An improved multistart based method for global optimization problems

Ioannis G. Tsoulos[*], Alexandros Tzallas

Department of Informatics and Telecommunications, University of Ioannina, Greece

Abstract

There is a great need to discover the global optimum of multimodal functions that arises from many scientifical and practical problems. A base method used to solve this problem is the Multistart method. The proposed method extends the traditional method by adding two major features: the first one is a rejection mechanism to prevent the method from spending time in unnecessary function evaluations and the second is a stopping rule aimed to terminate the method when the global minimum has been located with some certainty. The proposed method has been evaluated on a series of global optimization functions from the relevant literature and it is compared against some other optimization methods and the experimental results are presented.

Keywords: Global optimization, stochastic methods, termination rules.

1 Introduction

A novel method for the task of locating the global minimum of a continuous and differentiable function $f:S\rightarrow R, S\subset R^{n}$ is introduced here. The task of locating the global optimum can be formulated as, determine

$$x^{*}=\arg\min_{x\in S}f(x) \tag{1}$$

with S:

$$S=[a_{1},b_{1}]\otimes[a_{2},b_{2}]\otimes\ldots[a_{n},b_{n}]$$

Methods that discover the global minimum can be used in many areas such as: economics [1, 2], physics [3, 4], chemistry [5, 6], medicine [7, 8] etc. Global optimization methods usually are divided into two main categories: deterministic and stochastic methods. The most common methods of the first category are the

[*] Corresponding author. Email: itsoulos@uoi.gr

so called Interval methods [9, 10], where the set S is divided iteratively in subregions and some subregions that not contain the global solution are discarded using some pre defined criteria. On the other hand, in the second category there are Controlled Random Search methods [11, 12, 13], Simulated Annealing methods [14, 15, 16], Differential Evolution methods [17, 18], Particle Swarm Optimization methods [19, 20, 21], Ant Colony Optimization [22, 23], Genetic algorithms [24, 25, 26] etc. Also many hybrid methods have been proposed such as combing the Simplex method and Inductive search [27], methods that using PSO in conjuction with other methods [28, 29], etc. Recently many works have been appeared that take advantage of the GPU processing units to implement parallel global optimization methods [30, 31, 32].

This article introduces a novel method which is based on the multistart method to discover the global minimum of continuous functions. The method incorporates an efficient stopping rule as well as an asymptotic criterion to prevent the algorithm from unnecessary local optimization calls. The multistart method is one of the simplest global optimization technique which starts a local optimization method from different random samples and yields the lowest discovered minimum as the global one. Due to its simplicity the method has been used in many problems such as the TSP problem [33], the vehicle routing problem [34], the facility location problem [35], the maximum clique problem [36] etc. During the past year, the method multistart has been enhanced in the relevant literature with methods that target to discover all the local minima of a function [38, 39, 40], hybrid multistart techniques [41, 42], GRASP methods[43], new stopping rules [44, 45, 46], parallel techniques[47, 48] etc.

The rest of this article is organized as follows: in section 2 the proposed method is described in detail, in section 3 the experimental results are demonstrated and finally in section 4 some conclusions and guidelines for future work are provided.

2 Method description

At every iteration of the proposed method a number of samples is taken in the bounding box of the objective problem. Some of them are considered as starting points for a local search procedure and the rest are discarded. The method continues until the maximum number of iterations is reached or an asymptotic termination rule, defined in subsection 2.2 is satisfied. The main steps of the proposed algorithm are outlined in Algorithm 1. In the following subsection the main parts of the proposed algorithm which are the discarding procedure and the proposed stopping rule are described in detail.

2.1 Discarding procedure

The discarding procedure has two major elements:

- The first is the typical distance that is calculated after every local search

Algorithm 1 The main steps of the proposed algorithm.

```
1. **Initialization Step**
    (a) **Set** $K$, the maximum number of allowed iterations.
    (b) **Set** $N$, the number points that will be samples at each iteration.
    (c) **Set** $r_C = 0$, the distance for the gradient check algorithm.
    (d) **Set** $X^* = \emptyset$, the set of local minima discovered by the local search
        procedure.
2. **Main Step**
    (a) **For** $i = 1..N$ **do**
        i. **Sample** randomly a point $x \in$S.
        ii. **Check** if $x$ is a valid starting point for the local search procedure
            using the method gradientCheck(x) given in algorithm 2.
        iii. **If** gradientCheck(x)=false **then**
            A. **Start** a local search procedure $y = L(x)$
            B. **Update** the distance $r_C$ using the equation 2.
            C. If $x \notin X^*$ then $X^* = X^* \cup x$
        iv. **End if**
    (b) **End For**
3. **Termination Check Step.** Check the termination rule as described in
   subsection 2.2.
    (a) **If** the termination rule holds **then** terminate
    (b) **else goto** 2
    (c) **End if**
```

Algorithm 2 The procedure gradientCheck(x).

```
**boolean** gradientCheck(x)
    1. **Set** $d = \min_{y \in X^*} \|y - x\|$
    2. **Set** $z = \arg\min_{y \in X^*} \|y - x\|$
    3. **If** $d < r_C$ AND $(x - z)^T (\nabla f(x) - \nabla f(z)) > 0$ **then return** true
    4. **else return** false
**end** gradientCheck
```

and it is given by

$$r_C = \frac{1}{M} \sum_{i=1}^{M} \|x_i - x_{iL}\| \tag{2}$$

where the local search procedure $L(x)$ initiates from x_i and x_{iL} is the outcome of $L\left(x_i\right)$. If a point x is close enough to an already discovered local minima then it is highly possible that the point belongs to the so called region of attraction of the minima. The region of attraction of a local minimum z is defined as:

$$A\left(z\right) = \left\{x : x \in S,\ L(x) = z\right\} \tag{3}$$

- The second element is a gradient check performed between a candidate starting point and an already discovered local minimum. The value of the objective function $f(x)$ near to an already discovered local minimum can be calculated using:

  $$f(x) \simeq f(z) + \frac{1}{2}\left(x - z\right)^T B\left(x - z\right) \tag{4}$$

  where B is the Hessian matrix at the minimum z. By taking gradients in both sides of Equation 4 we obtain:

  $$\nabla f(x) \simeq B\left(x - z\right) \tag{5}$$

  Of course equation 5 holds for any other point y near to z

  $$\nabla f(y) \simeq B\left(y - z\right) \tag{6}$$

  By subtracting the equation 6 from 5 and by multiplying with $\left(x - y\right)^T$ we have the following equation:

  $$\left(x - y\right)^T \left(\nabla f(x) - \nabla f(y)\right) \simeq \left(x - y\right)^T B\left(x - y\right)^T > 0 \tag{7}$$

The distance r_C is used with conjunction with equation 7 to reject a point. The method rejects points from being start points if they are close enough to some other located local minima and in the same time equation 7 holds. A measure to identify if a point is close enough to some other located local minima is the distance r_C calculated after every local minimization.

In order to measure the efficiency of the above rejection procedure two tests were performed on two benchmark functions: Rastrigin and Exp16. In these test the proportion of local search starts over total number of samples were measured. The plot of this measurement for 20 iterations is plotted in Figures 1 and 2 respectively.

Figure 1: Plot for Rastrigin function and the discarding procedure.

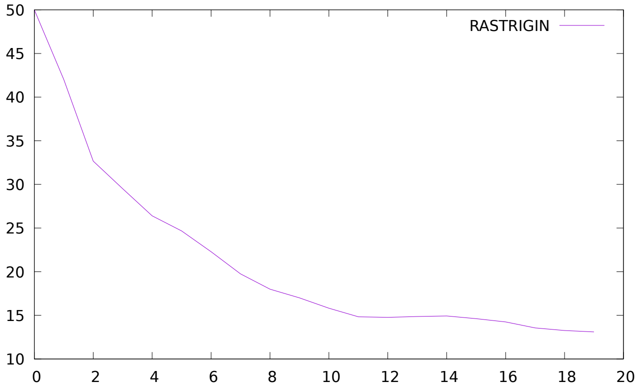


Figure 2: Plot for the function Exp16 and the discarding procedure.

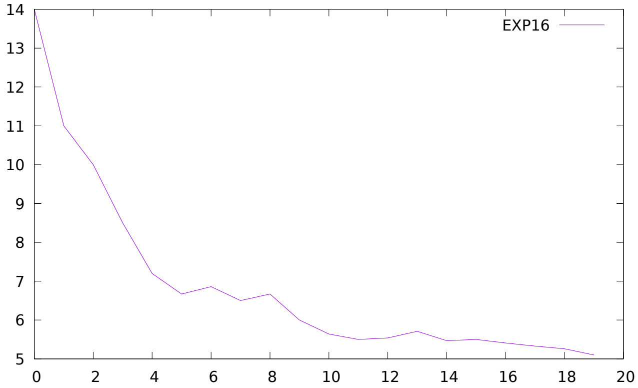
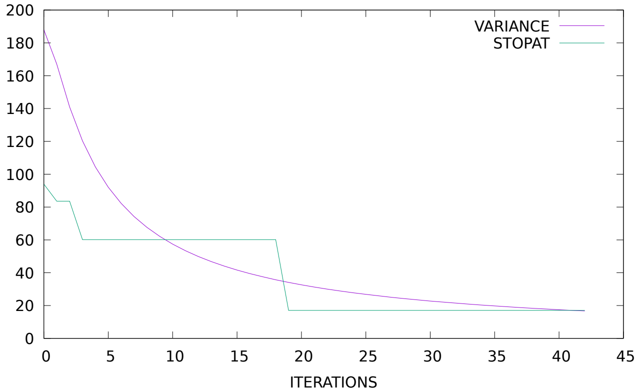


Figure 3: Plot of variance along with the stopping quantity for the problem of Potential with 20 atoms.



2.2 Stopping rule

A common way to terminate a global optimization procedure is to use the maximum number of allowed iterations, i.e. stop when iter $\geq K$. Although, it is a simple criterion but is not an efficient one since, if K is tool small, then the algorithm will terminate without locating the global optimum. Also, when K is too high, the optimization algorithm will spend computation time in unnecessary function calls. The termination rule used here is derived from [49]: denote with $\sigma^{(k)}$ the variance of $f(x^*)$, where k is the current iteration and x^* is the so far located global minimum. If the algorithm did not manage to locate new minimum for a number of generations, then probably the algorithm has located the global minimum and it should terminate. The termination rule stops the algorithm when

$$k \geq k_{\mathrm{min}} \text{ AND } \sigma^{(k)} \leq \frac{\sigma^{\left(k_{\mathrm{last}}\right)}}{2} \qquad (8)$$

where k_{last} stands for iteration where a new minimum was found. The value k_{min} is a predefined minimum number of iterations, in order to prevent the algorithm from premature termination. In figure 3 the values $\sigma^{(k)}$ denoted as VARIANCE and the value $\frac{\sigma^{\left(k_{\mathrm{last}}\right)}}{2}$ denoted as STOPAT are plotted. The objective function used is the EXP8 function. The value of k_{min} is set to 20 iterations and the maximum number of iterations is 200. The algorithm terminates successfully in 40 generation without spending unnecessary function calls for about 160 generations.

2.3 Constrained optimization problems

The proposed method can also be used to constrained optimization problems, which general formulation is:

$$\begin{aligned} \min_x \quad & f(x) \quad \text{subject to} \\ & g_i(x) \leq 0 \quad i = 1, \ldots, m \\ & h_j(x) = 0 \quad j = 1, \ldots, p \end{aligned} \tag{9}$$

where $x_i \in [a_i, b_i]$, $i = 1, \ldots, n$, which finds application in many practical fields such as physics [56], astronomy [57], chemistry [58], biology [59] etc. Many methods have been proposed to tackle this problem such as Lagrange multiplier methods [60], Trust region methods [61], interval methods [62], methods based on differential evolution [63] etc. The constrained optimization problem can transformed to a single function for the proposed method. The transformation is performed in a way that when a solution is infeasible then the function value is penalized according to the constraint violations. The steps for the transformation are the following:

1. **Set** $v_1(x) = f(x)$
2. **Set** $v_2(x) = \sum_{i=1}^{p} h_i^2(x)$
3. **Calculate** for the inequality constraints $g_i(x),\ i = 1, ..., m$ the quantity

   $$v_3(x) = \sum_{i=1}^{m} G_i^2\left(g_i(x)\right) \tag{10}$$

   where the function $G(x)$ is defined as follows:

   $$G(x) = \begin{cases} 0, & x \leq 0 \\ x, & x > 0 \end{cases} \tag{11}$$

4. The transformed objective function for the proposed method is given by:

   $$v(x) = v_1(x) + \lambda v_2(x) + \lambda v_3(x) \tag{12}$$

   where $\lambda > 0$.

3 Experiments

The proposed method was tested on a series of unconstrained and constrained optimization functions that are listed in the Appendix. The local search procedure in all experiments and methods was a BFGS variant of Powell[50]. The method was tested against four methods from the relevant literature:

1. Multistart method. This is simple method, who samples a series of points each time (the same number as the proposed method) and performs a local search procedure from every sample. In order to be the results comparable against the proposed method the same stopping rule was used also in the Multistart method. The code was implemented in ANSI C++.

2. Controlled Random Search denoted as CRS in the results. The implementation used here is based on the algorithm of Price[11] made in Ansi C++. The method uses the suggested $N = 25n$ from price value for the initial number of samples in the set. Also, the implementation utilizes the application of the Local search procedure after the termination of the algorithm to enhance the located global optimum.

3. Simulated Annealing, denoted as SA in the results. The used software here is the work of Corana[51].

4. Particle swarm optimization[19], denoted as PSO in the results. The implemented software here is coded in ANSI C++ and utilizes the same stopping rule as in the proposed method.

The experimental results are reported in Table 1. The number in the cells denotes average function calls for 30 runs using with seed for the random generator each time. The fraction in parentheses stands for the fraction of runs where the global optimum was found. If this number is missing then the global minimum was discovered in every independent run (100% success). The parameters used in the experiments are listed in Table 2. The bold letters were used to demonstrate the method that achieved the best average result. The proposed method seems to require less number of function calls than the other methods and the PSO technique shown to work well for small problems like BF1 or BRANIN but failed to work on bigger problems such Potential20, a problem with dimension 60 and thousands of local minima. On the other hand, the Simulated Annealing managed to solve with success all the optimization problems, although it requires more function calls than the proposed method and the same holds for the simple Multistart method.

In Table 3 the results from the comparison of Multistart and the proposed method for the constrained optimization test problems are listed. Again, the proposed method has managed to handle the test problems using less number of function evaluations than the Multistart method. The value of $\lambda = 100$ was used for the conducted experiments. Again, the proposed method overcomes the Multistart method to solve constrained optimization problems.

Also, in order to measure the efficiency of the proposed method for as the dimension of the objective functions increases an additional experiments was conducted: The function Exponential was used with different values of the dimension n from 2 to 20. The proposed method is tested against Multistart and the results are plotted in Figure 4. The average function calls required by the proposed method are in the range [4000,6000] when the Multistart requires 5 or 6 times more function calls.

Figure 4: Average number of function calls for the function Exponential.

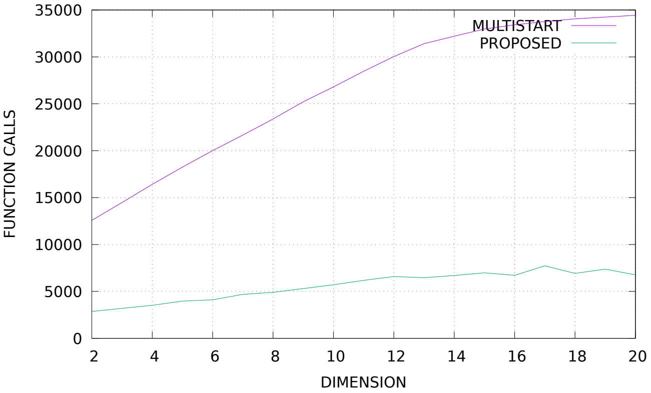
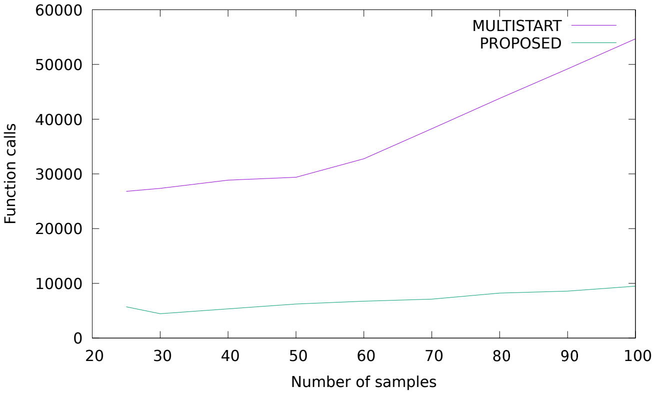


Figure 5: Average number of function calls as the number of samples increases for the function EXP10.



Additionally another experiment was conducted using the Exponential function with $n = 10$ with different values for the number of samples N and the results are plotted in Figure 5. Again the Multistart requires much more function calls than the proposed method and also the Multistart function calls tends to increase very rapidly as compared to the calls of the proposed method.

To compare the proposed method with all the rest methods for different function calls, the Wilcoxon signed-rank test was used. The results obtained with this statistical test are shown in Figure 6. The results indicate that the proposed method is superior than the most of unconstrained optimization function (multistart, CSA and SA). There was not, however, a significant difference between PSO and the proposed method.

Table 1: Average number of function calls for the proposed functions.

PROBLEM	MULTISTART	CRS	SA	PSO	PROPOSED
BF1	22533	2218	3845	**2494**	2833
BF2	18809	2207	3340	2641	**2629**
BRANIN	9735	1744	4816	**1636**	1753
CM4	27037	4746	9652	2988	**2293**
CAMEL	13688	1882	4820	**1639**	1732
DIFFPOWER10	1194776	78634	25918	15500(0.87)	**19572**
EASOM	5372	588	4807	866	**199**
EXP8	12022	13239	19233	3084	**2830**
EXP32	18294	93520	76842	5055	**3265**
GKLS250	17333(0.77)	1633	4120	**1459**	2415
GKLS350	10104	3329	7229	2259(0.97)	**243**
GRIEWANK2	13003	2111	3830	2595(0.83)	**1786**
GRIEWANK10	53372	32037	24118	8378(0.25)	**7184**
HANSEN	15294	3348	3323	4284	**1510**
HARTMAN3	14815	2898	7227	**2448**	11463
HARTMAN6	19459	9276	14440	4645	**3740**
POTENTIAL5	111631	95027	36084	**20100**	49601
POTENTIAL10	208405	193066	172166	19610(0.13)	**91094**
POTENTIAL20	280575	189591(0.53)	244314	18466(0.03)	**170524(0.97)**
RASTRIGIN	16968	1906	3343	2258	**675**
SHEKEL5	19224	6345	9635	4791	**3465**
SHEKEL7	20985	6528	9334	5722	**2976**
SHEKEL10	20284	6477	9998	5354	**3566**
SINU8	21860	16950	19241	5398	**549**
SINU32	39905	100887	13858	6887	**1296**
TEST2n4	15938	6754	9631	5219	**2890**
TEST2n5	18085	12717	12036	7672	**3262**
TEST2n6	19879	12822	14438	8039	**3451**
TEST2n7	21432	18620	16840	8220	**4002**
TEST30n3	24450	2768	9616	**2456**	10818
TEST30n4	26514	**3894**	10617	4528	13320

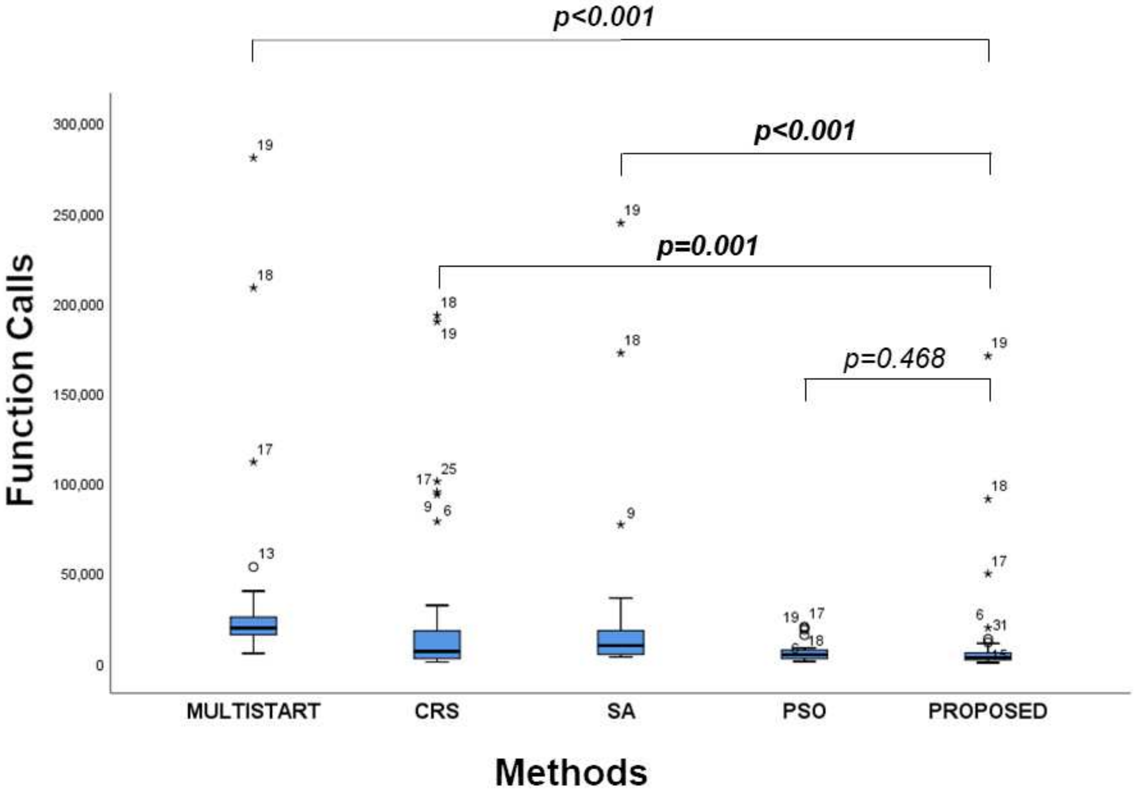
Table 2: Parameter values for the experiments.

PARAMETER	VALUE
K	200
N	25
k_{min}	20

Table 3: Average function calls from comparisons of the proposed method and Multistart for the constrained optimization problems.

PROBLEM	MULTISTART	PROPOSED
Levy	17491	1301
Salkin	48816	1010
Hess	27775	9524
Chootinan1	293459	15035
G15	318162	63542

Figure 6: Box plot representation and Wilcoxon rank-sum test results of the comparison between the multistart, CRS, SA, and PSO method and the proposed method. A p-value of less than 0.05 (2-tailed) was used for statistical significance and has been marked with bold.



4 Conclusions

A novel multistart based method is described and tested here for global optimization problems. The method utilizes an efficient discarding procedure to prevent the method from unnecessary function calls and an asymptotic stopping rule to stop the algorithm where there is a good probability that the global optimum has been discovered. The method was tested on a series of well known optimization problems from the relevant literature and proved to be efficient and fast.

Appendix - Test functions

In order to measure the effectiveness of the proposed approach we utilize several benchmark functions from the relevant literature [52, 53].

Unconstrained optimization functions

- **Bf1** function. The function Bohachevsky 1 is given by the equation

$$f(x)=x_1^2+2x_2^2-\frac{3}{10}\cos\left(3\pi x_1\right)-\frac{4}{10}\cos\left(4\pi x_2\right)+\frac{7}{10}$$

with $x\in[-100,100]^2$. The value of global minimum is 0.0.

- **Bf2** function. The function Bohachevsky 2 is given by the equation

$$f(x)=x_1^2+2x_2^2-\frac{3}{10}\cos\left(3\pi x_1\right)\cos\left(4\pi x_2\right)+\frac{3}{10}$$

  with $x\in[-50,50]^2$. The value of the global minimum is 0.0.

- **Branin** function. The function is defined by $f(x)=\left(x_2-\frac{5.1}{4\pi^2}x_1^2+\frac{5}{\pi}x_1-6\right)^2+10\left(1-\frac{1}{8\pi}\right)\cos(x_1)+10$ with $-5\le x_1\le 10,\ 0\le x_2\le 15$. The value of global minimum is 0.397887.with $x\in[-10,10]^2$. The value of global minimum is -0.352386.

- **CM** function. The Cosine Mixture function is given by the equation

$$f(x)=\sum_{i=1}^{n}x_i^2-\frac{1}{10}\sum_{i=1}^{n}\cos\left(5\pi x_i\right)$$

  with $x\in[-1,1]^n$. The value of the global minimum is -0.4 and in our experiments we have used $n=4$.

- **Camel** function. The function is given by

$$f(x)=4x_1^2-2.1x_1^4+\frac{1}{3}x_1^6+x_1x_2-4x_2^2+4x_2^4,\quad x\in[-5,5]^2$$

  The global minimum has the value of $f\left(x^*\right)=-1.0316$

- **DiffPower** function. The Sum of Different Powers function is defined

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

and the global minimum is is $f\left(x^*\right)=0$. The value $n=10$ was used in the conducted experiments and the associated function is denoted as Diffpower10.

- **Easom** function. The function is given by the equation

$$f(x)=-\cos\left(x_1\right)\cos\left(x_2\right)\exp\left(\left(x_2-\pi\right)^2-\left(x_1-\pi\right)^2\right)$$

with $x\in[-100,100]^2$. The value of the global minimum is -1.0

- **Exponential** function. The function is given by

$$f(x)=-\exp\left(-0.5\sum_{i=1}^{n}x_i^2\right),\quad -1\le x_i\le 1$$

The global minimum is located at $x^*=(0,0,...,0)$ with value -1. In our experiments we used this function with $n=8,32$ and the corresponding functions are denoted by the labels EXP8,EXP32.

- **Griewank2** function. The function is given by

$$f(x)=1+\frac{1}{200}\sum_{i=1}^{2}x_i^2-\prod_{i=1}^{2}\frac{\cos(x_i)}{\sqrt{(i)}},\quad x\in[-100,100]^2$$

The global minimum is located at the $x^*=(0,0,...,0)$ with value 0.

- **Griewank10** function. The function is given by the equation

$$f(x)=\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n}\cos\left(\frac{x_i}{\sqrt{i}}\right)+1$$

In our experiments we have used $n=10$ and the global minimum is 0.0 The function has several local minima in the specified range.

- **Gkls** function. $f(x)=\text{Gkls}(x,n,w)$, is a function with w local minima, described in [54] with $x\in[-1,1]^n$ and n a positive integer between 2 and 100. The value of the global minimum is -1 and in our experiments we have used $n=2,3$ and $w=50$. The corresponding functions are denoted by the labels GKLS250 and GKLS350.

- **Hansen** function. $f(x)=\sum_{i=1}^{5}i\cos\left[(i-1)x_1+i\right]\sum_{j=1}^{5}j\cos\left[(j+1)x_2+j\right]$, $x\in[-10,10]^2$. The global minimum of the function is -176.541793.

- **Hartman 3** function. The function is given by

$$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{3} a_{ij}\left(x_j - p_{ij}\right)^2\right)$$

with $x \in [0,1]^3$ and $a = \begin{pmatrix} 3 & 10 & 30 \\ 0.1 & 10 & 35 \\ 3 & 10 & 30 \\ 0.1 & 10 & 35 \end{pmatrix}$, $c = \begin{pmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{pmatrix}$ and

$$p = \begin{pmatrix} 0.3689 & 0.117 & 0.2673 \\ 0.4699 & 0.4387 & 0.747 \\ 0.1091 & 0.8732 & 0.5547 \\ 0.03815 & 0.5743 & 0.8828 \end{pmatrix}$$

The value of global minimum is -3.862782.

- **Hartman 6** function.

$$f(x) = -\sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} a_{ij}\left(x_j - p_{ij}\right)^2\right)$$

with $x \in [0,1]^6$ and $a = \begin{pmatrix} 10 & 3 & 17 & 3.5 & 1.7 & 8 \\ 0.05 & 10 & 17 & 0.1 & 8 & 14 \\ 3 & 3.5 & 1.7 & 10 & 17 & 8 \\ 17 & 8 & 0.05 & 10 & 0.1 & 14 \end{pmatrix}$, $c = \begin{pmatrix} 1 \\ 1.2 \\ 3 \\ 3.2 \end{pmatrix}$

and

$$p = \begin{pmatrix} 0.1312 & 0.1696 & 0.5569 & 0.0124 & 0.8283 & 0.5886 \\ 0.2329 & 0.4135 & 0.8307 & 0.3736 & 0.1004 & 0.9991 \\ 0.2348 & 0.1451 & 0.3522 & 0.2883 & 0.3047 & 0.6650 \\ 0.4047 & 0.8828 & 0.8732 & 0.5743 & 0.1091 & 0.0381 \end{pmatrix}$$

the value of global minimum is -3.322368.

- **Potential** function. The molecular conformation corresponding to the global minimum of the energy of N atoms interacting via the Lennard-Jones potential[55] is used as a test case here. The function to be minimized is given by:

$$V_{LJ}(r) = 4\epsilon\left[\left(\frac{\sigma}{r}\right)^{12} - \left(\frac{\sigma}{r}\right)^{6}\right] \tag{13}$$

In the current experiments three different cases were studied: $N = 5,\ 10,\ 20$

- **Rastrigin** function. The function is given by

$$f(x) = x_1^2 + x_2^2 - \cos(18x_1) - \cos(18x_2), \quad x \in [-1,1]^2$$

The global minimum is located at $x^* = (0,0)$ with value -2.0.

- **Shekel 7** function.

$$f(x) = -\sum_{i=1}^{7} \frac{1}{(x-a_i)(x-a_i)^T + c_i}$$

with $x \in [0,10]^4$ and $a = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 3 & 5 & 3 \end{pmatrix}$, $c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \end{pmatrix}$. The value of global minimum is -10.342378.

- **Shekel 5** function.

$$f(x) = -\sum_{i=1}^{5} \frac{1}{(x-a_i)(x-a_i)^T + c_i}$$

with $x \in [0,10]^4$ and $a = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \end{pmatrix}$, $c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \end{pmatrix}$. The value of global minimum is -10.107749.

- **Shekel 10** function.

$$f(x) = -\sum_{i=1}^{10} \frac{1}{(x-a_i)(x-a_i)^T + c_i}$$

with $x \in [0,10]^4$ and $a = \begin{pmatrix} 4 & 4 & 4 & 4 \\ 1 & 1 & 1 & 1 \\ 8 & 8 & 8 & 8 \\ 6 & 6 & 6 & 6 \\ 3 & 7 & 3 & 7 \\ 2 & 9 & 2 & 9 \\ 5 & 5 & 3 & 3 \\ 8 & 1 & 8 & 1 \\ 6 & 2 & 6 & 2 \\ 7 & 3.6 & 7 & 3.6 \end{pmatrix}$, $c = \begin{pmatrix} 0.1 \\ 0.2 \\ 0.2 \\ 0.4 \\ 0.4 \\ 0.6 \\ 0.3 \\ 0.7 \\ 0.5 \\ 0.6 \end{pmatrix}$. The value of global minimum is -10.536410.

- **Sinusoidal** function. The function is given by

$$f(x) = -\left(2.5 \prod_{i=1}^{n} \sin(x_i - z) + \prod_{i=1}^{n} \sin(5(x_i - z))\right), \quad 0 \le x_i \le \pi.$$

The global minimum is located at $x^* = (2.09435, 2.09435, ..., 2.09435)$ with $f(x^*) = -3.5$. In our experiments we used $n = 8, 32$ and $z = \frac{\pi}{6}$ and the corresponding functions are denoted by the labels SINU8 and SINU32 respectively.

- **Test2N** function. This function is given by the equation

  $$f(x) = \frac{1}{2}\sum_{i=1}^{n} x_i^4 - 16x_i^2 + 5x_i, \quad x_i \in [-5, 5].$$

  The function has 2^n in the specified range and in our experiments we used $n = 4, 5, 6, 7$. The corresponding values of global minimum is -156.664663 for $n = 4$, -195.830829 for $n = 5$, -234.996994 for $n = 6$ and -274.163160 for $n = 7$.

- Test30N function. This function is given by

  $$f(x) = \frac{1}{10}\sin^2(3\pi x_1)\sum_{i=2}^{n-1}\left((x_i-1)^2\left(1+\sin^2(3\pi x_{i+1})\right)\right)+(x_n-1)^2\left(1+\sin^2(2\pi x_n)\right)$$

  with $x \in [-10, 10]$. The function has 30^n local minima in the specified range and we used $n = 3, 4$ in our experiments. The value of global minimum for this function is 0.0

Constrained optimization test fuctions

The following functions were used from the relevant literature.

- **Levy** function. This problem is described in [64] and it is given by:

  $$\min_x f(x) = -x_1 - x_2$$

  with $x \in [0, 1]^2$, subject to

  $$g_1(x) = \left\lfloor (x_1-1)^2 + (x_2-1) \right\rfloor \left(\frac{1}{2a^2} - \frac{1}{2b^2}\right) + (x_1-1)(x_2-1)\left(\frac{1}{a^2} - \frac{1}{b^2}\right) - 1 \geq 0$$

  with $a = 2,\ b = 0.25$. The value of global minimum is $f_{\text{min}} = -1.8729$.

- **Salkin** function. This problem is described in [65] and it is given by:

  $$\max_x f(x) = 3x_1 + x_2 + 2x_3 + x_4 - x_5$$

  with $1 \leq x_1 \leq 4,\ 80 \leq x_2 \leq 88,\ 30 \leq x_3 \leq 35,\ 145 \leq x_4 \leq 150,\ 0 \leq x_5 \leq 2$ subject to

  $$\begin{aligned} g_1(x) &= 25x_1 - 40x_2 + 16x_3 + 21x_4 + x5 \leq 300 \\ g_2(x) &= x_1 + 20x_2 - 50x_3 + x_4 - x_5 \leq 200 \\ g_3(x) &= 60x_1 + x_2 - x_3 + 2x_4 + x_5 \leq 600 \\ g_4(x) &= -7x_1 + 4x_2 + 15x_3 - x_4 + 65x_5 \leq 700 \end{aligned}$$

  This global maximum is $f_{\text{max}} = 320$.

- **Hess** function. This problem is described in [66] and it is given by:

  $$\max_x f(x) = 25\left(x_1 - 2\right)^2 + \left(x_2 - 2\right)^2 + \left(x_3 - 1\right)^2 + \left(x_4 - 4\right)^2 + \left(x_5 - 1\right)^2 + \left(x_6 - 4\right)^2$$

  with $0 \leq x_1 \leq 5,\ 0 \leq x_2 \leq 1,\ 1 \leq x_3 \leq 5,\ 0 \leq x_4 \leq 6,\ 0 \leq x_5 \leq 5,\ 0 \leq x_6 \leq 10$ subject to:

  $$\begin{array}{rcl} g_1(x) & = & x_1 + x_2 - 2 \geq 0 \\ g_2(x) & = & -x_1 + x_2 + 6 \geq 0 \\ g_3(x) & = & x_1 - x_2 + 2 \geq 0 \\ g_4(x) & = & -x_1 + 3x_2 + 2 \geq 0 \\ g_5(x) & = & \left(x_3 - 3\right)^2 + x_4 - 4 \geq 0 \\ g_6(x) & = & \left(x_5 - 3\right)^2 + x_6 - 4 \geq 0 \end{array}$$

  The value of global maximum is $f_{\text{max}} = 310$.

- **Chootinan1** function. This problem is described in [67] and it is given by:

  $$\min_x f(x) = 5\sum_{i=1}^{4} x_i - 5\sum_{i=1}^{4} x_i^2 - \sum_{i=1}^{13} x_i$$

  with $0 \leq x_i \leq 1$ for $i = 1, .., 9, 13$, $0 \leq x_i \leq 100$ for $i = 10, 11, 12$ with the following constraints:

  $$\begin{array}{rcl} g_1(x) & = & 10 - (2x_1 + 2x_2 + x_{10} + x_{11}) \geq 0 \\ g_2(x) & = & 10 - (2x_1 + 2x_3 + x_{10} + x_{12}) \geq 0 \\ g_3(x) & = & 10 - (2x_2 + 2x_3 + x_{11} + x_{12}) \geq 0 \\ g_4(x) & = & 8x_1 - x_{10} \geq 0 \\ g_5(x) & = & 8x_2 - x_{11} \geq 0 \\ g_6(x) & = & 8x_3 - x_{12} \geq 0 \\ g_7(x) & = & 2x_4 + x_5 - x_{10} \geq 0 \\ g_8(x) & = & 2x_6 + x_7 - x_{11} \geq 0 \\ g_9(x) & = & 2x_8 + x_9 - x_{12} \geq 0 \end{array}$$

  The value of global minimum is $f_{\text{min}} = -15.0$.

- **G15** function. This problem is described in [68] and it is given by:

  $$\min_x f(x) = 1000 - x_1^2 - 2x_2^2 - x_3^2 - x_1x_2 - x_1x_3$$

  with $x \in [0, 10]^3$ subject to the following constraints:

  $$\begin{array}{rcl} h_1(x) & = & x_1^2 + x_2^2 + x_3^2 - 25 = 0 \\ h_2(x) & = & 8x_1 + 14x_2 + 7x_3 - 56 = 0 \end{array}$$

  The value of the global minimum is $f_{\text{min}} = 961.7150$

Compliance with Ethical Standards

All authors declare that they have no has no conict of interest.

Acknowledgments

The experiments of this research work was performed at the high performance computing system established at Knowledge and Intelligent Computing Laboratory, Dept of Informatics and Telecommunications, University of Ioannina, acquired with the project "Educational Laboratory equipment of TEI of Epirus" with MIS 5007094 funded by the Operational Programme "Epirus" 2014-2020, by ERDF and national finds.

References

[1] Zwe-Lee Gaing, Particle swarm optimization to solving the economic dispatch considering the generator constraints, IEEE Transactions on **18** Power Systems, pp. 1187-1195, 2003.

[2] C. D. Maranas, I. P. Androulakis, C. A. Floudas, A. J. Berger, J. M. Mulvey, Solving long-term financial planning problems via global optimization, Journal of Economic Dynamics and Control **21**, pp. 1405-1425, 1997.

[3] Q. Duan, S. Sorooshian, V. Gupta, Effective and efficient global optimization for conceptual rainfall-runoff models, Water Resources Research **28**, pp. 1015-1031 , 1992.

[4] P. Charbonneau, Genetic Algorithms in Astronomy and Astrophysics, Astrophysical Journal Supplement **101**, p. 309, 1995

[5] A. Liwo, J. Lee, D.R. Ripoll, J. Pillardy, H. A. Scheraga, Protein structure prediction by global optimization of a potential energy function, Biophysics **96**, pp. 5482-5485, 1999.

[6] P.M. Pardalos, D. Shalloway, G. Xue, Optimization methods for computing global minima of nonconvex potential energy functions, Journal of Global Optimization **4**, pp. 117-133, 1994.

[7] Eva K. Lee, Large-Scale Optimization-Based Classification Models in Medicine and Biology, Annals of Biomedical Engineering **35**, pp 1095-1109, 2007.

[8] Y. Cherruault, Global optimization in biology and medicine, Mathematical and Computer Modelling **20**, pp. 119-132, 1994.

[9] M.A. Wolfe, Interval methods for global optimization, Applied Mathematics and Computation **75**, pp. 179-206, 1996.

[10] T. Csendes and D. Ratz, Subdivision Direction Selection in Interval Methods for Global Optimization, SIAM J. Numer. Anal. **34**, pp. 922–938, 1997.

[11] W. L. Price, Global optimization by controlled random search, Journal of Optimization Theory and Applications **40**, pp. 333-348, 1983.

[12] Ivan Křivý, Josef Tvrdík, The controlled random search algorithm in optimizing regression models, Computational Statistics & Data Analysis **20**, pp. 229-234, 1995.

[13] M.M. Ali, A. Törn, and S. Viitanen, A Numerical Comparison of Some Modified Controlled Random Search Algorithms, Journal of Global Optimization **11**,pp. 377–385,1997.

[14] S. Kirkpatrick, CD Gelatt, , MP Vecchi, Optimization by simulated annealing, Science **220**, pp. 671-680, 1983.

[15] L. Ingber, Very fast simulated re-annealing, Mathematical and Computer Modelling **12**, pp. 967-973, 1989.

[16] R.W. Eglese, Simulated annealing: A tool for operational research, Simulated annealing: A tool for operational research **46**, pp. 271-281, 1990.

[17] R. Storn, K. Price, Differential Evolution - A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces, Journal of Global Optimization **11**, pp. 341-359, 1997.

[18] J. Liu, J. Lampinen, A Fuzzy Adaptive Differential Evolution Algorithm. Soft Comput **9**, pp.448–462, 2005.

[19] J. Kennedy and R. Eberhart, "Particle swarm optimization," Proceedings of ICNN'95 - International Conference on Neural Networks, 1995, pp. 1942-1948 vol.4, doi: 10.1109/ICNN.1995.488968.

[20] Riccardo Poli, James Kennedy kennedy, Tim Blackwell, Particle swarm optimization An Overview, Swarm Intelligence **1**, pp 33-57, 2007.

[21] Ioan Cristian Trelea, The particle swarm optimization algorithm: convergence analysis and parameter selection, Information Processing Letters **85**, pp. 317-325, 2003.

[22] M. Dorigo, M. Birattari and T. Stutzle, Ant colony optimization, IEEE Computational Intelligence Magazine **1**, pp. 28-39, 2006.

[23] K. Socha, M. Dorigo, Ant colony optimization for continuous domains, European Journal of Operational Research 185, pp. 1155-1173, 2008.

[24] D. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley Publishing Company, Reading, Massachussets, 1989.

[25] Z. Michaelewicz, Genetic Algorithms + Data Structures = Evolution Programs. Springer - Verlag, Berlin, 1996.

[26] S.A. Grady, M.Y. Hussaini, M.M. Abdullah, Placement of wind turbines using genetic algorithms, Renewable Energy **30**, pp. 259-270, 2005.

[27] Offord C., Bajzer Ž. (2001) A Hybrid Global Optimization Algorithm Involving Simplex and Inductive Search. In: Alexandrov V.N., Dongarra J.J., Juliano B.A., Renner R.S., Tan C.J.K. (eds) Computational Science - ICCS 2001. ICCS 2001. Lecture Notes in Computer Science, vol 2074. Springer, Berlin, Heidelberg. https://doi.org/10.1007/3-540-45718-6_73

[28] M. Pant, R. Thangaraj, C. Grosan and A. Abraham, "Hybrid Differential Evolution - Particle Swarm Optimization Algorithm for Solving Global Optimization Problems," 2008 Third International Conference on Digital Information Management, 2008, pp. 18-24, doi: 10.1109/ICDIM.2008.4746766.

[29] Niu B., Li L. (2008) A Novel PSO-DE-Based Hybrid Algorithm for Global Optimization. In: Huang DS., Wunsch D.C., Levine D.S., Jo KH. (eds) Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence. ICIC 2008. Lecture Notes in Computer Science, vol 5227. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-540-85984-0_20

[30] Y. Zhou and Y. Tan, "GPU-based parallel particle swarm optimization," 2009 IEEE Congress on Evolutionary Computation, 2009, pp. 1493-1500, doi: 10.1109/CEC.2009.4983119.

[31] L. Dawson and I. Stewart, "Improving Ant Colony Optimization performance on the GPU using CUDA," 2013 IEEE Congress on Evolutionary Computation, 2013, pp. 1901-1908, doi: 10.1109/CEC.2013.6557791.

[32] Barkalov, K., Gergel, V. Parallel global optimization on GPU. J Glob Optim 66, 3–20 (2016). https://doi.org/10.1007/s10898-016-0411-y

[33] Li W., A Parallel Multi-start Search Algorithm for Dynamic Traveling Salesman Problem. In: Pardalos P.M., Rebennack S. (eds) Experimental Algorithms. SEA 2011. Lecture Notes in Computer Science, vol 6630. Springer, Berlin, Heidelberg, 2011.

[34] Olli Bräysy, Geir Hasle, Wout Dullaert, A multi-start local search algorithm for the vehicle routing problem with time windows, European Journal of Operational Research **159**, pp. 586-605, 2004.

[35] Mauricio G.C. Resende, Renato F. Werneck,A hybrid multistart heuristic for the uncapacitated facility location problem, European Journal of Operational Research **174**, pp. 54-68, 2006.

[36] E. Marchiori, Genetic, Iterated and Multistart Local Search for the Maximum Clique Problem. In: Cagnoni S., Gottlieb J., Hart E., Middendorf M., Raidl G.R. (eds) Applications of Evolutionary Computing. EvoWorkshops 2002. Lecture Notes in Computer Science, vol 2279. Springer, Berlin, Heidelberg.

[37] Ya-Xiang Yuan, A Modified BFGS Algorithm for Unconstrained Optimization, IMA Journal of Numerical Analysis **11**, pp. 325–332, 1991.

[38] M.M. Ali, C. Storey, Topographical multilevel single linkage, J. Global Optimization 5 (1994) 349–358

[39] S. Salhi, N.M. Queen, A hybrid algorithm for identifying global and local minima when optimizing functions with many minima, European J. Oper. Res. 155(2004) 51–67.

[40] I. G. Tsoulos and I. E. Lagaris, MinFinder: Locating all the local minima of a function, Computer Physics Communications **174**, pp. 166-179, 2006.

[41] M. Perez, F. Almeida and J. M. Moreno-Vega, "Genetic algorithm with multistart search for the p-Hub median problem," Proceedings. 24th EUROMICRO Conference (Cat. No.98EX204), Vasteras, Sweden, 1998, pp. 702-707 vol.2.

[42] H. C. B. d. Oliveira, G. C. Vasconcelos and G. B. Alvarenga, "A Multi-Start Simulated Annealing Algorithm for the Vehicle Routing Problem with Time Windows," 2006 Ninth Brazilian Symposium on Neural Networks (SBRN'06), Ribeirao Preto, Brazil, 2006, pp. 137-142.

[43] Festa P., Resende M.G.C. (2009) Hybrid GRASP Heuristics. In: Abraham A., Hassanien AE., Siarry P., Engelbrecht A. (eds) Foundations of Computational Intelligence Volume 3. Studies in Computational Intelligence, vol 203. Springer, Berlin, Heidelberg.

[44] B. Betro‘, F. Schoen, Optimal and sub-optimal stopping rules for the multistart algorithm in global optimization, Math. Program. **57**, pp. 445–458, 1992.

[45] W.E. Hart, Sequential stopping rules for random optimization methods with applications to multistart local search, Siam J. Optim. **9**, pp. 270–290, 1998.

[46] I.E. Lagaris and I.G. Tsoulos, Stopping Rules for Box-Constrained Stochastic Global Optimization, Applied Mathematics and Computation **197**, pp. 622-632, 2008.

[47] J. Larson and S.M. Wild, Asynchronously parallel optimization solver for finding multiple minima, Mathematical Programming Computation **10**, pp. 303-332, 2018.

[48] H.P.J. Bolton, J.F. Schutte, A.A. Groenwold, Multiple Parallel Local Searches in Global Optimization. In: Dongarra J., Kacsuk P., Podhorszki N. (eds) Recent Advances in Parallel Virtual Machine and Message Passing Interface. EuroPVM/MPI 2000. Lecture Notes in Computer Science, vol 1908. Springer, Berlin, Heidelberg, 2000.

[49] Ioannis G. Tsoulos, Modifications of real code genetic algorithm for global optimization, Applied Mathematics and Computation **203**, pp. 598-607, 2008.

[50] M.J.D Powell, A Tolerant Algorithm for Linearly Constrained Optimization Calculations, Mathematical Programming **45**, pp. 547-566, 1989.

[51] [3] A. Corana, M. Marchesi, C. Martini, S. Ridella, Minimizing multimodal functions of continuous variables with the "Simulated Annealing" algorithm, ACM Trans. Math. Software 13 (1987) 262–280.

[52] M. Montaz Ali, Charoenchai Khompatraporn, Zelda B. Zabinsky, A Numerical Evaluation of Several Stochastic Algorithms on Selected Continuous Global Optimization Test Problems, Journal of Global Optimization **31**, pp 635-672, 2005.

[53] C.A. Floudas, P.M. Pardalos, C. Adjiman, W. Esposoto, Z. Gümüs, S. Harding, J. Klepeis, C. Meyer, C. Schweiger, Handbook of Test Problems in Local and Global Optimization, Kluwer Academic Publishers, Dordrecht, 1999.

[54] M. Gaviano, D.E. Ksasov, D. Lera, Y.D. Sergeyev, Software for generation of classes of test functions with known local and global minima for global optimization, ACM Trans. Math. Softw. **29**, pp. 469-480, 2003.

[55] J.E. Lennard-Jones, On the Determination of Molecular Fields, Proc. R. Soc. Lond. A **106**, pp. 463–477, 1924.

[56] O.A. Sauer, D.M. Shepard, T.R. Mackie, Application of constrained optimization to radiotherapy planning, Medical Physics **26**, pp. 2359-2366, 1999.

[57] F.P. Seelos, R.E. Arvidson, Bounded variable least squares - application of a constrained optimization algorithm to the analysis of TES Emissivity Spectra, in: 34th Annual Lunar and Planetary Science Conference, March 17-21, 2003, League City, Texas, abstract no.1817.

[58] M.J. Field, Constrained optimization of ab initio and semiempirical Hartree-Fock wave functions using direct minimization or simulated annealing , Journal of physical chemistry **95**, pp. 5104-5108, 1991.

[59] G.A. Williams, J.M. Dugan, R.B. Altman, Constrained global optimization for estimating molecular structure from atomic distances, Journal of Computational Biology **8**, pp. 523-547, 2001.

[60] P.E. Gill, W. Murray, The computation of Lagrange-multiplier estimates for constrained minimization, Mathematical Programming **17**, pp. 32-60, 1979.

[61] M.J.D. Powell, Y. Yuan, A trust region algorithm for equality constrained optimization, Mathematical Programming **49**, pp. 189-211, 2005.

[62] K. Ichida, Constrained optimization using interval analysis, Computers and Industrial Engineering **31**, pp. 933-937, 1996.

[63] R.L. Becerra, C.A.C. Coello, Cultured differential evolution for constrained optimization, Computer Methods in Applied Mechanics and Engineering **195**, pp. 4303-4322, 2006.

[64] A.V. Levy, A. Montalvo, The tunneling algorithm for global optimization of functions, SIAM Journal of Scientific and Statistical Computing **6**, pp. 15-29, 1985.

[65] H.M. Salkin, Integer programming, Edison Wesley Publishing Com., Amsterdam, 1975.

[66] R. Hess, A heuristic search for estimating a global solution of non convex programming problems, Operations Research **21**, pp. 1267-1280, 1973.

[67] P. Chootinan, A. Chen, Constraint handling in genetic algorithms using a gradient - based repair method, Computer and Operations Research **33**, pp. 2263-2281, 2006.

[68] J.J. Liang, T.P. Runarsson, E. Mezura-Montes, M. Clerc, P.N. Suganthan, C.A.C. Coello, K. Deb, Problem definitions and evaluation criteria for the CEC2006 special session on constrained real-parameter optimization, `http://www.ntu.edu.sg/home/EPNSugan/index_files/CEC-06/CEC06.htm`