

Performance Profile Assessment of Electromagnetism-like Algorithms for Global Optimization

Ana Maria A.C. Rocha and Edite M.G.P. Fernandes

Department of Production and Systems, University of Minho, 4710-057 Braga, Portugal

Abstract. This paper introduces a modification on the movement force vector of the Birbil and Fang's electromagnetism-like algorithm [1] for solving global optimization problems with bounded variables. The proposed movement vector combines the total force exerted on each point of the population, at the current iteration, with the rate of change in the force vector of a previous iteration. Several widely used benchmark problems were solved to compare the proposed modification with the original algorithm. A performance profile assessment shows the efficiency and robustness of the proposed modification.

Keywords: Global optimization, electromagnetism-like algorithm, movement force vector, performance profile

PACS: 02.60.Pn

INTRODUCTION

In this paper, we analyse the performance of a modification introduced in the movement force vector of the electromagnetism-like (EM) algorithm to compute a global solution of a nonlinear optimization problem. The algorithm as outline in [1], simulates the electromagnetism theory of physics by considering each point in a population as an electrical charge. The EM algorithm starts with a population of randomly generated points from the feasible region. The method uses an attraction-repulsion mechanism to move a population of points towards optimality. The charge of each point is related to the objective function value and determines the magnitude of attraction of the point over the population. The better the objective function value, the higher the magnitude of attraction. Each point is moved according to a vector that reflects the objective function value of the corresponding point relative to the other points in the population. Here, we denote the vector that is used to move each point in the population by movement force vector. The general scheme of the EM algorithm is presented below. We refer to [1, 2] for details.

Algorithm 1 (original EM algorithm)

Randomly generate a population of points in the feasible region

iteration \leftarrow 1

while termination criterion is not satisfied **do**

 Compute total force $F_{\text{iteration}}$ exerted on each point by other points in the population

 Move points, except the best one, according to $F_{\text{iteration}}$

 Locally explore the best point of the population

 iteration \leftarrow iteration + 1

end while

To improve the algorithm search ability and efficiency, a modification on the movement force vector is introduced in the original EM algorithm. This vector combines the total force exerted on the point, at the current iteration, with the rate of change in the force vector of a previous iteration. The proposed algorithm modification has been extensively experimented on a well-known benchmark problems set [3]. Performance profile plots, as outline in [4], have been made to assess the best behavior of the proposed modifications on the movement force vector when compared with the original EM algorithm.

The class of problems that we consider in this comparative study has the following form:

$$\text{minimize } f(x) \text{ subject to } x \in \Omega, \quad (1)$$

where $f: R^n \rightarrow R$ is a nonlinear function and $\Omega = \{x \in R^n : -\infty < l_k \leq x_k \leq u_k < \infty, k = 1, \dots, n\}$ is a bounded feasible region. We assume that the objective function f is not convex and possesses many local minima in the set Ω . This class of global optimization problems is very important in engineering applications.

DETAILS OF THE ALGORITHM

In the paper, $x^i \in R^n$ denotes the i th point of the population. At each iteration of the original EM algorithm, the total force vector $F_{\text{iteration}}^i$ exerted on each point x^i is calculated by adding the individual component forces, F_j^i , between any pair of points x^i and x^j . According to the electromagnetism theory, each force F_j^i is inversely proportional to the square of the distance between the points and directly proportional to the product of their charges. The charges of the points are computed according to their objective function values.

Each point x^i is then moved in the direction of a force, herein simply denoted by F^i , by a random step length and in a way that feasibility is maintained (see [1] for details). Thus, for each point x^i in the population, except the best one,

$$x_k^i = \begin{cases} x_k^i + \lambda \frac{F_k^i}{\|F^i\|} (u_k - x_k^i) & \text{if } F_k^i > 0 \\ x_k^i + \lambda \frac{F_k^i}{\|F^i\|} (x_k^i - l_k) & \text{otherwise} \end{cases}, \quad k = 1, 2, \dots, n. \quad (2)$$

The random step length λ is assumed to be uniformly distributed between 0 and 1. In the original EM algorithm, each force F^i is the total force exerted on the point computed at the current iteration, thus $F^i = F_{\text{iteration}}^i$.

A particle swarm optimization (PSO) algorithm, like the EM algorithm, is an easy to implement and computationally inexpensive population-based technique, that has been widely used to solve problem (1). Recently, Liu and Lin proposed in [5] a modification to the PSO algorithm that detects the velocity changes for each particle by including a delta momentum rule in the velocity updating equation.

Based on this idea, we propose a modification to the movement force vector, in the EM algorithm, so that past information on force vectors can be used to improve efficiency. Since $F_{\text{iteration}}^i$ represents the total force exerted on the point x^i , at the current time step iteration, the vector used to move the point may include a term for the rate of change in force. If we define the change in force, at a particular iteration I , by $\Delta F_I^i = F_I^i - F_{I-1}^i$ then the proposed movement vector can be written as

$$F^i = F_{\text{iteration}}^i + \beta \Delta F_I^i \quad (3)$$

where $0 \leq \beta < 1$ is a memory constant that controls the rate of change in force vector. Note that when $\beta = 0$, (3) resumes to the original EM. If information on the force of just one previous time step is to be considered then $I = \text{iteration}$. However, if information on two previous time steps are to be integrated in the movement vector then $I = \text{iteration} - 1$ is set in (3).

NUMERICAL EXPERIMENTS AND CONCLUSIONS

In this section, we report the numerical results obtained by running the modified EM algorithm on a collection of 50 benchmark global optimization test problems. Appendix B in [3] presents the problems in full detail. In ten problems, two values of the dimension (n) are tested, and in one, five values of n were tried, giving the final number of 64 problems. We compare the herein proposed modification on the movement force, as outline in the previous section - with both time steps - with the original EM algorithm, described in Algorithm 1.

In our implementation, the algorithms terminate when the number of objective function evaluations exceeds $100n^2$. The study also sets the number of points in the population to $\min\{200, 10n\}$. The EM algorithms were repeated for 30 runs for each problem. The following comparative study sets the memory constant β to 0.1 based on the parameter sensitivity analysis presented later on in the paper.

This comparison is based on the Dolan and Moré's performance profile approach as outline in [4]. The performance profiles give, for every $\tau \geq 1$, the proportion $\rho(\tau)$ of test problems on which each algorithm under comparison has a performance within a factor τ of the best. An explanation of our implementation of this approach follows.

Let \mathcal{P} be the set of all problems and \mathcal{S} the set of solvers used in the comparative study. Let $m_{(p,s)}$ be the performance metric found by solver $s \in \mathcal{S}$ on problem $p \in \mathcal{P}$ after a fixed number of function evaluations. Here, we use a metric that measures the relative improvement of the function values, a scaled distance to the known optimal function value f_{opt} , defined in [3] by $m_{(p,s)} = (f_{best(p,s)} - f_{opt}) / (f_{worst} - f_{opt})$, where f_{worst} denotes the worst function value found among all solvers on the problem p and $f_{best(p,s)}$ is the best function value found by solver s on problem p , after $100n^2$ function evaluations and over 30 runs.

Then, the overall assessment of the performance of a particular solver s is defined by $\rho_s(\tau) = \frac{1}{n_p} \text{size}\{p \in \mathcal{P} : r_{(p,s)} \leq \tau\}$, where n_p is the number of problems in the set \mathcal{P} and "size" is the number of problems in the set such that

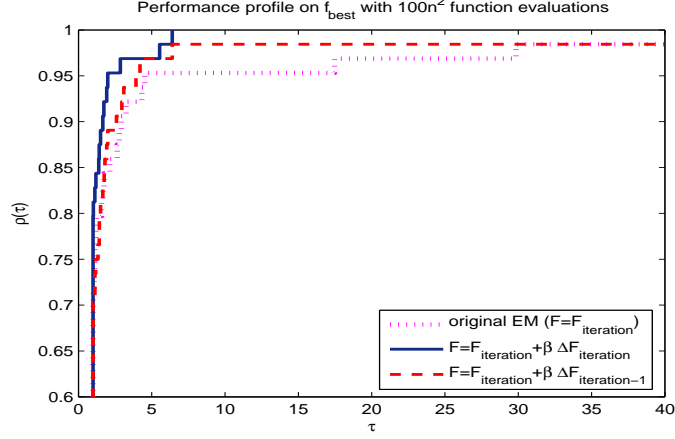


FIGURE 1. Performance profile on f_{best} with $100n^2$ function evaluations ($\beta = 0.1$).

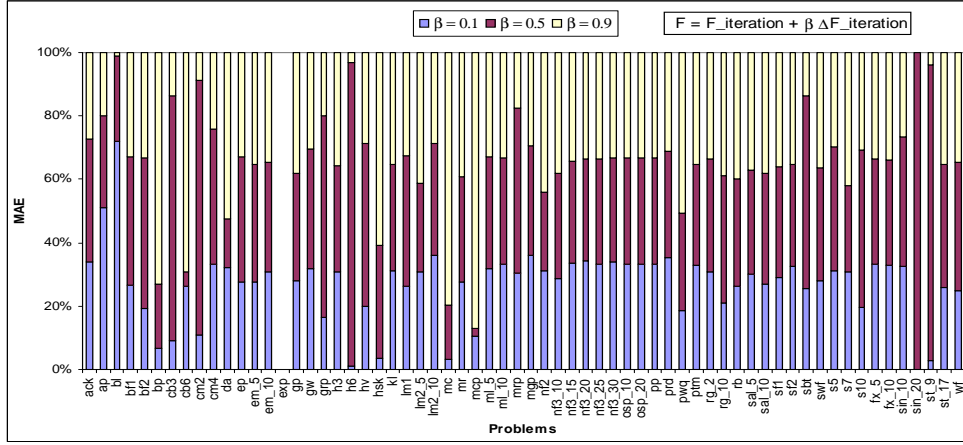


FIGURE 2. MAE comparison for three values of β

the performance ratio $r_{(p,s)}$ is less than or equal to τ for solver s . Hence, $\rho_s(\tau)$ is the probability (for solver $s \in \mathcal{S}$) that the performance ratio $r_{(p,s)}$ is within a factor $\tau \in R$ of the best possible ratio. The performance ratio adopted for this comparative study is

$$r_{(p,s)} = \begin{cases} 1 + m_{(p,s)} - \min\{m_{(p,s)} : s \in \mathcal{S}\}, & \text{if } \min\{m_{(p,s)} : s \in \mathcal{S}\} < \varepsilon \\ \frac{m_{(p,s)}}{\min\{m_{(p,s)} : s \in \mathcal{S}\}}, & \text{otherwise} \end{cases},$$

since $\min\{m_{(p,s)} : s \in \mathcal{S}\}$ may eventually be zero, for $p \in \mathcal{P}, s \in \mathcal{S}$ and $\varepsilon = 0.00001$. We refer to [6] for details.

Using the performance profile plot one can compare how well a solver can estimate the optimum relative to the others. The value of $\rho_s(1)$ gives the probability that the solver s will win over the others in the set. However, for large values of τ , the $\rho_s(\tau)$ measures the solver robustness.

Figure 1 contains the performance profiles on f_{best} for the three tested movement force vectors formulae. It shows that the modification introduced in the movement force that relies on the force variation $\Delta F_{iteration}^i$ performs better than the other two in comparison. It is also the most robust solving all problems to optimality. The figure also shows that the other two fail on approximately 1% of the problems for small values of τ .

To access the accuracy of the solutions found by an algorithm, the mean absolute error, MAE, defined by $(|f_{opt} - f_{avg}|)/n$ is usually used [5], where f_{avg} is the average best function values obtained after $100n^2$ function evaluations over a certain number of runs. Here, we use MAE to determine the effect of the memory constant on the algorithm efficiency. This analysis is focused only on the algorithm that performed better in the previous study. Three values of

TABLE 1. Numerical results obtained by Modified EM, GA, PSO and PS-EA

Problem	n	Modified EM		GA		PSO		PS-EA	
		f_{avg}	SD	f_{avg}	SD	f_{avg}	SD	f_{avg}	SD
Ackley	10	0.001	0.000	0.593	0.225	0.000	0.000	0.192	0.195
	30	0.000	0.000	1.099	0.250	0.000	0.000	0.377	0.099
Griewank	10	0.196	0.134	0.050	0.030	0.079	0.033	0.222	0.078
	30	0.105	0.046	1.234	0.110	0.011	0.014	0.821	0.140
Rastrigin	10	0.498	1.003	1.393	0.763	2.656	1.390	0.434	0.255
	30	0.332	0.708	10.439	2.639	32.476	6.995	3.053	0.999
Rosenbrock	10	4.859	0.717	46.318	33.822	4.371	2.381	25.303	29.796
	30	22.758	0.637	166.283	59.510	402.540	633.650	98.407	35.579
Schwefel	10	196.655	143.334	1.952	1.304	161.870	144.160	0.320	1.619
	30	4534.698	627.143	13.535	4.953	990.770	581.140	3.272	1.619

β were tested: 0.1, 0.5 and 0.9. Figure 2 is a 100% stacked column chart for the MAE values obtained by the three β values. For each problem, we can compare the percentage that each β value contributes to the total. Hence, the smaller the MAE (smaller the percentage) the better. Overall, the area of the bars corresponding to $\beta = 0.1$ is smaller than the others - 27% in contrast with 39% and 34% of the total - which indicates that 0.1 is the most efficient choice for β . It seems that larger values drive the points to new areas, instead of exploring locally, and prevent convergence to the solution. In the problem "exp", MAE is zero for the tested β values. A similar analysis carried out with the force variation $\Delta F_{iteration-1}^i$ generated the areas 27%, 35% and 38% of the total, for the β values 0.1, 0.5 and 0.9 respectively.

To compare the performance of our most efficient implementation with three other algorithms, the Genetic Algorithm (GA), the Particle Swarm Optimization (PSO) and a hybrid model of Evolutionary Algorithm and PSO (PS-EA), we use five benchmark problems: Ackley, Griewank, Rastrigin, Rosenbrock and Schwefel, with global function values 0. To be able to compare with the published results [7] for GA, PSO and PS-EA, we repeat the experiments using the same parameters therein used: population size is set to 125 and the maximum number of iterations is 500 when $n = 10$, and 1000 when $n = 30$. Each experiment is repeated 30 times with different random seeds. Table 1 contains the f_{avg} and SD (unbiased standard deviation of the computed function values). The modified EM algorithm outperforms GA in 6, out of 10 cases; outperforms PSO in 3 and is comparable in 4; and outperforms PS-EA in 6 problems.

This work proposes a modification on the movement force vector of the electromagnetism-like mechanism, including past force changes, for solving global optimization problems like (1), which improves on Birbil and Fang's EM algorithm. The modification is tested with a set of problems and compared using the Dolan and Moré's performance profiles. A sensitivity study is conducted to identify a good choice for the memory constant β . A performance assessment of the proposed algorithm when compared with other algorithms (GA, PSO and PS-EA) is also included. Future work will involve the extension of the proposed algorithm to constrained optimization problems.

ACKNOWLEDGMENTS

The authors are grateful to the two anonymous referees for their valuable suggestions.

REFERENCES

1. S.I. Birbil and S. Fang, *An electromagnetism-like mechanism for global optimization*, Journal of Global Optimization, 25, 2003, 263–282.
2. S.I. Birbil, S. Fang and R. Sheu, *On the convergence of a population-based global optimization algorithm*, Journal of Global Optimization, 30, 2004, 301–318.
3. M.M. Ali, C. Khompatraporn and Z.B. Zabinsky, *A numerical evaluation of several stochastic algorithms on selected continuous global optimization test problems*, Journal of Global Optimization, 31, 2005, 635–672.
4. E.D. Dolan and J.J. Moré, *Benchmarking optimization software with performance profiles*, Mathematical Programming, 91, 2002, 201–213.
5. J. Liu and J. Lin, *Evolutionary computation of unconstrained and constrained problems using a novel momentum-type particle swarm optimization*, Engineering Optimization, 39, 2007, 287–305.
6. A.I.F. Vaz and L.N. Vicente, *A particle swarm pattern search method for bound constrained global optimization*, Journal of Global Optimization, 39, 2007, 197–219.
7. D. Srinivasan and T.H. Seow, *Particle Swarm Inspired Evolutionary Algorithm (PS-EA) for Multiobjective Optimization Problems*, CEC' 03, 4 (2003), pp. 2292–2297.