Journal of Global Optimization **30:** 301–318, 2004.

On the Convergence of a Population-Based Global Optimization Algorithm

Ş. İLKER BİRBİL[1], SHU-CHERNG FANG[2], RUEY-LIN SHEU[3]
[1]*Erasmus Research Institute of Management (ERIM), Erasmus University, Postbus 1738, Rotterdam, 3000 DR, The Netherlands (e-mail: sibirbil@few.eur.nl);*
[2]*Industrial Engineering and Operations Research, North Carolina State University, Raleigh, NC, 27695-7906, U.S.A. (e-mail: fang@eos.ncsu.edu);*
[3]*Department of Mathematics, National Cheng-Kung University, Tainan, 701, Taiwan, R.O.C. (e-mail: rsheu@mail.ncku.edu.tw)*

(Received 17 December 2002; accepted 6 January 2003)

Abstract. In global optimization, a typical population-based stochastic search method works on a set of sample points from the feasible region. In this paper, we study a recently proposed method of this sort. The method utilizes an attraction-repulsion mechanism to move sample points toward optimality and is thus referred to as electromagnetism-like method (EM). The computational results showed that EM is robust in practice, so we further investigate the theoretical structure. After reviewing the original method, we present some necessary modifications for the convergence proof. We show that in the limit, the modified method converges to the vicinity of global optimum with probability one.

Key words: stochastic search method, population-based algorithm, convergence with probability one

1. Introduction

Global optimization problems arise in many practical applications [12, 15]. Despite its importance and the efforts invested so far, the situation with respect to algorithm development for solving general global optimization problems is still not satisfactory.

To locate a global optimum among many local optima, various stochastic search methods have been proposed. Commonly used algorithms include simulated annealing [8], multilevel methods [9], evolutionary methods [11] and partitioning methods [18]. These methods utilize a stochastic mechanism to search for better bounds on an objective function to be optimized. Some of these methods may combine the search process with local refinements like hill-climbing or gradient-based methods [6].

One school of the stochastic search uses a single point to guide the search. A well-known example is the simulated annealing. Whereas another school uses a population of solutions and proceeds according to relative efficiencies of the observed functional values, like the genetic algorithms.

Recently, Birbil and Fang proposed a new population-based stochastic search algorithm [2, 3]. The method is called *electromagnetism-like method*, (EM), which utilizes an attraction-repulsion mechanism to move a population of points toward optimality.

This proposed method works on the nonlinear optimization problems with bounded variables in the following form:

$$\begin{aligned}&\min\ f(x)\\&\text{s. t. }\ x\in S\end{aligned}\tag{1}$$

where $f:\Re^n\rightarrow\Re$ is a nonlinear function and

$$S=\{x\in\Re^n\,|\,-\infty<l_k\leqslant x_k\leqslant u_k<\infty, k=1,...n\}\tag{2}$$

is a bounded *feasible region.*

The original EM was compared with other well-known methods and shown to have substantial performance [2]. However it lacks a convergence proof. In this paper, we show that a modified EM exhibits global convergence with probability one [3].

There are several convergence results for stochastic search methods used in global optimization [1, 4, 13]. In an early work, Yakowitz and Fisher studied a general framework for the conditions of convergence of stochastic search algorithms [19]. Rudolph has discussed a framework to study the convergence properties of an evolutionary algorithm (EA). He proposed a set of conditions under which a typical EA converges to the set of ε-optimal solutions with probability one [14].

A successful stochastic search method has to be carefully designed such that regardless of the starting point, there exists a nonzero probability to visit any subset (with a positive volume) of the feasible region. Rudolph's Lemma 1 confirms the necessity of this general principle. We have a similar lemma (Lemma 1 of Section 4) showing that a specially designed version of EM indeed exhibits this important property. With this result, the convergence result follows in Theorem 3.

The paper is organized as follows. In Section 2, the main procedures of EM are reviewed. The required modifications for convergence results are given in Section 3. The proof of convergence of the modified EM appears in Section 4. Some concluding remarks are given in the final section.

2. Review of Electromagnetism-like Method (EM)

We assume that for the problem in (1), the following parameters are given: the dimension of the problem (n), the pointer to the function ($f(x)$), and the lower and upper bounds (l_k, u_k) for $k=1,2,\cdots,n$. Since EM works on a set of sample-points (population), there is an additional predetermined parameter, m, which denotes the number of points in the population.

The general scheme of the method is given in Algorithm 1. In this scheme an *iteration* of the algorithm corresponds to one pass of the while loop. There exist four procedures: *Initialize*, *Local*, *CalcF*, and *Move*. The first procedure, *Initialize*, is used for sampling m points from the feasible region and assigning them their initial objective function values. *Local* is a neighborhood search procedure, which can be applied to one or many points for local refinement at each iteration. The selection of these two procedures do not affect the convergence result that we will present. Therefore, we omit the description of these two procedures and refer the readers to [2]. The remaining procedures *CalcF* and *Move* are carefully explained in the following subsections.

ALGORITHM 1. EM(m)

```
Initialize()
while termination criteria are not satisfied do
  Local()
  CalcF()
  Move()
end while
```

We adopt the notation, $x^i \in \Re^n$, to specify the i^{th} point of the population. Among these points at the current iteration, there is a particular one that has the best objective function value. We call this point the *current best point* and denote it by x^{best}.

2.1. CALCULATION OF TOTAL FORCE VECTOR (*calcf*)

In each iteration of the algorithm, a total force vector is calculated at each point. This vector determines the direction of movement for the corresponding point at the subsequent iterations.

The steps of the *CalcF* procedure are given in Algorithm 2. In order to compute the force between two points, we assign a charge-like value, q^i, to each point (line 2, Algorithm 2). The *charge* of the point is calculated according to the relative efficiency of the objective function values in the current population, i.e.,

$$q^i = \exp\left[-n \times \frac{f(x^i) - f(x^{best})}{\sum_{k=1}^{m} [f(x^k) - f(x^{best})]} \right], \quad i = 1, 2, \ldots, m. \tag{3}$$

In this way the points that have better objective function values possess higher charges. Though we attach no sign to this value, it determines the magnitude of attraction or repulsion for the corresponding point i.

This leads to the computation of the vector, F^i_j, between any pair of points x^i and x^j (line 8, Algorithm 2). The magnitude of this "component force" is inversely proportional to the Euclidean distance between the points and directly proportional to the product of their charges. Recall that there are no signs attached to the charges, but we decide the direction of the force between two points after comparing their objective function values. This means, between two points the one which has a better objective function value attracts, whereas the point with worse objective function value repels (lines 10 and 12 respectively, Algorithm 2). The computation is given by

$$F_j^i = \begin{cases} (x^j - x^i)\frac{q^i q^j}{\|x^j - x^i\|^2}, & if\ f(x^j) < f(x^i) \\ (x^i - x^j)\frac{q^i q^j}{\|x^j - x^i\|^2}, & if\ f(x^i) \leqslant f(x^j) \end{cases}, \quad i = 1, 2, \ldots, m. \tag{4}$$

Finally, the total force vector F^i exerted on each point is calculated by adding the individual component forces, i.e.,

$$F^i = \sum_{j \neq i}^{m} F_j^i, \quad i = 1, 2, \ldots, m. \tag{5}$$

2.2. MOVEMENT ALONG THE TOTAL FORCE VECTOR (*move*)

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 in equation (6). In this equation, λ the random step length, is assumed to be uniformly distributed between 0 and 1. RNG denotes the allowed range of movement toward the lower bound l^k, or the upper bound u^k, for the corresponding dimension. Furthermore, the force exerted on each particle is normalized so that the feasibility is maintained.

$$x^i = x^i + \lambda \frac{F^i}{\|F^i\|}(RNG), i = 1, 2, \ldots, m \text{ and } i \neq best. \tag{6}$$

ALGORITHM 2. CalcF()

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

```
            end if
        end if
    end for
end for
```

ALGORITHM 3. Move()

```
 1: for $i = 1$ to $m$ do
 2:     if $i \neq best$ then
 3:         $\lambda \leftarrow U(0,1)$
 4: $F^i \leftarrow \frac{F^i}{\|F^i\|}$
 5:         for $k = 1$ to $n$ do
 6:             if $F_k^i > 0$ then
 7:                 $x_k^i \leftarrow x_k^i + \lambda F_k^i (u_k - x_k^i)$
 8:             else
 9:                 $x_k^i \leftarrow x_k^i + \lambda F_k^i (x_k^i - l_k)$
10:             end if
11:         end for
12:     end if
13: end for
```

Algorithm 3 shows the steps of the procedure. Note that the best point, x^{best}, is not moved and is carried to the subsequent iterations (line 2, Algorithm 3).

2.3. TERMINATION CRITERIA

In [2] the EM procedure was terminated by using a predetermined maximum number of iterations. 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 a certain number of iterations, the algorithm may be stopped. In the literature several other stopping conditions have also been studied [7, 17].

In this paper, we show that a revised version of EM terminates with an “ε-optimal” solution when the maximum number of iterations is set to be large enough (this concept will be revisited in Section 4).

3. Modifications on the Original Method

Before introducing the modifications, we discuss the necessity of the changes by elaborating on “premature convergence”.

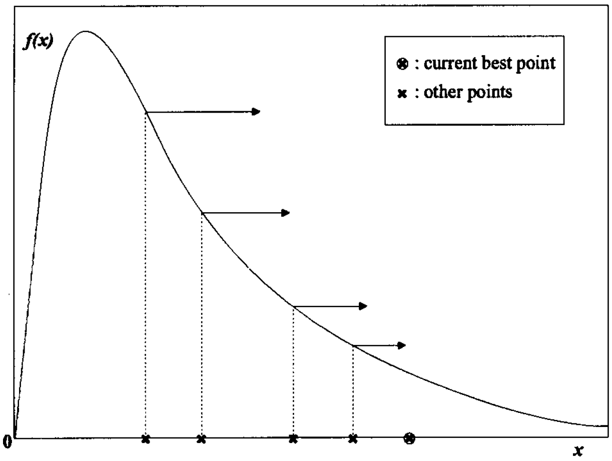

Figure 1. An example of premature convergence in one dimensional space.

3.1. PREMATURE CONVERGENCE

The premature convergence may occur when the forces exerted on the particles omit some parts of the feasible region. We illustrate this phenomenon by an example.

In Figure 1, the function has the optimum at 0, and it has a high peak that is close to this point. After the peak, the function is monotonically decreasing. If all the points in the current population were located on the right hand side of the peak, in the original EM all points would be directed toward the right, which would end up with a local minimizer, and the algorithm would converge prematurely.

In order to preclude premature convergence, we have to somehow "perturb" the current population so that at least one of the points will have a chance to move to the possibly omitted parts of the feasible region. Hence, one of the points in the population other than the current best point will be selected as the "perturbed point" and denoted by x^p. Next, the *CalcF* procedure is modified to take into account this perturbed point.

3.2. PERTURBED POINT AND MODIFIED *calcf*

The modified *CalcF* procedure is given in Algorithm 4. Note that a new parameter $\nu \in (0,1)$ is introduced, which will be described below.

The perturbed point, x^p is selected as the farthest point from the current best point, x^{best} in the current population (line 1, Algorithm 4), i.e.,

$$x^p = \operatorname{argmax}\{\|x^{best} - x^i\|, i = 1, 2, ..., m\}. \tag{7}$$

The calculation of the total force vector remains the same for all points except x^p. For x^p, the component forces are perturbed by a random number λ, i.e.,

$$F_j^p = \begin{cases} (x^j - x^p)\frac{\lambda q^p q^j}{\|x^j - x^p\|^2}, & if\ f(x^j) < f(x^p) \\ (x^p - x^j)\frac{\lambda q^p q^j}{\|x^j - x^p\|^2}, & if\ f(x^p) \leqslant f(x^j) \end{cases} \tag{8}$$

where λ is uniformly distributed between 0 and 1 (lines 12–13, Algorithm 4). Also, the directions of the component forces are perturbed; that is, if the random variable λ is less than the parameter ν then the direction of the component force is reversed (lines 14–16, Algorithm 4). Consequently, there exists one point in the population for which the direction of movement may be reversed.

We remark that Algorithm 4 may be replaced by a simpler procedure, which chooses any point from the population other than the x^{best} and perturbs it with a Gaussian distribution. In fact this is the most common way used by population-based algorithms [5]. However, we consider Algorithm 4 so that the modification is consistent with the main motivation of the proposed method, i.e., an attraction–repulsion mechanism is utilized for the modification as well. We also remark that any point other than the x^{best} can be selected as the perturbed point. We select the farthest point from the current best point, because intuitively the attractive force on this point due to x^{best} would be weaker than the attractive force on the other points. Hence, a perturbation on this point would be reasonable.

ALGORITHM 4. CalcF(ν)

```
 1: $x^p \leftarrow \mathrm{argmax}\{\|x^{best} - x^i\|, i = 1, 2, ..., m\}$
 2: **for** $i = 1$ to $m$ **do**
 3:   $q^i \leftarrow exp(-n\frac{f(x^i) - f(x^{best})}{\sum_{k=1}^{m}(f(x^k) - f(x^{best}))})$
 4:   $F^i \leftarrow 0$
 5: **end for**
 6: **for** $i = 1$ **to** $m$ **do**
 7:   **for** $j = 1$ **to** $m$ **do**
 8:     **if** $i \neq j$ **then**
 9:       **if** $x^i \neq x^p$ **then**
10:         $F_j^i \leftarrow (x^j - x^i)\frac{q^i q^j}{\|x^j - x^i\|^2}$
11:       **else**
12:         $\lambda \leftarrow U(0, 1)$
13:         $F_j^i \leftarrow (x^j - x^i)\frac{\lambda q^i q^j}{\|x^j - x^i\|^2}$
14:         **if** $\lambda < \nu$ **then**
15:           $F_j^i \leftarrow -F_j^i \{ReverseDirection\}$
16:         **end if**
17:       **end if**
18:       **if** $f(x^j) < f(x^i)$ **then**
19:         $F^i \leftarrow F^i + F_j^i \{Attraction\}$
```

```
      **else**
        $F^i \leftarrow F^i - F^i_j\{Repulsion\}$
      **end if**
    **end if**
  **end for**
**end for**
```

4. Convergence Results for the Modified EM

The steps of the modified EM is given in Algorithm 5. In this scheme N denotes the predetermined parameter, *maximum number of iterations*, i.e., the algorithm terminates when the total number of iterations exceeds N.

Notice that the neighborhood search procedure *Local* is not included in this revised version, since it does not effect the convergence proof. Our task is to show that in the long run (i.e., when $N \to \infty$) the modified method converges to the set of global optima with probability one. In the following sections, we give the convergence proof after introducing the notation.

ALGORITHM 5. EM(m, ν)

```
Initialize()
iteration ← 1
**while** iteration ⩽ N **do**
  CalcF($\nu$)
  Move()
  iteration ← iteration + 1
**end while**
```

4.1. NOTATION AND ASSUMPTIONS

Let $f:\Re^n \to \Re$ and $S \subset \Re^n$ be as in (1) and (2), respectively. Then $x^* \in \Re^n$ is called a *global minimum solution* on S, if

$$x^* \in S \text{ and } f(x^*) \leqslant f(x), \forall x \in S. \tag{9}$$

This leads to the definition of the set of points that are in the vicinity of global minimum. Given $\varepsilon > 0$, the *set of ε-optimal solutions* is defined by

$$B^*_\varepsilon = \{x \in S : |f(x) - f(x^*)| \leqslant \varepsilon\}. \tag{10}$$

In our derivation we make the following assumptions:

(1) B^*_ε contains an open ball of full dimensionality, i.e., $\mu(B^*_\varepsilon) > 0$, where μ is the Lebesgue measure on $\Re^n$.

(2) $f : S \to \Re$ is a lower bounded measurable function with respect to the Lebesgue measure μ.

(3) The collection of vectors (corresponding to the m ($m \geqslant n+1$) points in the current population) at every iteration generated by the algorithm has full rank, i.e.,

$$rank(\{x^1, x^2, ..., x^m\}) = n. \tag{11}$$

Recall that the proposed method (Algorithm 5) utilizes a stochastic search mechanism with a population of points. Thus, there exists an underlying stochastic process, which depends on the location of the m points in the feasible region S.

Formally, if we define Y_k as the random variable corresponding to the collection of m vectors at iteration k, then the stochastic process generated by the algorithm becomes the family of random variables $\{Y_k; k = 0, 1, 2, \cdots\}$. Also, the collection of these m vectors corresponds to the *state* of the process. Hence, we define the *state space* as

$$\mathbb{X}_m \triangleq \{\mathbf{x} : \mathbf{x} = (x^1, x^2, ..., x^m),\ x^i \in S,\ i = 1, 2, ..., m\}, \tag{12}$$

where $\mathbf{x}$ denotes a state. In this setting, the random variable Y_k gives the state of the process at iteration k.

Notice that, in the algorithm the location of the points at the next iteration depends only on the current population. Therefore, the stochastic process generated by the algorithm constitutes a time homogeneous Markov Chain [16]. This leads to the definition of the transition probability $\rho(\mathbf{x}, A)$; for any given $\mathbf{x} \in \mathbb{X}_m$, and $A \subset S$

$$\rho(\mathbf{x}, A) \triangleq P\{\chi_A(Y_{k+1}) \neq 0 \,|\, Y_k = \mathbf{x}\} \tag{13}$$

evaluates the conditional probability of making a transition from state $\mathbf{x}$ into a state that has *at least* one point in A, where

$$\chi_A(\mathbf{x}) \triangleq \sum_{i=1}^{m} 1_A(x^i) \tag{14}$$

gives the number of points in $A \cap \mathbf{x}$. $\mathbf{1}_A$ is the indicator function for set A, i.e., $\mathbf{1}_A(x^i)$ returns 1 if x^i is in A and returns 0 otherwise.

Remember that the main modification on the original method is the addition of the perturbed point, x^p. This point plays an essential role in the convergence proof, so we introduce additional notation related to x^p. Let $d \in \Re^n$ be any direction, then

$$L(x^p, d) \triangleq \{x(\eta) : x(\eta) = x^p + \eta d \in \Re^n,\ \eta \geqslant 0\} \tag{15}$$

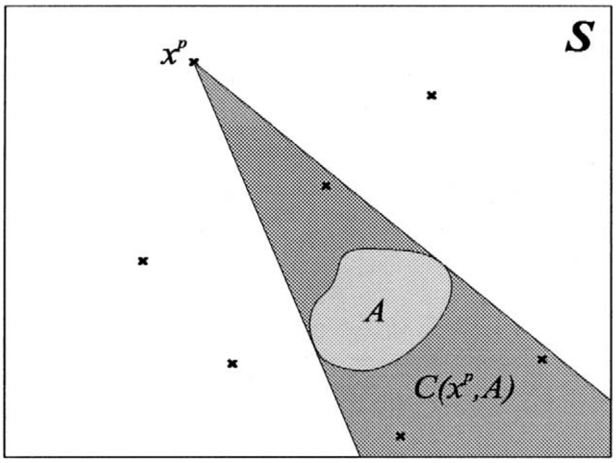

Figure 2. Truncated Cone in $\Re^2$.

denotes the *ray* originated at x^p and directed along d. Furthermore, for any given subset A of S, we define

$$C(x^p,A) \triangleq \{x : x \in L(x^p, y-x^p) \cap S, \text{ for some } y \in A\} \tag{16}$$

as the *truncated cone* pointed at x^p (Figure 2).

4.2. CONVERGENCE WITH PROBABILITY ONE

Before we present the convergence proof, let us first define the concept of convergence for the modified method. Given $\varepsilon > 0$, if there exists an integer $K(\varepsilon) \geqslant 0$ such that

$$\chi_{B_\varepsilon^*}(Y_k) \neq 0, \ \forall k > K(\varepsilon), \tag{17}$$

then the modified EM (Algorithm 5) is said to *converge* to the set of ε-optimal solutions.

4.3. CONVERGENCE PROOF

Recall the definition of $\rho(\mathbf{x}, A)$ in (13), we further define

$$\rho^*(A) \triangleq \inf_{\mathbf{x} \in \mathbb{X}_m} \{\rho(\mathbf{x}, A)\} \tag{18}$$

for a given subset A of S.

LEMMA 1. *Given $A \subset S$, if A contains an open ball of full dimensionality in S, then*

$$\rho^*(A) > 0. \tag{19}$$

Proof. Let $\mathbf{x} \in \mathbb{X}_m$ and without loss of generality let $m=n+1$. We want to show that at each iteration at least one point from $\mathbf{x}$ has a positive probability to move into set A.

By assumption 3, when we focus on the perturbed point x^p, the component forces F_j^p become a basis. Using a combination of the component forces, x^p may be moved into A in three steps:

(1) Some of the directions of the component forces are reversed (Algorithm 4, lines 14-16).
(2) A direction vector that falls into $C(x^p, A)$ is generated (Figure 2).
(3) An appropriate step length to displace x^p into A is calculated.

Next we show that there exist nonzero lower bounds on the probabilities of these steps.

Step 1. By hypothesis, there exists a ball $B_r \subset A$ with its radius $r > 0$ being small enough such that B_r lies in the nonnegative combination of the component forces at x^p. Let $\kappa_1(\mathbf{x})$ denote the probability for reversing the directions of some of the component forces so that B_r lies in the nonnegative combination of the component forces. Note that in Algorithm 4 (line 14) the probability of reversing each component force is $\nu \in (0,1)$. Therefore, the lower bound on the probability of Step 1 is

$$0 < \kappa_1^* \triangleq \min\{(1-\nu)^n, \nu^n\} \leqslant \kappa_1(\mathbf{x}). \tag{20}$$

Step 2. Let B_δ be a closed ball with radius $\delta > 0$ in B_r, i.e.,

$$B_\delta \subset B_r \subset A. \tag{21}$$

The proper selection of B_δ will become apparent shortly.
Let $H(\mathbf{x})$ be the set generated by the component forces, F_j^p, i.e.,

$$H(\mathbf{x}) \triangleq \{x : x = \sum_{j \neq p}^{m} \gamma_i F_j^p, \gamma_i \in [0,1]\}. \tag{22}$$

Equation (8) shows that the component forces are perturbed with uniformly distributed numbers between 0 and 1. As illustrated in Figure 3, let

$$T(\mathbf{x}) \triangleq \mu(H(\mathbf{x}) \cap C(x^p, B_\delta)) \tag{23}$$

then the probability of generating a total force vector that lies in $C(x^p, B_\delta)$ is

$$\kappa_2(\mathbf{x}) \triangleq \frac{\mu(T(\mathbf{x}))}{\mu(H(\mathbf{x}) \cap S)}. \tag{24}$$

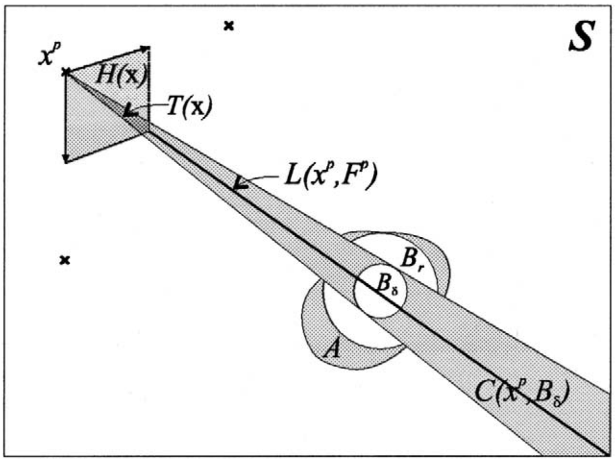

Figure 3. Calculation of the probability for Step 2 in $\Re^2$.

The value of $\kappa_2(\mathbf{x})$ decreases as the denominator increases and the numerator decreases. The upper bound of the denominator is the volume of S, i.e.,

$$\mu(H(\mathbf{x})\cap S)\leqslant\prod_{k=1}^{n}(u_k-l_k). \tag{25}$$

In (24), the volume of $T(\mathbf{x})$ depends on the volume of the $H(\mathbf{x})$, and the distance between B_δ and x^p. Let,

$$\alpha\triangleq\inf_{x\in B_\delta}\{\|x-x^p\|\}. \tag{26}$$

be the distance between B_δ and x^p. Since S is bounded, we have

$$\alpha\leqslant\alpha^*\triangleq\sup_{x,y\in S}\{\|x-y\|\}. \tag{27}$$

Also note that by (22), the volume of $H(\mathbf{x})$ decreases as the lengths of the component force vectors decrease. Let us define,

$$\beta\triangleq\min_{j\neq p}\|F_j^p\|. \tag{28}$$

By Equation (8) we have

$$\|F_j^p\|=\frac{q^p q^j}{\|x^j-x^p\|}, \tag{29}$$

where the charges of the points are evaluated as in equation (3). Note that in (3), the fraction in the exponential function is between 0 and 1. Hence, the lower bound on the charge of any point is $q^*\triangleq e^{-n}$. By using (27), we have

$$0<\beta^*\triangleq\frac{(q^*)^2}{\alpha^*}\leqslant\beta. \tag{30}$$

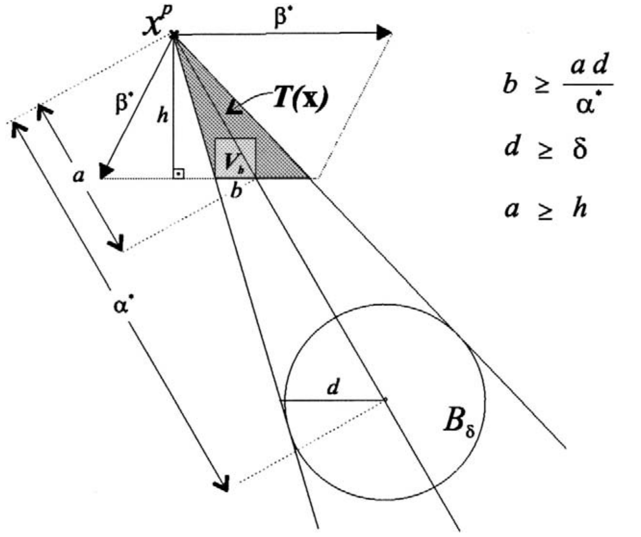

Figure 4. Calculation of the lower bound for $\mu(T(\mathbf{x}))$ in $\Re^2$.

In Figure 4, the distance between x^p and the center of B_δ is replaced by its over-estimate α^* and similarly the length of the component forces is illustrated with its under-estimate β^* so that the derivation of the lower bound for the numerator of (24) can be shown. As indicated in the figure, we can select δ small enough such that a cube with side b, V_b resides in $T(\mathbf{x})$. Note that this result can be further generalized to the n dimensional case as

$$b^n = \mu(V_b) \leqslant \mu(T(\mathbf{x})). \tag{31}$$

Next we show that b is bounded away from 0. After the cube is selected, a, d and h are defined as in Figure 4. From the figure, we see $a \geqslant h$, $d \geqslant \delta$ and

$$b \geqslant \frac{ad}{\alpha^*}. \tag{32}$$

With δ sufficiently small, the Pythagoras theorem implies

$$h^2 + (\beta^* - b)^2 \geqslant (\beta^*)^2 \tag{33}$$

$$a \geqslant h \geqslant \sqrt{2b\beta^* - b^2} \tag{34}$$

Substituting the lower bounds of a and d into (32) gives

$$b \geqslant \frac{\delta\sqrt{2b\beta^* - b^2}}{\alpha^*} \tag{35}$$

Hence,

$$b^2 \geqslant \frac{\delta^2(2\beta^* b - b^2)}{(\alpha^*)^2} \tag{36}$$

$$(\alpha^*)^2 b^2 \geqslant 2\beta^* b\delta^2 - b^2\delta^2 \tag{37}$$

$$b((\alpha^*)^2 + \delta^2) \geqslant 2\beta^*\delta^2 \tag{38}$$

$$b \geqslant \frac{2\beta^*\delta^2}{(\alpha^*)^2 + \delta^2}. \tag{39}$$

Consequently, the lower bound on b becomes

$$b^* \triangleq \frac{2\beta^*\delta^2}{(\alpha^*)^2 + \delta^2}. \tag{40}$$

This further leads to

$$0 < (b^*)^n \leqslant \mu(V_b) \leqslant \mu(T(\mathbf{x})). \tag{41}$$

Therefore, the lower bound on the probability of Step 2 is

$$0 < \kappa_2^* \triangleq \frac{(b^*)^n}{\prod_{k=1}^{n}(u_k - l_k)} \leqslant \kappa_2(\mathbf{x}). \tag{42}$$

Step 3. Let $L(x^p, F^p)$ be the ray originated at x^p and directed along F^p (Figure 3). In Algorithm 3 the perturbed point is moved along the total force vector by a uniformly distributed step length (between 0 and up/down to the boundaries). Hence, let

$$L(x^p, F^p) \cap \partial B_r \triangleq \{y_1, y_2\} \text{ and } L(x^p, F^p) \cap \partial S \triangleq \{z\}, \tag{43}$$

where ∂B_r and ∂S denote the boundaries of the B_r and S, respectively. Then, the probability of moving into A is

$$\kappa_3(\mathbf{x}) \triangleq \frac{\|y_1 - y_2\|}{\|x^p - z\|}. \tag{44}$$

As illustrated in Figure 5, if we define

$$r^* \triangleq \sqrt{r^2 - \delta^2}, \tag{45}$$

then

$$0 < 2r^* \leqslant \|y_1 - y_2\|. \tag{46}$$

Therefore, by using (27) the lower bound on the probability of Step 3 is

$$0 < \kappa_3^* \triangleq \frac{2r^*}{\alpha^*} \leqslant \kappa_3(\mathbf{x}). \tag{47}$$

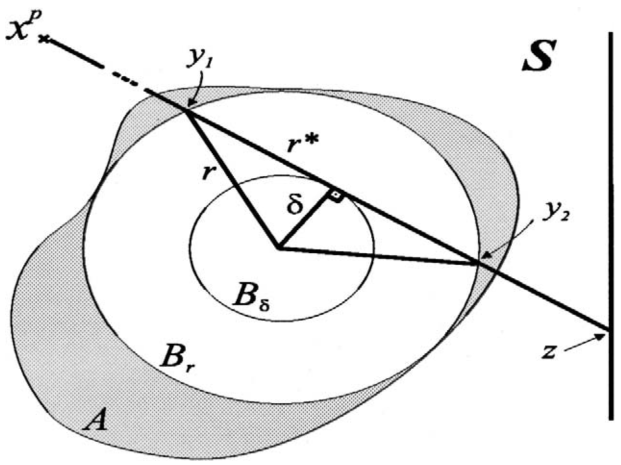

Figure 5. The calculation of r^* in $\Re^2$.

Therefore we have shown that

$$0 < \kappa_1^* \kappa_2^* \kappa_3^* \leqslant \rho^*(A). \tag{48}$$

This completes the proof of the lemma. □

Assumption 1 ensures that B_ε^* contains an open ball of full dimensionality. Therefore, there exists a nonzero probability for the population at any given iteration to move into the set of ε-optimal solutions in one iteration.

The next lemma shows that at any iteration, if one of the points is in B_ε^* then at subsequent iterations there always exists at least one point residing in B_ε^*. Intuitively, this reflects an *absorbing event* for the algorithm.

LEMMA 2. *Given any state* $\mathbf{x} \in \mathbb{X}_m$ *and* $k \geqslant 0$ *if* $\chi_{B_\varepsilon^*}(Y_k) \neq 0$ *then* $P\{\chi_{B_\varepsilon^*}(Y_{k+1}) \neq 0 \,|\, Y_k = \mathbf{x}\} = 1$.

Proof. Suppose at iteration k,

$$Y_k = \mathbf{x} \text{ and } \chi_{B_\varepsilon^*}(Y_k) \neq 0, \tag{49}$$

then we know that $x^{best} \in B_\varepsilon^*$. Algorithm 3 (line 2) ensures that unless another point observes a better objective function value than that of x^{best}, the current best point in B_ε^* remains at iteration $k+1$. If not, then x^{best} is replaced by this new point, which again resides in B_ε^*. Thus,

$$\chi_{B_\varepsilon^*}(Y_{k+1}) \neq 0. \tag{50}$$

This completes the proof of the lemma. □

We are now ready to prove that the modified EM converges to the set of ε-optimal solutions with probability one.

THEOREM 1. *Provided that the assumptions in Section* 4.1 *hold, Algorithm* 5 *converges to B^*_ε with probability one, i.e.,*

$$\lim_{k\uparrow\infty} P\{\chi_{B^*_\varepsilon}(Y_k)\neq 0\}=1. \tag{51}$$

Proof. The Markovian property of the stochastic process and Lemma 2 imply that

$$\begin{aligned} P\{\chi_{B^*_\varepsilon}(Y_k)=0\} &= P\{\chi_{B^*_\varepsilon}(Y_1)=0,\chi_{B^*_\varepsilon}(Y_2)=0,\cdots,\chi_{B^*_\varepsilon}(Y_k)=0\} \\ &= P\{\chi_{B^*_\varepsilon}(Y_1)=0\}\prod_{l=2}^{k} P\{\chi_{B^*_\varepsilon}(Y_l)=0\,|\,\chi_{B^*_\varepsilon}(Y_{l-1})=0\} \end{aligned} \tag{52}$$

Since we have a time homogeneous Markov Chain, it is sufficient to compute only

$$P\{\chi_{B^*_\varepsilon}(Y_l)=0\,|\,\chi_{B^*_\varepsilon}(Y_{l-1})=0\}. \tag{53}$$

Let us define

$$D=\{\mathbf{x}:\chi_{B^*_\varepsilon}(\mathbf{x})=0\}, \tag{54}$$

then we have,

$$\begin{aligned} P\{\chi_{B^*_\varepsilon}(Y_l)=0\,|\,\chi_{B^*_\varepsilon}(Y_{l-1})=0\} &= \frac{P\{\chi_{B^*_\varepsilon}(Y_{l-1})=0,\chi_{B^*_\varepsilon}(Y_{l-1})=0\}}{P\{\chi_{B^*_\varepsilon}(Y_l)=0\}} \\ &= \frac{\int_D P\{\chi_{B^*_\varepsilon}(Y_l)=0|Y_l=\mathbf{y}\}P\{Y_{l-1}=\mathbf{y}\}\mu(d\mathbf{y})}{\int_D P\{Y_{l-1}=\mathbf{y}\}\mu(d\mathbf{y})} \end{aligned} \tag{55}$$

Following (18), let $\rho^*(B^*_\varepsilon)=\rho^*$, and by Lemma 1, we have

$$P\{\chi_{B^*_\varepsilon}(Y_l)=0\,|\,Y_{l-1}=\mathbf{y}\}\leqslant(1-\rho^*),\ \forall\mathbf{y}\in D. \tag{56}$$

Hence (55) becomes

$$\begin{aligned} P\{\chi_{B^*_\varepsilon}(Y_l)=0\,|\,\chi_{B^*_\varepsilon}(Y_{l-1})=0\} &\leqslant \frac{(1-\rho^*)\int_D P\{Y_{l-1}=\mathbf{y}\}\mu(d\mathbf{y})}{\int_D P\{Y_{l-1}=\mathbf{y}\}\mu(d\mathbf{y})} \\ &= (1-\rho^*). \end{aligned} \tag{57}$$

Substituting (57) into (52), we get

$$P\{\chi_{B^*_\varepsilon}(Y_k)=0\}\leqslant(1-\rho^*)^k. \tag{58}$$

By Lemma 2, we have

$$\begin{aligned} \lim_{k\uparrow\infty} P\{\chi_{B^*_\varepsilon}(Y_k)\neq 0\} &= 1-\lim_{k\uparrow\infty} P\{\chi_{B^*_\varepsilon}(Y_k)=0\} \\ &\geqslant 1-\lim_{k\uparrow\infty}(1-\rho^*)^k=1. \end{aligned} \tag{59}$$

This completes the proof of the theorem. □

4.4. COMPUTATIONAL CONSIDERATIONS

The main theorem shows that Algorithm 5 eventually reaches an ε-optimal solution. If we consider the output of the algorithm in each iteration as a sequence of Bernoulli trials, then the average number of failures before the first success can be calculated [10]. By Lemma 1, the probability of success is $\rho^*(B^*_\varepsilon)$. Therefore the average number of failures before the first success is $(1-\rho^*(B^*_\varepsilon))/\rho^*(B^*_\varepsilon)$.

Again for computational study, we need to discuss two possible overflow problems. First, recall that the charge of each point is calculated by (3). Hence, if the objective function attains very high values, the fraction may become too small and cause an overflow problem in calculating the exponential function. This problem can be avoided by assigning a large floating point value (depending of the word length of the particular computer) to the points with very high objective function values. Second, if the distance between two points is close to zero, then there may be an overflow problem due to the denominator of the fractions in equations (4) and (8). This can be also avoided by setting a small enough number to the distance between these points according to the word length of the computer.

5. Conclusion

In this paper, a convergence property of the recently proposed *electroma-gnetism-like method*, (EM) has been studied [2]. Our main task has been to show that when the number of iterations is *large enough*, one of the points in the current population moves into the ε-neighborhood of the global optimum. In order to achieve this result, we have given a detailed mathematical construction, which could be easily applied to some of the other population-based stochastic search algorithms.

The proposed method deals with global optimization problems with bound constraints only. For further research, we will consider the method for optimization problems with general constraints. We are also interested in developing a discrete counterpart for combinatorial optimization problems.

Acknowledgment

We thank the anonymous referee for referring us to the article of Rudolph and for other useful comments.

References

1. Baba, N. (1981), Convergence of a random optimization method for constrained optimization problems, *Journal of Optimization Theory and Applications* 33, 451–461.
2. Birbil, Ş. İ. and Fang, S.-C. (2002), An electromagnetism-like mechanism for global optimization, *Journal of Global Optimization*, 25(3), 263–282.
3. Birbil, Ş. İ. (2002), Stochastic global optimization techniques, *PhD Thesis*, North Carolina State University, Raleigh, NC.

4. Dekkers, A. and Aarts, E. (1991), Global optimization and simulated annealing, *Mathematical Programming* 50, 367–393.
5. Fogel, D. B. (1994), Asymptotic convergence properties of genetic algorithms and evolutionary programming, *Cybernetics and Systems* 25(3), 389–407.
6. Hart, W. E. (1994), Adaptive global optimization with local search, *PhD Thesis*, University of California, San Diego, CA.
7. Huyer, W. and Neumaier, A. (1999), Global optimization by multilevel coordinate search, *Journal of Global Optimization* 14, 331–355.
8. Ingber, L. (1994), Simulated annealing: practice versus theory, *Journal of Mathematical Computation Modeling* 18, 29–57.
9. Kan, A. H. G. R. and Timmer, G. T. (1987), Stochastic global optimization methods Part II: Multi level methods, *Mathematical Programming* 39, 57–78.
10. Law, A. M. and Kelton, W. D. (2000), *Simulation Modeling and Analysis*, 3rd Edition, McGraw-Hill, Inc., New York.
11. Michalewic, Z. (1994), *Genetic Algorithms + Data Structures = Evolution Programs*, Springer, Berlin.
12. Moré, J. J. and Wu, Z. (1995), Global smoothing and continuation for large-scale molecular optimization, *Argonne National Laboratory*, Illinois: Preprint MCS-P539-1095.
13. Pintér, J. (1984), Convergence properties of stochastic optimization procedures, *Optimization* 15, 405–427.
14. Rudolph, G. (1996), Convergence of evolutionay algorithms in general search spaces. In: *International Conference on Evolutionary Computation*, pp. 50–54.
15. Schoen, F. (1989), A wide class of test functions for global optimization. In: *Proceedings of the 3rd International Conference on Genetic Algorithms*, pp. 51–60.
16. Taylor, H. M. and Karlin, S. M. (1998), *An Introduction to Stochastic Modeling*, Academic Press, San Diego, CA.
17. Törn, A., Ali, M. M. and Viitanen, S. (1999), Stochastic global optimization: Problem classes and solution techniques, *Journal of Global Optimization* 14, 437–447.
18. Wood, G. R. (1991), Multidimensional bisection and global optimization, *Computer and Mathematics with Applications* 21, 161–172.
19. Yakowitz, S. J. and Fisher, L. (1973), On sequential search for the maximum of an unknown function, *Journal of Mathematical Analysis and Applications* 41, 234–259.