A new Electromagnetism-like Algorithm with a Population Shrinking Strategy

ANA MARIA A. C. ROCHA and EDITE M.G.P. FERNANDES
Minho University
Department of Production and Systems
Campus de Gualtar, 4710-057 Braga
PORTUGAL

Abstract: The Electromagnetism-like (EM) algorithm, developed by Birbil and Fang [3] is a population-based stochastic global optimization algorithm that uses an attraction-repulsion mechanism to move sample points toward optimality. In order to improve the accuracy of the solutions the EM algorithm incorporates a random local search. In this paper we propose a new local search procedure based on a pattern search method, and a population shrinking strategy to improve efficiency. The proposed method is applied to some test problems and compared with the original EM algorithm.

Key–Words: Global optimization, Electromagnetism-like algorithm, Pattern search method, Population shrinking

1 Introduction

We consider the problem of finding a global solution of the problem:

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in \Omega, \end{array} \tag{1}$$

where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ is a nonlinear function and $\Omega = \{x \in \mathbb{R}^n | \infty < l_k \leq x_k \leq u_k < \infty, k = 1, \ldots, n\}$ is a bounded feasible region. Further, we assume that the derivatives are not available. Recently, Birbil and Fang proposed the electromagnetism-like (EM) algorithm that is a population-based stochastic search method for global optimization [3]. The algorithm simulates the electromagnetism theory of physics by considering each sampled point as an electrical charge. The method utilizes an attraction-repulsion mechanism to move a population of points toward optimality. In order to improve the accuracy of the solutions a typical EM algorithm incorporates a random local search. Here, we propose a modification to the EM algorithm using the original pattern search method of Hooke and Jeeves [5], that is simple to implement and does not require any derivative information. Another modification consists of incorporating a population shrinking strategy into the EM algorithm.

Thus, in order to solve problem (1), we modify the EM algorithm proposed in [3] twofold: (i) a pattern search method is used to provide at each iteration a local search about the best point of the population; (ii) a population shrinking strategy is implemented to reduce the number of points in the population whenever the concentration of all points around the best point is considered acceptable.

Four sets of experiments are carried out to illustrate the efficiency of the local pattern search method and the population shrinking strategy separately.

The remainder of the paper is organized as follows. Section 2 briefly describes the original EM algorithm and Section 3 introduces the pattern search method to be used as a local search. Section 4 is devoted to explain the main ideas of the new shrinking population strategy and Section 5 reports the numerical results. Some conclusions are drawn in Section 6.

2 Electromagnetism-like algorithm

The EM algorithm starts with a population of randomly generated points from the feasible region. Each point is considered as a charged particle that is released to the space. The charge of each point is related to the objective function value and determines the magnitude of attraction or repulsion of the point over the population. Points with lower objective function values attract others while those with higher function values repel. The charges are used to find a direction for each point to move in subsequent iterations.

Throughout the paper, the following notation is adopted: $x^i \in \mathbb{R}^n$ denotes the i th point of a population; x^{best} is the point that has the least objective function value; $x_k^i \in \mathbb{R}$ is the k th ($k = 1, \ldots, n$) coordinate of the point x^i of the population; m is the number of points in the population; $MaxIt$ is the maximum number of EM iterations; $LSIt$ denotes the maximum number of local search iterations; and δ is a local search parameter, $\delta \in [0, 1]$.

The EM mechanism is schematically shown in Algorithm 1 and relies on four main procedures (*Initialize*, *CalcF*, *Move* and *Local*).

Algorithm 1 (m, $MaxIt$, $LSIt$, δ)
Initialize()
iteration ← 1
while termination criteria are not satisfied **do**
  F ← *CalcF*()
  Move(F)
  Local($LSIt$, δ)
  iteration ← iteration + 1
end while

The procedure *Initialize* aims to randomly generate m points from the feasible region. Each coordinate of a point (x^i_k) $(k = 1, \ldots, n)$ is assumed to be uniformly distributed between the corresponding upper and lower bounds, i.e., $x^i_k = l_k + \lambda(u_k - l_k)$ where $\lambda \sim U(0, 1)$. The objective function values are computed for all the points in the population, and the best point, x^{best}, which is the point with the least function value is identified.

The *CalcF* procedure computes the force exerted on a point *via* other points. First a charge-like value, q^i, that determines the power of attraction or repulsion for the point x^i is determined. The charge of the point is calculated according to the relative efficiency of the objective function value of the corresponding point in the population, i.e.,

$$q^i = \exp\left(-n \frac{f(x^i) - f(x^{best})}{\sum_{k=1}^{m}(f(x^k) - f(x^{best}))}\right),$$

for $i = 1, \ldots, m$. The total force vector F^i exerted on each point is calculated by adding the individual component forces, F^i_j, between any pair of points x^i and x^j,

$$F^i = \sum_{j \neq i}^{m} F^i_j, \quad i = 1, 2, \ldots, m$$

where

$$F^i_j = \begin{cases} (x^j - x^i)\frac{q^i q^j}{\|x^j - x^i\|^2} & \text{if } f(x^j) < f(x^i) \\ (x^i - x^j)\frac{q^i q^j}{\|x^j - x^i\|^2} & \text{if } f(x^i) \leq f(x^j) \end{cases}.$$

The *Move* procedure uses the normalized total force vector exerted on the point x^i, so that feasibility can be maintained, to move it in the direction of the force by a random step length λ, i.e.,

$$x^i = x^i + \lambda \frac{F^i}{\|F^i\|}(RNG),$$

for $i = 1, \ldots, m$ and $i \neq best$, where RNG is a vector with components that define the allowed range of movement toward the lower bound l_k, or the upper bound u_k, for each coordinate k. The random parameter λ is assumed to be uniformly distributed between 0 and 1. Note that the best point, x^{best}, is not moved and is carried to the subsequent iterations.

Finally, the *Local* procedure presented in [3] is a random line search algorithm and is applied coordinate by coordinate to the best point only to explore the neighborhood of that point in the population. First, based on the parameter δ, the procedure computes the maximum feasible step length, $\delta(max_k(u_k - l_k))$. This quantity is used to guarantee that the local search generates only feasible points. Second, for each coordinate k, the best point is assigned to a temporary point y to store the initial information. Next, a random number is selected as a step length and the point y is moved along that direction. If an improvement is observed, within $LSIt$ iterations, the best point is replaced by y and the search along that coordinate k ends. The reader is referred to [2, 3, 4] for details.

3 Pattern search *Local* procedure

Birbil and Fang [3] show that the *Local* procedure is crucial in improving the accuracy of the average function values although at the cost of the number of function evaluations required. Another local search that is simple and does not use derivative information is the pattern search (PS) method, see for example [1, 7, 10]. The research about pattern search methods is flourishing. This method has been first applied to unconstrained optimization and then successfully extended to bound constrained [8], as well as to equality constrained problems [9]. Here, in the EM context, the original Hook and Jeeves pattern search algorithm is applied, at each iteration, to the current best point in the population [5, 10]. This algorithm is based on two moves: the exploratory move and the pattern move.

The exploratory move carries out a coordinate search about the best point, with a step length δ. If at the new point, y, $f(y) < f(x^{best})$, the iteration is successful. Otherwise, the iteration is unsuccessful and δ should be reduced. If the previous iteration was successful, the vector $y - x^{best}$ defines a promising direction and a pattern move is then implemented, meaning that the exploratory move is carried out about the point

$y + (y - x^{best})$ instead of y. Then, if the coordinate search is successful, the returned point is accepted as the new point; otherwise, the pattern move is rejected and the coordinate search is carried out about y. Here, a factor of 0.1 is used to reduce δ when the iteration is unsuccessful and the minimum step length allowed is 1E-08.

Our implementation of the pattern search method uses an exact penalty-like technique to maintain feasibility, i.e., problem (1) is replaced by the following

$$\min F(x) \equiv \begin{cases} f(x) & \text{if } x \in \Omega, \\ \infty & \text{otherwise}, \end{cases}$$

meaning that any generated point that is infeasible is rejected, since the corresponding function value is ∞.

4 A population shrinking strategy

The population shrinking strategy is the main contribution of this paper. The purpose of implementing a devise that is able to strategically reduce the number of points in the population, here denoted by shrinking the population, is to reduce the overall number of objective function evaluations without affecting the accuracy of the results. So, a strategy is designed to shrink the population as the iterative process progresses. The crucial point here is to decide when to shrink the population. When the concentration of all points in the population around the best point is considered acceptable, it seems that some points could be discarded without affecting the convergence rate to the solution. The points that will remain in the population should be the best ones.

One way to measure the concentration of the points in the population around the best point is to compute the standard deviation (SD) of the function values with respect to the best value,

$$\text{SD} = \sqrt{\frac{\sum_{i=1}^{m}(f(x^i) - f(x^{best}))^2}{m}}.$$

This quantity is used to decide when to shrink the population. Although one may think that the SD decreases monotonically as the iterative process converges to the solution, this is not always true. The idea is to shrink the population at a particular iteration l, when the $\text{SD}^{(l)}$ is below 10% of the SD of a reference iteration (SD^{ref}). Here, a reference iteration is the first iteration that was carried out with the same population of the iteration l. Thus, the population is shrunk if

$$deviation\ \ ratio = \frac{\text{SD}^{(l)}}{\text{SD}^{ref}} < 0.1\,.$$

Table 1: Numerical results

Tf	code	m_f	f_{best}	It_{EM}	f_e
GP	orig	-	3.000004	8	256
	orig-shri	5	3.000073	11	179
	ps	-	3.000004	5	306
	ps-shri	5	3.000013	6	350
H	orig	-	0.000038	8	265
	orig-shri	10	0.000042	9	231
	ps	-	0.000068	3	209
	ps-shri	10	0.000068	3	209
R	orig	-	0.008941	500	10533
	orig-shri	5	0.000053	380	2690
	ps	-	0.000685	500	31520
	ps-shri	10	0.000079	72	3678
M	orig-shri	20	-1.913215	9	284
	ps-shri	20	-1.913220	5	317
S	orig-shri	20	0.000002	12	308
	ps-shri	20	0.000001	4	269

Whenever the population is shrunk, the reference iteration changes. Although different shrinking factors have been tested, the constant 0.5 proved to be efficient. The shrinking process as described above is not activated if the number of points in the population is less or equal to $2n$. In the Algorithm 1 context, this shrinking procedure appears after the *Local* procedure.

5 Numerical results

Computational tests were performed on a PC with a 3GHz Pentium IV microprocessor and 1Gb of memory. We compare the performance of the two versions of the EM algorithm, as described in Sections 2 and 3, with the corresponding versions with the proposed shrinking strategy, using 5 well-known test functions selected from the literature. The algorithms terminate when the number of EM iterations exceeds $MaxIt$ or the relative error in the best objective function value, with respect to f_{global} is less than 0.01%.

Two multi-modal functions are selected: Goldstein and Price (GP) with $f_{global} = 3$, and Hump (H) with $f_{global} = 0$. The first has one global solution only and three local solutions. The second one has two global optima. The other 3 tested functions are uni-modal: Rosenbrock2 (R) with $f_{global} = 0$, McCormick (M) with $f_{global} = -1.9133$, and Spherical (S) with $f_{global} = 0$.

In the experiments, we used $m = 20$ (initial population), $LSIt = 10$, $MaxIt = 50$ (except

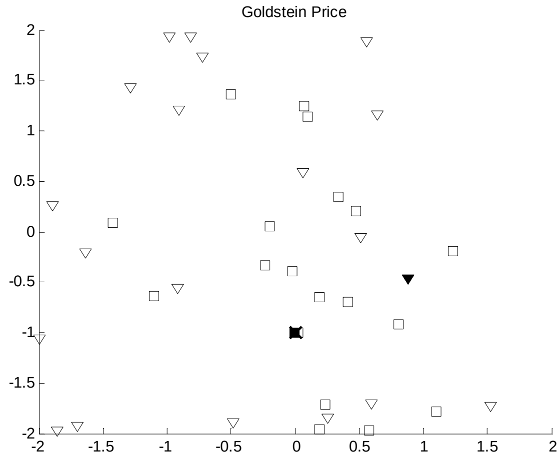

Figure 1: Original EM algorithm: points in the initial and final populations (GP)

for Rosenbrock2, where we chose 500 due to convergence problems) and δ=1E-03 (except for Rosenbrock2, where we used 1E-02). The results that are reported in Table 1 include the name of the test function, Tf, the number of points in the final (last iteration) population (when the shrinking strategy is used), m_f, the number of iterations of the EM algorithm, It_{EM}, and the total number of function evaluations, f_e.

The tested codes are denoted by: orig (original EM algorithm); orig-shri (original EM algorithm with shrinking strategy); ps (EM pattern search algorithm); ps-shri (EM pattern search algorithm with shrinking strategy). All random quantities were obtained with the seed number set to 0.

We noticed that the population shrinking process was never activated when solving problems McCormick and Spherical. The *deviation ratios* were never under 30%.

From Table 1 one can see that the implementations based on the pattern search algorithm require smaller number of EM iterations, although in some cases with larger number of function evaluations. The accuracy of the solutions is also good. The results produced by the shrinking process show a reduction on the iterations or on the function evaluations. In particular, with the function Rosenbrock2, the two codes based on shrinking significantly outperform the others in terms of convergence rate and accuracy of the solution. Codes orig and ps reach the maximum number of iterations (500) without meeting the default relative error criterion.

We use the functions Goldstein and Price and Hump to illustrate the population in the first and final iterations of the 4 tested codes. Figures 1 - 8 show the location of the randomly generated points in the initial population, represented by $\bigtriangledown$, and the best point

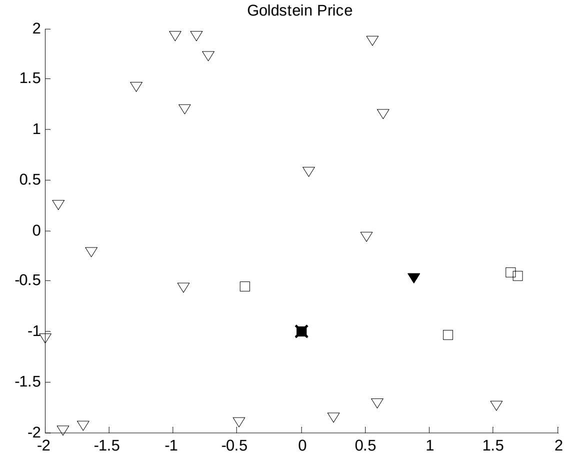

Figure 2: Original EM with shrinking strategy: points in the initial and final populations (GP)

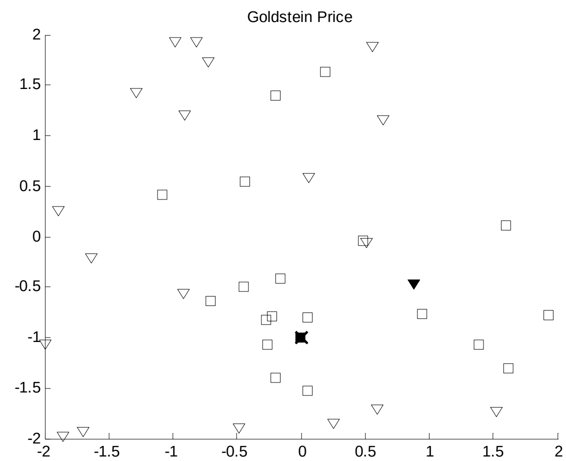

Figure 3: EM pattern search algorithm: points in the initial and final populations (GP)

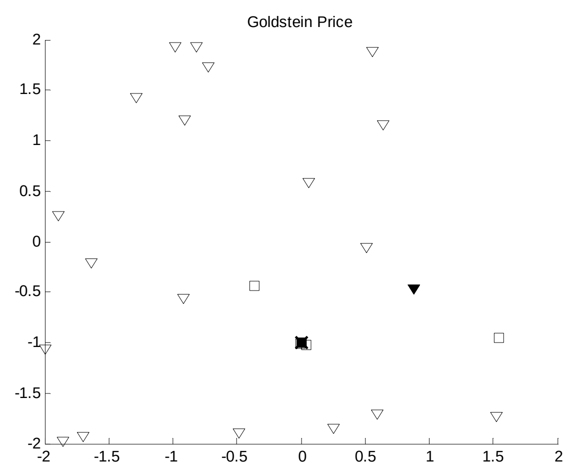

Figure 4: EM pattern search algorithm with shrinking strategy: points in the initial and final populations (GP)

represented by ▼; the location of the points in the final population, represented by □, and the best point represented by ■. The location of the known global solution is represented by $\times$.

First the function Goldstein and Price. Figures 1 and 2 correspond to the original EM algorithm without and with the shrinking strategy, respectively. In the second case, the population of 20 points is reduced to 10 points after the first iteration and reduced to 5 points in iteration 6. The algorithm takes 11 iterations to reach the solution (see Table 1). Figures 3 and 4 correspond to the EM pattern search algorithm without and with the shrinking strategy, respectively. Here, the population is reduced to 10 points after the first iteration and reduced again to 5 points in iteration 5.

The remaining Figures 5 - 8 contain similar illustrations for the function Hump. With the original EM algorithm, the shrinking strategy reduced the population in iteration 3 to 10 points which are maintained until the final iteration. In the EM pattern search algorithm, the population is reduced to 10 points in the third iteration and the iterative process terminates just after the reduction. Overall the reduction of the population did not affect the rate of convergence and accuracy of the results.

6 Conclusions

We have presented a modification to the Electromagnetism-like algorithm given in [3] for solving the global optimization problem (1). The original Hook and Jeeves pattern search method is proposed as the *Local* procedure and a population shrinking process is incorporated in the algorithm so that the population can be reduced over the iterative process, and fewer evaluations of the objective functions are required, without affecting the convergence rate. The numerical experiments and comparisons that were carried out show that the modified EM algorithm is efficient, although a more significant reduction on the function evaluations was expected. More numerical experiments have to be done. Another important challenge is to extend this modified EM algorithm, incorporating the shrinking strategy, to problems with large numbers of global and non-global optima in the region.

To reduce the need for large sets of points in the initial population we intend to apply the number-theoretic method, a deterministic process that produces a set of uniformly scattered points in the feasible region [6]. It seems that this new process has the ability to explore the search space uniformly and requires in practice smaller number of function evaluations.

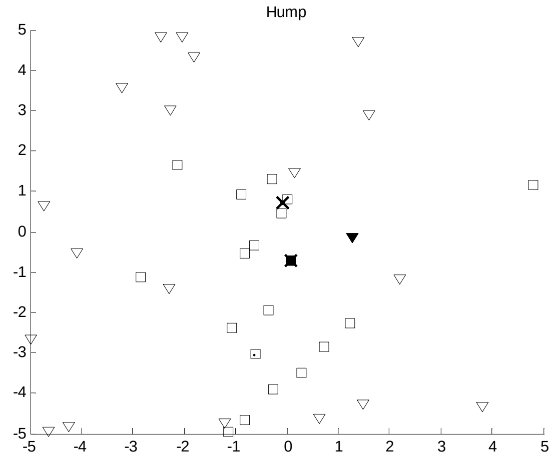

Figure 5: Original EM algorithm: points in the initial and final populations (H)

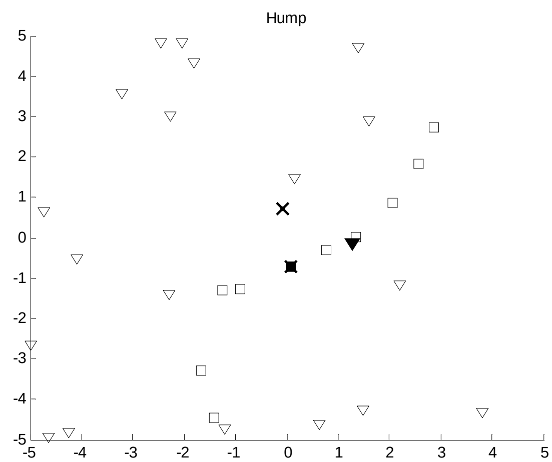

Figure 6: Original EM with shrinking strategy: points in the initial and final populations (H)

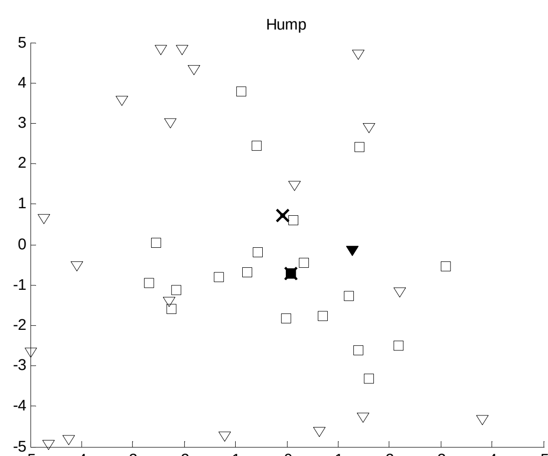

Figure 7: EM pattern search algorithm: points in the initial and final populations (H)

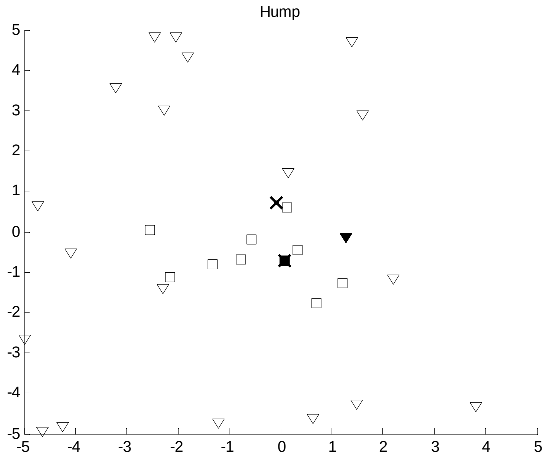

Figure 8: EM pattern search algorithm with shrinking strategy: points in the initial and final populations (H)

References:

[1] C. Audet and J.E. Dennis, Analysis of generalized pattern search, *SIAM J. Optim.* 13, 2003, pp. 889–903.

[2] S.I. Birbil, Stochastic Global Optimization Techniques, PhD Thesis, North Carolina State University 2002.

[3] S.I. Birbil SI and S. Fang, An Electromagnetism-like mechanism for global optimization, *J. of Global Optimization* 25, 2003, pp. 263–282.

[4] S.I. Birbil, S. Fang and R. Sheu, On the convergence of a population-based global optimization algorithm, *J. of Global Optimization* 30, 2004, pp. 2-3.

[5] R. Hooke and T.A. Jeeves, Direct search solution of numerical and statistical problems, *J. Assoc. Comput. Mach.* 8, 1961, pp. 212–229.

[6] Y-C. Jiao, C. Dang, Y. Leung and Y. Hao, A modification to the new version of the Prices algorithm for continuous global optimization problems, *J. of Global Optimization* 36, 2006, pp. 609-626.

[7] T.G. Kolda, R.M. Lewis and V. Torczon, Optimization by direct search: new perspectives on some classical and modern methods, *SIAM Review* 45, 2003, pp. 385–482.

[8] R.M. Lewis and V. Torczon, Pattern search algorithms for bound constrained minimization, *SIAM J. Optim.* 9, 1999, pp. 1082–1099.

[9] R.M. Lewis and V. Torczon, A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds, *SIAM J. Optim.* 12, 2002, pp. 1075–1089.

[10] V. Torczon, On the convergence of pattern search algorithms, *SIAM J. Optim.* 7, 1997, pp. 1–25.