Journal of Global Optimization **25:** 263–282, 2003.

An Electromagnetism-like Mechanism for Global Optimization

Ş. İLKER BİRBİL* and SHU-CHERNG FANG
*Department of Industrial Engineering and Operations Research Program, North Carolina State University, Raleigh, NC 27695-7906, USA (*Corresponding author: e-mail: sibirbil@eos.ncsu.edu)*

Abstract. This paper proposes a new heuristic for global optimization. The method utilizes an attraction-repulsion mechanism to move the sample points towards the optimality. The proposed scheme can be used either as a stand-alone approach or as an accompanying procedure for other methods. Some test results on nonlinear test functions in the category of "minor to moderate difficulty" are included. The ease of implementation and flexibility of the heuristic show the potential of this new approach.

Key words: Global optimization, Attraction–repulsion mechanism, Population-based heuristics

1. Introduction

In recent years global optimization has become a rapidly developing field. Many real life problems in areas such as physics, chemistry and molecular biology (Schoen, 1989; More and Wu, 1995) involve *nonlinear functions of many variables* with attributes (multi modality, discontinuouity, etc.) that are difficult to optimize by conventional mathematical tools, such as the gradient methods.

To overcome the difficulties described above, stochastic search methods, which rely heavily on computer power, have been developed starting in the 1980s. A random search algorithm may give a usable solution even when other algorithms fail because of irregularities or high dimensionality. We refer to Dixon and Szegö (1978), Kan and Timmer (1987a, b), Törn and Žilinskas (1989) and Boender (1985) for excellent surveys.

In this paper we study a special class of optimization problems with bounded variables in the form of:

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

where $[l, u] := \{x \in \Re^n \mid l_k \leqslant x_k \leqslant u_k, k = 1, ...n\}$. Because of the simplicity in maintaining feasibility, this model has been extensively studied by random search methods (Demirhan et al., 1999), and also by direct methods (Huyer and Neumaier, 1999) .

The organization of the paper is as follows. In Section 2, the motivation behind the proposed heuristic is introduced. The general scheme and the sub procedures

of the algorithm are discussed in Section 3. In Section 4, the computational results on a set of test problems are presented. Further research and conclusion are given in Section 5.

2. Motivation

In stochastic global optimization, population-based algorithms start with randomly sampling points from the feasible region. According to the objective function values of these sample points, the regions of attraction are determined. Then a mechanism is invoked for further exploitation of these candidate regions. In *Genetic Algorithms* this mechanism corresponds to the reproduction, crossover and mutation operators (Michalewicz, 1994), whereas in *two phase methods* the exploration of the feasible region is conducted by random sampling followed by hill-climbing (Kan and Timmer, 1987a; Törn and Viitanen, 1994) or gradient-based methods (Fletcher and Reeves, 1964).

Similarly, we construct a mechanism that encourages the points to converge to the highly attractive valleys, and contrarily, discourages the points to move further away from steeper hills. This idea enables us to make an analogy with the attraction–repulsion mechanism of the electromagnetism theory.

Similar to that in the elementary electromagnetism, we can think of each sample point as a charged particle that is released to a space. In our approach, the *charge* of each point relates to the objective function value, which we are trying to optimize. This charge also determines the magnitude of attraction or repulsion of the point over the sample population – the better the objective function value, the higher the magnitude of attraction.

After calculating these charges, we use them to find a direction for each point to move in subsequent iterations. We select this direction by evaluating a combination force exerted on the point via other points. Like the electromagnetic forces, this force is calculated by adding vectorially the forces from each of the other points calculated separately.

We need to state that though the analogy with electromagnetism theory motivates the idea, there are some notable differences, which we will make clear when we introduce the heuristic in subsequent sections.

Finally, similar to the hybrid population-based algorithms (Hart, 1994; Glover and Laguna, 1995), we may apply a local search procedure to improve some of the objective function values observed in the population.

3. Electromagnetism-like (EM) Heuristic

We deal with the functions of the form (1), with the following parameters given:

n dimension of the problem.
u_k upper bound in the kth dimension.
l_k lower bound in the kth dimension.
$f(x)$ pointer to the function that is minimized.

In this section, we introduce the general scheme of the EM heuristic and present its sub procedures. This section ends by an example that demonstrates the behavior of the algorithm visually.

3.1. GENERAL SCHEME FOR EM

The heuristic (EM) consists of four phases. These are *initialization* of the algorithm, calculation of the *total force* exerted on each particle,* *movement* along the direction of the force, and application of *neighborhood search* to exploit the local minima. The general scheme can be given as in Algorithm 1.

ALGORITHM 1. EM(m, $MAXITER$, $LSITER$, δ)
m: number of sample points
$MAXITER$: maximum number of iterations
$LSITER$: maximum number of local search iterations
δ: local search parameter, $\delta \in [0, 1]$

1: Initialize()
2: iteration $\leftarrow 1$
3: **while** iteration $< MAXITER$ **do**
4: Local($LSITER$, δ)
5: **F** $\leftarrow$ CalcF()
6: Move(**F**)
7: iteration $\leftarrow$ iteration $+ 1$
8: **end while**

3.2. INITIALIZATION

The procedure *Initialize* is used to sample m points randomly from the feasible domain, which is an n dimensional hyper-cube. Each coordinate of a point is assumed to be uniformly distributed between the corresponding upper bound and lower bound. After a point is sampled from the space, the objective function value for the point is calculated using the function pointer $f(x)$ (Algorithm 2, line 6). The procedure ends with m points identified, and the point that has the best function value is stored in x^{best} (line 8).

* We use the words *particle* and *point* interchangeably throughout the paper.

ALGORITHM 2. Initialize()

1: **for** $i = 1$ to m **do**
2: **for** $k = 1$ to n **do**
3: $\lambda \leftarrow U\ (0, 1)$
4: $x_k^i \leftarrow l_k + \lambda(u_k - l_k)$
5: **end for**
6: Calculate $f(x^i)$
7: **end for**
8: $x^{\text{best}} \leftarrow \text{argmin}\{f(x^i), \forall i\}$

3.3. LOCAL SEARCH

The procedure *Local* is used to gather the local information for a point x^i. The parameters, *LSITER* and δ that are passed to this procedure, represent the number of iterations and the multiplier for the neighborhood search, respectively.

The procedure iterates as follows: First, the maximum feasible step length (*Length*) is calculated according to the parameter δ (Algorithm 3, line 2). Second, for a given i, improvement in x^i is sought coordinate by coordinate (lines 5–13). For a given coordinate, the point x^i 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 the point y observes a better point within *LSITER* iterations, the point x^i is replaced by y and the neighborhood search for point i ends (lines 14–17). Finally the *current best* point is updated (line 22).

This is a simple random line search algorithm applied coordinate by coordinate. This procedure does not require any gradient information to perform the local search. Instead of using other powerful local search methods (Solis and Wets, 1981), we have utilized this procedure because we wanted to show that even with this trivial method, the algorithm shows promising convergence properties.

ALGORITHM 3. Local($LSITER, \delta$)

1: counter $\leftarrow 1$
2: Length $\leftarrow \delta(max_k\{u_k - l_k\})$
3: **for** $i = 1$ to m **do**
4: **for** $k = 1$ to n **do**
5: $\lambda_1 \leftarrow U\ (0, 1)$
6: **while** counter $< LSITER$ **do**
7: $y \leftarrow x^i$
8: $\lambda_2 \leftarrow U\ (0, 1)$
9: **if** $\lambda_1 > 0.5$ **then**
10: $y_k \leftarrow y_k + \lambda_2$(Length)
11: **else**

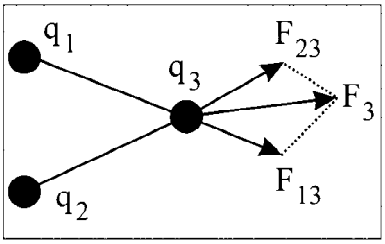

Figure 1. Superposition principle ($\vec{F}_{i3} = \frac{q_3 q_i}{4\pi\varepsilon_0\varepsilon_r r^2}\vec{e}_r \quad i = 1, 2$).

```
            y_k ← y_k − λ_2(Length)
        end if
        if f(y) < f(x^i) then
            x^i ← y
            counter ← LSITER − 1
        end if
        counter ← counter + 1
    end while
  end for
end for
x^best ← argmin{f(x^i), ∀i}
```

3.4. CALCULATION OF TOTAL FORCE VECTOR

The *superposition principle* (Figure 1) of electromagnetism theory states that the force exerted on a point via other points is inversely proportional to the distance between the points and directly proportional to the product of their charges (Cowan, 1968).

In each iteration we compute the charges of the points according to their objective function values. However, in our heuristic the charge of each point is not constant and changes from iteration to iteration.

The charge of each point i, q^i, determines point i's power of attraction or repulsion. This charge is evaluated as,

$$q^i = \exp\left(-n\frac{f(x^i) - f(x^{\text{best}})}{\sum_{k=1}^{m}(f(x^k) - f(x^{\text{best}}))}\right), \ \forall i. \tag{2}$$

In this way, points that have better objective values possess higher charges. We multiply the fraction by the dimension n, because in higher dimensions the number

of points in the population tends to get large. As a result of this, the fraction may become very small, and may cause overflow problems in calculating the exponential function.

We define the charge, q^i according to the relative efficiency of the objective function value of the corresponding point in the population. This is clearly not the unique nor the optimal choice for this calculation. An alternative calculation, which rank the points according to their objective function values, may be used here. Our experiments have shown that the proposed calculation in Eq. (2) is satisfactory for our study.

Notice that, unlike electrical charges, no signs are attached to the charge of an individual point in Eq. (2). Instead, we decide the direction of a particular force between two points after comparing their objective function values. Hence, the total force F^i exerted on point i is computed by the following equation:

$$F^i = \sum_{j \neq i}^{m} \left\{ \begin{array}{ll} (x^j - x^i) \dfrac{q^i q^j}{\left\| x^j - x^i \right\|^2} & if \quad f(x^j) < f(x^i) \\ (x^i - x^j) \dfrac{q^i q^j}{\left\| x^j - x^i \right\|^2} & if \quad f(x^j) \geqslant f(x^i) \end{array} \right\}, \forall i \tag{3}$$

As seen in Algorithm 4 (lines 7–8), between two points, the point that has a better objective function value attracts the other one. Contrarily, the point with worse objective function value repels the other (lines 9–10). Since x^{best} has the minimum objective function value, it acts as an absolute point of attraction, i.e., it attracts all other points in the population.

When we examine the algorithm carefully, we can see that the determination of a direction via total force vector resembles the statistical estimation of the gradient of f. However, the estimation computed by our heuristic is different since this direction depends on the Euclidean distance between the points. That is, the points that become close enough may lead each other to a direction other than the statistically estimated one.

ALGORITHM 4. CalcF():**F**

1: **for** $i = 1$ to m **do**
2: $\quad q^i \leftarrow \exp\left(-n \frac{f(x^i) - f(x^{\text{best}})}{\sum_{k=1}^{m} (f(x^k) - f(x^{\text{best}}))}\right)$
3: $\quad F^i \leftarrow 0$
4: **end for**
5: **for** $i = 1$ to m **do**
6: $\quad$ **for** $j = 1$ to m **do**
7: $\quad\quad$ **if** $f(x^j) < f(x^i)$ **then**
8: $\quad\quad\quad F^i \leftarrow F^i + (x^j - x^i) \frac{q^i q^j}{\left\| x^j - x^i \right\|^2}$ {Attraction}
9: $\quad\quad$ **else**
10: $\quad\quad\quad F^i \leftarrow F^i - (x^j - x^i) \frac{q^i q^j}{\left\| x^j - x^i \right\|^2}$ {Repulsion}

```
11:        end if
12:     end for
13: end for
```

3.5. MOVEMENT ACCORDING TO THE TOTAL FORCE

After evaluating the total force vector F^i, the point i is moved in the direction of the force by a random step length as given in Eq. (4). Here the random step length, λ, is assumed to be uniformly distributed between 0 and 1. Obviously, there are many other distributions that can be used in calculation of this step length. But for ease of computation, we have applied uniform distribution.

We have selected the step length randomly in order to ensure that the points have a nonzero probability to move to the unvisited regions along this direction. In our future work, we hope to show the convergence in probability by using this random step length.

In Eq. (4), RNG is a vector whose components denote the allowed feasible movement toward the upper bound, u^k, or the lower bound, l^k, for the corresponding dimension (Algorithm 5, lines 6–10). Furthermore, the force exerted on each particle is normalized so that we can maintain the feasibility. Thus,

$$x^i = x^i + \lambda \frac{F^i}{\|F^i\|}(RNG) \qquad i = 1, 2, ..., m \tag{4}$$

Algorithm 5 gives the pseudo-code of the *Move* procedure. Note that the best point, x^{best}, is not moved and is carried to the subsequent iterations (line 2). This suggests that we may avoid the calculation of the total force on the current best point in Algorithm 4 (yet the computational effort for calculating the total force on current best point is negligible).

ALGORITHM 5. Move(**F**)

```
 1: for i = 1 to m do
 2:    if i ≠ best then
 3:       λ ← U (0, 1)
 4:       F^i ← F^i / ‖F^i‖
 5:       for k = 1 to n do
 6:          if F^i_k > 0 then
 7:             x^i_k ← x^i_k + λ F^i_k (u_k − x^i_k)
 8:          else
 9:             x^i_k ← x^i_k + λ F^i_k (x^i_k − l_k)
10:          end if
11:       end for
```

12: **end if**
13: **end for**

3.6. TERMINATION CRITERIA

In our study we terminate the EM procedure by using a maximum number of iterations. According to our test results, in general 25 iterations per dimension (i.e., $MAXITER = 25n$) is satisfactory for converging to the optimum point for the moderate difficulty functions.

Another termination criterion that might be used is the successive number of iterations spent without changing the current best point. In other words, if the current best point is not changed for certain number of iterations, the algorithm may be stopped. However this decision has to be studied carefully since algorithm may be stopped before converging to the global optimum. On the other hand, unnecessary function evaluations may be avoided by stopping earlier.

In the literature several other stopping conditions are studied (Törn et al., 1999). One of the frequently used criteria is to terminate the algorithm when the observed objective function value is ε-close to the optimal value (Huyer and Neumaier, 1999). However, this criterion is not appropriate if the global optimum is not known in advance.

3.7. AN EXAMPLE

We use the *Spiky* ⋆ function to demonstrate a typical run of the algorithm. In the following figures, $\Diamond$ represents the current best point and * shows the location of the global optimum. In this example we have selected the population size to be 20.

Figure 2 shows the location of the points when the algorithm is started by randomly sampling points from the feasible region. Initially the current best point (x^{best}) is far away from the global optimum.

The points in the population move towards the region around the current best point. Note that some points are repelled closer to the global optimum (Figure 3).

In Figure 4, one of the points observes an objective function value better than the current best point, and this new point becomes the current best point. Thus, the points start to converge towards this new x^{best}.

Figure 5 demonstrates that the points in the population are directed towards the region around the current best point. And after this iteration, the global optimum is located.

⋆ We also include this function in our computational study, Section 4.

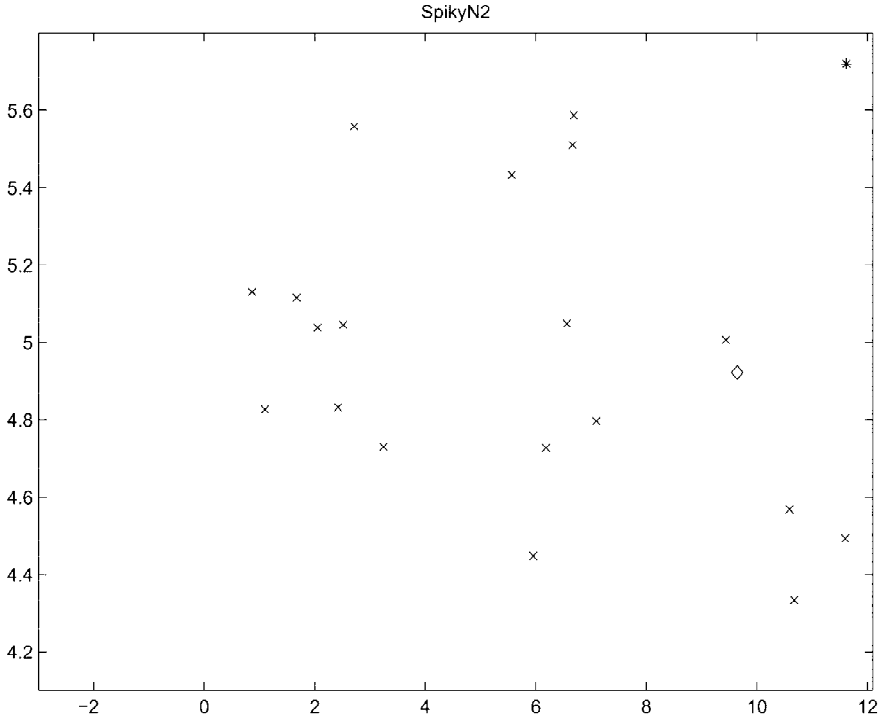

Figure 2. The starting position of the particles just after the procedure *Initialize*.

4. Computational Results

In a recent paper, Törn et al. (1999) classified the well-known global optimization problems as *unimodal*, *easy*, *moderately difficult*, and *difficult* problems. They also suggested that the test problems should represent different classes.

We have included some of the functions that Törn et al. (1999) had discussed and we have created a test function set consisting of 15 functions. The remaining functions are taken from Demirhan et al. (1999), and from the web site, http://solon.cma.univie.ac.at/~neum/glopt.html. We refer to this test function set as the *general test functions*.

Initially, we studied three versions of EM, which differ in the local search procedure. We demonstrate our results for the general test functions in terms of the average number of function evaluations over 25 runs. The average and best objective function values are also reported.

After selecting the best version, we applied EM to a well known test set, which consists of seven test functions (Dixon and Szegö, 1978). In addition to reporting our results, we compare the results with other methods recently known from the work by Huyer and Neumaier (1999).

All the computations were conducted on a Pentium-III 450 MHz PC. The algorithm is coded in C++ and it is available upon request.

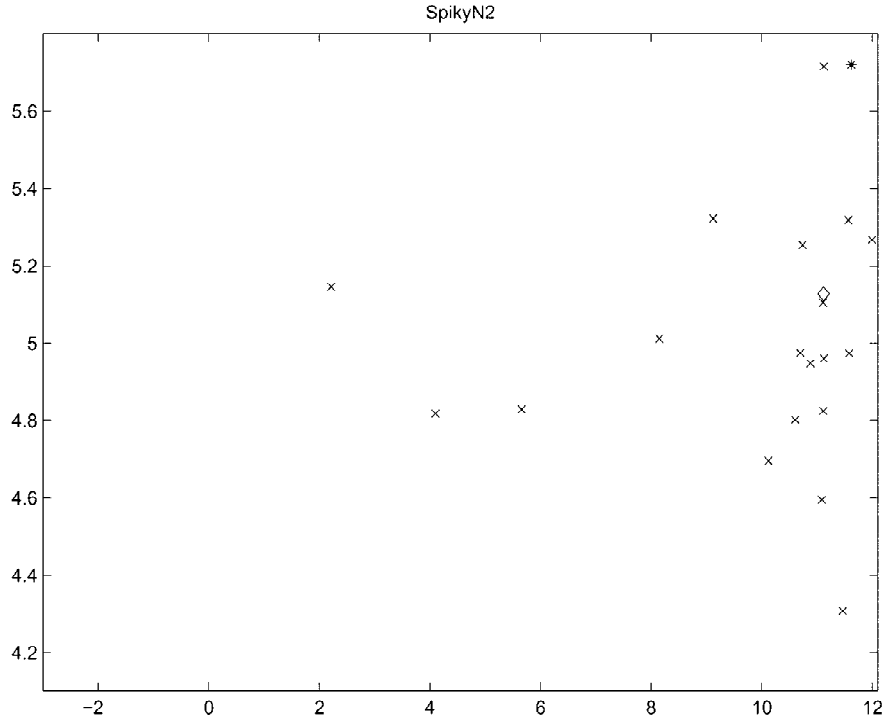

Figure 3. Points start to attract and repel each other.

4.1. GENERAL TEST FUNCTIONS

We tested EM with three different versions of the *Local* procedure. First, we excluded the local procedure completely from the general scheme. Second, we applied the procedure to all sample points. Finally, we experimented with the application of local search to the best point only.

The input parameters for the functions are given in Table 1. The functions are presented in alphabetical order, so the order of a function does not necessarily represent the difficulty of the corresponding function.

In order to analyze the effects of different parameters, we have selected two functions — one from each set — and constructed an experimental design. We include this experimental design in the Appendix.

4.1.1. *EM without Local procedure*

In order to examine the basic convergence properties of EM, we first excluded the *Local* procedure from the algorithm. This was achieved simply by assigning value 0 to the parameter LSITER. The immediate effect of this choice is the loss of the local information. In this case, even when a point is close to a local optimizer, it is not directed deeper into the valley. In certain sense, this approach behaves

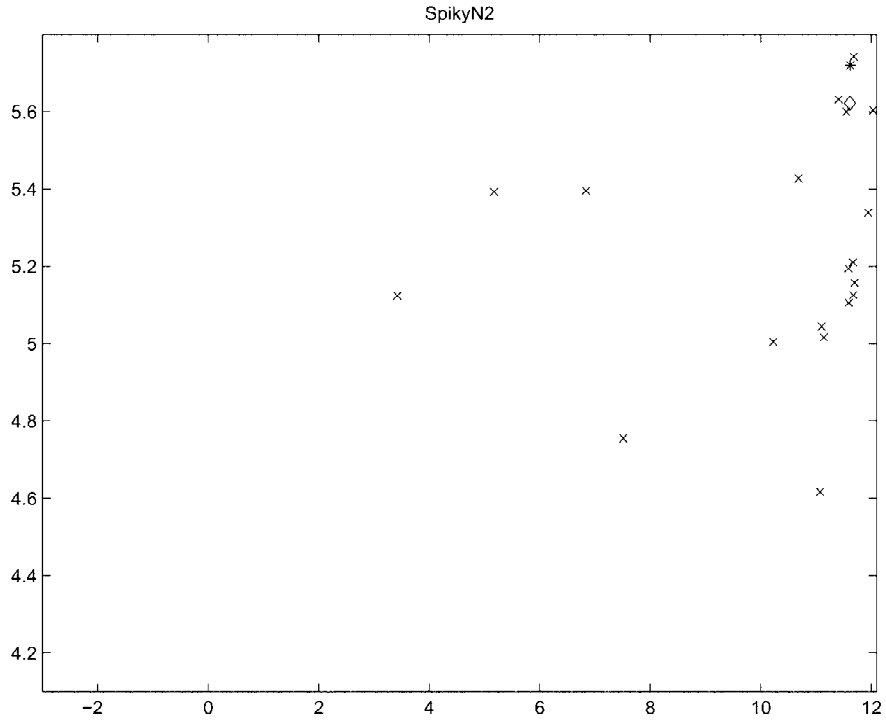

Figure 4. One of the repelled points observes a better region, and signals for the others.

Table 1. Parameters for General Test Functions

Function name	n	m	MAXITER	LSITER	δ
Complex	2	10	50	10	5.0e-3
Davis	2	20	50	30	5.0e-3
Epistacity(4)	4	30	50	10	1.0e-3
Epistacity(5)	5	40	100	20	1.0e-3
Griewank	2	30	100	20	1.0e-3
Himmelblau	2	10	50	5	1.0e-3
Kearfott	4	10	50	5	1.0e-3
Levy	10	20	75	5	1.0e-3
Rastrigin	2	20	50	10	5.0e-3
Sine Envelope	2	20	75	10	5.0e-4
Stenger	2	10	75	10	1.0e-3
Step	5	10	50	5	1.0e-3
Spiky	2	30	75	10	1.0e-3
Trid(5)	5	10	125	50	1.0e-2
Trid(20)	20	40	500	150	1.0e-3

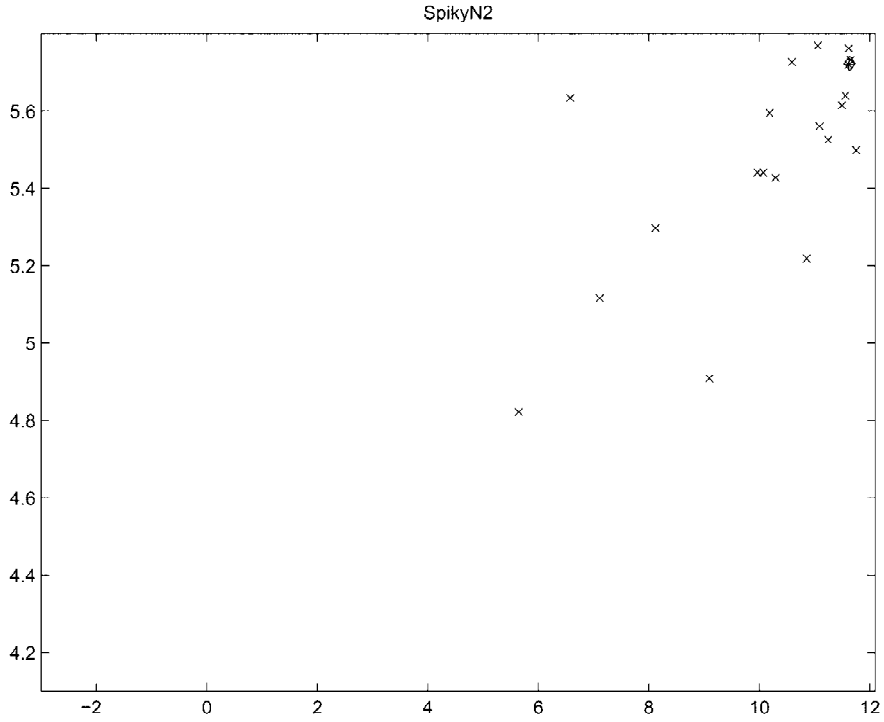

Figure 5. Best point attracts the others and global optimum is found.

Table 2. Results of EM without LOCAL procedure

Function name	Avg. evals.	Avg $f(x)$	Best $f(x)$	Known optimum
Complex	363	0.0175	0.0158	0.0
Davis	622	1.6157	1.5641	0.0
Epistacity(4)	1079	0.0379	0.0149	0.0
Epistacity(5)	2603	0.0355	0.0207	0.0
Griewank	1914	0.0896	0.0032	0.0
Himmelblau	84	0.0934	0.0759	0.0
Kearfott	231	0.0008	0.0000	0.0
Levy	835	0.1429	0.0303	0.0
Rastrigin	141	−1.9566	−1.9871	−2.0
Sine Envelope	962	0.0744	0.0400	0.0
Stenger	282	0.0020	0.0019	0.0
Step	728	0.0000	0.0000	0.0
Spiky	1702	−38.6378	−38.7251	−38.85
Trid(5)	968	−28.2997	−29.0324	−30.0
Trid(20)	43354	−33.2567	−177.6124	−1520.0

Table 3. Results of EM with LOCAL procedure applied to all points

Function name	Avg. evals.	Avg $f(x)$	Best $f(x)$	Known optimum
Complex	5534	0.0000	0.0000	0.0
Davis	8091	0.4088	0.1322	0.0
Epistacity(4)	32823	0.0003	0.0001	0.0
Epistacity(5)	189769	0.0001	0.0001	0.0
Griewank	50790	0.0000	0.0000	0.0
Himmelblau	3287	0.0000	0.0000	0.0
Kearfott	6190	0.0000	0.0000	0.0
Levy	44814	0.0001	0.0000	0.0
Rastrigin	10359	−2.0000	−2.0000	−2.0
Sine Envelope	15629	0.0116	0.0001	0.0
Stenger	8316	0.0000	0.0000	0.0
Step	5743	0.0000	0.0000	0.0
Spiky	10264	−38.8004	−38.8492	−38.85
Trid(5)	37154	−29.9979	−29.9999	−30.0
Trid(20)	1.5e+6	−1519.6117	−1519.9768	−1520.0

blindly. However, since no additional information is collected, the average number of function evaluation figures are not large.

The results in Table 2 show that the solutions for the functions *Davis* and *Trid(20)* are less satisfactory than those for the other functions. *Davis* is highly irregular in the neighborhood of the optimum, while the feasible region of *Trid(20)* is a 20-dimensional hypercube with bounds $[-400, 400]$ in each dimension. Therefore the number of evaluations in an experiment is not large enough to adequately explore the feasible space.

Our results show that even if we do not use the Local procedure, the average function values are still good. In most of the problems, EM is able to approximate the optimum. Nevertheless, the accuracy of the average function values is not good enough. This motivated the next experiment.

4.1.2. *EM with Local procedure applied to all points*

We next applied the Local procedure to all points in the population. This approach has two advantages; first, the attractive parts of the feasible region are more thoroughly examined, and second, the repelled points have better chance to lead to as yet undiscovered minimizers.

Notice that the performance of the algorithm improves but at the cost of the number of function evaluations required by the Local procedure (Table 3). Especially, the accuracy of the results for the functions *Davis*, *Levy*, and *Trid(20)*

Table 4. Results of EM with LOCAL procedure applied to current best point

Function name	Avg. evals.	Avg $f(x)$	Best $f(x)$	Known optimum
Complex	598	0.0000	0.0000	0.0
Davis	832	0.4538	0.2356	0.0
Epistacity(4)	1580	0.0002	0.0001	0.0
Epistacity(5)	4123	0.0002	0.0000	0.0
Griewank	2470	0.0000	0.0000	0.0
Himmelblau	520	0.0001	0.0000	0.0
Kearfott	712	0.0000	0.0000	0.0
Levy	2783	0.0001	0.0000	0.0
Rastrigin	792	−1.9898	−2.0000	−2.0
Sine Envelope	1007	0.0352	0.0097	0.0
Stenger	724	0.0000	0.0000	0.0
Step	870	0.0000	0.0000	0.0
Spiky	1520	−38.6684	−38.8486	−38.85
Trid(5)	1870	−29.9963	−29.9997	−30.0
Trid(20)	99731	−1519.4472	−1519.5543	−1520.0

are much better. However, the number of evaluations drastically increases for the problems with high dimensionality.

4.1.3. *EM with Local procedure applied to the current best point only*

To reduce the number of function evaluations, we tried applying *Local* procedure to the current best point only (Table 4). This choice balances the number of evaluations and the accuracy of the results.

For this case, the average number of evaluations is closer to those of the first version (Section 4.1.1), while the quality of the results is comparable to those of the second version (Section 4.1.2). However, when the function has good attractors spread over the feasible region as in functions *Sine Envelope* and *Spiky*, the points visiting the neighborhood of the global optimum may not be powerful enough to attract other points. In function *Davis*, the algorithm rapidly converges to the neighborhood of the optimum point, but the region around global optimum has many local minimizers in steep valleys. Thus, EM is trapped in one of these local minima.

4.2. FURTHER TEST SET

Our initial experiments show that EM rapidly converges to the minimizers. The local information gains more importance either when the function is highly irregu-

Table 5. Results of EM for Dixon and Szegö (1978) Functions

Function[a]	n	m	MAXITER	Avg. evals.	Avg $f(x)$	Best $f(x)$	f_{glob}
Shekel [S5][b]	4	40	150	3368	−9.7320	−10.1532	−10.1532
Shekel [S7]	4	40	150	1782	−10.4024	−10.4029	−10.4029
Shekel [S10]	4	40	150	5620	−10.5109	−10.5109	−10.5364
Hartman [H3]	3	30	75	1114	−3.8625	−3.8628	−3.8628
Hartman [H6]	6	30	75	2341	−3.3072	−3.3224	−3.3224
Goldstein Price [GP]	2	20	50	420	3.0001	3.0000	3.0000
Branin [BR]	2	20	50	315	0.3980	0.3979	0.3979
Six Hump Camel [C6]	2	20	50	233	−1.0316	−1.0316	−1.0316
Shubert [SHU]	2	20	50	358	−186.7227	−186.7309	−186.7309

[a]The labels of the functions are given in brackets (Huyer and Neumaier, 1999).
[b]Two of the 25 runs do not converge to the global optimum.

lar in the neighborhood of the global optimum, or when the global optimum is far from the highly attractive local minimizers. Although the application of the local search to all points provides the best results, the improvement over applying local search only to the current best point is not significant. Therefore, for most of the problems, applying local search to all points may turn out to be overkill. In the subsequent experimentation, we utilized the third version of EM for comparison with other global optimization methods.

For the comparison, we used the standard test functions given by Dixon and Szegö (1978). The performance of different methods on these functions has been extensively studied by Huyer and Neumaier (1999). We use their results in comparing EM with the existing approaches. As a stopping criterion we use the following Eq. (5), that is also applied by Huyer and Neumaier (1999):

$$\frac{(f(x^{\text{best}}) - f_{\text{glob}})}{\left|f_{\text{glob}}\right|} \leqslant 10^{-4} \tag{5}$$

where f_{glob} represents the known global optimum.

Table 5 shows our results with the corresponding parameters. EM does not exhibit any difficulty to converge to the global optimum in all functions, except *Shekel* [S5]. This function has an attractive local minimizer, which is far from the global optimum.

Table 6 shows the performance comparison of EM with other methods. Except for the last row, the figures in the table are taken from Huyer and Neumaier (1999). From the table we see that MCS achieves the best results followed generally speaking by the other methods (labeled with (f)), which use first or second order information in the neighborhood search. EM does as well as or better than the most of the older methods indicated in the upper part of the table.

Overall, our additional comments are in order:

Table 6. Comparison of EM with different methods in terms of number of function evaluations

Method	S5	S7	S10	H3	H6	GP	BR	C6	SHU
Bremmerman	–[a]	–[a]	–[a]	–[a]	–[a]	–[a]	250		
Mod. Bremmerman	–[a]	–[a]	–[a]	–[a]	515	300	160		
Zilinskas	–[a]	–[a]	–[a]	8641			5129		
Gomulka-Branin	5500	5020	4860						
Törn	3679	3606	3874	2584	3447	2499	1558		
Gomulka-Törn	6654	6084	6144						
Gomulka-V.M.	7085	6684	7352	6766	11125	1495	1318		
Price	3800	4900	4400	2400	7600	2500	1800		
Fagiuoli	2514	2519	2518	513	2916	158	1600		
De Biase-Frontini	620	788	1160	732	807	378	587		
Mockus	1174	1279	1209	513	1232	362	189		
Bélisle et al.[b]				339	302	4728	1846		
Boender et al.[f]	567	624	755	235	462	398	235		
Snyman-Fatti[f]	845	799	920	365	517	474		178	
Kostrowicki-Piela[g]	–[c]	–[c]	–[c]	200	200	120		120	
Yao[f]								1132	$\leqslant$ 6000
Perttunen[f]	516	371	250	264		82	97	54	197
Perttunen-Stuckman[f]	109	109	109	140	175	113	109	96	–[a]
Jones et al.[h]	155	145	145	199	571	191	195	285	2967
Storn-Price –[d]	6400	6194	6251	476	7220	1018	1190	416	1371
MCS –[e,f]	83	129	103	79	111	81	41	42	69
EM	3368	1782	5620	1114	2341	420	315	233	358

[a] Method converged to a local minimum.
[b] Average number of function evaluations when converges. For H6, converged only 70% of time.
[c] Global minimum not found within 12 000 function calls.
[d] Average over 25 cases. For H6, average over 24 cases only; one case did not converge within 12 000 function calls.
[e] The version that gives the best results is selected.
[f] Recent methods that use first or second order information.
[g] Requires closed form for a particular integral.
[h] Partitions the search space into hyper-rectangles.

Missing entry means that no result is available from the literature(Huyer and Neumaier, 1999).

- Unlike MCS and other methods indicated with (f), EM does not use the first or the second order information.
- EM is able to provide answers for all the problems.
- All the methods, other than EM that are included in Table 6, have some limitations due to their rigid structures. EM has a very flexible design which would permit using the desirable features from other methods.

- We could have included results using other methods, which are discussed in Section 1.2. (such as Genetic Algorithms, Simulated Annealing, etc.). However our earlier results (Demirhan et al., 1999) with these methods on some of the problems above have shown that EM outperforms those methods.
- In both sets of test problems, we have observed that EM performs well when there is a pattern in the function. However, if there are multiple attractive local minimizers spread around the feasible region, EM may be deceived and the performance of it may decrease.
- The combination of the general scheme with other powerful local search methods, such as Solis and Wets (1981), may increase the efficiency of the algorithm.
- The computational results to date suggest that we could improve our results by using first or second order information, such as Fletcher and Reeves (1964), in the local search.

5. Conclusion and Further Research

In this paper we have developed a new heuristic, EM, which is a powerful yet easy algorithm for global optimization. We have applied the algorithm to different test problems in the literature. Without using the first or second order information, EM converges rapidly to optimum when the number of function evaluations is used as a performance measure.

EM can be used as a stand-alone approach or as an accompanying algorithm for other methods. The strength of the algorithm lies in the idea of directing the sample points toward local optimizers by utilizing an attraction-repulsion mechanism.

Further research will be pursued on using the first or second order information of the functions instead of Local* sub procedure. Also, application of other local search methods as discussed by Solis and Wets (1981) might produce better results. Especially, using an adaptive local search algorithm, which gets better precision in the close neighborhood of the minimizers is preferable.

Another interesting study might be merging this approach with some partitioning algorithms (Pinter, 1992), since shrinking the feasible region shows better convergence properties. Here the movement along the total force is allowed to be up to the boundaries. Nevertheless, keeping the step length smaller, may accelerate the algorithm.

In this paper we have tried the test functions with dimensions up to 20. There are other test functions in the literature which have higher dimensions. The performance of EM for those functions might also be studied.

* In some of the problems, it is not necessary to apply this sub procedure. However in certain cases this procedure improves the results critically.

Table 7. Levels of the factors used in experimental design

Factors	A (m)	B ($MAXITER$)	C (δ)
Level 1	5n	25n	1.0e-4
Level 2	10n	50n	1.0e-3
Level 3	20n	75n	1.0e-2

Our future research includes a thorough study of the convergence properties of EM. We intend to examine the stochastic process generated by the algorithm, and to elaborate the convergence in probability.

Acknowledgment

We thank the anonymous referee for suggesting the statistical estimation of gradient of f.

Appendix: Effects of Different Parameters

The proposed approach depends on several parameters, hence in this appendix we study the effects of different parameters by conducting a general factorial design with three factors.

We have selected the number of sample points (m), the maximum number of iterations ($MAXITER$), and the local search parameter (δ) as our factors (A, B and C respectively). We have not added the maximum number of local search iterations since in the local search procedure the big *while* loop ends as soon as a better neighbor solution is found (Algorithm 3, lines 14–17).

The levels of the factors are given in Table 7. We select the levels of the first two factors according to the dimension of the problem (n). In each combination of the levels we have used 10 replicates, thus totally 270 runs have been taken.

We have used the function *Sine Envelope* from the first test set and the function *Shekel* [S5] from the second set. We have selected these functions because the computational results have shown that the efficiency of EM on these functions is not as good as its efficiency on other functions in the corresponding test sets.

Tables 8 and 9 show that the significant factors of the model are A, B and AC. These results suggest that when we run the heuristic for a *long time* or when we *increase the number of points* in the population the chance of finding the global optimum increases. Besides, if we change the local search parameter, δ, with the maximum number of iterations the results improve.

Table 8. Analysis of variance table for the function *Sine Envelope*

	Sum of squares	Degrees of freedom	Mean square	*F* value
A	5.421e-3	2	2.71e-3	4.11
B	2.2e-2	2	1.1e-2	16.35
C	1.037e-3	2	5.186e-4	0.79
AB	3.474e-3	4	8.684e-4	1.32
AC	6.470e-3	4	1.617e-3	2.45
BC	1.9e-3	4	4.751e-4	0.72
ABC	3.013e-3	8	3.767e-4	0.57
Error	0.16	243	6.596e-4	
Total	0.20	269		

Table 9. Analysis of Variance Table for the function *Shekel* [S5]

	Sum of squares	Degrees of freedom	Mean square	*F* value
A	48.58	2	24.29	3.01
B	71.41	2	35.71	4.43
C	42.42	2	21.21	2.63
AB	49.42	4	12.35	1.53
AC	83.16	4	20.79	2.58
BC	30.58	4	7.65	0.95
ABC	126.83	8	15.85	1.97
Error	1958.36	243	8.06	
Total	2410.76	269		

References

Boender, C. G. E., Kan, A. H. G. R. and Timmer G. T. (1985), A stochastic approach to global optimization, *Computational Mathematical Programming (NATO ASI Series)*, F15: 291–308.

Cowan, E. W. (1968), *Basic Electromagnetism*, Academic Press, New York.

Demirhan, M., Özdamar, L., Helvacıoĝlu, L. and Birbil, Ş. İ. (1999), FRACTOP: A geometric partitioning metaheuristic for global optimization, *Journal of Global Optimization* 14: 415–435.

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

Fletcher, R. and Reeves, C. (1964), Function minimization by conjugate directions, *Computer Journal* 7: 149–154.

Glover, J. K. F. and Laguna, M. (1995), Genetic algorithms and tabu search: Hybrids for optimization. *Computers and Operations Research* 22: 111–134.

Hart, W. E. (1994), Adaptive Global Optimization with Local Search. *PhD Thesis*, University of California, San Diego, 1994.

Huyer, W. and Neumaier, A. (1999), Global optimization by multilevel coordinate search. *Journal of Global Optimization* 14: 331–355.

Kan, A. H. G. R. and Timmer, G. T. (1987), Stochastic global optimization methods, Part I: Clustering methods, *Mathematical Programming* 39: 27–56.

Kan, A. H. G. R. and Timmer, G. T. (1987), Stochastic global optimization methods Part II: Multi level methods, *Mathematical Programming* 39: 57–78.

Michalewicz, Z. (1994), *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin.

More, B. J. and Wu, Z. (1995), *Global Smoothing and Continuation for Large-scale Molecular Optimization*. Argonne National Laboratory, Illinois: Preprint MCS-P539-1095.

Neumaier, A. (2001), Global optimization web site. Computational mathematics group, University of Vienna, http://solon.cma.univie.ac.at/~neum/glopt.html, Austria.

Pinter, J. (1992), Convergence qualification of adaptive partitioning algorithms in global optimization, *Mathematical Programming* 56: 343–360.

Schoen, F. (1989), A wide class of test functions for global optimization, *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 51–60.

Solis, F. J. and Wets, R. J-B. (1981), Minimization by random search techniques, *Mathematics of Operations Research* 6: 19–30.

Törn, A., Ali, M. M. and Viitanen, S. (1999), Stochastic global optimization: Problem classes and solution techniques, *Journal of Global Optimization* 14: 437–447.

Törn, A. and Viitanen, S. (1994), Topographical global optimization using pre-sampled points, *Journal of Global Optimization* 5: 267–276.

Törn, A. and Žilinskas, A. (1989), *Global Optimization*, Springer, Berlin.