Swarm and Evolutionary Computation 29 (2016) 47–72

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

journal homepage: www.elsevier.com/locate/swevo

Regular Paper

Hybrid self-adaptive cuckoo search for global optimization

Uroš Mlakar, Iztok Fister Jr.*, Iztok Fister

Faculty of Electrical Engineering and Computer Science, University of Maribor, Smetanova 17, 2000 Maribor, Slovenia

ARTICLE INFO

Article history:
Received 11 August 2015
Received in revised form
3 March 2016
Accepted 14 March 2016
Available online 29 March 2016

Keywords:
Cuckoo search
Global optimization
Self-adaptation
Hybridization

ABSTRACT

Adaptation and hybridization typically improve the performances of original algorithm. This paper proposes a novel hybrid self-adaptive cuckoo search algorithm, which extends the original cuckoo search by adding three features, i.e., a balancing of the exploration search strategies within the cuckoo search algorithm, a self-adaptation of cuckoo search control parameters and a linear population reduction. The algorithm was tested on 30 benchmark functions from the CEC-2014 test suite, giving promising results comparable to the algorithms, like the original differential evolution (DE) and original cuckoo search (CS), some powerful variants of modified cuckoo search (i.e., MOCS, CS-VSF) and self-adaptive differential evolution (i.e., jDE, SaDE), while overcoming the results of a winner of the CEC-2014 competition L-Shade remains a great challenge for the future.

1. Introduction

Nowadays, nature-inspired algorithms are applied for solving the hardest optimization problems in theory and practice. The main advantage of these algorithms is searching for solutions using the principle of 'trial-and-error'. Thus, the possible solutions are generated using the various variation operators and their quality estimated by appropriate fitness function after testing. Two inspirations have crucial impact on the development of nature-inspired algorithms, i.e., the Darwinian evolution, and the behavior of social living animals (e.g., flocks of birds and shoals of fish) and insects (e.g., bees and ant colonies). The former inspiration influenced the development of evolutionary algorithms (EAs) [1], while the latter on swarm intelligence (SI) based algorithms [2]. Typically, the nature-inspired algorithms are stochastic and maintain a population of solutions that have improved their qualities over the generations. The new solutions are generated by applying the appropriate variation operators, like crossover, mutation, and move. The first two are employed by EAs, while the latter by SI-based algorithms.

Normally, EAs are appropriate for solving the problems where no user experience and no problem-specific knowledge exist. Therefore, these algorithms have been applied to almost every domain of optimization, simulation and modeling. According to a representation of solutions, EAs are divided into the following types: genetic algorithms (GA) [3], genetic programming (GP) [4], evolution strategies (ES) [5] and differential evolution (DE) [6,7], and evolutionary programming (EP) [8].

The concept of swarm intelligence was used for the first time by Beni and Wang [9] by development of cellular robots. Today, these algorithms are applied in optimization, control of robots, routing in a new generation of mobile networks, and other domains, where robustness and flexibility are demanded. The more frequently used SI-based algorithms are ant colony optimization [10], particle swarm optimization (PSO) [11], artificial bee colony (ABC) [12], artificial immune systems (AIS) [13], firefly algorithm (FA) [14], and symbiotic organisms search (SOS) [15].

Cuckoo search (CS) belongs to a SI-based family of stochastic population-based algorithms. It was developed by Yang and Deb [16] and is inspired by the brood parasitism of some cuckoo species laying their eggs in the nests of other bird species. Each position of the nest in the CS algorithm represents a solution of the problem to be solved. The destiny of a particular nest in each generation can be twofold: a nest with high-quality eggs is preserved for the next generation, while the nest with the lower-quality eggs is abandoned and replaced by a completely new nest.

According to the literature, a lot of cuckoo search variants have been proposed since its introduction in 2009. Shortly after the introduction of basic cuckoo search, Yang and Deb [17] extended the basic CS to a multiobjective CS algorithm intended for solving design optimization problems. A modified cuckoo search was developed by Walton et al., where they added an information exchange between the top eggs. On the other hand, many binary cuckoo search variants have also been proposed [18–20]. Recently, a lot of new CS variants have been proposed by using the hybridization of other algorithms as for example [21–23]. Incorporating adaptation to CS may also become a hot topic for further development. For example, recent adaptative CS variants

* Corresponding author.
E-mail addresses: uros.mlakar@um.si (U. Mlakar), iztok.fister1@um.si (I. Fister Jr.), iztok.fister@um.si (I. Fister).

http://dx.doi.org/10.1016/j.swevo.2016.03.001
2210-6502/

are published in [24–27]. Loosely speaking, cuckoo search is used in myriads of applications in engineering, the real-world, energy management, finance industries too. A design optimization of truss structures was done by Gandomi et al. [28]. Additionally, Fateen and Bonilla-Petriciolet also applied CS algorithm within the chemistry domain [29,30], while Vázquez [31] employed the CS for training the spiking neural models. Gálvez et al. in [32] applied the CS for weighted Bayesian energy functional optimization.

Stochastic population-based algorithms are general enough to be appropriate for solving all classes of problems with which humans are confronted today. Unfortunately, the performances of these algorithms are the same when compared with regard to solving all classes of problems according to the No Free Lunch Theorem (NFL) [33]. However, the performances of general stochastic population-based algorithms can usually be improved in two ways. On the one hand, different algorithm parameter settings have a crucial impact on the performances of these algorithms. That is, the parameters that are valid by starting a search process become inappropriate when the process comes to a matured phase and vice versa. Consequently, the parameters should be changed during the run. In this sense, adaptation and self-adaptation [1] have been proposed within the computational intelligence community. The adaptation means that the parameters are changed according to a feedback from the search process, while the control parameters are saved into a representation of solutions and undergo acting the variation operators by self-adaptation. On the other hand, the stochastic population-based algorithms suffer from a lack of problem-specific knowledge that can generally be conducted by the so-called hybrid algorithms in forms of strategies/features, operators, construction and local search heuristics [34].

This paper proposes the hybrid self-adaptive CS algorithm (HSA-CS) using explicit control of exploration search strategies within the CS algorithm, self-adaptation of the CS control parameters and a population reduction feature. In the proposed HSA-CS, a search space is explored using three exploration search strategies, i.e., random long-distance search strategy, stochastic moderate-distance search strategy and stochastic short-distance search strategy [35]. While the first search strategy is devoted for exploring new solutions, the other two stochastic strategies are dedicated for exploiting the current solutions, where the search process is more directed towards the vicinity of the existing solutions [36]. Thus, the effects of both stochastic strategies are balanced using a special balancing parameter. The exploration/exploitation components of the CS search process are balanced by the control parameter setting. The self-adaptive control of parameter setting was employed by the HSA-CS. The motivation behind the population reduction [37] lies in the fact that the diversity of the population is high at the beginning and therefore more population members are needed. When the search process directs itself towards the more promising regions of the search space, the diversity of population decreases and the search process can reduce the population size accordingly.

The CEC-2014 benchmark function suite customized for global optimization was applied as a test-bed for assessing the quality of the HSA-CS. Recently, few CS algorithms have been developed for solving this problem suite due to the high complexity. As far as we know, Wang et al. [38] have been the only one's who have applied the CS to the similar problem suite. In this study, the results of the HSA-CS were compared with the other more promising variants of CS, like MOCS [39] and CS-VSF [38], the more powerful variants of DE, like jDE [40] and SaDE [41], and two under the first five qualified algorithms on the CEC-2014 competition, i.e., L-Shade [42] and MVMO [43]. Additionally, the original CS [16] and DE [6] have also been included in this comparative study. The study indicated the big potential for solving the global optimization problems of the proposed HSA-CS algorithm in the future.

The structure in the remainder of this paper is as follows. Section 2 describes the original CS algorithm. Section 3 discusses the proposed HSA-CS algorithm. Section 4 illustrates the experiments and results, while the paper is concluded with Section 5, where the performed work is summarized and directions for further development are outlined.

2. Cuckoo search algorithm

Cuckoo search (CS) is a contemporary stochastic nature-inspired population-based algorithm which was developed by Yang and Deb in late 2009 [16]. CS belongs to the SI-based algorithm family [44] that is inspired by the natural behavior of some cuckoo species along with their obligate brood parasitism. About this phenomenon, Payne in his book "The Cuckoos" [45] said:

> "The parasitic breeding behavior of cuckoos has fascinated people for centuries. The brood-parasitic cuckoos lay their eggs in the nests of other kinds of birds, and never rear their own young."

The cuckoo egg hatches earlier than the host's and the cuckoo young's develop faster. Thus, the cuckoo young evict the eggs of the host species. There are several strategies how to lay eggs into a host nest by the female cuckoo. On the one hand, female cuckoos have fast laying behavior. On the other hand, some parasitic cuckoo females are specialized in laying eggs that resemble the eggs of the host species. Consequently, the host birds cannot distinguish the cuckoo eggs from their own and raise these as if they were their own.

Interestingly, not all cuckoo species demonstrate this brood-parasitism. For instance, the anis and the Guire cuckoos are cooperative breeders, where several cuckoo pairs sharing the same nest lay their eggs together and also care for their own young. Some cuckoo species live in solitaire pairs, build own nests and rear their young's.

However, the CS algorithm mimics the brood-parasitism behavior of cuckoo species. To trap the behavior of cuckoos in nature and adapt it to be suitable for using as a computer algorithm, authors have idealized three rules [16]:

- Each cuckoo lays one egg at a time, and dumps it in a randomly chosen nest.
- The best nests with high-quality eggs will be carried over to the next generations.
- The number of available host nests is fixed and any egg laid by a cuckoo may be discovered by the host bird with a probability $p_a \in (0, 1)$. In this case, the host bird can either get rid of the egg, or simply abandon the nest and build a completely new one.

Algorithm 1. Original cuckoo search algorithm.

Input: Population of nests $\mathbf{x}_i = (x_{i,1}, \ldots, x_{i,D})^T$ for $i = 1 \ldots NP$, MAX_FE.
Output: The best solution $\mathbf{x}_{best}$ and its corresponding value $f_{min} = \min(f(\mathbf{x}))$.

```
generate_initial_host_nest_locations;
FE = 0;
while termination_condition_not_meet do
  for i = 1 to NP do
    u_i = generate_new_solution(x_i);
    f_trial = evaluate_the_new_solution(u_i);
    FE = FE + 1;
    j = ⌊rand(0, 1)*NP + 1⌋;
    if f_trial < f_j then
      x_j = u_i; f_j = f_trial ;//replace the j-th random selected solution
    end if
```

```
12:     if rand(0, 1) < p_a then
13:         init_nest(x_worst);
14:     end if
15:     if f_trial < f_min then
16:         x_best = u_i; f_min = f_trial; // replace the global best solution
17:     end if
18:   end for
19: end while
```

The pseudo-code of original cuckoo search algorithm is presented in Algorithm 1 which consists of the following main steps:

- Initialize a population of solutions (function *generate_initial_host_nest_locations* in line 1): In this step, host nests are positioned within the search space randomly.
- Generate a new solution (function *generate_new_solution* in line 5): The algorithm obtains a new position of i-th cuckoo randomly by Lévy flight.
- Evaluation of solution (function *evaluate_the_new_solution* in line 6): Fitness of solution is evaluated.
- Replacement of the randomly selected solution (lines 9–11): Some solution j is randomly selected and replaced with the i-th trial solution if the trial solution is better.
- Reinitialization of the worst solution (lines 12–14): The worst nests can be abandoned and the new ones are built according to probability p_a.
- Save the global best solution (lines 15–17): Maintaining the best solution is performed during this step.

A solution to the original cuckoo search algorithm corresponding to cuckoo nests represents the position of the cuckoo egg within the search space. Mathematically, this position is defined as

$$\mathbf{x}_i^{(t)} = \{x_{i,j}^{(t)}\}, \quad \text{for } i = 1, \ldots, NP \quad \text{and} \quad j = 1, \ldots, D, \tag{1}$$

where NP denotes the number of cuckoo nests in the population, D the dimension of the problem to be solved, and t the generation number.

As can be seen from Algorithm 1, the local random walk (implemented in *generate_new_solution* line 5 in Algorithm 1) intended primarily for exploitation of the current solutions is carried out by using the Lévy flight distribution expressed as

$$\mathbf{x}_i^{(t+1)} = \mathbf{x}_i^{(t)} + \alpha L(s, \lambda), \tag{2}$$

where

$$L(s, \lambda) = \frac{\lambda \Gamma(\lambda) \sin(\pi\lambda/2)}{\pi} \frac{1}{s^{1+\lambda}}, \quad (s \gg s_0 > 0). \tag{3}$$

Term $L(s, \lambda)$ determines the characteristic scale and $\alpha > 0$ denotes a scaling factor of the step size s.

Interestingly, the CS algorithm employs the 'one-to-random' replacement operator that somewhat resembles to the 'one-to-one' replacement operator in differential evolution (DE) [6]. In CS algorithm, the randomly selected candidate solution j competes with the trial solution for a place in the next generation by the former operator, while the trial solution replaces the corresponding candidate solution i in the case of fitness improvement by the latter operator.

The global random walk intended primarily for exploration of the search space is implemented in function *init_nest* (line 13 in Algorithm 1) and is mathematically expressed as

$$x_{i,j}^{(t+1)} = (Ub_j - Lb_j) * U_j(0, 1) + Lb_j, \tag{4}$$

where Lb_j and Ub_j are the lower and upper bounds of the specific variable, respectively, and $U_j(0, 1)$ is a random number drawn from a uniform distribution from interval $[0, 1]$. Let us explain that the global random search is interleaved with the local search according to the probability p_a in the original cuckoo search algorithm.

Yang and Deb stated in [16] that the main advantage which distinguishes CS from the other meta-heuristic algorithms is the fine balance between exploration and exploitation in CS search process and the lesser number of parameters. However, the obtained less than average results, compared to other state-of-the-art meta-heuristics, proved that the CS does not offer a good balance between exploration and exploitation. They also asserted that there are actually two algorithm parameters, i.e., population size NP and probability p_a, and once the former parameter is fixed only the latter parameter remains to be properly set by the user. Unfortunately, very little was said about those parameters controlling the Lévy flight distribution as proposed in Eq. (2) in Yang's and Deb's paper. This distribution is controlled using two additional parameters: scale s and stability factor λ. A range of stable distributions is obtained for $0 < \lambda \leq 2$ that extend from a Cauchy distribution for $\lambda = 1$, a normal distribution for $\lambda = 2$ to Lévy flights when $1 < \lambda < 2$. Furthermore, the random value drawn from Lévy flights distribution is scaled by scaling factor α in the proposed equation. In summary, we have three additional parameters in order to totally control the step size. However, their proper settings depend on the characteristics of the problem to be solved. The default values of these parameters are $\alpha = 1.0$, $s = 1.0$ and $\lambda = 1.5$ as proposed by the authors of the original CS algorithm. Despite its simplicity and generality, however, the original algorithm is hard to apply to complex real-world problems without any adaptation and/or hybridization [34].

3. Hybrid self-adaptive cuckoo search algorithm

This section presents the proposed hybrid self-adaptive cuckoo search (HSA-CS), where the original CS, as described in the previous section, is modified by adding the following mechanisms:

- balancing the exploration strategies,
- self-adaptation of CS control parameters, and
- population reduction feature.

In the remainder of this paper, the proposed self-adaptive and hybrid mechanisms are described thoroughly, accompanied by pseudo-code.

3.1. Balancing the exploration strategies

According to Črepinšek et al. in [36], the exploration/exploitation is achieved by selection, mutation and crossover operators in EAs. A balancing between exploration and exploitation is performed by a control parameter setting that is obviously problem dependent.

The proposed HSA-CS algorithm implements three different strategies for exploring the search space (also exploration strategies). Each of these strategies is controlled by their own control parameters. Furthermore, a launching of the strategies is controlled by specific control parameters. Consequently, both sets of control parameters have a great impact on the exploration/exploitation components of the CS search process. According to the distance to how far the trial solution can be generated from the parent solution, there are three exploration strategies in the HSA-CS [35]:

- the random long-distance exploration,
- the stochastic short-distance exploration and
- the stochastic moderate-distance exploration.

These strategies determine how different from the parent solution

the generated trial solution could be expected after applying the specific exploration strategy. The first exploration strategy presents the global random walk according to Eq. (4). Primarily, this strategy is devoted for exploring new solutions and thus induced a raising of the population diversity. The second strategy improves the current solution using the local random walk (RW) [46] according to Eq. (2) and directs the search process to exploit the neighborhood of the already discovered solution. The third exploration strategy 'rand_to_best/1/bin' is borrowed from the DE algorithm [6] and consists of two terms, i.e., the distance to the current best solution and the distance between two random selected solutions. Intuitively, the first term has more exploitive, while the second more explorative effect on the search process.

Moreover, this strategy also introduces a crossover operation. Typically, the SI-based algorithms employ only a mutation strategy, where all elements of a trial solution are changed after using the operator. In addition to mutation in EAs, a crossover operator is applied to the trial solution in order to preserve some good elements of this solution from being changed. While the mutation is typically an unary operation, the crossover operation demands an interaction between two or more population members and enables a flow of information inside the population. This flow is controlled by the crossover rate (*CR*) that limits the number of changed elements in the new solution.

The proposed DE mutation strategy is expressed as

$$\mathbf{u}_i^{(t)} = \mathbf{x}_i^{(t)} + F_i(\mathbf{x}_{best}^{(t)} - \mathbf{x}_i^{(t)}) + (\mathbf{x}_{r1}^{(t)} - \mathbf{x}_{r2}^{(t)}), \tag{5}$$

where F_i denotes a scaling factor regulating the magnitude of the change, $\mathbf{x}_{best}^{(t)}$ is the current best solution and, $\mathbf{x}_{r1}^{(t)}$ and $\mathbf{x}_{r2}^{(t)}$ denote the randomly selected solutions from a cuckoo population.

Introducing the crossover operator to the proposed DE mutation strategy in Eq. (5) within the CS algorithm has a crucial impact on the performances, as turned out during the experimental work. Mathematically, this crossover can be expressed as follows:

$$w_{i,j}^{(t)} = \begin{cases} u_{i,j}^{(t)} & \text{rand}_j(0,1) \le CR \vee j = j_{rand}, \\ x_{i,j}^{(t)} & \text{otherwise}, \end{cases} \tag{6}$$

where $CR \in [0.0, 1.0]$ controls the fraction of parameters that are copied to the trial solution. The condition $j = j_{rand}$ ensures that the trial vector differs from the original solution $\mathbf{x}_i^{(t)}$ in at least one element. Let us notice that when $CR = 1.0$ the whole mutated vector $\mathbf{u}_i$ is copied to the trial vector $\mathbf{v}_i$. In this case, no crossover takes place.

Finally, replacement of the randomly selected solution is performed that can be mathematically expressed as follows:

$$\mathbf{x}_k^{(t+1)} = \begin{cases} \mathbf{w}_i^{(t)} & \text{if } f(\mathbf{w}_i^{(t)}) \le f(\mathbf{x}_k^{(t)}) \wedge k \ne i, \\ \mathbf{x}_i^{(t)} & \text{otherwise}, \end{cases} \tag{7}$$

where $k = \text{rand}(0, NP)$ is a randomly selected integer number drawn from uniform distribution in interval $[0, NP)$.

The main weakness of the majority of SI-based algorithms is fast convergence toward a local optimum. Therefore, the biggest challenge for developers of these algorithms is how to maintain the diversity of the population over the generations. The following scheme is applied for balancing between two stochastic exploration search strategies that is controlled by a balancing probability p_b in the HSA-CS algorithm

$$\text{if} \begin{cases} U(0,1) \le p_b \Rightarrow \textit{moderate\_distance\_strategy} \\ \text{otherwise} \Rightarrow \textit{short\_distance\_strategy} \end{cases} \tag{8}$$

where $U(0, 1)$ denotes a uniformly generated random value between 0 and 1. The random long-distance exploration strategy is applied similarly as by the original CS algorithm. This means that three exploration strategies take care about the diversity of the population in the proposed HSA-CS.

3.2. Self-adaptation of CS control parameters

The self-adaptation of control parameters is typically used when their proper values are unknown in advance or they need to be changed during the stochastic search process [1]. The following control parameters are self-adapted in the HSA-CS. The short-distance exploration strategy is controlled by two control parameters, i.e., the scaling factor α and the stability factor λ, while the moderate-distance exploration strategy needs the scaling factor F and the crossover rate CR. All these control parameters are embedded within the representation of solutions and undergo acting of the variation operators. Consequently, the solutions in HSA-CS are represented as follows:

$$\mathbf{x}_i^{(t)} = (x_{i,1}^{(t)}, \ldots, x_{i,D}^{(t)}, \alpha_i^{(t)}, \lambda_i^{(t)}, F_i^{(t)}, CR_i^{(t)})^T, \quad \text{for } i = 1, \ldots, NP. \tag{9}$$

The mentioned parameters of the short-distance exploration search strategy are self-adapted according to the following equations:

$$\alpha_i^{(t+1)} = \begin{cases} r_1, & \text{if } r_2 < \tau_0, \\ \alpha_i^{(t)}, & \text{otherwise}, \end{cases} \lambda_i^{(t+1)} = \begin{cases} r_3, & \text{if } r_4 < \tau_1, \\ \lambda_i^{(t)}, & \text{otherwise}, \end{cases} \tag{10}$$

$$F_i^{(t+1)} = F_i^{(t)}, \quad CR_i^{(t+1)} = CR_i^{(t)},$$

while a self-adaptation of the moderate-distance exploration search strategy is expressed as follows:

$$F_i^{(t+1)} = \begin{cases} r_5, & \text{if } r_6 < \tau_2, \\ F_i^{(t)}, & \text{otherwise}, \end{cases}$$

$$CR_i^{(t+1)} = \begin{cases} r_7, & \text{if } r_8 < \tau_3, \\ CR_i^{(t)}, & \text{otherwise}, \end{cases}$$

$$\alpha_i^{(t+1)} = \alpha_i^{(t)}, \quad s_i^{(t+1)} = s_i^{(t)}, \tag{11}$$

where $\tau_0, \ldots, \tau_3$ are the so-called learning rates determining when the corresponding self-adaptive control parameter needs to be changed, and $r_1, \ldots, r_8$ are random numbers drawn from uniform distribution within the interval $[0, 1]$.

3.3. Population reduction feature

The population size parameter *NP* used by the nature-inspired population-based algorithms plays a significant role in controlling the convergence rate. The small-sized populations tend towards faster convergence but also increase the risk of getting trapped in a local optimum. On the other hand, the larger-sized populations converge slower but provide better exploration of the search space. A lot of studies have been made on the impact of varying the *NP* parameter throughout the stochastic search process [37,40] but these methods usually define complex metrics that require additional meta-control parameters, which are also hard for tuning. In this paper, we used the idea as proposed in [42], where a linear reduction of the population was utilized.

The population size is modified by the linear population reduction according to the following equation:

$$NP^{(t+1)} = round\left[\left(\frac{NP_{max} - NP_{min}}{MAX\_FE}\right) * FE + NP_{min}\right], \tag{12}$$

where NP_{max} is the starting population size, NP_{min} is the user specified minimal population size, and *FE* and *MAX_FE* are the current number of function evaluation and maximum number of function evaluations, respectively.

3.4. Pseudo-code of the HSA-CS algorithm

The pseudo-code of the HSA-CS algorithm is presented in Algorithm 2. The main difference between the original CS and the HSA-CS algorithms lies in the representation of solutions because the representation in the modified CS algorithm is adjusted in order to enable self-adapting the control parameters (Eq. (9)). Additionally, balancing the stochastic exploration strategies according to parameter p_b (lines 10–14 in Algorithm 2) is implemented that demands modifying the value of the control parameters according to Eqs. (10) and (11). Finally, the linear population reduction feature is incorporated within the HSA-CS (line 26 in Algorithm 2).

Algorithm 2. Hybrid cuckoo search with population reduction.

1: **procedure** HYBRID SELF-ADAPTIVE CUCKOO SEARCH
2: generate_initial_host_nest_locations;
3: **while** termination_condition_not_set **do**
4: **for** $i = 1$ **to** NP **do**
5: $j = \lfloor \text{rand}(0, 1)_* NP + 1 \rfloor$;
6: **if** $rand(0, 1) \leq p_e$ **then**
7: j = getBestPosition; // the top p_e% of best solutions
8: **end if**
9: **if** $rand(0, 1) \leq p_b$ **then**
10: $\mathbf{w}_j$ = moderate_distance_strategy($\mathbf{x}_j$);
11: **else**
12: $\mathbf{w}_j$ = short_distance_strategy($\mathbf{x}_j$);
13: **end if**
14: f_{trial} = evaluate_the_new_solution($\mathbf{w}_j$);
15: $FE = FE + 1$;
16: **if** $f_{trial} \leq f_j$ **then**
17: $\mathbf{x}_j = \mathbf{w}_j$; $f_j = f_{trial}$;
18: **end if**
19 **if** $rand(0, 1) \leq p_a$ **then**
20: abandon_worst_nest($\mathbf{x}_{worst}$); // random new solution
21: **end if**
22: **if** $f_{trial} < f_{min}$ **then**
23: $\mathbf{x}_{best} = \mathbf{u}_i$; $f_{min} = f_{trial}$; // replace the global best solution
24: **end if**
25: **end for**
26: reduce_population;
27: **end while**
28: **end procedure**

The other additional features of the HSA-CS algorithm are as follows. This algorithm introduces a new elitist parameter p_e that denotes the probability of the random selected solution from a set capturing some percentages of the best solutions in the population. The higher the value of this parameter, the more elitist solutions can participate in this selection process. Thus, it is expected that exploitation in the vicinity of the best solutions is encouraged. A replacement of the local best solutions, the global best solution as well as abandoning the worst solution is similar as by the original CS algorithm. The HSA-CS is finished by a linear population reduction (line 27 in Algorithm 2).

The original CS algorithm uses five parameters when controlling of the Lévy flight distribution is taken into account. Two parameters are necessary for the proper operation of the algorithm, while the other three parameters (i.e., α_i, λ_i, s) control the stochastic moderate-distance exploration strategy. Let us assume that the step size s is fixed. As a result, there are four parameters that are self-adapted and therefore easy to set. However, introducing the DE stochastic moderate-distance strategy demands two control parameters $F_i^{(t)}$ and $CR_i^{(t)}$. The linear population reduction feature operates using two parameters NP_{max} and NP_{min}, where setting of parameter NP_{max} depends on the dimension of the problem, while the parameter NP_{min} is fixed.

In HSA-CS, a new solution is generated either by the stochastic short-distance exploration strategy according to Eq. (3) or by the stochastic moderate-distance exploration strategy according to Eq. (5) based on the value of the balancing probability p_b. Occasionally, the random long-distance exploration strategy is launched according to the probability p_a.

4. Experiments and results

The purpose of our experimental work was twofold. On the one hand, we wanted to show that the HSA-CS significantly outperforms the results of the original CS algorithm, while on the other hand, to demonstrate that these obtained results are comparable to other standard and powerful algorithms, like the original CS [16], DE [6], jDE [47], SaDE [41], L-Shade [42], and MVMO [43]. In line with this, the mentioned algorithms were applied to CEC-2014 benchmark suite.

The experimental work was divided into the following tests by addressing:

- the influence of the population reduction feature,
- the influence of the dimensionality of the problem,
- the influence of the balancing probability p_b,
- the influence of the elitist parameter p_e,
- the influence of the probability p_a,
- the influence of the HSA-CS features,
- the comparative study,
- the HSA-CS time complexity, and
- solving the real-world problems.

The proposed parameter setting for HSA-CS is presented in Table 1 which is divided into four columns: the parameter name, its rational initial value and the interval of the rational values from which the parameter values can be drawn either by developer at the beginning of a run or by the HSA-CS modifying the same parameter self-adaptively during the run. The self-adaptive parameters are indicated in the last column by indicator “Yes”.

While two of the original CS parameters $\alpha_i^{(0)}$ and $\lambda_i^{(0)}$ are self-adaptive, the third parameter (i.e., the scale parameter in Lévy flight distribution) was fixed to $s = 1.0$. The initial values of self-adaptive parameters $F_i^{(0)}$ and $Cr_i^{(0)}$ are set as proposed in [47].

The parameter maximum population size NP_{max} depends on the dimension of the problem. The higher the dimension of the problem, the higher the population size. However, its optimal value needs to be discovered during experiments by the developer. As a starting value of the parameter, we propose $NP_{max} = 200$. This parameter should be then varied up to 1000 in steps of 200 in order to determine the optimal value for the population size. Let us notice that the population size must be set highly enough because of using the linear population reduction. The population reduction feature was borrowed from the L-Shade algorithm, where the parameter minimum population size had been set to the minimum number of vectors needed for the proper execution of the algorithm, i.e., $NP_{min} = 3$. In our opinion, the rational setting of this parameter is located in the interval between 3 and 10. The $NP_{min} = 10$ is proposed for the initial value. The motivation of this setting was to stop the population reduction at this minimum

Table 1
Parameters for the HSA-CS algorithm.

Parameter	Initial values	Proposed interval	Self-adaptation
$\alpha_i^{(0)}$	0.9	[0.0,1.0]	Yes
$\lambda_i^{(0)}$	1.5	[1.2,1.8]	Yes
s	1.0	n/a	No
$F_i^{(0)}$	0.5	[0.1,1.0]	Yes
$Cr_i^{(0)}$	0.9	[0.0,1.0]	Yes
NP_{max}	200	[200,1000]	No
NP_{min}	10	[3, 10]	No
p_a	0.1	[0.05,0.2]	No
p_b	0.8	[0.5,1.0]	No
p_e	0.1	[0.05,0.2]	No

population size, providing sufficient population diversity.

The proper setting of parameters p_a, p_b and p_e strongly depends on the problem to be solved. Therefore, some experiments are needed to find their optimal setting for the specific problem. In Table 1, the rational values for these parameters can be seen. In order to help the potential developers, we also propose the intervals in which the optimal settings of these parameters should be found in our opinion.

For evaluating the effectiveness of the proposed algorithm the following measures were used:

- best solution found over 51 runs,
- worst solution found over 51 runs,
- mean error over 51 runs,
- standard deviation over 51 runs, and
- median of 51 runs.

In the reminder of this paper, the PC configuration, on which experiments were conducted, is presented, a description of the benchmark suite is specified and finally, the results of the mentioned algorithms are illustrated.

4.1. PC configuration

All runs were made on an IBM System x3550 M4 Server, with the following configuration:

- Processor: Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00 GHz.
- RAM: 65 GB.
- Operating system: Linux Ubuntu 14.04.2 LTS.

The majority of the tested algorithms were implemented in the C++ programming language. The code of the MOCS was downloaded from the Internet site [48] in Matlab code, while the code for L-Shade was taken from the CEC-2014 competition webpage.

4.2. Test suite

The CEC 2014 test suite (Table 2) consists of 30 benchmark functions that are divided into four classes:

- unimodal functions (1–3),
- simple multi-modal functions (4–16),
- hybrid functions (17–22), and
- composition functions (23–30).

Unimodal functions have a single global optimum and no local optimums. The unimodal functions in this suite are non-separable and rotated. Multi-modal functions are either separable or non-separable. In addition, they are also rotated or/and shifted. To develop the hybrid functions, the variables are randomly divided into some subcomponents and then different basic functions are used for different subcomponents [49]. Composition functions consist of a sum of two or more basic functions. In this suite, hybrid functions are used as the basic functions to construct composition functions. The characteristics of these hybrid and composition functions depend on the characteristics of the basic functions.

4.3. Influence of the population reduction feature

This experiment was intended to show how the population reduction feature influences the overall performance of the HSA-CS. In line with this, the starting population size was varied in the interval $NP_{max} \in \{100, 200, 400, 600, 800, 1000\}$, while the following dimensions of the problem $D \in \{10, 30, 50\}$ were taken into account. In fact, the purpose of this experiment was to investigate how the starting population size NP_{max} depends on the dimensionality of the problem.

In order for the easier readability of the paper, at most three test problems from each function subgroup from Table 2 are included in Table 3 representing the results of this experiment by optimizing the functions of dimension $D=30$. The rest of the results by optimizing the functions of this dimension as well as the

Table 2
Summary of the CEC'14 test functions.

Subgroup	No.	Functions	$F_i^* = F_i(x_*)$
Unimodal functions	1	Rotated high conditioned elliptic function	100
	2	Rotated bent cigar function	200
	3	Rotated Discus Function	300
Simple multimodal functions	4	Shifted and rotated Rosenbrock's function	400
	5	Shifted and rotated Ackley's function	500
	6	Shifted and rotated Weierstrass function	600
	7	Shifted and rotated Griewank's function	700
	8	Shifted Rastrigin's function	800
	9	Shifted and rotated Rastrigin's function	900
	10	Shifted Schwefel's function	1000
	11	Shifted and rotated Schwefel's function	1100
	12	Shifted and rotated Katsuura function	1200
	13	Shifted and rotated HappyCat function	1300
	14	Shifted and rotated HGBat function	1400
	15	Shifted and rotated expanded Griewank's plus Rosenbrock's function	1500
	16	Shifted and rotated expanded Scaffer's F6 function	1600
Hybrid functions	17	Hybrid function 1 ($N=3$)	1700
	18	Hybrid function 2 ($N=3$)	1800
	19	Hybrid function 3 ($N=4$)	1900
	20	Hybrid function 4 ($N=4$)	2000
	21	Hybrid function 5 ($N=5$)	2100
	22	Hybrid function 6 ($N=5$)	2200
Composition functions	23	Composition function 1 ($N=5$)	2300
	24	Composition function 2 ($N=3$)	2400
	25	Composition function 3 ($N=3$)	2500
	26	Composition function 4 ($N=5$)	2600
	27	Composition function 5 ($N=5$)	2700
	28	Composition function 6 ($N=5$)	2800
	29	Composition function 7 ($N=3$)	2900
	30	Composition function 8 ($N=3$)	3000

Table 3
Influence of the starting population size NP_{max} on the quality of solutions by dimensionality of the problem $D = 30$.

Class	Func.	NP_{max}	Best	Worst	Mean	Std	Median
1	f_1	100	1.37E+03	7.16E+04	2.09E+04	1.75E+04	1.36E+04
		200	6.39E+03	6.92E+04	2.16E+04	1.17E+04	2.00E+04
		400	1.13E+03	6.78E+04	2.14E+04	1.53E+04	1.89E+04
		600	2.38E+03	6.29E+04	**1.66E+04**	1.23E+04	1.43E+04
		800	2.73E+03	5.66E+04	2.12E+04	1.26E+04	1.70E+04
		1000	6.85E+03	7.08E+04	2.33E+04	1.53E+04	1.77E+04
	f_3	100	0.00E+00	9.59E−03	2.28E−03	2.60E−03	8.82E−04
		200	3.56E−07	5.61E−03	5.62E−04	9.53E−04	2.07E−04
		400	3.09E−06	2.69E−03	1.62E−04	3.81E−04	4.85E−05
		600	5.95E−06	3.11E−04	**7.19E−05**	7.21E−05	4.46E−05
		800	4.14E−06	3.59E−04	8.98E−05	8.09E−05	6.17E−05
		1000	7.42E−07	2.96E−03	1.62E−04	4.05E−04	9.32E−05
2	f_4	100	8.84E−06	6.34E+01	**1.34E+00**	8.79E+00	7.54E−03
		200	4.24E−05	6.81E+01	2.76E+00	1.32E+01	3.91E−02
		400	7.87E−03	6.78E+01	3.01E+00	1.07E+01	6.81E−01
		600	2.21E−03	6.91E+01	3.71E+00	1.31E+01	8.97E−01
		800	3.93E−02	7.20E+01	6.68E+00	1.87E+01	1.19E+00
		1000	1.09E−01	7.16E+01	1.31E+01	2.53E+01	1.77E+00
	f_{14}	100	1.36E−01	6.56E−01	2.54E−01	7.26E−02	2.40E−01
		200	1.55E−01	3.08E−01	2.28E−01	3.36E−02	2.26E−01
		400	1.58E−01	2.98E−01	2.30E−01	3.01E−02	2.36E−01
		600	1.60E−01	2.87E−01	**2.27E−01**	2.96E−02	2.25E−01
		800	1.74E−01	2.74E−01	2.33E−01	2.51E−02	2.32E−01
		1000	1.56E−01	2.72E−01	2.23E−01	2.68E−02	2.25E−01
	f_{16}	100	7.95E+00	1.02E+01	**9.28E+00**	4.83E−01	9.29E+00
		200	8.22E+00	1.05E+01	9.50E+00	5.00E−01	9.58E+00
		400	7.87E+00	1.07E+01	9.59E+00	5.81E−01	9.55E+00
		600	6.99E+00	1.08E+01	9.69E+00	6.32E−01	9.84E+00
		800	8.86E+00	1.09E+01	9.90E+00	4.83E−01	9.85E+00
		1000	8.85E+00	1.11E+01	1.01E+01	4.64E−01	1.02E+01
3	f_{19}	100	2.99E+00	9.32E+00	5.81E+00	1.51E+00	5.69E+00
		200	2.53E+00	8.18E+00	4.77E+00	1.04E+00	4.66E+00
		400	3.04E+00	6.78E+00	**4.27E+00**	6.67E−01	4.30E+00
		600	2.91E+00	6.04E+00	4.67E+00	6.40E−01	4.74E+00
		800	3.32E+00	5.78E+00	4.83E+00	5.38E−01	4.84E+00
		1000	3.38E+00	7.43E+00	5.00E+00	6.48E−01	4.96E+00
	f_{20}	100	7.81E+00	1.87E+02	4.35E+01	3.58E+01	2.81E+01
		200	5.60E+00	5.21E+01	1.63E+01	9.32E+00	1.35E+01
		400	4.90E+00	2.07E+01	1.20E+01	4.02E+00	1.18E+01
		600	4.78E+00	1.99E+01	9.97E+00	3.36E+00	9.30E+00
		800	4.71E+00	1.91E+01	9.77E+00	2.74E+00	9.49E+00
		1000	3.36E+00	2.08E+01	**8.78E+00**	3.03E+00	8.12E+00
	f_{22}	100	2.16E+01	2.79E+02	1.33E+02	8.55E+01	1.47E+02
		200	4.03E+00	2.62E+02	7.51E+01	6.36E+01	3.33E+01
		400	2.16E+01	1.55E+02	5.57E+01	4.95E+01	2.71E+01
		600	2.11E+01	1.51E+02	4.73E+01	4.30E+01	2.86E+01
		800	2.14E+01	1.50E+02	**4.20E+01**	3.53E+01	2.87E+01
		1000	2.17E+01	1.58E+02	4.36E+01	3.86E+01	2.82E+01
4	f_{26}	100	1.00E+02	1.00E+02	1.00E+02	4.59E−02	1.00E+02
		200	1.00E+02	1.00E+02	1.00E+02	3.69E−02	1.00E+02
		400	1.00E+02	1.00E+02	1.00E+02	2.96E−02	1.00E+02
		600	1.00E+02	1.00E+02	1.00E+02	3.39E−02	1.00E+02
		800	1.00E+02	1.00E+02	1.00E+02	2.83E−02	1.00E+02
		1000	1.00E+02	1.00E+02	1.00E+02	3.14E−02	1.00E+02
	f_{27}	100	3.01E+02	4.94E+02	3.81E+02	4.15E+01	4.00E+02
		200	3.00E+02	4.02E+02	3.44E+02	4.54E+01	3.16E+02
		400	3.00E+02	4.02E+02	3.22E+02	4.11E+01	3.01E+02
		600	3.00E+02	4.02E+02	**3.13E+02**	3.25E+01	3.01E+02
		800	3.00E+02	4.03E+02	3.20E+02	3.99E+01	3.01E+02
		1000	3.00E+02	4.02E+02	3.15E+02	3.45E+01	3.01E+02
	f_{28}	100	6.65E+02	1.23E+03	8.27E+02	7.98E+01	8.20E+02
		200	6.34E+02	8.75E+02	**7.77E+02**	5.01E+01	7.85E+02
		400	6.88E+02	8.48E+02	7.95E+02	2.70E+01	7.98E+02
		600	6.56E+02	8.47E+02	7.91E+02	2.96E+01	7.93E+02
		800	7.65E+02	8.55E+02	7.98E+02	1.95E+01	7.95E+02
		1000	6.70E+02	8.56E+02	7.99E+02	2.66E+01	7.97E+02

results obtained by optimizing the functions of dimensions $D=10$ and $D=50$ are presented in appendix.

In Table 2, a total of 11 problems are illustrated in order to show how functions of different classes behaved during the optimization. As a result, functions f_1 and f_3 were selected from the unimodal function class, functions f_4, f_{14}, and f_{16} from the simple multimodal class, and functions f_{19}, f_{20}, and f_{22} from the hybrid function class, while functions f_{26}, f_{27}, and f_{28} from the

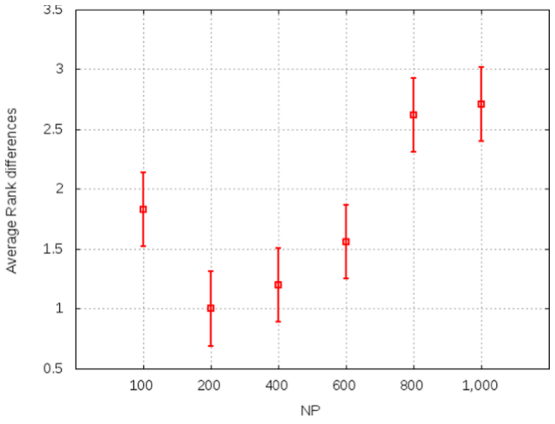
composition function class. Table 3 is divided into rows presenting the function classes, function numbers, and starting population sizes according to their best, worst, mean, standard deviation, and median values. The best values are represented as bold.

In order to estimate the results statistically, Friedman tests [50] were conducted. The Friedman test is a two-way analysis of variances by ranks, where the test statistic is calculated and converted to ranks in the first step, and after that the post-hoc tests are conducted using the calculated ranks in the second step. Here, a low value of rank means a better algorithm [51]. The second step is performed only if a null hypothesis of Friedman test is rejected. The null hypothesis states that medians between the ranks of all algorithms are equal [52].

According to Demšar [53], the Friedman test is as yet a safe and robust non-parametric test for the comparisons of more algorithms over multiple data sets that together with the corresponding Nemenyi post-hoc test enables a neat presentation of statistical results [54]. The main drawback of the Friedman test is that it makes the whole multiple comparisons over data sets and therefore it is unable to establish proper comparisons between some of the algorithms

NP_{max}	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
100	3.44	[3.13,3.75]	†	$\ll 0.05$	†
200	2.61	[2.30,2.92]	‡	∞	‡
400	2.81	[2.50,3.12]		0.0139	†
600	3.17	[2.86,3.48]		0.0100	†
800	4.32	[4.01,4.63]	†	$\ll 0.05$	†
1,000	4.22	[3.91,4.53]	†	0.0001	

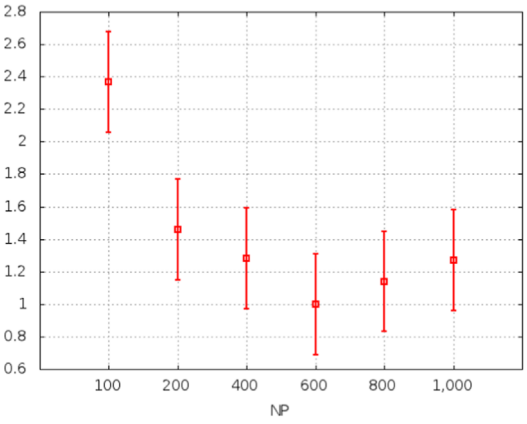
(a) $D = 10$



(b) $D = 10$

NP_{max}	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
100	3.92	[3.61,4.23]	†	$\ll 0.05$	†
200	3.01	[2.70,3.32]		0.0473	†
400	2.83	[2.52,3.14]		0.0221	†
600	2.55	[2.24,2.86]	‡	∞	‡
800	2.69	[2.38,3.00]		0.4670	
1,000	2.82	[2.51,3.13]		0.9512	

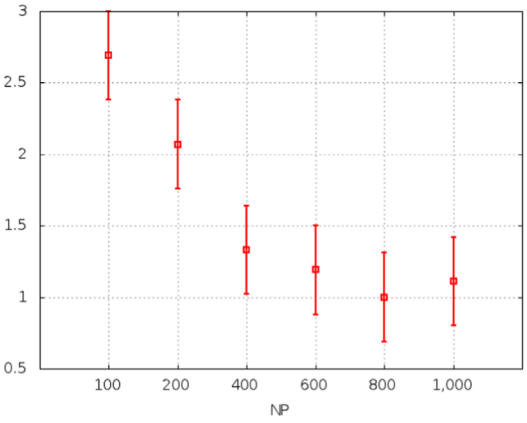
(c) $D = 30$



(d) $D = 30$

NP_{max}	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
100	4.07	[3.76,4.38]	†	$\ll 0.05$	†
200	3.45	[3.14,3.76]	†	$\ll 0.05$	†
400	2.70	[2.39,3.01]		0.0516	
600	2.57	[2.26,2.88]		0.3238	
800	2.37	[2.06,2.68]	‡	∞	‡
1,000	2.49	[2.18,2.80]		0.6958	

(e) $D = 50$



(f) $D = 50$

Fig. 1. The results of Friedman non-parametric tests including post-hoc tests by population reduction analysis.

Table 4
Influence of dimensionality of the problem—Part 1/2.

Func.	Dim.	Best	Worst	Mean	Std	Median
1	10	9.21E−08	4.42E−05	4.96E−06	8.55E−06	1.62E−06
	30	6.85E+03	7.08E+04	2.33E+04	1.53E+04	1.77E+04
	50	1.75E+05	6.33E+05	3.44E+05	1.09E+05	3.19E+05
2	10	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	30	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	50	0.00E+00	7.15E−06	3.37E−07	1.21E−06	2.40E−08
3	10	7.67E−08	2.00E−03	2.06E−04	3.29E−04	1.08E−04
	30	7.42E−07	2.96E−03	1.62E−04	4.05E−04	9.32E−05
	50	7.52E−03	8.34E−02	3.47E−02	1.56E−02	3.36E−02
4	10	8.72E−05	3.48E+01	1.57E+01	1.67E+01	4.34E+00
	30	1.09E−01	7.16E+01	1.31E+01	2.53E+01	1.77E+00
	50	8.70E−02	1.53E+02	6.56E+01	4.01E+01	8.79E+01
5	10	3.54E−02	2.00E+01	1.63E+01	7.02E+00	2.00E+01
	30	2.00E+01	2.03E+01	2.00E+01	5.11E−02	2.00E+01
	50	2.00E+01	2.04E+01	2.00E+01	7.66E−02	2.00E+01
6	10	3.05E−02	1.68E−01	8.13E−02	3.47E−02	7.52E−02
	30	3.16E−01	8.32E+00	1.73E+00	1.79E+00	9.89E−01
	50	2.38E−01	3.58E+00	1.07E+00	8.48E−01	7.10E−01
7	10	6.58E−06	1.71E−02	2.19E−03	3.71E−03	4.81E−04
	30	0.00E+00	1.95E−07	1.05E−08	3.50E−08	0.00E+00
	50	0.00E+00	1.38E−05	1.30E−06	2.52E−06	4.13E−07
8	10	0.00E+00	1.05E−06	1.42E−07	2.24E−07	5.17E−08
	30	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	50	0.00E+00	9.95E−01	1.95E−02	1.38E−01	0.00E+00
9	10	1.00E+00	5.00E+00	2.46E+00	1.00E+00	2.08E+00
	30	9.95E+00	3.49E+01	2.12E+01	4.98E+00	2.00E+01
	50	3.18E+01	8.86E+01	4.89E+01	1.10E+01	4.81E+01
10	10	2.16E−08	1.87E−01	4.90E−02	5.57E−02	6.25E−02
	30	6.25E−02	1.35E+00	3.83E−01	4.30E−01	1.87E−01
	50	1.87E−01	4.59E+00	1.61E+00	8.50E−01	1.57E+00
11	10	7.36E+00	2.34E+02	3.51E+01	4.14E+01	2.34E+01
	30	1.09E+03	3.06E+03	2.06E+03	3.83E+02	2.09E+03
	50	4.06E+03	6.31E+03	5.11E+03	6.31E+02	5.11E+03
12	10	2.19E−02	3.27E−01	1.60E−01	7.87E−02	1.67E−01
	30	7.85E−02	6.67E−01	2.77E−01	1.16E−01	2.51E−01
	50	1.11E−01	6.59E−01	2.58E−01	1.07E−01	2.42E−01
13	10	4.53E−02	1.36E−01	9.93E−02	2.02E−02	1.01E−01
	30	1.49E−01	3.06E−01	2.19E−01	3.72E−02	2.15E−01
	50	2.41E−01	4.13E−01	3.21E−01	3.89E−02	3.22E−01
14	10	3.88E−02	1.74E−01	1.11E−01	2.57E−02	1.11E−01
	30	1.56E−01	2.72E−01	2.23E−01	2.68E−02	2.25E−01
	50	2.12E−01	3.30E−01	2.78E−01	2.38E−02	2.77E−01
15	10	3.05E−01	1.02E+00	5.73E−01	1.57E−01	5.80E−01
	30	1.58E+00	3.93E+00	2.79E+00	4.23E−01	2.85E+00
	50	4.12E+00	7.87E+00	6.13E+00	1.01E+00	6.17E+00

considered [51]. Therefore, a Wilcoxon two paired non-parametric test is applied as a post-hoc test after the control method (i.e., the algorithm with the lowest rank) is determined using the Friedman test. On the one hand, the Nemenyi test is very conservative and it may not find any difference in most of the experimentations [55]. On the other hand, the Wilcoxon test was also preferred by Benavoli et al. in [56] against the post-hoc tests based on mean-ranks. As a result, the Nemenyi test is used for graphical presentation of the results, while the Wilcoxon test is more powerful. Both tests were conducted using a significance level 0.05 in this study.

These tests captured the results of optimizing all the functions in the test suite according to all observed dimensions. In summary, six classifiers (i.e., the results according to different population sizes) were compared according to $30_*5 = 150$ variables, where the first number in the expression denotes the number of functions and the second the number of measures taken into consideration. Three Friedman tests were performed regarding the different dimensions.

The results of the Friedman non-parametric tests are presented in Fig. 1 which is divided into three parts consisting of one table and a diagram dedicated to each considered dimension. Each table contains the results of Friedman tests together with corresponding Nemenyi and Wilcoxon post-hoc tests. The results of the Nemenyi post-hoc test are presented as intervals of critical differences. The best algorithm found by the Friedman test is used as a control method with which all the other algorithms are compared using the Wilcoxon two paired non-parametric test. The results of the Wilcoxon test are illustrated by corresponding p-values, where a significant difference between two algorithms appears if $p < 0.05$. The best algorithm in the Nemenyi post-hoc test as well as the control method in Wilcoxon test is denoted with ‡ symbol in the table, while the significant difference between the control method and corresponding algorithm is depicted by † symbol.

The results of the Nemenyi post-hoc test are graphically illustrated in the corresponding diagrams. In each diagram, the points show the average ranks and the lines indicate confidence intervals (critical differences) for the algorithms under consideration. The lower the rank value, the better the algorithm. On the other hand, two algorithms are significantly different when their intervals do not overlap. Obviously, the diagrams are organized regarding the dimension of the functions.

As can be seen from Fig. 1, the best results determined with minimum ranks for $D=10$ were obtained by starting population size $NP_{max} = 200$ (diagram (c)), while for $D=30$ by $NP_{max} = 600$ (diagram (d)). Interestingly, setting the starting population size to $NP_{max} = 800$, the best results were gained for $D=50$ (diagram (f)). In summary, the starting population size depends on the dimension of the problem. As a matter of fact, the higher the dimension of the problem, the larger the starting population size.

This experiment was designed to investigate the impact of the dimensionality of the problem on the quality of the obtained results. In line with this, three dimensions of the benchmark functions, i.e., $D = \{10, 30, 50\}$, were considered in this test. The HSA-CS used the same parameters as proposed in Table 1. The results of the experiment are assembled in Tables 4 and 5, where each part presents a half of the results obtained by optimizing the observed functions. For each function, the results of three dimensions are aggregated according to the best, worst, mean, standard deviation and median values, respectively.

4.4. Influence of problem dimensionality

The goal of this experiment was to discover how the HSA-CS algorithm behaves when confronted with harder problems (functions with higher dimensions). Obviously, it was expected that this algorithm would be able to produce better results by solving the problems of lower dimensions. In line with this, mean values in the tables needed to be increased when the dimension of the problem becomes higher. Although this fact held for the majority of test functions, the HSA-CS negated this assumption for some functions. The functions at hand were f_4, f_5, f_7, and f_{23}. For instance, function f_4 reported the best mean value by dimension $D=30$ and not by $D=10$ as it is expected.

4.5. Influence of balancing probability

This experiment was intended to show the behavior of stochastic exploration strategies in the search process controlled by the parameter balancing probability p_b. Thus, the elitist parameter was fixed at $p_e=0.1$, the probability $p_a=0.1$, while the balancing probability was varied in the interval $p_b \in [0.0, 1.0]$ in steps of 0.1. As a result, the 11 instances of HSA-CS algorithms were obtained. The instance $p_b=0.0$ indicates a situation, when the HSA-CS works with a stochastic short-distance strategy. In contrast, when the

Table 5
Influence of dimensionality of the problem—Part 2/2.

Func.	Dim.	Best	Worst	Mean	Std	Median
16	10	1.08E+00	2.15E+00	1.67E+00	2.61E−01	1.70E+00
	30	8.85E+00	1.11E+01	1.01E+01	4.64E−01	1.02E+01
	50	1.70E+01	1.99E+01	1.88E+01	6.84E−01	1.88E+01
17	10	2.33E+00	6.56E+01	1.56E+01	1.02E+01	1.56E+01
	30	2.23E+02	1.73E+03	1.21E+03	3.59E+02	1.25E+03
	50	1.82E+03	6.90E+03	3.09E+03	9.04E+02	2.88E+03
18	10	4.60E−02	1.14E+00	3.89E−01	2.51E−01	3.39E−01
	30	2.90E+01	1.13E+02	5.92E+01	1.53E+01	5.70E+01
	50	6.68E+01	1.80E+02	1.13E+02	2.10E+01	1.13E+02
19	10	1.01E−01	1.12E+00	3.73E−01	1.92E−01	3.44E−01
	30	3.38E+00	7.43E+00	5.00E+00	6.48E−01	4.96E+00
	50	1.10E+01	3.64E+01	1.44E+01	3.46E+00	1.37E+01
20	10	2.08E−01	8.42E−01	4.64E−01	1.57E−01	4.47E−01
	30	3.36E+00	2.08E+01	8.78E+00	3.03E+00	8.12E+00
	50	4.43E+01	1.55E+02	1.12E+02	2.67E+01	1.10E+02
21	10	1.83E−01	1.22E+00	7.69E−01	2.64E−01	8.02E−01
	30	1.35E+02	5.93E+02	2.88E+02	8.70E+01	2.72E+02
	50	9.13E+02	2.45E+03	1.58E+03	3.49E+02	1.56E+03
22	10	2.13E−01	2.76E+00	1.06E+00	5.70E−01	8.92E−01
	30	2.17E+01	1.58E+02	4.36E+01	3.86E+01	2.82E+01
	50	4.21E+01	4.02E+02	2.55E+02	8.35E+01	2.71E+02
23	10	3.29E+02	3.29E+02	3.29E+02	1.08E−05	3.29E+02
	30	3.15E+02	3.15E+02	3.15E+02	3.81E−06	3.15E+02
	50	3.44E+02	3.44E+02	3.44E+02	0.00E+00	3.44E+02
24	10	1.00E+02	1.11E+02	1.08E+02	1.99E+00	1.08E+02
	30	2.23E+02	2.26E+02	2.24E+02	4.13E−01	2.24E+02
	50	2.55E+02	2.72E+02	2.64E+02	5.64E+00	2.66E+02
25	10	1.09E+02	1.96E+02	1.19E+02	1.17E+01	1.17E+02
	30	2.03E+02	2.04E+02	2.03E+02	3.03E−01	2.03E+02
	50	2.00E+02	2.22E+02	2.06E+02	8.40E+00	2.00E+02
26	10	1.00E+02	1.00E+02	1.00E+02	2.25E−02	1.00E+02
	30	1.00E+02	1.00E+02	1.00E+02	3.14E−02	1.00E+02
	50	1.00E+02	1.00E+02	1.00E+02	4.11E−02	1.00E+02
27	10	1.10E+00	2.68E+00	1.97E+00	3.34E−01	2.00E+00
	30	3.00E+02	4.02E+02	3.15E+02	3.45E+01	3.01E+02
	50	3.02E+02	4.47E+02	3.55E+02	4.47E+01	3.62E+02
28	10	3.57E+02	4.64E+02	3.66E+02	2.71E+01	3.57E+02
	30	6.70E+02	8.56E+02	7.99E+02	2.66E+01	7.97E+02
	50	1.01E+03	1.24E+03	1.13E+03	4.45E+01	1.13E+03
29	10	2.22E+02	2.24E+02	2.22E+02	5.07E−01	2.22E+02
	30	7.21E+02	1.06E+03	8.18E+02	7.43E+01	8.13E+02
	50	9.04E+02	1.50E+03	1.09E+03	1.31E+02	1.08E+03
30	10	4.63E+02	4.87E+02	4.66E+02	5.18E+00	4.64E+02
	30	7.93E+02	2.97E+03	1.58E+03	4.84E+02	1.45E+03
	50	7.97E+03	1.14E+04	9.25E+03	6.33E+02	9.17E+03

balancing probability is set to $p_b=1.0$, the algorithm employs the stochastic moderate-distance strategy only. Indeed, it is expected that the optimal value of the balancing property parameter p_b would be discovered, where these two strategies are properly balanced to direct the search process towards the best solutions. On the other hand, this value indicates which of the components of the search process is more important during the optimization.

The results of these experiments are depicted in Fig. 2 that is organized similarly as in Section 4.3 according to the observed dimensions $D = \{10, 30, 50\}$ into three parts consisting of numerical results obtained by the statistical tests (i.e., Friedman, Nemenyi and Wilcoxon) and corresponding graphical presentation of Nemenyi post-hoc statistical test.

In summary, the experiments showed that although the original CS algorithm operates with two exploration search strategies (i.e., local search and reinitialization of the worst solution), it suffers from a fine grained moderate-distance exploration strategy able to discover more promising solutions within the search space. Particularly important in this is the application of a crossover operator that does not corrupt the already good elements in the trial solution.

4.6. Influence of the elitist parameter

The purpose of this experiment was to show the influence of the elitist parameter p_e on the obtained results. This parameter determines a percentage of the best solutions that can enter into the one-to-random selection process. Thus, it is expected that the multiple offspring of the so-called elitist parent solution would replace the randomly selected solution and indeed the exploitative power of the search process is increased. However, too higher values for this parameter could harmfully influence the results of the optimization.

In this experiment, the balancing probability was fixed at $p_b=0.9$, the probability $p_a=0.1$, while the values of the elitist parameter were varied within the interval $p_e \in [0.0, 0.25]$ in steps of 0.05. As a result, six instances were obtained for each of the observed dimensions $D = \{10, 30, 50\}$. The results of optimizing the CEC-2014 benchmark suite by the HSA-CS algorithm are represented in Fig. 3.

Fig. 3 accumulates the results of the HSA-CS obtained by optimizing the six instances of 30 benchmark functions for each dimensions considered. As a matter of fact, the $3 \times 6 \times 30 \times 51 = 27,540$ runs of the algorithm were needed in order to accomplish the figure. Obviously, the results in the table are organized as described in Section 4.3, i.e., the numerical results of three statistical tests are presented in the table together with the graphical results of the Nemenyi test for each of three dimensions.

In summary, it can be seen that the impact of the elitist parameter p_e was not significant by optimizing the instances of lower dimensions, i.e., $D=10$ and $D=30$. However, the difference between instances became significant by optimization of benchmark functions of dimension $D=50$. Here, the best results were obtained by $p_e=0.20$. This means the elitism is on average considered by each fifth population member. The results decreased when the value of this parameter decreased/increased from this optimal setting. The worst results were gained when no elitism was considered by the HSA-CS, i.e., by $p_e=0.0$.

4.7. Influence of the probability p_a

The probability p_a controls the launching of a random long-distance exploration strategy. The higher the value of this parameter, the closer the search process is to the random search. Yang and Deb in [16] proposed $p_a=0.25$ for the optimal setting of this parameter. The aim of this experiment was to find the promising settings of this parameter and either confirm or reject this proposed parameter setting.

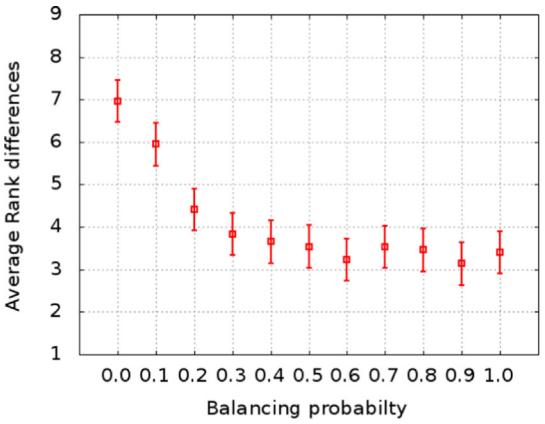
In line with this, this probability was varied within the interval $p_a \in [0.0, 0.25]$ in steps of 0.05. As a result, the six instances for optimizing the CEC-2014 benchmark functions needed to be solved by the HSA-CS. Totally, the $3 \times 6 \times 30 \times 51 = 27,540$ independent runs must be performed for completion of this task. Thus, the balanced probability was fixed at $p_b=0.9$ and the elitist parameter to $p_e = 0.2$ during the experiments.

The results of the experiments are gathered in Fig. 4 that is organized according to the description in Section 4.3. Regarding the dimensions of the observed problems, the table is divided into three parts. For each part, the numerical results of the statistical tests are accumulated within the table, while the graphical results of the Nemenyi tests are presented in the corresponding diagrams.

From Fig. 4, it can be seen that the results obtained by setting the probability parameter $p_a=0.1$ outperformed the results obtained by other settings when optimizing the benchmark functions of dimension $D=10$. Interestingly, the setting of the same

p_b	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
0.0	6.96	[6.46,7.46]	†	$\ll 0.05$	†
0.1	5.95	[5.45,6.45]	†	0.0031	†
0.2	4.41	[3.91,4.91]	†	0.4764	
0.3	3.83	[3.33,4.33]		0.4691	
0.4	3.65	[3.15,4.15]		0.6261	
0.5	3.54	[3.04,4.04]		0.7083	
0.6	3.23	[2.73,3.73]		0.6528	
0.7	3.53	[3.03,4.03]		0.6932	
0.8	3.46	[2.96,3.96]		0.3306	
0.9	3.14	[2.64,3.64]	‡	∞	‡
1.0	3.40	[2.90,3.90]		0.1993	

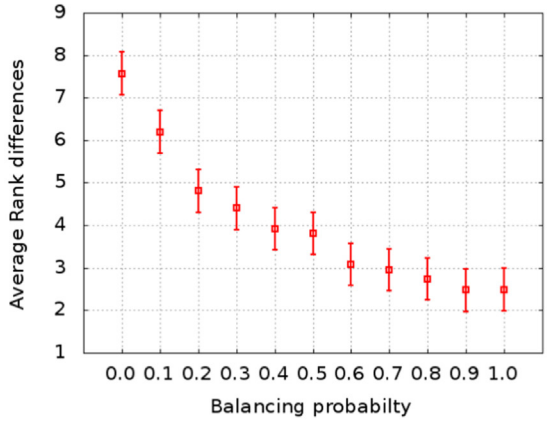
(a) $D = 10$



(b) $D = 10$

p_b	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
0.0	7.57	[7.07,8.07]	†	$\ll 0.05$	†
0.1	6.20	[5.70,6.70]	†	$\ll 0.05$	†
0.2	4.81	[4.31,5.31]	†	$\ll 0.05$	†
0.3	4.40	[3.90,4.90]	†	0.0208	†
0.4	3.92	[3.42,4.42]	†	0.0140	†
0.5	3.81	[3.31,4.31]	†	0.0311	†
0.6	3.08	[2.58,3.58]		0.0026	†
0.7	2.95	[2.45,3.45]		0.0266	†
0.8	2.74	[2.24,3.24]		0.0570	
0.9	2.47	[1.97,2.97]	‡	∞	‡
1.0	2.49	[1.99,2.99]		0.2667	

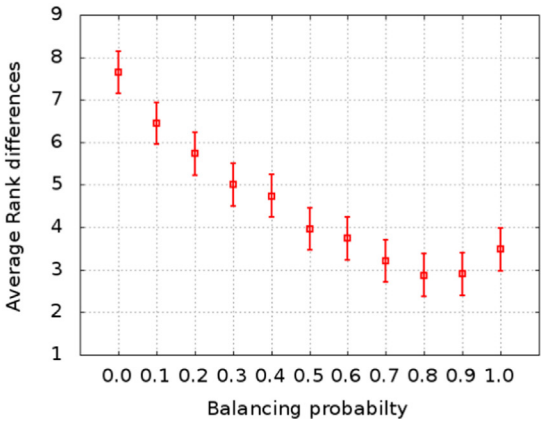
(c) $D = 30$



(d) $D = 30$

p_b	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
0.0	7.65	[7.15,8.15]	†	$\ll 0.05$	†
0.1	6.45	[5.95,6.95]	†	$\ll 0.05$	†
0.2	5.73	[5.23,6.23]	†	$\ll 0.05$	†
0.3	5.00	[4.50,5.50]	†	0.0016	†
0.4	4.74	[4.24,5.24]	†	0.0179	†
0.5	3.96	[3.46,4.46]	†	0.0234	†
0.6	3.74	[3.24,4.24]		0.0341	†
0.7	3.21	[2.71,3.71]		0.4464	
0.8	2.87	[2.37,3.37]	‡	∞	‡
0.9	2.90	[2.40,3.40]		0.2742	
1.0	3.48	[2.98,3.98]		0.1866	

(e) $D = 50$



(f) $D = 50$

Fig. 2. An influence of the balancing probability p_b ($p_e = 0.1$, $p_a = 0.1$).

parameter $p_a = 0.05$ exposed the better results by optimizing the benchmark functions of higher dimensions, i.e., D=30 and D=50. In summary, it can be concluded that the random long-distance exploration search has a favorable impact on the results of the HSA-CS. Using this feature improves the results of the HSA-CS significantly regarding the Wilcoxon 2-paired non-parametric tests when optimizing the benchmark functions of lower dimensions, i.e., D=10 and D=30, and substantially when the benchmark functions of higher dimension (i.e., D=50) was taken into consideration. Indeed, two facts can be highlighted: (1) a setting of this parameter decreases when the dimensionality of the problem increases, and (2) a setting of this parameter to p_a=0.25 is not optimal when the global optimization problems are addressed.

4.8. Influence of the HSA-CS features

The proposed HSA-CS is composed of three features, namely self-adapting the control parameters of the algorithm (SA), balancing the stochastic exploration strategies (BA), and population reduction (PR). The aim of this experiment was to investigate how

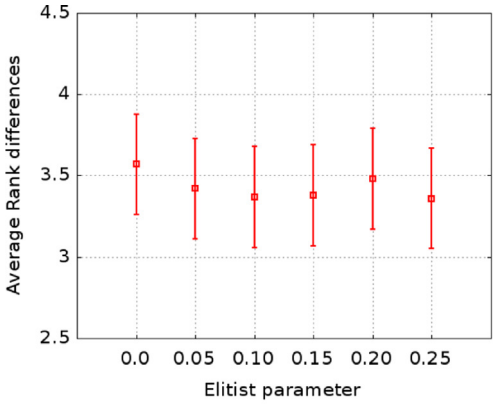
each strategy or combination of strategies affects the results of the original CS by sequential activation of these features.

The experiment was divided into three steps. In the first step, all three features were applied sequentially to the original CS and thus an improvement was calculated. The improvement was calculated using the Friedman non-parametric test, where an increasing of the rank of the CS algorithm modified by the specific features determines the improvement factor (IF). Then, the improvement factor of the best modified CS algorithm was identified that most improved the results of the original CS algorithm. In the second step, the remaining two features were applied to the best modified algorithm determined in the first step sequentially and the improvement factors were calculated. Naturally, the better among the two CS modified by two features was preserved for the third step. In the last step, the results of the proposed HSA-CS using all three features were compared with the results of the original CS and the corresponding improvement factor was calculated.

The results are presented in Table 6, where the improvement factors are presented by applying the specific features to the original CS algorithm. The modified CS algorithms are denoted as CS underlined by the added specific features. For instance, CS_{+SA} denotes the original CS algorithm modified by the self-adapting

p_e	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
0.00	3.57	[3.26,3.88]		0.3252	
0.05	3.42	[3.11,3.73]		0.3789	
0.10	3.37	[3.06,3.68]		0.5574	
0.15	3.38	[3.07,3.69]		0.1142	
0.20	3.48	[3.17,3.79]		0.8839	
0.25	3.36	[3.05,3.67]	‡	∞	‡

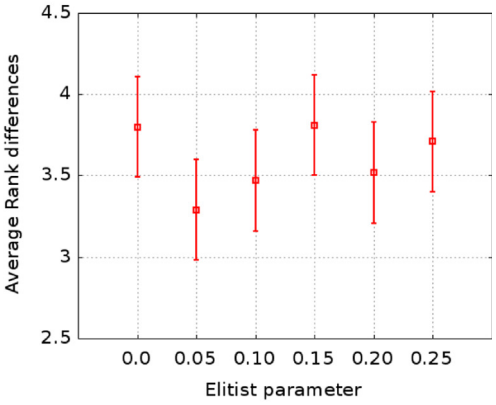
(a) $D = 10$



(b) $D = 10$

p_e	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
0.00	3.80	[3.49,4.11]		0.1947	
0.05	3.29	[2.98,3.60]	‡	∞	‡
0.10	3.47	[3.16,3.78]		0.9305	
0.15	3.81	[3.50,4.12]		0.7163	
0.20	3.52	[3.21,3.83]		0.7955	
0.25	3.71	[3.40,4.02]		0.3536	†

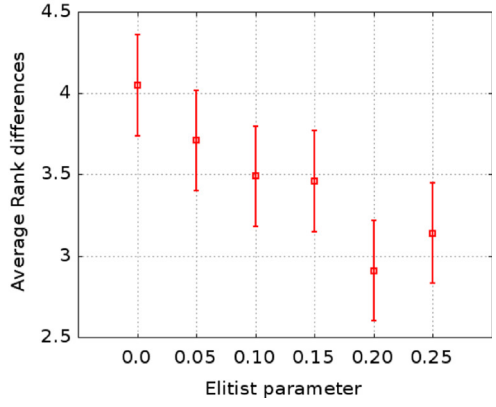
(c) $D = 30$



(d) $D = 30$

p_e	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
0.00	4.05	[3.74,4.36]	†	$\ll 0.05$	†
0.05	3.71	[3.40,4.02]	†	0.004321	†
0.10	3.49	[3.18,3.80]		0.03012	†
0.15	3.46	[3.15,3.77]		0.4785	
0.20	2.91	[2.60,3.22]	‡	∞	‡
0.25	3.14	[2.83,3.45]		0.5523	

(e) $D = 50$



(f) $D = 50$

Fig. 3. An influence of the elitist parameter ($p_b = 0.9$, $p_a = 0.1$).

feature. Let us notice that the results are ordered according to the dimensions of the observed benchmark functions.

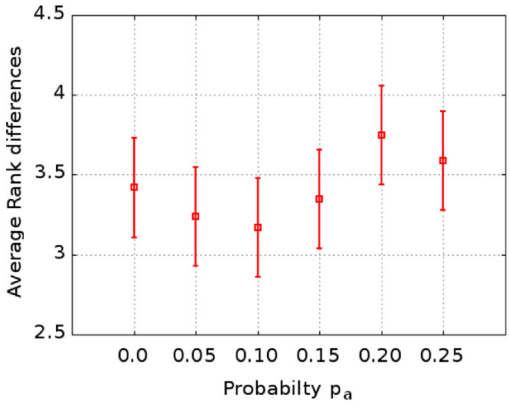
From Table 6, it can be shown that the self-adapting feature has the biggest impact on the results of the HSA-CS by all considered dimensions of the benchmark functions. For instance, using this feature, the CS_{+SA} improves the results of the original CS algorithm for 50% by optimizing the benchmark functions of dimension $D=10$ according to improvement factor. The next important feature is the population reduction as determined by the tests, where, e.g., the modified CS_{+SA+PB} improves the results of the original CS algorithm by 114% by optimizing the benchmark functions of dimension $D=30$ according to improvement factor. Finally, the proposed HSA-CS outperformed the results of the original CS algorithm by 207% by optimizing the benchmark functions of dimension $D=50$ according to improvement factor.

4.9. Comparative study

In order to test the efficiency of the HSA-CS algorithm, it was compared with the other more promising variants of the CS algorithm as well as the more powerful variants of the DE. Usually, the DE variants have achieved the prestigious results in solving the

p_a	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
0.00	3.42	[3.11,3.73]		0.0217	†
0.05	3.24	[2.93,3.55]		0.2765	
0.10	3.17	[2.86,3.48]	‡	∞	‡
0.15	3.35	[3.04,3.66]		0.5481	
0.20	3.75	[3.44,4.06]		0.0131	†
0.25	3.59	[3.28,3.90]		0.6455	

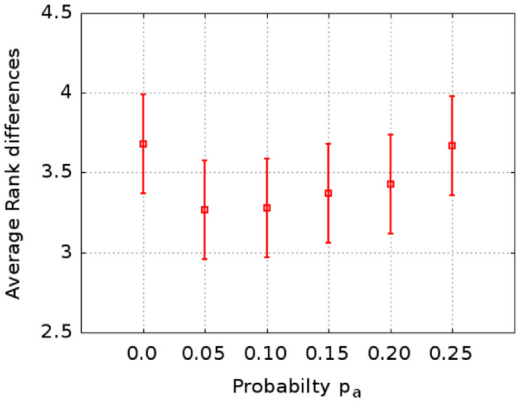
(a) $D = 10$



(b) $D = 10$

p_a	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
0.00	3.68	[3.37,3.99]		0.0019	†
0.05	3.27	[2.96,3.58]	‡	∞	‡
0.10	3.28	[2.97,3.59]		0.2727	
0.15	3.37	[3.06,3.68]		0.6285	
0.20	3.43	[3.12,3.74]		0.4892	
0.25	3.67	[3.36,3.98]		0.1896	

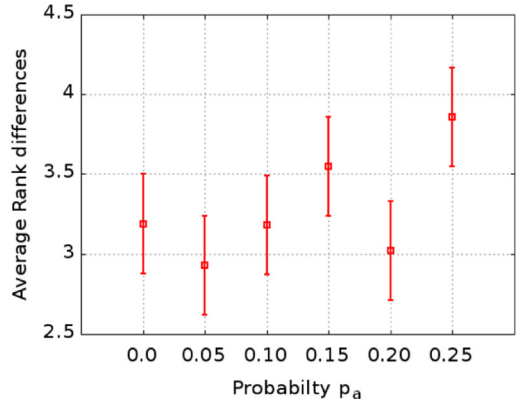
(c) $D = 30$



(d) $D = 30$

p_a	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
0.00	3.19	[2.88,3.50]		0.3867	
0.05	2.93	[2.62,3.24]	‡	∞	‡
0.10	3.18	[2.87,3.49]		0.1139	
0.15	3.55	[3.24,3.86]	†	$\ll 0.05$	†
0.20	3.02	[2.71,3.33]		0.7508	
0.25	3.86	[3.55,4.17]	†	$\ll 0.05$	†

(e) $D = 50$



(f) $D = 50$

Fig. 4. An influence of the probability p_a ($p_b = 0.9$, $p_e = 0.2$).

Table 6
Influence of features on the HSA-CS results.

Dimension	CS	CS_{+SA}		CS_{+SA+PB}		HSA-CS	
		Rank	IF (%)	Rank	IF (%)	Rank	IF (%)
10	6.77	4.46	52	3.14	112	2.16	208
30	7.08	5.11	39	3.30	114	2.32	205
50	6.7	4.79	40	3.08	118	2.18	207

CEC-2014 benchmark function suites over recent years. Let us notice that two variants of the modified CS algorithm were taken into account, i.e., the modified cuckoo search (MOCS) by Walton et al. [39] and the cuckoo search with a varied scaling factor (CS-VSF) by Wang et al. [38]. The former is considered today as one of the better modification of the original CS algorithm, while the latter was applied to the CEC-2013 benchmark function suite and is therefore suitable for solving the global optimization problems. Two powerful and well-known self-adapting DE variants were used in this study, i.e., jDE proposed by Brest et al. [47] and the SaDE by Qin et al. in [41].

The last two algorithms in the study were MVMO by Erlich et al. [43], and L-Shade by Tanabe et al. [42] which were two under the best five qualified algorithms at the Special section and competition on single objective real-parameter numerical optimization at Congress of Computation Intelligence 2014 (CEC-2014). However, the original CS and DE algorithms were included in the study in order to show how the various modifications, adaptations and hybridization affected the results of their originals.

The purpose of this experiment was twofold, i.e., to significantly improve the results of the original CS algorithm by solving the CEC-2014 benchmark function suite, on the one hand and to show that the results of the HSA-CS algorithm are comparable with the other observed algorithms, on the other hand. The following parameter settings for the original CS were used as proposed by Yang and Deb [16]: p_a=0.25, $NP = 100$, $\alpha = 0.9$, $\lambda = 1.5$ during testing. The MOCS operated with the population size $NP = 100$, while the probability p_a was set to p_a=0.25 for all tests. The same value of the second parameter was also employed for the CS-VSF that for different dimensions used the different population sizes, i.e., $NP = 30$, $NP = 50$ and $NP = 50$ for dimensions D=10, D=30 and D=50, respectively.

Additionally, the scaling factor F=0.5 and the crossover rate $CR = 0.9$ were applied for the DE algorithm during the tests. The same initial values were used for jDE. The SaDE is capable of self-adapting the mutation strategies as well as corresponding algorithm parameters. In our SaDE implementation, the following mutation strategies were applied: 'rand/1/bin', 'rand/2/bin', 'rand-to-best/2/bin' and 'current-to-rand/1'. The learning period was set as $LP = 20$, while the scale parameters F_i were changed using the normal distribution $N(0.5, 0.3)$ with mean value 0.5 and standard deviation 0.3, and crossover rates CR_i were randomly drawn from normal distribution $N(CRm_k, 0.1)$ with mean value CRm_k and standard deviation 0.1. For L-Shade the parameters were set according to the accompanying article [42] (i.e., $NP_{max} = 18_{*}D$, $mem_{size} = 6$, $arc_{rate} = 2.6$, $best_{rate} = 0.11$). Considering the MVMO algorithm, the results were taken from the Special Session webpage. The maximum number of fitness function evaluation $MAX\_FE = 10,000 \cdot D$ was used as a termination condition for all algorithms during the tests. All DE algorithms worked on a 100 numbered population (i.e., $NP = 100$). In summary, $3 \times 8 \times 30 \times 51 = 36,720$ independent runs were needed for completing the whole test.

The detailed numerical results for optimizing the functions by the mentioned algorithms are depicted in Table 7, where the best results are marked in bold face. The table accumulates the results as obtained by optimization of the benchmark functions of dimension D=30, while the tables illustrating the results obtained by optimizing the functions of dimensions D=10 and D=50 can be seen in appendix.

The results in Table 7 show that the L-Shade outperformed the results obtained by the other algorithms by 21 times. The HSA-CS was the winner 5 times, while the DE, SaDE, and MVMO reached the better result 5 times, MOCS and CS-VSF one time. The algorithm CS did not perform better on any function in the test suite. The results of Friedman's non-parametric statistical tests are presented in Fig. 5.

Fig. 5 is also organized as described in Section 4.3, where the tables represent the results of Friedman non-parametric tests, and two post-hoc tests, i.e., Nemenyi and Wilcoxon. The best algorithms in the Nemenyi tests are denoted using the character '‡', while the significant differences are marked by † sign. In the Wilcoxon tests, HSA-CS is used as a control algorithm. Here, the '+' sign denotes that the HSA-CS significantly outperformed the results of the specific algorithm, while the '−' sign means that the HSA-CS is significantly worse than the observed algorithm.

In summary, the statistical analysis of the results obtained by the algorithms in the comparative study showed that the L-Shade is really the best, because it outperformed the results of all the other significantly by solving the CEC-2014 benchmark functions of all observed dimensions. On the other hand, the same is also valid by the HSA-CS and MVMO that were only outperformed by the L-Shade. Interestingly, this difference is not significant according to the Nemenyi post-hoc test for D=10 and D=50. However, the three mentioned algorithms are in a class above the rest. The remaining algorithms can be grouped in two groups: To the first quality group belong algorithms like DE, SaDE and jDE, while in the second group algorithms like CS, MOCS and CS-VSF.

As a result, the two hypotheses set at the beginning of the section can be accepted partially, i.e.,

- the HSA-CS significantly outperformed the results of the original CS,
- the results of the HSA-CS were significantly better than the results of the other algorithms in tests, except L-Shade, and substantially better than the results of the MVMO.

Although the HSA-CS was significantly worse compared to L-Shade according to the Wilcoxon test, this difference was not too convincing as evidenced by the Friedman non-parametric test, especially, for dimensions D=10 and D=50. Thus, two facts can be highlighted:

- the HSA-CS outperformed the results of the other DE algorithms significantly,
- the HSA-CS was significantly better than the other contemporary variants of the modified CS algorithm.

The present study demonstrates that the HSA-CS could be successfully applied to the hardest global optimization problems in the future.

4.10. HSA-CS time complexity

This subsection deals with an analysis of the HSA-CS time complexity as defined in [49]. The running time, obtained by evaluating the benchmark function f_{18}, is compared with a running time of the test program presented in Algorithm 3.

Algorithm 3. Test program for computing the algorithm complexity.

Table 7
Comparison of mean and std values for algorithms used in the study on 30D.

F	Meas.	CS	MOCS	CS-VSF	DE	SaDE	jDE	L-Shade	MVMO	HCS-SA
f_1	Mean	3.50E+07	2.93E+06	9.29E+07	6.74E+04	3.73E+03	6.12E+04	**0.00E+00**	1.07E−03	1.16E+04
	Std	2.49E+07	1.51E+06	2.36E+07	5.29E+04	3.26E+03	7.64E+04	**0.00E+00**	1.09E−03	8.79E+03
f_2	Mean	1.95E+07	8.18E+03	6.72E+08	**0.00E+00**	**0.00E+00**	2.27E−15	**0.00E+00**	2.38E−05	**0.00E+00**
	Std	5.49E+07	4.77E+03	1.41E+08	**0.00E+00**	**0.00E+00**	7.87E−15	**0.00E+00**	1.19E−05	**0.00E+00**
f_3	Mean	3.10E+04	5.61E+02	3.62E+04	**0.00E+00**	**0.00E+00**	4.09E−14	**0.00E+00**	1.11E−03	**0.00E+00**
	Std	1.36E+04	5.86E+02	8.47E+03	**0.00E+00**	**0.00E+00**	2.60E−14	**0.00E+00**	1.03E−03	**0.00E+00**
f_4	Mean	2.03E+02	9.41E+01	2.88E+02	1.36E+00	**0.00E+00**	8.53E+00	**0.00E+00**	**0.00E+00**	5.08E−02
	Std	6.69E+01	4.33E+01	3.58E+01	8.77E+00	**0.00E+00**	2.16E+01	**0.00E+00**	**0.00E+00**	1.33E−01
f_5	Mean	2.00E+01	2.06E+01	2.08E+01	2.09E+01	2.03E+01	2.03E+01	2.01E+01	**2.00E+01**	2.00E+01
	Std	2.28E−03	1.45E+00	6.69E−02	4.47E−02	4.03E−02	3.26E−02	3.65E−02	**2.00E+01**	5.53E−03
f_6	Mean	3.23E+01	2.16E+01	2.90E+01	2.11E−02	1.49E+01	5.31E+00	**1.38E−07**	3.62E+00	1.91E+00
	Std	3.27E+00	8.88E+00	1.63E+00	1.49E−01	9.42E−01	4.04E+00	**9.79E−07**	3.04E+00	2.40E+00
f_7	Mean	1.79E+00	3.91E+00	7.01E+00	**0.00E+00**	**0.00E+00**	2.96E−04	**0.00E+00**	2.99E−03	**0.00E+00**
	Std	2.19E+00	9.89E+00	1.39E+00	**0.00E+00**	**0.00E+00**	1.48E−03	**0.00E+00**	0.00E+00	**0.00E+00**
f_8	Mean	1.71E+02	3.80E+01	1.68E+02	1.23E+02	**0.00E+00**	1.19E−01	**0.00E+00**	8.58E−01	**0.00E+00**
	Std	3.46E+01	2.58E+01	1.24E+01	2.54E+01	**0.00E+00**	3.30E−01	**0.00E+00**	9.95E−01	**0.00E+00**
f_9	Mean	2.80E+02	1.39E+02	2.09E+02	1.78E+02	3.58E+01	3.81E+01	**6.78E+00**	2.51E+01	1.84E+01
	Std	5.16E+01	1.09E+02	8.46E+00	9.09E+00	7.01E+00	5.71E+00	**1.47E+00**	2.39E+01	3.35E+00
f_{10}	Mean	2.66E+03	7.98E+02	4.61E+03	5.23E+03	1.11E+00	3.17E+00	**1.63E−02**	1.79E+01	6.35E−01
	Std	5.34E+02	1.11E+03	2.99E+02	6.27E+02	2.02E+00	3.18E+00	**1.56E−02**	9.76E+00	6.84E−01
f_{11}	Mean	4.13E+03	2.77E+03	6.28E+03	6.68E+03	2.28E+03	2.71E+03	**1.23E+03**	1.54E+03	1.86E+03
	Std	5.35E+02	1.28E+03	2.48E+02	2.27E+02	3.45E+02	2.75E+02	**1.81E+02**	1.59E+03	3.46E+02
f_{12}	Mean	5.11E−01	6.15E−01	3.58E−01	2.41E+00	4.59E−01	4.77E−01	1.61E−01	**7.21E−02**	1.48E−01
	Std	2.56E−01	2.91E−01	5.56E−01	2.46E−01	5.23E−02	5.41E−02	2.27E−02	**6.24E−02**	4.33E−02
f_{13}	Mean	4.81E−01	3.24E−01	4.74E−01	3.37E−01	3.02E−01	2.84E−01	**1.24E−01**	1.57E−01	1.48E−01
	Std	1.17E−01	7.27E−02	6.32E−02	4.38E−02	3.69E−02	3.55E−02	**1.73E−02**	1.62E−01	2.43E−02
f_{14}	Mean	3.08E−01	6.31E+00	3.66E−01	2.71E−01	2.68E−01	3.02E−01	2.42E−01	**1.99E−01**	2.05E−01
	Std	5.64E−02	1.58E+01	7.72E−02	2.97E−02	1.40E−01	4.15E−02	2.95E−02	**1.99E−01**	3.24E−02
f_{15}	Mean	9.80E+01	4.11E+01	5.11E+01	1.53E+01	4.86E+00	5.36E+00	**2.15E+00**	2.86E+00	2.66E+00
	Std	3.02E+01	1.60E+01	1.18E+01	1.15E+00	4.17E−01	7.43E−01	**2.49E−01**	2.69E+00	5.08E−01
f_{16}	Mean	1.27E+01	3.97E+04	1.25E+01	1.23E+01	1.03E+01	1.03E+01	**8.50E+00**	1.02E+01	9.62E+00
	Std	5.01E−01	1.14E+05	1.88E−01	2.35E−01	3.42E−01	3.23E−01	**4.54E−01**	9.84E+00	5.44E−01
f_{17}	Mean	1.48E+06	1.74E+05	1.72E+06	1.29E+03	8.55E+02	1.62E+03	**1.88E+02**	9.01E+02	1.22E+03
	Std	1.21E+06	1.34E+05	6.49E+05	1.96E+02	2.80E+02	1.49E+03	**7.42E+01**	1.03E+03	4.00E+02
f_{18}	Mean	7.67E+03	6.94E+02	2.47E+03	5.18E+01	4.92E+01	1.86E+01	**5.91E+00**	2.89E+01	6.11E+01
	Std	6.70E+03	9.51E+02	2.66E+03	6.09E+00	2.57E+01	1.04E+01	**2.86E+00**	2.08E+01	1.99E+01
f_{19}	Mean	5.33E+01	5.88E+02	1.39E+01	4.39E+00	5.26E+00	4.97E+00	3.68E+00	**3.08E+00**	4.86E+00
	Std	3.63E+01	1.55E+03	2.06E+00	2.22E−01	1.15E+00	9.61E−01	6.74E−01	**3.02E+00**	7.68E−01
f_{20}	Mean	3.93E+04	2.31E+04	7.43E+03	3.29E+01	1.85E+01	1.36E+01	**3.08E+00**	1.09E+02	1.17E+01
	Std	2.20E+04	5.55E+04	3.31E+03	7.82E+00	4.14E+00	6.64E+00	**1.45E+00**	5.69E+01	4.09E+00
f_{21}	Mean	3.54E+05	7.58E+04	2.43E+05	6.03E+02	4.31E+02	2.98E+02	**8.68E+01**	4.67E+02	3.31E+02
	Std	3.48E+05	6.28E+04	1.08E+05	2.23E+02	1.32E+02	2.25E+02	**8.90E+01**	4.89E+02	1.26E+02
f_{22}	Mean	9.47E+02	8.08E+02	3.03E+02	3.87E+01	1.65E+02	1.38E+02	**2.76E+01**	1.45E+02	4.83E+01
	Std	3.31E+02	3.05E+02	1.06E+02	3.33E+01	7.11E+01	5.38E+01	**1.77E+01**	1.46E+02	4.52E+01
f_{23}	Mean	3.29E+02	**3.03E+02**	3.21E+02	3.15E+02	3.15E+02	3.15E+02	3.15E+02	3.15E+02	3.15E+02
	Std	7.51E+00	**3.04E+01**	9.75E−01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	3.15E+02	0.00E+00
f_{24}	Mean	2.78E+02	2.25E+02	2.52E+02	**2.17E+02**	2.25E+02	2.26E+02	2.24E+02	2.25E+02	2.24E+02
	Std	3.11E+01	5.90E+00	2.91E+00	**9.11E+00**	4.31E+00	3.34E+00	1.05E+00	2.25E+02	7.20E−01
f_{25}	Mean	2.23E+02	2.08E+02	2.21E+02	2.03E+02	2.03E+02	2.04E+02	**2.03E+02**	2.03E+02	2.03E+02
	Std	9.39E+00	1.60E+01	2.19E+00	6.27E−02	5.52E−01	8.81E−01	**4.91E−02**	2.03E+02	4.55E−01
f_{26}	Mean	1.00E+02	2.43E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	**1.00E+02**	1.00E+02	1.00E+02
	Std	1.63E−01	1.66E+02	6.64E−02	4.69E−02	3.55E−02	5.05E−02	**1.54E−02**	1.00E+02	2.35E−02
f_{27}	Mean	4.27E+02	1.32E+03	5.18E+02	3.41E+02	5.46E+02	4.01E+02	**3.00E+02**	4.01E+02	3.49E+02
	Std	1.96E+01	1.07E+03	4.00E+01	4.84E+01	1.11E+02	5.44E+01	**5.39E−06**	4.01E+02	4.97E+01
f_{28}	Mean	3.49E+03	3.34E+03	9.00E+02	7.97E+02	8.08E+02	8.38E+02	8.40E+02	8.77E+02	**7.91E+02**
	Std	5.48E+02	8.02E+02	7.88E+01	3.91E+01	3.78E+01	2.99E+01	1.39E+01	8.78E+02	**3.69E+01**
f_{29}	Mean	5.44E+05	2.11E+03	**2.28E+02**	6.30E+02	8.41E+05	8.66E+02	7.17E+02	7.36E+02	3.45E+05
	Std	2.61E+06	4.70E+02	**2.90E+00**	2.04E+02	2.66E+06	1.62E+02	5.08E+00	7.42E+02	1.71E+06
f_{30}	Mean	2.49E+04	2.96E+03	7.10E+02	**4.85E+02**	2.34E+03	2.79E+03	1.25E+03	2.00E+03	1.95E+03
	Std	2.26E+04	8.46E+02	1.14E+02	**1.12E+02**	1.38E+03	1.22E+03	6.14E+02	2.08E+03	7.96E+02

1: **procedure** T0
2: **for** $i = 1$ **to** $10e6$ **do**
3: $x = 0.55 + i$; $x = x + x$; $x = \frac{x}{2}$; $x = x^2$;
4: $x = \text{sqrt}(x)$; $x = \log(x)$; $x = \exp(x)$; $x = \frac{x}{x+2}$
5: **end for**
6: **end procedure**

The computing time of the test program is denoted as $T0$. Let us assume that variable $T1$ presents the time required for evaluating the benchmark function f_{18} and variable $T2$ the time of HSA-CS execution for function f_{18} within 200,000 evaluations for each dimension. Variable $\hat{T2}$ is an average of $T2$ values obtained in five independent runs. The computational complexity of the algorithm HSA-CS is reflected by the measured and calculated variables $T0$, $T1$, $\hat{T2}$ and $(\hat{T2} - T1)/T0$ for each of the observed dimension $D = \{10, 30, 50\}$ (Table 8).

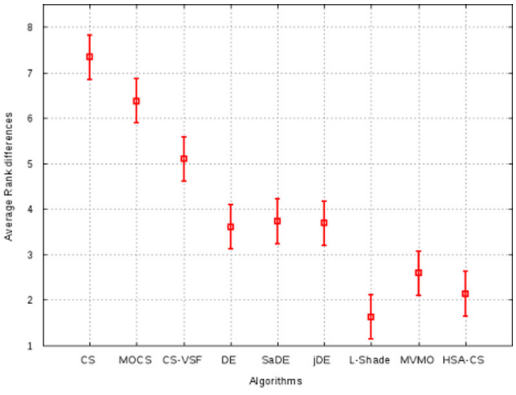
The variables in the table indicate how the computing complexity relates with the dimension of the problem. Obviously, this calculation is independent of the computing system and programming language in which the measured algorithm is implemented.

4.11. HSA-CS by solving the real-world problems

In this study, the proposed HSA-CS was applied to real-world problems, more specifically, on a benchmark of non-linear constrained optimization problems obtained from the literature. The

Algorithms	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
CS	7.34	[6.85,7.83]	†	$\ll 0.05$	+
MOCS	6.38	[5.89,6.87]	†	$\ll 0.05$	+
CS-VSF	5.10	[4.61,5.59]	†	$\ll 0.05$	+
DE	3.61	[3.12,4.10]	†	$\ll 0.05$	+
SaDE	3.73	[3.24,4.22]	†	$\ll 0.05$	+
jDE	3.69	[3.20,4.18]	†	$\ll 0.05$	+
L-Shade	1.63	[1.14,2.12]	‡	$\ll 0.05$	-
MVMO	2.59	[2.10,3.08]		0.3225	
HSA-CS	2.14	[1.65,2.63]		∞	

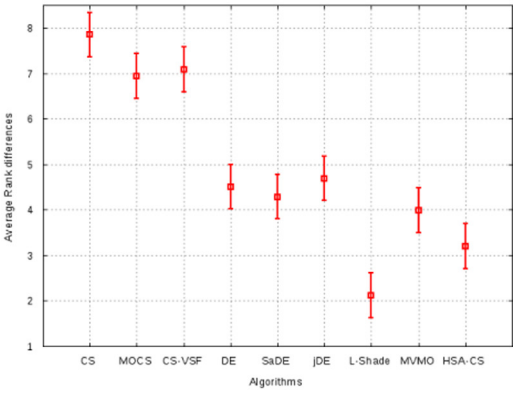
(a) $D = 10$



(b) $D = 10$

Algorithms	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
CS	7.85	[7.36,8.34]	†	$\ll 0.05$	+
MOCS	6.94	[6.45,7.43]	†	$\ll 0.05$	+
CS-VSF	7.09	[6.60,7.58]	†	$\ll 0.05$	+
DE	4.51	[4.02,5.00]	†	$\ll 0.05$	+
SaDE	4.29	[3.80,4.78]	†	$\ll 0.05$	+
jDE	4.69	[4.20,5.18]	†	$\ll 0.05$	+
L-Shade	2.12	[1.63,2.61]	‡	$\ll 0.05$	-
MVMO	3.99	[3.50,4.48]	†	0.3480	
HSA-CS	3.20	[2.71,3.69]	†	∞	

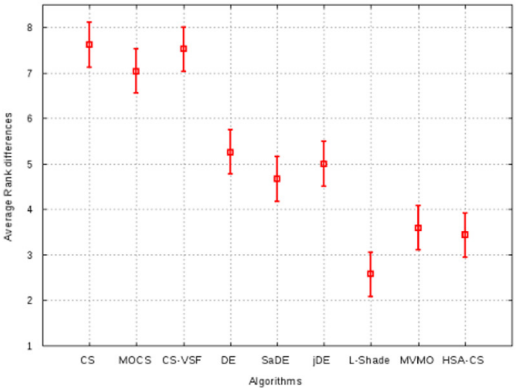
(c) $D = 30$



(d) $D = 30$

Algorithms	Fri.	Nemenyi		Wilcoxon	
		CD	S.	p-value	S.
CS	7.62	[7.13,8.11]	†	$\ll 0.05$	+
MOCS	7.04	[6.55,7.53]	†	$\ll 0.05$	+
CS-VSF	7.52	[7.03,8.01]	†	$\ll 0.05$	+
DE	5.26	[4.77,5.75]	†	$\ll 0.05$	+
SaDE	4.67	[4.18,5.16]	†	$\ll 0.05$	+
jDE	5.00	[4.51,5.49]	†	$\ll 0.05$	+
L-Shade	2.57	[2.08,3.06]	‡	$\ll 0.05$	-
MVMO	3.59	[3.10,4.08]	†	0.1195	
HSA-CS	3.43	[2.94,3.92]		∞	

(e) $D = 50$



(f) $D = 50$

Fig. 5. Comparison of algorithms.

Table 8
Computational complexity (all times are in seconds).

D	$T0$	$T1$	$\hat{T2}$	$(\hat{T2} - T1)/T0$
10	0.036	0.051	0.839	21.89
30		0.193	3.674	96.69
50		0.539	5.779	145.56

study based on the work of Gandomi et al. [28], where the authors compared the results of their modified CS with the results of other algorithms by solving the 13 well-known constrained optimization problems. For our purposes, we selected two which in our opinion were the more interesting problems from this benchmark, i.e.,

- pressure vessel design and
- speed reducer.

These two problems are described in the remainder of the paper. Then, the results of experiments are illustrated.

4.11.1. Pressure vessel design

This problem deals with the designing of a compressed air storage tank with a working pressure of 1000 psi and minimum volume of 750 ft^3. The pressure vessel is capped at both ends by hemispherical heads. The shell is to be made into two halves which are joined by two longitudinal welds to form a cylinder. The design variables are defined by the vector $\mathbf{x} = [x_1, x_2, x_3, x_4]^T$. x_1 represents the shell thickness, x_2 is the spherical head thickness, while x_3 and x_4 are the radius and the length of the shell, respectively. The objective in this problem is to minimize the manufacturing cost of the pressure vessel, which is expressed as

$$f(\mathbf{x}) = 0.6224x_1x_3x_4 + 1.7781x_2x_3x_3 + 3.1661x_1x_1x_4 + 19.84x_1x_1x_3, \tag{13}$$

subject to

$$-x_1 + 0.0193x_3 \le 0, \quad -x_2 + 0.00954x_3 \le 0,$$

$$-\pi x_3x_3x_4 - \frac{4\pi x_3x_3x_3}{3} + 1,296,000 \le 0, \quad x_4 - 240 \le 0. \tag{14}$$

The objective function is a sum of material cost, welding cost and forming cost. The design variables are bounded as $0.0625 \le x_1, x_2 \le 99_*0.0625$, where x_1 and x_2 are discrete values, and $10 \le x_3, x_4 \le 200$.

Table 9
Parameter setting by real-problem optimization.

Algorithm	NP	MAX_FE
(a) Pressure vessel		
Gandomi [28]	25	15,000
Kaveh [58]	20	?
He [59]	70	200,000
Coello [60]	30	150,000
Huang [61]	50 × 6	240,000
HSA-CS	100	20,000
(b) Speed reducer		
Gandomi [28]	50	250,000
Ray [62]	?	?
Akhtar [63]	100	20,000
Ku [64]	?	?
Montes [65]	100	30,000
HSA-CS	100	35,000

4.11.2. Speed reducer

This problem is a structural optimization problem, where the objective is to minimize the total weight of the speed reducer. The constraints cover the limitations on the bending stress of the gear teeth, surface stress, transverse deflections of shafts due to transmitted force, and stresses in these shafts. More information can be found in [28]. The mathematical formulation is given as:

$$f(\mathbf{x}) = 0.7854x_1x_2^2(3.3x_3^2 + 14.9334x_3 - 43.0934) - 1.508x_1(x_6^2 + x_7^2) + 7.477(x_6^3 + x_7^3) + 0.7854(x_4x_6^2 + x_5x_7^2) \tag{15}$$

subject to

$$\frac{27}{x_1x_2^2x_3} - 1 \le 0, \quad \frac{397.5}{x_1x_2^2x_3^2} - 1 \le 0,$$
$$\frac{1.93x_4^3}{x_2x_3x_6^4} - 1 \le 0, \quad \frac{1.93x_5^3}{x_2x_3x_7^4} - 1 \le 0,$$
$$\frac{\sqrt{\left(\frac{745x_4}{x_2x_3}\right)^2 + 16.9\times 10^6}}{110x_6^3} - 1 \le 0,$$
$$\frac{\sqrt{\left(\frac{745x_5}{x_2x_3}\right)^2 + 157.9\times 10^6}}{85x_7^3} - 1 \le 0,$$
$$\frac{x_2x_3}{40} - 1 \le 0, \quad \frac{5x_2}{x_1} - 1 \le 0,$$
$$\frac{x_1}{12x_2} - 1 \le 0, \quad \frac{1.5x_6 + 1.9}{x_4} - 1 \le 0,$$
$$\frac{1.1x_7 + 1.9}{x_5} - 1 \le 0, \tag{16}$$

where $2.6 \le x_1 \le 3.6$, $0.7 \le x_2 \le 0.8$, $17 \le x_3 \le 28$, $7.3 \le x_4 \le 8.3$, $7.8 \le x_5 \le 8.3$, $2.9 \le x_6 \le 3.9$, $5.0 \le x_7 \le 5.5$.

4.11.3. Results of solving the real-world optimization problems

The results of the HSA-CS obtained by optimizing two real-world design optimization problems were compared with the other stochastic nature-inspired population-based algorithms as proposed and reported by Gandomi et al. in [28]. Unfortunately, this comparison is not completely fair because the algorithms used in this comparative study typically employed different algorithm parameters. Thus, the most crucial is the parameter maximum number of fitness function evaluations that usually terminates the

Table 10
Real world design optimization results for the pressure vessel design and three-bar truss problem.

Algorithm	Best	Worst	Mean	Std	CV
Pressure vessel design					
Gandomi et al. [28]	**6059.714**	6495.347	6447.76	502.693	No
Kaveh et al. [58]	6059.73	6150.13	6081.78	67.242	No
He at al. [59]	6061.078	6363.804	6147.133	86.454	No
Coello and Corte's [60]	6061.123	6734.085	7368.06	457.9959	No
Huang et al. [61]	6059.734	6085.23	6371.046	43.013	No
HSA-CS*	**6059.714**	**6370.779**	**6076.084**	44.549	No
Speed reducer					
Gandomi et al. [28]	3000.98	3009	3007.19	4.96	No
Ray et al. [62]	2732.9	2780.3	2758.88	N/A	Yes
Akhtar et al. [63]	3008.08	3028.28	3012.12	N/A	No
Ku et al. [64]	2876.118	N/A	N/A	N/A	No
Montes et al. [65]	3025.005	3088.7778	3078.5918	N/A	No
HSA-CS	**2996.21**	**2996.21**	**2996.21**	0	No

algorithm run [57]. In order to show how fair this comparative study was, the used parameter settings used in different papers are accumulated in Table 9. The table consists of settings of two parameters, i.e., the population size (NP) and the maximum number of fitness function evaluations (MAX_FE) for each of two observed problems. The results of the algorithms were compared by the following measures: minimum, maximum, mean and standard deviation. In tables, constraint violations for the specific algorithm are denoted with string "yes" in column "CV". The sign '?' in the tables denote that the corresponding setting of the parameter was not reported in the paper or the paper could not be obtained by the authors.

From Table 9, it can be seen that the algorithms in the study applied different population sizes and different termination conditions, in general. Consequently, it is hard to argue that comparison in this way was fair performed.

The results of the HSA-CS by solving the pressure vessel design problems (Table 10) were compared with the results obtained by the following algorithms: the CS algorithm [28], ant colony optimization [58], co-evolutionary particle swarm optimization [59], hybridizing genetic algorithm with artificial immune system [60] and co-evolutionary differential evolution [61]. Despite the HSA-CS, the following algorithms appeared by solving the speed reducer problem: the CS [28], the new swarm algorithm using a sharing of information among individuals [62], the socio-behavior simulation model [63], Taguchi-aided search method [64] and the simple evolutionary algorithm [65].

The best results by solving the first problem were obtained by both CS algorithms. Interestingly, these algorithms achieved the optimal design. On the other hand, the second real-world problem was solved best by the HSA-CS, while the other algorithms achieved worse results. In summary, the results of these experiments showed that HSA-CS is also suitable for optimizing the real-world design optimization problems.

5. Conclusion

The nature-inspired algorithms are too general for optimal solving all the problems with which humans are confronted today. In order to improve the results of such a general problem solver by solving a certain problem, domain specific knowledge needs to be conducted during the solving. Typically, this knowledge can be conducted using adaptation and hybridization.

As a general problem solver, the cuckoo search (CS) was used which belongs to the domain of swarm intelligence. The original CS algorithm was adapted and hybridized with three features: a self-adaptation of the CS algorithm parameters, a mechanism for balancing the stochastic exploration strategies within the CS search process, and linear population reduction. In this way, a novel hybrid self-adaptive CS algorithm (HSA-CS) was obtained which was applied to a CEC-2014 benchmark function suite.

During the extensive experimental work, the characteristics of the built-in features were analyzed in detail. It turned out that the results of the original CS algorithm were significantly outperformed by the novel HSA-CS when solving the CEC-2014 benchmark function suite. Moreover, the obtained results of the HSA-CS were also comparable to the other standard and powerful algorithms, like DE, jDE, SaDE, and MVMO, while outperforming the results of the winner of the competition CEC-2014 algorithm L-Shade remains a great challenge for the future.

The HSA-CS employs a deterministic setting of the balancing probability. Unfortunately, this parameter demands a lot of work when the optimal value of it should be found. Therefore, the adaptive (or self-adaptive) control of this parameter would be performed in future work.

Appendix

See Tables 11–21.

Table 11
Influence of the starting population size NP_0 on the quality of solutions by dimensionality of the problem $D = 10$ (1/3).

Func.	NP_0	Best	Worst	Mean	Std	Median
f_1	100	0.00E+00	8.37E−06	1.75E−07	1.16E−06	0.00E+00
	200	0.00E+00	2.98E−06	5.90E−08	4.13E−07	0.00E+00
	400	0.00E+00	5.32E−07	**1.13E−08**	7.37E−08	0.00E+00
	600	0.00E+00	9.39E−07	5.31E−08	1.41E−07	0.00E+00
	800	0.00E+00	1.10E−05	8.41E−07	1.77E−06	2.01E−07
	1000	9.21E−08	4.42E−05	4.96E−06	8.55E−06	1.62E−06
f_2	100	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	200	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	400	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	600	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	800	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	1000	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
f_3	100	2.48E−08	3.74E−03	5.09E−04	9.01E−04	2.17E−04
	200	1.15E−06	2.51E−03	2.20E−04	4.37E−04	7.20E−05
	400	4.35E−06	5.45E−04	**8.00E−05**	1.09E−04	2.25E−05
	600	2.48E−07	1.63E−03	1.50E−04	2.55E−04	7.09E−05
	800	7.99E−05	1.59E−03	4.70E−04	3.25E−04	3.69E−04
	1000	7.67E−08	2.00E−03	2.06E−04	3.29E−04	1.08E−04
f_4	100	0.00E+00	3.48E+01	**1.46E+01**	1.69E+01	6.48E−02
	200	8.80E−07	3.48E+01	1.57E+01	1.67E+01	4.34E+00
	400	3.79E−07	3.48E+01	2.09E+01	1.66E+01	3.48E+01
	600	1.57E−05	3.48E+01	1.67E+01	1.71E+01	4.34E+00
	800	4.93E−03	3.48E+01	1.92E+01	1.72E+01	3.48E+01
	1000	8.72E−05	3.48E+01	1.57E+01	1.67E+01	4.34E+00
f_5	100	2.70E−05	2.00E+01	1.73E+01	6.57E+00	2.00E+01
	200	1.08E−02	2.00E+01	1.73E+01	6.65E+00	2.00E+01
	400	1.27E−02	2.00E+01	1.78E+01	6.09E+00	2.00E+01
	600	1.15E−01	2.00E+01	1.80E+01	5.68E+00	2.00E+01
	800	1.95E+00	2.00E+01	1.83E+01	4.51E+00	2.00E+01
	1000	3.54E−02	2.00E+01	**1.63E+01**	7.02E+00	2.00E+01
f_6	100	2.05E−03	9.07E−01	2.50E−02	1.25E−01	5.67E−03
	200	4.27E−03	7.85E−02	**1.46E−02**	1.09E−02	1.27E−02
	400	8.55E−03	8.00E−02	2.49E−02	1.37E−02	2.09E−02
	600	1.12E−02	9.53E−02	3.40E−02	1.67E−02	3.16E−02
	800	2.20E−02	1.47E−01	6.32E−02	3.02E−02	5.64E−02
	1000	3.05E−02	1.68E−01	8.13E−02	3.47E−02	7.52E−02
f_7	100	0.00E+00	3.66E−02	3.98E−03	6.81E−03	5.20E−05
	200	5.39E−08	2.52E−02	1.91E−03	4.50E−03	8.51E−05
	400	3.17E−07	1.06E−02	9.63E−04	2.49E−03	7.65E−05
	600	1.80E−07	1.45E−02	**9.42E−04**	2.61E−03	8.53E−05
	800	6.67E−06	2.39E−02	3.72E−03	5.08E−03	9.38E−04
	1000	6.58E−06	1.71E−02	2.19E−03	3.71E−03	4.81E−04
f_8	100	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	200	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	400	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	600	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	800	0.00E+00	1.69E−07	4.89E−08	4.93E−08	3.22E−08
	1000	0.00E+00	1.05E−06	1.42E−07	2.24E−07	5.17E−08
f_9	100	2.84E−07	4.03E+00	1.77E+00	1.02E+00	1.99E+00
	200	2.98E−05	2.99E+00	**1.43E+00**	7.77E−01	1.04E+00
	400	2.63E−05	3.21E+00	1.64E+00	8.24E−01	1.99E+00
	600	8.03E−02	4.02E+00	2.00E+00	7.86E−01	2.02E+00
	800	1.08E+00	4.08E+00	2.41E+00	6.86E−01	2.24E+00
	1000	1.00E+00	5.00E+00	2.46E+00	1.00E+00	2.08E+00
f_{10}	100	6.25E−02	6.77E+00	1.07E+00	1.58E+00	2.50E−01
	200	0.00E+00	5.00E−01	1.62E−01	1.06E−01	1.87E−01
	400	0.00E+00	3.12E−01	1.04E−01	7.69E−02	6.25E−02
	600	5.98E−08	1.87E−01	6.98E−02	5.75E−02	6.25E−02
	800	2.44E−07	1.87E−01	5.02E−02	4.79E−02	6.25E−02
	1000	2.16E−08	1.87E−01	**4.90E−02**	5.57E−02	6.25E−02

Table 12
Influence of the starting population size NP_0 on the quality of solutions by dimensionality of the problem $D = 10$ (2/3).

Func.	NP_0	Best	Worst	Mean	Std	Median
f_{11}	100	1.88E−01	1.42E+02	1.50E+01	2.60E+01	6.97E+00
	200	1.68E−01	6.21E+01	**1.30E+01**	1.16E+01	1.07E+01
	400	5.82E−01	7.01E+01	1.44E+01	1.27E+01	1.05E+01
	600	5.44E−01	6.54E+01	1.85E+01	1.36E+01	1.53E+01
	800	1.68E+00	2.32E+02	5.14E+01	5.12E+01	3.46E+01
	1000	7.36E+00	2.34E+02	3.51E+01	4.14E+01	2.34E+01
f_{12}	100	4.86E−03	9.75E−02	**3.64E−02**	1.84E−02	3.45E−02
	200	7.67E−03	1.49E−01	6.45E−02	3.18E−02	5.82E−02
	400	3.71E−02	2.04E−01	1.04E−01	4.56E−02	9.31E−02
	600	2.89E−02	2.47E−01	1.21E−01	5.06E−02	1.18E−01
	800	8.81E−02	4.06E−01	1.85E−01	6.60E−02	1.83E−01
	1000	2.19E−02	3.27E−01	1.60E−01	7.87E−02	1.67E−01
f_{13}	100	4.54E−02	1.57E−01	1.03E−01	2.48E−02	1.01E−01
	200	4.51E−02	1.46E−01	**9.07E−02**	2.10E−02	9.36E−02
	400	5.46E−02	1.31E−01	9.18E−02	1.99E−02	9.17E−02
	600	5.84E−02	1.25E−01	9.44E−02	1.89E−02	9.67E−02
	800	4.85E−02	1.40E−01	1.00E−01	2.20E−02	1.04E−01
	1000	4.53E−02	1.36E−01	9.93E−02	2.02E−02	1.01E−01
f_{14}	100	3.27E−02	2.32E−01	1.03E−01	3.92E−02	1.07E−01
	200	3.31E−02	1.81E−01	1.03E−01	3.53E−02	9.67E−02
	400	4.88E−02	1.98E−01	1.04E−01	2.78E−02	1.02E−01
	600	4.66E−02	1.74E−01	**9.73E−02**	2.73E−02	9.83E−02
	800	6.15E−02	2.22E−01	1.16E−01	2.91E−02	1.17E−01
	1000	3.88E−02	1.74E−01	1.11E−01	2.57E−02	1.11E−01
f_{15}	100	2.35E−01	7.90E−01	4.74E−01	1.08E−01	4.78E−01
	200	2.24E−01	7.29E−01	4.73E−01	1.05E−01	4.62E−01
	400	2.38E−01	7.37E−01	**4.71E−01**	1.16E−01	4.92E−01
	600	2.81E−01	8.46E−01	5.33E−01	1.30E−01	5.20E−01
	800	3.74E−01	9.04E−01	5.82E−01	1.13E−01	5.93E−01
	1000	3.05E−01	1.02E+00	5.73E−01	1.57E−01	5.80E−01
f_{16}	100	3.45E−01	1.96E+00	**1.22E+00**	3.54E−01	1.31E+00
	200	5.87E−01	2.03E+00	1.30E+00	3.22E−01	1.28E+00
	400	7.26E−01	2.12E+00	1.42E+00	3.26E−01	1.48E+00
	600	6.84E−01	2.22E+00	1.53E+00	3.13E−01	1.51E+00
	800	9.69E−01	2.19E+00	1.62E+00	3.05E−01	1.61E+00
	1000	1.08E+00	2.15E+00	1.67E+00	2.61E−01	1.70E+00
f_{17}	100	4.16E−01	2.86E+02	4.98E+01	6.49E+01	1.64E+01
	200	1.15E+00	1.59E+02	2.34E+01	3.53E+01	1.12E+01
	400	9.50E−01	1.61E+02	1.55E+01	2.66E+01	7.52E+00
	600	2.09E+00	4.15E+01	**9.53E+00**	6.85E+00	8.41E+00
	800	2.61E+00	2.33E+01	1.34E+01	5.34E+00	1.36E+01
	1000	2.33E+00	6.56E+01	1.56E+01	1.02E+01	1.56E+01
f_{18}	100	8.73E−03	2.05E+00	3.07E−01	3.60E−01	2.14E−01
	200	1.58E−02	1.20E+00	2.10E−01	2.12E−01	1.64E−01
	400	1.03E−02	1.08E+00	**1.93E−01**	1.65E−01	1.63E−01
	600	5.85E−02	1.06E+00	2.83E−01	2.07E−01	2.19E−01
	800	5.19E−02	1.44E+00	3.01E−01	2.05E−01	2.65E−01
	1000	4.60E−02	1.14E+00	3.89E−01	2.51E−01	3.39E−01
f_{19}	100	2.77E−02	1.52E+00	3.09E−01	3.70E−01	1.50E−01
	200	5.01E−02	1.07E+00	**2.23E−01**	2.29E−01	1.69E−01
	400	3.82E−02	1.02E+00	2.32E−01	1.86E−01	1.79E−01
	600	4.71E−02	6.54E−01	2.45E−01	1.29E−01	2.20E−01
	800	1.48E−01	8.44E−01	3.46E−01	1.25E−01	3.25E−01
	1000	1.01E−01	1.12E+00	3.73E−01	1.92E−01	3.44E−01
f_{20}	100	1.50E−02	6.33E−01	**2.18E−01**	1.56E−01	1.60E−01
	200	7.01E−02	6.18E−01	2.62E−01	1.37E−01	2.18E−01
	400	6.67E−02	6.34E−01	3.29E−01	1.47E−01	2.83E−01
	600	1.02E−01	6.75E−01	3.49E−01	1.39E−01	3.30E−01
	800	2.60E−01	7.92E−01	5.29E−01	1.41E−01	5.44E−01
	1000	2.08E−01	8.42E−01	4.64E−01	1.57E−01	4.47E−01

Table 13
Influence of the starting population size NP_0 on the quality of solutions by dimensionality of the problem $D = 10$ (3/3).

Func.	NP_0	Best	Worst	Mean	Std	Median
f_{21}	100	2.23E−02	5.82E+01	1.53E+00	8.02E+00	3.68E−01
	200	3.66E−02	9.27E−01	**3.89E−01**	2.34E−01	3.71E−01
	400	5.00E−02	1.13E+00	4.49E−01	2.44E−01	4.02E−01
	600	1.19E−01	1.17E+00	5.19E−01	2.26E−01	5.34E−01
	800	2.88E−01	1.43E+00	8.30E−01	2.47E−01	8.38E−01
	1000	1.83E−01	1.22E+00	7.69E−01	2.64E−01	8.02E−01
f_{22}	100	2.01E−02	7.86E−01	1.98E−01	1.66E−01	1.22E−01
	200	2.51E−02	5.57E−01	**1.59E−01**	1.14E−01	1.22E−01
	400	1.19E−01	9.91E−01	3.80E−01	1.92E−01	3.38E−01
	600	1.70E−01	1.47E+00	5.67E−01	2.56E−01	5.22E−01
	800	2.45E−01	1.67E+00	8.61E−01	3.20E−01	7.80E−01
	1000	2.13E−01	2.76E+00	1.06E+00	5.70E−01	8.92E−01
f_{23}	100	3.29E+02	3.29E+02	**3.29E+02**	1.32E−05	3.29E+02
	200	3.29E+02	3.29E+02	**3.29E+02**	1.32E−05	3.29E+02
	400	3.29E+02	3.29E+02	**3.29E+02**	1.32E−05	3.29E+02
	600	3.29E+02	3.29E+02	**3.29E+02**	1.38E−05	3.29E+02
	800	3.29E+02	3.29E+02	**3.29E+02**	1.32E−05	3.29E+02
	1000	3.29E+02	3.29E+02	**3.29E+02**	1.08E−05	3.29E+02
f_{24}	100	1.00E+02	1.12E+02	**1.07E+02**	2.94E+00	1.07E+02
	200	1.00E+02	1.10E+02	**1.07E+02**	2.18E+00	1.08E+02
	400	1.00E+02	1.10E+02	**1.07E+02**	1.93E+00	1.08E+02
	600	1.00E+02	1.10E+02	**1.07E+02**	1.94E+00	1.08E+02
	800	1.01E+02	1.11E+02	1.08E+02	1.63E+00	1.08E+02
	1000	1.00E+02	1.11E+02	1.08E+02	1.99E+00	1.08E+02
f_{25}	100	1.00E+02	2.01E+02	1.19E+02	1.94E+01	1.13E+02
	200	1.00E+02	2.01E+02	**1.14E+02**	1.86E+01	1.12E+02
	400	1.00E+02	2.01E+02	1.15E+02	1.34E+01	1.14E+02
	600	1.00E+02	2.01E+02	1.18E+02	1.30E+01	1.16E+02
	800	1.09E+02	1.31E+02	1.19E+02	5.36E+00	1.17E+02
	1000	1.09E+02	1.96E+02	1.19E+02	1.17E+01	1.17E+02
f_{26}	100	1.00E+02	1.00E+02	**1.00E+02**	2.12E−02	1.00E+02
	200	1.00E+02	1.00E+02	**1.00E+02**	1.67E−02	1.00E+02
	400	1.00E+02	1.00E+02	**1.00E+02**	1.78E−02	1.00E+02
	600	1.00E+02	1.00E+02	**1.00E+02**	1.99E−02	1.00E+02
	800	1.00E+02	1.00E+02	**1.00E+02**	2.37E−02	1.00E+02
	1000	1.00E+02	1.00E+02	**1.00E+02**	2.25E−02	1.00E+02
f_{27}	100	7.53E−01	4.00E+02	1.79E+01	7.73E+01	1.77E+00
	200	1.21E+00	3.31E+00	**1.86E+00**	4.58E−01	1.83E+00
	400	8.48E−01	4.86E+00	1.90E+00	7.27E−01	1.71E+00
	600	8.94E−01	7.86E+00	2.05E+00	9.32E−01	1.92E+00
	800	1.28E+00	2.62E+00	1.90E+00	3.24E−01	1.90E+00
	1000	1.10E+00	2.68E+00	1.97E+00	3.34E−01	2.00E+00
f_{28}	100	3.57E+02	4.81E+02	4.02E+02	5.30E+01	3.69E+02
	200	3.57E+02	4.81E+02	3.88E+02	4.87E+01	3.57E+02
	400	3.57E+02	4.81E+02	3.80E+02	4.16E+01	3.57E+02
	600	3.57E+02	4.69E+02	3.78E+02	3.80E+01	3.57E+02
	800	3.57E+02	4.70E+02	3.72E+02	3.33E+01	3.57E+02
	1000	3.57E+02	4.64E+02	**3.66E+02**	2.71E+01	3.57E+02
f_{29}	100	2.15E+02	2.26E+02	2.23E+02	1.70E+00	2.23E+02
	200	2.22E+02	2.24E+02	**2.22E+02**	6.14E−01	2.22E+02
	400	2.22E+02	2.24E+02	**2.22E+02**	5.88E−01	2.22E+02
	600	2.22E+02	2.24E+02	**2.22E+02**	4.84E−01	2.22E+02
	800	2.22E+02	2.23E+02	**2.22E+02**	4.22E−01	2.22E+02
	1000	2.22E+02	2.24E+02	**2.22E+02**	5.07E−01	2.22E+02
f_{30}	100	4.54E+02	5.87E+02	4.88E+02	3.76E+01	4.65E+02
	200	4.62E+02	5.39E+02	4.72E+02	1.77E+01	4.63E+02
	400	4.62E+02	5.50E+02	4.68E+02	1.55E+01	4.63E+02
	600	4.62E+02	5.10E+02	4.67E+02	1.20E+01	4.63E+02
	800	4.62E+02	5.07E+02	4.67E+02	1.21E+01	4.63E+02
	1000	4.63E+02	4.87E+02	**4.66E+02**	5.18E+00	4.64E+02

Table 14
Influence of the starting population size NP_0 on the quality of solutions by dimensionality of the problem $D = 30$ (1/3).

Func.	NP_0	Best	Worst	Mean	Std	Median
f_1	100	1.37E+03	7.16E+04	2.09E+04	1.75E+04	1.36E+04
	200	6.39E+03	6.92E+04	2.16E+04	1.17E+04	2.00E+04
	400	1.13E+03	6.78E+04	2.14E+04	1.53E+04	1.89E+04
	600	2.38E+03	6.29E+04	**1.66E+04**	1.23E+04	1.43E+04
	800	2.73E+03	5.66E+04	2.12E+04	1.26E+04	1.70E+04
	1000	6.85E+03	7.08E+04	2.33E+04	1.53E+04	1.77E+04
f_2	100	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	200	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	400	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	600	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	800	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	1000	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
f_3	100	0.00E+00	9.59E−03	2.28E−03	2.60E−03	8.82E−04
	200	3.56E−07	5.61E−03	5.62E−04	9.53E−04	2.07E−04
	400	3.09E−06	2.69E−03	1.62E−04	3.81E−04	4.85E−05
	600	5.95E−06	3.11E−04	**7.19E−05**	7.21E−05	4.46E−05
	800	4.14E−06	3.59E−04	8.98E−05	8.09E−05	6.17E−05
	1000	7.42E−07	2.96E−03	1.62E−04	4.05E−04	9.32E−05
f_4	100	8.84E−06	6.34E+01	**1.34E+00**	8.79E+00	7.54E−03
	200	4.24E−05	6.81E+01	2.76E+00	1.32E+01	3.91E−02
	400	7.87E−03	6.78E+01	3.01E+00	1.07E+01	6.81E−01
	600	2.21E−03	6.91E+01	3.71E+00	1.31E+01	8.97E−01
	800	3.93E−02	7.20E+01	6.68E+00	1.87E+01	1.19E+00
	1000	1.09E−01	7.16E+01	1.31E+01	2.53E+01	1.77E+00
f_5	100	2.00E+01	2.00E+01	**2.00E+01**	2.95E−03	2.00E+01
	200	2.00E+01	2.01E+01	**2.00E+01**	1.33E−02	2.00E+01
	400	2.00E+01	2.01E+01	**2.00E+01**	2.80E−02	2.00E+01
	600	2.00E+01	2.03E+01	**2.00E+01**	6.01E−02	2.00E+01
	800	2.00E+01	2.04E+01	2.01E+01	8.52E−02	2.00E+01
	1000	2.00E+01	2.03E+01	**2.00E+01**	5.11E−02	2.00E+01
f_6	100	3.26E−01	6.93E+00	2.76E+00	1.55E+00	2.53E+00
	200	1.44E−01	8.04E+00	1.67E+00	1.84E+00	9.55E−01
	400	7.88E−02	1.01E+01	1.76E+00	2.16E+00	9.16E−01
	600	1.48E−01	9.02E+00	**1.43E+00**	1.97E+00	7.37E−01
	800	2.01E−01	8.49E+00	1.55E+00	1.53E+00	1.06E+00
	1000	3.16E−01	8.32E+00	1.73E+00	1.79E+00	9.89E−01
f_7	100	0.00E+00	1.23E−02	5.80E−04	2.37E−03	1.15E−08
	200	0.00E+00	2.21E−02	4.37E−04	3.07E−03	0.00E+00
	400	0.00E+00	9.86E−03	1.93E−04	1.37E−03	0.00E+00
	600	0.00E+00	1.65E−06	4.04E−08	2.33E−07	0.00E+00
	800	0.00E+00	1.63E−07	**4.69E−09**	2.33E−08	0.00E+00
	1000	0.00E+00	1.95E−07	1.05E−08	3.50E−08	0.00E+00
f_8	100	0.00E+00	6.96E+00	2.20E+00	1.74E+00	1.99E+00
	200	0.00E+00	1.99E+00	3.90E−01	5.94E−01	0.00E+00
	400	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	600	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	800	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	1000	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
f_9	100	1.29E+01	3.48E+01	2.52E+01	5.15E+00	2.59E+01
	200	9.96E+00	3.08E+01	2.01E+01	4.55E+00	1.99E+01
	400	1.19E+01	3.29E+01	**1.97E+01**	4.64E+00	1.99E+01
	600	1.10E+01	3.11E+01	1.99E+01	4.72E+00	2.08E+01
	800	9.14E+00	3.16E+01	2.08E+01	4.33E+00	2.06E+01
	1000	9.95E+00	3.49E+01	2.12E+01	4.98E+00	2.00E+01
f_{10}	100	3.56E+00	1.33E+02	1.27E+01	1.76E+01	9.76E+00
	200	1.26E+00	1.20E+01	5.44E+00	2.43E+00	5.19E+00
	400	1.46E−01	4.76E+00	1.79E+00	1.11E+00	1.41E+00
	600	1.46E−01	4.55E+00	8.78E−01	9.04E−01	2.71E−01
	800	1.25E−01	2.42E+00	5.90E−01	6.55E−01	2.29E−01
	1000	6.25E−02	1.35E+00	**3.83E−01**	4.30E−01	1.87E−01

Table 15
Influence of the starting population size NP_0 on the quality of solutions by dimensionality of the problem $D = 30$ (2/3).

Func.	NP_0	Best	Worst	Mean	Std	Median
f_{11}	100	9.04E+02	2.09E+03	**1.47E+03**	2.77E+02	1.50E+03
	200	6.10E+02	2.15E+03	1.49E+03	3.41E+02	1.48E+03
	400	9.35E+02	2.36E+03	1.67E+03	3.24E+02	1.64E+03
	600	7.44E+02	2.57E+03	1.82E+03	4.03E+02	1.88E+03
	800	1.07E+03	2.63E+03	1.98E+03	3.46E+02	2.05E+03
	1000	1.09E+03	3.06E+03	2.06E+03	3.83E+02	2.09E+03
f_{12}	100	5.61E−02	3.07E−01	**1.70E−01**	5.29E−02	1.62E−01
	200	6.90E−02	3.34E−01	1.98E−01	6.73E−02	1.85E−01
	400	1.14E−01	4.30E−01	2.38E−01	7.70E−02	2.41E−01
	600	5.61E−02	5.33E−01	2.65E−01	9.55E−02	2.43E−01
	800	9.76E−02	4.90E−01	2.51E−01	9.66E−02	2.46E−01
	1000	7.85E−02	6.67E−01	2.77E−01	1.16E−01	2.51E−01
f_{13}	100	1.67E−01	3.05E−01	2.33E−01	3.32E−02	2.31E−01
	200	1.14E−01	3.03E−01	**2.12E−01**	3.57E−02	2.09E−01
	400	1.60E−01	2.71E−01	2.17E−01	2.97E−02	2.18E−01
	600	1.09E−01	2.73E−01	2.15E−01	3.51E−02	2.15E−01
	800	1.35E−01	3.06E−01	2.15E−01	3.19E−02	2.19E−01
	1000	1.49E−01	3.06E−01	2.19E−01	3.72E−02	2.15E−01
f_{14}	100	1.36E−01	6.56E−01	2.54E−01	7.26E−02	2.40E−01
	200	1.55E−01	3.08E−01	2.28E−01	3.36E−02	2.26E−01
	400	1.58E−01	2.98E−01	2.30E−01	3.01E−02	2.36E−01
	600	1.60E−01	2.87E−01	2.27E−01	2.96E−02	2.25E−01
	800	1.74E−01	2.74E−01	2.33E−01	2.51E−02	2.32E−01
	1000	1.56E−01	2.72E−01	**2.23E−01**	2.68E−02	2.25E−01
f_{15}	100	1.49E+00	3.49E+00	2.51E+00	4.53E−01	2.51E+00
	200	1.63E+00	3.07E+00	**2.36E+00**	3.79E−01	2.34E+00
	400	1.57E+00	3.51E+00	2.52E+00	4.52E−01	2.48E+00
	600	1.97E+00	3.43E+00	2.66E+00	3.82E−01	2.54E+00
	800	1.73E+00	3.62E+00	2.73E+00	4.81E−01	2.76E+00
	1000	1.58E+00	3.93E+00	2.79E+00	4.23E−01	2.85E+00
f_{16}	100	7.95E+00	1.02E+01	**9.28E+00**	4.83E−01	9.29E+00
	200	8.22E+00	1.05E+01	9.50E+00	5.00E−01	9.58E+00
	400	7.87E+00	1.07E+01	9.59E+00	5.81E−01	9.55E+00
	600	6.99E+00	1.08E+01	9.69E+00	6.32E−01	9.84E+00
	800	8.86E+00	1.09E+01	9.90E+00	4.83E−01	9.85E+00
	1000	8.85E+00	1.11E+01	1.01E+01	4.64E−01	1.02E+01
f_{17}	100	9.15E+02	2.62E+03	1.65E+03	4.11E+02	1.62E+03
	200	7.14E+02	3.11E+03	1.42E+03	4.50E+02	1.32E+03
	400	5.92E+02	2.41E+03	1.26E+03	3.55E+02	1.22E+03
	600	5.05E+02	2.10E+03	1.33E+03	3.59E+02	1.37E+03
	800	4.90E+02	2.17E+03	**1.15E+03**	3.67E+02	1.08E+03
	1000	2.23E+02	1.73E+03	1.21E+03	3.59E+02	1.25E+03
f_{18}	100	5.31E+01	2.27E+02	1.28E+02	4.29E+01	1.23E+02
	200	3.19E+01	2.05E+02	8.87E+01	3.72E+01	8.73E+01
	400	3.50E+01	1.34E+02	6.97E+01	1.92E+01	6.79E+01
	600	2.92E+01	1.15E+02	6.32E+01	1.78E+01	6.10E+01
	800	3.71E+01	1.73E+02	6.40E+01	2.09E+01	6.10E+01
	1000	2.90E+01	1.13E+02	**5.92E+01**	1.53E+01	5.70E+01
f_{19}	100	2.99E+00	9.32E+00	5.81E+00	1.51E+00	5.69E+00
	200	2.53E+00	8.18E+00	4.77E+00	1.04E+00	4.66E+00
	400	3.04E+00	6.78E+00	**4.27E+00**	6.67E−01	4.30E+00
	600	2.91E+00	6.04E+00	4.67E+00	6.40E−01	4.74E+00
	800	3.32E+00	5.78E+00	4.83E+00	5.38E−01	4.84E+00
	1000	3.38E+00	7.43E+00	5.00E+00	6.48E−01	4.96E+00
f_{20}	100	7.81E+00	1.87E+02	4.35E+01	3.58E+01	2.81E+01
	200	5.60E+00	5.21E+01	1.63E+01	9.32E+00	1.35E+01
	400	4.90E+00	2.07E+01	1.20E+01	4.02E+00	1.18E+01
	600	4.78E+00	1.99E+01	9.97E+00	3.36E+00	9.30E+00
	800	4.71E+00	1.91E+01	**9.77E+00**	2.74E+00	9.49E+00
	1000	3.36E+00	2.08E+01	8.78E+00	3.03E+00	8.12E+00

Table 16
Influence of the starting population size NP_0 on the quality of solutions by dimensionality of the problem $D = 30$ (3/3).

Func.	NP_0	Best	Worst	Mean	Std	Median
f_{21}	100	2.25E+02	1.24E+03	6.49E+02	2.37E+02	6.15E+02
	200	1.32E+02	9.83E+02	4.52E+02	2.11E+02	4.00E+02
	400	6.51E+01	8.26E+02	3.71E+02	1.60E+02	3.55E+02
	600	1.33E+02	6.41E+02	3.12E+02	1.11E+02	2.94E+02
	800	1.29E+02	4.89E+02	**2.81E+02**	8.66E+01	2.80E+02
	1000	1.35E+02	5.93E+02	2.88E+02	8.70E+01	2.72E+02
f_{22}	100	2.16E+01	2.79E+02	1.33E+02	8.55E+01	1.47E+02
	200	4.03E+00	2.62E+02	7.51E+01	6.36E+01	3.33E+01
	400	2.16E+01	1.55E+02	5.57E+01	4.95E+01	2.71E+01
	600	2.11E+01	1.51E+02	4.73E+01	4.30E+01	2.86E+01
	800	2.14E+01	1.50E+02	**4.20E+01**	3.53E+01	2.87E+01
	1000	2.17E+01	1.58E+02	4.36E+01	3.86E+01	2.82E+01
f_{23}	100	3.15E+02	**3.15E+02**	3.15E+02	0.00E+00	3.15E+02
	200	3.15E+02	3.15E+02	**3.15E+02**	0.00E+00	3.15E+02
	400	3.15E+02	3.15E+02	**3.15E+02**	0.00E+00	3.15E+02
	600	3.15E+02	3.15E+02	**3.15E+02**	0.00E+00	3.15E+02
	800	3.15E+02	3.15E+02	**3.15E+02**	0.00E+00	3.15E+02
	1000	3.15E+02	3.15E+02	**3.15E+02**	3.81E−06	3.15E+02
f_{24}	100	2.22E+02	2.31E+02	2.25E+02	1.68E+00	2.25E+02
	200	2.23E+02	2.29E+02	2.25E+02	1.12E+00	2.24E+02
	400	2.23E+02	2.27E+02	**2.24E+02**	9.32E−01	2.24E+02
	600	2.23E+02	2.29E+02	**2.24E+02**	9.32E−01	2.24E+02
	800	2.23E+02	2.26E+02	**2.24E+02**	6.61E−01	2.24E+02
	1000	2.23E+02	2.26E+02	**2.24E+02**	4.13E−01	2.24E+02
f_{25}	100	2.03E+02	2.08E+02	2.04E+02	1.04E+00	2.03E+02
	200	2.03E+02	2.04E+02	**2.03E+02**	3.22E−01	2.03E+02
	400	2.03E+02	2.05E+02	**2.03E+02**	3.56E−01	2.03E+02
	600	2.03E+02	2.04E+02	**2.03E+02**	3.73E−01	2.03E+02
	800	2.03E+02	2.04E+02	**2.03E+02**	2.75E−01	2.03E+02
	1000	2.03E+02	2.04E+02	**2.03E+02**	3.03E−01	2.03E+02
f_{26}	100	1.00E+02	1.00E+02	**1.00E+02**	4.59E−02	1.00E+02
	200	1.00E+02	1.00E+02	**1.00E+02**	3.69E−02	1.00E+02
	400	1.00E+02	1.00E+02	**1.00E+02**	2.96E−02	1.00E+02
	600	1.00E+02	1.00E+02	**1.00E+02**	3.39E−02	1.00E+02
	800	1.00E+02	1.00E+02	**1.00E+02**	2.83E−02	1.00E+02
	1000	1.00E+02	1.00E+02	**1.00E+02**	3.14E−02	1.00E+02
f_{27}	100	3.01E+02	4.94E+02	3.81E+02	4.15E+01	4.00E+02
	200	3.00E+02	4.02E+02	3.44E+02	4.54E+01	3.16E+02
	400	3.00E+02	4.02E+02	3.22E+02	4.11E+01	3.01E+02
	600	3.00E+02	4.02E+02	**3.13E+02**	3.25E+01	3.01E+02
	800	3.00E+02	4.03E+02	3.20E+02	3.99E+01	3.01E+02
	1000	3.00E+02	4.02E+02	3.15E+02	3.45E+01	3.01E+02
f_{28}	100	6.65E+02	1.23E+03	8.27E+02	7.98E+01	8.20E+02
	200	6.34E+02	8.75E+02	**7.77E+02**	5.01E+01	7.85E+02
	400	6.88E+02	8.48E+02	7.95E+02	2.70E+01	7.98E+02
	600	6.56E+02	8.47E+02	7.91E+02	2.96E+01	7.93E+02
	800	7.65E+02	8.55E+02	7.98E+02	1.95E+01	7.95E+02
	1000	6.70E+02	8.56E+02	7.99E+02	2.66E+01	7.97E+02
f_{29}	100	4.94E+02	1.45E+03	9.14E+02	1.53E+02	8.82E+02
	200	4.01E+02	1.18E+03	8.52E+02	1.36E+02	8.19E+02
	400	7.19E+02	1.11E+03	8.49E+02	9.62E+01	8.24E+02
	600	7.34E+02	1.11E+03	8.25E+02	7.46E+01	8.12E+02
	800	7.27E+02	1.09E+03	8.28E+02	8.32E+01	8.10E+02
	1000	7.21E+02	1.06E+03	**8.18E+02**	7.43E+01	8.13E+02
f_{30}	100	6.53E+02	4.32E+03	1.95E+03	9.95E+02	1.60E+03
	200	7.22E+02	4.11E+03	1.75E+03	7.17E+02	1.48E+03
	400	5.59E+02	4.44E+03	1.71E+03	8.12E+02	1.52E+03
	600	5.75E+02	4.51E+03	1.80E+03	7.82E+02	1.60E+03
	800	5.33E+02	3.05E+03	1.73E+03	5.91E+02	1.70E+03
	1000	7.93E+02	2.97E+03	**1.58E+03**	4.84E+02	1.45E+03

Table 17
Influence of the starting population size NP_0 on the quality of solutions by dimensionality of the problem $D = 50$ (1/3).

Func.	NP_0	Best	Worst	Mean	Std	Median
f_1	100	5.90E+04	3.60E+05	**1.80E+05**	7.73E+04	1.61E+05
	200	8.97E+04	6.13E+05	2.16E+05	1.03E+05	1.99E+05
	400	8.00E+04	6.62E+05	2.70E+05	1.26E+05	2.23E+05
	600	9.82E+04	6.87E+05	2.87E+05	1.27E+05	2.51E+05
	800	1.27E+05	8.03E+05	3.00E+05	1.20E+05	2.84E+05
	1000	1.75E+05	6.33E+05	3.44E+05	1.09E+05	3.19E+05
f_2	100	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	200	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	400	0.00E+00	0.00E+00	**0.00E+00**	0.00E+00	0.00E+00
	600	0.00E+00	1.85E−08	3.62E−10	2.56E−09	0.00E+00
	800	0.00E+00	1.61E−06	5.01E−08	2.25E−07	0.00E+00
	1000	0.00E+00	7.15E−06	3.37E−07	1.21E−06	2.40E−08
f_3	100	1.48E−02	3.33E−01	8.78E−02	5.64E−02	7.05E−02
	200	1.29E−02	1.78E−01	6.24E−02	3.53E−02	5.36E−02
	400	1.11E−02	1.75E−01	4.51E−02	2.94E−02	3.60E−02
	600	1.27E−02	8.70E−02	3.92E−02	1.85E−02	3.27E−02
	800	9.04E−03	1.39E−01	**2.98E−02**	2.10E−02	2.46E−02
	1000	7.52E−03	8.34E−02	3.47E−02	1.56E−02	3.36E−02
f_4	100	3.12E−06	1.71E+02	**2.73E+01**	4.54E+01	4.38E+00
	200	3.78E−04	1.15E+02	2.80E+01	3.58E+01	1.09E+01
	400	4.14E−01	1.54E+02	4.91E+01	4.32E+01	1.86E+01
	600	6.15E−02	1.55E+02	5.58E+01	4.77E+01	2.02E+01
	800	9.05E+00	1.56E+02	5.87E+01	4.95E+01	2.09E+01
	1000	8.70E−02	1.53E+02	6.56E+01	4.01E+01	8.79E+01
f_5	100	2.00E+01	2.00E+01	**2.00E+01**	8.66E−03	2.00E+01
	200	2.00E+01	2.01E+01	**2.00E+01**	1.94E−02	2.00E+01
	400	2.00E+01	2.03E+01	**2.00E+01**	4.31E−02	2.00E+01
	600	2.00E+01	2.04E+01	**2.00E+01**	6.20E−02	2.00E+01
	800	2.00E+01	2.03E+01	**2.00E+01**	5.66E−02	2.00E+01
	1000	2.00E+01	2.04E+01	**2.00E+01**	7.66E−02	2.00E+01
f_6	100	3.34E+00	1.88E+01	9.78E+00	3.08E+00	9.46E+00
	200	1.04E+00	2.07E+01	5.79E+00	3.12E+00	5.02E+00
	400	2.27E−01	1.60E+01	2.25E+00	2.25E+00	1.75E+00
	600	2.33E−01	4.65E+00	1.54E+00	9.07E−01	1.37E+00
	800	1.93E−01	3.67E+00	**8.88E−01**	7.91E−01	5.96E−01
	1000	2.38E−01	3.58E+00	1.07E+00	8.48E−01	7.10E−01
f_7	100	0.00E+00	9.86E−03	1.02E−03	2.81E−03	1.13E−07
	200	0.00E+00	9.86E−03	7.74E−04	2.37E−03	1.23E−07
	400	0.00E+00	9.86E−03	5.32E−04	2.14E−03	7.36E−08
	600	0.00E+00	3.25E−05	1.34E−06	4.61E−06	1.61E−07
	800	0.00E+00	6.19E−06	**6.00E−07**	1.17E−06	1.31E−07
	1000	0.00E+00	1.38E−05	1.30E−06	2.52E−06	4.13E−07
f_8	100	3.98E+00	1.69E+01	1.07E+01	3.47E+00	1.09E+01
	200	0.00E+00	9.95E+00	3.71E+00	1.93E+00	2.98E+00
	400	0.00E+00	2.98E+00	7.22E−01	9.87E−01	0.00E+00
	600	0.00E+00	9.95E−01	7.80E−02	2.67E−01	0.00E+00
	800	0.00E+00	9.95E−01	**1.95E−02**	1.38E−01	0.00E+00
	1000	0.00E+00	9.95E−01	**1.95E−02**	1.38E−01	0.00E+00
f_9	100	3.68E+01	9.95E+01	6.89E+01	1.37E+01	6.97E+01
	200	3.58E+01	7.56E+01	5.10E+01	9.06E+00	4.98E+01
	400	3.11E+01	7.76E+01	4.72E+01	1.12E+01	4.73E+01
	600	3.06E+01	6.85E+01	4.75E+01	8.25E+00	4.63E+01
	800	3.02E+01	6.37E+01	**4.64E+01**	8.94E+00	4.54E+01
	1000	3.18E+01	8.86E+01	4.89E+01	1.10E+01	4.81E+01
f_{10}	100	5.60E+00	4.90E+02	1.45E+02	1.28E+02	1.33E+02
	200	4.26E+00	2.41E+02	1.72E+01	3.57E+01	1.03E+01
	400	9.70E−01	1.16E+01	5.21E+00	2.08E+00	4.59E+00
	600	2.25E−01	8.30E+00	3.05E+00	1.64E+00	2.87E+00
	800	2.37E−01	6.59E+00	2.06E+00	1.26E+00	1.60E+00
	1000	1.87E−01	4.59E+00	**1.61E+00**	8.50E−01	1.57E+00

Table 18
Influence of the starting population size NP_0 on the quality of solutions by dimensionality of the problem $D = 50$ (2/3).

Func.	NP_0	Best	Worst	Mean	Std	Median
f_{11}	100	3.03E+03	5.99E+03	**4.49E+03**	6.01E+02	4.51E+03
	200	2.83E+03	6.06E+03	4.57E+03	6.33E+02	4.51E+03
	400	2.98E+03	6.15E+03	4.61E+03	5.95E+02	4.52E+03
	600	3.53E+03	6.01E+03	4.82E+03	6.42E+02	4.80E+03
	800	4.10E+03	6.10E+03	5.01E+03	5.06E+02	4.87E+03
	1000	4.06E+03	6.31E+03	5.11E+03	6.31E+02	5.11E+03
f_{12}	100	7.70E−02	4.07E−01	2.38E−01	6.77E−02	2.20E−01
	200	1.24E−01	3.85E−01	**2.27E−01**	6.64E−02	2.23E−01
	400	1.25E−01	4.15E−01	2.60E−01	7.16E−02	2.68E−01
	600	1.19E−01	5.22E−01	2.69E−01	9.46E−02	2.42E−01
	800	1.12E−01	5.46E−01	2.72E−01	9.39E−02	2.56E−01
	1000	1.11E−01	6.59E−01	2.58E−01	1.07E−01	2.42E−01
f_{13}	100	2.45E−01	4.73E−01	3.63E−01	5.48E−02	3.61E−01
	200	2.26E−01	4.57E−01	3.30E−01	5.39E−02	3.26E−01
	400	2.55E−01	4.15E−01	3.32E−01	4.15E−02	3.30E−01
	600	2.08E−01	3.99E−01	3.22E−01	4.22E−02	3.25E−01
	800	2.51E−01	3.97E−01	**3.20E−01**	3.63E−02	3.22E−01
	1000	2.41E−01	4.13E−01	3.21E−01	3.89E−02	3.22E−01
f_{14}	100	2.31E−01	6.66E−01	3.10E−01	6.39E−02	3.01E−01
	200	2.22E−01	6.74E−01	2.94E−01	5.97E−02	2.83E−01
	400	2.17E−01	3.52E−01	2.76E−01	2.47E−02	2.76E−01
	600	2.19E−01	3.32E−01	2.76E−01	2.64E−02	2.83E−01
	800	2.01E−01	3.34E−01	**2.74E−01**	2.21E−02	2.78E−01
	1000	2.12E−01	3.30E−01	2.78E−01	2.38E−02	2.77E−01
f_{15}	100	5.15E+00	1.64E+01	8.02E+00	1.98E+00	7.71E+00
	200	3.63E+00	8.26E+00	5.57E+00	9.84E−01	5.67E+00
	400	3.23E+00	7.44E+00	**5.14E+00**	9.01E−01	5.16E+00
	600	3.82E+00	7.88E+00	5.31E+00	8.52E−01	5.18E+00
	800	3.54E+00	7.97E+00	5.93E+00	9.91E−01	5.95E+00
	1000	4.12E+00	7.87E+00	6.13E+00	1.01E+00	6.17E+00
f_{16}	100	1.69E+01	1.93E+01	**1.80E+01**	5.04E−01	1.80E+01
	200	1.68E+01	1.93E+01	1.81E+01	4.95E−01	1.81E+01
	400	1.66E+01	1.97E+01	1.85E+01	6.28E−01	1.86E+01
	600	1.63E+01	1.98E+01	1.85E+01	6.89E−01	1.87E+01
	800	1.73E+01	2.00E+01	1.87E+01	6.02E−01	1.88E+01
	1000	1.70E+01	1.99E+01	1.88E+01	6.84E−01	1.88E+01
f_{17}	100	1.74E+03	2.47E+04	7.80E+03	4.18E+03	6.89E+03
	200	1.67E+03	1.02E+04	4.85E+03	2.22E+03	4.39E+03
	400	2.19E+03	7.74E+03	3.86E+03	1.27E+03	3.45E+03
	600	1.87E+03	7.37E+03	3.38E+03	1.09E+03	3.03E+03
	800	1.83E+03	5.42E+03	**3.18E+03**	7.77E+02	3.19E+03
	1000	1.82E+03	6.90E+03	3.09E+03	9.04E+02	2.88E+03
f_{18}	100	7.89E+01	2.75E+03	2.43E+02	3.58E+02	1.97E+02
	200	8.49E+01	2.74E+02	1.76E+02	4.40E+01	1.70E+02
	400	6.49E+01	2.03E+02	1.24E+02	2.82E+01	1.28E+02
	600	5.61E+01	1.97E+02	1.14E+02	2.60E+01	1.11E+02
	800	6.21E+01	1.67E+02	**1.10E+02**	2.27E+01	1.07E+02
	1000	6.68E+01	1.80E+02	1.13E+02	2.10E+01	1.13E+02
f_{19}	100	7.92E+00	7.65E+01	1.76E+01	9.80E+00	1.54E+01
	200	1.08E+01	7.95E+01	1.62E+01	9.84E+00	1.42E+01
	400	8.47E+00	2.01E+01	**1.31E+01**	1.68E+00	1.29E+01
	600	9.41E+00	1.80E+01	1.35E+01	1.56E+00	1.36E+01
	800	1.17E+01	1.70E+01	1.36E+01	1.44E+00	1.33E+01
	1000	1.10E+01	3.64E+01	1.44E+01	3.46E+00	1.37E+01
f_{20}	100	1.13E+02	4.90E+02	2.43E+02	8.03E+01	2.33E+02
	200	1.06E+02	3.60E+02	2.03E+02	5.09E+01	2.04E+02
	400	8.05E+01	2.38E+02	1.44E+02	3.63E+01	1.43E+02
	600	5.95E+01	2.19E+02	1.35E+02	3.58E+01	1.28E+02
	800	5.14E+01	1.79E+02	1.15E+02	2.59E+01	1.14E+02
	1000	4.43E+01	1.55E+02	**1.12E+02**	2.67E+01	1.10E+02

Table 19
Influence of the starting population size NP_0 on the quality of solutions by dimensionality of the problem $D = 50$ (3/3).

Func.	NP_0	Best	Worst	Mean	Std	Median
f_{21}	100	7.41E+02	1.14E+04	3.02E+03	1.75E+03	2.56E+03
	200	7.23E+02	4.05E+03	1.85E+03	6.39E+02	1.78E+03
	400	3.76E+02	2.97E+03	1.68E+03	3.98E+02	1.68E+03
	600	8.34E+02	2.35E+03	**1.51E+03**	3.72E+02	1.52E+03
	800	9.27E+02	3.09E+03	1.66E+03	3.99E+02	1.66E+03
	1000	9.13E+02	2.45E+03	1.58E+03	3.49E+02	1.56E+03
f_{22}	100	2.56E+02	7.80E+02	4.99E+02	1.27E+02	5.02E+02
	200	4.25E+01	6.90E+02	3.84E+02	1.38E+02	4.00E+02
	400	4.34E+01	4.96E+02	3.14E+02	9.23E+01	3.25E+02
	600	3.01E+01	4.85E+02	2.82E+02	1.10E+02	2.88E+02
	800	3.42E+01	4.63E+02	2.66E+02	1.03E+02	2.80E+02
	1000	4.21E+01	4.02E+02	**2.55E+02**	8.35E+01	2.71E+02
f_{23}	100	3.44E+02	3.44E+02	**3.44E+02**	8.53E−06	3.44E+02
	200	3.44E+02	3.44E+02	**3.44E+02**	8.53E−06	3.44E+02
	400	3.44E+02	3.44E+02	**3.44E+02**	9.34E−06	3.44E+02
	600	3.44E+02	3.44E+02	**3.44E+02**	8.53E−06	3.44E+02
	800	3.44E+02	3.44E+02	**3.44E+02**	0.00E+00	3.44E+02
	1000	3.44E+02	3.44E+02	**3.44E+02**	0.00E+00	3.44E+02
f_{24}	100	2.55E+02	2.78E+02	2.69E+02	5.61E+00	2.71E+02
	200	2.55E+02	2.75E+02	2.66E+02	6.02E+00	2.69E+02
	400	2.56E+02	2.73E+02	2.64E+02	5.55E+00	2.65E+02
	600	2.55E+02	2.72E+02	**2.63E+02**	6.13E+00	2.65E+02
	800	2.55E+02	2.73E+02	2.65E+02	6.07E+00	2.68E+02
	1000	2.55E+02	2.72E+02	2.64E+02	5.64E+00	2.66E+02
f_{25}	100	2.06E+02	2.23E+02	2.11E+02	3.30E+00	2.10E+02
	200	2.05E+02	2.22E+02	2.09E+02	3.22E+00	2.08E+02
	400	2.00E+02	2.22E+02	2.12E+02	6.51E+00	2.09E+02
	600	2.00E+02	2.22E+02	2.10E+02	7.75E+00	2.08E+02
	800	2.00E+02	2.22E+02	2.09E+02	9.11E+00	2.00E+02
	1000	2.00E+02	2.22E+02	**2.06E+02**	8.40E+00	2.00E+02
f_{26}	100	1.00E+02	1.01E+02	**1.00E+02**	6.63E−02	1.00E+02
	200	1.00E+02	1.00E+02	**1.00E+02**	4.72E−02	1.00E+02
	400	1.00E+02	1.00E+02	**1.00E+02**	4.53E−02	1.00E+02
	600	1.00E+02	1.00E+02	**1.00E+02**	5.06E−02	1.00E+02
	800	1.00E+02	1.00E+02	**1.00E+02**	4.76E−02	1.00E+02
	1000	1.00E+02	1.00E+02	**1.00E+02**	4.11E−02	1.00E+02
f_{27}	100	4.61E+02	8.96E+02	6.70E+02	9.52E+01	6.62E+02
	200	4.08E+02	6.60E+02	5.14E+02	6.23E+01	5.13E+02
	400	3.03E+02	5.02E+02	4.06E+02	5.71E+01	4.00E+02
	600	3.03E+02	5.07E+02	3.75E+02	5.68E+01	3.70E+02
	800	3.02E+02	5.46E+02	3.60E+02	5.22E+01	3.57E+02
	1000	3.02E+02	4.47E+02	**3.55E+02**	4.47E+01	3.62E+02
f_{28}	100	9.68E+02	2.19E+03	1.33E+03	2.41E+02	1.25E+03
	200	9.98E+02	1.74E+03	1.16E+03	1.14E+02	1.14E+03
	400	1.03E+03	1.22E+03	1.13E+03	3.80E+01	1.14E+03
	600	1.04E+03	1.21E+03	1.13E+03	4.44E+01	1.13E+03
	800	1.04E+03	1.22E+03	**1.12E+03**	4.40E+01	1.12E+03
	1000	1.01E+03	1.24E+03	1.13E+03	4.45E+01	1.13E+03
f_{29}	100	8.77E+02	4.23E+07	1.56E+06	7.71E+06	1.28E+03
	200	8.42E+02	1.98E+03	1.24E+03	2.85E+02	1.20E+03
	400	8.40E+02	1.55E+03	1.08E+03	1.72E+02	1.04E+03
	600	8.33E+02	1.53E+03	1.10E+03	1.91E+02	1.05E+03
	800	7.87E+02	1.56E+03	**1.07E+03**	1.63E+02	1.04E+03
	1000	9.04E+02	1.50E+03	1.09E+03	1.31E+02	1.08E+03
f_{30}	100	8.66E+03	1.88E+04	1.06E+04	1.74E+03	1.03E+04
	200	8.12E+03	1.61E+04	9.76E+03	1.14E+03	9.64E+03
	400	8.13E+03	1.09E+04	**9.24E+03**	6.07E+02	9.19E+03
	600	8.13E+03	1.11E+04	9.31E+03	6.00E+02	9.27E+03
	800	7.90E+03	1.14E+04	9.35E+03	8.17E+02	9.26E+03
	1000	7.97E+03	1.14E+04	9.25E+03	6.33E+02	9.17E+03

Table 20
Comparison of mean and std values for algorithms used in the study on 10D.

F	Meas.	CS	MOCS	CS-VSF	DE	SaDE	jDE	L-Shade	MVMO	HCS-SA
f_1	Mean	1.19E+07	3.09E+06	3.15E+05	**0.00E+00**	**0.00E+00**	1.64E−12	**0.00E+00**	4.95E−04	**0.00E+00**
	Std	1.07E+07	2.86E+06	1.87E+05	**0.00E+00**	**0.00E+00**	5.04E−12	**0.00E+00**	2.24E−04	**0.00E+00**
f_2	Mean	2.06E+07	7.90E+02	8.55E+03	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	6.18E−09	**0.00E+00**
	Std	4.51E+07	7.90E+02	1.07E+04	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	0.00E+00	**0.00E+00**
f_3	Mean	1.38E+04	4.70E+02	8.69E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	6.68E+03	8.22E+02	3.79E+02	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_4	Mean	6.83E+01	2.58E+01	**3.50E+00**	1.19E+01	2.60E+01	2.00E+01	2.94E+01	9.55E+00	1.68E+01
	Std	3.35E+01	2.01E+01	**4.66E+00**	1.55E+01	1.52E+01	1.71E+01	1.25E+01	0.00E+00	1.70E+01
f_5	Mean	2.00E+01	1.80E+01	2.01E+01	2.03E+01	1.98E+01	2.01E+01	**1.41E+01**	1.66E+01	1.77E+01
	Std	1.15E−01	5.05E+00	1.93E−01	1.20E−01	1.28E+00	2.72E−02	**8.68E+00**	2.00E+01	6.30E+00
f_6	Mean	8.13E+00	4.86E+00	1.36E+00	**0.00E+00**	2.31E+00	4.14E−01	1.75E−02	3.44E−03	7.04E−02
	Std	1.09E+00	2.18E+00	8.62E−01	**0.00E+00**	6.12E−01	6.36E−01	1.24E−01	1.84E−04	2.41E−01
f_7	Mean	6.57E+00	1.36E+00	3.37E−01	2.47E−01	2.49E−02	1.20E−02	3.04E−03	1.86E−02	**3.33E−06**
	Std	4.33E+00	1.79E+00	1.01E−01	1.39E−01	9.78E−03	1.04E−02	6.44E−03	1.72E−02	**1.62E−05**
f_8	Mean	3.60E+01	6.84E+00	9.24E+00	1.71E+01	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
	Std	1.25E+01	3.23E+00	2.10E+00	3.63E+00	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**	**0.00E+00**
f_9	Mean	4.34E+01	3.77E+01	1.82E+01	2.29E+01	5.04E+00	6.06E+00	**2.34E+00**	3.49E+00	2.53E+00
	Std	1.25E+01	6.27E+01	3.25E+00	3.75E+00	9.93E−01	1.56E+00	**8.32E−01**	3.98E+00	9.60E−01
f_{10}	Mean	8.84E+02	2.30E+02	1.90E+02	5.82E+02	3.45E+00	2.13E+00	**8.57E−03**	2.14E+00	8.08E−02
	Std	3.20E+02	2.60E+02	6.33E+01	1.60E+02	6.94E+00	7.11E+00	**2.15E−02**	3.75E−01	7.03E−02
f_{11}	Mean	1.03E+03	6.40E+02	7.61E+02	9.92E+02	2.95E+02	3.95E+02	**3.21E+01**	9.63E+01	6.31E+01
	Std	2.79E+02	3.47E+02	1.12E+02	1.68E+02	9.48E+01	1.62E+02	**3.79E+01**	1.22E+02	5.78E+01
f_{12}	Mean	4.05E−01	3.19E−01	4.68E−01	8.70E−01	3.90E−01	4.04E−01	6.82E−02	**4.22E−02**	7.61E−02
	Std	2.97E−01	2.01E−01	2.19E−01	1.32E−01	5.65E−02	9.56E−02	1.90E−02	**4.23E−02**	3.41E−02
f_{13}	Mean	4.83E−01	4.41E−01	2.07E−01	1.50E−01	1.29E−01	1.49E−01	5.16E−02	**3.55E−02**	7.08E−02
	Std	2.79E−01	1.65E−01	3.67E−02	2.49E−02	2.36E−02	2.37E−02	1.49E−02	**3.43E−02**	1.59E−02
f_{14}	Mean	6.82E−01	6.77E−01	1.76E−01	1.57E−01	1.17E−01	1.60E−01	**8.14E−02**	8.91E−02	1.02E−01
	Std	9.03E−01	7.88E−01	3.65E−02	2.83E−02	3.65E−02	3.84E−02	**2.53E−02**	8.21E−02	3.38E−02
f_{15}	Mean	2.10E+01	3.87E+00	2.22E+00	2.01E+00	7.69E−01	9.67E−01	**3.66E−01**	4.35E−01	4.09E−01
	Std	1.08E+01	1.49E+00	3.84E−01	3.27E−01	1.12E−01	1.80E−01	**6.85E−02**	4.07E−01	1.08E−01
f_{16}	Mean	3.53E+00	5.60E+03	2.81E+00	2.66E+00	2.22E+00	2.38E+00	**1.24E+00**	1.45E+00	1.46E+00
	Std	3.26E−01	2.31E+04	1.81E−01	1.97E−01	2.45E−01	2.01E−01	**3.00E−01**	1.46E+00	3.76E−01
f_{17}	Mean	4.85E+04	2.94E+04	3.59E+03	**5.08E−01**	4.81E+01	1.52E+01	9.77E−01	9.36E+00	2.80E+01
	Std	6.24E+04	4.03E+04	2.77E+03	**6.75E−01**	4.02E+01	1.16E+01	1.07E+00	1.12E+01	3.98E+01
f_{18}	Mean	1.10E+04	5.02E+03	6.53E+02	**1.45E−01**	1.43E+00	1.35E+00	2.44E−01	7.83E−01	1.45E−01
	Std	1.15E+04	3.57E+03	6.68E+02	**2.50E−01**	5.56E−01	7.82E−01	3.11E−01	1.01E+00	1.16E−01
f_{19}	Mean	5.93E+00	2.02E+02	1.54E+00	3.86E−01	3.82E−01	5.37E−01	**7.73E−02**	1.58E−01	1.78E−01
	Std	1.77E+00	6.36E+02	3.26E−01	1.11E−01	1.54E−01	1.85E−01	**6.34E−02**	1.06E−01	1.79E−01
f_{20}	Mean	4.35E+03	2.57E+03	3.63E+01	**6.95E−02**	4.43E−01	1.84E−01	1.85E−01	3.13E−01	2.07E−01
	Std	5.72E+03	2.95E+03	1.69E+01	**1.53E−01**	1.58E−01	1.05E−01	1.78E−01	1.54E−01	1.19E−01
f_{21}	Mean	7.92E+03	2.79E+03	4.86E+02	2.62E−01	6.17E+00	5.09E−01	4.08E−01	1.93E+00	**2.00E−01**
	Std	9.07E+03	2.87E+03	2.57E+02	2.70E−01	5.07E+00	3.57E−01	3.06E−01	3.31E−01	**2.40E−01**
f_{22}	Mean	6.98E+01	1.67E+02	1.52E+01	2.39E−01	4.09E+00	1.12E+00	**4.41E−02**	2.63E−01	1.82E−01
	Std	5.77E+01	7.28E+01	6.29E+00	2.05E−01	1.55E+00	1.47E+00	**2.79E−02**	2.04E−01	1.00E−01
f_{23}	Mean	3.43E+02	**3.06E+02**	3.25E+02	3.29E+02	3.29E+02	3.29E+02	3.29E+02	3.29E+02	3.29E+02
	Std	9.62E+00	**5.87E+01**	2.87E+01	0.00E+00	0.00E+00	2.32E−13	8.53E−06	3.29E+02	1.32E−05
f_{24}	Mean	1.57E+02	1.67E+02	1.25E+02	1.28E+02	1.12E+02	1.13E+02	**1.07E+02**	1.09E+02	1.08E+02
	Std	1.53E+01	2.69E+01	3.98E+00	3.80E+00	1.82E+00	2.45E+00	**2.26E+00**	1.10E+02	2.09E+00
f_{25}	Mean	1.84E+02	1.86E+02	1.42E+02	1.39E+02	1.61E+02	1.68E+02	1.33E+02	**1.16E+02**	1.18E+02
	Std	2.24E+01	3.41E+01	8.08E+00	3.24E+01	3.96E+01	3.64E+01	4.00E+01	**1.17E+02**	1.39E+01
f_{26}	Mean	1.01E+02	1.29E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	1.00E+02	**1.00E+02**	1.00E+02
	Std	2.45E−01	1.02E+02	3.29E−02	2.52E−02	2.42E−02	3.10E−02	1.61E−02	**1.00E+02**	1.69E−02
f_{27}	Mean	2.81E+01	4.37E+02	**4.04E+00**	7.43E+01	1.63E+02	1.63E+02	5.81E+01	1.72E+01	1.56E+01
	Std	5.19E+01	2.06E+02	**8.34E−01**	1.39E+02	1.82E+02	1.74E+02	1.33E+02	1.60E+00	6.83E+01
f_{28}	Mean	7.14E+02	6.84E+02	3.84E+02	3.80E+02	4.02E+02	3.78E+02	3.81E+02	**3.61E+02**	4.02E+02
	Std	1.26E+02	2.00E+02	1.84E+01	3.94E+01	4.87E+01	2.20E+01	3.14E+01	**3.57E+02**	5.36E+01
f_{29}	Mean	1.57E+03	8.04E+04	2.09E+02	2.15E+02	2.23E+02	2.23E+02	2.22E+02	**1.81E+02**	2.19E+02
	Std	2.24E+03	3.95E+05	1.68E+00	2.42E+01	1.96E+00	6.72E−01	4.58E−01	**1.60E+02**	1.77E+01
f_{30}	Mean	1.88E+03	1.72E+03	**2.62E+02**	4.64E+02	4.80E+02	4.72E+02	4.65E+02	4.92E+02	4.80E+02
	Std	6.91E+02	3.47E+02	**1.54E+01**	7.69E+00	3.26E+01	1.49E+01	1.32E+01	4.87E+02	2.95E+01

Table 21
Comparison of mean and std values for algorithms used in the study on 50D.

F	Meas.	CS	MOCS	CS-VSF	DE	SaDE	jDE	L-Shade	MVMO	HCS-SA
f_1	Mean	1.94E+07	2.98E+06	2.52E+08	1.30E+06	1.21E+05	4.76E+05	1.24E+03	**7.54E−03**	1.74E+05
	Std	1.04E+07	1.42E+06	4.41E+07	4.59E+05	6.54E+04	1.85E+05	1.50E+03	**7.39E−03**	7.06E+04
f_2	Mean	3.64E+04	5.49E+03	7.61E+09	2.54E+00	**0.00E+00**	6.10E−10	**0.00E+00**	1.52E−05	**0.00E+00**
	Std	1.26E+04	3.70E+03	1.04E+09	8.83E+00	**0.00E+00**	9.02E−10	**0.00E+00**	1.30E−05	**0.00E+00**
f_3	Mean	4.06E+04	1.72E+03	1.34E+05	4.01E−01	2.07E+01	2.50E−09	**0.00E+00**	4.54E−03	**0.00E+00**
	Std	1.13E+04	1.06E+03	1.21E+04	1.52E+00	5.73E+01	8.54E−09	**0.00E+00**	4.75E−03	**0.00E+00**
f_4	Mean	2.66E+02	1.20E+02	1.27E+03	5.42E+01	2.67E+01	9.23E+01	5.89E+01	**5.77E+00**	2.78E+01
	Std	7.18E+01	5.05E+01	1.56E+02	3.51E+01	4.33E+01	1.68E+01	4.52E+01	**0.00E+00**	3.14E+01
f_5	Mean	2.00E+01	2.40E+01	2.11E+01	2.11E+01	2.04E+01	2.05E+01	2.02E+01	**2.00E+01**	2.00E+01
	Std	1.44E−03	1.00E+01	5.39E−02	3.44E−02	2.74E−02	3.48E−02	4.55E−02	**2.00E+01**	1.70E−03
f_6	Mean	5.92E+01	4.28E+01	5.93E+01	6.53E−01	3.08E+01	1.69E+01	**2.64E−01**	1.05E+01	2.38E+00
	Std	3.97E+00	1.74E+01	2.68E+00	9.46E−01	1.95E+00	6.58E+00	**5.18E−01**	9.67E+00	1.25E+00
f_7	Mean	6.87E−02	8.25E+00	7.58E+01	2.41E−04	8.70E−04	3.94E−04	**0.00E+00**	6.67E−03	5.32E−04
	Std	3.76E−02	2.08E+01	8.69E+00	1.71E−03	2.74E−03	1.97E−03	**0.00E+00**	7.40E−03	2.15E−03
f_8	Mean	2.66E+02	8.42E+01	3.82E+02	2.04E+02	**0.00E+00**	3.98E−02	2.58E−09	2.28E+00	**0.00E+00**
	Std	4.37E+01	5.62E+01	1.67E+01	3.87E+01	**0.00E+00**	1.99E−01	7.41E−09	1.99E+00	**0.00E+00**
f_9	Mean	5.88E+02	3.13E+02	4.48E+02	3.54E+02	8.35E+01	9.67E+01	**1.14E+01**	7.69E+01	4.33E+01
	Std	7.67E+01	2.15E+02	1.49E+01	1.37E+01	1.57E+01	1.20E+01	**2.11E+00**	8.06E+01	5.91E+00
f_{10}	Mean	4.05E+03	1.53E+03	1.04E+04	9.65E+03	5.94E−01	2.97E+00	**1.22E−01**	8.08E+01	1.42E+00
	Std	7.23E+02	2.11E+03	3.84E+02	1.62E+03	1.20E+00	1.95E+00	**4.09E−02**	1.75E+01	1.11E+00
f_{11}	Mean	7.31E+03	5.23E+03	1.25E+04	1.29E+04	5.23E+03	5.89E+03	**3.22E+03**	3.57E+03	4.29E+03
	Std	8.47E+02	2.23E+03	3.07E+02	4.29E+02	2.84E+02	4.48E+02	**3.27E+02**	3.66E+03	4.44E+02
f_{12}	Mean	6.45E−01	8.40E−01	1.67E−01	3.30E+00	4.80E−01	5.56E−01	2.19E−01	**4.93E−02**	1.61E−01
	Std	2.23E−01	2.72E−01	3.59E−01	2.79E−01	4.37E−02	5.59E−02	2.79E−02	**3.28E−02**	4.78E−02
f_{13}	Mean	5.07E−01	5.10E−01	7.84E−01	4.63E−01	4.02E−01	3.91E−01	**1.60E−01**	2.84E−01	2.55E−01
	Std	8.43E−02	1.03E−01	8.64E−02	4.52E−02	5.43E−02	3.13E−02	**1.81E−02**	2.79E−01	3.69E−02
f_{14}	Mean	3.46E−01	1.62E+01	2.02E+01	3.21E−01	2.88E−01	3.89E−01	2.97E−01	**2.38E−01**	2.72E−01
	Std	5.53E−02	4.05E+01	2.93E+00	6.33E−02	1.04E−01	1.19E−01	2.45E−02	**2.35E−01**	2.50E−02
f_{15}	Mean	1.85E+02	1.05E+02	4.78E+03	3.12E+01	1.14E+01	1.19E+01	5.15E+00	**5.00E+00**	5.97E+00
	Std	4.66E+01	4.00E+01	2.08E+03	1.03E+00	1.41E+00	1.01E+00	5.03E−01	**5.00E+00**	1.10E+00
f_{16}	Mean	2.20E+01	6.38E+04	2.23E+01	2.21E+01	1.84E+01	1.89E+01	**1.69E+01**	1.89E+01	1.82E+01
	Std	6.34E−01	1.66E+05	2.45E−01	2.91E−01	3.35E−01	2.42E−01	**4.76E−01**	1.89E+01	6.22E−01
f_{17}	Mean	1.75E+06	2.70E+05	9.48E+06	1.36E+04	1.33E+04	2.34E+04	1.40E+03	**6.65E+02**	2.86E+03
	Std	1.10E+06	1.78E+05	2.64E+06	1.04E+04	1.24E+04	1.21E+04	5.08E+02	**6.91E+02**	7.17E+02
f_{18}	Mean	2.41E+03	1.99E+03	8.72E+03	1.34E+02	1.92E+02	1.48E+03	9.73E+01	**3.40E+01**	1.07E+02
	Std	1.68E+03	1.29E+03	3.53E+03	1.03E+01	6.99E+01	1.63E+03	1.37E+01	**2.92E+01**	2.08E+01
f_{19}	Mean	6.71E+01	1.94E+02	7.43E+01	1.17E+01	1.27E+01	1.32E+01	8.30E+00	**6.76E+00**	1.14E+01
	Std	2.60E+01	3.94E+02	1.56E+01	7.61E−01	1.75E+00	4.45E+00	1.80E+00	**6.43E+00**	2.89E+00
f_{20}	Mean	6.46E+04	3.42E+04	3.65E+04	9.75E+01	6.83E+01	5.07E+01	**1.39E+01**	2.29E+01	1.03E+02
	Std	2.68E+04	9.47E+04	9.68E+03	1.20E+01	2.07E+01	2.06E+01	**4.52E+00**	2.11E+01	2.17E+01
f_{21}	Mean	1.15E+06	1.93E+05	3.22E+06	2.73E+03	1.86E+03	1.11E+04	**5.15E+02**	5.39E+02	1.41E+03
	Std	9.20E+05	1.16E+05	9.34E+05	1.31E+03	1.44E+03	1.04E+04	**1.48E+02**	5.00E+02	3.29E+02
f_{22}	Mean	1.95E+03	1.50E+03	1.33E+03	7.01E+02	7.32E+02	6.01E+02	**1.14E+02**	4.58E+02	2.57E+02
	Std	4.13E+02	5.64E+02	1.28E+02	3.77E+02	1.54E+02	1.48E+02	**7.43E+01**	4.92E+02	1.11E+02
f_{23}	Mean	3.52E+02	**3.32E+02**	3.72E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02	3.44E+02
	Std	5.49E+00	**2.89E+01**	3.87E+00	1.08E−05	0.00E+00	2.38E−13	1.57E−05	3.44E+02	8.53E−06
f_{24}	Mean	3.27E+02	**2.56E+02**	3.34E+02	2.70E+02	2.70E+02	2.69E+02	2.75E+02	2.58E+02	2.69E+02
	Std	5.27E+01	**1.44E+01**	4.93E+00	3.11E+00	2.80E+00	2.03E+00	6.55E−01	2.57E+02	4.50E+00
f_{25}	Mean	2.56E+02	2.19E+02	2.60E+02	2.05E+02	2.07E+02	2.07E+02	**2.05E+02**	2.06E+02	2.11E+02
	Std	2.69E+01	1.16E+01	5.76E+00	3.22E−01	1.76E+00	1.85E+00	**3.61E−01**	2.06E+02	8.19E+00
f_{26}	Mean	1.01E+02	4.09E+02	1.01E+02	1.04E+02	1.23E+02	1.18E+02	1.02E+02	1.04E+02	**1.00E+02**
	Std	1.09E+00	5.23E+02	1.50E−01	1.93E+01	7.09E+01	5.96E+01	1.38E+01	1.00E+02	**5.94E−02**
f_{27}	Mean	2.04E+03	2.47E+03	2.09E+03	3.63E+02	9.58E+02	5.71E+02	**3.33E+02**	5.11E+02	3.96E+02
	Std	1.24E+02	1.88E+03	5.99E+01	4.21E+01	1.94E+02	8.65E+01	**3.00E+01**	5.05E+02	5.68E+01
f_{28}	Mean	7.19E+03	6.81E+03	1.08E+03	**1.07E+03**	1.23E+03	1.13E+03	1.11E+03	1.33E+03	1.16E+03
	Std	9.88E+02	1.17E+03	1.50E+02	**4.39E+01**	1.56E+02	3.03E+01	2.88E+01	1.35E+03	5.29E+01
f_{29}	Mean	1.11E+06	6.32E+03	**3.04E+02**	1.49E+06	9.16E+06	1.13E+07	7.95E+02	1.11E+03	7.06E+05
	Std	4.01E+06	3.66E+03	**1.38E+01**	7.36E+06	1.58E+07	1.69E+07	2.38E+01	1.12E+03	4.99E+06
f_{30}	Mean	3.51E+04	1.45E+04	**1.66E+03**	8.44E+03	9.98E+03	8.76E+03	8.66E+03	9.42E+03	9.09E+03
	Std	1.18E+04	1.76E+03	**1.74E+02**	4.02E+02	9.96E+02	5.68E+02	4.09E+02	9.43E+03	8.34E+02

References

[1] A.E. Eiben, J.E. Smith, Introduction to Evolutionary Computing, Springer-Verlag, Berlin, Heidelberg, 2003.
[2] C. Blum, D. Merkle (Eds.), Swarm Intelligence: Introduction and Applications, Natural Computing Series, Springer, Berlin, Heidelberg, 2008. http://dx.doi.org/10.1007/978-3-540-74089-6.
[3] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, 1st edition, Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989.
[4] J. Koza, Genetic Programming 2—Automatic Discovery of Reusable Programs, MIT Press, Cambridge, USA, 1994.
[5] T. Bäck, Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms, Oxford University Press, Oxford, UK, 1996.
[6] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.
[7] A. Zamuda, J. Brest, Self-adaptive control parameters' randomization frequency and propagations in differential evolution, Swarm Evol. Comput. 25 (2015) 72–99, http://dx.doi.org/10.1016/j.swevo.2015.10.007.
[8] L. Fogel, A. Owens, M. Walsh, Artificial Intelligence Through Simulated Evolution, John Wiley & Sons, New York, 1966 ⟨https://books.google.si/books?id=QMLaAAAAMAAJ⟩.
[9] G. Beni, J. Wang, Swarm intelligence in cellular robotic systems, in: P. Dario, G. Sandini, P. Aebischer (Eds.), Robots and Biological Systems: Towards a New Bionics?, NATO ASI Series, vol. 102, Springer, Berlin, Heidelberg, 1993, pp. 703–712. http://dx.doi.org/10.1007/978-3-642-58069-7_38.
[10] M. Dorigo, G. Di Caro, The ant colony optimization meta-heuristic, in: C. David, M. Dorigo, F. Glover (Eds.), New Ideas in Optimization, McGraw Hill, London, UK, 1999, pp. 11–32.
[11] J. Kennedy, R. Eberhart, Particle swarm optimization, in: Proceedings of IEEE International Conference on Neural Networks, vol. 4, 1995, pp. 1942–1948. http://dx.doi.org/10.1109/ICNN.1995.488968.
[12] D. Karaboga, B. Basturk, A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm, J. Glob. Optim. 39 (3) (2007) 459–471, http://dx.doi.org/10.1007/s10898-007-9149-x.
[13] D. Dasgupta, Information processing in the immune system, in: C. David, M. Dorigo, F. Glover (Eds.), New Ideas in Optimization, McGraw Hill, London, UK, 1999, pp. 161–167.
[14] I. Fister, I. Fister Jr., X.-S. Yang, J. Brest, A comprehensive review of firefly algorithms, Swarm Evol. Comput. 13 (2013) 34–46, http://dx.doi.org/10.1016/j.swevo.2013.06.001 ⟨http://www.sciencedirect.com/science/article/pii/S2210650213000461⟩.
[15] M.-Y. Cheng, D. Prayogo, Symbiotic organisms search: a new metaheuristic optimization algorithm, Comput. Struct. 139 (2014) 98–112, http://dx.doi.org/10.1016/j.compstruc.2014.03.007 ⟨http://www.sciencedirect.com/science/article/pii/S0045794914000881⟩.
[16] X.-S. Yang, S. Deb, Cuckoo search via lévy flights, in: World Congress on Nature & Biologically Inspired Computing, 2009, NaBIC 2009, Coimbatore, India, IEEE, 2009, pp. 210–214.
[17] X.-S. Yang, S. Deb, Multiobjective cuckoo search for design optimization, Comput. Oper. Res. 40 (6) (2013) 1616–1624.
[18] R. Soto, B. Crawford, R. Olivares, J. Barraza, F. Johnson, F. Paredes, A binary cuckoo search algorithm for solving the set covering problem, in: Bioinspired Computation in Artificial Systems, IWINAC 2015, Elche, Spain, Springer International Publishing, 2015, pp. 88–97.
[19] A. Gherboudj, A. Layeb, S. Chikhi, Solving 0–1 knapsack problems by a discrete binary version of cuckoo search algorithm, Int. J. Bio-Inspired Comput. 4 (4) (2012) 229–236.
[20] D. Rodrigues, L.A. Pereira, T. Almeida, J.P. Papa, A. Souza, C.C. Ramos, X.-S. Yang, BCS: a binary cuckoo search algorithm for feature selection, in: 2013 IEEE International Symposium on Circuits and Systems (ISCAS), Beijing, China, IEEE, 2013, pp. 465–468.
[21] W. Long, X. Liang, Y. Huang, Y. Chen, An effective hybrid cuckoo search algorithm for constrained global optimization, Neural Comput. Appl. 25 (3–4) (2014) 911–926.
[22] G. Kanagaraj, S. Ponnambalam, N. Jawahar, A hybrid cuckoo search and genetic algorithm for reliability-redundancy allocation problems, Comput. Ind. Eng. 66 (4) (2013) 1115–1124.
[23] K. Perumal, J. M. Ungati, G. Kumar, N. Jain, R. Gaurav, P.R. Srivastava, Test data generation: a hybrid approach using cuckoo and tabu search, in: Swarm, Evolutionary, and Memetic Computing, Visakhapatnam, India, Springer, Berlin, Heidelberg, 2011, pp. 46–54.
[24] P. Ong, Adaptive cuckoo search algorithm for unconstrained optimization, Sci. World J. (2014), http://dx.doi.org/10.1155/2014/943403.
[25] X. Li, M. Yin, Modified cuckoo search algorithm with self adaptive parameter method, Inf. Sci. 298 (2015) 80–97.
[26] Y. Zhang, L. Wang, Q. Wu, Modified adaptive cuckoo search (MACS) algorithm and formal description for global optimisation, Int. J. Comput. Appl. Technol. 44 (2) (2012) 73–79.
[27] R. Gomathi, D. Sharmila, A novel adaptive cuckoo search for optimal query plan generation, Sci. World J. (2014).
[28] A.H. Gandomi, S. Talatahari, X.-S. Yang, S. Deb, Design optimization of truss structures using cuckoo search algorithm, Struct. Des. Tall Spec. Build. 22 (17) (2013) 1330–1349.
[29] S.-E.K. Fateen, A. Bonilla-Petriciolet, Unconstrained Gibbs free energy minimization for phase equilibrium calculations in nonreactive systems, using an improved cuckoo search algorithm, Ind. Eng. Chem. Res. 53 (26) (2014) 10826–10834.
[30] S.-E.K. Fateen, A. Bonilla-Petriciolet, On the effectiveness of nature-inspired metaheuristic algorithms for performing phase equilibrium thermodynamic calculations, Sci. World J. (2014).
[31] R.A. Vázquez, Training spiking neural models using cuckoo search algorithm, in: 2011 IEEE Congress on Evolutionary Computation (CEC), New Orleans, LA, USA, IEEE, 2011, pp. 679–686.
[32] A. Gálvez, A. Iglesias, L. Cabellos, Cuckoo search with Lévy flights for weighted Bayesian energy functional optimization in global-support curve data fitting, Sci. World J. 2014 (2014) 1–11, http://dx.doi.org/10.1155/2014/138760.
[33] D.H. Wolpert, W.G. Macready, No free lunch theorems for optimization, Trans. Evol. Comput. 1 (1) (1997) 67–82, http://dx.doi.org/10.1109/4235.585893.
[34] I. Fister, D. Strnad, X.-S. Yang, I. Fister Jr., Adaptation and hybridization in nature-inspired algorithms, in: Adaptation and Hybridization in Computational Intelligence, Springer Publishing Company, Incorporated, Berlin, 2015, pp. 3–50.
[35] I. Fister, I. Fister Jr., J. Brest, V. Žumer, Memetic artificial bee colony algorithm for large-scale global optimization, in: 2012 IEEE Congress on Evolutionary Computation (CEC), Brisbane, Australia, IEEE, 2012, pp. 1–8.
[36] M. Črepinšek, S.-H. Liu, M. Mernik, Exploration and exploitation in evolutionary algorithms: a survey, ACM Comput. Surv. 45 (3) (2013) 35:1–35:33, http://dx.doi.org/10.1145/2480741.2480752.
[37] J. Brest, M.S. Maucec, Population size reduction for the differential evolution algorithm, Appl. Intell. 29 (2008) 228–247, http://dx.doi.org/10.1007/s10489-007-0091-x.
[38] L. Wang, Y. Yin, Y. Zhong, Cuckoo search with varied scaling factor, Front. Comput. Sci. 9 (4) (2015) 623–635, http://dx.doi.org/10.1007/s11704-015-4178-y.
[39] S. Walton, O. Hassan, K. Morgan, M. Brown, Modified cuckoo search: a new gradient free optimisation algorithm, Chaos Solitons Fractals 44 (9) (2011) 710–718.
[40] J. Brest, M.S. Maučec, Self-adaptive differential evolution algorithm using population size reduction and three strategies, Soft Comput.—A Fusion of Foundations, Methodologies and Applications 15 (2011) 2157–2174, http://dx.doi.org/10.1007/s00500-010-0644-5.
[41] A.K. Qin, V.L. Huang, P.N. Suganthan, Differential evolution algorithm with strategy adaptation for global numerical optimization, IEEE Trans. Evol. Comput. 13 (2) (2009) 398–417, http://dx.doi.org/10.1109/TEVC.2008.927706.
[42] R. Tanabe, A.S. Fukunaga, Improving the search performance of shade using linear population size reduction, in: 2014 IEEE Congress on Evolutionary Computation (CEC), Beijing, China, IEEE, 2014, pp. 1658–1665.
[43] I. Erlich, J. Rueda, S. Wildenhues, F. Shewarega, Evaluating the mean-variance mapping optimization on the IEEE-CEC 2014 test suite, in: IEEE Congress on Evolutionary Computation (CEC), 2014, pp. 1625–1632. http://dx.doi.org/10.1109/CEC.2014.6900516.
[44] I. Fister Jr., X.-S. Yang, I. Fister, J. Brest, D. Fister, A brief review of nature-inspired algorithms for optimization, Electrotech. Rev. 80 (3) (2013) 116–122.
[45] R.B. Payne, The Cuckoos, Oxford University Press, New York, NY, USA, 2005.
[46] S.S. Rao, Engineering Optimization: Theory and Practice, John Wiley & Sons, NJ, USA, 2009.
[47] J. Brest, S. Greiner, B. Bošković, M. Mernik, V. Žumer, Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems, IEEE Trans. Evol. Comput. 10 (6) (2006) 646–657.
[48] S. Walton, Modified Cuckoo Search (MCS) Open Source Gradient Free Optimiser ⟨https://code.google.com/p/modified-cs⟩, 2015.
[49] J. Liang, B. Qu, P. Suganthan, Problem Definitions and Evaluation Criteria for the CEC 2014 Special Session and Competition on Single Objective Real-Parameter Numerical Optimization, Computational Intelligence Laboratory, Zhengzhou University, Zhengzhou China and Technical Report, Nanyang Technological University, Singapore.
[50] M. Friedman, A comparison of alternative tests of significance for the problem of m rankings, Ann. Math. Stat. 11 (1940) 86–92, http://dx.doi.org/10.1214/aoms/1177731944.
[51] J. Derrac, S. García, D. Molina, F. Herrera, A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms, Swarm Evol. Comput. 1 (1) (2011) 3–18, http://dx.doi.org/10.1016/j.swevo.2011.02.002 ⟨http://www.sciencedirect.com/science/article/pii/S2210650211000034⟩.
[52] N. Veček, M. Mernik, M. Črepinšek, A chess rating system for evolutionary algorithms: a new method for the comparison and ranking of evolutionary algorithms, Inf. Sci. 277 (2014) 656–679, http://dx.doi.org/10.1016/j.ins.2014.02.154 ⟨http://www.sciencedirect.com/science/article/pii/S002002551400276X⟩.
[53] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[54] P.B. Nemenyi, Distribution-free multiple comparisons (Ph.D. thesis), Princeton University, 1963.
[55] S. García, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, J. Mach. Learn. Res. (2008) 2677–2694.
[56] A. Benavoli, G. Corani, F. Mangili, Should we really use post-hoc tests based on mean-ranks? J. Mach. Learn. Res. 17 (5) (2016) 1-10.

[57] M. Mernik, S. Liu, D. Karaboga, M. Črepinšek, On clarifying misconceptions when comparing variants of the artificial bee colony algorithm by offering a new implementation, Inf. Sci. 291 (2015) 115–127, http://dx.doi.org/10.1016/j.ins.2014.08.040.
[58] A. Kaveh, S. Talatahari, An improved ant colony optimization for constrained engineering design problems, Eng. Comput. 27 (1) (2010) 155–182.
[59] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, Eng. Appl. Artif. Intell. 20 (1) (2007) 89–99.
[60] C. Coello, N. Cortés, Hybridizing a genetic algorithm with an artificial immune system for global optimization, Eng. Optim. 36 (5) (2004) 607–634, http://dx.doi.org/10.1080/03052150410001704845.
[61] F. Zhuo Huang, L. Wang, Q. He, An effective co-evolutionary differential evolution for constrained optimization, Appl. Math. Comput. 186 (1) (2007) 340–356, http://dx.doi.org/10.1016/j.amc.2006.07.105 ⟨http://www.sciencedirect.com/science/article/pii/S0096300306009441⟩.
[62] T. Ray, P. Saini, Engineering design optimization using a swarm with an intelligent information sharing among individuals, Eng. Optim. 33 (6) (2001) 735–748.
[63] S. Akhtar, K. Tai, T. Ray, A socio-behavioural simulation model for engineering design optimization, Eng. Optim. 34 (4) (2002) 341–354.
[64] K.J. Ku, S. Rao, L. Chen, Taguchi-aided search method for design optimization of engineering systems, Eng. Optim. 30 (1) (1998) 1–23.
[65] E. Mezura Montes, C.A. Coello Coello, R. Landa-Becerra, Engineering optimization using simple evolutionary algorithm, in: Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, 2003, pp. 149–156. http://dx.doi.org/10.1109/TAI.2003.1250183.