A Comparison of Several Algorithms and Representations for Single Objective Optimization

Crina Grosan

Department of Computer Science
Faculty of Mathematics and Computer Science
Babeş-Bolyai University, Kogălniceanu 1
Cluj-Napoca, 3400, Romania.
cgrosan@cs.ubbcluj.ro

In this paper we perform two experiments. In the first experiment we analyze the convergence ability to using different base for encoding solutions. For this purpose we use the bases 2 to 16. We apply the same algorithm (with the same basic parameters) for all considered bases of representation and for all considered test functions. The algorithm is an (1+1) ES. In the second experiment we will perform a comparison between three algorithms which use different bases for solution representation. Each of these algorithms uses a dynamic representation of the solutions in the sense that the representation is not fixed and is changed during the search process. The difference between these algorithms consists in the technique adopted for changing the base over which the solution is represented. These algorithms are: Adaptive Representation Evolutionary Algorithms (AREA) [1], Dynamic Representation Evolution Strategy (DRES) and Seasonal Model Evolution Strategy (SMES) [2].

AREA change the alphabet if the number of successive harmful mutations for an individual exceeds a prescribed threshold. In DRES algorithm the base is changed at the end of each generation with a fixed probability. In SMES algorithm the base in which solution is encoded is changed after a fixed (specified) number of generations.

Test functions used in these experiments are are well known benchmarking problems ([3]): Ackley's function (f_1), Griewangk's function (f_2), Michalewicz function (f_3), Rosenbrock's function (f_4), Rastrigin's function(f_5) and Schwefel's function (f_6).

The essential role of these experiments is to show that using only one base for solution encoding (without change it during the search process) there are cases when the optimum cannot be found. Changing the representation base provides a new way of searching through the solution space. The second experiment show us which technique used for changing the base is suitable.

The number of space dimension was set to 30 for each test function. Each algorithm is run 100 times for each test function in each experiment and with any considered parameters.

In first experiment for test functions f_1, f_2 and f_4 the best results are obtained using binary encoding.

For test functions f_3, f_5 and f_6 the best result is obtained by encoding solutions in the base 4.

K. Deb et al. (Eds.): GECCO 2004, LNCS 3102, pp. 788–789, 2004.

In second experiment a comparison between AREA DRES and SMES is performed. The difference between these algorithms consists in the strategy used for changing the current base in which the solutions is encoded with another one. All considered algorithms used a population with a single individual. Parameters used for AREA are: *Number of alphabets* = 31; MAX_HARMFUL_MUTATION = 50 and *Number of mutation / chromosome* = 2. DRES and SMES use the same parameters as AREA uses. The probability of changing an alphabet in DRES is 0.02. The number of generations after SMES changes the alphabet is 50. The results obtained by these three algorithms are presented in Table 1.

Table 1. Results obtained by AREA, DRES and SMES for test functions f_1-f_6

Function	**Mean best**		
	AREA	**DRES**	**SMES**
f_1	1.6510	2.3978	2.1838
f_2	0.6328	0.7085	0.8307
f_3	-26.803	-26.6668	-26.949
f_4	146.756	161.53	156.776
f_5	8.1164	9.3483	10.2461
f_6	-11894.6	-11986.1	-11956.9

However, AREA significantly outperforms the standard evolutionary algorithms on the well-known difficult (multimodal) test functions. This advantage of AREA makes it very suitable for real-world applications where we have to deal with highly multi-modal functions. Had only one base been used for solution encoding the gain of AREA over standard ES would have been minimal. Thus, the AREA individuals use a dynamic system of alphabets that may be changed during (and without halting) the search process. If an individual gets stuck in a local optimum - from where it is not able to "jump"- , the individual representation is changed, hoping that this new representation will help the individual to escape from the current position and to explore farther and more efficiently the search space.

References

1. Grosan C., Oltean M.: Adaptive Representation Evolutionary Algorithm – a new technique for single objective optimization. In Proceedings of First Balcanic Conference in Informatics (BCI), Thessaloniki, Greece (2003) 345-355
2. Kingdon J, Dekker L. "The Shape of Space", Proceedings of the First IEE/IEEE International Conference on Genetic Algorithms in Engineering Systems: Innovations and Applications (GALESIA '95) IEE, London, (1995) 543-548
3. Yao, X., Liu, Y., lin, G.: Evolutionary programming made faster. IEEE Transaction on Evolutionary Computation, Vol. 3(2) (1999) 82-102