

A Hybrid Evolutionary Algorithm Based on Alopex and Estimation of Distribution Algorithm and Its Application for Optimization^{*}

Shaojun Li , Fei Li, and Zhenzhen Mei

Institute of Automation, East China University of Science and Technology,
Shanghai, 200237, P.R. China
lishaojun@ecust.edu.cn

Abstract. Alopex is a correlation-based algorithm, which shares characteristics of both gradient descent approach and simulated annealing. It has been successfully applied to continuous and combinatorial optimization problems for years. Estimation of Distribution Algorithms (EDAs) is a class of novel evolutionary algorithms (EAs) proposed in recent years. Compared with the traditional EAs, it possesses unique evolutionary characteristics. In this paper, a hybrid evolutionary algorithm (EDA-Alopex) is proposed, which integrates the merits of both Alopex and EDA, and obtains more evolutionary information than these two approaches. The new algorithm is tested with several benchmark functions; numerical case study results demonstrate that EDA-Alopex outperforms both EDA and AEA, especially for the complex multi-modal functions. Finally, the proposed algorithm is investigated on high-dimensional and multi-peaks benchmark functions, and it also achieves satisfactory results.

Keywords: Alopex, Estimation of distribution algorithm, Meta-heuristics, Evolutionary algorithm.

1 Introduction

Alopex (Harth and Tzanakou, 1974) [1-3] was first proposed to solve combinatorial optimization and pattern match problems. Inspired by the Alopex algorithm, a new evolutionary algorithm (EA) was proposed by the author in [2], called as Apolox-based evolutionary algorithm (AEA). In the AEA, the correlation calculation and Boltzmann-type probability in Alopex algorithm are introduced to evolve swarms, and an adaptive annealing schedule is adopted instead of the one used in the original Alopex algorithm. These are the two major differences between the AEA and the Alopex. Moreover, during the iterative progress of the AEA, two populations need to be generated to calculate the correlation which is used for heuristic information. Therefore, to some extent, the information contained in the two populations determines the convergence

^{*} This work was supported by Chinese National Programs for High Technology Research and Development(Grant No.2007AA04Z171) and National Natural Science Foundation of China(Grant No.20976048).

speed and solution quality. Comprehensive comparisons had been carried out between the AEA and genetic algorithm (GA) [4], particle swarm optimization (PSO) [5], as well as differential evolution (DE) [6]. The results show that the performance of AEA is superior to these EAs[2].

Estimation of Distribution Algorithm (EDA) [7-9] was proposed by Mühlenbein and Paaß (1996) as an extended version of GA, which is a novel kind of EAs. The solutions of EDA are generated by sampling from probability model derived from the promising individual set instead of crossover, mutation or differential operations implemented in the traditional EAs [8]. In EDAs, the global probability information is used as heuristic information.

Inspired by the search information extracted by both EDA and AEA is fundamentally distinct, in this paper, EDA is embedded into AEA to form a hybrid algorithm [10, 11] EDA-Alopex, for the purpose of improving the population diversity and acquiring more heuristic information from searching process. In EDA-Alopex, EDA is used to generate two populations, which are then passed to AEA for correlation calculation, providing more information for the evolution of AEA and in favor of an evolving towards to the promising domain. Due to the EDA is computation inexpensive, there is no significant increase of runtime of AEA, however, a relatively substantial improvement of performance is achieved, especially for the convergence rate.

The outline of the paper is organized as follows. In section 2, the basic principles of AEA and EDA will be illustrated for completeness. In section 3, the detailed steps of the EDA-Alopex are listed. In section 4, ten benchmark functions and different test methods are used to investigate the performance of the hybrid algorithm. Conclusions are given in the last section.

2 Principles of AEA and EDA

AEA is a new swarm intelligence evolutionary algorithm based on Alopex algorithm. With regard to the basic Alopex algorithm [3], the temperature ‘T’ for the first annealing cycle is preset subjectively. It is a great blindness of setting an initial temperature empirically, especially for the optimization problem, of which no priori domain knowledge is obtained. In AEA, adaptive annealing strategy is adopted. There is no need to set the initial temperature. From the beginning, the process of annealing is self-regulation, which not only improves the performance of the algorithm but also makes the algorithm simpler. The schedule of annealing applied in AEA is described by the following Eq.5.

The main process of the population-based AEA is expressed as follows [2, 3]: consider a minimization for the functions $F(x_1, x_2 \cdots x_N)$, where, $x_1, x_2 \cdots x_N$ are variables. Suppose there are two populations x_1, x_2 , the population size is L and variable dimension is N . Randomly select two individuals from x_1 and x_2 , which represent two state-vector at moment $t-1$ and $t-2$, respectively. And then the product $C = \Delta x \Delta F$ between vector difference Δx and corresponding function value ΔF is calculated, which describes the correlation between different individuals in the population. And then the probability is used to determine walk direction of each variable of individual, aiming at making each variable walk towards to the direction that can reduce the function value

and enhancing the climbing ability. Then each individual in population $X1$ is renewed according to Eq.1 to form new individuals. Determine whether the function value of the new individuals get improvement, the lesser one is retained as the individual for the next generation. The formula is as follows:

$$X1^k(i, j)' = X1^k(i, j) + (X2^k(i, j) - X1^k(i, j)) \cdot \delta(i, j) \cdot \text{rand}(0, 1) \quad (1)$$

$$\delta(i, j) = \begin{cases} 1 & \text{with probability } p^k(i, j) \\ -1 & \text{with probability } 1 - p^k(i, j) \end{cases} \quad (2)$$

$$p^k(i, j) = \frac{1}{1 + \exp\left(\frac{\pm C^k(i, j)}{T^k}\right)} \quad (3)$$

$$C^k(i, j) = [X1^k(i, j) - X2^k(i, j)] \times [F(X1^k(i, :)) - F(X2^k(i, :))] \quad (4)$$

$$T^k = \frac{1}{LN} \sum_{i=1}^L \sum_{j=1}^N C^k(i, j) \quad (5)$$

Where, $X1^k(i, j)$ denotes the j -th variable of i -th individual in population $X1$ at the k -th iteration. $\text{rand}(0, 1)$ is a random number which subjects to uniform distribution between 0 and 1. $\delta(i, j)$ is walk direction of j -th variable of i -th individual. The selection of positive and negative sign in the Eq.3 depends on the purpose of optimization. The positive or negative sign minimize or maximize the object to be optimized, respectively. In Eq.4, $F(X1^k(i, :))$ denotes function value of i -th individual in population $X1$. In Eq.5, T^k is the annealing temperature during the k -th iteration.

According to the dependence between variables, there are several kinds of EDAs have been proposed. In this paper, UMDAc [12] (Univariate Marginal Distribution Algorithm for Continuous Domains) is employed, in which interdependence between variables are not considered. In UMDAc, the n -dimensional joint probability distribution is factorized as a product of n independent normal distributions [13]. Suppose the k -th iteration, the joint density function of n -dimensional variable is computed as follows:

$$f^k(x) = \prod_{i=1}^N \frac{1}{\sqrt{2\pi}\sigma_i^k} \exp\left(-\frac{1}{2} \left(\frac{x_i^k - \mu_i^k}{\sigma_i^k}\right)^2\right) \quad (6)$$

In Eq.6, the mean μ_i^k and the standard deviation σ_i^k need to be estimated. In this chapter, maximum likelihood estimation is adopted for the estimation of parameters.

Overall speaking, process of EDA can be divided into four parts: selection, modeling, sampling and replacement. Firstly, best individuals are selected from the initial population to form a promising individual set. There are two problems for selection need to be determined: Selection ratio and selection method. The number of selected individuals divided by total population denoted the selection ratio ω . As for selection method, truncation selection [14] or tournament selection [12] can be applied. Secondly, the probability can be established by estimating the mean μ and the standard deviation σ . The third step is sampling, which guarantees the descendants sampled from the probability

model to scatter on the most favorable search area. Finally, whether the new population replaces the old one totally or not is determined by the scheme of replacement in which the replacement ratio v within the range of 0 to 1 is put up.

3 EDA-Alopex Algorithm

EDA and AEA evolve in different ways. In EDA, there is no operation of individuals and the probability model is used to guide the evolution of population, which is the largest distinction compared with the traditional swarm intelligent algorithm. In EDA, evolutionary information is obtained through the probability model which characterizes the global statistical information [13]. On the contrary, the most parts of information acquired by AEA are local correlation information [3]. On the foundation of AEA and EDA, a hybrid evolutionary algorithm called EDA-Alopex is proposed in this chapter. In the new algorithm, EDA is embedded into AEA to expect to provide more search information for the evolution of AEA and enhance the diversity of the population.

From the main process of AEA described in section 2, we can know that two populations X_1 and X_2 is produced for the computation of the correlation C , which is the main driving force for the evolution of population. Therefore, the more evolution information contained in the two populations, the greater probability owned to find the optimal solution. Inspired by the unique evolutionary model adopted by EDA and only need to allocate less time for the implementation of EDA, EDA is embedded into AEA to generate the two populations, which brings a richer population and improves the convergence speed of AEA.

The step size of original AEA expressed in section 2 is fixed, but varied step length changing with search process is more reasonable in the perspective of evolution. In order to complete AEA, a varied step size changing with iterations is applied in this paper. In the early stage of search process, a relatively lager step size promotes the swarm skim over an extensive search area. As for the later period of search process, due to the difference between individuals is little; the population is easier to be trapped at a local optimum, which can be alleviated through using of a relatively larger step size. The scheme of step size varying with iterations is given as follows:

$$X1^k(i, j)' = X1^k(i, j) + (X2^k(i, j) - X1^k(i, j)) \cdot \delta(i, j) \cdot \text{rand}(0, 1) \times \gamma^k \quad (7)$$

$$\gamma^k = \begin{cases} \alpha 0 \times \exp\left(\frac{k}{M} \log\left(\frac{\alpha 1}{\alpha 0}\right)\right) & k \leq M/2 \\ \beta 0 \times \exp\left(\frac{2k}{M} \log\left(\frac{\beta 1}{\beta 0}\right)\right) & k > M/2 \end{cases} \quad (8)$$

Where: γ^k is an shrinkage factor, decreasing or increasing exponentially with the iterations, M denotes the total number of iterations, $\alpha 0$, $\alpha 1$, $\beta 0$ and $\beta 1$ are constant.

Here, the iterative steps for EDA-Alopex are described as follows: also a minimization for the functions $F(x_1, x_2 \dots x_N)$ is used as an example for explanation.

Step 1. Initialize the parameters for EDA-Alopex. A given maximum generation or optimization target for objective function is preset, initialize population randomly and

calculate the objective function value of every individual in population, set the number of iteration $k := 0$

Step2. According to the objective function value, select the promising solutions to form the promising solutions set. Truncation selection is applied in this chapter.

Step3. Based on the promising solutions set, compute the mean μ and the standard deviation σ of every variable, and then establish a probability model described by Eq.6.

Step4. Generate new solutions by sampling from the constructed probabilistic model; in according with the replace ratio v , replace the old population totally or in part to form a new population X_1 .

Step5. Re-arrange the order of the individuals in population X_1 to form a second population X_2 , according to Eq. 3-5 given above, the probability p , the correlation C and annealing temperature T are calculated respectively.

Step6. Each individual in population X_1 is renewed according to Eq.7 expressed above to form an intermediate population X_1' , value of objective function $F(X_1')$ for population X_1' and the corresponding function value $F(X_1)$ for population X_1 is compared, the lesser one is retained as the individual for the next generation.

Step7. If the given termination criterion is met, yes, go to step 8, else $k = k + 1$, go to step2.

Step8. Output the best individual and the best value.

As can be seen from the above steps, from step 1 to step 4, an EDA is implemented. The step 5 and 6 are steps for AEA. Each generation, the excellent solutions of population are used to construct probability model; then the population retaining the promising individuals of previous generation and the best individuals sampled from the probability model are passed to AEA for the correlation calculation. Therefore, compared with the single EDA or AEA, EDA-Alopex obtains a more diversity population.

In EDA-Alopex, for the existence of replacement ratio v , part of population comes from the offspring generated by EDA and the other part of population stems from descendents yielded by AEA for every generation. Therefore, two different evolutionary strategies are applied to drive the population moving towards the optimal solution. Actually, some individuals in population evolve in EDA way, while the other individuals evolve in AEA way. So offspring generated by EDA-Alopex include not only probability information which describes evolution from the macro level but also an Alopex heuristic information which is similar to gradient descent. For this reason, the hybrid algorithm can integrate the advantage of both AEA and EDA, guiding the population towards a more promising search space.

4 Simulation Results and Analysis

To verify the performance of EDA-Alopex, several benchmark functions, listed in table1, are used in the test. The comparisons between PSO, DE, AEA, and

EDA-Alopes are carried out. The parameters of PSO are given as follows: the initial inertia weight, $w_{start}=0.95$, the final inertia weight, $w_{end}=0.4$, the percentage of maximum iteration for a linearly changing weight, $w_{for}=0.7$, acceleration factor, $c1=c2=2$, constriction factor $Chi=1$ and the maximum velocity is fixed at one-tenth of variable interval. The parameters involved in DE are listed as follows: the mutation operator, $F=0.8$, DE strategy is DE/rand-to-best/1/bin, crossover probability, $CR=0.5$. In the EDA-Alopes, the value of $\alpha_0, \alpha_1, \beta_0$ and β_1 are fixed at 1.2, 0.8, 0.8, 1.2, respectively. The value of ω and v is set as 0.5, 0.9, separately.

In order to make a fair comparison, a total of 50 runs for each algorithm are implemented and the population size for the first two problems f_1 - f_2 and the remaining eight functions f_3 - f_{10} are fixed at 50 and 100, respectively. The maximum generations for all the four algorithms is set as 3000. At the same time, for the purpose of examining the performance of algorithms from another point of view, the target values of each test function is preset. The global minimum of the test functions f_1 - f_4 is -1, -1.03, 0, and -4189.82, respectively. All the other test functions own a global minimum of 0. A trial is considered to be succeeded if the value obtained is less than or equal to the target within the maximum iterations, otherwise it is believed the current search is failed. The target for every test function is also listed in table 1.

Table 1. Test functions

No	Name	Function	D	Target	Bounds
f_1	<i>Easom</i>	$-\cos(x_1) \times \cos(x_2) \times \exp(-(x_1 - \pi)^2 + (x_2 - \pi)^2)$	2	-0.99	[-100,100]
f_2	<i>Six hump</i>	$\left(4 - 2.1x_1^2 + x_1^{\frac{4}{3}}\right)x_1^2 + x_1x_2 + (-4 + 4x_2^2)x_2^2$	2	-1.03	[-5,5]
f_3	<i>Ellipsoidal</i>	$\sum_{i=1}^n (x_i - i)^2$	10	0.001	[-10,10]
f_4	<i>Schwefel</i>	$-\sum_{i=1}^n x_i \sin(\sqrt{ x_i })$	10	-4189.8	[-500,500]
f_5	<i>Exponential</i>	$1 - \left(\exp\left(-0.5 \sum_{i=1}^n x_i^2\right)\right)$	30	0.001	[-1,1]
f_6	<i>Griewank</i>	$1 + \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right)$	30	0.1	[-600,600]
f_7	<i>Rastrigin</i>	$10n + \sum_{i=1}^n \left[x_i^2 - 10 \cos(2\pi x_i)\right]$	30	100	[-5.12,5.12]
f_8	<i>Rosenbrock</i>	$\sum_{i=1}^{n-1} \left[100(x_{i+1} - x_i)^2 + (x_i - 1)^2\right]$	30	100	[-2.048,2.048]
f_9	<i>Sphere</i>	$\sum_{i=1}^n x_i^2$	30	0.001	[-5.12,5.12]
f_{10}	<i>Ackley</i>	$-20 \exp\left(-0.02 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	30	1	[-30,30]

To make a fair comparison, the average best functions value (ABFV); standard deviations (SD) over 50 runs of the total ten problems for the four algorithms are quoted in table 2.

Table 2. Comparative average best function value and standard deviation over 50 runs

No	EDA-Alopex	AEA	PSO	DE
f_1	-1±0	-1±0	-1±0	-0.98±1.41E-01
f_2	-1.03±0	-1.03±0	-1.03±2.24E-16	-1.03±0
f_3	0±0	0±0	8.00E-02±2.74E-01	0±0
f_4	-4.19E+03±1.84E-12	-4.19E+03±1.84E-12	-3.40E+03±2.26E+02	-4.15E+03±7.42E+01
f_5	0±0	2.42E-16±4.86E-17	1.04E-16±2.66E-17	1.11E-16±0
f_6	0±0	5.61E-03±8.31E-03	1.51E-02±1.60E-02	.34E-07±6.85E-07
f_7	1.80E+01±1.96E+01	7.53E+00±6.03E+00	4.21E+01±1.07E+01	1.19E+02±8.16E+00
f_8	1.83E+01±1.21E-01	2.37E+01±2.54E-01	2.38E+01±7.68E+00	1.35E+01±4.63E-01
f_9	1.25E-239±0	8.07E-52±9.83E-52	8.13E-31±2.38E-30	2.00E-32±1.31E-32
f_{10}	4.00E-15±0	2.10E-14±4.00E-15	1.50E-14±4.00E-15	8.85E-15±2.22E-15

In table 2, the best values obtained by four algorithms are marked by bold. The number of no worse solutions compared to the other three algorithms given by EDA-Alopex is 8, by AEA is 5, by PSO is 2, by DE is 3. Totally, most of solutions given by EDA-Alopex is superior to the one obtained by other algorithms. In table 3, rate of successful minimizations (RS), average generations (AG), average objective function evaluation numbers (AFEN) over 50 runs are listed to investigate the convergence and efficiency of algorithms. '-' denotes that the algorithm can not reach the target within the maximum iteration.

Table 3. Comparison of convergence and efficiency of four algorithms over 50 runs

No	EDA-Alopex			AEA			PSO			DE		
	RS	AG	AFEN	RS	AG	AFEN	RS	AG	AFEN	RS	AG	AFEN
f_1	50	12	1321	50	55	2882	50	335	16810	50	58	2964
f_2	50	7	778	50	17	892	50	62	3147	50	21	1106
f_3	50	32	6520	50	97	9786	43	880	88112	50	95	9574
f_4	50	193	38796	50	501	50244	-	0	0	38	891	89161
f_5	50	33	6696	50	100	10092	50	1149	114992	50	304	30518
f_6	50	60	12116	50	244	24540	50	1559	156016	50	634	63544
f_7	50	770	154108	50	773	77448	50	848	84886	2	2650	265050
f_8	50	16	3380	50	35	3568	50	343	34440	50	228	22852
f_9	50	51	10288	50	195	19642	50	1454	145522	50	454	45538
f_{10}	50	40	8192	50	130	13132	50	1330	133106	50	392	39250

In table 3, it can be clearly observed that 100% success for all the test problems given by EDA-Alopex and AEA. However, EDA-Alopex needs less iteration to reach the target value compared with AEA, especially for the number of objective function calls; a substantial decline is witnessed except function 7.

On the whole, EDA-Alopex achieves a balance between convergence speed and convergence precision. With regard to test function 7, the current parameter settings of EDA-Alopex are not recommended. We have confirmed that the new algorithm can achieve a better optimization result compared with the result given by AEA via using other parameters combination. To further validate the performance of the algorithms, three multi-apices and high-dimensional functions are employed for a scalability study. The test functions are Ackley (F1), Schwefel (F2) and Griewank (F3), the dimension is fixed at 100, 30 and 100 respectively and the global optimum of three test functions is 0, -12569.48 and 0, respectively. Number of maximum iterations is still fixed at 3000. The average best functions value (ABFV), standard deviations (SD) over 50 runs are listed in table 4.

Table 4. Optimization results for high-dimension test functions

No	EDA-Alopex	AEA	PSO	DE
F1	4.20E-14±7.00E-15	2.65E+00±2.17E-01	1.55E-01±2.90E-01	7.65E-02±1.04E-02
F2	-1.24E+04±2.59E+02	-1.17E+04±5.38E+02	-7.88E+03±5.63E+02	-9.18E+03±8.96E+02
F3	5.42E-04±2.24E-03	1.13E-03±2.87E-03	1.28E-02±3.60E-02	9.53E-02±2.39E-02

As can be seen from table 4, in terms of Ackley function, ABFV and SD obtained by EDA-Alopex are significantly better than the other three algorithms. In terms of the difficulty level of optimization, Schwefel function is a difficult test problem for optimizing. However, ABFV obtained by EDA-Alopex is closest to the global optimal solution. With regard to the function of Griewank, EDA-Alopex obtains the best results compared with other three algorithms. Consequently, EDA-Alopex also demonstrates a good performance for the optimization of the function of the high-dimensional and multiple local minimum. Overall speaking, EDA-Alopex outperforms others, considering in a comprehensive point of view.

5 Conclusion

In this paper, a new hybrid algorithm EDA-Alopex is proposed, which combines the Alopex with the EDA. In the EDA-Alopex, EDA is embedded into AEA, aiming at improving the performance of AEA. As a global optimization algorithm, AEA shows better performance compared with the basic EAs, such as GA, POS, and DE, and it also has a high potential of parallelism. In the EDA-Alopex, two populations used for correlation calculation in AEA are generated by EDA, for the purpose of improving the population diversity. Therefore, with a higher possibility, the offspring generated by the EDA-Alopex can contain not only the global probability information but also the local correlation information. Furthermore, an adaptive variation scheme for step size

with iterations is introduced in the EDA-Alopex, which improves the convergence speed of the EDA-Alopex.

Ten widely used benchmark functions with different dimensions are utilized to investigate the performance of the EDA-Alopex. Each test problem possesses different characteristics, which examines the performance of the EDA-Alopex in a comprehensive way. Also different test methods are employed for comparison. The case study results show that the EDA-Alopex can achieve faster speed and better solution, however, without a substantial cost in the execution time compared with AEA. Consequently, based on the analysis above, it can be concluded that the proposed algorithm can achieve better performance than traditional EAs.

References

1. Harth, E., Tzanakou, E.: Alopex: A Stochastic Method for Determining Visual Receptive Fields. *Vision Research* 14, 1475–1482 (1974)
2. Li, S.J.: An Alopex based Evolutionary Optimization Algorithm. *Pattern Recognition and Artificial Intelligence* 22(3), 452–456 (2009) (in Chinese)
3. Unnikrishnan, K.P., Venugopal, K.P.: A Correlation-Based Learning Algorithm for Feed-Forward and Recurrent Neural Networks. *Neural Computation* 6(3), 469–490 (1994)
4. Holland, J.H.: *Adaptation in Natural and Artificial Systems*. The university of Michigan Press, Ann Arbor (1975), 2nd ed. MIT Press, Cambridge (1992)
5. Kennedy, J., Eberhart, R.: Particle Swarm Optimization. In: *Proceeding of the 1995 IEEE International Conference on Neural Networks*, pp. 1942–1948. IEEE Service Center, Piscataway (1995)
6. Storn, R., Price, K.: Differential Evolution - A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces. *Journal of Global Optimization* 11(4), 341–359 (1997)
7. Mühlenbein, H., Paaß, G.: From Recombination of Genes to the Estimation of Distributions. In: Mühlenbein, H., Bendisch, J., Voigt, H. (eds.) *PPSN 1996*. LNCS, vol. 1141, pp. 178–187. Springer, Heidelberg (1996)
8. Mühlenbein, H., Höns, R.: The Estimation of Distributions and the Minimum Relative Entropy Principle. *Evolutionary Computation* 13(1), 1–27 (2005)
9. Zhou, S.D., Sun, Z.Q.: A Survey on Estimation of Distribution Algorithms. *Acta Automatica Sinica* 33(2), 113–124 (2007) (in Chinese)
10. Abraham, A., Corchado, E., Corchado, J.M.: Hybrid Learning Machines. *Neurocomputing* 72(13–15), 2729–2730 (2009)
11. Lozano, M., García-Martínez, C.: Hybrid Metaheuristics with Evolutionary Algorithms Specializing in Intensification and Diversification: Overview and Progress Report. *Computers & Operations Research* 37(3), 481–497 (2010)
12. González, C., Lozano, J.A., Larrañaga, P.: Mathematical Modeling of UMDac Algorithm with Tournament Selection Behavior on Linear and Quadratic functions. *International Journal of Approximate Reasoning* 31(3), 313–340 (2002)
13. Sun, J.Y., Zhang, Q.F., Edward, P.K.T.: DE/EDA: A New Evolutionary Algorithm for Global Optimization. *Information Sciences* 169(3–4), 249–262 (2005)
14. Grahl, J., Minner, S., Rothlauf, F.: Behavior of UMDac with Truncation Selection on Monotonous Functions. In: *IEEE Congress on Evolutionary Computation (CEC)*, vol. 3, pp. 2553–2559 (2005)