Les Cahiers du GERAD ISSN: 0711–2440

Comparative Analysis of Continuous Global Optimization Methods

D. Kovačević, N. Mladenović,
P. Milošević, B. Petrović, V. Dobrić

G–2013–41

June 2013

Comparative Analysis of Continuous Global Optimization Methods

Darko Kovačević

Faculty of Organizational Sciences
University of Belgrade
Belgrade, Serbia

darko.kovacevic@nbs.rs

Nenad Mladenović

GERAD & Department of Mathematics
Brunel University-West London
London, UK

nenad.mladenovic@brunel.ac.uk

Pavle Milošević
Bratislav Petrović
Vladimir Dobrić

Faculty of Organizational Sciences
University of Belgrade
Belgrade, Serbia

pavle.milosevic@fon.bg.ac.rs
bratislav.petrovic@fon.bg.ac.rs
vdobric@gmail.com

June 2013

Les Cahiers du GERAD

G–2013–41

Abstract: In this paper we evaluate the performance and compare 19 different heuristics for solving continuous global optimization. They are all based on the following metaheuristics: Simulated annealing, Variable neighborhood search, Particle swarm optimization, and Differential evolution. Codes of methods are taken from their authors. The comparison on usual test instances (convex and non-convex) is performed on the same computer. Dimensions of test functions are changed from 10 to 100, thus effectively covering small and large scale problems. The results measured by computational efforts and ranked statistics show that the recent DE-VNS heuristic outperforms the other 18 algorithms on selected problems. Its better performances are noted in solving non-convex problems.

Key Words: Continuous Global Optimization, Metaheuristics, Differential Evolution, Variable Neighborhood Search, Comparison.

1 Introduction

In this paper, we consider an unconstrained global optimization problem in a continuous space. The general form of the problem is given below:

$$(\min) f(x), x \in X \subseteq R^n \tag{1}$$

where $f : R^n \to R$ is generally nonlinear, non-convex function defined on R^n. In cases where f is not convex function and/or X is not convex set, the defined problem is not easy to solve. Since the classic mathematical tools usually cannot help, one need to use approximate methods.

In the last 30 years several metaheuristics (or framework for building heuristics) have been developed, such as Simulated Annealing (SA) [13,18], Tabu Search [7,8], Variable Neighborhood Search (VNS) [19], etc. Several new optimization techniques have emerged in the past two decades that mimic biological evolution, or the way biological entities communicate in nature. Some of these algorithms have been used successfully in many areas with many constraints and non-linear processes. The most representative algorithms include Particle Swarm Optimization (PSO) [12], Genetic Algorithm [9], Differential Evolution (DE) [24, 29], etc. Particular interest is on global optimization of numerical, real valued "black box" problems for which exact and analytical methods are not applicable. There are a considerable number of papers devoted to comparing different optimization approaches. Typically, such comparison has been based on numerical benchmark problems [32] but in recent years, comparisons are evident to many real life problems, especially in engineering [4] and biology [21]. Many studies verify that one class of algorithms outperformed another on a given set of problems. To the best of our knowledge, no numerical comparison of such a large number of algorithms to the nonlinear continuous multidimensional global optimization problems, been presented previously.

The main objective of this paper is to evaluate whether one of tested heuristics would outperform others on benchmark problem instances. In addition to that, we are particularly interested in the behavior of algorithms depending on the problem size. "Curse of dimensionality" is one of the fundamental flaws of many heuristics, which, at the first glance, have promising results. Bearing that in mind, the results could also reveal whatever the algorithms would have particular preferences or difficulties regarding the specific problem or dimension as well as success rate in achieving the global optimum.

Overall, our experimental study suggests that DE variants are more efficient and robust in terms of number of function evaluations and precision. More particularly, the recent hybrid DE-VNS appears to be the best method on multimodal problems. However, some of SA variants behave similarly to the DE variants on selected convex problems, while VNS based variants show a remarkable convergence rate on some low dimensional problems.

This paper is organized as follows. In Section 2 we briefly give steps of the methods that will be used in this study: SA, VNS, PSO, and DE. Section 3 brings briefly overview of selected nonlinear continuous global optimization problems. In Section 4 the discussion of results is presented. Section 5 is devoted to statistical analysis of the obtained results and finally in Section 6 concludes our work.

2 Algorithms for comparison

In our comparative analysis, nineteen different algorithms are taken in consideration. These algorithms are based on rules given by well-known metaheuristics: Simulated Annealing, Variable Search Neighborhood, Particle Swarm Optimization, and Differential Evolution.

2.1 Heuristics based on Simulated Annealing

The basic variant of Simulated Annealing (SA) is presented by Kirkpatrick et al. [13]. The technique starts with an initial solution h, which simply assigns random values to all the parameters satisfying the initial constraints. The control variable analogous to temperature is marked as T and it decries to predefined value $T_{\min}$. A candidate solution h' is created by copying the parameters of h and then adding random values to each of the parameters. E is the current error and p stands for each of the kinetic parameters. Value of parameter T can be lowered by subtracting or multiplying current T by a value that is less than 1.

Three algorithms with different settings of control parameters are included. The first version will set the value of maximum temperature $T_{\max} = 100$ and the number of trial points $pop = 100$ (SA 100 100), which is actually a variant of the fast algorithm that quickly converges to the solution. Due to the speed of convergence, the global properties may be compromised, so we introduce the following variations to the algorithm.

Algorithm 1 General Simulated Annealing

Initialize h, pop
Set $T \leftarrow T_{\max}$
WHILE stopping criteria
    Set $i \leftarrow 1$
    WHILE until $i > pop$
        $h' \leftarrow h$
        For each kinetic parameter p to h', $p \leftarrow p + k \cdot ln(\sqrt{E+1}) \cdot N(\bar{x}, \sigma)$
        $\Delta E = Error(h') - Error(h)$
        IF $\Delta E \leq 0$ **then**
            Set $h \leftarrow h'$
        ELSE
            Set $h \leftarrow h'$ with probability $e^{-\frac{\Delta E}{T}}$
        END IF
        Set $i \leftarrow i + 1$
    END WHILE
    Lower T until $T = T_{\min}$
END WHILE
Stopping criteria: x^* is an approximate solution of the problem

In the second variant will use the same number of trial points $pop = 100$ but we will increase the value of the maximum temperature to $T_{\max} = 500$ (SA 500 100), allowing the higher initial energy of the system and thus provide an easier escape from local optima. As the latest version of general SA will use the values of control parameters $T_{\max} = 100$, or a lower initial energy of the system which will compensated by a large population $pop = 500$ (SA 100 500), that will enable diverse solutions for child offsprings. The last version used in comparison is SA with reheating (SAR) that adjusts the temperature of the system $T_{\max}$ depending on the speed of convergence, leaving a population trial points at $pop = 100$. Unlike the SAR version proposed in [1], the system is reheated to $T_{\max}$.

Besides these basic SA algorithms, the Simplex-Simulated annealing (SIMPSA) is included in the comparison. Control parameters of interest are cool rate $CoolR$, which controls the speed of convergence of the system, and the acceptance rate $AccR$, with the method of estimation of initial temperature of the system proposed in [2], and adding random fluctuations to current function values of vertices [23]. Two SIMPSA variants, with control parameter settings $CoolR = 1$, $AccR = 0.95$ (SIMPSA 1), and $CoolR = 2$, $AccR = 0.3$ (SIMPSA 2), are included in the comparison. In the first case the algorithm has the acceptance rate by a larger number of dimensions and a slower convergence. In the other case we force slower population clustering with lower acceptance rate value, and thus reduce the convergence speed that we have introduced with higher cool rate.

2.2 Heuristics based on Variable Neighborhood Search

Mladenović and Hansen [19] proposed Variable Neighborhood Search, the metaheuristic based on systematic change of neighborhoods. It explores increasingly distant neighborhoods of the current best solution. If better solution is found, VNS jumps from the current solution to the new one. Many VNS extensions for continuous global optimization are made using this idea.

Mladenović et al. [20] have presented Glob-VNS, algorithm that utilizes the idea of using several geometric neighborhood structures and random distributions in the shaking step. Neighborhood structure $N_k(x)$ is defined as:

$$N_k(x) = \{y \in S | \rho_k(x, y) \leq r_k\} \tag{2}$$

or

$$N_k(x) = \{y \in S | r_{k-1} \leq \rho_k(x, y) \leq r_k\} \tag{3}$$

It is determined by the geometry of neighborhood structure and its radius r_k. Values of radius can be specified by the user or generated automatically during the search, with a condition that the values must be monotonically nondecreasing with k. Geometry of neighborhood structures is defined by choice of metric function ρ_k – authors use l_p distance as metric, usually l_1, l_2, and l_∞. The uniform distribution is the mostly used for obtaining y from $N_k(x)$. Other distributions can be used in shaking step, too.

Gaussian VNS (Gauss-VNS) is presented by Carrizosa et al. [3]. The main idea of this approach is defining a class of probability distributions $P_k(x)$, instead of class neighborhoods $N_k(x)$. It is assumed that each distribution is a n-variate Gaussian distribution centered at x and covariance matrix $\sum_k$. The next trial point in shaking step is generated by $P_k(x)$. Gauss-VNS is particularly user-friendly, because only the sequence of variances for defining a covariance matrix should be specified by the user.

Algorithm 2 Gauss-VNS

```
Select a set of covariance matrices ∑_k, k = 1, ..., k_max
Chose an arbitrary initial point x
Set x* ← x, f* ← f(x)
WHILE stopping criteria
    Set k ← 1
    WHILE until k > k_max
        Shake: Generate y from a Gaussian distribution with mean x* and covariance matrix ∑_k
        Apply some local search method from y to obtain a local minimum y'
        IF f(y') < f* then
            Set x* ← y', f* ← (y')
        END IF
        Set k ← k + 1
    END WHILE
END WHILE
Stopping criteria: x* is an approximate solution of the problem
```

2.3 Heuristics based on Particle Swarm

Kennedy and Eberhart [12] introduced Particle Swarm Optimization (PSO). PSO has roots in two main areas: bird flocking, fish schooling, and swarming theory; and evolutionary computation, genetic algorithms and evolutionary programming. The search process can be described as particles being "flown" through the hyper dimensional space. They adjust its position based on position of best particle and its best position found so far. Each particle is adjusted to move closer to the best particle in a predefined neighborhood according to the following equations:

$$\nu_i \leftarrow \omega \cdot \nu_i + U(0, \varphi_1) \otimes (p_i - x_i) + U(0, \varphi_2) \otimes (p_g - x_i) \tag{4}$$

$$x_i \leftarrow x_i + \nu_i \tag{5}$$

where ν_i is velocity vector, x_i position vector, p_i the better position vector, p_g the better position vector for the good neighbor, $U(0, \varphi_i)$ is a vector of random numbers uniformly distributed in $[0, \varphi_i]$, φ_1 and φ_2 are cognitive and social acceleration coefficients, and $\otimes$ is component-wise multiplication. Population size is referred as *popSize*. At the original PSO variant ω is fixed at 1. Later, Shi and Eberhart [27] refer this parameter as inertia weight.

Algorithm 3 Particle Swarm

Initialize a population array of particles with random positions and velocities
WHILE stopping criteria
    Set $i \leftarrow 1$
    WHILE until $i > popSize$
        Update velocity vector
        $\nu_i \leftarrow \nu_i + U(0, \varphi_1) \otimes (p_i - x_i) + U(0, \varphi_2) \otimes (p_g - x_i)$
        Update position vector $x_i \leftarrow x_i + \nu_i$
        IF $f(x_i) < f(p_i)$ **then**
            Set $p_i \leftarrow x_i$
            IF $f(p_i) < f(p_g)$ **then**
                Set $p_g \leftarrow p_i$
            END IF
        END IF
        Set $i \leftarrow i + 1$
    END WHILE
END WHILE
Stopping criteria: x^* is an approximate solution of the problem

The PSO GBest topology (for "global best") is the static topology. At PSO GBest, the best neighbor in the entire population influenced the target particle.

In contrast to PSO GBest, where the neighborhood is the entire swarm, PSO LBest [5] utilizes a neighborhood with smaller size K. For $K = 2$, it is a simple ring lattice where each particle is connected to neighboring members in the population array. PSO LBest can be generalized to $K > 2$. The main advantage of PSO LBest is parallel search. A smaller neighborhood size usually leads to slower convergence, but increases diversity – a larger part of the search space is covered for smaller neighborhoods.

2.4 Heuristics based on Differential Evolution

Storn and Price [24, 29] proposed Differential Evolution, simple and straightforward metaheuristic that consists three main parts: strategy, crossover and selection. There are two basic strategy approaches: "$DE/rand/1/bin$", characterized by slow convergence speed and stronger exploration capability, and "$DE/best/1/bin$", which has the high convergence speed and performs well on the unimodal problems. Crossover determines whether the target or the trial vector survives to the next generation. At last phase, we have the selection based on the choice of better solutions. The configuration and adaptation of mutation parameter F and crossover parameter CR are crucial for the performance of DE based algorithms [17]. The parameter adaptation techniques are divided into deterministic, adaptive, and self-adaptive control rules, e.g. Smith and Fogarty [28]. Deterministic rules modify the parameters according to certain predetermined rationales without utilizing any feedback from the search process. Adaptive rules incorporate some form of the feedback from the search procedure to guide the parameter adaptation. Self-adaptive rules directly encode parameters into the individuals and evolve them together with the encoded solutions. Four deterministic and four self-adaptive DE algorithms are taken in consideration.

Since it is well known that "$DE/rand/1/bin$" version of the strategy has the possibility for good searching solution space and therefore better performance in solving multimodal global optimization problem, which is the main focus of this paper, we compare three variants of this strategy. We will use the values of control parameters $F = 0.5$, $CR = 0.3$ (DERand 0.5 0.3), and $F = 0.5$, $CR = 0.5$ (DERand 0.5 0.5), which proved to be promising strategy according to [25]. In addition to these variations, we will observe the pure stochastic variation of the algorithm. Control parameters are defined as $F = Unif(0.4, 1)$ and $CR = Unif(0, 1)$ (DERand rand), which proved to be a good variant of DE algorithms for searching solution space.

Although the main focus of this paper is global multimodal problems, one DE variant that uses "$DE/best/1/bin$" strategy is taken in consideration, because this algorithm is well-known and is commonly used in comparisons. The values of control parameters $F = 0.5$ and $CR = 0.3$ are chosen on the basis of

Algorithm 4 Differential Evolution using "$DE/rand/1/bin$" strategy

Randomly initialize a population of N individuals $P_G = \{X_i, \ldots, X_N\}, i = 1, \ldots, N$
Evaluate the population
Set F, CR
WHILE stopping criteria
    Set $k \leftarrow 1$
    WHILE until $k > N$
        Applying strategy "$DE/rand/1/bin$" with F and CR parameters
        Evaluate child vector $f\left(y^k_{child}\right)$
        IF $f\left(y^k_{child}\right) < f\left(y^k_{parent}\right)$ **then**
            $parent = child$
        END IF
        Set $k \leftarrow k + 1$
    END WHILE
END WHILE
Stopping criteria: x^* is an approximate solution of the problem

good behavior version of "$DE/rand/1/bin$" at these values of control parameters. That algorithm will be referred as DEBest in this paper.

Qin et al. [25] presented Self-adaptive DE (SaDE) that instead of employing the computationally expensive trial-and-error search for the most suitable strategy and its parameter values, maintains a strategy candidate pool. Strategy candidate pool includes several trial vector generation strategies with effective, and yet diverse characteristics. During evolution, one strategy will be chosen from the candidate pool and applied to perform the mutation operation. Four trial vector generation strategies are included in the strategy candidate pool: "$DE/rand/1/bin$", "$DE/rand/2/bin$", "$DE/rand-to-best/2/bin$", and "$DE/current-to-rand/1/bin$". In SaDE algorithm, the parameter F is approximated by a normal distribution with mean value 0.5 and standard deviation 0.3. This setting enables to maintain both exploitation and exploration power throughout the evolution process. Parameter CR is taken from normal distribution $N(\mu_{CR}, 0.1)$, where μ_{CR} is initialized as 0.5. To adapt CR to proper values, the authors update μ_{CR} every 25 generations based on the recorded successful CR values since the last μ_{CR} update.

Zhang and Sanderson [34] have presented their DE based algorithm named JADE. The main contribution of JADE is the implementation of a new mutation strategy – "$DE/current-to-pbest$", that is a generalization of the classic "$DE/current-to-best$" strategy. This strategy has an optional external archive and updating control parameters in an adaptive manner. After each generation, the parent solutions that fail in the selection process are added to the archive. If the archive size exceeds a certain threshold, then some solutions are randomly removed from it. The role of the archive is to provide information about the progress direction and to improve the diversity of the population. Crossover probability CR_i is randomly taken from a normal distribution of mean μ_{CR} and standard deviation 0.1, and then truncated to [0, 1]. Mean μ_{CR} is initialized to be 0.5 and then updated at the end of each generation as:

$$\mu_{CR} = (1 - c) \cdot \mu_{CR} + c \cdot mean_A\left(CR_{succ}\right) \tag{6}$$

where c is a constant between 0 and 1, $mean_A(\,)$ is the arithmetic mean and CR_{succ} as the set of all successful crossover probabilities CR_i. The mutation factor F_i of is randomly taken from a Cauchy distribution with location parameter μ_F and scale parameter 0.1, and then truncated to be 1 if $F_i \geq 1$ or regenerated if $F_i \leq 0$. The location parameter μ_F is initialized to be 0.5 and then updated at the end of each generation as:

$$\mu_F = (1 - c) \cdot \mu_F + c \cdot mean_L\left(F_{succ}\right) \tag{7}$$

where $mean_L(\,)$ is the Lehmer mean and F_{succ} is the set of all successful mutation factors.

Wang et al. [33] proposed novel optimization method, called composite DE (CoDE). This method uses three trial vector generation strategies and three control parameter settings and randomly combines them to generate trial vectors. The strategy candidate pool is consisted of following strategies: "$DE/rand/1/bin$", "$DE/rand/2/bin$", and "$DE/current - to - rand/1$". The three control parameter settings are: $[F = 1.0, CR = 0.1]$, $[F = 1.0, CR = 0.9]$, and $[F = 0.8, CR = 0.2]$. At each generation, each of these trial vector generation strategies is used to create a new trial vector with a control parameter setting randomly chosen from the parameter candidate pool. Thus, three trial vectors are generated for each target vector, and they are compared in the next step. The best trial vector enters the next generation if it is better than its target vector.

In addition to the above method, the results of the analysis incorporate self-adaptive DE based on competitive settings, referred as DEbr18 [30]. This algorithm gives a higher probability of selecting those values of control parameters that were in previous iterations proved more successful in finding better child offspring. A number of settings for control parameter CR and F are denoted by H. Among them values are chosen at random with probability $p_h, h = 1, 2, \ldots, 3, H$. The h^{th} setting is successful if it generates such a trial point y that $f(y) < f(x_i)$. When n_h is the current number of the $f(y)$ setting successes, probability p_h can be calculated as the relative frequency as:

$$p_h = \frac{n_h + n_0}{\sum_{j=1}^{H} (n_j + n_0)} = \frac{n_h + n_0}{n_0 \cdot H + \sum_{j=1}^{H} n_j} \tag{8}$$

where $n_0 > 0$ is a constant. If any of probabilities drop below a given threshold $\delta > 0$ and $\delta < 1$, current values of p_h are reset to its starting values $p_h = 1/H$. Thus the premature convergence of probabilities p_h is avoided.

2.5 DE-VNS hybrid

Kovačević et al. [14] proposed hybrid approach based on DE with estimating crossover parameter CR using neighborhood search approach DE-VNS. The authors introduce a family of adaptive distributions, that depend on variable neighborhood parameter – par. Crossover parameter CR values are chosen based on par. Implementing the idea of VNS, the search around the current vector starts from the closest neighborhoods, and if a better solution is not found, progressively increasing of the neighborhood is applied on CR. When the algorithm finds a child vector better than the parent vector, in the next iterations it will be required that neighborhood parameter par remains at low values which imply the crossover by just few dimensions i.e. closest dimensional neighborhoods. In this way, it is ensured that the entire population would not have converged too quickly, and therefore more detailed search of area around the population vector. For these reasons, the iterations in which the algorithm cannot find a more satisfactory solution will gradually increase the value of the neighborhood factor using the $stepfactor$, allowing finding new solutions in further dimensional distances around the parent vector. In the case of finding a favorable child vector, algorithm resets the distribution according to the objective function. The authors used extended version – two sided power distribution (TSP) [21] as a proxy for beta distribution [11], because the benefits of the control parameters comprehensible set evaluation. The problem of selecting the values of the parameter F is solved by roulette methods which gradually give greater probability of drawing the successful values of F from TSP. DE-VNS strategy is based on the modification of "$DE/rand/1/bin$" strategy that comprise the characteristics of "$DE/best/1/bin$", named "$DE/Rand - Local - Best/1/bin$".

3 Test functions

For the numerical experiments, we use seven common benchmark functions, whose global optima are known. They are selected from [26]. The focus will be on multi-modal problems, however, in order to demonstrate the robustness of the proposed approaches on a wide class of problems, several easy (convex) objective functions will be considered as well.

3.1 Schwefel's function

Schwefel's function (presented at Figure 1) is deceptive in that the global minimum is geometrically distant, over the parameter space, from the next best local minima. Therefore, the search algorithms are potentially

Algorithm 5 DE-VNS

Randomly initialize a population of N individuals $P_G = \{X_i, \ldots, X_N\}, i = 1, \ldots, N$
Evaluate the population
Setting initial roulette probability for F parameter
WHILE stopping criteria
  Set $k \leftarrow 1$
  WHILE until $k > N$
    Calculate roulette probability for F parameters $p_h = \frac{n_h + n_0}{\sum_{j=1}^{H}(n_j + n_0)}$
    Sampling CR from adaptive beta distribution as:
    $$CR = F(x|a,m,b,par) = \begin{cases} \frac{m-a}{b-a}\left(\frac{x-a}{m-a}\right)^{\frac{1}{par}}, & a < x < m \\ 1 - \frac{b-m}{b-a}\left(\frac{b-x}{b-m}\right)^{\frac{1}{par}}, & m < x < b \end{cases}$$
    Applying strategy "$DE/Rand - Local - Best/1/bin$" with obtained F and CR parameters
    Evaluate child vector $f(y^k_{child})$
    IF $f\left(y^k_{child}\right) < f\left(y^k_{parent}\right)$ **then**
      $par^k_{new} = \max\left(par_{\min}, par^k_{old} - \left(f(y^k_{child}) = f(y^k_{parent})\right)\right)$
      $parent = child$
    ELSE
      $par^k_{new} = par^k_{old} + stepfactor$
      $par^k_{new} = \min(par^k_{new}, par_{\max})$
    END IF
    Set $k \leftarrow k + 1$
  END WHILE
END WHILE
Stopping criteria: x^* is an approximate solution of the problem

prone to convergence in the wrong direction.

$$F_1 = \sum_{i=1}^{n}\left[-x_i sin\left(\sqrt{|x_i|}\right)\right] \tag{9}$$

with $-500 \leq x_i \leq 500$ and $\min F_1(420.9687, 420.9687, \ldots, 420.9687) = -418.9829 \cdot n$.

3.2 Ackley's function

Ackley's function (presented at Figure 2) is a widely used multimodal test function. This function has an exponential term that covers its surface with numerous local minima. The complexity of Ackley's function is moderate. In order to obtain good results for this function, the search strategy must combine the exploratory and exploitative components efficiently.

$$F_2 = -20exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=0}^{n}x_i^2}\right) - exp\left(\sqrt{\frac{1}{n}}\sum_{i=0}^{n}cos(2\pi x_i)\right) + 20 \tag{10}$$

with $-32 \leq x_i \leq 32$ and $\min F_2(0, 0, \ldots, 0) = 0$.

3.3 Griewank's function

Griewank's function (presented at Figure 3) is similar to the function of Rastrigin. It has many widespread local minima regularly distributed over the solution space.

$$F_3 = \frac{1}{4000}\sum_{i=1}^{n}x_i^2 - \prod_{i=1}^{n}cos\left(\frac{x_i}{\sqrt{i}}\right) + 1 \tag{11}$$

with $-600 \leq x_i \leq 600$ and $\min F_3(0, 0, \ldots, 0) = 0$.

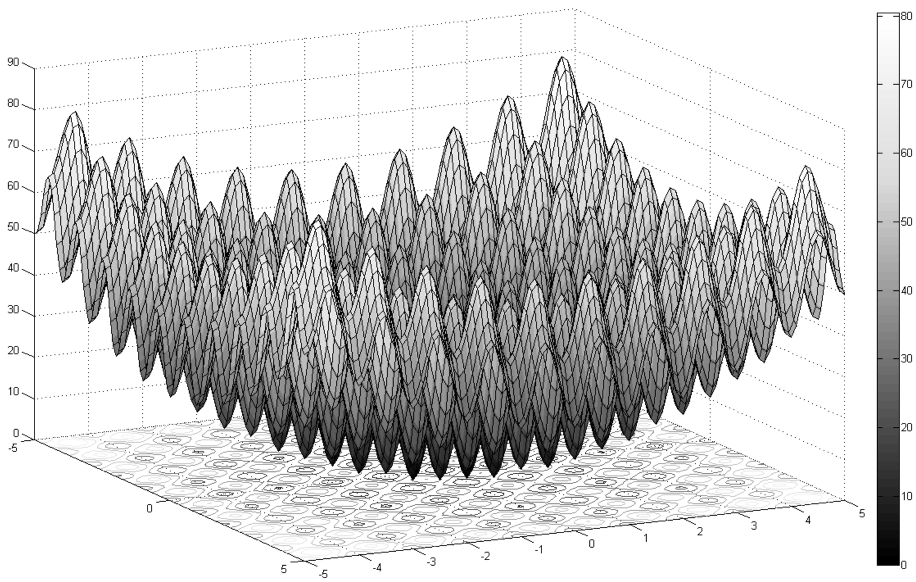


Figure 1: Schwefel's function

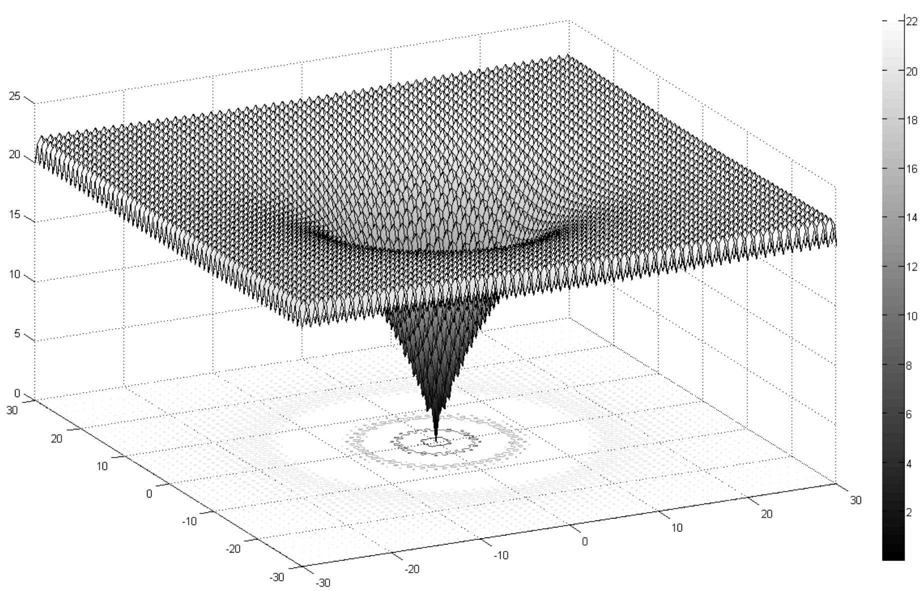


Figure 2: Ackley's function

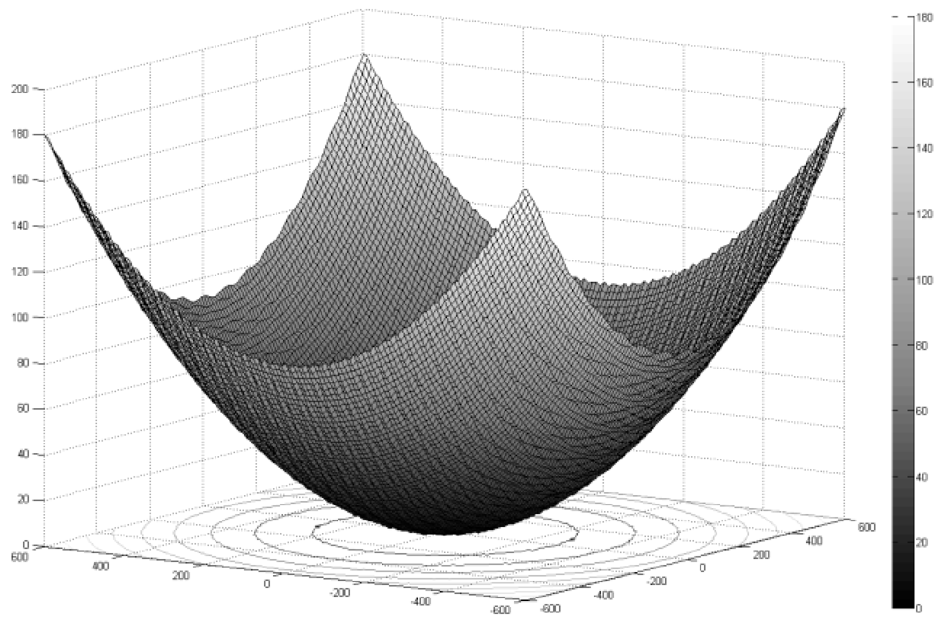


Figure 3: Griewank's function

3.4 Rastrigin's function

Rastrigin's function (presented at Figure 4) is based on the function of De Jong with the addition of cosine modulation in order to produce frequent local minima. Thus, the test function is highly multimodal. However, the locations of the minima are regularly distributed.

$$F_4 = 10n + \sum_{i=1}^{n} \left(x_i^2 = 10cos(2\pi x_i) \right) \tag{12}$$

with $-5.12 \leq x_i \leq 5.12$ and $\min F_4(0, 0, \ldots, 0) = 0$.

3.5 Molecular potential energy (MPE) function

MPE function [16] (presented at Figure 5) is the functional form similar to general potential energy functions, whose global minimum is known. The number of local minima of this function increases exponentially with the size of the problem.

$$F_5 = \sum_{i=1}^{n} \left(1 + cos(3x_i) + \frac{(-1)^i}{\sqrt{10.60099896 - 4.141720682 \cdot cos(x_i)}} \right) \tag{13}$$

with $0 \leq x_i \leq 5$ and $\min F_5(0, 0, \ldots, 0) = -0.0411183034 \cdot n$.

3.6 Rosenbrock's function

Rosenbrock's valley (presented at Figure 6) is a classic optimization problem, also known as banana function or the second function of De Jong. The global optimum lies inside a long, narrow, parabolic shaped flat valley. To find the valley is trivial, however convergence to the global optimum is difficult and hence this problem has been frequently used to test the performance of optimization algorithms. The function has the following definition:

$$F_6 = \sum_{i=1}^{n-1} \left[100 \cdot (x_{i+1} - x_i^2)^2 + (1 - x_i)^2 \right] \tag{14}$$

with $-5 \leq x_i \leq 5$ and $\min F_6(0, 0, \ldots, 0) = 0$.

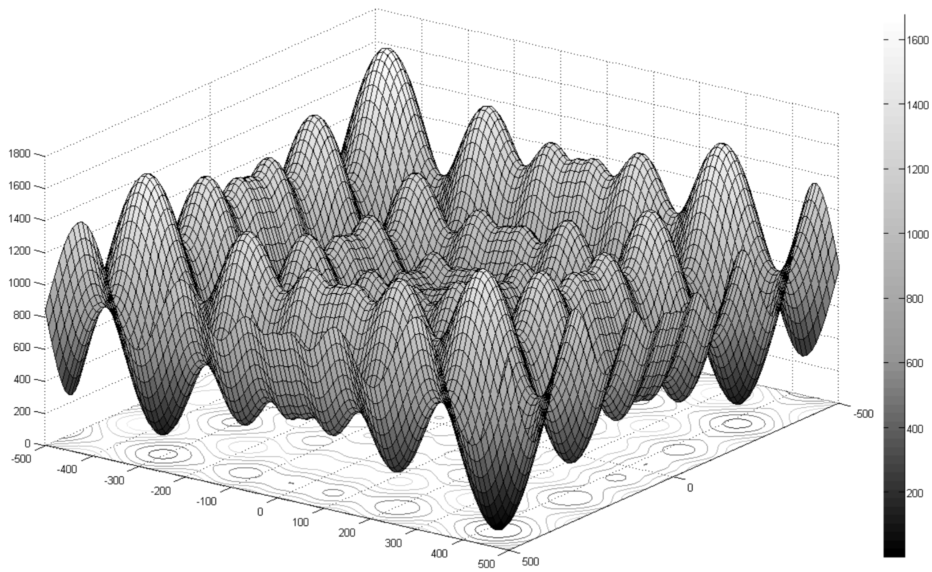


Figure 4: Rastrigin's function

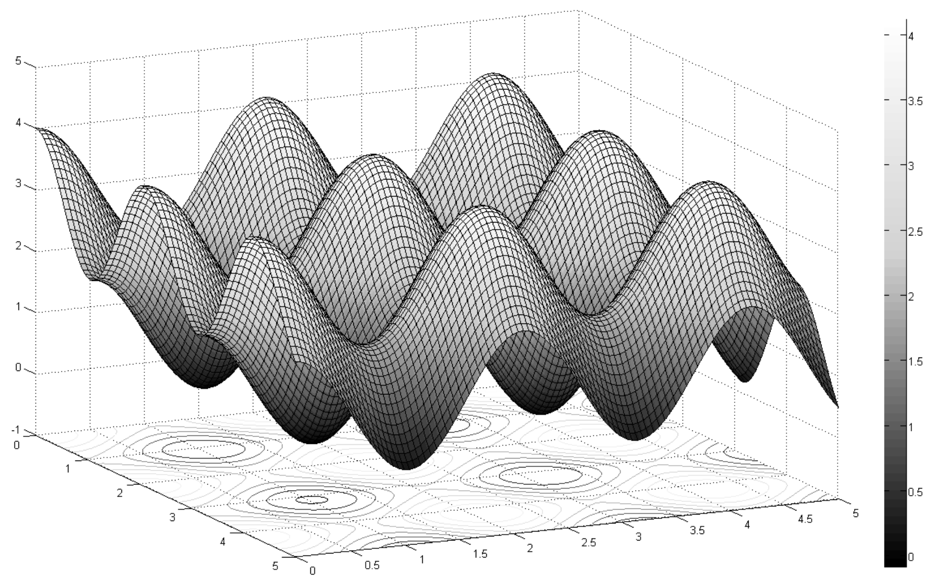


Figure 5: MPE function

3.7 Sphere function

The sum of different powers is a commonly used unimodal test function. Sphere function (presented at Figure 7) is a simple and strongly convex function used in the development of the theory of evolutionary strategies. It has the following definition:

$$F_7 = \sum_{i=1}^{n} x_i^2 \tag{15}$$

with $-1 \leq x_i \leq 1$ and $\min F_7(0, 0, \ldots, 0) = 0$.

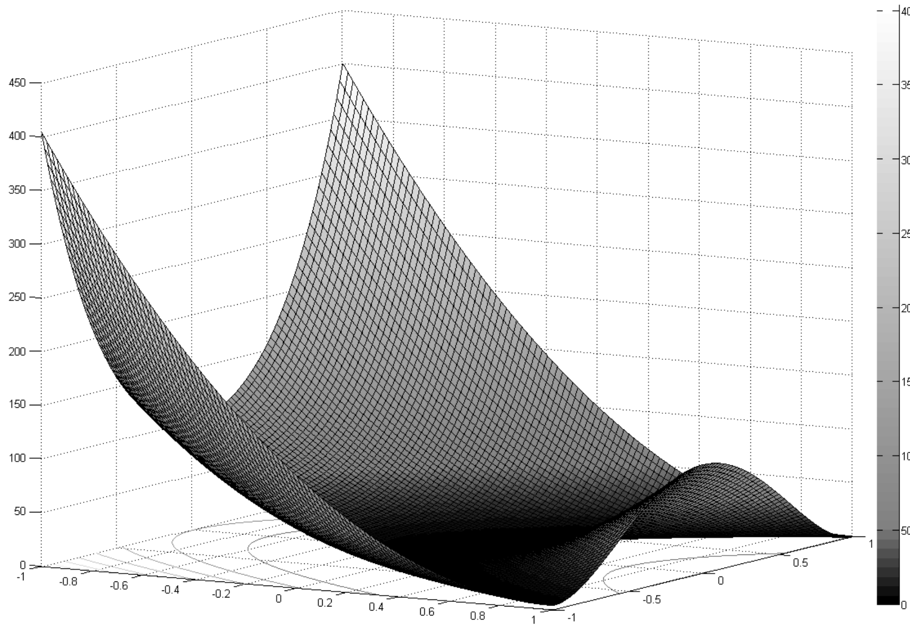


Figure 6: Rosenbrock's function

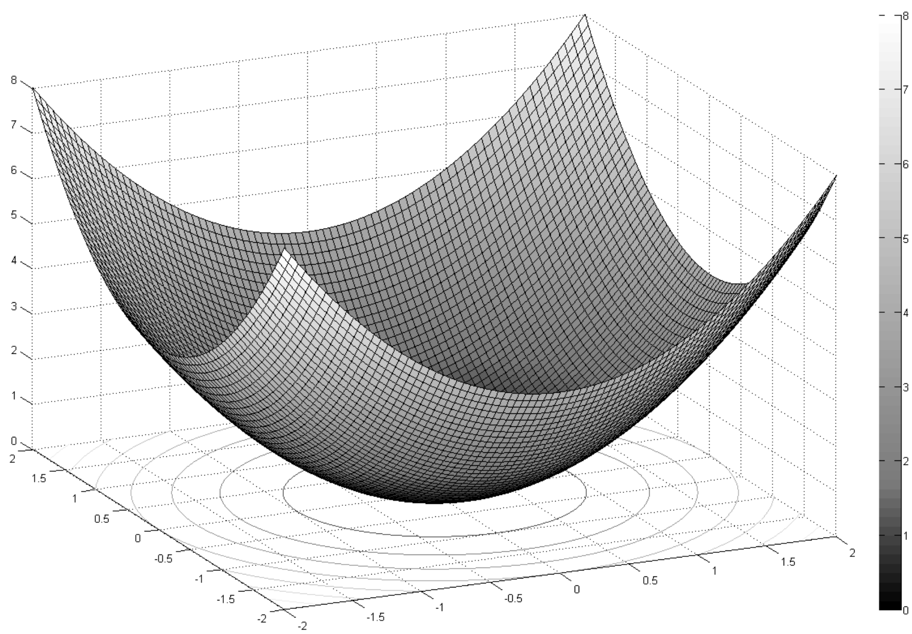


Figure 7: Sphere function

4 Computational results

Codes of all 19 methods are taken from authors who originally proposed their heuristics. They are then run on our computer own Pentium dual core computer using Matlab environment. In analyzing the behavior of the selected algorithms we will be focused on the unconstrained continuous multimodal global optimization problems. To show their robustness, we vary the dimension (D) from 10 to 100; in that way we will cover large scale problem instances as well. A value of 10^{-6} is used for the predefined tolerance around the global optimum. The other stopping condition is the maximum number of function evaluations. For the Rosenbrock function it is set to $50,000^*D$ and $10,000^*D$ for all other functions. Each problem is repeated 25 times, in order to obtain credible data.

The number of function evaluations (FEs) is the usual indicator used to compare methods. In addition to FEs we will be also indicate the success rate (SR), or the percentage of successfully achieved optima within the predefined tolerance.

Additional parameters for the DE-VNS are: (i) $parmin = 0$; (ii) $parmin = 0.7$;

$$stepfactor = \frac{1}{10 \cdot D \cdot \log_2(D)} \tag{16}$$

Parameters a, b, and m were defined as follows: $a = 0$, $b = 1$, $m = 0$. For all DE variants used in this study, the maximum number of parameter evaluation $evalmax$ and population pop were set as it is shown in Table 1.

Table 1: Values of $evalmax$ and pop parameters for all DE variants

Dimension	$evalmax$	pop
10	1e + 5	34
20	2e + 5—1e + 6	44
30	1.5e + 6	50
50	5e + 5—2.5e + 6	80
100	1e + 6	100

In addition, both PSO variants' swarm consists of 50 particles. The inertia weight varies linearly from 0.9 in the first velocity update to 0.4 in the final velocity update. The cognitive and social acceleration coefficients are: $\varphi_1 = \varphi_2 = 1.49618$. Also, for the PSO, parameter $Lbest$, the neighborhood of each particle is consisted of two particles, $K = 2$. These settings are noted as promising in [22].

4.1 Comparative analysis

In Tables 2 to 8, the test results for each function are shown, varying the different values of the dimension D. Gray areas indicate that for a given problem, the tolerance around the global optimum is not reached in 100% of cases. Tables also display the basic information for function evaluations: minimum FEs number ($evalmin$), average FEs number ($evalavg$) and maximum FEs number ($evalmax$). Another indicator that we are interested in is named $fmin_D$ and it presents the difference between the average cost function and known global optima in cases where tolerance around the global optima is not achieved, or tolerance in others. The best results are marked with bold fields, provided that the heuristic achieved SR percentage of 100%. The last column in all of these tables is overall score (OS) for each of selected algorithms. This variable is calculated as follows:

$$OS = \log\left(\sum_{D} \frac{eval\ avg}{D} fmin_D\right), \begin{cases} D = 10, 20, 30, 50, & \text{for Rosenbrock} \\ D = 10, 20, 50, 100, & \text{in other cases} \end{cases} \tag{17}$$

The overall score is the logarithm of the sum of elements that characterize optimization for selected dimensional problem. The average number of function evaluations is the base of each calculation. Value of $evalavg$ is divided by the dimension D. In that way, both the small and the large dimensional problems get the same importance. The fact that the global minimum is not reached is indicated by value $fmin_D$. Success rate is not included in the formula, so there is no double counting. Optimization is considered better if OS value is less. The logarithm is used for normalizing results. Algorithms are ranked using this criterion function in each of these tables. Before we give general conclusions, we analyze performances of all 19 heuristics on 7 different test functions, varying the dimension value D.

Schwefel functions (Table 2). Five algorithms have SR of 100%: DE-VNS, SaDE, DERand 0.5 0.3, DEbr18, and DERand 0.5 0.5; DE-VNS is the fastest (tolerance is achieved in less FEs) for all dimensions; all three

Table 2: Results for Schwefel's functions

SCHWEFEL'S

		10					20					50					100					Overall
		eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	score
1	DE-VNS	**7,901**	**8,901**	**10,101**	**100%**	**1.00E-06**	**21,201**	**23,431**	**27,301**	**100%**	**1.00E-06**	**70,401**	**79,081**	**89,101**	**100%**	**1.00E-06**	**172,201**	**189,791**	**210,001**	**100%**	**1.00E-06**	-2.2564
2	SaDE	12,001	12,101	13,001	100%	1.00E-06	25,001	27,601	29,001	100%	1.00E-06	84,001	87,501	93,001	100%	1.00E-06	220,001	233,901	249,001	100%	1.00E-06	-2.1753
3	DERand 0.5 0.3	9,001	9,441	9,901	100%	1.00E-06	23,601	24,601	25,601	100%	1.00E-06	94,301	101,161	105,301	100%	1.00E-06	350,501	385,181	408,701	100%	1.00E-06	-2.0942
4	DEbr18	13,801	14,711	15,301	100%	1.00E-06	42,901	43,851	44,801	100%	1.00E-06	161,201	165,541	168,001	100%	1.00E-06	417,301	420,211	421,101	100%	1.00E-06	-1.9517
5	DERand 0.5 0.5	8,901	9,471	10,401	100%	1.00E-06	26,001	28,331	30,501	100%	1.00E-06	134,501	150,211	167,801	100%	1.00E-06	661,001	762,971	853,801	100%	1.00E-06	-1.8861
6	SA 100 100	6,001	13,001	29,001	100%	1.00E-06	22,001	27,601	37,001	100%	1.00E-06	66,001	91,601	106,001	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	5.52E-06	-1.2237
7	SA 100 500	13,001	14,101	15,001	100%	1.00E-06	26,001	28,901	38,001	100%	1.00E-06	85,001	108,901	134,001	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	6.31E-06	-1.1667
8	SA 500 100	11,001	14,001	16,001	100%	1.00E-06	26,001	29,401	33,001	100%	1.00E-06	81,001	105,201	144,001	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	6.46E-06	-1.1576
9	SAR	5,101	12,251	27,401	100%	1.00E-06	15,201	28,611	69,101	100%	1.00E-06	80,401	101,861	113,801	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	7.38E-06	-1.1050
10	JADE	9,001	56,401	100,000	52%	7.11E+01	31,001	150,401	200,000	28%	9.48E+01	102,001	263,901	500,000	60%	5.92E+01	223,001	615,601	1,000,000	52%	1.07E+02	6.3194
11	SIMPSA 1	7,101	17,191	63,101	100%	1.00E-06	19,301	36,721	97,801	100%	1.00E-06	113,601	166,001	227,501	100%	1.00E-06	520,201	837,801	1,000,000	68%	4.25E+02	6.5511
12	DERand rand	8,501	9,021	9,901	100%	1.00E-06	22,801	62,801	200,000	80%	3.55E+01	69,201	287,841	500,000	52%	1.07E+02	1,000,000	1,000,000	1,000,000	0%	6.69E+02	6.8702
13	SIMPSA 2	7,101	19,451	47,901	100%	1.00E-06	16,201	56,941	149,501	100%	1.00E-06	209,801	305,791	429,201	100%	1.00E-06	921,401	992,231	1,000,000	12%	3.10E+03	7.4886
14	CoDE	25,001	25,801	27,001	100%	1.00E-06	76,001	80,101	85,001	100%	1.00E-06	338,001	356,601	376,001	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	3.24E+03	7.5105
15	Lbest	100,000	100,000	100,000	0%	4.87E+02	200,000	200,000	200,000	0%	1.26E+03	500,000	500,000	500,000	0%	5.99E+03	1,000,000	1,000,000	1,000,000	0%	1.65E+04	8.3845
16	GBest	100,000	100,000	100,000	0%	7.46E+02	200,000	200,000	200,000	0%	2.07E+03	500,000	500,000	500,000	0%	7.78E+03	1,000,000	1,000,000	1,000,000	0%	1.97E+04	8.4812
17	DEBest	6,601	53,771	100,000	48%	8.29E+01	38,601	77,391	200,000	80%	2.37E+01	500,000	500,000	500,000	0%	1.19E+06	1,000,000	1,000,000	1,000,000	0%	2.10E+04	10.0843
18	Glob-VNS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	Gauss-VNS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 3: Results for Ackley's functions

ACKLEY'S

		10					20					50					100					Overall
		eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	score
1	DE-VNS	9,401	10,441	13,501	100%	1.00E-06	22,701	26,241	30,001	100%	1.00E-06	78,101	86,231	98,901	100%	1.00E-06	**182,801**	**204,521**	**237,601**	**100%**	**1.00E-06**	-2.2128
2	DERand 0.5 0.3	11,401	11,771	12,101	100%	1.00E-06	27,501	28,211	29,001	100%	1.00E-06	94,301	96,011	97,501	100%	1.00E-06	293,201	297,981	304,001	100%	1.00E-06	-2.1257
3	DERand 0.5 0.5	10,701	11,091	11,401	100%	1.00E-06	28,001	29,021	29,701	100%	1.00E-06	118,401	120,331	122,101	100%	1.00E-06	458,101	465,251	473,001	100%	1.00E-06	-2.0169
4	DEbr18	18,301	19,251	20,001	100%	1.00E-06	55,401	57,261	58,601	100%	1.00E-06	208,201	211,941	216,901	100%	1.00E-06	524,001	530,321	537,501	100%	1.00E-06	-1.8437
5	CoDE	30,001	30,501	31,001	100%	1.00E-06	80,001	83,101	86,001	100%	1.00E-06	228,001	239,601	248,001	100%	1.00E-06	444,001	457,901	470,001	100%	1.00E-06	-1.7805
6	Gauss-VNS	-	50,149	-	100%	1.00E-05	-	158,412	-	100%	1.00E-05	-	1,143,721	-	100%	1.00E-05	-	-	-	-	-	-0.4460
7	DERand rand	10,701	11,351	11,701	100%	1.00E-06	25,301	26,451	27,501	100%	1.00E-06	**69,601**	**71,561**	**73,201**	**100%**	**1.00E-06**	142,101	230,500	1,000,000	90%	3.56E-04	-0.0838
8	SA 100 100	8,001	9,001	10,001	100%	1.00E-06	18,001	19,801	22,001	100%	1.00E-06	148,001	160,701	168,001	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	1.02E-04	0.0104
9	SA 500 100	19,001	20,501	22,001	100%	1.00E-06	36,001	38,601	42,001	100%	1.00E-06	170,001	185,301	203,001	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	1.05E-04	0.0262
10	SA 100 500	18,001	20,001	21,001	100%	1.00E-06	37,001	38,601	40,001	100%	1.00E-06	158,001	191,501	208,001	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	1.12E-04	0.0529
11	SAR	**7,801**	**8,651**	**10,101**	**100%**	**1.00E-06**	19,201	21,571	24,401	100%	1.00E-06	148,401	179,101	214,001	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	1.30E-04	0.1158
12	Glob-VNS	-	188,670	-	100%	1.00E-05	-	433,194	-	100%	1.00E-05	-	4,791,075	-	100%	1.00E-05	-	-	-	-	-	0.1346
13	SIMPSA 2	10,701	21,231	38,001	100%	1.00E-06	200,000	200,000	200,000	0%	2.73E-06	500,000	500,000	500,000	0%	1.85E-05	1,000,000	1,000,000	1,000,000	0%	1.79E-04	0.3020
14	SIMPSA 1	9,501	16,941	33,601	100%	1.00E-06	200,000	200,000	200,000	0%	2.79E-06	500,000	500,000	500,000	0%	1.87E-05	1,000,000	1,000,000	1,000,000	0%	9.10E-04	0.9693
15	GBest	86,301	90,031	100,000	80%	2.02E-06	200,000	200,000	200,000	0%	4.98E-04	500,000	500,000	500,000	0%	6.05E-04	1,000,000	1,000,000	1,000,000	0%	4.33E-03	1.7353
16	JADE	9,001	9,801	10,001	100%	1.00E-06	**15,001**	**16,201**	**18,001**	**100%**	**1.00E-06**	30,001	78,001	500,000	88%	8.70E-02	1,000,000	1,000,000	1,000,000	0%	1.53E+00	4.1885
17	LBest	100,000	100,000	100,000	0%	4.73E-04	200,000	200,000	200,000	0%	2.12E-03	500,000	500,000	500,000	0%	5.36E-03	1,000,000	1,000,000	1,000,000	0%	2.37E+00	4.3765
18	SaDE	9,001	9,901	10,001	100%	1.00E-06	19,001	20,601	21,001	100%	1.00E-06	500,000	500,000	500,000	0%	1.25E+00	1,000,000	1,000,000	1,000,000	0%	2.55E+00	4.5798
19	DEBest	8,701	9,411	10,001	100%	1.00E-06	53,801	55,211	58,001	100%	1.00E-06	500,000	500,000	500,000	0%	6.34E-02	1,000,000	1,000,000	1,000,000	0%	1.20E+01	5.0825

Table 4: Results for Griewank's functions

GRIEWANK'S

		10					20					50					100					Overall
		eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	score
1	DERand 0.5 0.3	**16,101**	**17,501**	**20,001**	**100%**	**1.00E-06**	**21,301**	**23,971**	**26,901**	**100%**	**1.00E-06**	**68,401**	**71,001**	**74,001**	**100%**	**1.00E-06**	212,601	216,321	218,501	100%	1.00E-06	-2.1850
2	DE-VNS	11,801	22,101	47,401	100%	1.00E-06	22,701	24,961	28,101	100%	1.00E-06	67,401	73,821	80,201	100%	1.00E-06	**158,801**	**168,961**	**180,701**	**100%**	1.00E-06	-2.1789
3	DERand 0.5 0.5	20,801	27,611	34,301	100%	1.00E-06	22,401	24,541	28,401	100%	1.00E-06	86,901	89,231	91,601	100%	1.00E-06	329,401	338,611	344,201	100%	1.00E-06	-2.0382
4	Roulette	31,401	35,861	43,201	100%	1.00E-06	46,001	51,441	56,701	100%	1.00E-06	153,701	158,431	172,201	100%	1.00E-06	383,801	390,411	399,401	100%	1.00E-06	-1.8784
5	DEbr18	57,001	63,201	69,001	100%	1.00E-06	63,001	79,801	104,001	100%	1.00E-06	165,001	173,501	186,001	100%	1.00E-06	320,001	330,301	339,001	100%	1.00E-06	-1.7674
6	DERand rand	13,601	15,323	100,000	88%	2.40E-03	19,401	20,468	200,000	92%	7.30E-04	50,701	55,001	500,000	92%	1.03E-03	99,501	284,421	1,000,000	80%	2.20E-03	1.0722
7	JADE	19,001	31,801	101,001	92%	1.08E-03	11,001	108,001	201,001	76%	2.96E-03	21,001	261,501	501,001	68%	5.39E-03	43,001	332,001	1,001,001	72%	5.10E-03	1.8100
8	SaDE	16,001	38,201	100,000	80%	2.46E-03	13,001	70,901	200,000	68%	3.60E-03	40,001	226,301	500,000	60%	1.20E-02	100,001	195,401	1,000,000	92%	1.72E-03	1.9022
9	LBest	64,801	96,571	100,000	12%	4.22E-02	113,401	161,741	200,000	60%	4.93E-03	311,101	335,071	500,000	88%	8.22E-06	664,401	803,081	1,000,000	60%	1.18E-04	2.6518
10	SIMPSA 2	100,000	100,000	100,000	0%	5.10E-02	11,201	125,061	200,000	40%	1.60E-02	34,201	223,591	500,000	60%	2.90E-02	69,101	537,331	1,000,000	52%	3.00E-02	2.9547
11	GBest	100,000	100,000	100,000	0%	5.90E-02	87,801	172,781	200,000	32%	2.70E-02	414,701	467,771	500,000	40%	1.49E-02	923,701	992,371	1,000,000	12%	4.03E-03	3.0016
12	SAR	5,501	90,641	100,000	8%	4.43E-02	14,001	165,101	200,000	20%	4.76E-02	54,501	366,811	500,000	32%	2.19E-02	913,201	978,101	1,000,000	32%	8.13E-03	3.0147
13	SA 100 500	100,000	100,000	100,000	0%	5.04E-02	200,000	200,000	200,000	0%	4.52E-02	66,001	249,601	500,000	60%	9.35E-03	823,001	941,901	1,000,000	40%	1.40E-02	3.0548
14	SA 500 100	100,000	100,000	100,000	0%	5.71E-02	26,001	183,501	200,000	8%	5.67E-02	70,001	390,601	500,000	28%	1.79E-02	845,001	920,301	1,000,000	80%	3.93E-03	3.1028
15	SA 100 100	100,000	100,000	100,000	0%	6.30E-02	29,001	183,801	200,000	8%	4.29E-02	50,001	415,101	500,000	20%	4.06E-02	811,001	928,101	1,000,000	60%	3.70E-03	3.1447
16	SIMPSA 1	100,000	100,000	100,000	0%	9.10E-02	200,000	200,000	200,000	0%	3.50E-02	36,901	362,721	500,000	20%	2.70E-02	79,601	360,541	1,000,000	68%	6.10E-03	3.1696
17	DEBest	29,601	51,121	73,301	100%	1.00E-06	41,801	69,971	90,301	100%	1.00E-06	500,000	500,000	500,000	0%	3.98E-01	1,000,000	1,000,000	1,000,000	0%	4.11E+02	6.6141
18	Glob-VNS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	Gauss-VNS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 5: Results for Rastrigin's functions

RASTRIGIN'S

		10					20					50					100					Overall
		eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	score
1	DE-VNS	10,001	10,691	11,401	100%	1.00E-06	25,001	27,811	31,101	100%	1.00E-06	85,001	89,741	95,601	100%	1.00E-06	**200,101**	**220,541**	**243,701**	**100%**	**1.00E-06**	-2.1898
2	JADE	13,001	13,901	15,001	100%	1.00E-06	33,001	34,301	35,001	100%	1.00E-06	108,001	111,201	114,001	100%	1.00E-06	231,001	247,801	257,001	100%	1.00E-06	-2.1075
3	DEbr18	15,101	15,641	16,801	100%	1.00E-06	44,401	45,261	46,101	100%	1.00E-06	165,601	167,591	171,301	100%	1.00E-06	414,401	418,761	422,501	100%	1.00E-06	-1.9444
4	SA 100 100	**5,001**	**9,001**	**15,001**	**100%**	**1.00E-06**	**12,001**	**21,201**	**29,001**	**100%**	**1.00E-06**	**67,001**	**75,801**	**92,001**	**100%**	**1.00E-06**	890,001	983,601	1,000,000	20%	1.05E-06	-1.8597
5	SAR	4,601	9,651	17,701	100%	1.00E-06	16,101	22,531	35,101	100%	1.00E-06	74,501	87,471	113,001	100%	1.00E-06	959,201	992,561	1,000,000	32%	1.01E-06	-1.8595
6	SA 100 500	12,001	14,001	20,001	100%	1.00E-06	25,001	29,501	37,001	100%	1.00E-06	80,001	92,201	100,001	100%	1.00E-06	941,001	990,301	1,000,000	28%	1.02E-06	-1.8289
7	SA 500 100	13,001	13,501	15,001	100%	1.00E-06	25,001	29,601	37,001	100%	1.00E-06	80,001	92,301	102,001	100%	1.00E-06	957,001	992,401	1,000,000	28%	1.28E-06	-1.7600
8	Glob-VNS	-	52,471	-	100%	1.00E-05	-	213,597	-	100%	1.00E-05	-	1,334,842	-	100%	1.00E-05	-	5,388,075	-	100%	1.00E-05	-0.0155
9	Gauss-VNS	-	85,589	-	100%	1.00E-05	-	287,075	-	100%	1.00E-05	-	1,524,701	-	100%	1.00E-05	-	6,248,753	-	100%	1.00E-05	0.0641
10	SIMPSA 1	8,901	22,381	50,201	100%	1.00E-06	41,801	64,191	107,901	100%	1.00E-06	221,501	276,351	324,001	100%	1.00E-06	816,501	905,611	1,000,000	92%	9.98E-02	2.9563
11	SIMPSA 2	13,401	24,181	37,201	100%	1.00E-06	41,801	65,221	97,401	100%	1.00E-06	254,901	325,671	383,701	100%	1.00E-06	957,901	984,931	1,000,000	52%	8.90E-01	3.9428
12	SaDE	13,001	14,901	16,001	100%	1.00E-06	34,001	37,201	42,001	100%	1.00E-06	125,001	206,401	500,000	80%	1.99E-01	1,000,000	1,000,000	1,000,000	0%	3.28E+00	4.5266
13	DERand rand	10,401	11,531	12,301	100%	1.00E-06	200,000	200,000	200,000	40%	6.96E-01	500,000	500,000	500,000	0%	6.57E+00	1,000,000	1,000,000	1,000,000	0%	3.28E+01	5.6031
14	CoDE	31,001	34,001	36,001	100%	1.00E-06	117,001	124,901	132,001	100%	1.00E-06	500,000	500,000	500,000	0%	3.04E+01	1,000,000	1,000,000	1,000,000	0%	2.12E+02	6.3851
15	GBest	100,000	100,000	100,000	0%	2.19E+00	200,000	200,000	200,000	0%	1.07E+01	500,000	500,000	500,000	0%	5.48E+01	1,000,000	1,000,000	1,000,000	0%	1.79E+02	6.3923
16	LBest	100,000	100,000	100,000	0%	3.58E+00	200,000	200,000	200,000	0%	1.86E+01	500,000	500,000	500,000	0%	1.01E+02	1,000,000	1,000,000	1,000,000	0%	2.67E+02	6.5908
17	DERand 0.5 0.3	11,601	13,151	14,201	100%	1.00E-06	73,601	78,551	86,601	100%	1.00E-06	500,000	500,000	500,000	0%	1.43E+02	1,000,000	1,000,000	1,000,000	0%	5.58E+02	6.8457
18	DEBest	21,701	25,001	28,301	100%	1.00E-06	200,000	200,000	200,000	0%	4.41E+01	500,000	500,000	500,000	0%	3.59E+02	1,000,000	1,000,000	1,000,000	0%	5.37E+02	6.9730
19	DERand 0.5 0.5	19,901	23,451	28,601	100%	1.00E-06	200,000	200,000	200,000	0%	2.44E+01	500,000	500,000	500,000	0%	2.48E+02	1,000,000	1,000,000	1,000,000	0%	7.42E+02	7.0064

Table 6: Results for MPE functions

MPE

		10					20					50					100					Overall
		eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	score
1	DE-VNS	6,101	6,831	7,601	100%	1.00E-06	16,801	19,121	20,901	100%	1.00E-06	52,701	59,221	66,701	100%	1.00E-06	122,901	143,831	160,701	100%	1.00E-06	-2.3704
2	DEbr18	27,901	27,901	27,901	100%	1.00E-06	66,401	72,041	75,901	100%	1.00E-06	249,401	262,921	267,301	100%	1.00E-06	623,801	646,181	663,601	100%	1.00E-06	-1.7420
3	Glob-VNS	-	8,102	-	100%	1.00E-05	-	26,647	-	100%	1.00E-05	-	202,280	-	100%	1.00E-05	-	830,343	-	100%	1.00E-05	-0.8389
4	Gauss-VNS	-	5,015	-	100%	1.00E-05	-	21,172	-	100%	1.00E-05	-	143,309	-	100%	1.00E-05	-	1,183,873	-	100%	1.00E-05	-0.7887
5	SA 100 100	5,001	14,101	21,001	100%	1.00E-06	200,000	200,000	200,000	0%	2.35E-06	500,000	500,000	500,000	0%	2.58E-05	1,000,000	1,000,000	1,000,000	0%	1.43E-04	0.2339
6	SAR	5,501	9,121	11,701	100%	1.00E-06	200,000	200,000	200,000	0%	2.07E-06	500,000	500,000	500,000	0%	2.63E-05	1,000,000	1,000,000	1,000,000	0%	1.44E-04	0.2368
7	SA 100 500	13,001	15,801	21,001	100%	1.00E-06	200,000	200,000	200,000	0%	2.30E-06	500,000	500,000	500,000	0%	2.49E-05	1,000,000	1,000,000	1,000,000	0%	1.47E-04	0.2409
8	SA 500 100	9,001	15,601	24,001	100%	1.00E-06	200,000	200,000	200,000	0%	2.10E-06	500,000	500,000	500,000	0%	2.56E-05	1,000,000	1,000,000	1,000,000	0%	1.47E-04	0.2429
9	JADE	10,001	11,301	13,001	100%	1.00E-06	27,001	97,601	200,000	60%	8.22E-02	89,001	131,901	500,000	80%	8.23E-02	190,001	520,901	1,000,000	60%	3.22E-02	2.8954
10	SaDE	11,001	11,701	12,001	100%	1.00E-06	25,001	43,501	200,000	88%	8.17E-03	88,001	460,601	500,000	12%	8.20E-02	1,000,000	1,000,000	1,000,000	0%	3.04E-01	3.5813
11	DERand rand	43,101	54,751	64,901	100%	1.00E-06	206,701	241,821	276,901	100%	1.00E-06	500,000	500,000	500,000	0%	5.03E-02	1,000,000	1,000,000	1,000,000	0%	3.98E-01	3.6515
12	SIMPSA 2	17,401	46,341	98,801	100%	1.00E-06	39,501	74,311	130,801	100%	1.00E-06	308,601	398,421	500,000	68%	1.42E-01	662,801	872,621	1,000,101	80%	9.41E-01	3.9706
13	SIMPSA 1	15,201	38,201	99,301	100%	1.00E-06	45,001	84,601	152,001	100%	1.00E-06	272,301	361,081	500,000	80%	1.19E-01	793,401	918,871	1,000,000	72%	1.21E+00	4.0775
14	CoDE	21,001	22,901	24,001	100%	1.00E-06	69,001	73,101	77,001	100%	1.00E-06	372,001	390,201	402,001	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	4.23E+00	4.6263
15	LBest	100,000	100,000	100,000	0%	3.04E-01	200,000	200,000	200,000	0%	8.00E-01	500,000	500,000	500,000	0%	2.27E+00	1,000,000	1,000,000	1,000,000	0%	4.65E+00	4.9043
16	GBest	100,000	100,000	100,000	0%	3.26E-01	200,000	200,000	200,000	0%	8.43E-01	500,000	500,000	500,000	0%	2.34E+00	1,000,000	1,000,000	1,000,000	0%	4.83E+00	4.9213
17	DERand 0.5 0.3	45,001	49,761	51,601	100%	1.00E-06	200,000	200,000	200,000	0%	6.74E-02	500,000	500,000	500,000	0%	1.05E+01	1,000,000	1,000,000	1,000,000	0%	3.37E+01	5.6462
18	DERand 0.5 0.5	100,000	100,000	100,000	0%	6.38E-04	200,000	200,000	200,000	0%	1.90E+00	500,000	500,000	500,000	0%	1.83E+01	1,000,000	1,000,000	1,000,000	0%	4.55E+01	5.8171
19	DEBest	8,501	16,261	23,401	100%	1.00E-06	200,000	200,000	200,000	0%	1.5607	500,000	500,000	500,000	0%	1.75E+01	1,000,000	1,000,000	1,000,000	0%	5.27E+01	5.8556

Table 7: Results for Rosenbrock's functions

ROSENBROCK'S

		10					20					30					50					Overall
		eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	score
1	DEbr18	29,001	33,001	39,001	100%	1.00E-06	132,001	143,101	150,001	100%	1.00E-06	280,001	333,601	359,001	100%	1.00E-06	619,001	946,201	1,073,001	100%	1.00E-06	-1.3926
2	DE-VNS	45,001	53,901	69,001	100%	1.00E-06	183,001	234,301	415,001	100%	1.00E-06	281,001	424,301	516,001	100%	1.00E-06	729,001	1,142,001	1,442,001	100%	1.00E-06	-1.2669
3	DERand rand	57,001	62,001	65,001	100%	1.00E-06	233,001	243,701	261,001	100%	1.00E-06	568,001	577,401	589,001	100%	1.00E-06	1,766,001	1,807,701	1,870,001	100%	1.00E-06	-1.1320
4	SIMPSA 1	13,000	49,700	264,000	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	5.64E-06	1,500,000	1,500,000	1,500,000	0%	5.06E-04	2,500,000	2,500,000	2,500,000	0%	1.76E-02	2.9578
5	SIMPSA 2	23,000	56,300	317,000	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	4.43E-06	1,500,000	1,500,000	1,500,000	0%	4.41E-04	2,500,000	2,500,000	2,500,000	0%	1.81E-02	2.9662
6	CoDE	52,001	55,501	59,001	100%	1.00E-06	193,001	202,301	218,001	100%	1.00E-06	410,001	426,401	446,001	100%	1.00E-06	853,001	1,167,601	2,500,000	84%	3.99E-01	3.9693
7	JADE	15,001	33,801	101,001	100%	1.00E-06	38,001	43,301	52,001	100%	1.00E-06	70,001	79,401	96,001	100%	1.00E-06	157,001	645,901	2,500,000	80%	7.91E-01	4.0094
8	DERand 0.5 0.5	500,000	500,000	500,000	0%	4.54E+00	1,000,000	1,000,000	1,000,000	0%	8.77E-02	1,500,000	1,500,000	1,500,000	0%	6.49E-02	2,244,001	2,463,801	2,500,000	32%	2.74E-04	5.3705
9	SaDE	263,001	434,401	501,001	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	8.00E-01	1,500,000	1,500,000	1,500,000	0%	2.18E+00	2,500,000	2,500,000	2,500,000	0%	9.48E+00	5.7945
10	LBest	500,000	500,000	500,000	0%	4.37E-02	1,000,000	1,000,000	1,000,000	0%	6.02E-01	1,500,000	1,500,000	1,500,000	0%	4.15E+00	2,500,000	2,500,000	2,500,000	0%	7.79E+00	5.7987
11	SA 100 500	500,000	500,000	500,000	0%	4.70E+00	1,000,000	1,000,000	1,000,000	0%	4.55E+00	1,500,000	1,500,000	1,500,000	0%	6.21E+00	2,500,000	2,500,000	2,500,000	0%	6.03E+00	6.0312
12	DERand 0.5 0.3	455,001	494,401	500,000	20%	5.68E-02	1,000,000	1,000,000	1,000,000	0%	1.99E+00	1,500,000	1,500,000	1,500,000	0%	6.25E+00	2,500,000	2,500,000	2,500,000	0%	1.81E+01	6.1202
13	DEBest	104,001	112,401	125,001	100%	1.00E-06	551,001	572,601	598,001	100%	1.00E-06	1,500,000	1,500,000	1,500,000	0%	1.15E-04	2,500,000	2,500,000	2,500,000	0%	2.71E+01	6.1323
14	SA 500 100	500,000	500,000	500,000	0%	8.67E+00	1,000,000	1,000,000	1,000,000	0%	4.81E+00	1,500,000	1,500,000	1,500,000	0%	4.67E+00	2,500,000	2,500,000	2,500,000	0%	9.05E+00	6.1335
15	SA 100 100	500,000	500,000	500,000	0%	4.98E+00	1,000,000	1,000,000	1,000,000	0%	8.72E+00	1,500,000	1,500,000	1,500,000	0%	5.76E+00	2,500,000	2,500,000	2,500,000	0%	1.01E+01	6.1697
16	GBest	500,000	500,000	500,000	0%	4.52E-01	1,000,000	1,000,000	1,000,000	0%	2.77E+00	1,500,000	1,500,000	1,500,000	0%	4.81E+00	2,500,000	2,500,000	2,500,000	0%	4.18E+01	6.3962
17	SAR	500,000	500,000	500,000	0%	5.42E+00	1,000,000	1,000,000	1,000,000	0%	5.01E+00	1,500,000	1,500,000	1,500,000	0%	4.49E+00	2,500,000	2,500,000	2,500,000	0%	3.56E+01	6.4021
	Gauss-VNS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
	Glob-VNS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 8: Results for Sphere functions

SPHERE

		10					20					50					100					Overall
		eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	eval-min	eval- avg	eval max	SR	fmin	score
1	JADE	6,001	6,301	7,001	100%	1.00E-06	11,001	11,101	12,001	100%	1.00E-06	**21,001**	**22,001**	**23,001**	**100%**	**1.00E-06**	**44,001**	**45,401**	**48,001**	**100%**	**1.00E-06**	-2.6821
2	SaDE	7,001	7,001	7,001	100%	1.00E-06	13,001	14,201	16,001	100%	1.00E-06	40,001	43,301	46,001	100%	1.00E-06	104,001	110,701	117,001	100%	1.00E-06	-2.4707
3	SIMPSA 2	**4,000**	**4,900**	**6,000**	**100%**	**1.00E-06**	12,000	14,400	15,000	100%	1.00E-06	42,000	50,700	64,000	100%	1.00E-06	137,000	181,400	255,000	100%	1.00E-06	-2.3938
4	SIMPSA 1	5,000	5,900	7,000	100%	1.00E-06	13,000	15,400	18,000	100%	1.00E-06	48,000	58,000	67,000	100%	1.00E-06	135,000	234,000	520,000	100%	1.00E-06	-2.3134
5	DE-VNS	8,001	8,201	9,001	100%	1.00E-06	19,001	21,001	22,001	100%	1.00E-06	64,001	72,101	83,001	100%	1.00E-06	153,001	165,001	189,001	100%	1.00E-06	-2.3043
6	DERand 0.5 0.3	8,001	8,501	9,001	100%	1.00E-06	20,001	20,401	21,001	100%	1.00E-06	70,001	70,801	71,001	100%	1.00E-06	224,001	226,901	230,001	100%	1.00E-06	-2.2553
7	DERand 0.5 0.5	8,001	8,001	8,001	100%	1.00E-06	20,001	21,101	22,001	100%	1.00E-06	88,001	89,401	91,001	100%	1.00E-06	347,001	353,501	364,001	100%	1.00E-06	-2.1440
8	DEbr18	13,001	13,701	15,001	100%	1.00E-06	39,001	40,601	42,001	100%	1.00E-06	153,001	155,901	162,001	100%	1.00E-06	394,001	397,801	402,001	100%	1.00E-06	-1.9790
9	CoDE	19,001	20,701	21,001	100%	1.00E-06	57,001	58,301	60,001	100%	1.00E-06	167,001	172,101	178,001	100%	1.00E-06	331,001	338,101	344,001	100%	1.00E-06	-1.9278
10	SA 100 100	5,001	5,101	6,001	100%	1.00E-06	**9,001**	**10,701**	**12,001**	**100%**	**1.00E-06**	42,001	48,101	53,001	100%	1.00E-06	962,001	994,001	1,000,000	20%	1.04E-06	-1.9092
11	SA 500 100	11,001	12,201	13,001	100%	1.00E-06	23,001	24,401	26,001	100%	1.00E-06	67,001	74,101	91,001	100%	1.00E-06	980,001	997,501	1,000,000	20%	1.16E-06	-1.8107
12	SA 100 500	11,001	11,901	13,001	100%	1.00E-06	23,001	24,901	27,001	100%	1.00E-06	65,001	71,001	75,001	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	1.49E-06	-1.7264
13	DERand rand	13,001	13,701	14,001	100%	1.00E-06	43,001	44,801	46,001	100%	1.00E-06	242,001	247,501	253,001	100%	1.00E-06	991,001	999,201	1,000,000	20%	1.28E-06	-1.6707
14	LBest	35,301	39,231	42,001	100%	1.00E-06	85,801	91,181	95,401	100%	1.00E-06	258,501	265,941	270,901	100%	1.00E-06	352,101	812,061	1,000,000	40%	1.08E-06	-1.6468
15	SAR.	5,001	5,201	6,001	100%	1.00E-06	9,001	11,501	13,001	100%	1.00E-06	73,001	98,901	115,001	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	2.22E-04	0.3462
16	GBest	28,101	31,031	32,201	100%	1.00E-06	66,801	72,641	76,501	100%	1.00E-06	208,701	212,951	220,701	100%	1.00E-06	1,000,000	1,000,000	1,000,000	0%	4.70E+00	4.6721
17	DEBest	24,001	25,501	27,001	100%	1.00E-06	109,001	114,601	119,001	100%	1.00E-06	500,000	500,000	500,000	0%	7.30E-02	1,000,000	1,000,000	1,000,000	0%	4.63E+02	6.6655
18	Gauss-VNS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
19	Glob-VNS	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

$DE/rand/1/bin$ versions also achieved tolerance in less than 10,000 FEs at 10 D problem; DERand 0.5 0.3 and DERand 0.5 0.5 perform very well at 20 D problem; the third variant, DERand rand, is also fast, but it struggles with the precision SR; in solving 50 and 100 dimensions, SaDE is the second best.

Ackley functions (Table 3). For lower dimensions of, SA 100 100, SAR, JADE, and SaDE are notably fast. DEBest has shown good result only at 10 D Ackley. At large dimensions, algorithms based on Differential Evolution outperforms others, especially good are DE-VNS and DERand 0.5 0.3.

Greiwank functions (Table 4). The most algorithms have difficulties in reaching tolerance, even at low dimensions; 10 algorithms do not have SR equal to 100% at any dimension. DERand 0.5 0.3, DE-VNS, and DERand 0.5 0.5 have shown very good results in both SR and speed. Code and DEbr18 are reaching the tolerance slower, but within the defined test setting. JADE is very fast, especially for $D = 20$ and 50, but it doesn't have maximal success rate at final problem.

Rastrigin functions (Table 5). Only DE-VNS, JADE and DEbr18 solved all instances (with SR 100%). Glob VNS and Gauss VNS also reach 100%, but with more iterations than it is defined in test settings. All four Simulated Annealing algorithms are fast at 10, 20, and 50 D problems. SA 100 100 is the fastest of all algorithms on these dimensions for Rastrigin. On 100 D they have SR 100% or lower. DE-VNS and JADE have shown great results at the largest dimension considered in this paper.

MPE functions (Table 6). VNS-based algorithms (Gauss-VNS, Glob-VNS and DE-VNS) and DEbr18 have $SR = 100\%$ at MPE for all dimensions. Gauss-VNS is the fastest at 10 D. At other dimensions DE-VNS outperforms other algorithms. At smaller dimension problems all VNS-based algorithms have similar number of FEs, while at larger DE-VNS is faster than others significantly.

Rosenbrock functions (Table 7). DEbr18, DE-VNS, and DERand rand have shown best performance. All of these algorithms have SR 100% and they are among the fastest algorithms. Theirs number of FEs is of the same order of magnitude as CoDE's, but CoDE doesn't have SR 100% at 50 D Rosenbrock. JaDE is considerably faster than others on 20 and 30 dimensions, but at maximum dimensions has the same problem as CoDE.

Sphere function (Table 8). Tested algorithms do not have greater problems with achieving SR 100%, so the focus of analysis is on the number of FEs. Predefined tolerance at 10 dimensional is reached within 5000 FEs by SIMPSA 2, SA 100 100, and SAR. These algorithms and JADE are the fastest on 20 D problem. For larger dimensions, JADE is by far the best.

Summarizing results reported in Tables 2 to 8, one can get the following observations:

(i) In terms of precision and robustness, DE-VNS and DEbr18 algorithms have shown the best properties: only these two algorithms solved all test problems (on all test functions, for all dimensions), i.e., their $SR = 100\%$.

(ii) DE-VNS is clearly the best algorithm for global optimization, since its worst behavior is on convex test functions (Tables 7 and 8). DE-VNS is at 10 out of 28 problems ranked as the fastest. It should be emphasized that DE-VNS achieved desired tolerance in the fewest FEs for all largest non-convex test instances.

(iii) These two algorithms are followed by CoDE that does not reach 100% success rate at 5 out of 28 problems.

(iv) The second fastest method is JADE. It was the fastest at 7 problems, but most of them are convex.

Rank statistics. In Table 9 and at Figure 8, all 19 algorithms are ranked for each test function. The last column, "*all*", is the arithmetic mean of rankings. DE-VNS is ranked as the first, with mean ranking 1.86, and with the top results for multimodal problems. The second is DEbr18 with rank = 3.71. Third and fifth are Gauss and Glob-VNS. These results should not be taken for granted, because only data for three test functions were available from papers and compared. Self-adaptive DE algorithms JADE, SaDE and CoDE are ranked fourth, seventh, and thirteenth. JADE has proven the best at Sphere and Rastrigin's function in accordance with an overall score. SaDE performs well at Sphere and Schwefel's function. On these problems CoDE algorithm has a slow convergence rate. CoDE algorithm almost always converged toward the solution, but convergence speed is not very fast. Other DE algorithms are ranked sixth, eighth and tenth. Among them, DERand 0.5 0.3 is proven as the best, although its results differ from good (Griewank's, Ackley's, Schwefel's), till very bad (Rastrigin's, MPE). SA 100 100 is ranked ninth, and it has shown the best performance among SA algorithms. It is followed by SA 100 500, and SA 500 100, while SAR has lowest ranking. SA has generally shown well in case of MPE and Rastrigin's function. SIMPSA algorithms are ranked twelfth and fifteenth. Better ranked SIMPSA 2 have shown as good for unimodal problems, it is ranked fifth and third at Rosenbrock and Sphere. PSO algorithms LBest and GBest are next. They have shown a rather slow convergence. They perform best for Griewank's function. In last place is DEBest algorithm, based on "$DE/best/1/bin$", due its characteristic that converges prematurely at multimodal problems.

Table 9: Rank of compared heuristics

		Schwefel's	Ackley's	Griewank's	Rastrigin's	MPE	Rosenbrock's	Sphere	All
1	DE-VNS	1	1	2	1	1	2	5	1.8571
2	DEbr18	4	4	4	3	2	1	8	3.7143
3	Gauss-VNS		6		9	4			6.3333
4	JADE	10	16	7	2	9	7	1	7.4286
5	Glob-VNS		12		8	3			7.6667
6	DERand 0.5 0.3	3	2	1	17	17	12	6	8.2857
7	SaDE	2	18	8	12	10	9	2	8.7143
8	DERand 0.5 0.5	5	3	3	19	18	8	7	9.0000
9	SA 100 100	6	8	15	4	5	15	10	9.0000
10	DERand rand	12	7	6	13	11	3	13	9.2857
11	SA 100 500	7	10	13	6	7	11	12	9.4286
12	SIMPSA 2	13	13	10	11	12	5	3	9.5714
13	CoDE	14	5	5	14	14	6	9	9.5714
14	SA 500 100	8	9	14	7	8	14	11	10.1429
15	SIMPSA 1	11	14	16	10	13	4	4	10.2857
16	SAR	9	11	12	5	6	17	15	10.7143
17	LBest	15	17	9	16	15	10	14	13.7143
18	GBest	16	15	11	15	16	16	16	15.0000
19	DEBest	17	19	17	18	19	13	17	17.1429

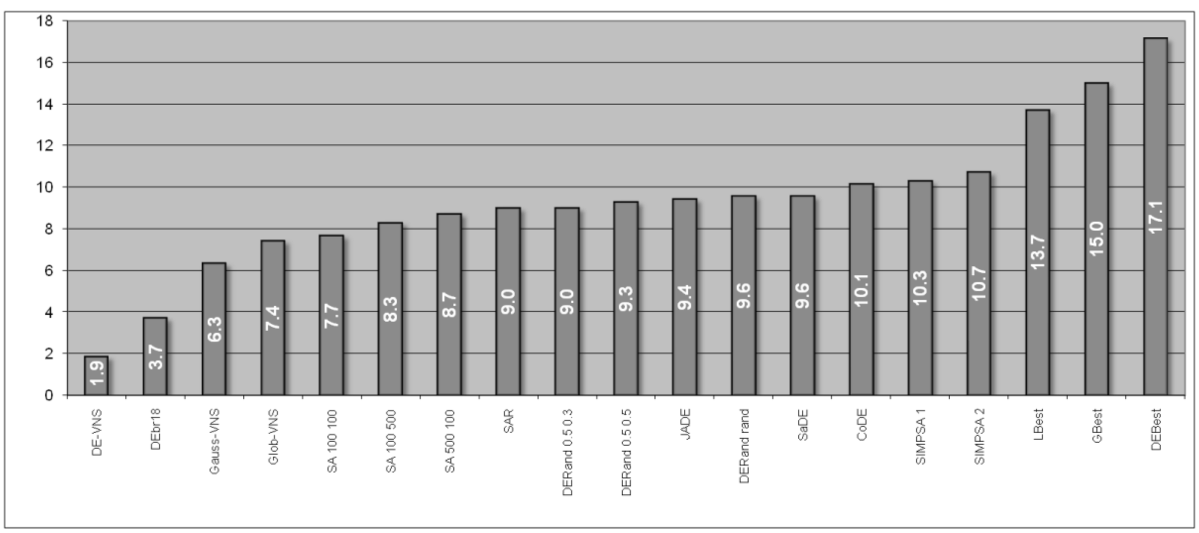


Figure 8: The average rank of comparing heuristics

Composite scores for all algorithms are presented in Table 10. For each 50 and 100 dimensional problem we compare the speed of convergence. Five algorithms that have proved to be the fastest for that problem are taken into consideration. An additional condition is that the algorithm must have success rate equal to 100% of that problem. Data are also presented at Figure 9.

Table 10: Composite score of compared heuristics

		Schwefel's	Ackley's	Griewank's	Rastrigin's	MPE	Rosenbrock's	Sphere	All
1	DE-VNS	-2.2564	-2.2128	-2.1789	-2.1898	-2.3704	-1.2669	-2.3043	-14.7795
2	DEbr18	-1.9517	-1.8437	-1.8784	-1.9444	-1.7420	-1.3926	-1.9790	-12.7318
3	Gauss-VNS		-0.4460		0.0641	-0.7887			-1.1707
4	Glob-VNS		0.1346		-0.0155	-0.8389			-0.7197
5	SA 100 100	-1.2237	0.0104	3.1447	-1.8597	0.2339	6.1697	-1.9092	4.5662
6	SA 100 500	-1.1667	0.0529	3.0548	-1.8289	0.2409	6.0312	-1.7264	4.6579
7	SA 500 100	-1.1576	0.0262	3.1028	-1.7600	0.2429	6.1335	-1.8107	4.7772
8	SAR	-1.1050	0.1158	3.0147	-1.8595	0.2368	6.4021	0.3462	7.1513
9	DERand 0.5 0.3	-2.0942	-2.1257	-2.1850	6.8457	5.6462	6.1202	-2.2553	9.9519
10	DERand 0.5 0.5	-1.8861	-2.0169	-2.0382	7.0064	5.8171	5.3705	-2.1440	10.1088
11	JADE	6.3194	4.1885	1.4897	-2.1075	2.8954	4.0094	-2.6821	14.1128
12	DERand rand	6.8702	-0.0830	0.8962	5.6031	3.6515	-1.1320	-1.6707	14.1353
13	SaDE	-2.1753	4.5798	1.9022	4.5266	3.5813	5.7945	-2.4707	15.7384
14	CoDE	7.5105	-1.7805	-1.7674	6.3851	4.6263	3.9693	-1.9278	17.0155
15	SIMPSA 1	6.5511	0.9693	3.1696	2.9563	4.0775	2.9578	-2.3134	18.3682
16	SIMPSA 2	7.4886	0.3020	2.9547	3.9428	3.9706	2.9662	-2.3938	19.2311
17	LBest	8.3845	4.3765	2.6518	6.5908	4.9043	5.7987	-1.6468	31.0598
18	GBest	8.4812	1.7353	3.0016	6.3923	4.9213	6.3962	4.6721	35.6000
19	DEBest	10.0843	5.0825	6.6141	6.9730	5.8556	6.1323	6.6655	47.4073

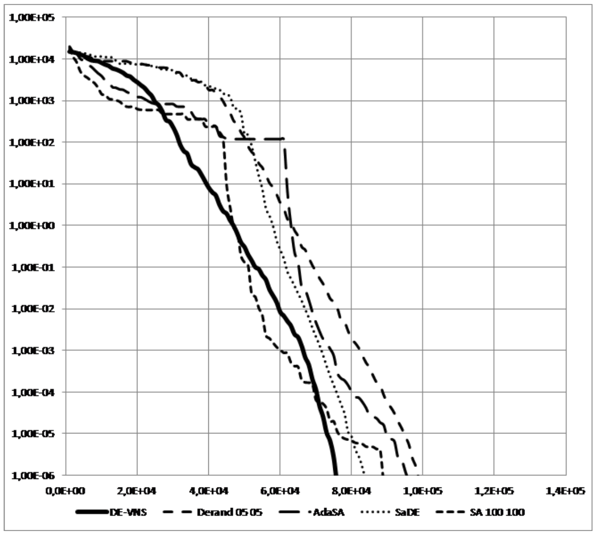
4.2 Statistical tests

The results of the statistical analysis given in this subsection do not contain Glob-VNS and Gauss-VNS heuristics, since the results were not available for all the all test problems. The results with the inclusion of these two heuristics would be biased, so we excluded them from the further analysis. Therefore, we continue comparison of the remaining 17 heuristics. To prove that the rank distributions of heuristics do not have the same first moments, i.e. the same mean, we use the Kruskal-Wallis test [15]. The function compares the medians of the samples, and returns the p value for the null hypothesis that all samples are drawn from the same population (or equivalently, from different populations with the same distribution).

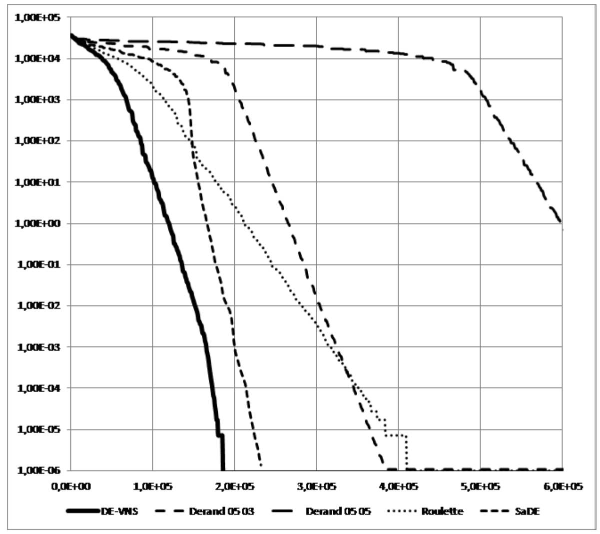
Table 11: Kruskal-Wallis ANOVA test

Source	SS	df	MS	Chi-sq	Prob > Chi-sq
Columns	62066.1	15	3879.13	52.31	9.75462e-6
Error	77936.9	102	764.09		
Total	140003	118			

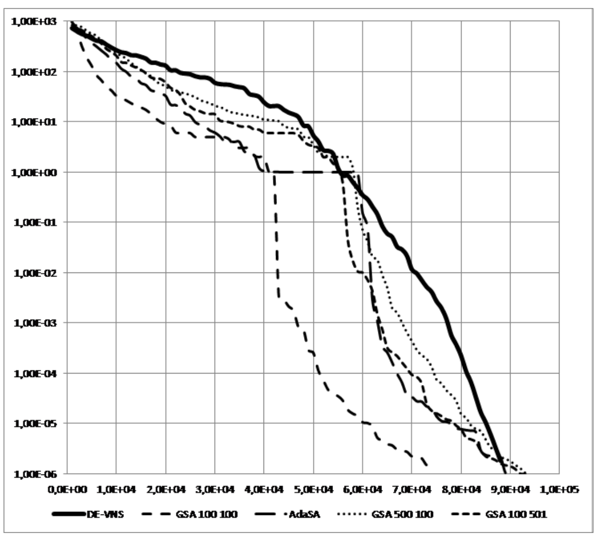
As it can be seen in Table 11, the probability $P = 9.75e-6$ casts doubt on the null hypothesis and suggests that at least one sample median is significantly different from the others. Since this is a generalized conclusion concerning the results of all heuristics, we have no indications which ranks behave statistically different. For this reason we use multiple comparison Wilcoxon rank sum test [6, 10] which performs a two-sided rank sum test of the null hypothesis that the data in the vectors are independent samples from identical continuous distributions with equal medians, against the alternative that they do not have equal medians, for all combinations of the used heuristics. The results can be seen in Table 12. It appears that, the null hypothesis (that the ranks of other heuristics are independent samples from identical continuous distributions with equal medians as DE-VNS) should be rejected (with the confidence level $\alpha = 0.1$). It is notable the natural clustering of algorithms from the same family of heuristics. For example, DE variants (SaDE, CoDE and JADE) with mutual p-value 0.5, 0.63 and 0.73 respectively, from which we cannot reject the null hypothesis of the identical distribution. Similar is happening with variants DERand 0.5 0.3, DERand 0.5 0.5, and DERand rand. The only exception concerns the DE variants – DEBest, that with a significance level $\alpha = 0.1$ does not match with any other heuristics in terms of the first moment distribution of ranks. This is an expected result, since DEBest does not show good results in the case of multi-modal optimization problems and has a problem of premature convergence. Variants of Simulated Annealing (SIMPSA, SA and


1,00E+05
1,00E+04
1,00E+03
1,00E+02
1,00E+01
1,00E+00
1,00E-01
1,00E-02
1,00E-03
1,00E-04
1,00E-05
1,00E-06
0,0E+00 2,0E+04 4,0E+04 6,0E+04 8,0E+04 1,0E+05 1,2E+05 1,4E+05
DE-VNS
Derand 05 05
AdaSA
SaDE
SA 100 100

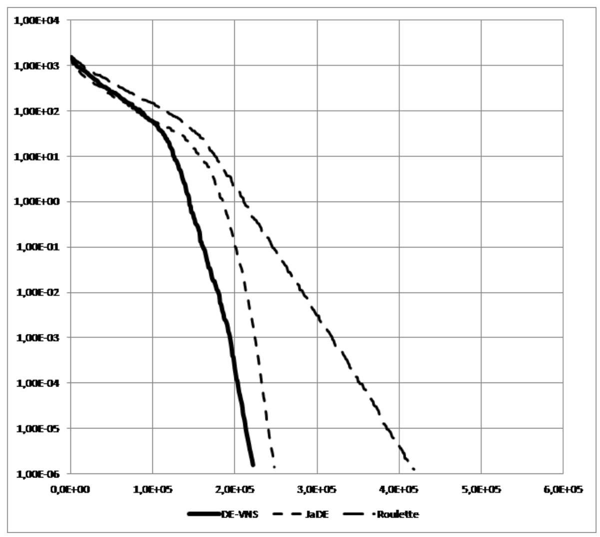
(a)


1,00E+05
1,00E+04
1,00E+03
1,00E+02
1,00E+01
1,00E+00
1,00E-01
1,00E-02
1,00E-03
1,00E-04
1,00E-05
1,00E-06
0,0E+00 1,0E+05 2,0E+05 3,0E+05 4,0E+05 5,0E+05 6,0E+05
DE-VNS
Derand 05 03
Derand 05 05
Roulette
SaDE

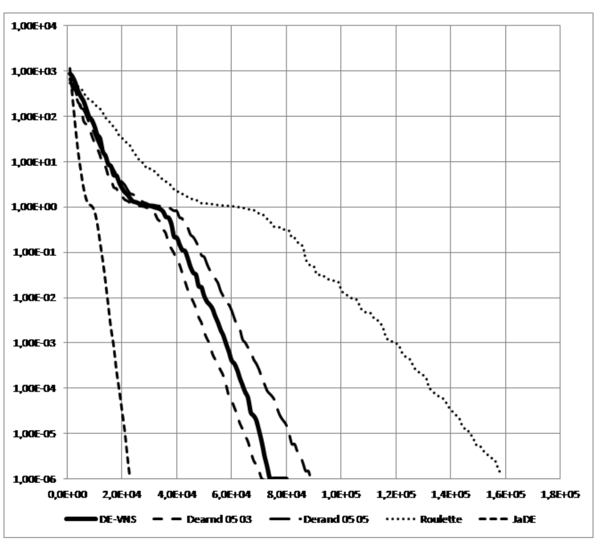
(b)


1,00E+03
1,00E+02
1,00E+01
1,00E+00
1,00E-01
1,00E-02
1,00E-03
1,00E-04
1,00E-05
1,00E-06
0,0E+00 1,0E+04 2,0E+04 3,0E+04 4,0E+04 5,0E+04 6,0E+04 7,0E+04 8,0E+04 9,0E+04 1,0E+05
DE-VNS
GSA 100 100
AdaSA
GSA 500 100
GSA 100 501

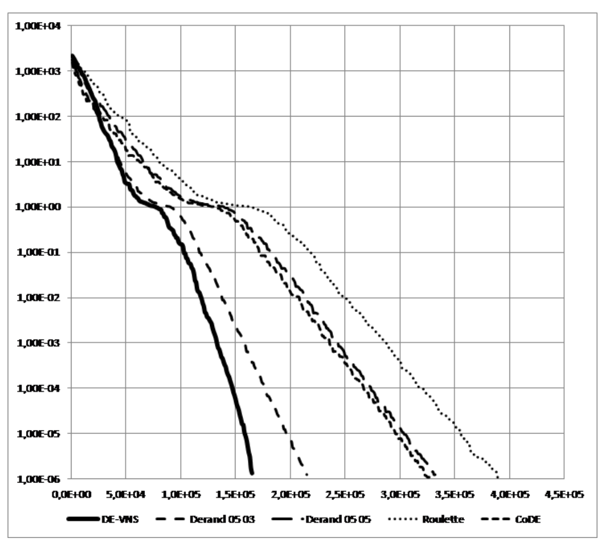
(c)


1,00E+04
1,00E+03
1,00E+02
1,00E+01
1,00E+00
1,00E-01
1,00E-02
1,00E-03
1,00E-04
1,00E-05
1,00E-06
0,0E+00 1,0E+05 2,0E+05 3,0E+05 4,0E+05 5,0E+05 6,0E+05
DE-VNS
JaDE
Roulette

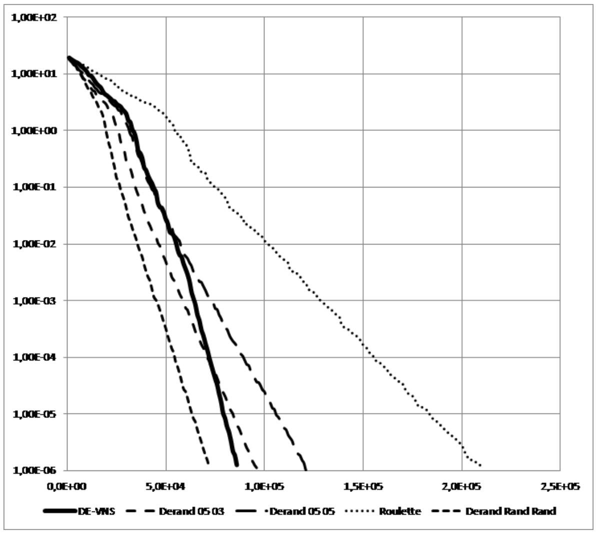
(d)


1,00E+04
1,00E+03
1,00E+02
1,00E+01
1,00E+00
1,00E-01
1,00E-02
1,00E-03
1,00E-04
1,00E-05
1,00E-06
0,0E+00 2,0E+04 4,0E+04 6,0E+04 8,0E+04 1,0E+05 1,2E+05 1,4E+05 1,6E+05 1,8E+05
DE-VNS
Dearnd 05 03
Derand 05 05
Roulette
JaDE

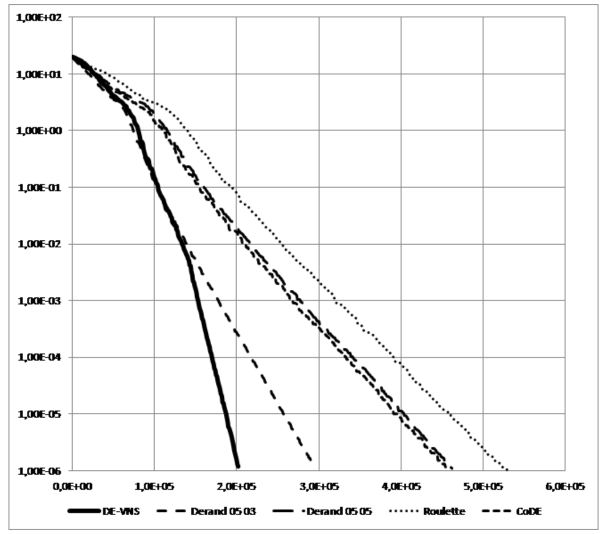
(e)


1,00E+04
1,00E+03
1,00E+02
1,00E+01
1,00E+00
1,00E-01
1,00E-02
1,00E-03
1,00E-04
1,00E-05
1,00E-06
0,0E+00 5,0E+04 1,0E+05 1,5E+05 2,0E+05 2,5E+05 3,0E+05 3,5E+05 4,0E+05 4,5E+05
DE-VNS
Derand 05 03
Derand 05 05
Roulette
CoDE

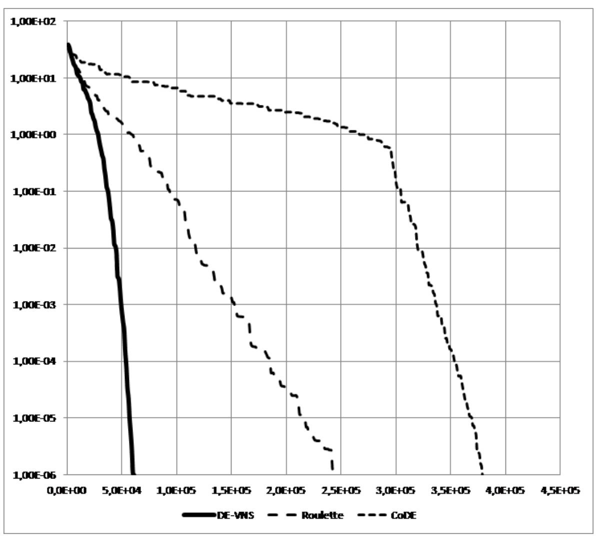
(f)


DE-VNS
Derand 05 03
Derand 05 05
Roulette
Derand Rand Rand

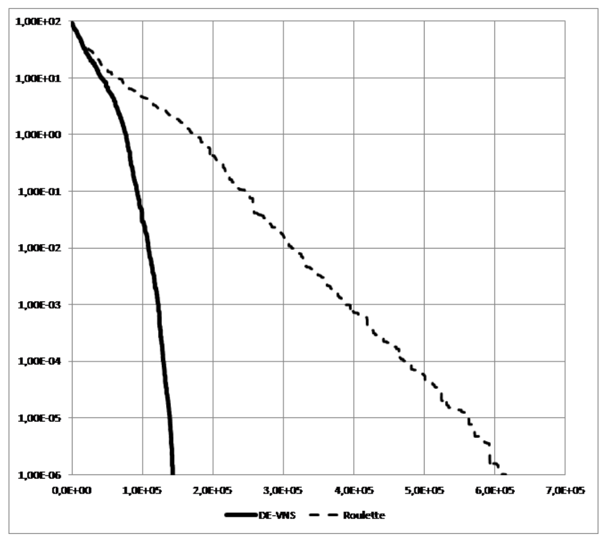
(g)


DE-VNS
Derand 05 03
Derand 05 05
Roulette
CoDE

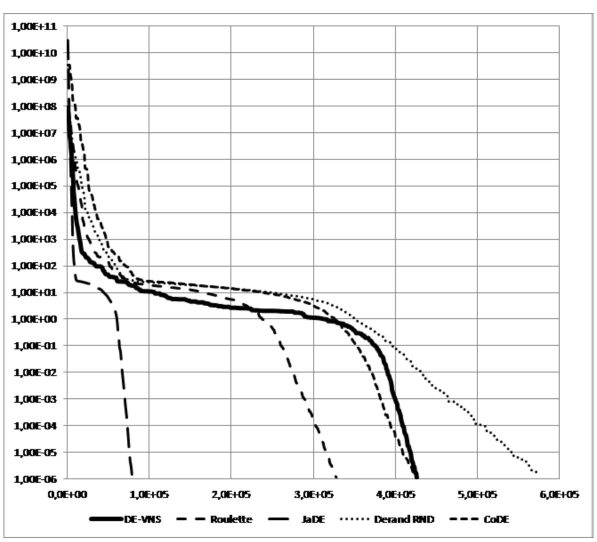
(h)


DE-VNS
Roulette
CoDE

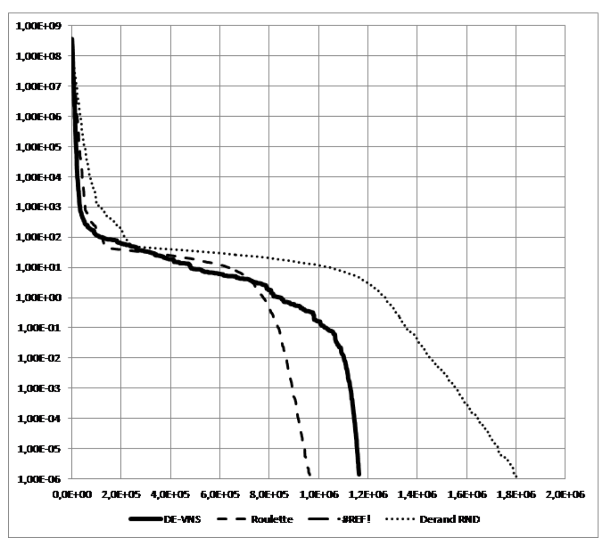
(i)


DE-VNS
Roulette

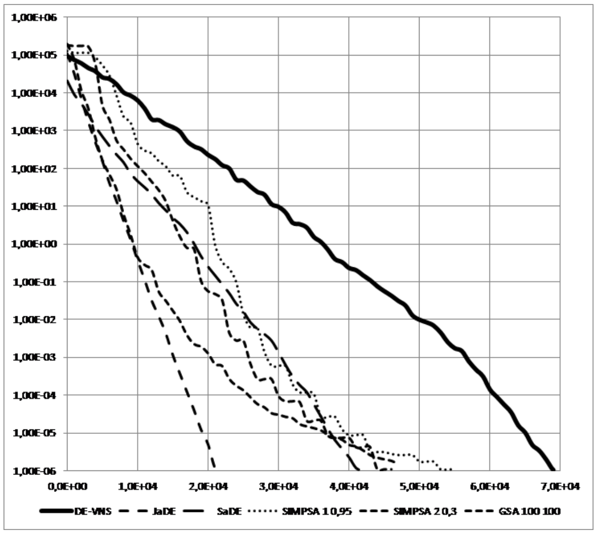
(j)


DE-VNS
Roulette
JaDE
Derand RND
CoDE

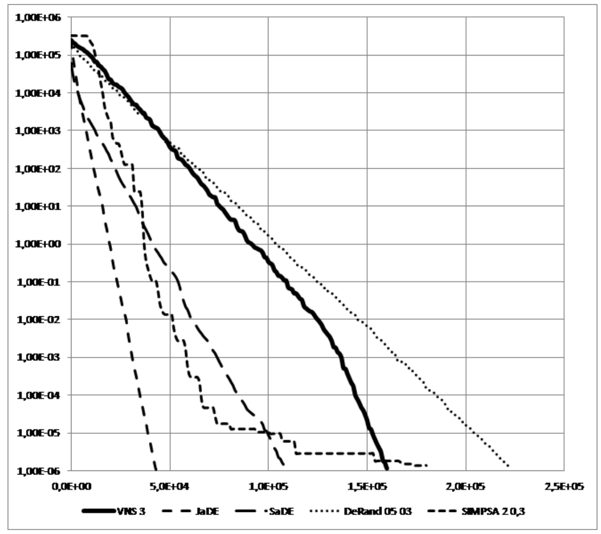
(k)


DE-VNS
Roulette
#REF!
Derand RND

(l)



(m)



(n)

Figure 9: Convergence for (a) Schwefel 50 D; (b) Schwefel 100 D; (c) Rastrigin 50 D; (d) Rastrigin 100 D; (e) Greiwank 50 D; (f) Greiwank 100 D; (g) Ackley 50 D; (h) Ackley 100 D; (i) MPE 50 D; (j) MPE 100 D; (k) Rosenbrock 30 D; (l) Rosenbrock 50 D; (m) Sphere 50 D; (n) Sphere 100 D

Table 12: Multiple comparison Wilcoxon rank sum test

	DE-VNS	DEbr18	JADE	DERand 0.5 0.3	SaDE	DERand 0.5 0.5	SA 100 100	DERand rand	SA 100 500	SIMPSA 2	CoDE	SA 500 100	SIMPSA 1	SAR	LBest	GBest	DEBest
DE-VNS	1.00	0.097	0.023	0.035	0.006	0.003	0.002	0.001	0.001	0.002	0.002	0.001	0.002	0.001	0.001	0.001	0.001
DEbr18		1.000	0.199	0.412	0.101	0.104	0.010	0.017	0.002	0.012	0.004	0.002	0.009	0.002	0.001	0.001	0.001
JADE			1.000	0.881	0.505	0.776	0.734	0.431	0.395	0.274	0.636	0.269	0.242	0.269	0.032	0.010	0.001
DERand 0.5 0.3				1.000	0.830	0.510	0.738	0.647	0.597	0.689	0.649	0.513	0.599	0.513	0.241	0.232	0.009
SaDE					1.000	0.971	0.981	0.645	0.764	0.436	0.742	0.738	0.397	0.506	0.087	0.033	0.007
DERand 0.5 0.5						1.000	0.764	0.731	0.506	0.633	0.687	0.332	0.600	0.475	0.195	0.197	0.077
SA 100 100							1.000	0.923	0.688	0.878	1.000	0.603	0.734	0.431	0.057	0.006	0.002
DERand rand								1.000	0.947	1.000	0.718	0.587	0.642	0.730	0.034	0.004	0.002
SA 100 500									1.000	0.844	1.000	0.569	0.640	0.745	0.029	0.003	0.001
SIMPSA 2										1.000	0.680	0.923	0.691	0.732	0.059	0.004	0.002
CoDE											1.000	0.720	1.000	0.612	0.028	0.002	0.002
SA 500 100												1.000	0.827	0.779	0.030	0.003	0.002
SIMPSA 1													1.000	0.827	0.174	0.030	0.002
SAR														1.000	0.244	0.079	0.005
LBest															1.000	0.340	0.022
GBest																1.000	0.015
DEBest																	1.000

SAR), even with a significance level of $\alpha = 0.5$, accept the hypothesis of same ranks distribution moments, which indicates the similarity of behavior of these variants for different values of the control parameters. It can be also seen that Particle Swarm Optimizer variants (LBest and GBest), belong to the group with the same distributions.

Based on this statistical analysis, we can extract 5 overlapping groups of heuristics, which have a similar distribution of median ranks. The first group contains DE-VNS, suggesting the best results for the problem under consideration. The second group includes representatives of the advanced version of the DE algorithm (DEbr18, JADE, SaDE, DERand 0.5 0.3 and DERand 0.5 0.5). In the third group are mixed heuristics (JADE, DERand 0.5 0.3, DERand0.5 0.5, SaDE, CoDE, SA 100 100, SA 100 500, SA 500 100, DERand rand, SAR and SIMPSA variants).The fourth group consists of PSO variants (GBest and LBest) while in the fifth group is only DEBest.

5 Conclusions

In this paper we perform extensive comparative analysis of 19 algorithms for solving continuous box constrained global optimization problem. All 19 methods are based on some metaheuristic principle, such as Simulated annealing (SA), Differential evolution (DE), Variable neighborhood search (VNS) and Particle swarm optimization (PSO). We believe that we collected currently best methods from the literature, and compared them at the same test instances (small and large), on the same computer, using the same programming language (Matlab) and using the same evaluation parameters. In addition, we performed some statistical tests to evaluate performances of heuristics more rigorously. It appears that all heuristics are naturally divided in 5 groups. The best method, in terms of the number of function evaluation and precision, appeared to be the recent hybrid between DE and VNS (DE-VNS). On the basis of tests conducted, with a confidence of $\alpha = 0.1$, DE-VNS belongs to the separate ranking group. Probably the most interesting our observation is the fact that heuristics that follow the same metaheuristic principles are clustered in the same quality group. For example, group of DE heuristics (DEbr18, JADE and SaDE) is in the second quality group. Their rank statistics are better than ranks of other methods. In addition, it has few parameters to set in some variants and in some have fully adaptable control parameters. VNS variants, on some low dimensional problems, show a remarkable convergence speed and success rate of 100%, but we cannot generalize this statement since the results of these heuristics were available for only some test instances. SA variants behave well, but only on selected convex problems. PSO variants did not show satisfactory results, especially on multimodal problems with higher dimensions. DEBest is by far the last, given that this heuristic does not behave well regarding multiple optima problems and have a problem of premature convergence. Probably a larger set of test instances is required to get more rigorous conclusions of comparative analysis. This task remains for the future work.

References

[1] Anagnostopoulos A., Michel L., Van Hentenryck P., Vergados Y.: A simulated annealing approach to the traveling tournament problem. J. of Sched. 9, 177–193, (2006)

[2] Cardoso, M.E., Salcedo, R.L., de Azevedo, S.E.: The simplex-simulated annealing approach to continuous non-linear optimization., Comput. Chem. Eng. 20(9), 1065–1080 (1996)

[3] Carrizosa, E., Drazic, M., Drazic, Z., Mladenović, N.: Gaussian variable neighborhood search for continuous optimization, Comput. & Oper. Res. 39, 2206–2213 (2012)

[4] Chandrasekarand, K., Ramana, N.V.: Performance Comparison of GA, DE, PSO and SA Approaches in Enhancement of Total Transfer Capability using FACTS Devices, J. of Electr. Eng. & Technol. 7(4), 493–500 (2012)

[5] Eberhart, R.C., Kennedy, J.: A new optimizer using particle swarm theory, In: Proc. of the 6th Int. Symp. on Micro Mach. and Hum. Sci. (1995), doi:10.1109/MHS.1995.494215

[6] Gibbons, J.D.: Nonparametric Statistical Inference. Marcel Dekker, New York (1985)

[7] Glover F.: Tabu Search – Part I, ORSA J. Comut. 1(3), 190–206 (1989)

[8] Glover F.: Tabu Search – Part II, ORSA J. Comut. 2(1), 4–32 (1990)

[9] Goldberg D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning, Addison-Wesley, Reading (1989)

[10] Hollander, M., Wolfe, D.A.: Nonparametric Statistical Methods. Wiley, New York (1999)

[11] Johnson, D.: The triangular distribution as a proxy for the beta distribution in risk analysis, The Statistician 46, 387–398 (1997)

[12] Kennedy, J., Eberhart, R.C.: Particle swarm optimization. In: Proc. of the IEEE Int. Conf. on Neural Netw. (1995) doi: 10.1109/ICNN.1995.488968

[13] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing, Sci. 220, 671–680 (1983)

[14] Kovačević, D., Mladenović, N., Petrović, B., Milošević, P.: DE-VNS: Self adaptive Differential Evolution with Crossover Neighborhood Search for Continious Global Optimization, Comput. & Oper. Res. (submitted)

[15] Kruskal, W.H., Wallis, W.A.: Use of ranks in one-criterion variance analysis, J. of the Am. Stat. Assoc. 47, 583–621 (1952)

[16] Lavor, C., Maculan, N.: A function to test methods applied to global minimization of potential energy of molecules, Num. Alg. 35, 287–300 (2004)

[17] Liu, J., Lampinen, J.: On setting the control parameter of the differential evolution method, In: Proc. 8th Int. Conf. Soft Computing, 11–18 (2002)

[18] Metropolis, N., Rosenbluth, A., Teller, A., Teller, E.: Equation of state calculations by fast computing machines, J. Chem. Phys. 21, 1087–1092 (1953)

[19] Mladenović, N., Hansen, E.: Variable neighborhood search, Comput. & Oper. Res. 24(1), 1097–1100 (1997)

[20] Mladenović, N., Drazic, M., Kovačević-Vujcic, V., Cangalovic, M.: General variable neighborhoods search for the continuous optimization, Eur. J. of Oper. Res. 191(3), 753–770 (2008)

[21] Murtuza Baker, S., Schallau, K., Junker, B.H.: Comparison of different algorithms for simultaneous estimation of multiple parameters in kinetic metabolic models, J. of Integr. Bioinforma. 7, 133 (2010)

[22] Poli, R., Kennedy, J., Blackwell, T.: Particle swarm optimization: an overview, Swarm Intelli. 1(1), 33–57 (2007)

[23] Press W.H., Teukolsky, S.A.: Simulated annealing optimization over continuous spaces, Comput. Phys. 5(4), 426 (1991).

[24] Price, K., Storn, R., Lampinen, J.: Differential Evolution: A Practical Approach to Global Optimization. Springer, Berlin (2005)

[25] Qin A.K., Huang, V.L., Suganthan, P.N.: Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. on Evol. Comput. 13(2), 398–417 (2009)

[26] Salomon, R.: Re-evaluating genetic algorithm performance under coordinate rotation of benchmark functions: a survey of some theoretical and practical aspects of genetic algorithms, Biosyst. 39(3), 263–278 (1996)

[27] Shi, Y., Eberhart, R.C., A modified particle swarm optimizer, In: Proc. of the IEEE Int. Conf. on Evol. Comput. (1998), doi: 10.1109/ICEC.1998.699146

[28] Smith, J.E., Fogarty, T.C.: Operator and parameter adaptation in genetic algorithms, Soft Comput. 1(2), 81–87 (1997)

[29] Storn, R., Price, K.: Differential Evolution: A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces, J. Glob. Optim. 11, 341–359 (1997)

[30] Tvrdik, J.: Differential Evolution with Competitive Setting of its Control Parameters, TASK Quart. 11, 169–179 (2007)

[31] Van Dorp, J., Kotz, S.: A Novel Extension of the Triangular Distribution and its Parameter Estimation, The Statistician 51, 63–79 (2002)

[32] Vesterstroem, J., Thomsen, R.: A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems, In: Proc. IEEE Congr. Evolutionary Computation, (2004) doi: 10.1109/CEC.2004.1331139

[33] Wang Y., Cai, Z., Zhang, Q.: Differential evolution with composite trial vector generation strategies and control parameters, IEEE Trans. on Evol. Comput. 15(1), 55–66 (2011)

[34] Zhang J., Sanderson, A.C.: JADE: Adaptive differential evolution with optional external archive, IEEE Trans. on Evol. Comput. 13(5), 945–958 (2009)