DIFFERENTIAL EVOLUTION ALGORITHM IN MODELS OF TECHNICAL OPTIMIZATION

Roman Knobloch and Jaroslav Mlýnek
Department of Mathematics
Technical University of Liberec
Studentská 2, 46117 Liberec
The Czech Republic
E-mail: roman.knobloch@tul.cz, jaroslav.mlynek@tul.cz

KEYWORDS
Optimization, search space, cost function, differential evolution algorithm, global convergence, asymptotic convergence, mathematical model.

ABSTRACT

At present, evolutionary optimization algorithms are increasingly used in the development of new technological processes. Evolutionary algorithms often allow the optimization procedure to be performed even in cases where classical optimization algorithms fail (e.g. gradient methods) and where an acceptable solution is sufficient to solve the optimization task. The article focuses on possibilities of using a differential evolution algorithm in the optimization process. This algorithm is often referred to in the literature as a global optimization procedure. However, we show by means of a practical example that the convergence of the classic differential algorithm to the global extreme is not generally assured and is largely dependent on the specific cost function. To remove this weakness, we designed a modified version of the differential evolution algorithm. The improved version, named the modified differential evolution algorithm, is described in the article. It is possible to prove asymptotic convergence to the global minimum of the cost function for the modified version of the algorithm.

INTRODUCTION

New technological procedures are often developed using mathematical models describing the essential features of the solved problem. The model is then used to transform the real world problem into an optimization task. Strong assumptions are often required when using classic optimization methods (e.g. convexity of the searched space, convexity of the evaluation function, knowledge of the appropriate position of the initial solution). Otherwise, these methods do not often lead to the required solution.

Evolutionary optimization algorithms are primarily utilized in situations when other usual methods fail to converge to the optimized state. Recently, use of the evolutionary optimization algorithms has been considerably expanding, see e.g. (Simon 2013), (Affenzeller et al. 2009)). The evolutionary algorithms are in particular appropriate for problems with a complicated structure of the search space and in case of intricate cost functions. Evolutionary algorithms are in general more computationally demanding and they are therefore suitable for calculations that are not time limited (e.g. off-line calculations of trajectories of an industrial robot, see (Mlýnek et al. 2020)). Their use in time critical calculations is rather limited. For example, their utilization for online decision making processes (e.g. online calculations of trajectories of industrial robot depending on the evaluation of current conditions) is not so frequent. Nevertheless, parallel programming tools are often used to speed up calculations with good results. Nowadays, parallel programming tools form a part of most used programming languages.

The differential evolution algorithm is one of the frequently used algorithms for solving practical optimization tasks. This algorithm was first introduced by Storn and Price in (Storn and Price 1997) and (Price et al. 2005). This algorithm is often referred to as a global optimization method (see (Storn and Price 1997), (Price 1996)). However, such statements are always justified. We demonstrate by an example of a specific cost function that this algorithm is prone to premature local convergence and its convergence to the minimum of the cost function is not assured. The issue of suitable choice of optional algorithm parameters is solved, for example, in (Červenka and Boudná 2018). In this article we propose a suitable modification of the differential evolution algorithm that eliminates the premature convergence to a local minimum. Additionally, it is possible to prove asymptotic convergence to the global minimum of the cost function.

The differential evolution algorithms now constitute a larger group of similar algorithms that differ in implementation details. We concentrate on the standard *DE/rand/1/bin* algorithm which is best known and mostly used. That is why it is termed as the classic differential evolution algorithm in (Price at al. 2005). Hereafter it is referenced to as CDEA. The new proposed modification of CDEA is termed to as the modified differential algorithm denoted by abbreviation MDEA.

CLASSIC DIFFERENTIAL EVOLUTION ALGORITHM AND GLOBAL CONVERGENCE

In this part we briefly describe the operation of CDEA. Generally, CDEA seeks for the minimum of the cost function by constructing whole generations of

Communications of the ECMS, Volume 35, Issue 1,
Proceedings, ©ECMS Khalid Al-Begain, Mauro Iacono,
Lelio Campanile, Andrzej Bargiela (Editors)
ISBN: 978-3-937436-72-2 / 78-3-937436-73-9(CD) ISSN 2522-2414

individuals. Each individual is an ordered set of specific values corresponding to one point in the cost function domain. In this way each individual represents a potential solution to the optimization task. The quality of this individual is determined by the evaluation of the cost function corresponding to this individual. The next generation is formed from the existing generation by means of mutation and crossover operators. Specifically, we go successively through all individuals in the generation *G*. To each individual y_m^G (termed as the *target individual*) we select randomly three other (different) individuals y_{r1}^G, y_{r2}^G, y_{r3}^G from the current generation. We form in a specific way (including randomness) a combination of these three random individuals and the target individual. This combination is termed as the *trial individual* and denoted y_m^{trial}. Then we evaluate the cost function for the target y_m^G and trial individual y_m^{trial} and compare the results. The individual with lower value of the cost function advances to the position of the target individual of the next generation y_m^{G+1}. When this procedure is completed for all target individuals in generation *G*, we have constructed the new generation of individuals numbered $G + 1$.
The next part illustrates CDEA operation in a definite way in the form of pseudo code.

Input:
Optimization task parameters:
f denotes the cost function, *D* is the dimension of the cost function domain, $< x_{j\min}, x_{j\max} >$ is a domain of each cost function variable x_j.

CDEA parameters:
NP denotes the generation size (the number of individuals in each generation), *NG* is the number of calculated generations, *F* stands for mutation factor ($F \in < 0, 2 >$), and CR denotes the crossover probability ($CR \in < 0, 1 >$). The symbol *G* stands for the generation number, index *m* is the number of the individual in the generation, index *j* describes the *j*-th component of a specific individual y_m.

Computation:
1. Create an initial generation ($G = 0$) of *NP* individuals y_m^G, $1 \le m \le NP$, (e.g. by use of relation (1)).
2. a) Evaluate all individuals y_m^G of the *G*-th generation (calculate $f(y_m^G)$ for each individual y_m^G). b) Store the individuals y_m^G and their evaluations $F(y_m^G)$ into matrix **B** (each matrix row contains parameters of individual y_m^G and its evaluation $F(y_m^G)$. That is matrix be has *NP* rows and D+1 columns ($1 \le m \le NP$).
3. *while* $G \le NG$

a) *for* $m := 1$ *step* 1 to *NP* *do*
(i) randomly select index $s_m \in \{1, 2, \ldots, D\}$,
(ii) randomly select indexes $r_1, r_2, r_3 \in \{1, \ldots, NP\}$,
where $r_l \ne m$ for $1 \le l \le 3$;
$r_1 \ne r_2, r_1 \ne r_3, r_2 \ne r_3$;
(iii) *for* $j := 1$ *step* 1 to D *do*
if rand $(0,1) \le CR$ or $j = s_m$) *then*
$y_{m,j}^{trial} := y_{r_3,j}^G + F\left(y_{r_1,j}^G - y_{r_2,j}^G\right)$
else
$y_{m,j}^{trial} := y_{m,j}^k$
end if
end for (*j*)
(iv) *if* $f\left(y_m^{trial}\right) \le f\left(y_m^k\right)$ *then* $y_m^{G+1} := y_m^{trial}$
else $y_m^{G+1} := y_m^G$
end if
end for (*m*)

b) store individuals y_m^{G+1} and their evaluations $f\left(y_m^{G+1}\right)$ $\left(1 \le m \le NP\right)$ of the new $(G+1)$-st generation in the matrix **B**, $G := G + 1$
end while (*G*).

Output:
The row of matrix **B** that contains the corresponding value $\min\{F\left(y_m^G\right); y_m^G \in \mathbf{B}\}$ represents the best found individual y_{opt}.

Comments

The individual y_{opt} in pseudo-code of CDEA is the final solution of the optimization problem.

One way of possible forming the initial generation ($G = 0$) of individuals y_m^0 is given by relation

$$y_{m,j}^0 := x_{j\min} + \text{rand}\,(0, 1) \cdot \left(x_{j\max} - x_{j\min}\right). \qquad (1)$$

Values $x_{j\min}$ and $x_{j\max}$ are lower and upper limit of variable x_j. The function rand $(0, 1)$ randomly generates a value from a closed interval $< 0, 1 >$.

Counterexample to Global Convergence of CDEA (Premature Convergence)

It is not difficult to find counterexamples to the global convergence of the CDEA. Let us consider for instance the following two graphs of cost functions with the domain in Euclidean space R^2, see Figure 1. Even for the cost function shown in Figure 1 above the probability that the CDEA finds the global minimum of the cost function is less than one. The reason is that the CDEA

can converge in some cases relatively fast to the local minimum missing completely the global minimum. This results in concentrating the individuals in subsequent generations around the local minimum. As soon as the size of the generation falls under some critical value, the generation is too small to produce trial individuals that could hit the region in the vicinity of the global minimum. This situation is called a *premature convergence*. In this case even increasing the number of generations does not lead to increasing the chance to identify the global minimum. Moreover, the probability that the CDEA finds the global minimum falls with the decreasing measure of the global minimum region. The probability of finding the global minimum for the cost function in Figure 1 below is substantially smaller than for the cost function in Figure 1 above. Additionally, by a sufficient reduction of the measure of the global minimum region this probability can be made as close to zero as possible.

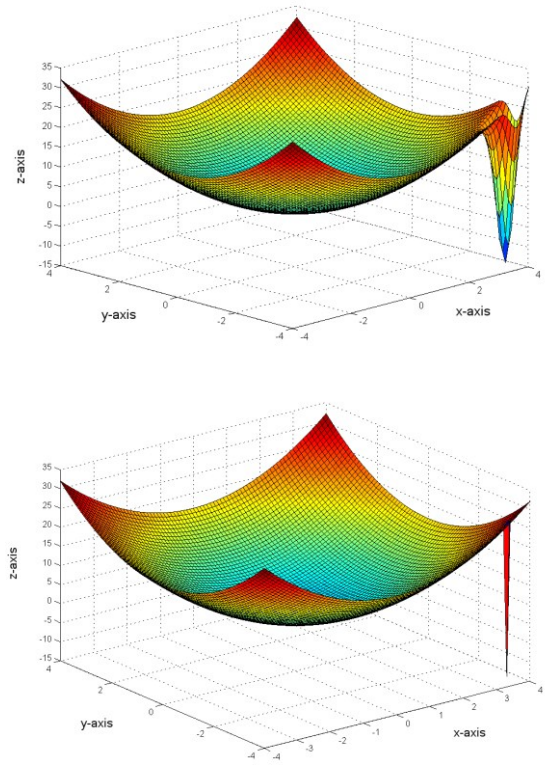

Figure 1: Examples of cost functions with domains in R^2

Numerical Example

We can present a specific cost function to demonstrate the limited ability of CDEA to converge to the global minimum of the cost function. To keep things simple we consider the domain of the cost function as a subset of the two dimensional Euclidean space R^2. We will construct the cost function $F(x_1, x_2)$ as a composition of two simple functions

$$F(x_1, x_2) = F_B(x_1, x_2) + F_M(x_1, x_2). \qquad (2)$$

The term $F_B(x_1, x_2)$ represents the base function. This function is smooth and has one shallow minimum. It can be defined for instance in the following way

$$F_B(x_1, x_2) = x_1^2 + x_2^2,$$

with the domain $D(F_B) = \langle -H, H\rangle \times \langle -H, H\rangle$, where H determines the boundary values of the domain.

The term $F_M(x_1, x_2)$ denotes a modifier function. This function should be relatively steep and with a rather small domain. We use the function $F_M(x_1, x_2)$ to modify the underlying base function $F_B(x_1, x_2)$. The role of the function $F_M(x_1, x_2)$ is to realize the global minimum of the cost function $F(x_1, x_2)$ in relation (2). To be able to construct the function $F_M(x_1, x_2)$ effectively, we introduce another auxiliary function F_P,

$$F_P(x_1, x_2) = x_1^2 + x_2^2 - 1,$$

with the domain $D(F_P) = \{x_1, x_2: x_1^2 + x_2^2 \leq 1\}$. It is obvious that the function $F_P(x_1, x_2)$ is defined exclusively on a unit circle and has values from the closed interval $\langle -1, 0\rangle$. The graph of the function $F_P(x_1, x_2)$ is a circular paraboloid presented in Figure 2.

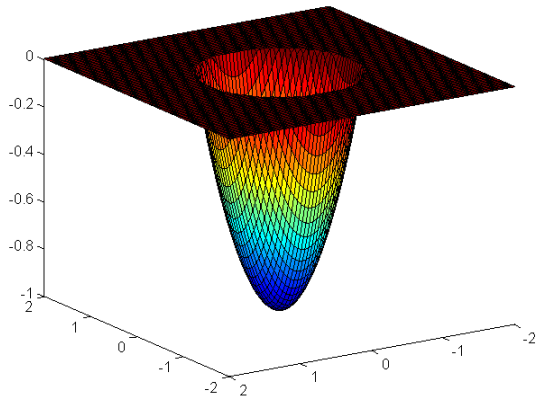

Figure 2: Graph of the auxiliary function $F_p(x_1, x_2)$

The function $F_M(x_1, x_2)$ is then formed as

$$F_M(x_1, x_2) = \lambda_h \cdot F_P\left(\frac{1}{\rho}(x_1 - x_{G1}), \frac{1}{\rho}(x_2 - x_{G2})\right).$$

Here the number λ_h defines the height of the resulting circular paraboloid, ρ denotes the radius of the domain on which the modifier function $F_M(x_1, x_2)$ is defined. Obviously, the modifier function $F_M(x_1, x_2)$ is defined only for points that are closer to the point $[x_{G1}, x_{G2}]$ than the radius of its domain ρ. The coordinates x_{G1}, x_{G2} specify the point, where the modifier function $F_M(x_1, x_2)$ attains its minimum. The overall cost function $F(x_1, x_2)$ is then defined according to the relation (2) by the composite formula

$$F(x_1, x_2) = x_1^2 + x_2^2 + \lambda_h \cdot F_P\left(\frac{1}{\rho}(x_1 - x_{G1}), \frac{1}{\rho}(x_2 - x_{G2})\right) \qquad (3)$$

We have to choose the parameters λ_h, ρ, x_{G1} and x_{G2} in a reasonable way to obtain the required result. It is clear that we can control the dimensions of the modifier function domain by parameter ρ. The point $[x_{G1}, x_{G2}]$ is to be placed relatively close to the boundary of the cost function domain. This means it is relatively far from the point [0, 0] representing the local minimum of the cost function analogously to the cost functions presented in Figure 1. Since the base function $F_B(x_1, x_2)$ is positive definite, it attains a positive value $F_B(x_{G1}, x_{G2})$ at the point $[x_{G1}, x_{G2}]$. This means we have to take the parameter λ_h sufficiently large, so that the global minimum is essentially lower than the local minimum at the point [0, 0].

We performed numerical experiments with following parameters: $D(F_B) = \langle -H, H\rangle \times \langle -H, H\rangle = \langle -4, 4\rangle \times \langle -4, 4\rangle$ with measure $\mu(D(F)) = 8^2 = 64$, $x_{G1} = x_{G2} = 3$. This means that the global minimum of the cost function F is at point [3,3] and local minimum at point [0,0]. Number of individuals in each generation $NP = 200$, number of generation $NG = 160$, value of parameter $F = 0.8$ and value of parameter $CR = 0.9$. We realized 200 numerical experiments with value of parameter $\rho = \frac{1}{10}$ (then $\mu(D(F_M)) = \pi\rho^2 \doteq 0.0314$) and 200 numerical experiments with value of parameter $\rho = \frac{1}{16}$ (then $\mu(D(F_M)) \doteq 0.01223$). The results illustrating the limited ability of CDEA to identify the global minimum are summarized in Table 1.

Table 1: Experimental testing of CDEA

CDEA	Local minimum hits	Global minimum hits	Success rate in %
$\rho = \frac{1}{10}$	163	37	18.5
$\rho = \frac{1}{16}$	185	15	7.5

Based on the values given in Table 1, it can be assumed that the decreasing value of ρ (and thus value of $\mu(D(F_M))$) will significantly decrease the success rate of the CDEA algorithm in finding the global minimum.

MODIFIED DIFFERENTIAL EVOLUTION ALGORITHM

As illustrated in the previous part, CDEA does not in general guarantee the convergence to the global minimum of the cost function. This is caused by the too fast convergence of CDEA to the local minimum (premature convergence) resulting in rapid reduction of the generation size (which means a loss of diversity). The most straightforward way how to limit the premature convergence is to replace some individuals with the highest values of the cost function by random individuals in each generation. Though these random individuals reduce partially the convergence speed, they increase substantially the diversity of the generation. In technical terms, it is necessary to make one simple change in the CDEA scheme. We present only the differences with respect to CDEA. See the pseudocode description of CDEA in chapter "Classic Differential Evolution Algorithm and Global Convergence".

Input:

We add another parameter R that determines the ratio of random individuals in each generation, $R \in < 0, 1 >$, e.g., $R = 0.1$ means that 10% of individuals in each generation are generated randomly.

Computation:

We add another procedure to the part 3., specifically:

c) determine in matrix B the quantity $\lfloor NP.R \rfloor$ of individuals with the highest cost function values and replace these individuals by randomly generated individuals (e.g. by use of relation (1)) from the search space). Note that here the symbol $\lfloor x \rfloor$ denotes the integer part of the real number x.

This modified algorithm will be called the Modified Differential Evolution Algorithm (MDEA). We applied the numerical experiments on MDEA with the same input parameters as in the previous chapter on CDEA. In addition, the value of R parameter is equal to $R = 0.1$. The results are summarized in Table 2.

Table 2: Experimental testing of MDEA

MDEA	Local minimum hits	Global minimum hits	Success rate in %
$\rho = \frac{1}{10}$	34	166	83.0
$\rho = \frac{1}{16}$	70	130	65.0

Another positive feature of the algorithm MDEA is that if we increase the number of generations NG the global minimum will be identified with an increased probability. This probability can come close to 1 for a sufficiently high number G of generations. We call this aspect of the MDEA an *asymptotic global convergence*. We describe this topic in the following chapter.

ASYMPTOTIC GLOBAL CONVERGENCE OF MDEA

In this part we present several theoretical concepts and statements that can be used to prove the asymptotic global convergence of MDEA. More specifically, we will show that when the number of generations $G \to \infty$ then the probability that MDEA identifies the global minimum of the cost function approaches 1.

Optimal Solution Set
We would like to find the minimum of the cost function with the lowest value

$$\min\{F(x): x \in S\}, \quad (4)$$

where S is a measurable search space of a finite measure representing all possible configurations of variables $x = (x_1, x_2, \dots x_n)$. We suppose that the global minimum of function F exists on S. We define a solution set S^* as

$$S^* = \{x^*: F(x^*) = \min\{F(x): x \in S\}\},$$

where x^* represent global minima of the function F. We introduce an optimal solution set S_ε^* as

$$S_\varepsilon^* = \{x \in S: |F(x) - F(x^*)| < \varepsilon\},$$

where $\varepsilon > 0$ is a small positive real number. Denoting by μ the Lebessgue measure, we suppose that for each ε it holds $\mu(S_\varepsilon^*) > 0$.

Convergence in Probability
To examine the global convergence of MDEA we need to introduce a concept of the convergence in probability defined in (Hu et al. 2013).

Definition: Let $\{G(k), k = 1,2, \dots\}$ be a generation sequence created by a differential evolution algorithm to solve optimization task (4). We say that the algorithm converges to the optimal solution set in probability if

$$\lim_{k \to \infty} p\{G(k) \cap S_\varepsilon^* \neq \emptyset\} = 1, \quad (5)$$

where p denotes the probability of an event.

Now, we can use this concept to formulate the following statement.

Proposition: Let us suppose that for each generation $G(k)$ of a differential evolution algorithm there exists at least one individual y such that

$$p\{y \in S_\varepsilon^*\} \geq \alpha > 0,$$

where α is a small positive value. Then the algorithm converges to the optimal set S_ε^* in probability. That is relation (5) holds.

The proof of this proposition is stated in full in (Knobloch et al. 2017).

It holds that for each generation G of the MDEA it is true

$$p\{y \in S_\varepsilon^*\} = \alpha \geq 0, \quad (6)$$

where α is a small positive value. Here $p\{y \in S_\varepsilon^*\}$ denotes the probability that y belongs to S_ε^*. The validity of relation (6) necessarily results from the generation of random individuals in each generation G of MDEA. It follows that MDEA converges to $y \in S_\varepsilon^*$ for any small real positive number ε. This implies the asymptotic global convergence of MDEA. Thus, we know that MDEA converges to the global minimum. The asymptotic convergence of MDEA is proved in detail in (Knobloch et al. 2017), see also (Hu et al. 2013). Probability estimates of reaching the global minimum after performing G generations of MDEA are given in (Knobloch and Mlýnek 2020). These estimates help to decide after how many generations to finish the MDEA calculation.

CONCLUSIONS

CDEA is a universal optimization algorithm that is frequently used in technical projects, economy studies, natural sciences and other important areas of interest. Nevertheless, it has some principal limitations. The main weakness of CDEA is a possible premature convergence of the computing process to a local minimum of the cost function. We demonstrated this fact by means of a simple example.
Identification of this weakness was the starting point for a search of an improved version of the algorithm that would provide better chances regarding the convergence to the global minimum of the cost function. MDEA is a result of these efforts.
MDEA is not prone to the premature convergence because a certain ratio of random individuals in each generation makes it immune to the loss of generation diversity. From the theoretical point of view, we proved that MDEA converges asymptotically to the global minimum of the cost function in probabilistic sense.
The use of MDEA has proved successful to the authors in solving complicated practical optimization problems. For example, it is the task of optimizing the placement of infrared heaters over a metal thin walled mould in the production of artificial leather for the automotive industry (Slush Moulding technology).
The cost function of this optimization problem is a function of many variables (often 300 and more) and has many local minima. Gradient methods, genetic algorithms and also CDEA found only a local minimum of the corresponding cost function (this optimization problem is described in more detail in (Mlýnek and Knobloch 2018) and (Mlýnek et al. 2016). MDEA has also proved successful in optimizing the fibre winding procedures using a fibre-processing head and a non-bearing frame moved by an industrial robot (for more details see (Mlýnek et al. 2020)).

ACKNOWLEDGMENTS

This article was supported by project "Modular platform for autonomous chassis of specialized electric vehicles for freight and equipment transportation", Reg. No. CZ.02.1.01/0.0/0.0/16_025/0007293.

REFERENCES

Affenzeller, M.; Winkler, S.; Wagner, S.; and A. Beham. 2009. "Genetic Algorithms and Genetic Programming." CRC Press, Boca Raton.

Červenka, M. and H. Boudná. 2018. "Visual Guide of F and CR Parameters Influence on Differential Evolution Solution Quality." *Proceedings of 24th International Conference Engineering Mechanics 2018*, Svratka, Czech Republic, 141-144, DOI: 10.21495/91-8-141.

Hu, Z.; S. Xiong; Q. Su; and X. Deng. 2013. "Sufficient Conditions for Global Convergence of Differential Evolution Algorithm." *Journal of Applied Mathematics*, Article ID 139196.

Knobloch, R.; J. Mlýnek; and R. Srb. 2017. "The Classic Differential Evolution Algorithm and Its Convergence Properties". *Applications of Mathematics*, Vol. 62, No. 2, 197-208.

Knobloch, R. and J. Mlýnek. 2020. "Probabilistic Analysis of the Convergence of the Differential Evolution Algorithm." *J. Neural Network World*, Vol. 30, 249-263, DOI: 10.14311/NNW.2020.30.017.

Mlýnek, J.; R. Knobloch; and R. Srb. 2016. "Optimization of a Heat Radiation Intensity and Temperature Field on the Mould Surface." *Proceedings of 30th European Conference on Modelling and Simulation*, Regensburg, Germany, ISBN: 978-0-9932440-2-5, DOI: 10.7148/2016-0425.

Mlýnek, J. and R. Knobloch. 2018. "Model of shell Metal Mould Heating in the Automotive Industry." *Applications of Mathematics*, Vol. 63, No. 2, 111-124.

Mlýnek, J.; M. Petrů; T. Martinec, T. ; and S.S.R. Koloor. 2020. "Fabrication of High-Quality Polymer Composite Frame by a New Method of Fiber Winding Process." *J. Polymers*, Volume 12(5), 30 pages, https://doi.org/10.3390/polym12051037, Open Access.

Price, K.V. 1996. "Differential Evolution - A Fast and Simple Numerical Optimizer." *Proceedings of North American Fuzzy Information Processing. Berkeley*, 524-527.

Price, K.V.; R.M. Storn ; and J.A. Lampien. 2005. "Differential Evolution, A Practical Approach to Global Optimization." Springer-Verlag, Berlin Heidelberg.

Simon, D. 2013. "Evolutionary Optimization Algorithms." John Wiley & Sons, Hoboken, New Jersey.

Storn, R.M. and K.V. Price. 1997. "Differential Evolution - A Simple and Efficient Heuristics for Global Optimization over Continuous Spaces." *Journal of Global Optimization*. Kluwer Academic Publishers, 11, 341-359.

AUTHOR BIOGRAPHIES

ROMAN KNOBLOCH was born in Turnov, the Czech Republic. He finished his studies at the Charles University in Prague, the Faculty of Mathematics and Physics, where he studied physics and teaching of mathematics and physics. His main areas of interest are: modelling of physical phenomena, modern optimization methods and heat and transport phenomena in continuum mechanics. He works as an assistant professor at the Technical University of Liberec where he also graduated his PhD study programme. His e-mail address is roman.knobloch@tul.cz

JAROSLAV MLÝNEK was born in Trnava, Czechoslovakia and went to the Charles University in Prague, where he studied numerical mathematics at the Faculty of Mathematics and Physics and he graduated in 1981. In his work he focuses on the computational problems of heating and thermal losses in components of electrical machines and on optimization procedures. Currently he works as an associate professor at the Technical University of Liberec. His e-mail address is: jaroslav.mlynek@tul.cz