An EVEBO-Based BTS Localization Algorithm

Koorosh Navi, Manoochehr Kelarestaghi and Farshad Eshghi
Electrical & Computer Eng. Dept., Kharazmi University, 15719-14911 Tehran, Iran

Keywords: BTS Localization, EVEBO, Election Inspired, Evolutionary Algorithm, Meta-Heuristics, NP-Hard.

Abstract: In this paper, we use EVEBO, an election-inspired optimization algorithm, to solve the BTS (i.e. transceiver) localization problem. The proposed method tries to solve the classic and very important problem of achieving maximum coverage with minimum number of BTSs in a specified geographical area. It also tries to reduce the over-coverage rate, one of the undesirable phenomena in cellular networks. The EVEBO's merit in solving the problem is measured by a common fitness function, and speed of convergence. Simulation results show that EVEBO solves the problem in much less number of evaluations compared to the best results reported in the literature for square-coverage transceivers. We also show that it can be used in a scenario involving more challenging non-square-coverage (almost circular) transceiver type with satisfactory results.

1 INTRODUCTION

Optimization problems are one of the most common problems in science and engineering. Achieving the exact solution, in too many cases, is almost impossible due to the high complexity and multiplicity of dimensions of the problem in hand. So, solving the problem comes down to searching for the best possible solution.

NP-hard problems, Mathematical Programming, Regression Analysis, and etc are different types of optimization problems. (Luke, 2009)

NP-hard problems are the ones we try to solve by searching for and finding the optimum solution(s) and not necessarily the exact one, because of the inherent complexity involved. Adding more dimensions to a problem, makes it harder to solve (Luke, 2009). Therefore, researchers try to devise new algorithms and techniques regularly, in hope of solving them nicely.

Some of the most popular methods to solve NP-hard problems, particularly those involve multiple solutions, are Evolutionary Algorithms (EA), which try to evolve a population of individuals (solutions) by iterative execution of a set of simple operations. They do this based on a fitness function that shows how much an individual in population is adequate. The operation will continue until a termination condition is satisfied (Eiben, 2003).

EAs are a subset of a more general category called Meta-Heuristics, which try to find better and better solutions by using simple search mechanisms in the problem's defined search space, based on the evolution of a population of solutions (Bianchi et al., 2009). Genetic Algorithm (GA), Evolution Strategy (ES), Evolution Programming (EP), and Genetic Programming (GP) technique (Beyer, 2013) and other population-based methods such as: Differential Evolution (DE) (Simon, 2013), and Swarm intelligence algorithms like Particle Swarm optimization (PSO), and Ant Colony are the most common evolutionary algorithms (Yu and Gen, 2010).

Ever-increasing wireless devices usage has made the optimization problem of estimating the number of base transceiver stations (BTSs) in an area very relevant (called BTS localization). In this kind of problems, the goal is to find the location of the transceivers in a specified area that ensures maximum coverage with minimum number of units. Due to its NP-hard complexity nature (Vega-Rodríguez et al., 2007), most of the BTS localization solutions in the literature are meta-heuristic-based algorithms. In this work, we employ EVolutive-Election Based Optimization (EVEBO), a most recently devised evolutionary algorithm, to solve the similar problem due to its capability in solving large-size(dimension) problems and its low dependency on initial problem parameters values.

We will show that our new proposed method which uses EVEBO algorithm on binary representation space can find the optimum solution in a acceptable evaluation numbers (computational efforts) and also shows better result in some competitions. Be-

Navi K., Kelarestaghi M. and Eshghi F.
An EVEBO-Based BTS Localization Algorithm.
DOI: 10.5220/0006500902650272
In *Proceedings of the 9th International Joint Conference on Computational Intelligence (IJCCI 2017)*, pages 265-272
ISBN: 978-989-758-274-5

sides that we will try to minimize over-coverage rate (coverage overlap), one of the unpleasant parameters in this issue.

It should be mentioned that we consider this problem as a permutation NP-hard (which based on a binary representation) issue and thus we try to find the optimum solution regardless of considering any special telecommunication parameters.

The rest of this paper is organized as follows. In the next section, the motivation of authors regarding solving this classic problem will be expressed. Thereafter, some relevant works are briefly explained. A short description of EVEBO algorithm come in section 4. Application of the EVEBO along with some adaptations towards the BTS localization problem will be introduced in section 5. Simulation results and comparisons against other meta-heuristic methods appear in section 6. Concluding remarks are mentioned in the last section.

2 MOTIVATION

Increasing mobile usage in the daily life necessitates more wireless coverage. Accordingly, one of the most challenging optimization problems in recent years, and years to come, is to estimate the minimum required number of BTSs and their locations for a maximum coverage in a specific geographical area.

Regarding maximizing the coverage area, specifying the locations of a set of BTSs is an NP-hard optimization problem (Vega-Rodríguez et al., 2007). Moreover, looking at the problem as to select the locations of BTSs from all possible locations, makes this problem to be seen as a permutation optimization problem.

We believe that EVEBO, a recently introduced election-inspired evolutionary algorithm, is a good candidate for solving this kind of problems due to its distributed nature and low sensitivity to initial values of problem parameters.

3 RELATED WORKS

Solving BTS localization problem will lead to determination of a set of optimal BTS locations in a geographical area ensuring a satisfactory coverage (not necessarily a total coverage) with minimum number of transceivers.

Find a set of optimized BTS locations in a radio network design using a distributed steady state (GA) in (Alba, 2004) is one of the first EA-based efforts for solving the problem.

Another radio network design by some meta-heuristics approaches such as GA, DE, PBIL (Population-Based Incremental Learning) which is a combination of GA and competitive learning, CHC (it is similar to EAs but with no mutation), and Simulated Annealing (SA) were introduced in (Vega-Rodríguez et al., 2007). These methods try to achieve the maximum coverage with minimum number of square-coverage BTSs, without considering the over-coverage in their fitness functions. Simulation results show that PBIL, SA and CHC have obtained the optimal fitness function value (with full coverage), while DE gives better convergence speed performance (less evaluations). (Calégari et al., 1997) tried to solve the problem by proposing a parallel island-based GA wherein the population of solutions are distributed over some workstations called islands.

There are other related works that try to find adequate solutions by considering specific telecommunication factors such as signal strength, power level at specified locations, and etc. Amongst them, a global optimization for multiple transmitter locations is proposed by (Nelson et al., 2006) which uses particle swarm optimization to find the transmitter locations. It tries to minimize the difference between the true received power, measured by a set of sensor nodes in the specified area, and the estimated power with considering initial smart clustering conditions. An Expectation-Maximization (EM) technique for locating multiple transmitters, based on power levels observed by a set of arbitrarily-placed receivers, is proposed by (Nelson and Gupta, 2007). Finally, (Nelson et al., 2009) proposes a quasi EM method for estimating multiple transmitter locations based on received signal strength measurements by a sensor network of randomly located receivers in the area. Simulation results show its outperformance compared to methods presented by (Nelson et al., 2006) and (Nelson and Gupta, 2007).

4 EVEBO: EVOLUTIVE ELECTION-BASED OPTIMIZATION

One of the most recent meta-heuristic evolutionary algorithms is EVEBO which was inspired by mankind electoral systems and was modified for becoming capable of solving some optimization problems (Pourghanbar et al., 2015). The idea of EVEBO is based on a common society feature, called collective behaviour, which uses the aggregation of human opinions to determine a winner person(s) over a number of nominated candidates, through an official election. Similar to other EAs, EVEBO starts with a population of initial individuals (non-optimized solutions) in a problem-specific defined search space called Belief Space (BS). The individuals are sorted using a fitness function values. Some better solutions are marked as candidates and each of the remaining individuals moves towards each candidate inversely related to their inter-distance in the BS (campaign). Then, each individual votes for its closest candidate(s) in the BS. The candidate(s) (solution(s)) with the highest number of votes is(are) elected as the winner(s). In the next step, the post-election, each winner moves towards his fans (individuals who voted for him) and other non-fan individuals. The whole process comprising campaign, voting, and post-election steps is considered one epoch of running EVEBO. The above cycle continues until a termination condition is satisfied. A brief description of how EVEBO transforms a real life election phenomenon into a tool for solving an optimization problem is illustrated in Figure 1.

Election / Entities	Real life election	EVEBO
Individuals	Eligible persons	Population of solutions
Legislations	Specified election rules which are validated and controlled by a committee	Modification rules applied on solutions to satisfy problem conditions
Candidates	Election candidates	Better solutions in the population based on their fitness value
Campaign	Campaigning by: 1- Fans 2- Candidates	1- Each solution moves towards the nearest non-candidate based on their fitness function values. 2- Each solution moves towards each candidate based on their distance
Voting	People vote for their favorite candidate(s)	Each solution is associated with its nearest candidate(s) based on a distance.
Winner(s)	Winner candidate(s) is(are) determined by counting votes.	Candidate solution(s) with highest number of associates is(are) the winner(s).
Post-Election	Winner(s) should adapt themselves to fans' (and also non-fans') expectations.	The winner solution(s) is(are) manipulated by other solutions. Then all other solutions will be updated, just once, according to the updated winner(s).
Tolerance	If elected winner(s) cannot satisfy people expectations in number of consecutive elections, people try to change the electoral system (Revolution).	If winner(s) solution'(s) fitness value(s) would not get better in number of consecutive evaluations, the electoral system will be changed.

Figure 1: Election entities correspondence between real life election and EVEBO.

EVEBO evaluation steps are described as follows.

- **INITIALIZATION.** A population of initial solutions will be generated randomly with respect to the predefined conditions for problem in hand.
- **UTILITY MATRIX.** All evolutionary algorithms involve some stochastic(Simon, 2013). In EVEBO, each individual in the population has a probabilistic parameter called Utility Matrix (UM). The number of rows in UM is equal to the number of solutions in the population and the number of its columns is equal the dimension of solutions. UM elements are randomly generated $\in [0,1]$ at the beginning of each evaluation. Each element of the UM determines the probability with which modifications in the corresponding element of the solution takes place (for more details see (Pourghanbar et al., 2015)).
- **ELECTORAL SYSTEM SELECTION.** In this step, an electoral system will be randomly selected from those systems which are suited, in terms of modality (uni-modal (one optima) or multi-modal (more than one optima)), to the specific problem in hand. For uni-modal problems which is our interest herein, we randomly select one from FPTP (First Past the Post), TRS (Two-Round System), and IRS (Instant-Runoff System) electoral systems described in Figure 2.

FPTP (First Past The Post): The candidate solution with highest number of votes is the evaluation winner	TRS (Two-Round System): The candidates with absolute majority of votes is the evaluation winner	IRS (Instant-Runoff System): Individuals vote for an ordered list. Candidates with lowest number of first places in the list will be removed until one candidate is left.

Figure 2: Electoral systems suited to problems seeking one optimal solution (uni-modal).

- **ELECTORAL LEGISLATIONS.** Based on the nature of the problem, some pre-defined conditions should be verified and enforced on solutions. For instance, those individuals which are not valid based on our problem conditions are removed and replaced by another random ones (to ensure that the population size remains fixed).
- **SETTING CANDIDATES AND CAMPAIGNING.** After selecting the electoral system, the number of candidates will be determined according to the population size. The candidates will be selected based on their fitness values. Then, as a result of campaigning, each non-candidate individual will be affected by its nearest (in distance) individual, and its corresponding solution is updated according to their fitness values. Finally, for each candidate, a Gaussian Impact Function (GIF) of distance is calculated. All non-candidate individuals are updated based on these GIF values.

- **VOTING.** Each individual votes for the nearest candidate(s). The candidate(s) with the highest number of votes is(are) selected as the winner solution(s) in this evaluation.
- **POST-ELECTION.** Because the winner solution(s) is(are) not necessarily the optimum one(s), so, in this step, the winner solution(s) is(are) updated according to its(their) distance(s) from all other individuals.

As mentioned before, solution update in this algorithm is stochastic and based on a utility matrix. Also, EVEBO employs a tolerance parameter which controls the solution progress in evaluations (Tolerance in Figure 1).

The flowchart of Figure 3 illustrates the implementation of EVEBO.

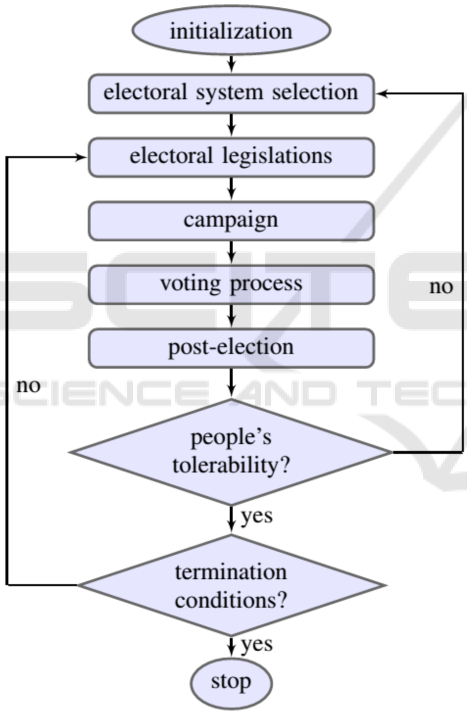

Figure 3: EVEBO's functional flowchart (Pourghanbar et al., 2015).

5 BTS LOCALIZATION USING EVEBO

In this work, we use EVEBO to solve BTS-localization problem with some modification in the original algorithm. First we try to estimate the square-coverage BTS locations in a discretized area aiming at maximum coverage with minimum number of transceivers. To be more realistic, thereafter, we solve the problem for circular-coverage BTSs as well. When a circular or any non-square-coverage transceiver is employed, it is impossible to get full coverage without having any over-coverage (Calégari et al., 1997) (Figure 4).

In what follows, first, we express the essentials of the problem and then proceed with applying EVEBO steps to it.

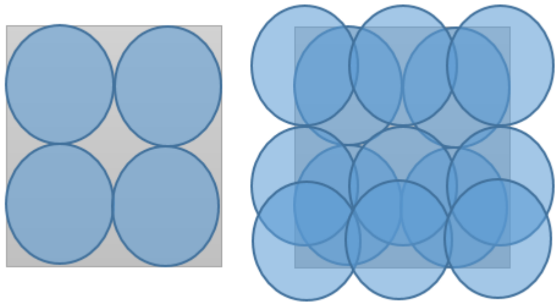

Figure 4: Coverage and over-coverage in non-square transceiver cellular networks. Left: Partial coverage and a no over-coverage case. Right: Full coverage and partial over-coverage case.

- **SOLUTION REPRESENTATION:** We consider each individual solution $\mathbf{X}$ as a binary vector of n elements (the total number of different potential BTS locations) where each element x_i can be 1 or 0 which shows whether a transceiver has been positioned in that location or not:

$$X = \left[\begin{array}{lll} x_1 & x_2 & \ldots x_n \end{array}\right] \quad (1)$$

In fact, the one dimensional vector $\mathbf{X}$ is the linear representation of the 2-D grid of BTS locations. This will reduce the complexity of the problem significantly by replacing euclidean distance computations with Hamming distance. Each vector's element is a pointer to the two dimensional BTS location in the grid space. A genotype/phenotype solution representation of the problem is illustrated in Figure 5.

Besides, the binary/1-D representation of the problem has the benefit of fairer comparison with the related works using similar representations.

- **FITNESS FUNCTION DEFINITION.** Our problem has 3 objective parameters: Coverage, over-coverage and number of BTSs. These parameters can be unified into one objective criterion $f(.)$, called fitness function, as proposed by (Calégari et al., 1997):

$$f(\mathbf{X}) = \frac{(CR(\mathbf{X}) - k.OCR(\mathbf{X}))^2}{N_{BTS}} \quad (2)$$

in which $\mathbf{X}$ is an individual solution in our population described before, and CR, OCR, and N_{BTS} are

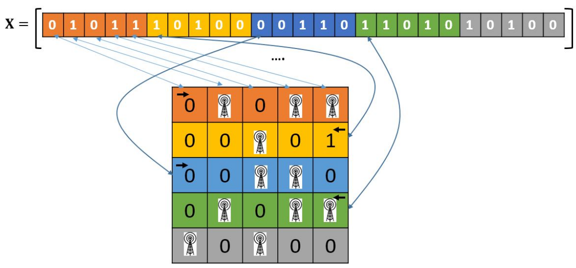

Figure 5: An example of corresponding binary vector **X** (genotype solution representation) to the 2-D placement of transceivers in a discretized grid area (phenotype solution representation).

coverage rate, over-coverage rate, and the number of BTSs respectively.

Our goal is to maximize $f(\mathbf{X})$. There is another non-objective weighing parameter in the fitness function, k, where $k \in [0,1]$ states the importance of the over-coverage rate in the fitness function. It is obvious that setting $k = 0$ will ignore the over-coverage factor in the problem.

- **UNCERTAINTY PARAMETER.** In this work, we replace the utility matrix with a stochastic vector $\mathbf{P}(\mathbf{X})$, $p_i \in [0,1]$, for each $\mathbf{X}$ in the population. For large problems like ours, creating a matrix for each solution would increase memory usage and computational complexity. This was confirmed by our simulation results which are not presented herein for the sake of brevity.

The application of modified-EVEBO to the BTSs localization problem proceeds through the following steps:

1. **Initial Individuals.** A population of initial solutions is generated randomly. The size of the population (number of solutions) is a constant value (See Table 2).
2. **Electoral System Selection.** In a flat grid area with no limitation in BTS locations(i.e. uniform nature of the problem), the uni-modal consideration is a right choice since there are not much difference between solutions out of a multi-modal consideration of the problem. Amongst FPTP, TRS and IRS electoral systems, one is randomly selected. If the best solution does not improve (based on the fitness value) during a predetermined number of evaluations, represented by a *Tolerance* threshold, the electoral system is switched to another one (corresponding to revolution in EVEBO).
3. **Legislations.** For the problem in hand, all individuals in the population will be verified for duplicate solutions.
4. **Fitness Calculation and Candidates Setting.** In this step, the fitness values of all solutions in the population will be calculated. Thereafter, a number of highest-in-fitness solutions will be selected as the candidates in the current evaluation. Then, $\mathbf{P}(\mathbf{X})$, the uncertainty parameter, is generated for each solution $\mathbf{X}$ in the population.
5. **Campaigning.** Each solution $\mathbf{X}$ takes a move, with the probabilistic influence of $\mathbf{P}(\mathbf{X})$, towards its nearest (in terms of hamming distance) neighbor/each candidate by randomly selecting $m(\mathbf{X})$ elements of $\mathbf{X}$ and setting them equal to the corresponding elements in that neighbor/candidate (Figure 6). The value of $m(\mathbf{X})$ is calculated as below:

$$m(\mathbf{X}) = \lfloor \frac{fit(NN/C)}{fit(NN/C)) + fit(\mathbf{X})} * size(\mathbf{X}) \rfloor \quad (3)$$

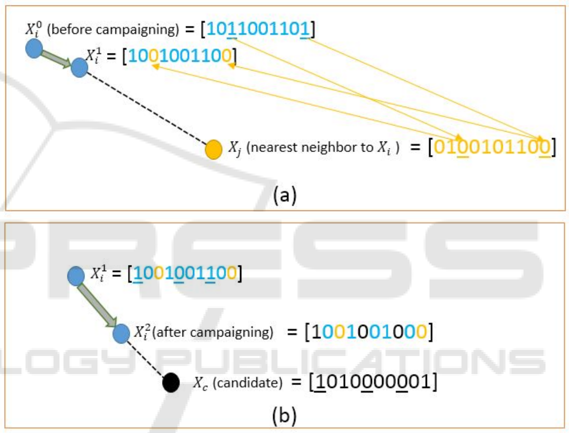

Figure 6: An example of campaigning effects on an individual solution $\mathbf{X}_i$ in a binary belief space: a) $\mathbf{X}_i^0$ moves under effect of and towards its nearest neighbor (based on Hamming distance) to the new position $\mathbf{X}_i$ ($m(\mathbf{X}_i)$ =2). b) $\mathbf{X}_i^1$ moves towards a candidate to the new position $\mathbf{X}_i^2$ ($m(\mathbf{X_i})$ =3).

where NN and C represent the nearest neighbor and each candidate respectively. It is obvious that the higher is $m(\mathbf{X})$, the closer $\mathbf{X}$ moves towards its nearest neighbor or a candidate. It should be noted that, herein, the above campaigning method has replaced the continuous Gaussian Impact Function (GIF) in the original EVEBO. Since GIF is a continues function, so it works just when proximity criteria is a continues distance in an euclidean space. However, as was mentioned earlier, we use binary vectors with Hamming-distance proximities in this problem for ease of implementation and fairer comparison to the related works.

6. **Voting:** Each individual $\mathbf{X}$ identifies and votes for its nearest candidate. The candidate with the high-

est number of votes will be announced as the winner solution in the current evaluation.

7. **Post Election:** Finally, the winner solution is updated by considering the impacts of all other individuals. Particularly, a light mutation-like correction will be probably (according to $\mathbf{P}(\mathbf{X})$) performed on the winner. As a result, some new BTSs will be randomly placed and some already existing ones will be randomly removed. It should be noted that each occurred mutation is also counted as one more evaluation. The offsprings will remain in the next evaluation only if they are better than their parents (Elitism in EAs (Yu and Gen, 2010)).

The pseudo-code of the above steps appear in Algorithm 1.

Algorithm 1: EVEBO-Based BTS localization.

Parameters initialization.
Generating initial solutions and first fitness calculation.
$Evaluation \leftarrow 0$
while Termination conditions not reached **do**
    System Selection
    $Tolerance \leftarrow t_0$
    while $Tolerance > 0$ **do**
        Legislations
        for each solution $\mathbf{X}_i$ in population **do**
            $p(\mathbf{X}) \leftarrow rand \in [0,1]$
        Set candidates based on fitness function.
        (*) Campaigning - Update solutions
        Voting - Set Winner
        (*)Post election: Update winner and other individual
        if Mutation **then**
            $Evaluation \leftarrow Evaluation + 1$
        Fitness values calculation
        if $fit(winner) > fit(Best)$ **then**
            $Best \leftarrow winner$
            Elitism: Always save best
        if $fit(winner) <= fit(Best)$ **then**
            $Tolerance \leftarrow Tolerance - 1$
        $Evaluation \leftarrow Evaluation + 1$

(*) All changes on the solution $\mathbf{X}_i$ will be enforced with probability $P(\mathbf{X}_i)$.

6 SIMULATION RESULTS

To verify the performance of EVEBO-based BTS localization algorithm, it is compared to the methods introduced in (Vega-Rodríguez et al., 2007). Herein, we intend to estimate the optimum number and locations of square-coverage BTSs in a discretized square flat area. For a fair comparison, we use the classic $287 * 287$-grid-point area and the maximum possible of 349 BTS locations as in (Vega-Rodríguez et al., 2007). Finally, we will consider a more realistic and challenging (in the sense of coverage and over-coverage) circular-coverage BTS case. As mentioned earlier, there are no report of any previous meta-heuristic-based works on BTS-localization problem with circular-coverage BTSs(the only reports of circular-coverage BTSs involve communication signal parameters). All simulations were done in Java with external JFree chart library.

6.1 Numerical Results

Table 1. summarizes the simulation results of the proposed EVEBO method and the two best methods in (Vega-Rodríguez et al., 2007). We chose two best methods (one for its better fitness value and another for less number of evaluations) in (Vega-Rodríguez et al., 2007). For a fair comparison and statistical purposes, 30 independent runs have been performed. To be consistent with the results in (Vega-Rodríguez et al., 2007), the over-coverage was excluded by setting $k = 0$ in the equation (2). So now the fitness function will be only a relation between coverage and the number of transceivers.

Table 1 suggests that EVEBO and PBIL outperform DE in fitness value, coverage, and convergence speed. EVEBO is doing significantly better in convergence speed (computational efforts) compared to PBIL. This outperformance becomes noticeably important in problems of larger sizes.

The fitness value, coverage rate, and number of BTSs in PBIL and our proposed EVEBO-based algorithm are the same. This is due to employing the same grid size, the same list of predefined BTS locations, and an identical fitness function. It is important to note that equal optimum results does not mean same solutions since the final optimized BTS locations most probably differ due to non-deterministic nature of these two algorithms.

There is another work which presents its coverage results only in graphical forms ((Calégari et al., 1997)) and for this reason it cannot be used as a comparison reference.

Now we start the circular-coverage BTS case. In our simulation, for ease of visualization we assume that BTSs can be placed on any 3-by-3-pixel boxes which gives us 9604 different potential BTS locations which is also the size of our solution $\mathbf{X}$. To display the circular-coverage area on our adopted discretized

Table 1: A comparison between the best results in (Vega-Rodríguez et al., 2007) and our proposed EVEBO-based method with square-coverage BTSs with 30 independent runs.

Parameters / Method	PBIL	DE	EVEBO
· Fitness Value	204.082	163.48	204.082
· Number of BTSs	49	49	49
· Coverage	100%	89.5%	100%
· Number of Evaluations	276,345	9,363	8,079

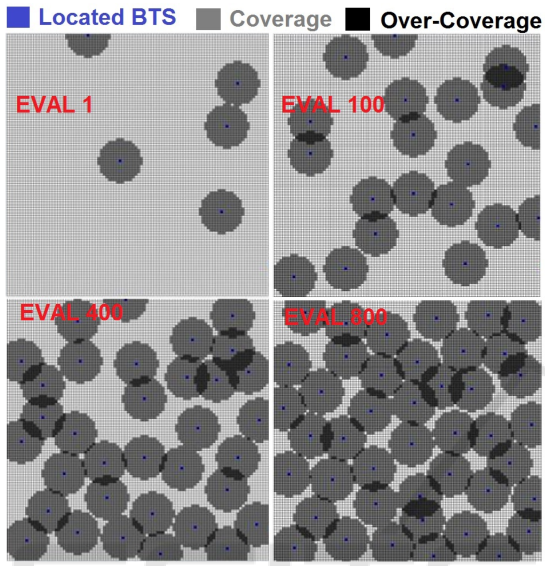

Figure 7: Some simulation evaluations for a specific independent run.

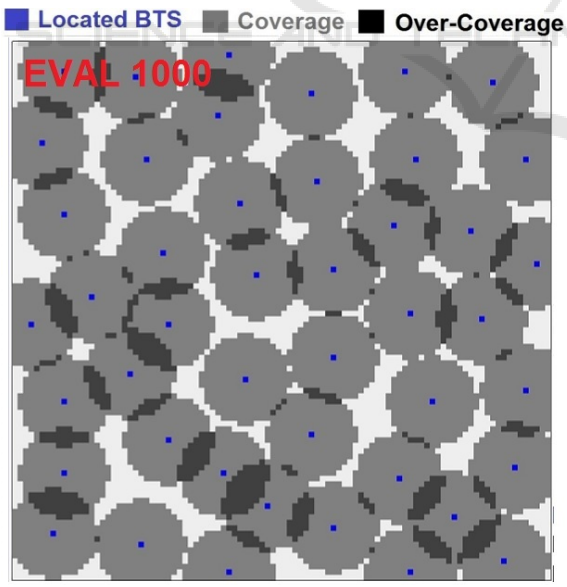

Figure 8: Final result after the 1000th evaluation ($k = 0.5$).

area with square pixels, we use the midpoint circle algorithm in (Pitteway, 1967) and (Van Aken, 1984).

All initialization parameters are tabulated in Table 2. In this case, by adopting $k = 0.5$ the over-coverage factor in the fitness function is also considered. To make results more reliable, each run is repeated 50 times and the corresponding results is shown in Table 3.

The results in Table 3 show that the proposed method does not show much variations throughout 50 runs. While satisfactory coverage and over-coverage rates are achieved, they cannot reach perfection due to their conflicting natures (see Figure 4).

For clarity, some intermediate results corresponding to the 1st, 100th, 400th, and 800th evaluation of an independent run are shown in Figure 7. The final solution corresponding to the 1000th evaluation is illustrated in Figure 8. Also for this run, the average rates of coverage and over-coverage in each evaluation (average rates over all solutions in the population) is calculated and plotted on two diagrams shown in Figure 9. As seen, coverage rate increases almost monotonically while EVEBO tries to keep over-coverage low. Both rates demonstrate stabilization after some number of evaluations.

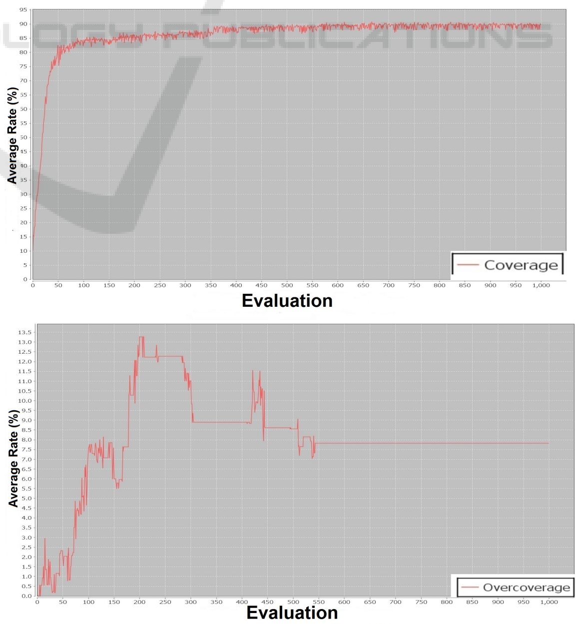

Figure 9: Average rates of coverage and over-coverage over all solutions in the population in each evaluation($k = 0.5$).

We repeated all simulations with combinations of different initial values of parameters in Table 2.

Table 2: Initial parameters values for circular-BTS-coverage localization problem.

Parameter	*Value*
Population size	200
Over-coverage parameter (k)	0.5
Tolerance	4
BTS coverage radius	24 Pixels
Termination condition	# of Evals.>1000

Table 3: Simple statistics of circular-BTS-coverage localization results for 50 separate runs each after 1000 evaluations ($k = 0.5$).

Results	*Min*	*Max*	*Average*
Fitness	141.42	152.35	147.37
Active BTS	42	51	48
Coverage	81.41%	91%	86.53%
Over-Coverage	4.31%	8.06%	5.64%

Specifically, we tried population sizes equal to 100, 400, and 500 and tolerance values of 2, 5, and 10, and we observed no significant changes in final solutions and number of evaluations. This demonstrates, at least in a simulation sense, low sensitivity of EVEBO algorithm to initial values of parameters.

7 CONCLUSION

In this paper, EVEBO was employed to solve the prevalent problem of BTS localization in cellular networks. We show that EVEBO is a good candidate for solving this kind of problems due to its speed of convergence and low sensitivity to initial values of parameters. Simulation results show EVEBO's superiority in comparison to the best of the related works. In addition, we have also tried the more challenging circular-coverage BTS where we show that EVEBO can produce stable results with adequately fast convergence. We have shown that EVEBO can produce better than or repeat the best results in the literature in much less computational effort. Herein, we have also shown good over-coverage rate results of the EVEBO which is not usually addressed in similar works.
For future works, we are interested in running EVEBO to optimize radio network design by considering signal and telecommunication factors in real, larger and more challenging terrains.

REFERENCES

Alba, E. (2004). Evolutionary algorithms for optimal placement of antennae in radio network design. In *Parallel and Distributed Processing Symposium, 2004. Proceedings. 18th International*, page 168. IEEE.

Beyer, H.-G. (2013). *The theory of evolution strategies*. Springer Science & Business Media.

Bianchi, L., Dorigo, M., Gambardella, L. M., and Gutjahr, W. J. (2009). A survey on metaheuristics for stochastic combinatorial optimization. *Natural Computing*, 8(2):239–287.

Calégari, P., Guidec, F., Kuonen, P., and Kobler, D. (1997). Parallel island-based genetic algorithm for radio network design. *Journal of Parallel and Distributed Computing*, 47(1):86–90.

Eiben, J. S. (2003). *Introduction to Evolutionary Computing*. Springer.

Luke, S. (2009). *Essentials of metaheuristics*, volume 113. Lulu Raleigh.

Nelson, J. K. and Gupta, M. R. (2007). An em technique for multiple transmitter localization. In *Information Sciences and Systems, 2007. CISS'07. 41st Annual Conference on*, pages 610–615. IEEE.

Nelson, J. K., Gupta, M. R., Almodovar, J. E., and Mortensen, W. H. (2009). A quasi em method for estimating multiple transmitter locations. *IEEE Signal Processing Letters*, 16(5):354–357.

Nelson, J. K., Hazen, M. U., and Gupta, M. R. (2006). Global optimization for multiple transmitter localization. In *Military Communications Conference, 2006. MILCOM 2006. IEEE*, pages 1–7. IEEE.

Pitteway, M. L. (1967). Algorithm for drawing ellipses or hyperbolae with a digital plotter. *The Computer Journal*, 10(3):282–289.

Pourghanbar, M., Kelarestaghi, M., and Eshghi, F. (2015). Evebo: A new election inspired optimization algorithm. In *Evolutionary Computation (CEC), 2015 IEEE Congress on*, pages 916–924. IEEE.

Simon, D. (2013). *Evolutionary optimization algorithms*. John Wiley & Sons.

Van Aken, J. R. (1984). An efficient ellipse-drawing algorithm. *IEEE Computer Graphics and Applications*, 4(9):24–35.

Vega-Rodríguez, M., Gómez-Pulido, J., Alba, E., Vega-Pérez, D., Priem-Mendes, S., and Molina, G. (2007). Evaluation of different metaheuristics solving the rnd problem. *Applications of Evolutionary Computing*, pages 101–110.

Yu, X. and Gen, M. (2010). *Introduction to evolutionary algorithms*. Springer Science & Business Media.