International Conference
20th EURO Mini Conference
"Continuous Optimization and Knowledge-Based Technologies" (EurOPT-2008)
May 20–23, 2008, Neringa, LITHUANIA

ISBN 978-9955-28-283-9
L. Sakalauskas, G.W. Weber and E. K. Zavadskas (Eds.): EUROPT-2008 Selected papers. Vilnius, 2008, pp. 198–203

ON CHARGE EFFECTS TO THE ELECTROMAGNETISM-LIKE ALGORITHM

Ana Maria A. C. Rocha*, Edite M. G. P. Fernandes

Department of Production and Systems,
University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal
E-mail:{arocha;emgpf}@dps.uminho.pt

Abstract: This paper presents modifications of the electromagnetism-like (EM) algorithm for solving global optimization problems with box constraints. The modifications are concerned with the charges associated with each point in the population. The purpose here is to improve efficiency and solution accuracy by exploring the attraction-repulsion mechanism of the EM algorithm. Several widely used benchmark problems were solved in a performance evaluation of the new algorithm when compared with the original one. The modified algorithm has also been compared with other heuristic population-based methods.

Keywords: global optimization; electromagnetism-like algorithm; charge computation; performan-ce profiles.

1. Introduction

In this paper, we consider the problem of finding a global solution of a nonlinear optimization problem with box constraints

$$\min f(x), \quad \text{subject to} \quad l \le x \le u\,. \tag{1}$$

Most real problems of this type are non-differentiable and multimodal. Hence, gradient based methods cannot be used to find the global solution. To overcome this issue, many researchers have been proposing stochastic global methods that do not require any derivative computation. These methods generate a random sample of points and based on their objective function values move the sample in order to converge to optimality. Recently (Birbil and Fang, 2003) proposed a stochastic population-based method known as electromagnetism-like (EM) algorithm. This algorithm simulates the electromagnetism theory of physics by considering each point in the population associated with an electrical charge. The method uses an attraction-repulsion mechanism to move a population of points towards optimality. Here, we are concerned with improving EM efficiency and solution accuracy. To this purpose, we suggest some modifications to the computation of the charge associated to each point so that the attraction-repulsion technique can accelerate convergence. The proposed modifications are extensively experimented on a well-known benchmark problems set. Performance profile plots, as outline in (Dolan and Moré, 2002), have been made to assess the best objective function behaviour, as well as the mean absolute error, of the new algorithm when compared with the original EM algorithm.

The paper is organized as follows. In Section 2 we describe the original EM algorithm and justify the adopted charge calculation modifications. Section 3 contains the results of the carried out numerical experiments including a comparison with other heuristics. We conclude the paper in Section 4.

2. The electromagnetism-like mechanism

To introduce the charge modifications, we first describe the original EM algorithm. The following notation is used. The population size is represented by m, $x^i \in \Re^n$ denotes the ith point in the population, $x^i_k \in \Re$ is the kth coordinate of the point x^i ($k=1,\ldots,n$), and the best point in the population, the point with least objective function value, is represented by x^{best}. The corresponding function value is denoted by f_{best}.

* Corresponding author. E-mail: arocha@dps.uminho.pt

2.1. Original electromagnetism algorithm

The EM algorithm consists of four main procedures: *Initialize*, *Compute F*, *Move* and *Local Search*. A brief explanation follows.

Initialize aims to randomly generate the m points of the population inside the feasible region. Each coordinate of a point is assumed to be uniformly distributed between the corresponding upper and lower bounds. The objective function values of the points in the population are computed and the best point is identified.

In the procedure *Compute F*, the total force exerted on each point via other points is computed. According to the Coulomb's law, the force between any pair of points x^i and x^j is inversely proportional to the square of the distance between the points and directly proportional to the product of their charges. The charge q^i of the point x^i is computed according to the objective function value and determines the power of attraction or repulsion for that point. In the original EM algorithm (Birbil and Fang, 2003; Birbil *et al.*, 2004), the charge of a point is computed by

$$q^i = \exp\left(-n(f(x^i) - f_{best}) / \sum_{j=1}^{m}(f(x^j) - f_{best})\right), \quad i = 1,...,m, \tag{2}$$

where points with better function values have higher charges. This is a scaled distance of the function value $f(x^i)$ to the function value of the best point in the population. In this case charges are positive and vary from 0 to 1. Each individual component force F^i_j, between any pair of points x^i and x^j, depends on the objective function values at x^i and x^j. Since charges (2) are positive, the direction of the force F^i_j is $\overrightarrow{x^i x^j}$ if $f(x^j) < f(x^i)$, meaning that x^j attracts x^i, and is $\overrightarrow{x^j x^i}$ if $f(x^j) \geq f(x^i)$ and x^j repels x^i. Finally, the forces exerted on point x^i by all other points, x^j, $j \neq i \in \{1,...,m\}$, are combined by means of vector summation, to give the total force vector F^i exerted on that point x^i,

$$F^i = \sum_{j \neq i}^{m} F^i_j = \begin{cases} (x^j - x^i) q^i q^j / (\| x^j - x^i \|^3), & \text{if } f(x^j) < f(x^i) \\ (x^i - x^j) q^i q^j / (\| x^j - x^i \|^3), & \text{if } f(x^j) \geq f(x^i) \end{cases}, \; i = 1,...,m.$$

The procedure *Move* uses the total force vector, F^i, to move the point x^i in the direction of the force by a random step length λ, see (3). The best point, x^{best}, is not moved and is carried out to the subsequent iteration. To maintain feasibility, the force exerted on each point is normalized and scaled by the allowed range of movement towards the lower or the upper bounds. Thus, for each coordinate, $k=1,...,n$,

$$x^i_k = x^i_k + \begin{cases} \lambda(F^i_k / \| F^i \|)(u_k - x^i_k), & \text{if } F^i_k > 0 \\ \lambda(F^i_k / \| F^i \|)(x^i_k - l_k), & \text{otherwise} \end{cases}, \text{ for } i = 1,...,m \text{ and } i \neq best \tag{3}$$

and the random step length λ is assumed to be uniformly distributed between 0 and 1.

The procedure *Local Search* performs a local refinement and can be applied to one point or to all points in the population. The algorithm presented in (Birbil and Fang, 2003) is a random line search that is applied coordinate by coordinate only to the best point in the population. First, based on the parameter, δ, the procedure computes $s_{max} = \delta \, max_k (u_k - l_k)$, which is the maximum feasible step length, to guarantee that the local search always generates feasible points. Second, the best point is assigned to a temporary point y to store the initial information. Next, for each coordinate k, a random number λ between 0 and 1 is selected as a step length and y_k is moved along that direction, $y_k = x_k^{best} + \lambda \, s_{max}$. If an improvement is observed, within n_{it}^{max} iterations, the best point is replaced by y and the search along that coordinate ends.

2.2. Different point charge calculations

Formula (2) is not the only one in literature to compute charges associated with points in a EM algorithm context. An alternative has been adopted in hybridization of recent heuristics, the scatter search algorithm (Debels *et al.*, 2006) and the differential evolution algorithm (Kaelo and Ali, 2007). The approach therein adopted assumes that each point has another point exerting force on it. So, the charges are not computed independently but depend on the point they exert force on, *i.e.*, $q^{i,j}$ depends on the relative difference of the objective function values at x^i and x^j. In this case, each charge $q^{i,j} \in [-1,1]$ and when charge is positive, x^j attracts x^i, while x^j repels x^i when the charge is negative. Our approach however is similar to that of the original EM algorithm. Each point of a population will have $m-1$ points exerting force on it. Thus the charge q^i of each point x^i is based on the (scaled) distance of its function value to the function value of the best point in the population, but, in our case, this scaling is relative to the spread of values, *i.e.*, the difference between the worst (f_{worst}) and the best function values in the population,

$$q^i = \exp\left(-n(f(x^i) - f_{best})/(f_{worst} - f_{best})\right), \quad i = 1,...,m. \tag{4}$$

This formula also defines charges in the set (0,1] and points with better function values also have higher charges (close to 1). Although the scaling factor here is quite different from the one in (2), at the end the resulting charges have similar behaviour due to the "exp" function. Like formula (2), this new proposal depends on the problem dimension. The immediate result is that the charges of points different from the best point approach zero more rapidly as n increases.

Formula (5) also provides nonnegative charge values that increase from 0 to 1 as the points x^i have function values closer to f_{best}. However, in this case, the decrease of the charges to zero, as the function values go far away from f_{best} is slower than that of formula (4),

$$q^i = 1/\left(n(f(x^i) - f_{best})/(f_{worst} - f_{best}) + 1\right), \quad i = 1,...,m. \tag{5}$$

In both proposed formulae, the charges depend on the problem dimension. However, it is not clear that this dependency is crucial to the algorithm performance. In the next section, we carry out an extensive experimental procedure to compare the new charge computation formulae with the original one and analyze the n dependency effect on algorithm performance. This comparative study relies on the best objective function value and on the mean absolute error of the mean best solutions.

3. Numerical experiments

In this section, we report the numerical results obtained by running the original EM algorithm – q^i from (2) – and the modified with the two proposals for the charges – (4) and (5) – on a set of global optimization problems with box constraints. We use a set of 64 benchmark global optimization test problems, produced in full detail in the Appendix B of (Ali *et al*., 2005). As problem dimensions vary from 2 to 30 we set the population size dependent on the problem dimension: $m = \min\{200, 10n\}$. The used algorithm constants are $\delta = 0.001$ and $n_{it}^{max} = 10$.

3.1. Performance assessment by Dolan & Moré´s profiles

To evaluate and compare the performance of the two charge proposals within the electromagnetism-like algorithm, performance profiles as proposed in (Dolan and Moré, 2002) are used. The performance profiles give, for every value of $\tau > 1$, the proportion $\rho(\tau)$ of test problems on which each algorithm under comparison has a performance within a factor τ of the best.

The performance profile plot represents the cumulative distribution function of a performance ratio that is computed from an appropriate metric, for example the computing time required to solve a problem. A brief discussion of our implementation of this performance assessment follows. Assume that P is the set of all problems, S is the set of solvers – algorithm implementation codes – used in the comparative study, and $m(p,s)$ is the chosen performance metric, found by solver $s \in S$ on problem $p \in P$ after a fixed number of function evaluations. Then, the performance ratio is defined by

$$r(p,s) = \begin{cases} 1 + m(p,s) - \min\{m(p,s) : s \in S\}, & \text{if} \quad \min\{m(p,s) : s \in S\} < 0.000001 \\ m(p,s)/\min\{m(p,s) : s \in S\}, & \text{otherwise} \end{cases},$$

and the cumulative distribution function of the performance ratio is represented by $\rho(s;\tau) = |P_\tau| \,/\, |P|$, where $|.|$ denotes the cardinality of a set, and $P_\tau = \{p \in P: r(p,s) \le \tau\}$. Thus, $\rho(s;\tau)$ gives the probability (for solver s) that the performance ratio is within a factor τ of the best possible ratio. To evaluate and compare the best behaviour of the solvers, the following performance metric, which gives a scaled distance of the best value computed by solver s on problem p, f_{best}, to the known optimal function value f^*, is used

$$m(p,s) = (f_{best} - f^*)/(f_{worst(s)} - f^*), \tag{6}$$

where $f_{worst(s)}$ is the worst function value computed among all solvers on problem p ($f_{worst(s)}$ and f_{best} are evaluated after running each problem 30 times and $f_{best} = \min\{f^i_{best},\ i=1,\ldots,30\}$). Figure 1 contains the plots of the performance profiles after $100n^2$ function evaluations, over the 30 runs, for the q^i formulae previously presented, see (2), (4) and (5). Formulae (4) and (5) were also tested without using n (n=1). Formula (5) with dependency on n is the most efficient for values of τ inferior to approximately 7, and formula (4) with dependency on n gives the best results for larger values of τ. It solves all problems to optimality, meaning that, according to the metric (6), the best computed function value has a high accuracy. Further, with this comparative study we are able to show that the dependency on n of formulae (4)

and (5) is important since it provides better results (see Figure 1). To analyze the performance of the algorithm convergence, the most appropriate metric is the mean absolute error, defined by $MAE = (|f_{avg} - f^*|) / n$, since is gives the distance of the mean best function values over the 30 runs, $f_{avg} = (\sum_{i=1}^{30} f_{best}^i)/30$, to the optimal value. Figure 2 plots the performance profiles of the five cases in comparison. We note that formula (4), to compute the charges q^i, wins over the others. The version that depends on n is the best for small values of τ while the other, with n=1, is the best for the remaining values ($\tau > 9$).

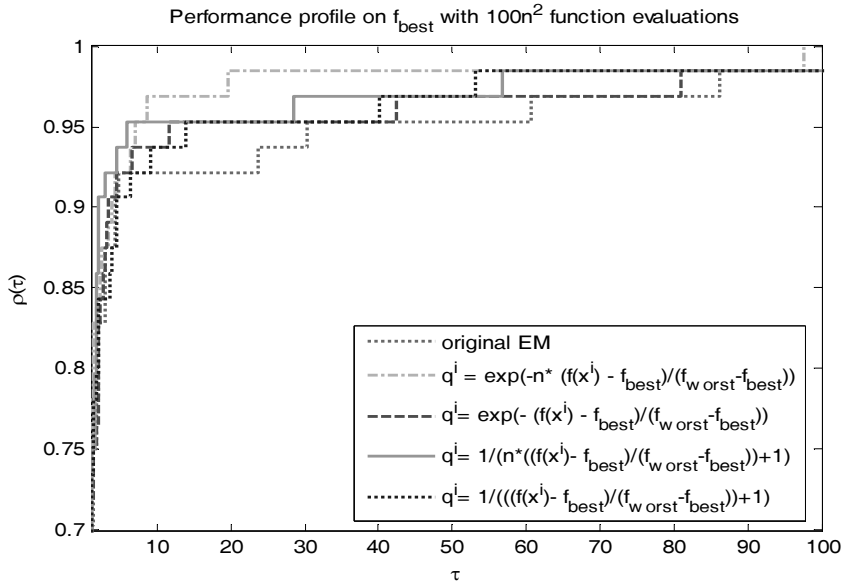


Fig. 1. Performance profiles on f_{best}, for (2), (4) and (5), after $100n^2$ function evaluations

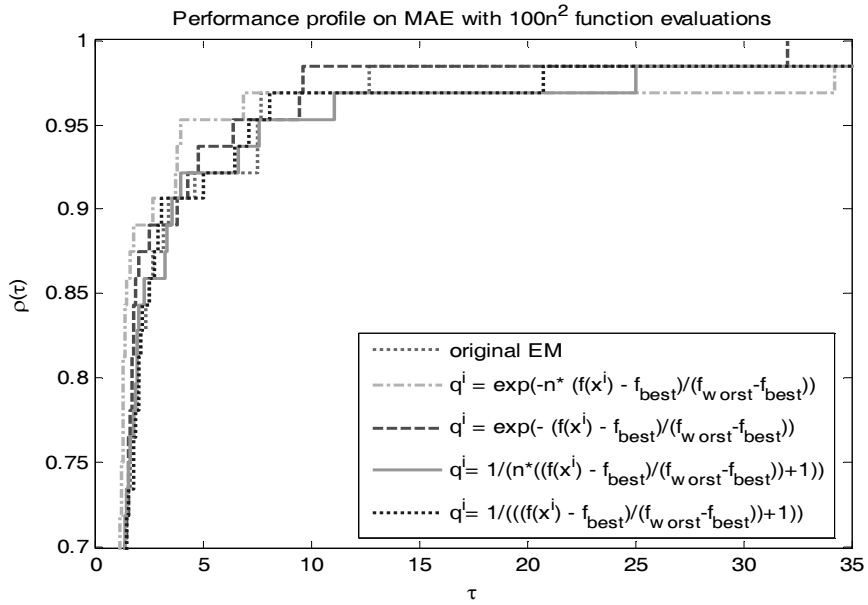
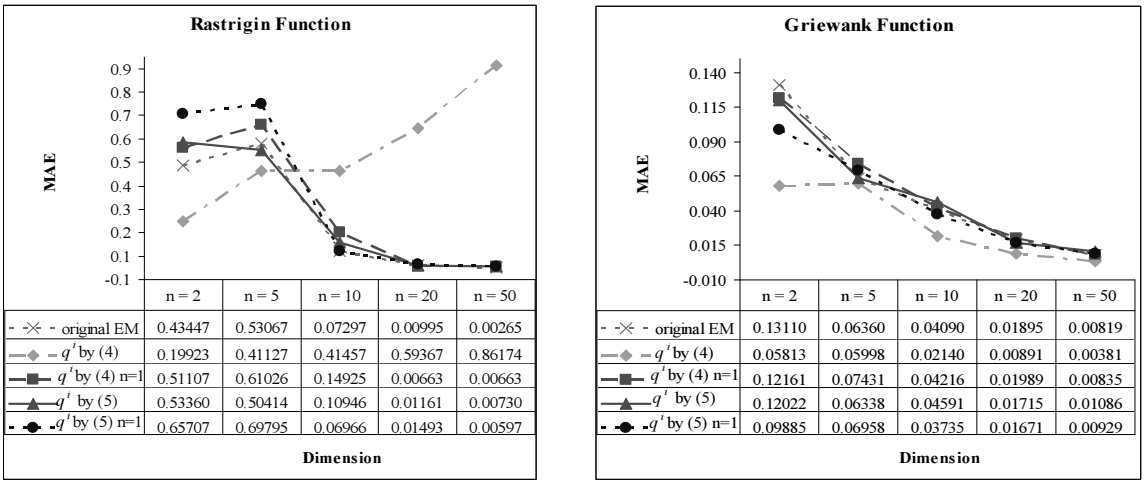


Fig. 2. Performance profiles on *MAE* for (2), (4) and (5), after $100n^2$ function evaluations

3.2. Behaviour of the modified EM algorithm for different problem dimensions

Here we pick two problems and analyze the error behaviour, with respect to f^*, of the modified EM algorithm as problem dimension increases. The measurement of error is based on *MAE* and the chosen problems are: Rastrigin and Griewank with 5 values of n (2, 5, 10, 20, 50) (Ali *et al*., 2005). Figure 3 illustrates the results of applying formulae (2), (4) and (5) to the 5 versions of each problem. Except in one case, the error does not increase as n increases, meaning that the tested versions of the EM algorithm are suitable for high dimension problems.



Rastrigin Function	n = 2	n = 5	n = 10	n = 20	n = 50
original EM	0.43447	0.53067	0.07297	0.00995	0.00265
q^i by (4)	0.19923	0.41127	0.41457	0.59367	0.86174
q^i by (4) n=1	0.51107	0.61026	0.14925	0.00663	0.00663
q^i by (5)	0.53360	0.50414	0.10946	0.01161	0.00730
q^i by (5) n=1	0.65707	0.69795	0.06966	0.01493	0.00597

Dimension

Griewank Function	n = 2	n = 5	n = 10	n = 20	n = 50
original EM	0.13110	0.06360	0.04090	0.01895	0.00819
q^i by (4)	0.05813	0.05998	0.02140	0.00891	0.00381
q^i by (4) n=1	0.12161	0.07431	0.04216	0.01989	0.00835
q^i by (5)	0.12022	0.06338	0.04591	0.01715	0.01086
q^i by (5) n=1	0.09885	0.06958	0.03735	0.01671	0.00929

Dimension

Fig. 3. Comparison of error behaviour using ***MAE*** for 5 values of ***n***

3.3. Comparison with other stochastic population-based algorithms

To compare the performance of the modified EM algorithm, based on (4) and (5) to compute the charges, with other stochastic population-based algorithms – particle swarm optimization (PSO), evolutionary algorithm (EA) and differential evolution (DE) –, five well-known problems for benchmarking purposes of optimization algorithms are used (Krink *et al*., 2004): Schaffer F6 (n = 2), Sphere (n = 5), Griewank (n = 50), Rastrigin (n = 50) and Rosenbrock (n = 50). We replicate the conditions therein presented. Each problem is run 30 times and the number of function evaluations is kept constant for a fair comparison with those results: 100000 for problems Schaffer F6 and Sphere and 500000 for the other three. We refer to (Krink *et al.*, 2004) for details concerning the other algorithm parameters. For example, the population sizes therein reported are: m = 50 in DE, m = 100 in EA, m = 20 in PSO. In our modified EM algorithm m = 50 is used. Table 1 reports the mean best function values (f_{avg}) and the standard error (± *se*) of the results, over the 30 runs. These two quantities represent the performance of the convergence and the stability of the algorithm respectively. Values below 1e-12 are reported as 0 in the table. From the results, we may conclude that DE is indeed the best algorithm in comparison. The modified EM algorithm outperforms PSO in the optimization of 4 problems and outperforms EA in 2 problems. Comparing the four versions of the modified EM algorithm, formula (4) with dependency on n gives slightly better results.

Table 1. Mean ± *se* of results

	Griewank	Rastrigin	Rosenbrock	Schaffer F6	Sphere
q^i by (4)	0.25732±0.13119	6.667e-7±7.581e-7	47.58370±21.78979	0.00777±0.00395	0±0
q^i by (4), n=1	0.63627±0.40296	0.43115±0.89311	42.58174±22.75698	0.00874±0.00297	3.333e-8±1.826e-7
q^i by (5)	0.54585±0.44669	0.23216±0.50147	51.07712±23.75423	0.00842±0.00336	0±0
q^i by (5), n=1	0.63993±0.38538	0.16583±0.45881	57.228570±31.68917	0.00812±0.00362	0±0
PSO	1.54900±0.06695	13.1162±1.44815	5142.45±2929.47	0.00453±0.00090	2.511e-8±0
EA	0.00624±0.00138	32.6679±1.94017	79.8180±10.4477	0±0	0±0
DE	0±0	0±0	35.3176±0.27444	0±0	0±0

Finally, we compare the modified EM algorithm, based on formula (4), with the DE algorithm that relies on two types of hybrid mutation – integrated (DEIM) and adapted (DEAM) – as outline in (Kaelo and Ali, 2007), as well as with the original EM (Birbil and Fang, 2003). In the comparison, we use the same set of 9 problems taken from (Dixon and Szegö, 1978), with population sizes (m), maximum number of iterations (N_{it}^{max}) and stopping criterion similar to those used in (Kaelo and Ali, 2007). Table 2 contains the average number of function evaluations computed over the 25 runs. The modified EM algorithm

is better than DEIM and DEAM in 6 problems and better than EM in 3 problems. Nonetheless, our modified EM algorithm wins over the others in terms of the total average number of function evaluations (10978 against 14063, 14566 and 15551).

Table 2. Average number of function evaluations

	Shekel5	Shekel7	Shekel10	Hartman3	Hartman6	Goldprice	Branin	Six H C	Shubert
m	40	40	40	30	30	20	20	20	20
$\mathbf{N}_{it}^{max}$	150	150	150	75	75	50	50	50	50
q^i by (4)	1879	755	2242	1139	2851	430	339	239	1104
DEIM	3662	2716	2656	634	1547	577	814	441	1016
DEAM	3560	2709	3037	631	1801	594	778	442	1014
EM	3368	1782	5620	1114	2341	420	315	233	358

4. Conclusions

We have made a preliminary study on the effect of charges to the performance of the electromagnetism-like algorithm for solving global optimization problems like (1). The results confirm that the proposed formula (4), based on the "exp" function, is the best since the corresponding algorithm is proven to be robust, with a high level of accuracy. The behaviour of the modified EM algorithm for different problem size dimensions has also been analyzed. A comparison with other population-based heuristics shows that the modified EM algorithm is competitive. Further research will consider the use of a greedy selection mechanism within the EM algorithm to accelerate convergence to the solution and the extension to equality and inequality constrained optimization problems.

References

Ali, M. M.; Khompatraporn, C. and Zabinsky, Z. B. 2005. A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems, *Journal of Global Optimization* 31: 635–672.

Birbil, S. I. and Fang, S. 2003. An electromagnetism-like mechanism for global optimization, *Journal of Global Optimization* 25: 263–282.

Birbil, S. I.; Fang, S. and Sheu, R. 2004. On the convergence of a population-based global optimization algorithm, *Journal of Global Optimization* 30: 301–318.

Debels, D.; DeReyck, B.; Leus, R. and Vanhoucke, M. 2006. A hybrid scatter search/electromagnetism metaheuristic for project scheduling, *European Journal of Operational Research* 169; 638–653.

Dixon, L. C. W. and Szegö, G. P. 1978. The global optimization problem: an introduction, in *Towards Global Optimization* 2: 1–15, North-Holland.

Dolan, E. D. and Moré, J. J. 2002. Benchmarking optimization software with performance profiles, *Mathematical Programming* 91: 201–213.

Kaelo, P. and Ali, M. M. 2007. Differential evolution algorithms using hybrid mutation, *Computational Optimization and Applications* 37: 231–246.

Krink, T.; Filipic, B.; Fogel G. B. and Thomsen, R. 2004. Noisy optimization problems – a particular challenge for differential evolution? in *Proceedings of 2004 Congress on Evolutionary Computation*, IEEE Press, 332–339.