A Logarithmic Mutation Operator to Solve Constrained Optimization Problems

A. Petrowski
Dépârtment Informatique,
nstitut National des Télécommunications,
9, rue Charles Fourier, 91011 Evry, France;
e-mail: *Alain.Petrowski@int-evry.fr*

S. Ben Hamida
CMAP, URA CNRS 756,
Ecole Polytechnique,
Palaiseau 91128, France;
e-mail: *sana@cmapx.polytechnique.fr*

1 The optimization method

We propose a method based on improving the exploration of the search space to solve constrained optimization problems. This exploration is performed with a mutation operator, whose distribution function is logarithmic. A BLX-0.5 crossover is also used. These operators are associated to a rudimentary constraint handling method implemented by a dedicated ranking selection such that feasible individuals are more likely selected than unfeasible ones.

2 Results

Experiments were made for eight reference functions selected from the test cases proposed in [3]. The results were compared with those given for the methods presented in [1], [2] and [3]. The number of generations G for all problems and all methods is bounded by 5000. The results for each experiment with our method are given for 30 independent runs.

All experiments have always given feasible solutions for all problems. The method we present here provides better results than any earlier reported method, except for the test case G2.

Table 1 summarizes experimental results obtained by improving exploration and by the best other methods, for each test case. Method 1 refers to "Genocop II" method 2 refers to "Behavioral Memory", method 3 refers to the method of "Searching the Boundary of Feasible Region" and method 4 refers to "Homomorphous Mapping".

This constrained optimization method is easy to apply; it does not require any computation overhead; it does not require any additional parameter; it does not need evaluation of unfeasible solutions.

References

[1] Koziel, S. and Michalewicz, Z. (1999). *Evolutionary Computation*, Vol.7, No1, pp. 19–44.

[2] Michalewicz, Z. (1995). *Proceedings of the* 6^{th} *International Conference on Genetic Algorithms*, pp. 151–158. Morgan Kaufmann.

[3] Michalewicz, Z. and Schoenauer, M. (1996). *Evolutionary Computation*, Vol.4, No1, pp. 1–32.

		G1	G2	G4	G6	G7	G8	G9	G10
Exact Opt.		-15.000	0.803553	-30665.5	-6961.814	24.30621	0.095825	680.63006	7049.3309
G		5000	5000	2000	1000	5000	100	5000	5000
Improving	best	**-15.000**	0.80248	**-30665.5**	**-6961.81**	**24.384**	**0.095825**	**680.630**	**7070.33**
Exploration	median	**-15.000**	0.79430	**-30665.5**	**-6961.81**	**24.509**	**0.095825**	**680.637**	**7259.64**
	worst	**-15.000**	0.75832	**-30665.5**	**-6961.81**	**24.974**	**0.095825**	**680.657**	**8429.79**
Best results		meth. 2	meth. 3	meth. 4	meth. 4	meth. 4	meth. 4	meth. 2	meth. 1
obtained	best	**-15.000**	**0.80355**	-30662.5	-6901.5	25.132	**0.095825**	680.640	7377.976
with other	m/a	**-15.000**		-30643.8	-6191.2	26.619	0.0871551		8206.151
methods	worst	**-15.000**	**0.80296**	-30617.0	-4236.7	38.682	0.029143	680.889	9652.901

Table 1: Experimental results.