

INT J COMPUT COMMUN, ISSN 1841-9836  
Vol.7 (2012), No. 4 (November), pp. 688-695

## An Electromagnetism-Like Approach for Solving the Low Autocorrelation Binary Sequence Problem

J. Kratica

**Jozef Kratica**

1. Mathematical Institute, Serbian Academy of Sciences and Arts  
Kneza Mihaila 36/III, 11 000 Belgrade, Serbia  
E-mail: [jkratica@mi.sanu.ac.rs](mailto:jkratica@mi.sanu.ac.rs)

**Abstract:** In this paper an electromagnetism-like approach (EM) for solving the low autocorrelation binary sequence problem (LABSP) is applied. This problem is a notoriously difficult computational problem and represents a major challenge to all search algorithms. Although EM has been applied to the topic of optimization in continuous space and a small number of studies on discrete problems, it has potential for solving this type of problems, since movement based on the attraction-repulsion mechanisms combined with the proposed scaling technique directs EM to promising search regions. Fast implementation of the local search procedure additionally improves the efficiency of the overall EM system.

**Keywords:** low autocorrelation binary sequence problem, electromagnetism-like metaheuristic, combinatorial optimization.

### 1 Introduction

Low autocorrelation binary sequence problem (LABSP) is a very hard combinatorial optimization problem with quite a simple formulation. The mathematical formulation of LABSP is based on a binary sequence  $s$  of length  $n$ . Let  $s \in \{-1, 1\}^n$ , i.e.  $s$  be represented by  $(s_1, s_2, \dots, s_n)$ , where  $s_i \in \{-1, 1\}$  for  $1 \leq i \leq n$ . Each sequence  $s$  is associated with the value of its energy function which is defined as follows:

$$E(s) = \sum_{j=1}^{n-1} C_j^2(s), \text{ where} \quad (1)$$

$$C_j(s) = \sum_{i=1}^{n-j} s_i s_{i+j}$$

Now, the low autocorrelation problem for binary sequences with length  $n$ , can be formulated as finding a sequence  $s$  of length  $n$  whose energy function is as minimal as possible. The second measure of quality of the sequence  $s$  is a merit factor

$$F(s) = \frac{n^2}{2E(s)}, \quad (2)$$

defined by Bernasconi in [2]. Mathematically, LABSP can be formulated as  $\max_{s \in \{-1, 1\}^n} F(s)$ . Both formulations are equivalent, and either of them can be used when it is convenient.

### 2 Previous work

LABSP has been deeply studied since the 1960s by both the communities of physics and artificial intelligence. There are two reasons behind this interest:

- It arises in many diverse areas including statistical mechanics and configuration state analysis [2], calibration of surface profile metrology tools [1], satellite and space applications [8], digital signal processing [16], etc.;

- LABSP is also a significant challenge of exact and/or heuristic applications, since it is known that the problem has "bit-flip" neighborhood structure of combinatorial landscapes [5, 6]. With this type of neighborhood, it is extremely steep around the optimum, which is sometimes referred to as "golf hole" landscapes and it poses a very difficult optimization problem. In that case, small changes in argument values usually cause drastic difference in objective value. For example, alteration of only one bit in binary sequence  $s$  can affect the objective value change by several tens of percents. From these reasons the LABSP is also listed as a problem 005 in the CSPLIB library.

Although Golay in [9] estimated that  $\lim_{n \rightarrow \infty} F(s) = 12.32$ , it is not well enough, because for dimensions between 21 and 60 the merit factor varies from  $F(s) = 5.627$  for  $n = 23$  up to  $F(s) = 9.85$  for  $n = 27$ , which is obviously far from the estimated limit 12.32.

The state-of-the-art exact method given in [12, 13] is based on exhaustive search and it solves problem optimally up to  $n = 60$ . The experimental research was carried out for several days on a multiprocessor cluster of 160 CPUs. Up to now it is the largest dimension with known optimal solution.

A hybrid evolutionary approach described in [4] combines the evolutionary search described in [14] and Kerningham-Lin heuristic defined in [11]. That evolutionary approach uses a specially defined termination criterion based on statistical analysis of known optimal solutions and their asymptotic behavior.

A detailed analysis of different stand-alone local search strategies is given in [7]. That analysis is later used in embedding the best local search strategy within other metaheuristic approaches. The results indicate that pure evolutionary algorithm cannot cope with the complexity of the problem and the assistance of local-search operators it is required to provide optimal or suboptimal results consistently. As a best choice for solving LABSP a memetic algorithm endowed with a tabu search local searcher is proposed, and that approach consistently finds optimal sequences in considerably less time than approaches previously reported in the literature.

Another metaheuristic method for solving LABSP, based on the stochastic local search (SLS), is presented in [10]. In-depth analysis of LABSP fitness landscape and the white-box visualization get insights on how SLS can be effective and lead to a slightly better strategy.

Local search algorithm described in [15], on the other hand, uses a quite different strategy compared to previous local search approaches, which is based on the randomized form of backtracking. In that way, the optimization problem is reduced to a series of constraint satisfaction problems which are to be solved iteratively, with decreasing upper bounds on the given objective function. Experimental results indicate that the algorithm is time consuming. For example, the average running time for  $n = 40$  is over 1000 seconds.

### 3 Proposed EM method

An electromagnetism-like (EM) metaheuristic is a powerful algorithm for global optimization that converges rapidly to optimum [3]. The method is also used for combinatorial optimization as a stand-alone approach or as an accompanying algorithm for other methods.

EM is a population based algorithm that can solve nonlinear optimization problems. In the following text each member  $p_k$ ,  $k = 1, 2, \dots, m$  of the population maintained by the algorithm will be referred to as EM point (or solution), and the population itself will be referred as a solution set.

The proposed EM algorithm for solving LABSP is given by the following pseudo code:

EM points in the first iteration are randomly initialized from  $[-1, 1]^n$  (function `Random_Init()`).

```

Algorithm EM
1 Random_Init()
  while iteration < max_iteration do
    foreach EM point  $p_k$  in the solution_set do
      2 Calculate_objective_value( $p_k$ )
      3 Local_search( $p_k$ )
      4 Scale_solution( $p_k$ )
      5 Calculate_charge_and_forces()
      6 Move()

```

**Algorithm 1:** EM pseudo code

For a given EM point  $p_k$ , sequence  $s$  is obtained by rounding, i.e.  $s_i = \begin{cases} 1, & p_{ki} \geq 0 \\ -1, & p_{ki} \leq 0 \end{cases}$ , for each coordinate  $i = 1, \dots, n$ . Energy  $E(s)$  and merit factor  $F(s)$  are computed by using (1) and (2).

### 3.1 Local search and scaling

This step is used to move the sample points towards the local optima that are near them. Points are pushed towards the local valleys using a neighborhood search procedure. The local search method used in this algorithm is simple but effective. Regarding the importance of the local search step, it is described in Algorithm 2.

The proposed local search procedure uses the first improvement strategy, which means that when an improvement is detected, the improvement is immediately applied and local search continues. If for each member of sequence  $s$  swap produces energy value greater or equal than the original one, the local search ends with no further improvement.

In this implementation, scaling procedure is also applied, which additionally moves points towards solutions obtained by local search. It is considered only with some factor  $\lambda \in (0, 1)$  to prevent falling into a local optimum and become trapped there. An EM point  $p_k$  is moved by the following formula

$$p_k \leftarrow \lambda \cdot p_{k'} + (1 - \lambda) \cdot p_k \quad (3)$$

where  $p_{k'}$  denotes sequence  $s$  of the  $k$ -th EM point in the current iteration when the local search procedure finished its work.

Choosing appropriate value of the scale factor  $\lambda$  is significant for governing the search process. In the extremal case, when  $\lambda$  is close to 1, the search process will likely fall into a local optimum and become trapped. Another extremal case, when  $\lambda$  is equal to 0, obviously represents no-scaling situation. Experiments showed that  $\lambda = 0.1$  is a good compromise which yields satisfactory results.

### 3.2 Attraction-repulsion mechanism

As can be seen from the literature, the strength of the EM algorithm lies in the idea of directing the sample points towards local optima utilizing an attraction-repulsion mechanism. Therefore, after applying the local search procedure to each solution in the current population, the solutions must be moved towards promising regions in order to get closer to the optimal solution.

```

Function Local_search( $p_k$ )
1 repeat
2    $impr \leftarrow \text{false}$ 
3    $i \leftarrow 0$ 
4   while  $i < n$  and not  $impr$  do
5      $i \leftarrow i + 1$ 
6     for  $j = 1$  to  $n$  do
7        $C_j^{tmp} = \begin{cases} C_j - 2 \cdot s_j \cdot s_{i+j}, & j < n - i \\ C_j^{tmp} = C_j, & j \geq n - i \end{cases}$ 
8        $C_j^{new} = \begin{cases} C_j^{tmp} - 2 \cdot s_j \cdot s_{j-i}, & j \geq i \\ C_j^{tmp}, & j < i \end{cases}$ 
9        $E^{new}(s) = \sum_{j=1}^{n-1} (C_j^{new}(s))^2$ 
10      if  $E^{new}(s) < E(s)$  then
11         $impr \leftarrow \text{true}$ 
12         $s_i \leftarrow -s_i$ 
13        for  $j = 1$  to  $n$  do
14           $C_j \leftarrow C_j^{new}$ 
15           $E(s) \leftarrow E^{new}(s)$ 
16 until not  $impr$ 
17  $F(s) \leftarrow \frac{n^2}{2E(s)}$ 

```

Algorithm 2: Local search pseudo code

In this process, each sample point is considered as a charged particle. The charge of each sample point is calculated by the following formula:

$$q_i = \exp \left( -n \frac{f(p_i) - f(p^{best})}{\sum_{k=1}^m f(p_k) - f(p^{best})} \right). \quad (4)$$

The amount of charge relates to the value of the objective function ( $f(p_k) = E(s)$ ) at the point, which also determines the magnitude of attraction or repulsion of the point over the sample population.

According to the superposition principle of electromagnetism theory, the force exerted on a point via another point is inversely proportional to the distance between the points and directly proportional to the product of their charges. Mathematically, the power of attraction or repulsion of charges is calculated as follows:

$$F_i = \sum_{j=1, j \neq i}^m F_i^j, \text{ where} \quad (5)$$

$$F_i^j = \begin{cases} \left( \frac{q_i q_j}{\|p_j - p_i\|^2} \right) \cdot (p_j - p_i), & f(p_j) < f(p_i) \\ \left( \frac{q_i q_j}{\|p_j - p_i\|^2} \right) \cdot (p_i - p_j), & f(p_j) \geq f(p_i) \end{cases}$$

where  $\|p_i - p_j\|$  is the Euclidean distance between EM points  $p_i$  and  $p_j$ .

As mentioned before, by using the Move() procedure of the electromagnetism approach, current solutions are shifted towards the best ones. All the EM points are moved with the exception of the current best solution. Detailed explanations about movement are given in Algorithm 3.

```

Function Move()
foreach EM point  $p_k$  in the solution_set do
  if  $p_k \neq p_{best}$  then
1    $\beta \leftarrow \text{Random}[0, 1]$ 
2    $F_i \leftarrow F_i / \|F_i\|$ 
3   for  $i = 1$  to  $n$  do
4     if  $F_{ki} > 0$  then
       $p_{ki} \leftarrow p_{ki} + \beta \cdot F_{ki} \cdot (1 - p_{ki})$ 
5     else
       $p_{ki} \leftarrow p_{ki} + \beta \cdot F_{ki} \cdot (p_{ki} + 1)$ 

```

**Algorithm 3:** Move pseudo code

As can be seen from Algorithm 3, the movement of each EM point is in the direction of total force exerted on it by a random step length  $\beta$ . This length is generated from uniform distribution between  $[0,1]$ . As can be seen in [3], the candidate solutions have a nonzero probability to move to the unvisited solution along this direction when random step length is selected. Moreover, normalizing the total force exerted on each candidate solution implies that infeasible solutions cannot be produced.

## 4 Experimental results

In this section, the proposed EM solution procedure on LABSP is tested for  $n$  up to 40 nodes, for which the optimal solutions are known in the literature.

Each numerical experiment was repeated 20 times and the results are summarized in Table 1, which is organized as follows:

- The first three columns contain  $n$ , optimal solution value (merit factor  $F(s)$ ) and the EM best solution obtained in 20 runs;
- The average running time ( $t$ ) and number of iterations  $iter$  used to reach the final EM solution for the first time are given in the fourth and fifth columns, while the total running time  $t_{tot}$  necessary to finish EM is given in the sixth column.

- The last two columns ( $agap$  and  $\sigma$ ) contain information on the average solution quality:  $agap$  is a percentage gap defined as  $agap = \frac{1}{20} \sum_{i=1}^{20} gap_i$ , where  $gap_i = 100 * \frac{EM_{best} - EM_i}{EM_{best}}$  and  $EM_i$  represents the EM solution (merit factor  $F(s)$ ) obtained in the  $i$ -th run, while  $\sigma$  is the standard deviation of  $gap_i$ ,  $i = 1, 2, \dots, 20$ , obtained by formula  $\sigma = \sqrt{\frac{1}{20} \sum_{i=1}^{20} (gap_i - agap)^2}$ .

The computational results were performed on an Intel 2.5 GHz single processor with 1GB memory, under Windows operating system. All EM runs were made with the following empirically determined parameters:  $m = 10$ ,  $iter_{max} = 100000$  and  $\lambda = 0.1$ . These values cause most charges to exhibit convergent behavior with a few individuals diverging, thereby providing a good balance between local and global search. In this case all these values were chosen experimentally as a matter of convenience because they provide good results.

Table 1: Computational results

$n$	$Opt_{sol}$	$EM_{best}$	$t$ (sec)	$t_{tot}$ (sec)	$agap$ (%)	$\sigma$ (%)
3	4.500000	opt.	0.0010	1.5503	0.000	0.000
4	4.000000	opt.	0.0010	4.3613	0.000	0.000
5	6.250000	opt.	0.0010	4.8269	0.000	0.000
6	2.571429	opt.	0.0010	11.6996	0.000	0.000
7	8.166667	opt.	0.0010	9.8925	0.000	0.000
8	4.000000	opt.	0.0010	16.4588	0.000	0.000
9	3.375000	opt.	0.0010	17.3339	0.000	0.000
10	3.846154	opt.	0.0010	20.3198	0.000	0.000
11	12.100000	opt.	0.0017	16.8205	18.462	34.585
12	7.200000	opt.	0.0010	17.4573	0.000	0.000
13	14.083333	opt.	0.0031	17.3925	11.429	25.806
14	5.157895	opt.	0.0010	26.4173	0.000	0.000
15	7.500000	opt.	0.0900	22.1761	1.739	7.715
16	5.333333	opt.	0.0010	23.1151	0.000	0.000
17	4.515625	opt.	0.0031	25.3597	0.000	0.000
18	6.480000	opt.	0.0052	28.1081	2.424	7.453
19	6.224138	opt.	0.0249	23.8753	1.212	3.681
20	7.692308	opt.	0.3983	26.3590	8.235	12.230
21	8.480769	opt.	0.0865	26.3949	19.226	12.095
22	6.205128	opt.	0.0266	33.8676	3.404	7.048
23	5.627660	opt.	0.1937	30.9995	2.745	3.847
24	8.000000	opt.	0.2528	32.9262	19.003	13.390
25	8.680556	opt.	0.8880	32.2269	13.759	12.523
26	7.511111	opt.	1.6412	36.4104	4.887	9.350
27	9.851351	opt.	0.9698	35.7136	29.084	18.230
28	7.840000	opt.	1.1606	37.8760	17.338	11.389
29	6.782258	opt.	3.7152	36.8581	6.531	6.318
30	7.627119	opt.	1.7120	42.7611	13.471	10.877
31	7.171642	opt.	2.6340	43.7440	9.258	6.876
32	8.000000	opt.	2.7331	46.6714	15.540	11.667
33	8.507813	opt.	4.4168	49.8097	15.667	9.911
34	8.892308	opt.	6.9574	51.4971	25.079	9.309
35	8.390411	opt.	1.8435	52.5394	19.264	8.260
36	7.902439	opt.	2.4458	55.0151	17.239	8.781
37	7.959302	opt.	6.4090	55.7682	15.507	7.856
38	8.298851	opt.	2.6012	61.6151	20.118	9.361
39	7.681818	opt.	7.8379	62.7660	13.623	8.275
40	7.407407	opt.	9.4135	73.1222	14.369	6.279

Observing the data shown in Table 1, it is remarkable that EM identifies optimal solutions in all cases. Moreover, the EM performs very efficiently, since the total running time was less than 74 seconds with one million objective function evaluations. Note that, most of this time is spent after the EM reach optimal solution merely to satisfy the finishing criterion. Also mind that in the case when  $n = 40$ , search space is  $2^{40}$  and EM searched only  $1.17 \cdot 10^{-7}$  part of it to reach the optimal solution.

## 5 Conclusions and Future Works

In this article, a hybrid approach combining an electromagnetism-like method (EM) with a scaling technique for solving the LABSP is proposed. The fast local search procedure and the applied scaling scheme were adapted to facilitate the use of EM to boost the performance of the proposed algorithm. To show the efficiency of the proposed hybrid EM, a number of experiments was carried out and the results were compared with the optimal solutions taken from the literature. The obtained results clearly indicate that EM is a useful tool for solving this problem.

As a direction for future studies, it can be interesting to parallelize the EM and run it on a powerful multiprocessor computer. Another orientation of future research can be incorporation of this method in some exact solution framework.

## Acknowledgments

This research was partially supported by Serbian Ministry of Education and Science under grants 174010 and 174033.

## Bibliography

- [1] S.K. Barber, P. Soldate, E.H. Anderson, R. Cambie, W.R. McKinney, P.Z. Takacs, D.L. Voronov, V.V. Yashchuk, Development of Pseudorandom Binary Arrays for Calibration of Surface Profile Metrology Tools, *Journal of Vacuum Science and Technology B: Microelectronics and Nanometer Structures*, Vol.27, No.6, pp.3213–3219, 2009.
- [2] J. Bernasconi, Low Autocorrelation Binary Sequences: Statistical Mechanics and Conguration State Analysis, *Journal Physique*, Vol.48, pp.559–567, 1987.
- [3] S.I. Birbil, S.C. Fang, An Electromagnetism-like Mechanism for Global Optimization, *Journal of Global Optimization*, Vol.25, pp.263–282, 2003.
- [4] F. Brglez, X.Y. Li, M.F. Stallman, B. Miltzer, Reliable Cost Prediction for Finding Optimal Solutions to LABS Problem: Evolutionary and Alternative Algorithms, *Fifth International Workshop on Frontiers in Evolutionary Algorithms*, Cary, NC, USA 2003.
- [5] A.V. Eremeev, C.R. Reeves, Non-parametric Estimation of Properties of Combinatorial Landscapes, *Lecture notes on Computer Science*, Vol.2279, pp.31–40, 2002.
- [6] F. Ferreira, J. Fontanari, P. Stadler, Landscape Sstatistics of the Low Autocorrelated Binary String Problem, *Journal of Physics A: Mathematical and General*, Vol.33, pp.8635–8647, 2000.
- [7] J.E. Gallardo, C. Cotta, A.J. Fernandez, Finding Low Autocorrelation Binary Sequences with Memetic Algorithms, *Applied Soft Computing*, Vol.9, No.4, pp.1252–1262, 2009.

- [8] R. Garelo, N. Boujnah, Y. Jia, Design of Binary Sequences and Matrices for Space Applications, *Proceedings of the 2009 International Workshop on Satellite and Space Communications - IWSSC'09*, art. no. 5286416, pp.88–91, 2009.
- [9] M.J.E. Golay, The Merit Factor of Long Low Autocorrelation Binary Sequences, *IEEE Transactions on Information Theory*, Vol.28, pp.543–549, 1982.
- [10] S. Halim, R.H.C. Yap, F. Halim, Engineering Stochastic Local Search for the Low Autocorrelation Binary Sequence Problem, *Lecture Notes in Computer Science*, Vol.5202, pp.640–645, 2008.
- [11] B.W. Kernighan, S. Lin, An Efficient Heuristic Procedure for Partitioning Graphs, *Bell System Technical Journal*, pp.291–307, 1970.
- [12] S. Mertens, Exhaustive Search for Low-autocorrelation Binary Sequences, *Journal of Physics A: Mathematical and General*, Vol.29, pp.473–481, 1996.
- [13] S. Mertens, H. Bauke, Ground States of the Bernasconi Model with Open Boundary Conditions, Available online at <http://odysseus.nat.uni-magdeburg.de/~mertens/bernasconi/open.dat>
- [14] B. Militzer, M. Zamparelli, D. Beule, Evolutionary Search for Low Autocorrelated Binary Sequences, *IEEE Transactions on Evolutionary Computation*, Vol.2, pp.34–39, 1998.
- [15] S. Prestwich, Exploiting Relaxation in Local Search for LABS, *Annals of Operations Research*, Vol.156, pp.129-141, 2007.
- [16] A. Ukil, Low Autocorrelation Binary Sequences: Number Theory-Based Analysis for Minimum Energy Level, Barker codes, *Digital Signal Processing: A Review Journal*, Vol.20, No.2, pp.483–495, 2010.