Numerical Experiments with a Population Shrinking Strategy within a Electromagnetism-like Algorithm

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

***Abstract*—This paper extends our previous work done with a modified Electromagnetism-like (EM) algorithm to a benchmark global optimization collection of test problems. The EM algorithm is a population-based stochastic method that uses an attraction-repulsion mechanism to move sample points towards optimality. The modifications include a local search based on the original pattern search method of Hooke and Jeeves and a shrinking strategy that aims to reduce the population size as the iterative process progresses. Performance profiles are used to compare the proposed modifications with the original EM algorithm considering the average number of function evaluations and the best function value.**

***Index Terms*—Global optimization, Electromagnetism-like algorithm, Population shrinking, Performance profiles.**

I. 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 $f : \mathbb{R}^n \to \mathbb{R}$ 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. We also assume that the derivatives are not available. To solve a global optimization problem like (1), Birbil and Fang [4] proposed the electromagnetism-like (EM) algorithm. This is a population-based stochastic search method that 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 towards optimality. To improve the accuracy of the solutions, the EM algorithm incorporates a random local search that is applied to the best point in the population [3].

Recently [12], we proposed a modification to the EM algorithm using the well-known derivative-free pattern search method of Hooke and Jeeves [7] as a local search procedure. In general, we were able to achieve better accuracy results, although at a cost of function evaluations. Population-based methods are computationally expensive as they require large number of function evaluations. While this is not a critical issue in some problems, there are practical engineering applications where function evaluations are costly and should be avoided. One way to reduce the total number of function evaluations is to define a population with less points.

Beginning the iterative process with rather small populations does not seem a good strategy as population-based methods rely on the random scatter of the points in the feasible region to search for promising regions, as well as to prevent the algorithm to get stuck at nonglobal solutions.

However, a reduction of the population size at different stages of the iterative process does not indeed affect the algorithm convergence to the global solution. Better accuracy solutions are obtained even with a reduction in the computational requirements. Thus, this paper performs an extensive practical evaluation of two modifications to the EM algorithm of [4]:

(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.

The remaining paper is organized as follows. Section II introduces the original EM algorithm, Section III summarizes the pattern search method of Hooke and Jeeves and Section IV explains the main ideas of the population shrinking strategy. Section V reports the numerical results and includes a comparison of the proposed modifications with the original EM algorithm. Some conclusions are drawn in Section VI.

II. 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 ith point of a population; x^{best} is the point that has the least objective function value; $x^i_k \in \mathbb{R}$ is the kth ($k = 1, \ldots, n$) coordinate of the point x^i of the population;

Ana Maria A. C. Rocha is with the Department of Production and Systems, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal (corresponding author E-mail: arocha@dps.uminho.pt).

Edite M. G. P. Fernandes is with the Department of Production and Systems, University of Minho, Campus de Gualtar, 4710-057 Braga, Portugal (E-mail: emgpf@dps.uminho.pt).

m is the number of points in the population; N_{ls}^{max} 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, N_{ls}^{max}, δ)
Initialize()
iteration $\leftarrow$ 1
while termination criterion is not satisfied **do**
  F $\leftarrow$ *CalcF*()
  Move(F)
  Local(N_{ls}^{max}, δ)
  iteration $\leftarrow$ iteration + 1
end while

The procedure *Initialize* aims to randomly generate m points from the feasible region. Each coordinate of a point ($x_k^i, k = 1,\ldots,n$) is assumed to be uniformly distributed between the corresponding upper and lower bounds, i.e.,

$$x_k^i = l_k + \lambda(u_k - l_k)$$

where $\lambda \sim U(0,1)$. The objective function values are computed for all the points in the population, and the point with the least function value, x^{best}, is identified.

The procedure *CalcF* 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.,

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

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

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

where

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

The procedure *Move* 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.,

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

for $i = 1,\ldots,m$ and $i \neq best$. 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 procedure *Local* presented in [4] is a random line search algorithm and is applied coordinate by coordinate only to the best point to explore the neighborhood of that point in the population. First, based on the parameter δ, the procedure computes the maximum feasible step length, $\delta(\max_k(u_k - l_k))$. Second, for each coordinate k, the best point is assigned to a temporary point y to store the initial information. Then, a random number is selected as a step length and the point y is moved along that direction. If an improvement is observed, within N_{ls}^{max} iterations, the best point is replaced by y and the search along that coordinate k ends. The reader is referred to [3], [4], [5] for details.

The termination criterion can include various conditions, such as solution accuracy, a limit on iteration evaluations and/or a limit on objective function evaluations.

III. Hooke and Jeeves pattern search method

Birbil and Fang [4] show that the procedure *Local* is crucial in improving the accuracy of the computed solutions. However, an increase in the number of function evaluations has been observed. A local procedure applied to all points in the population is costly in terms of function evaluations, and has been noticed that only a small set of points should be considered for a local refinement. Assuming that the local search method is applied only on the best point of the population, at each iteration, then a deterministic direct search method with guaranteed convergence properties seems indeed the most appropriate.

One of the most popular direct search methods that can be used as a local procedure in the EM algorithm context is the pattern search (PS) method [2], [9], [14]. This method has been applied to unconstrained optimization and then successfully extended to bound constrained [10], as well as to equality constrained problems [11]. We chose to implement the original Hooke and Jeeves pattern search method [7], [14].

To generate feasible points in this local search procedure, the pattern search method is applied to the penalty-type function

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

instead of $f(x)$. Any generated trial point that is not feasible will never be accepted since the corresponding function value is ∞.

This pattern search method carries out two types of moves: the exploratory move and the pattern move. A brief explanation follows. The exploratory move carries out a coordinate search about the best point, with a step length of δ. If at the new point, y, $P(y) < P(x^{best})$, the iteration is successful. Otherwise, the iteration is unsuccessful and δ should be reduced by a factor of ε_δ. 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. The minimum step length allowed is δ^{min}.

IV. A POPULATION SHRINKING STRATEGY

The purpose of implementing a population shrinking strategy is to reduce the number of points in the population, as the iterative process progresses, and consequently to reduce the overall number of objective function evaluations. Since the accuracy of the results should not be affected, the crucial point here is to choose the best way to shrink the population, i.e., to decide when and how to shrink the population. When the concentration of all the points in the population around the best point is considered acceptable, it seems that some points could be discarded without affecting the convergence to the solution.

One way to measure the concentration of the points in the population around the best point is to compute the spread of the function values with respect to the best value, represented by SPR and defined as

$$\text{SPR} = \left(\frac{\sum_{i=1}^{m} (f(x^i) - f(x^{best}))^2}{m} \right)^{1/2}.$$

Thus, this quantity is used to decide when to shrink the population. Since, in practice, the SPR does not monotonically decrease as the iterative process converges to the solution, the idea is to shrink the population, at a particular iteration, whenever the SPR of that iteration is below a certain percentage of the SPR of a reference iteration (SPR^{ref}). We define the reference iteration as the first iteration of the set of iterations that have the same population of that particular iteration. Thus, the population is shrunk if

$$\text{SPR} < \epsilon \, \text{SPR}^{ref} \tag{2}$$

holds. In practice, $\epsilon = 0.1$ proved to be a good choice. When condition (2) holds, the population is halved, and the reference iteration is updated. We remark that the proposed shrinking process is only activated when the population has at least $2n$ points. The algorithm that corresponds to this EM mechanism with the shrinking strategy is as follows:

Algorithm 2 (m, N_{ls}^{max}, δ, ϵ)
Initialize()
iteration $\leftarrow$ 1
SPR^{ref} $\leftarrow$ *Compute*(SPR)
while termination criterion is not satisfied **do**
  F $\leftarrow$ *CalcF*()
  Move(F)
  Local(N_{ls}^{max}, δ)
  if $m > 2n$ **then**
  Compute(SPR)
    if SPR $< \epsilon$ SPR^{ref} **then**
      $m = m/2$ and discard m points
      SPR^{ref} $\leftarrow$ SPR
    end if
  end if
  iteration $\leftarrow$ iteration + 1
end while

V. NUMERICAL EXPERIMENTS

Computational tests were performed on a PC with a 3GHz Pentium IV microprocessor and 1Gb of memory. We compare the original EM algorithm, as described in Section II, with the proposed modifications. There are 4 codes to be compared. To simplify the notation we use the following abbreviations: `EM` (original EM algorithm); `EM-Shri` (original EM algorithm with the inclusion of the shrinking strategy); `PS` (EM with the Hooke and Jeeves pattern search algorithm); `PS-Shri` (EM with the Hooke and Jeeves pattern search algorithm and the shrinking strategy). In our numerical experiments we use a collection of 50 benchmark global optimization test problems, produced in full detail in the Appendix B of [1]. Some problems are defined for different values of n, giving a total of 64 problems.

The algorithms terminate when the number of function evaluations exceeds N_{fe}^{max} or when the relative error in the best objective function value, with respect to f_{global} (global objective function value), is less than 0.01%. The values for the constants are: $N_{fe}^{max} = 100n^2$, $N_{ls}^{max} = 10$, $\delta = 0.001$, and in the pattern search method $\delta^{min} = 1 \times 10^{-8}$ and $\varepsilon_\delta = 0.1$.

Since problem dimensions in the test set vary from 2 to 30, we decided to use the number of points in the initial population dependent on n. Thus, we set $m = \min\{200, 10n\}$.

A. Illustration of the shrinking strategy

We choose the Goldstein and Price (GP) problem, with $f_{global} = 3$ at $(0, -1)$, to illustrate the behaviour of the population shrinking strategy. The problem has one global solution and three local solutions. The initial population has $m = 20$ points. In this experiment, all random quantities were obtained with the seed number set to 0.

We show in Fig. 1 and 2:

i) the location of the randomly generated points in the initial population, represented by ▼, with the best point represented by $\times$;
ii) the location of the points in the final population, represented by ■, with the best point represented by $+$;
iii) the location of the known global solution represented by $\circ$.

The graph on the left of Fig. 1 shows the initial and final populations of the `EM` algorithm. It takes 8 iterations to reach the solution, 256 function evaluations and $f_{best} = 3.000004$. The graph on the right illustrates `EM-Shri` and 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, 179 function evaluations and $f_{best} = 3.000073$. On the left of Fig. 2, we may observe the initial and final populations of the `PS` algorithm. It takes 5 iterations to reach $f_{best} = 3.000004$, after 306 function evaluations. On the right, the `PS-Shri` algorithm takes 6 iterations to reach $f_{best} = 3.000013$, and 350 function evaluations. Here, the population is reduced to 10 points after the first iteration and reduced again to 5 points in iteration 5.

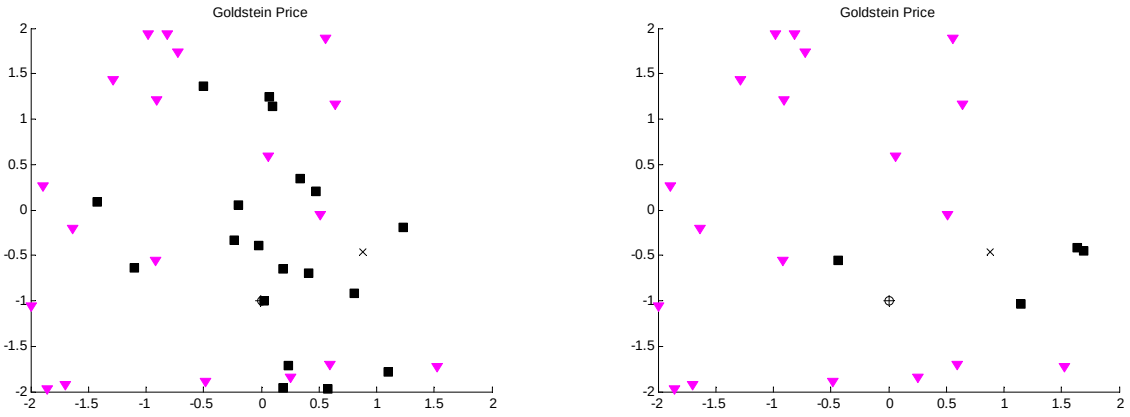

Fig. 1. Initial and final populations in `EM` and `EM-Shri` algorithms.

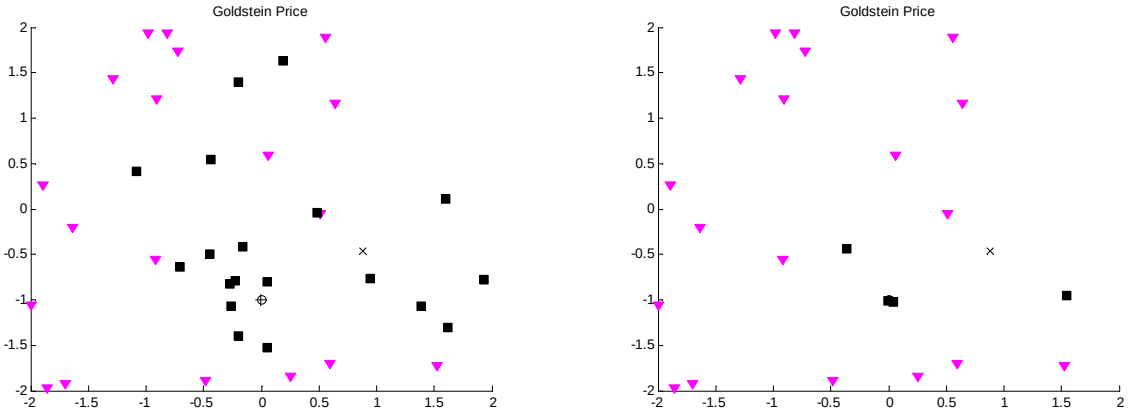

Fig. 2. Initial and final populations in `PS` and `PS-Shri` algorithms.

In previous work [13], we tested the shrinking strategy only on problems of dimension 2. They were two multi-modal functions (includes the GP problem) and three uni-modal functions. In two problems, the shrinking strategy was not activated since condition (2) was never achieved. Here, we extend the implementation of the algorithms to the 64 problems in the test set. The population was never shrunk on 33 problems when running `EM-Shri`, and on 41 problems when running `PS-Shri`.

B. Performance profiles

To evaluate and compare the performance of the four electromagnetism-like algorithms (`EM`, `EM-Shri`, `PS` and `PS-Shri`) on the set of selected global problems, we use performance profiles as outline in [6]. The performance profile plot represents the cumulative distribution function of a performance ratio, which is computed from an appropriate metric. Dolan and Moré in [6] proposed the use of the computing time required to solve a problem, but other metrics can be used. Using the performance profile plot one can compare how well the implementation of an algorithm can estimate the optimum relative to the others.

A brief explanation of this performance assessment follows. Let $\mathcal{P}$ be the set of all problems and $\mathcal{S}$ the set of solvers - implementation of the algorithms - used in the comparative study. Let $m_{(p,s)}$ be the performance metric of solver $s \in \mathcal{S}$ when solving problem $p \in \mathcal{P}$, according to the termination criteria previously defined. Every problem is solved 30 times with each solver, with different seed numbers (based on time) to randomly generate the initial population. Our first experience uses the average number of function evaluations, over the 30 runs, as the performance metric.

Thus, the performance ratios used in this comparative study are defined by

$$r_{(p,s)} = \frac{m_{(p,s)}}{\min\{m_{(p,s)} : s \in \mathcal{S}\}}$$

for $p \in \mathcal{P}, s \in \mathcal{S}$, and the overall assessment of the performance of a particular solver s is given 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 $\mathcal{P}$. The "size" is the number of problems in the set such that the performance ratio $r_{(p,s)}$ is less than or equal to τ for solver s. Thus, $\rho_s(\tau)$ is the probability (for solver $s \in \mathcal{S}$) that the performance ratio $r_{(p,s)}$ is within a factor $\tau \in \mathbb{R}$ of the best possible ratio.

If one is interested in the best assessment of the solvers, then f_{best}, the best function value found by solver s on problem p,

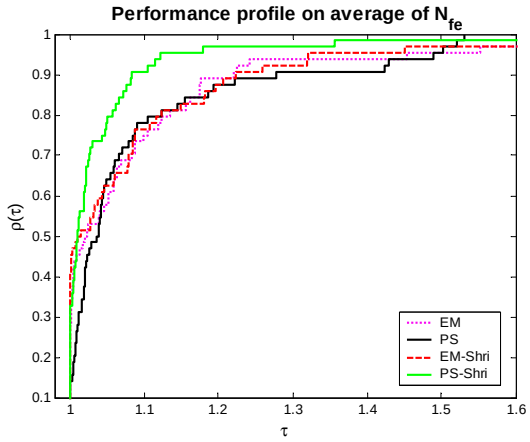

Fig. 3. Performance profile on the average number of function evaluations.

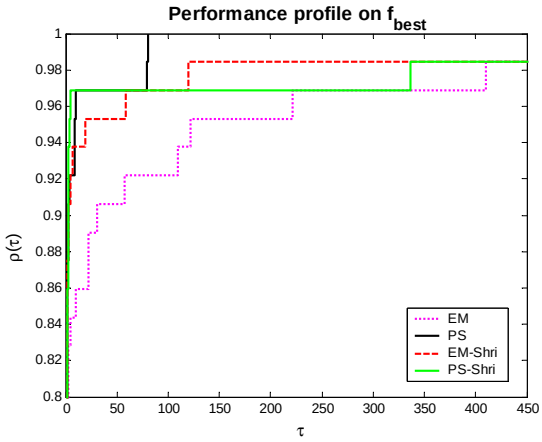

Fig. 4. Performance profile on the best function value.

over the 30 runs, should be used as the performance metric. However, in this case $\min\{m_{(p,s)} : s \in \mathcal{S}\}$ can be zero for a particular problem, or even negative, and the performance ratios used in this comparative study should then be given by

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

for $\epsilon = 0.00001$ (see [15] for a more complete discussion).

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. The solver with largest $\rho_s(\tau)$ is the one that solves more problems in the set $\mathcal{P}$.

The performance plots presented in Fig. 3 summarize the comparison on the selected 64 problems of the average number of function evaluations (N_{fe}). We may conclude that the `PS-Shri` algorithm is indeed the best one relative to the others. The introduction of the shrinking strategy improves efficiency of the `PS` algorithm. In the `EM` version, the shrinking strategy seems to slightly improve robustness.

When the comparative study is concerned with the best function value f_{best} ($f_{best} = \min(f^i_{best})$, $i = 1, \ldots, 30$), the algorithms `EM-Shri` and `PS-Shri` do not differ very much. The performance plots turn out to be very similar, see Fig. 4, except the ones corresponding to `EM` and `PS` algorithms. For large values of τ ($\tau > 80$) the algorithm `PS` solves all the problems to optimality.

C. Error behavior on the problem Neumaier 3

The performance of the proposed modifications to the EM algorithm is herein further examined on a problem that has varied dimension sizes. To show the algorithms behavior as n increases, we selected the problem Neumaier 3, represented by $NF3$ in [1]. This problem has global minima that can be expressed as

$$f_{global} = -\frac{n(n+4)(n-1)}{6},$$

$$\text{for } x_k^{global} = k(n+1-k),\ k = 1, \ldots, n.$$

Our analysis uses five values of n: $10, 15, 20, 25, 30$. To measure the accuracy of the solutions found by the algorithms we use the mean absolute error (MAE) defined by

$$MAE = \frac{|f_{global} - f_{avg}|}{n}$$

where f_{avg} is the average value of the best function values obtained over the 30 runs.

Table I lists values of f_{global}, f_{best}, f_{avg}, MAE, N_{fe} (average number of function evaluations, over the 30 runs), m and m_{final} (the average number of points in the final population, over 30 runs). Note that the initial population has $m = \min\{200, 10n\}$ points.

The effect of the problem dimension on the algorithms performance is not surprising since some global optimization algorithms have poor searching ability as n increases.

The algorithms based on the shrinking strategy achieve in general the lowest numerical errors. Using MAE as a measurement of error, Fig. 5 shows how the errors grow from $n = 10$ to $n = 30$. We may observe that with the original EM algorithm the error grows faster with n than with the other algorithms. The shrinking strategy has reduced the dependency of the algorithm on n. This is particularly important in the `EM` algorithm. We also note that a significant reduction on the N_{fe} is obtained when the shrinking strategy is coupled with the `PS` algorithm, see Table I. We would like to point out that the f_{best} values obtained by the `PS-Shri` algorithm are extremely close to the analytical results.

VI. Conclusions

We have presented a detailed comparative study of two types of modifications to introduce in the Electromagnetism-like algorithm given in [4] for solving the global optimization problem (1). Our results demonstrate favorable performance of the proposed modifications.

The original Hooke and Jeeves pattern search method is proposed as a procedure *Local* within the EM algorithm, and a population shrinking process is incorporated in the algorithm so that the population can be reduced over the iterative process. Fewer objective function evaluations are required and the convergence rate of the resulting algorithm is not affected.

TABLE I
PERFORMANCE OF PROPOSED MODIFICATIONS TO EM ALGORITHM ON PROBLEM NEUMAIER 3

n	code	f_{global}	f_{best}	f_{avg}	N_{fe}	MAE	m	m_{final}
10	EM	-210	-206.62341	-199.02158	10066	1.09784	100	100
	PS		-209.99193	-209.98659	6607	0.00134		100
	EM-Shri		-209.98218	-209.97975	8167	0.00203		12
	PS-Shri		-209.99328	-209.98618	4436	0.00138		12
15	EM	-665	-643.38018	-622.90264	22567	2.80649	150	150
	PS		-664.94572	-664.93901	17421	0.00407		150
	EM-Shri		-664.83907	-664.77173	22517	0.01522		18
	PS-Shri		-664.94919	-664.93874	11385	0.00408		18
20	EM	-1520	-1398.81653	-1366.80792	40100	7.65960	200	200
	PS		-1519.86269	-1519.67868	39437	0.01607		200
	EM-Shri		-1518.32553	-1517.88224	40024	0.10589		25
	PS-Shri		-1519.86621	-1519.85636	27585	0.00718		25
25	EM	-2900	-2678.89641	-2608.16469	62643	11.67341	200	200
	PS		-2899.71944	-2896.58852	62696	0.13646		200
	EM-Shri		-2881.94390	-2878.90144	62539	0.84394		50
	PS-Shri		-2899.74243	-2899.69538	57611	0.01218		50
30	EM	-4930	-4520.75077	-4425.00997	90121	16.83300	200	200
	PS		-4927.98962	-4919.48719	90342	0.35043		200
	EM-Shri		-4871.86254	-4864.56731	90039	2.18109		50
	PS-Shri		-4929.54779	-4928.43544	89923	0.05215		50

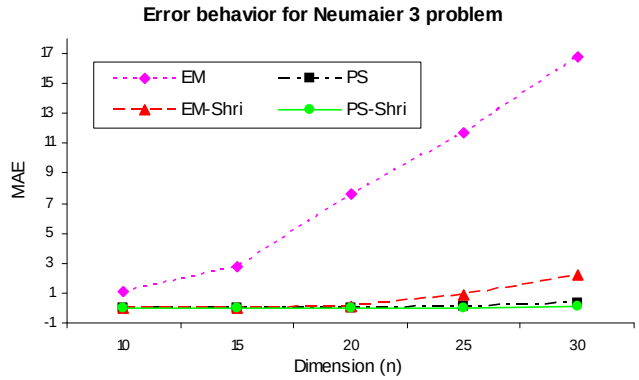

Fig. 5. MAE behavior on problem Neumaier 3 (for 5 values of n)

When the performance assessment of the algorithms is based on the average number of function evaluations, the shrinking strategy is more effective when coupled with the EM algorithm based on the Hooke and Jeeves pattern search local procedure. If one is interested in the performance assessment of the best function value then the shrinking strategy significantly improves the EM algorithm performance.

Another important conclusion that we can draw from our numerical experiments is that the shrinking strategy reduces the dependency of the algorithm on the number of variables of the problem. This strategy seems rather appropriate for solving large global optimization problems with bounded variables.

To further 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 [8]. It seems that this new process has the ability to explore the search space uniformly and requires in practice reduced number of function evaluations.

REFERENCES

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

[2] C. Audet and J. E. Dennis, "Analysis of generalized pattern search," *SIAM Journal on Optimization*, vol. 13, pp. 889–903, 2003.

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

[4] S. I. Birbil and S. Fang, "An Electromagnetism-like mechanism for global optimization," *Journal of Global Optimization*, vol. 25, n. 3, pp. 63–282, 2003.

[5] S. I. Birbil, S. Fang, and R. Sheu, "On the Convergence of a Population-Based Global Optimization Algorithm," *Journal of Global Optimization*, vol. 30, pp. 301–318, 2004.

[6] E. D. Dolan and J. J. Moré, "Benchmarking optimization software with performance profiles," *Mathematical Programming*, vol. 91, pp. 201–213, 2002.

[7] R. Hooke and T. A. Jeeves, "Direct search solution of numerical and statistical problems," *Journal of Association Computing Machinery*, vol. 8, pp. 212–229, 1961.

[8] Y.-C. Jiao, C. Dang, Y. Leung, and Y. Hao, "A modification to the new version of the Price' s algorithm for continuous global optimization problems," *Journal of Global Optimization*, vol. 36 pp. 609-626, 2006.

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

[10] R. M. Lewis and V. Torczon, "Pattern search algorithms for bound constrained minimization," *SIAM Journal on Optimization*, vol. 9, pp. 1082–1099, 1999.

[11] R. M. Lewis and V. Torczon, "A globally convergent augmented Lagrangian pattern search algorithm for optimization with general constraints and simple bounds," *SIAM Journal on Optimization*, vol. 12, pp. 1075–1089, 2002.

[12] A. M. A. C. Rocha and E. M. G. P. Fernandes, "A Modified Electromagnetism-like Algorithm Based on a Pattern Search Method," ECC07, Springer-Verlag, 6 pages, 2007 (in press).

[13] A. M. A. C. Rocha and E. M. G. P. Fernandes, "A new electromagnetism-like algorithm with a population shrinking strategy," in *Proc. 6th. WSEAS International Conference on System Science and Simulation in Engineering*, ISBN: 978-960-6766-14-4, 2007, pp. 307–312.

[14] V. Torczon, "On the convergence of pattern search algorithms," *SIAM Journal on Optimization*, vol. 7, pp. 1–25, 1997.

[15] A. I. F. Vaz and L. N. Vicente, "A particle swarm pattern search method for bound constrained global optimization," *Journal of Global Optimization*, vol. 39, pp. 197–219, 2007.