Communications to SIMAI Congress,
ISSN 1827-9015, Vol. 1 (2006)
DOI: 10.1685/CSC06130

Differential Evolution for Constrained Optimization Problems

R. Rascunà
University of Sussex
Pevensey II 4A23, UK
R.Rascuna@sussex.ac.uk

This work on constrained optimization problems presents preliminary results using Differential Evolution (DE) (Storn and Rice, 1997) as a tool to search in complex fitness landscape. Penalty functions were not used and a simple feasibility rule was implemented to choose between feasible and infeasible solutions. Equality constraints are dealt with a fixed tolerance and no extra diversity mechanism was used.

To prove the effectiveness of the approach a set of well known functions (Runarsson and Yao, 2000) was used and the comparison was made against the SMES algorithm (Mezura Montes and Coello Coello, 2005), which represents a state-of-the-art evolutionary algorithm in constrained optimization problems. The aims of this work is to maintain a simple approach and verify whether simplicity could also be an effective way in constrained optimization problems.

Differential Evolution is a search method that uses vectors of real numbers to represent its individuals. The idea of DE is to generate new vectors as a weighted sum of the difference between two or more vectors taken from the population. Since there is no mutation the number of parameters required by DE is minimal compared to other algorithms. In order to choose between feasible and infeasible solutions the same operator implemented in SMES was used. As stated in (Mezura Montes and Coello Coello, 2005) this operator works as follow: 1) between two feasible solutions, the one with the highest fitness value wins; 2) if one solution is feasible and the other one is infeasible, the feasible solution wins; 3) if both solutions are infeasible, the one with the lowest sum of constraint violation is preferred.

The strategy used for DE is named rand/1/exp. The population is represented by vectors of fixed length N. The initial population is created randomly respecting the boundaries of each variable. The parameters used are: F as a weighting constant, and CR as crossover probability. The following algorithm is repeated until the maximum number of function evaluations is reached:

1. For each individual $\mathbf{v}$ in the population
2. Select randomly three individuals $\mathbf{x}^1$, $\mathbf{x}^2$ and $\mathbf{x}^3$
3. Compute the individual $\mathbf{v}'$ with the following algorithm:

   $i =$ **rand** $(1, N)$;
   $L = 1$;
   do
   $\quad \mathbf{v}'_i = \mathbf{x}^1_i + F \cdot (\mathbf{x}^2_i - \mathbf{x}^3_i)$
   $\quad i = (i + 1)$ **mod** N
   $\quad L = L + 1$
   while rand() < CR **and** (L < N)

4. Copy in the new population either $\mathbf{v}'$ or $\mathbf{v}$ depending on the feasibility-based comparison defined earlier. The algorithm repeat from 1. until the new population is completely filled.

SMES is an $(\mu+\lambda)$-ES algorithm, where μ parents generate λ offsprings and all compete to survive in the next generation. It uses self-adaptive mutation and an hybrid panmictic recombination. The results obtained by SMES are very competitive when compared to other state-of-the-art techniques.

The key features of SMES are: 1) *Diversity Mechanism*, infeasible solutions survive in next generations, this allow recombination between feasible solutions and infeasible ones. 2) *Dynamic Tolerance*, a dynamic mechanism is used to deal with equality constraints. The tolerance value ϵ is decreased at every generation.

The settings used in the experiment are the following: both algorithms were allowed to perform 240000 objective function evaluations. The population size of DE is fixed to 50 individuals, crossover probability $CR = 0.9$ and weighting factor $F = 0.7$. These setting were chosen after several experiments. The authors also suggest similar parameters (Storn and Rice, 1997). The tolerance for equality constraints is fixed to 0.0001.

SMES uses populations size of $\mu = 100$, $\lambda = 300$ and the tolerance is updated with the following formula $\epsilon(t + 1) = \epsilon(t)/1.00195$. The initial value is set to $\epsilon_0 = 0.001$. Accordingly with the formula, in the last generation, the final value will be $\epsilon_T = 0.0004$. For function g03 and g13 those parameters where changed due to some difficulties reported to produce feasible solutions (Mezura Montes and Coello Coello, 2005).

To prove the effectiveness of the approach a set of 13 functions is considered. These functions represent a well-know benchmark in constrained optimization problems (Runarsson and Yao, 2000). Results summarized in Table 0.1 are obtained considering 30 independent runs for each algorithm. Considering that both algorithms are allowed to perform the same maximum number of objective function evaluations, DE is able to obtain the optimal solution in functions g05, g07, g09, and g10 in comparison with SMES.

The results obtained by DE on the “mean”, “median” and “worst” solutions also present an improvement over SMES. For function g02, DE was not able to find the optimum, however the result is better than that one achieved by SMES.

SMES obtains a better results for function g01, because of a better “mean” and “worst” solution, and for function g13, because DE is not able to find a feasible solution.

In conclusion, it was shown that even with a simple evolutionary algorithm with few parameters and a fixed tolerance to handle equality constrains it is possible to compete with a state-of-the-art algorithm such as SMES. Further works are needed to investigate the relations between the fixed tolerance value and the ability to find feasible solutions. It is also important to prove the approach with more test functions to ensure the quality of the algorithm.

REFERENCES

1. E. Mezura Montes and C. Coello Coello, A simple multimembered evolution strategy to solve constrained optimization problems. (2005). *IEEE Trans. Evol. Comp.* Vol 9, No. 1, 1–17.

2. T. P. Runarsson and X. Yao, Stochastic ranking for constrained evolutionary optimization. (2000). *IEEE Tran. Evol. Comput* Vol 4, Sept., 284–294.

Problem	Optimal	Best		Mean	
		SMES	DE	SMES	DE
g01	-15	**-15**	**-15**	-15	-14.433
g02	0.803619	0.803601	0.803616	0.785238	0.803590
g03	1.000	**1.000**	**1.000**	**1.000**	**1.000**
g04	-30665.539	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**
g05	5126.498	5126.599	**5126.498**	5174.492	**5126.498**
g06	-6961.814	**-6961.814**	**-6961.814**	-6961.284	**-6961.814**
g07	24.306	24.327	**24.306**	24.475	**24.306**
g08	0.095825	**0.095825**	**0.095825**	**0.095825**	**0.095825**
g09	680.63	680.632	**680.63**	680.643	**680.63**
g10	7049.25	7051.903	**7049.25**	7253.047	7103.039
g11	0.75	**0.75**	**0.75**	**0.75**	**0.75**
g12	1.000	**1.000**	**1.000**	**1.000**	**1.000**
g13	0.053950	0.053986	0.053941	0.166385	0.331386
Problem	Optimal	Median		Worst	
		SMES	DE	SMES	DE
g01	-15	**-15**	**-15**	-15	-12
g02	0.803619	0.792549	0.803597	0.751322	0.803466
g03	1.000	**1.000**	**1.000**	**1.000**	**1.000**
g04	-30665.539	**-30665.539**	**-30665.539**	**-30665.539**	**-30665.539**
g05	5126.498	5160.198	**5126.498**	5304.167	**5126.498**
g06	-6961.814	**-6961.814**	**-6961.814**	-6952.482	**-6961.814**
g07	24.306	24.426	**24.306**	24.843	**24.306**
g08	0.095825	**0.095825**	**0.095825**	**0.095825**	**0.095825**
g09	680.63	680.642	**680.63**	680.719	**680.63**
g10	7049.25	7253.603	**7049.25**	7638.366	7250.967
g11	0.75	**0.75**	**0.75**	**0.75**	**0.75**
g12	1.000	**1.000**	**1.000**	**1.000**	**1.000**
g13	0.053950	0.061873	0.438802	0.468294	0.438802

Table 0.1: Statistical results of best, mean, median and worst solutions obtained in 30 independent runs, with 240000 objective function evaluations.

3. R. Storn and K. Price, Differential Evolution "A Simple and Efficient Heuristic for Global Optimization over Continuous Spaces".(1997) *Journal of Global Optimization.* Vol. 11, No.342, 341–359.