

Shake – Regicide: A new heuristic for the diversity control of evolutionary algorithms

Justino Ramirez-Ortegon, Mariano Rivera and Arturo Hernandez
Centro de Investigacion en Matematicas A.C.
Guanajuato, Gto. Mexico 36000

Abstract. Evolutionary algorithms have been very successful at solving global optimization problems. Two competing goals govern the performance of evolutionary algorithms: exploration and exploitation. This paper proposes a new heuristic to keep population diversity: the shake and the regicide. The shake heuristic improves the exploration by perturbing the whole population. The regicide heuristic (kill the leader) reduces the risk of being, early, trapped by a local minimum. Experiments demonstrate that the Shake-Regicide heuristic improves significantly the precision of the results (in about 3 orders of magnitude) of standard differential evolution, genetic algorithm and evolution strategy.

1 Introduction

The problem of computing the global optimum arises in almost every field of business, engineering, and sciences. For example, engineers need to optimize the configuration parameters of a production process. An enormous effort has been applied to solve global continuous problems [1]. We consider the following unconstrained global continuous optimization problem:

$$\min_{x \in \mathfrak{R}^n} f(x), \quad (1)$$

where $f: \mathfrak{R}^n$ to \mathfrak{R} is a real-valued function defined on $x \in \mathfrak{R}^n$, and it can be non differentiable function. A significant problem faced during optimization occurs when f is non-convex, perhaps with many local minima. The problem becomes a major challenge when local minima are present. Evolutionary algorithms (EA) [2, 3] have been used to solve global optimization problems, and proved very robust. For example: Genetic Algorithms (GA) [4], Evolution Strategy (ES) [5] and Differential Evolution (DE) [6]. EA analogized the species evolution process manifested in the biological adaptation to changing environments; such process is emulated by EAs for finding the optimum through the evolution of a synthetic population of candidate solutions.

Two competing goals govern the design of EA [7]: the exploration and the exploitation. Exploration is needed to ensure that the entire search space is thoroughly examined resulting in a reliable global optimum estimation. On the other hand, exploitation is important since the refinement of the current solution will often produce a better solution. Different mechanisms to improve exploration have recently been discussed in the literature. Mutation operators based on physic phenomenon, such

like electromagnetic theory [8], and statistical properties of the population [9, 10] have proved some success. Hybrid algorithms that combine DE and GA with an Estimation Distribution Algorithm (EDA) [11, 12] to promote exploration are used successfully in low dimensional space (dimension 5 and 10). This paper, however, proposes a new heuristic that promotes exploration by spawning new individuals in promissory neighborhoods of the search space, named here Shake–Regicide Heuristic (SRH). The goal of this paper is to introduce the new heuristic, and to assess the performance of three EAs modified with SRH. The dimensionality of the problems tested is 30, 50 and 100. The SRH promotes the diversity (Shake) and, at the same time, eliminates hegemonic individuals (Regicide). SRH is inspired by the behavior of social individuals [13] and economy laws [14, 15].

Standard genetic or evolution inspired algorithms promote the reproduction and replication of individuals with better fitness value with respect to the average fitness of the population (*hegemonic individuals*). Even at initial generations, when the search space is not well explored yet, the selection operator favors the fittest individuals. In the absence of additional diversity to balance the selection pressure, the search suffers of premature convergence, generally ending at a local minima. The regicide strategy removes hegemonic individuals from the population; they can only reproduce or inherit their genetic material during a few generations. This reasoning is based on the guess that at initial stages the population has low probability of being near the global optimum when high exploration is recommendable.

Shake strategy consists of a hybrid technique inspired by the Differential Evolution algorithm and a procedure for estimating the population distribution. The Shake heuristic generates new random individuals in a neighborhood of the population. The new individuals can explore neighborhoods of the space which the reproduction or mutation operators have difficulties to identify. A neighborhood is identified by statistical means (explained later). We evaluate the proposed combined SRH in a benchmark functions set included in the Appendix. Our results show that the heuristic is competitive and robust to avoid local minima (with high probability).

This paper is organized in the following way. Section 2 introduces the Shake – Regicide Heuristic. Section 3 lists the pseudo-code of the Genetic Algorithm, Evolution Strategy, and Differential Evolution algorithms modified with the Shake - Regicide Heuristic. The experimental results are reported in Section 4. Conclusions and final remarks are provided in Section 5.

2 Shake – Regicide Heuristic

The Dynasty Period (DP), Shake Heuristic (SH) and Regicide Heuristic (RH) are presented in this section. The context of these heuristics is the framework of Evolutionary Algorithms. However, the SRH could be straightforward generalized to other evolutionary or distribution-estimation based algorithms, as our experiments illustrate.

Firstly, we introduce our notation. Let $X^k = \{x_1^k, x_2^k, \dots, x_m^k\}$ be the vector of m individuals (population) at the k -th iteration, where the vector

$x_i^k = \{x_i^{1^k}, x_i^{2^k}, \dots, x_i^{n^k}\}$ in n dimension, denotes the i -th individual. Then, $f: \mathcal{R}^n \rightarrow \mathcal{R}$ defines the fitness function and $Y^k = \{f(x_1^k), f(x_2^k), \dots, f(x_m^k)\}$ the fitness vector with x_B^k as the best individual in the k -th iteration, i.e. $f(x_B^k) \leq f(x_i^k) \forall i$.

2.1 Dynasty or Hegemonic Periods

Although hegemonic individuals are, implicitly, promoted in the naturist evolution, the societies, no necessarily humans, have implemented mechanisms for its control. For instance, legal frameworks forbid economical monopolies and dictators are commonly eliminated in conspiracies developed at corridors of power. The Sherman Act [9] is an example of a legal mechanism for preventing the consolidation hegemonic individuals in economy. The Sherman Act provides: "Every contract, combination in the form of trust or otherwise, or conspiracy, in restraint of trade or commerce among the several States, or with foreign nations, is declared to be illegal" and "Every person who shall monopolize, or attempt to monopolize, or combine or conspire with any other person or persons, to monopolize any part of the trade or commerce among the several States, or with foreign nations, shall be deemed guilty of a felony". The Sherman Act promotes the elimination of monopolies (Hegemony individuals in economy) because monopolies constrain the free competence between independent companies and decrease the capability decision of consumers.

Moreover, a dynasty is a continuous succession of rules that belong to the same family (or group, for our purposes). The dynasty period describes the era during which that family reigned. A long dynasty period produces a government with high hegemony [13], thus the local economy and social situation decrease for the population. Examples of long dynastic periods are the Chinese dynasty between I to X century, the Ming and Quing dynasties, the Romanov dynasty in Russia, and the Ottoman dynasty in Turkey.

In EA, we call "leader" or "king" the best individual of the population. A dynasty period is the number of generation during which a king or a king's son is the best individual in the population. In the SRH, a dynasty period is regulated by the λ parameter. The Regicide heuristic is applied to the population in order to emulate the dead or decline of dynastic periods. Thus, the fittest individual is systematically eliminated every λ generation. The Shake heuristic emulates convulsionary periods, those like revolutions. Regicide is applied immediately after a Shake.

2.2 Shake Heuristic

The Shake heuristic generates a new candidate population with larger variance in a neighborhood of the space around the mean of population. There is only one candidate for each individual in the population. A candidate replaces the current individual if it gets better fitness value. The first step of the Shake is the computation of the mean, μ , and standard deviation, σ , for each dimension of the population. If σ is less

than a given threshold ρ , the candidate population \hat{X} is generated with uniform distribution in the interval $[\mu-\sigma, \mu+\sigma]$. Next, X and \hat{X} are compared, individual to individual, preserving the best one of every pair. The Shake heuristic is formally defined in the next algorithm:

Shake Heuristic

1. Assign $\rho=0.1$ and compute the parameters μ^j and σ^j .

$$\mu^j = \frac{1}{m} \sum_{i=1}^m x_i^j, \quad i = 1, \dots, m \text{ and } j = 1, \dots, n \quad (2)$$

$$\sigma^j = \sqrt{\frac{1}{m} \sum_{i=1}^m (x_i^j - \mu^j)^2}, \quad i = 1, \dots, m \text{ and } j = 1, \dots, n \quad (3)$$

2. If $\sigma^j < \rho$ then $\sigma^j = \rho$.

3. Generate a new vector $\hat{X} = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m\}$, where \hat{x}_i^j is randomly generated with an uniform distribution.

$$\hat{x}_i^j \sim U(\mu^j - \sigma^j, \mu^j + \sigma^j), \quad i = 1, \dots, m \text{ and } j = 1, \dots, n \quad (4)$$

4. If $f(\hat{x}_i) \leq f(x_i)$ $i=1, \dots, m$, then $x_i \leftarrow \hat{x}_i$.

2.3 Regicide Heuristic

Regicide Heuristic eliminates the fittest individual x_B . Next, a substitute \hat{x}_R is generated close to the centroid of the population (a more democratic individual). Finally, \hat{x}_R substitutes (or murders, in social terms) the best of the population (x_B). The regicide strategy is defined in the next algorithm:

Regicide Heuristic

1. Assign $\varepsilon = 0.01$.

2. Compute the individual \hat{x}_R , where

$$\hat{x}_R^j = \mu^j + \varepsilon \sigma^j D, \quad j = 1, \dots, n \quad (5)$$

and $D \sim U(-1, 1)$, μ^j and σ^j are computed by equations (2) and (3).

3. Perform $x_B^k \leftarrow \hat{x}_R$.

3 Evolutionary Algorithms

The pseudo-code of a GA, ES, and DE algorithms with the Shake-Regicide Heuristic is provided here. These are versions used for the Experiments.

GA-SRH Algorithm

1. Set $k=0$ (generation) and create initial random population $X^k = \{x_1^k, x_2^k, \dots, x_m^k\}$ and $Y^k = \{f(x_1^k), f(x_2^k), \dots, f(x_m^k)\}$ where $f(x_i^k) \leq f(x_{i+1}^k)$.

2. Repeat until stop_condition==true

2.1 Create $S = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_r\}$ (Selection) with $r = m/2$ and

$$\hat{x}_i = \begin{cases} \arg \min_{x_p^k, x_q^k} \{f(x_p^k), f(x_q^k)\} & i = 4, 5, \dots, r \\ x_i^k & i = 1, 2, 3 \end{cases} \quad (6)$$

where x_p^k and x_q^k are randomly selected in X^k .

Perform $Y_S = \{f(\hat{x}_1), f(\hat{x}_2), \dots, f(\hat{x}_r)\}$.

2.2 Create $R = \{\bar{x}_{r+1}, \bar{x}_{r+2}, \dots, \bar{x}_m\}$ (Reproduction) where

$$\bar{x}_i = \alpha \hat{x}_p + (1 - \alpha) \hat{x}_q, \quad i = r+1, r+2, \dots, m \quad (7)$$

where \hat{x}_p and \hat{x}_q are randomly selected in S and $\alpha \sim U(0,1)$.

2.3 Create $M = \{\tilde{x}_{r+1}, \tilde{x}_{r+2}, \dots, \tilde{x}_m\}$ (Mutation) where

$$\tilde{x}_i^j = \begin{cases} \bar{x}_i^j + N(0,1) & j = J \\ \bar{x}_i^j & \text{other} \end{cases}, \quad i = r+1, r+2, \dots, m \quad (8)$$

where $J \sim U(1, n)$. Compute $Y_M = \{f(\tilde{x}_{r+1}), f(\tilde{x}_{r+2}), \dots, f(\tilde{x}_m)\}$.

2.4 Set $k=k+1$, perform $X^k = S \cup M$ where $f(x_i^k) \leq f(x_{i+1}^k)$.

2.5 Every λ iterations, perform Shake Heuristic and Regicide Heuristic.

ES-SRH Algorithm

1. Set $k=0$ (generation) and create initial random population $X^k = \{x_1^k, x_2^k, \dots, x_m^k\}$ and $Y^k = \{f(x_1^k), f(x_2^k), \dots, f(x_m^k)\}$ where $f(x_i^k) \leq f(x_{i+1}^k)$.

2. Repeat until stop_condition==true

2.1 Create $R = \{\bar{x}_1, \bar{x}_2, \dots, \bar{x}_r\}$ (Reproduction) with $r=2m$

$$\bar{x}_i = \alpha x_q^k + (1 - \alpha) x_q^k, \quad i = 1, 2, \dots, r \quad (9)$$

where x_p^k and x_q^k are randomly selected in X^k and $\alpha \sim U(0,1)$.

2.2 Create $M = \{\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_r\}$ (Mutation) where

$$\tilde{x}_i^j = \begin{cases} \bar{x}_i^j + N(0, \sigma^j) & j = J \\ \bar{x}_i^j & \text{other} \end{cases}, i=1,2,\dots,r \quad (10)$$

where $J \sim U(1,n)$ and σ is computing in (3). Compute $Y_M = \{f(\tilde{x}_1), f(\tilde{x}_2), \dots, f(\tilde{x}_r)\}$.

2.3 Create $S = X^k \cup M = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_{m+r}\}$ (Selection) and perform $f(\hat{x}_i) \leq f(\hat{x}_{i+1})$, $i=1, \dots, m+r-1$.

2.4 Set $k=k+1$, perform $X^k = \{\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m\}$.

2.5 Every λ iterations, perform Shake Heuristic and Regicide Heuristic.

DE-SRH Algorithm

1. Set $k=0$ (generation) and create initial random population $X^k = \{x_1^k, x_2^k, \dots, x_m^k\}$ and $Y^k = \{f(x_1^k), f(x_2^k), \dots, f(x_m^k)\}$ where $f(x_i^k) \leq f(x_{i+1}^k)$.

2. Repeat until stop_condition==true

2.1 For $i=1, \dots, m$.

2.1.1 Select $\alpha, \beta, \gamma \sim U(1,m)$ and $\alpha \neq \beta \neq \gamma$, perform

$$\hat{x}_i = x_\alpha^k + F(x_\beta^k - x_\gamma^k) \quad (11)$$

with $F=0.3$.

2.1.2 If $f(\hat{x}_i) \leq f(x_i^k)$, perform $x_i \leftarrow \hat{x}_i$.

2.2 Every λ iterations, perform Shake Heuristic and Regicide Heuristic.

5 Experiments

In this section, SRH was applied to global minimization problems in order to verify its performance and compared with standard versions of GA, ES and DE. This is the benchmark functions were solved with the three algorithms, each algorithm with and without using the SRH (the standard GA does not include step 2.5 of GA-SRH; the standard ES does not include step 2.5 of ES-SRH; and the standard DE does not include the step 2.2 of DE-SRH).

Table 1. Comparison of GA and GA-SRH.

Function	Best		Mean		Worst	
	GA	GA-SRH	GA	GA-SRH	GA	GA-SRH
f_0 n=30	183.02	0.0015	391.43	0.0120	764.79	0.0205
f_0 n=50	663.22	0.0458	951.58	0.0745	1552.36	0.1181
f_0 n=100	1680.70	0.3516	2415.95	0.4632	3332.87	0.5857
f_1 n=30	28.53	0.0355	66.84	0.2423	133.66	0.5015
f_1 n=50	69.29	0.4489	155.51	0.8033	241.01	1.13
f_1 n=100	282.29	3.61	485.66	4.38	770.74	5.45
f_2 n=30	26.96	0.4497	69.86	1.60	128.07	3.30
f_2 n=50	74.16	4.21	137.44	10.87	238.85	14.51
f_2 n=100	198.91	47.16	300.36	66.29	418.42	80.39
f_3 n=30	0.2548	0.0007	0.4899	0.0014	0.7057	0.0025
f_3 n=50	0.4932	0.0037	0.7088	0.0052	0.8724	0.0084
f_3 n=100	0.7022	0.0115	0.8610	0.0191	0.9737	0.0257
f_4 n=30	0.6685	0.0222	1.19	0.0716	1.75	0.1101
f_4 n=50	1.03	0.1501	1.58	0.1884	2.06	0.2556
f_4 n=100	1.54	0.35	1.87	0.44	2.10	0.59

Table 2. Comparison of ES ($\mu + \lambda$) and ES-SRH ($\mu + \lambda$).

Function	Best		Mean		Worst	
	ES	ES-SRH	ES	ES-SRH	ES	ES-SRH
f_0 n=30	38.64	0.0210	95.58	0.0323	180.98	0.0558
f_0 n=50	142.63	0.0757	232.11	0.1080	333.53	0.1447
f_0 n=100	444.25	0.4974	597.20	0.6270	759.51	0.7452
f_1 n=30	18.95	0.0832	27.23	0.3510	34.58	0.6176
f_1 n=50	43.66	0.8146	59.03	1.03	85.57	1.34
f_1 n=100	126.31	3.89	146.78	5.08	182.76	6.04
f_2 n=30	4.46	1.47	11.68	4.33	17.69	8.04
f_2 n=50	17.84	9.43	26.77	13.95	42.48	24.12
f_2 n=100	48.54	24.80	66.05	35.59	90.01	47.56
f_3 n=30	0.0666	0.0017	0.1593	0.0025	0.2543	0.0036
f_3 n=50	0.1352	0.0038	0.2567	0.0062	0.4227	0.0092
f_3 n=100	0.2383	0.0152	0.3344	0.0198	0.5103	0.0243
f_4 n=30	0.2975	0.0896	0.5138	0.1603	0.7747	0.2211
f_4 n=50	0.5025	0.1978	0.6972	0.2721	0.9178	0.3261
f_4 n=100	0.6054	0.4256	0.8157	0.4805	1.04	0.5607

We performed 30 Monte Carlo independent runs of each algorithm for each benchmark function. The test problems are: Sphere model (f_0), generalized Rosenbrock (f_1), generalized Rastrigin (f_2), Ackley function (f_3) and generalized Griewank (f_4). The optimum value of these functions is 0.0. A population of 200 individuals was randomly initialized on the allowed range of either decision variable, and 500,000 fitness function evaluations. Shake is always applied after the regicide every 10 generations. Tables 1 to 5 summarize the results and comparison of problems in dimension 30, 50 and 100. In every case, the modified EA with SRH obtained a better result than the standard version.

Table 3. Comparison of ES ($\mu + \lambda$) and ES-SRH ($\mu + \lambda$). We used panmictic reproduction [16].

Function	Best		Mean		Worst	
	ES	ES-SRH	ES	ES-SRH	ES	ES-SRH
f_0 n=30	1.62	0.0049	8.94	0.0199	23.13	0.0471
f_0 n=50	25.43	0.0451	50.31	0.0822	87.51	0.1167
f_0 n=100	223.65	0.3223	297.04	0.4604	442.13	0.6376
f_1 n=30	10.15	0.2251	13.02	0.3246	16.52	0.4260
f_1 n=50	25.87	0.6850	31.29	1.13	39.65	1.43
f_1 n=100	83.78	4.39	101.79	5.77	126.70	7.65
f_2 n=30	0.2393	2.09E-9	0.7908	0.0014	2.0240	0.0231
f_2 n=50	2.24	0.0376	5.04	0.6402	7.46	1.80
f_2 n=100	21.06	8.81	29.56	14.84	41.62	22.20
f_3 n=30	0.0054	0.0009	0.0149	0.0019	0.0310	0.0029
f_3 n=50	0.0253	0.0036	0.0523	0.0062	0.0918	0.0089
f_3 n=100	0.1126	0.0173	0.1559	0.0225	0.2237	0.0281
f_4 n=30	0.0471	1.24E-4	0.0882	0.0203	0.1359	0.1045
f_4 n=50	0.1388	0.0543	0.2336	0.1225	0.3575	0.1950
f_4 n=100	0.3442	0.2815	0.4705	0.3884	0.5689	0.5000

Table 4. Comparison of DE and DE-SRH.

Function	Best		Mean		Worst	
	DE	DE-SRH	DE	DE-SRH	DE	DE-SRH
f_0 n=30	208.83	0.0019	522.37	0.0103	944.19	0.0200
f_0 n=50	915.08	0.0311	1397.02	0.0695	2070.54	0.0985
f_0 n=100	2335.25	0.3687	4100.89	0.4663	5832.83	0.6130
f_1 n=30	24.33	0.0840	82.96	0.2415	157.74	0.4242
f_1 n=50	115.46	0.5641	325.35	0.8765	625.48	1.28
f_1 n=100	515.29	3.83	1127.68	4.88	1751.66	5.55
f_2 n=30	41.35	0.0559	69.04	0.5756	96.98	2.20
f_2 n=50	145.41	3.93	195.31	8.83	231.89	23.30
f_2 n=100	290.44	37.96	521.48	59.74	622.95	85.15
f_3 n=30	0.3204	0.0008	0.5554	0.0017	0.8449	0.0030
f_3 n=50	0.5228	0.0041	0.7847	0.0058	0.9354	0.0081
f_3 n=100	0.9160	0.0159	0.9902	0.0198	1.05	0.0239
f_4 n=30	0.9735	0.0188	1.46	0.0600	1.86	0.1021
f_4 n=50	1.65	0.1314	2.02	0.1886	2.39	0.4463
f_4 n=100	1.81	0.3263	2.28	0.4197	2.64	0.5369

Table 5. Comparison of DE and DE-SRH. Both DE versions ran with panmictic reproduction [16].

Function	Best		Mean		Worst	
	DE	DE-SRH	DE	DE-SRH	DE	DE-SRH
f_0 n=30	0.0	0.0	0.0	0.0	0.0	0.0
f_0 n=50	3.78E-42	3.19E-43	1.09E-41	6.69E-42	1.96E-41	1.26E-42
f_0 n=100	2.26E-14	1.60E-16	3.16E-14	3.03E-16	5.09E-14	4.28E-16
f_1 n=30	4.05E-13	4.68E-14	1.607E-6	9.38E-14	3.08E-5	2.54E-13
f_1 n=50	2.20E-13	2.34E-14	5.52E-13	5.75E-14	1.16E-12	1.12E-13
f_1 n=100	6.18E-9	2.67E-10	1.75E-8	6.90E-10	4.27E-8	2.51E-9
f_2 n=30	110.61	45.88	125.16	102.29	138.69	134.94
f_2 n=50	260.53	131.78	289.64	210.53	308.06	293.39

f_2 n=100	684.80	357.06	728.57	452.29	753.91	554.98
f_3 n=30	0.0	0.0	0.0	0.0	0.0	0.0
f_3 n=50	0.0	0.0	2.34E-20	5.42E-21	5.42E-20	8.35E-21
f_3 n=100	2.14E-17	2.54E-18	4.38E-17	5.65E-18	1.15E-16	1.06E-17
f_4 n=30	1.98E-6	2.34E-7	1.98E-6	2.34E-7	1.98E-6	2.34E-7
f_4 n=50	1.98E-6	3.89E-7	2.11E-6	3.92E-7	3.89E-6	3.99E-7
f_4 n=100	5.60E-6	4.71E-6	5.80E-6	4.71E-6	5.90E-6	4.71E-6

6 Conclusions and Remarks

Exploration is an important issue in Evolutionary Algorithm because poor exploration leads EAs to local minima. A novel heuristic, named SRH, is proposed in this paper. SRH is inspired in social investigations and economic laws to promotes wider exploration of the EA. The shake heuristic promotes exploration by spawning new individuals in a neighborhood around the population mean. After a shake, the regicide heuristic is applied to eliminate hegemonic individuals.

Genetic Algorithm discussion: the GA version with SRH clearly improves the standard GA by several orders of magnitude. Note that both GA versions use elitism of three individuals. The regicide heuristic replaces the best elite with one individual generated about the centroid, and the other two are copied to the new population. However, removing the elite did not decrement the chance to converge to the optimum value, in fact, GA-SRH is always better.

Evolution Strategy discussion: ES with standard reproduction (no panmictic) is shown in Table 2. Table 3 show results for panmictic reproduction. Again, in both tables, the use of the SR heuristic improves the results. Note that ES with SRH (table 2, column EE-SRH), performs better than the ES with panmictic reproduction (table 3, column ES). Panmictic reproduction is the natural approach of ES to increase the diversity, however, the SRH itself is able to improve the panmictic approach.

Differential Evolution discussion: in these experiments, the use of panmictic reproduction did not improve that much the SRH results. However, SRH version is still better.

Note that the search mechanism of DE and ES is self-adaptable but GA is not.

References

1. Hager, W. W., Huang, S., Pardalos, P. M., Prokopyev, O. A.: Multiscale Optimization Methods and Applications (Hardcover). Springer Verlag, 1 edition (2005).
2. Davis, L.: Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York (1991).
3. Goldberg, D.E., Genetic Algorithms in search, Optimization, and Machine Learning. Addison- Wesley, Reading, MA, (1989).
4. Holland, J. H. , Adaptation in natural and artificial systems. The University of Michigan Press, Ann Harbor, MI, (1975).
5. Back, T., Evolutionary algorithms in theory and practice, Oxford University Press, (1996).

6. Storn, R., Price, K., Differential evolution- A simple and efficient adaptive schemes for global optimization over continuous space, Technical Report TR-95-012, International Computer Sciences Institute, Berkeley, California, (1995).
7. Talbi E-G.: A taxonomy of hybrid metaheuristics. In: Journal of Heuristics 8(5), (2002) 541-564.
8. Kaelo, P., Ali, M. M.: Differential evolution algorithms using hybrid mutation, In: Computational Optimization and Applications, 37 (2), (2007) 231-246.
9. Dong, H., He, J., Huang, H., Hou W.: Evolutionary programming using a mixed mutation strategy. In: Information Sciences. 177(1): (2007) 312-327.
10. Zhang, Q., Sun, J., Tsang, E.: An Evolutionary Algorithm with Guides Mutation for the maximum clique problem. In: IEEE Transaction on Evolutionary Computation, vol. 9, no. 2 April (2005) 192-200.
11. Sun, J., Zhang, Q., Tsang E. P. K.: DE/EDA: A new evolutionary algorithm for global optimization. In: Information Sciences 169 (2005) 249-262.
12. Robles, V., J. M. Peña, Perez, M. S., Herrero, O.: Extending the GA-EDA Hybrid Algorithm to study Diversification and Intensification in GAs and EDAs. In: Lecture Notes in Computer Science No. 3646 (2005) 339 – 350.
13. Kennedy, P., The Rise and Fall of the Great Powers, Random House, (1987)
14. Sherman Act, ch. 647, 26 Stat. 209, 15 U.S.C., July 2, (1890)
15. Hart-Scott-Rodino Antitrust Improvements Act. Public Law 94-435, USA (1976)
16. Yapo, P.O., Gupta, H. V. and Sohooshian, S. Multi-Objective Global Optimization for Hydrologic Models. In: Journal of Hydrology, 204, 83-97 (1998)

Appendix: Benchmark functions

(1) Sphere model:

$$f_0(x) = \sum_{i=1}^n x_i^2, \quad -100 \leq x_i \leq 100,$$

where $f_{\min}=0$ and $x_{\min}=(0, \dots, 0)$.

(2) Generalized Rosenbrock's function:

$$f_1(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2) + (x_i - 1)^2) \quad -30 \leq x_i \leq 30,$$

where $f_{\min}=0$ and $x_{\min}=(1, \dots, 1)$.

(3) Generalized Rastrigin's function:

$$f_2(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10) \quad -5.12 \leq x_i \leq 5.12,$$

where $f_{\min}=0$ and $x_{\min}=(0, \dots, 0)$.

(4) Ackley's function:

$$f_3(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e, \quad -32 \leq x_i \leq 32,$$

where $f_{\min}=0$ and $x_{\min}=(0, \dots, 0)$.

(5) Generalized Griewank function

$$f_4(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(x_i / \sqrt{i}\right) + 1, \quad -600 \leq x_i \leq 600,$$

where $f_{\min}=0$ and $x_{\min} = (0, \dots, 0)$.