J Glob Optim (2009) 44:227–234
DOI 10.1007/s10898-008-9320-z

A novel hybrid approach combining electromagnetism-like method with Solis and Wets local search for continuous optimization problems

M. Gol Alikhani · N. Javadian · R. Tavakkoli-Moghaddam

Received: 5 August 2007 / Accepted: 12 June 2008 / Published online: 8 July 2008

Abstract The electromagnetism-like method (EM) is a meta-heuristic algorithm utilizing an attraction-repulsion mechanism to move sample points towards optimality in continuous optimization problems. Traditionally, the EM uses two algorithms known as the original and revised EMs. This paper presents a novel hybrid approach for EM by employing a well-known local search, called Solis and Wets. To show the performance of our proposed hybrid EM, a number of experiments are carried out on a set of well-known test problems and the related results are compared with two forgoing algorithms.

Keywords Electromagnetism-like method · Solis and Wets localsearch · Continuous optimization

1 Introduction

In recent years, global optimization has become a rapidly developing field and many stochastic search methods have been proposed in order to find a global optimum among many local optima. One of these methods has recently been proposed by Birbil and Fang [1] and is known as electromagnetism-like method (EM). This algorithm uses an attraction–repulsion mechanism to move sample points toward optimality. This mechanism is similar to the electromagnetism theory for charged particles [2]. The EM applies to optimization problems with continuous variables in the following form, as given in Eqs. 1 and 2.

$$\underset{\text{s.t.}}{\text{Min}} \quad f(x) \tag{1}$$

M. Gol Alikhani (✉) · N. Javadian
Department of Industrial Engineering, Mazandaran University of Science and Technology, Babol, Iran
e-mail: alikhani_mohsen1@yahoo.com

R. Tavakkoli-Moghaddam
Department of Industrial Engineering, College of Engineering, University of Tehran, Tehran, Iran

$$x \in [L, U], \tag{2}$$
$$[L, U] := \{x \in R^n | L_k \leq x_k \leq U_k; k = 1, 2, \ldots, n\}$$

The original EM was revised by Birbil et al. [3] in order to make it convergent and after some modifications in the original EM, they proved that their new revised EM exhibits global convergence with probability one. Thus, there are two algorithms, namely the original and revised EMs. The benefits of each algorithm are as follows:

- Original EM: The results of this algorithm are better and more satisfactory than the revised algorithm (shown in Sect. 5).
- Revised EM: This algorithm exhibits global convergence with probability one, which is proved in [3].

In this paper, we propose a novel hybrid approach by combining the revised EM with a strong local search method, known as Solis and Wets [4]. The benefits of our new proposed hybrid EM are as follows:

- The results obtained by this proposed hybrid EM are better and more satisfactory than other two forgoing algorithms, the original and revised ones. We show this fact by applying our proposed hybrid EM to a number of test problems.
- Our proposed approach uses the modifications of the revised EM. Thus, it exhibits global convergence with probability one, the same as the revised EM.

This paper is organized as follows: In Sect. 2, the original EM is reviewed. We review the revised EM in Sect. 3 and then we propose our new hybrid EM approach in Sect. 4. In Sect. 5, we compare the computational results of these three algorithms on a set of test problems. Finally, the conclusions are presented in Sect. 6.

2 The original EM

Figure 1 shows the original EM, namely Algorithm 1.

The full description for each step of Algorithm 1 and their procedures are referred to [1]; however, we summarize each step to be more acquainted with this algorithm.

- *Initialize:* In this step, m sample points are selected at random from the feasible region, which is an n-dimensional hyper-cube. Then, the objective function value (OFV) of each

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, $\delta \in [0, 1]$

```
Initialize()
iteration ← 1
while iteration < MAXITER do
    Local(LSITER, δ)
    F ← CalcF( )
    Move(F)
    iteration ← iteration + 1
end while
```

Fig. 1 Pseudo code of the original EM sample point is computed. This step ends with m points identified, and the point that has the best OFV is stored in x^{best}.

- *Calculate force:* In this step, a charged-like value is assigned to each point (q^i). The charge of a point is computed according to the efficiency of the OFV of that point (points with better OFV have more charge than other points). The charges are computed by Eq. 3.

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

Then, the force between two points is computed using a mechanism that is similar to electromagnetism theory for the charged particles. In this mechanism, 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 and a point that has a better OFV (i.e., bigger q^i) attracts the other point and the point with the worse OFV repels the other. The computation of this force is given by Eq. 4.

$$F_j^i = \begin{cases} \left(x^j - x^i\right) \dfrac{q^i q^j}{\left\|x^j - x^i\right\|^2}, & \text{if } f\left(x^j\right) < f\left(x^i\right) \\ \left(x^i - x^j\right) \dfrac{q^i q^j}{\left\|x^j - x^i\right\|^2}, & \text{if } f\left(x^i\right) \leq f\left(x^j\right) \end{cases}, \quad 1 = 1, 2, \ldots, m \tag{4}$$

At the end of this step, the vector of the total force exerted on each point from other points is computed. This vector determines the direction of movement for corresponding point in Step "move (**F**)" of Algorithm 1 as shown in Fig. 1.

Move points along the total force vector: In this step, points are moved along the total force vector that is computed in the previous step. The movement is according to Eq. 5. In this equation, λ the random step length is uniformly distributed between 0 and 1. RNG denotes the allowed range of movement toward the lower or upper bound for the corresponding dimension.

$$x^i = x^i + \lambda \frac{F^i}{\left\|F^i\right\|} (RNG), \quad i = 1, 2, \ldots, m \text{ and } 1 \neq best \tag{5}$$

Local search: This step is used to move the sample points toward the local minimums that are near them. In this step, points are pushed toward the local valleys using a neighborhood search procedure. The local search method used in this algorithm is very simple. Powerful local search methods (e.g., Solis and Wets [4]) are not used in this algorithm. The procedure of each step of the algorithm is referred to [1]; however, regarding the importance of the local search step, we describe it in Fig. 2, namely Algorithm 2. For detailed description of each step of the above algorithm, please see [1].

3 The revised EM

Birbil et al. [3] showed that the original EM may converge prematurely and end up with a local minimizer in some problems. In order to preclude this premature convergence, they applied some modifications in Step "Calc**F**( )" of the original EM (Algorithm 1) and proposed a new revised EM, namely Algorithm 3, as explained in Fig. 3.

In the revised EM, one of the points other than the current best point is considered as the "perturbed" point. This perturbed point, x^p, is selected as the farthest point from the

Fig. 2 Pseudo code of the local search used in Algorithm 1

ALGORITHM 2: Local($LSITER$, δ)

m:	number of sample points
n:	number of dimensions of the space
$LSITER$:	maximum number of local search iterations
δ:	local search parameter, $\delta \in [0, 1]$

```
counter ←1
Length←δ(max k{u_k − l_k})
for i = 1 to m do
    for k = 1 to n do
        λ1 ←U (0, 1)
        while counter <LSITER do
            y ←x^i
            λ2 ←U (0, 1)
            if λ1 > 0.5 then
                y^k ←y^k + λ2 (Length)
            else
                y^k ←y^k −λ2 (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}
```

ALGORITHM 3: Revised EM(m, $MAXITER$)

m:	number of sample points
$MAXITER$:	maximum number of iterations

```
Initialize()          (the same as original EM)
iteration ←1
while iteration <MAXITER do
    F←CalcF( )   (with some modifications in original EM)
    Move(F)      (the same as original EM)
    iteration ←iteration + 1
end while
```

Fig. 3 Pseudo code of Algorithm 3

current best point, x^{best}, in the current population. The force exerted to the perturbed point is computed according to Eq. 6, in which parameter λ is uniformly distributed between 0 and 1.

$$F_j^p = \begin{cases} \left(x^j - x^p\right) \dfrac{\lambda q^p q^j}{\left\|x^j - x^p\right\|^2}, & \text{if } f\left(x^j\right) < f\left(x^p\right) \\ \left(x^p - x^j\right) \dfrac{\lambda q^p q^j}{\left\|x^j - x^p\right\|^2}, & \text{if } f\left(x^p\right) \leq f\left(x^j\right) \end{cases} \tag{6}$$

In the revised EM, the direction of the component forces exerted to x^p is also perturbed. That is, if parameter λ is less than parameter $\nu \in (0, 1)$, then the direction of the component force is reversed. After these modifications, Birbil et al. [3] showed that the perturbed point have a chance to move to the possibly omitted parts of the feasible region and their new revised EM exhibits global convergence with probability one.

Since the local search step does not effect on the convergence proof, Birbil et al. [3] omitted this step from Algorithm 1 for analytic convenience of their convergence proof. Fig. 3 shows a pseudo code of the revised EM (Algorithm 3). For detailed description of each step of this algorithm, please see [3].

4 Our novel hybrid EM

As we discussed earlier, there are two algorithms for electromagnetism-like method (EM). Each algorithm has an important advantage as follows:

- Original EM: The results of this algorithm are better and more satisfactory than the revised EM, as shown in Sect. 5 of this paper.
- Revised EM: This algorithm exhibits global convergence with probability one, in which it is proved in [3].

In this section, we propose a novel hybrid EM by applying the revised EM embedded with a powerful local search method, known as Solis and Wets. Our proposed hybrid EM has two main advantages as follows:

- It is convergent, whose proof is the same as the revised EM given in [3]. The reason, which we can use the same proof for our proposed algorithm, is because we use those modifications in Step "Calc**F**( )" of the revised EM. These modifications explained in Sect. 3 ensure when the number of iterations is large enough, regardless of the starting population, there exists a nonzero probability to visit any subset of the feasible region and one of the points in the current population moves into the $\acute{\varepsilon}$-neighborhood of the global optimum.
- The results reported by this algorithm are better and more satisfactory than the other two forgoing algorithms, as shown in Sect. 5.

Figure 4 shows a pseudo code of our new proposed hybrid EM, namely Algorithm 4.

The local search method used in our proposed hybrid EM is known as the Solis and Wets method. Solis and Wets [4] described a class of local and global search algorithms with proofs

ALGORITHM 4: Our proposed hybrid EM (*m*, *MAXITER*, *LSITER*, ρ, *MAXF*, *MAXS*, *expf*, *conf*)

m:	number of sample points
MAXITER:	maximum number of iterations
LSIT ER:	maximum number of local search iterations
ρ, ***MAXF***, ***MAXS***, ***expf***, ***conf***:	local search parameters

```
Initialize()          (the same as original and revised EMs)
iteration ←1
while iteration <MAXITER do
    Local(LSITER, ρ, MAXF, MAXS, expf, conf)  (with Solis and Wets method)
    F←CalcF( )        (the same as the revised algorithm)
    Move(F)           (the same as the original and revised EMs)
    iteration←iteration + 1
end while
```

Fig. 4 Pseudo code of Algorithm 4

```
ALGORITHM 5: Solis and Wets(LSITER, ρ, MAXF, MAXS, expf, conf)
LSITER:   maximum number of local search iterations
ρ:        Local search parameter
MAXF:     maximum number of failures for decreasing ρ
MAXS:     maximum number of Successes for increasing ρ
expf:     Exponential factor for increasing ρ
conf:     Contraction factor for  decreasing ρ
choose initial point x
set b (bias vector with dimensionality of search space) to 0
while LSITER not exceeded and ρ not too small  do
   for each dimension i of the solution space  do
      add deviate D_i = normal deviate with mean b_i and standard deviation ρ
   End for
   If new solution is better and new solution is feasible  then
      failures =0
      successes = successes +1
       b = 0.4D + 0.2b
   else
       for each dimension I do
          add –D_i to the original solution
       End for
       If new solution is better and new solution is feasible  then
           failures =0
           successes = successes +1
            b = b - 0.4D
       else
           failures = failures +1
           successes = 0
           b = 0.5b
       End if
   End if
   If successes >= MAXS  then
       failures = 0
       successes = 0
       ρ= expf * ρ
   End if
   If failures >= MAXF  then
       failures = 0
      successes = 0
      ρ= conf * ρ
   End if
End while
```

Fig. 5 Pseudo code of Algorithm 5

of convergence in the limit of infinite search time. This local search method used in this paper is a randomized hill climber with an adaptive step size. Each step starts with a current point x. A deviate d is chosen from a distribution whose standard deviation is given by a parameter ρ. If $x - d$ or $x + d$ is better (and also a feasible solution), a move is made to the better point and a "success" is recorded. Otherwise, a "failure" is recorded. After several successes in a row, ρ is increased to move more quickly. After several failures in a row, ρ is decreased to focus the search. Additionally, a bias term is included to give the search momentum in directions that yield success. Figure 5 shows a pseudo code for our implementation of the Solis and Wets local search, namely Algorithm 5.

It is worth noting that either $x - d$ or $x + d$ may fall outside the feasible solution space, especially when a sample point is near to the boundaries of the solution space and ρ has a big value. In this case even if the new solution ($x - d$ or $x + d$) is a better solution, the movement is not made to that point and a "failure" is recorded. Recording a failure leads to reducing the parameter ρ so that the local search domain is reduced and the chance of feasible $x - d$ or $x + d$ is increased in the next iterations.

An important feature of this type of local search is that it does not rely on gradient information. So, it does not have the difficulties of calculating the gradient vector. However, after some iteration, the algorithm can estimate the appropriate direction toward the local optima and can continue the search in that direction (by using the bias vector). In other words, the Solis and Wets mechanism estimates the direction of movement based on the information obtained on the previous iterations, and then guide the search in that direction.

Another important feature of the Solis and Wets local search is its adaptive length of movement in each step. It means that this local search estimates if the current point is far from the local optima or near it. This estimation is based on the information of the previous iterations. If this method estimates that the current point is far from the local optima, the length of movement is increased. On the other hand, if this method estimates that the current point is near to the local optima, the length of movement is decreased and the search is focused on the nearer areas.

5 Computational results

We apply these three algorithms (i.e., Algorithms 1, 3, and 4) on a set of well-known test problems drawn from [5,6]. The global minimum value for all six functions, as shown in Table 1, is zero. We use each test problem with two different space dimensions ($n = 2$ and $n = 10$). Each of three above-mentioned algorithm is run for $n = 2$ in 5 s and for $n = 10$ in 30 s on a Pentium-IV 3 GHz PC. These algorithms are coded in VB and they are available upon request.

We run each algorithm 20 times for each function. Then, we record the average and minimum of the 20 obtained results (i.e., the average of results and the best result). It is worth noting that the number of local searches in each iteration of the original EM and our proposed hybrid EM is 50 times applied to all points.

In Table 1, it is clear that when the dimension of space is 2 (i.e., $n = 2$), the original EM and our proposed hybrid EM return the optimal results for all instances even in such a short time (5 sec.). However, the revised EM returns the optimal result in some problems and results that are near to the optimum for other problems.

As shown in Table 2, when the dimension of space increases ($n = 10$), our new proposed hybrid EM returns the best results, and the original EM returns better results than the revised EM.

Table 1 Computational results obtained for $n = 2$ in 5 s

Function name	Original EM		Revised EM		Proposed hybrid EM	
	Best	Average	Best	Average	Best	Average
Griewank	0.000	0.000	0.008	0.039	0.000	0.000
Levy	0.000	0.000	0.000	0.000	0.000	0.000
Rastrigin	0.000	0.000	0.002	0.038	0.000	0.000
Rosenbrock	0.000	0.000	0.000	0.003	0.000	0.000
Sum Squares	0.000	0.000	0.000	0.000	0.000	0.000
Zakharov	0.000	0.000	0.000	0.000	0.000	0.000

Table 2 Computational results obtained for $n = 10$ in 30 s

Function name	Original EM		Revised EM		Proposed hybrid EM	
	Best	Average	Best	Average	Best	Average
Griewank	0.039	0.079	0.561	0.764	0.028	0.075
Levy	0.000	0.000	0.008	0.021	0.000	0.000
Rastrigin	0.000	0.199	2.176	5.754	0.000	0.006
Rosenbrock	0.014	0.717	3.521	9.879	0.011	0.077
Sum Squares	0.000	0.000	0.028	0.068	0.000	0.000
Zakharov	0.000	0.000	0.000	0.000	0.000	0.000

An important conclusion obtained from Tables 1 and 2 is the importance of the local search procedure in the electromagnetism- like method (EM). Even a simple local search used in the original EM can improve the quality of results very well; while the revised EM returns the worse results due to the lack of any local search.

Despite the superior performance of our proposed hybrid EM, the simple local search of the original EM has an important advantage that cannot be ignored and that is its simple structure. Comparing the Solis and Wets local search used in our proposed hybrid EM, the simple local search of the original EM has only two parameters ($\boldsymbol{LSITER}$ and $\boldsymbol{\delta}$), which makes it easier in implementation and faster for the user in order to find the appropriate parameters for a given function.

6 Conclusion

We have proposed a novel hybrid approach combining an electromagnetism-like method (EM) with a strong local search method, known as Solis and Wets. To show the efficiency of our proposed hybrid EM, a number of experiments are carried out and the associated results are compared with the results taken from the literature. This comparison shows that the results of our hybrid EM outperform two original and revised EMs considered in this paper. We have also described another advantage of our proposed hybrid EM. This advantage is using the modifications of the revised EM. Thus, it can be proved that our hybrid EM exhibits global convergence with probability one, in which the proof is given in [3].

References

1. Birbil, S.I., Fang, S.-C.: An electromagnetism-like mechanism for global optimization. J. Glob. Optim. **25**(3), 263–282 (2002). doi:10.1023/A:1022452626305
2. Cowan, E.W.: Basic Electromagnetism. Academic Press, New York (1968)
3. Birbil, S., Fang, G., Sheu, R.-L.: On the convergence of a population-based global optimization algorithm. J. Glob. Optim. **30**, 301–318 (2005). doi:10.1007/s10898-004-8270-3
4. Solis, F.J., Wets, J.-B.: Minimization by random search techniques. Math. Oper. Res. **6**, 19–30 (1981)
5. Website of http://www-optima.amp.i.kyoto-u.ac.jp/member/student/hedar/Hedarfiles, global optimization test problems library
6. Törn, A., Ali, M.M., Viitanen, S.: Stochastic global optimization: problem classes and solution techniques. J. Glob. Optim. **14**, 437–447 (1999). doi:10.1023/A:1008395408187