Proceedings of the International Multiconference on
Computer Science and Information Technology, pp. 65 – 71

ISBN 978-83-60810-14-9
ISSN 1896-7094

Applying Differential Evolution to the Reporting Cells Problem

Sónia Almeida-Luz
Polytechnic Institute of Leiria,
School of Technology and Management
Leiria, Portugal
sluz@estg.ipleiria.pt

Miguel A.Vega-Rodríguez,
Juan A. Gómez-Pulido,
Juan M. Sánchez-Pérez
University of Extremadura, Dept. Technologies
of Computers and Communications
Escuela Politécnica, Campus Universitario
s/n 10071 Cáceres, Spain
{mavega, jangomez,sanperez}@unex.es

***Abstract*—The Location Management problem corresponds to the management of the network configuration with the objective of minimizing the costs involved. It can be defined using several different schemes that principally consider the location update and the paging costs. The Location Area and Reporting Cells are two common strategies used to solve the location management problem. In this paper we present a new approach that uses the Differential Evolution algorithm applied to the reporting cells planning problem, with the objective of minimizing the involved location management costs. With this work we want to define the best values to the differential evolution parameters and respective scheme, using 12 distinct test networks, as well as compare our results with the ones obtained by other authors. The final results obtained with this approach are very encouraging.**

I. Introduction

THE use of mobile networks is growing every day and being applied to the most of newly and renovated applications for data transfer, voice and fax services among many others mobile services. Because of this, communication networks [1] must support a big number of users and their respective applications maintaining a good response without loose quality and availability. With the goal that mobile networks keep this quality it is necessary to consider the mobility management when making design of the network infrastructure.

Mobility management is a very important point because it includes the process of hand off management that enables the mobile network to locate roaming mobile terminals, and also the process of location management that enables the mobile network to find the current location of a mobile terminal in order to make or receive calls. We are mainly concerned with location management because it involves the user movements and tracing, with the objective of minimizing the involved costs.

The location management is partitioned in two main operations: location update that corresponds to the notification of current location, performed by mobile terminals when they change their location in the mobile network, and location paging (inquiry) that represents the operation of determining the location of the mobile user terminal, performed by the network when it tries to direct an incoming call to the user.

There exist several strategies of location management, which are divided in two main groups: static and dynamic schemes [2]. The static schemes consider the same behavior of the network for all users, while the dynamic schemes consider different network topologies for different users based on the individual user's call and mobility patterns. A survey of different dynamic techniques based on users' behavior such as timer-based, distance-based, movement-based (among others) may be seen in [2]. As static techniques, the most common ones are always-update, never-update, and location area schemes [2], [3], among others. The reporting cells (RC) represents another static location management strategy that also is very used.

In this paper we present a new approach to solve the reporting cells problem and minimize the involved costs, using the Differential Evolution (DE) based algorithm. Section II presents an overview of the reporting cells planning and the respective involved costs. In section III, the DE based algorithm is described including its parameters and respective possible schemes. In section IV are explained the details of algorithm implementation and network preparation. Section V includes the experimental results and presents the results produced by the four distinct experiments. In section VI analysis and comparisons with other authors' results and other artificial life techniques are shown. Finally section VII includes conclusions and future work.

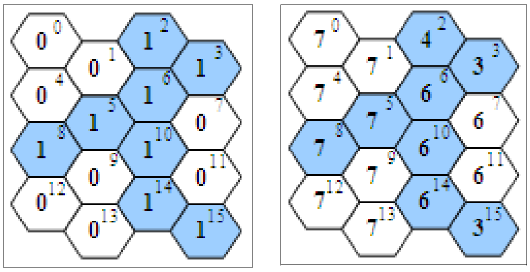
II. Reporting Cells Planning Problem

In this section we will explain the reporting cells scheme and how it is applied in the calculus of the location management cost.

A. Reporting Cells Scheme

The reporting cells planning scheme was proposed by Bar and Kessler in [4] with the objective of minimizing the cost of tracking mobile users.

This strategy is characterized by defining a subset of cells as reporting cells and the others as non-reporting cells (nRC), as it is possible to see in Fig. 1a (RC represented with value 1 and in blue color and nRC represented with value 0 and in white color). The mobility terminals only perform a



a) b)

Fig 1. a) Reporting Cells planning b) Vicinity values

new location update when they change their location and move to one reporting cell. If an incoming call must to be routed to the mobile user, the search can be restricted to his last reporting cell known and their respective neighbors non-reporting cells.

It is necessary to calculate for each cell the vicinity factor, which represents the maximum number of cells that the user must page when an incoming call occurs.

The vicinity value of a reporting cell corresponds to the number of non-reporting cells that are reachable from this reporting cell, without crossing other reporting cells, and adding the reporting cell itself. For example, considering the calculus of vicinity factor for the cell number 5 (RC) in Fig. 1a, we must count the number of neighbors that are nRCs (cells 0, 1, 4, 9, 12 and 13) and also include the RC itself, which makes a total of 7 neighbors. This total number of neighbors will correspond to the vicinity factor of this RC.

If we are calculating the vicinity value of a non-reporting cell it is necessary to consider the maximum vicinity value among the reporting cells from where this one can be reached. This means that if a non-reporting cell belongs to the neighborhood of more than one reporting cell, the calculus has to be done for all the reporting cells and then, the maximum number is set as the vicinity factor of the respective non-reporting cell. If we consider the cell number 9 (nRC), in Fig. 1a, we can observe that it belongs to the neighborhood of at least two RCs (more precisely four cells: number 5, 8, 10 and 14). Because of that, the calculus of vicinity factor must be done for all those RCs and after this, the maximum number will be considered as the vicinity factor for this nRC. The vicinity factor for cells 5, 8, 10 and 14 is respectively 7, 7, 6 and 6, so the maximum value that represents the vicinity factor of cell number 9 is 7.

Considering the reporting cells planning of the Fig. 1a and calculating all the vicinity factors, the result will be the one presented on Fig. 1b.

B. Location Management Cost

The location management (LM) cost is principally divided in two fundamental operations of location update and location paging. The location update (LU) cost corresponds to the cost involved with the location updates performed by mobile terminals in the network, when they change their location and must register the new one. The location paging (P) cost is caused by the network when it tries to locate a user's mobile terminal, during the location inquiry, and normally the number of paging transactions is directly related to the number of incoming calls. LM involves several other parameters and components that are considered to be equal for all strategies and does not make influence when comparing the results obtained by different strategies. Because of that these cost are not considered for the total cost.

From earlier studies and experiments [3], [5] we have seen that the generic formula to obtain the LM cost is:

$$Cost = \beta * N_{LU} + N_P \qquad (1)$$

The total cost of location updates is given by N_{LU}, the total cost of paging operations corresponds to N_P and β is a ratio constant used in a location update relatively to a paging transaction in the network. To each location update is imputed a much higher cost than to each paging operation, because of the complex process that must be executed for each location update performed, and also because most of the time a mobile user moves without making any call [6]. Due to all of that, the cost of a location update is normally considered to be 10 times greater than the cost of paging, that is, β=10 [5].

In the reporting cells scheme the location updates only are performed when a mobile user enters in a reporting cell and the vicinity factor of each cell must be considered. Because of that the generic formula given by (1) must be readjusted and it is formulated as [7]:

$$Cost = \beta * \sum_{i \in s} N_{LU}(i) + \sum_{i=0}^{N} N_P(i) * V(i) \qquad (2)$$

Here we can see that $N_{LU}(i)$ corresponds to the number of location updates associated to the reporting cell i, S indicates the subset of cells defined as reporting cells, $N_P(i)$ is the number of incoming calls attributed for cell i, N is the total number of cells that compound the mobile network configuration and $V(i)$ is the vicinity factor attributed for cell i.

We will use this formula with the objective of minimize the LM cost, using the reporting cells strategy.

III. Differential Evolution

The Differential Evolution (DE) is a population-based algorithm, created by Ken Price and Rainer Storn [8], whose main objective is functions optimization. This algorithm is one strategy based on evolutionary algorithms that has some specific characteristics. It has a key strategy to generate new individuals by calculating vector differences between other randomly-selected individuals of the population.

A. DE Parameters

DE algorithm uses four important parameters: population size *NI*, mutation *F*, crossover *Cr* and selection operators as well as different schemes presented in the following sub-section. For further information about the four parameters, refer to [8].

B. DE Schemes

DE can be implemented using 10 different schemes suggested by Price and Storn [8]. These schemes, exposed in Table I, are classified based on notation *DE/x/y/z*, where *x* specifies the vector to be mutated, *y* correspond to the number of difference vectors used in mutation of *x* (normally 1 or 2) and *z* represents the crossover scheme. The vector *x* may be chosen randomly ('rand') or as the best of current popula-

TABLE I. DE SCHEMES

Scheme	Mutant vector generation
DE/best/1/exp	$x_i = x_{best} + F(x_{r1} - x_{r2})$
DE/rand/1/exp	$x_i = x_{r3} + F(x_{r1} - x_{r2})$
DE/randtobest/1/exp	$x_i = x_{r3} + F(x_{best} - x_{r3}) + F(x_{r1} - x_{r2})$
DE/best/2/exp	$x_i = x_{best} + F(x_{r1} + x_{r2} - x_{r3} - x_{r4})$
DE/rand/2/exp	$x_i = x_{r5} + F(x_{r1} + x_{r2} - x_{r3} - x_{r4})$
DE/best/1/bin	$x_i = x_{best} + F(x_{r1} - x_{r2})$
DE/rand/1/bin	$x_i = x_{r3} + F(x_{r1} - x_{r2})$
DE/randtobest/1/bin	$x_i = x_{r3} + F(x_{best} - x_{r3}) + F(x_{r1} - x_{r2})$
DE/best/2/bin	$x_i = x_{best} + F(x_{r1} + x_{r2} - x_{r3} - x_{r4})$
DE/rand/2/bin	$x_i = x_{r5} + F(x_{r1} + x_{r2} - x_{r3} - x_{r4})$

tion ('best') and z may be binomial ('bin') or exponential ('exp') depending on the type of crossover used.

C. DE Algorithm

The pseudo-code of the DE algorithm, using the *DE/best/1/exp* is presented in Fig. 2. It starts by defining and evaluating the initial population through calculating the fitness value for each individual. After that, until the termination condition is not reached, the necessary individuals are picked and a new one is produced according to the selected DE scheme and respective rules. This new individual is evaluated and compared with the old one. Just the one with the best fitness value will be chosen and pass for population of the next generation.

IV. IMPLEMENTATION DETAILS

In this section we explain the fitness function implemented to evaluate the solutions obtained, present the test networks used and expose the original definition of parameters.

```
Initialize the population
Evaluate the initial population
While (termination condition not satisfied) {
        Randomly select ind. xr1 ≠ xbest
        Randomly select ind. xr2 ≠ xr1 and ≠ xbest
        Generate trial ind.: xtrial = xbest + F(xr1 – xr2)
        Use Cr to define the amount of genes changed
            in trial individual
        Evaluate the trial individual
        Deterministic selection
}
```

Fig 2. DE Algorithm Pseudo Code with Scheme *DE/best/1/exp*

A. Fitness Function

In this study the fitness function is used for measuring the total location management cost of each potential solution, which is defined according to the equation (2). This means that for each potential solution generated, it is calculated the fitness value, which corresponds to the network configuration by means of reporting cells and non-reporting cells.

B. Test Networks

Other authors have studied the reporting cells strategy, but most of them do not present the test networks used, so it is not possible to compare our approach with them. However, in [7] it is presented a set of twelve test networks, representing four groups defined by size, that have been generated and are available in [9] as benchmark. In this work we used these twelve test networks with the objective of compare final results. In Table II it is shown, as an example, the test network 1 that represents a 4x4 cells configuration. The first column indicates the cell identification, the second column corresponds to the number of location updates *NLU* and the third represents the number of incoming calls *NP*.

C. Parameters Definition

The initial definition of parameters is an important step because it represents the basis for the algorithm evolution. First it is defined the initial population of candidate solutions that corresponds to the individuals.

Each individual is compound by N genes, where the N value is the number of cells in the network and each gene represents the information about the cell type, which can be a reporting cell or a non-reporting cell.

To define the initial population we have set, with a probability of fifty percent, the type of each cell as RC or nRC.

Initially it is also necessary to set the DE algorithm parameters and that has been done with a number of individuals NI equal to 10, the mutation factor F set to 0.5 and the crossover value Cr defined as 0.1. For the DE scheme it has been selected the *DE/rand/1/bin*. The number of generations represents the terminal condition when the algorithm is executed and it is set to 1000.

Throughout the different experiments, the parameters values have been adjusted with the specific objective of obtaining the best results for each test network.

TABLE II. TEST NETWORK 1

Cell	NLU	NP	Cell	NLU	NP
0	452	484	**8**	647	366
1	767	377	**9**	989	435
2	360	284	**10**	1105	510
3	548	518	**11**	736	501
4	591	365	**12**	529	470
5	1451	1355	**13**	423	376
6	816	438	**14**	1058	569
7	574	415	**15**	434	361

TABLE III. EXPERIMENT 1: DETERMINING THE BEST NI

Test Network	NI – Fitness Evaluation									
N. (Dim)	10	25	50	75	100	125	150	175	200	225
1 (4x4)	99,137	100,881	98,535	98,535	98,535	98,535	98,535	98,535	98,535	98,535
2 (4x4)	101,250	98,879	97,156	97,156	97,156	97,156	97,156	97,156	97,156	97,156
3 (4x4)	98,106	101,403	95,038	95,038	95,038	95,038	95,038	95,038	95,038	95,038
4 (6x6)	195,283	185,092	176,800	173,701	173,701	173,701	173,701	173,701	173,701	173,701
5 (6x6)	205,859	192,426	185,937	182,331	182,331	182,331	182,331	182,331	185,059	182,331
6 (6x6)	193,540	186,423	175,321	174,519	174,519	174,519	174,519	174,519	174,519	174,519
7 (8x8)	338,196	321,575	315,097	310,888	308,853	309,342	308,401	308,401	308,991	308,401
8 (8x8)	319,912	304,750	294,548	287,149	289,051	287,149	287,149	287,149	289,935	287,149
9 (8x8)	292,467	276,299	270,171	265,272	264,204	264,204	264,204	264,316	264,204	264,204
10 (10x10)	425,866	405,127	394,168	388,206	386,775	387,551	386,695	386,474	387,543	386,893
11 (10x10)	390,793	377,363	361,299	360,210	361,581	359,224	358,778	359,224	359,697	358,944
12 (10x10)	401,704	385,022	380,846	375,233	376,631	374,711	375,001	375,722	373,733	374,220

V. EXPERIMENTAL RESULTS

In this section we explain the four distinct experiments applied to each test network with the objective of study in more detail the best configuration of DE. For each experiment, and for all combination of parameters, 30 independent runs have been performed in order to assure its statistical relevance. In each experiment the final results, of the best fitness values obtained (lower cost value), are presented and explained the decisions taken.

A. Experiment 1 – Determining NI

The number of individuals that will compound the initial population must be the first experiment because it is the basis of the algorithm implementation. In order to accomplish that, we have fixed, as referred in *IV.C. Parameters Definition,* the values of crossover *Cr*=0.1, the mutation *F*=0.5, DE scheme as *DE/rand/1/bin* and the stop criterion as 1000 generations, considering our experience from earlier experiments that we have performed [3].

With this experiment we have concluded that, increasing of *NI* value, it is possible to observe a positive evolution of the results (see fitness results in Table III), but just until the value of *NI* =175, because after that we start observing worse results and stop increase in *NI* =225. Considering this and the average evolution we concluded that *NI* =175 would be the elected value for the second experiment.

B. Experiment 2 – Determining Cr

The second experiment has the objective of selecting the *Cr* value that obtains the best results. To proceed with this experiment we fixed the value of *NI* to 175 (from experiment 1), and maintained the other parameters as defined in the beginning of experiment 1.

This experiment has been executed using different values for *Cr*: 0.1, 0.25, 0.50, 0.75 and 0.90. Analyzing the results obtained, we could conclude that best values were obtained with *Cr*=0.1 and *Cr*=0.25. Because of that, with the objective of taking more complete conclusions, we decided to execute the algorithm with values 0.15 and 0.20. Finally, as it is possible to see in , we could conclude that *Cr*=0.15 is the one that performs better.

TABLE IV. EXPERIMENT 2: DETERMINING THE BEST CR

Test Network	Cr – Fitness Evaluation						
N. (Dim)	0.1	0.15	0.20	0,25	0.50	0.75	0.90
1 (4x4)	98,535	98,535	98,535	98,535	98,535	98,535	98,535
2 (4x4)	97,156	97,156	97,156	97,156	97,156	97,156	97,258
3 (4x4)	95,038	95,038	95,038	95,038	95,038	95,038	98,216
4 (6x6)	173,701	173,701	173,701	173,701	173,701	177,647	177,889
5 (6x6)	182,331	182,331	187,990	183,264	184,679	183,991	185,966
6 (6x6)	174,519	174,519	174,519	175,321	175,182	175,321	178,255
7 (8x8)	308,401	308,401	308,401	311,646	313,378	313,607	319,069
8 (8x8)	287,149	287,149	289,573	289,051	293,248	302,812	309,609
9 (8x8)	264,204	264,204	265,452	264,786	272,249	266,876	275,489
10 (10x10)	387,318	386,681	388,357	386,959	393,510	393,492	420,650
11 (10x10)	360,262	358,669	360,072	360,128	360,596	367,508	374,405
12 (10x10)	373,695	374,966	374,554	374,921	377,190	383,782	391,001

TABLE V. EXPERIMENT 3 - DETERMINING THE BEST F

Test Network	F – Fitness Evaluation				
N. (Dim)	0.1	0,25	0.50	0.75	0.90
1 (4x4)	98,535	98,535	98,535	98,535	98,727
2 (4x4)	97,156	97,156	97,156	97,156	97,156
3 (4x4)	95,038	95,038	95,038	95,038	95,038
4 (6x6)	174,112	173,701	173,701	173,701	176,530
5 (6x6)	182,331	182,331	182,331	182,331	182,331
6 (6x6)	174,519	175,321	174,519	174,519	174,519
7 (8x8)	310,162	310,426	308,401	311,492	308,401
8 (8x8)	293,093	304,911	287,149	292,913	295,557
9 (8x8)	265,494	264,643	264,204	268,312	265,750
10 (10x10)	388,849	389,438	386,681	389,125	387,533
11 (10x10)	359,221	360,072	358,669	358,167	361,441
12 (10x10)	373,298	375,087	374,966	371,829	375,232

C. *Experiment 3 – Determining F*

The determination of the best value for mutation, *F*, is the purpose of the third experiment. So, in order to perform this experiment it was fixed the value of *NI* to 175 (from experiment 1), the value of *Cr* to 0.15 (from experiment 2) and the others maintained as in the two earlier experiments.

After finishing these executions and examining the results (see Table V) we conclude that *F*=0.5 is the one that permits to obtain the best results.

D. *Experiment 4 – Determining DE Scheme*

Finally, with this fourth experiment we pretend to select the most adequate DE scheme, the one that permits to obtain the best results (lower fitness value). For that, we fixed the best values for each parameter (defined in the three earlier experiments) as: *NI*=175, *Cr*=0.15 and *F*=0.5, and executed the algorithm applying the ten DE schemes presented in section *III.B*.

Once finished all the executions, and observing the respective results shown in Table VI, it was possible to conclude that the scheme *DE/rand/1/bin* is the one with a better performance, because it is the one that obtains better fitness values for all the test networks. With these results we may say that the binomial schemes perform better than the exponential ones and that it is also better to choose randomly the individuals used to create the trial individual.

Finishing these four experiments we had determined the best DE configuration, applied to the reporting cells plan ning problem, setting the parameters as *NI* =175, *Cr* =0.15, *F* =0.5 and *DE/rand/1/bin* as the most adequate DE scheme.

VI. ANALYSIS AND COMPARISONS

In this section we pretend to analyze the results obtained, compare them with those shown in [7] and present the configuration for best solutions.

Finally we apply the best DE configuration to the test networks presented in [10] and compare results produced with this DE configuration, with the ones of other artificial life techniques.

A. *Analysis and Comparison of Results*

Analyzing the experimental results we could conclude that with this approach it is possible to obtain the same minimum fitness values (considered optimal in [7]), as the ones obtained in [7] with a Hopfield Neural Network with Ball Dropping (HNN+BD) and a Geometric Particle Swarm Optimization (GPSO) for ten of the twelve test networks used.

For the other two test networks the results are very similar because: for the test-network-10 our fitness value is 386,951 and in [7] the one obtained by the HNN+BD is 386,351; and for the test-network-12 our fitness value is 371,829 and in [7] the value obtained by HNN+BD and GPSO is 370,868. Relatively to the average values it is possible to say that they are very similar.

In fig. 3 the configuration for each test-network solution is shown and it is possible to observe that most of them split each one in subnetworks.

B. *Comparison with other Artificial Life Techniques*

After obtaining the best DE configuration applied to the reporting cells planning problem we decide to apply it in

TABLE VI. EXPERIMENT 4: DETERMINING THE DE SCHEME

Test Network	DE Scheme – Fitness Evaluation									
	Exponential Crossover					Binomial Crossover				
N. (Dim)	Best1	Rand1	RandTBest1	Best2	Rand2	Best1	Rand1	RandTBest1	Best2	Rand2
1 (4x4)	98,535	98,535	98,535	98,535	99,008	98,535	98,535	98,535	98,727	98,535
2 (4x4)	97,156	97,156	97,156	97,156	97,156	97,156	97,156	97,156	97,156	97,156
3 (4x4)	95,038	95,038	95,038	95,038	95,038	95,038	95,038	95,038	95,038	95,038
4 (6x6)	173,701	173,701	173,701	176,530	178,038	176,041	173,701	173,701	173,701	173,701
5 (6x6)	182,331	182,331	187,801	190,779	191,279	182,331	182,331	182,331	182,331	182,331
6 (6x6)	174,519	175,182	183,992	177,276	177,892	181,850	174,519	174,519	174,519	174,519
7 (8x8)	322,973	319,772	328,327	323,391	332,472	320,236	308,401	308,401	309,855	308,730
8 (8x8)	304,214	307,139	313,010	310,708	316,849	305,236	287,149	287,149	287,149	287,149
9 (8x8)	277,408	279,177	290,646	291,684	289,936	269,984	264,204	264,316	265,164	264,353
10 (10x10)	420,701	423,017	420,452	421,353	425,228	394,176	386,681	386,951	393,471	386,695
11 (10x10)	385,950	380,824	384,661	387,273	380,241	366,156	358,167	359,486	367,202	359,517
12 (10x10)	394,636	388,468	395,290	395,404	394,767	379,227	371,829	376,015	379,544	376,165

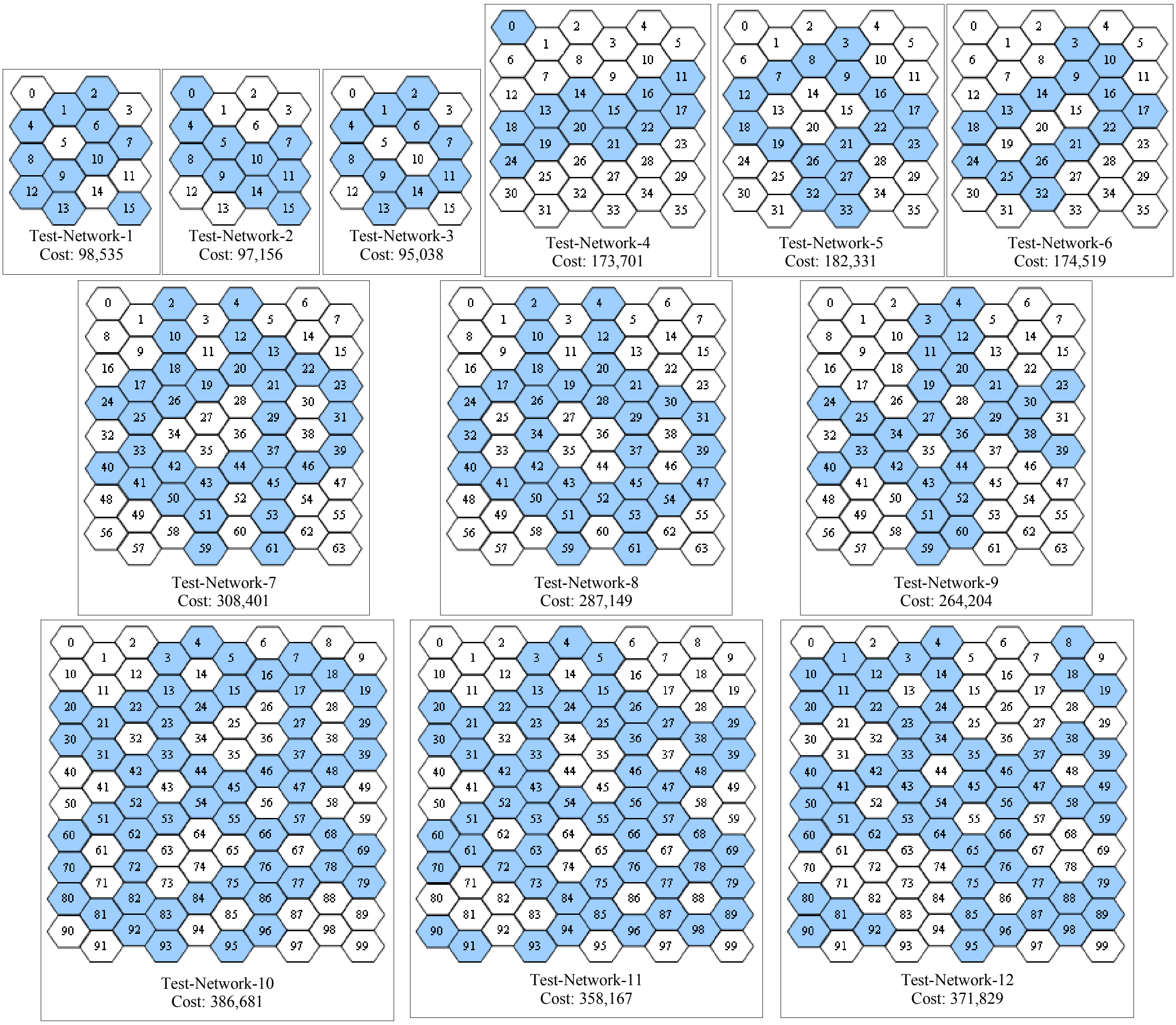


Fig 3. Test-Network Solution with Reporting Cells Configuration

other test networks, and analyze the performance with the objective of comparing with other artificial life techniques. We decided to use 3 test networks presented in [10] (also referred in [7]) in order to compare our results with those produced by the application of Genetic Algorithms (GA), Tabu Search (TS) and Ant Colony algorithm (AC).

Making the additional experiments we conclude that our approach performs well. Using the Test-Network-1 (4×4 instance provided in [10]) we obtain the same fitness value as the GA, TS and AC, that is, 92,883 with a total of 10 reporting cells.

Our approach generates a better solution when solving the Test-Network-2 (6×6 instance provided in [10]), comparing with GA and AC. The fitness value obtained by the GA is 229,556 with a total of 26 reporting cells in the network [7], [10], while, the cost obtained by DE in this work is 211,278 with 24 reporting cells. TS presents the same cost 211,278 and the fitness value obtained by AC is 211,291.

Finally, for the Test-Network-3 (8×8 instance in [10]) DE again surpasses the results obtained in [10] by the GA, TS and AC. Concretely, the fitness value obtained by the GA and TS is 436,283, the one obtained by AC is 436,886 while, the cost obtained by DE in this work is 436,269 with a total of 39 reporting cells.

VII. Conclusion

In this paper we have discussed the use of differential evolution algorithm (DE) applied to the reporting cells planning problem. To the best of our knowledge, this is the first time that DE is employed for this task (this is another contribution of this paper).

We have studied in detail the best configuration of DE including parameters and scheme. After more than 10,000 runs, they are *NI*=175, *Cr*=0.15, *F*=0.5 and *DE/ran/1/bin* as the best DE scheme.

We have shown that our approach produces interesting results because when compared with ones of other authors, that use HNN+BD and GSPO, they are equal or very similar.

Comparing the performance of DE algorithm with other artificial life techniques as genetic algorithm (GA), tabu search (TS), and ant colony algorithm (AC), we may say that it performs well because improves the results obtained by those ones.

As future work we pretend applying other evolutionary algorithms to the RC problem comparing their results with the ones accomplished by the DE algorithm. We also have the intention of testing our approach with test networks generated by using real data.

Acknowledgment

This work was developed in part thanks to the project OPLINK (TIN2005-08818-C04-03). Thanks also to the Polytechnic Institute of Leiria, for the economic support offered to Sónia Almeida-Luz to make this research.

References

[1] K. Pahlavan, A. H. Levesque, *Wireless Information Networks* , John Wiley & Sons, Inc, 1995.

[2] V. W. S. Wong, V. C. M. Leung, "Location Management for Next-Generation Personal Communications Networks", IEEE Network, Oct. 2000, vol. 14, no. 5, pp. 18-24.

[3] Almeida-Luz, M. A. Vega-Rodríguez, J. A. Gómez-Pulido, J. M. Sánchez-Pérez, "Defining the Best Parameters in a Differential Evolution Algorithm for Location Area Problem in Mobile Networks", In EPIA 2007 – 13th Portuguese Conference on Artificial Intelligence; (Eds). APPIA, Dec. 2007, ISBN: 978-98-995-6180-9; pp. 219-230.

[4] A. Bar-Noy, I. K. "Tracking mobile users in wireless communications networks", *IEEE Transactions on Information Theory* , 1993, vol. 39, pp. 1877 – 1886.

[5] J. Taheri, A. Y. Zomaya, "A Genetic Algorithm for Finding Optimal Location Area Configurations for Mobility Management", *30th Anniversary of the IEEE Conference on Local Computer Networks (LCN)*, Nov. 2005, pp. 568-577.

[6] P. R. L. Gondim, "Genetic Algorithms and the Location Area Partitioning Problem in Cellular Networks", *IEEE 46th Vehicular Technology Conf. Mobile Technology for the Human Race* , May 1996, vol. 3, pp. 1835-1838.

[7] E. Alba, J. García-Nieto, J. Taheri, A. Zomaya, "New Research in Nature Inspired Algorithms for Mobility Management in GSM Networks", EvoWorkshops 2008, LNCS 4974, pp. 1-10, Springer-Verlag Berlin Heidelberg, 2008.

[8] Price, R. Storn: Web Site of Differential Evolution: http://www.icsi.berkeley.edu/~storn/code.html (on June 2008)

[9] Test networks benchmark: http://oplink.lcc.uma.es/problems/mmp.html (on June 2008)

[10] R. Subrata, A. Y. Zomaya, "A comparison of three artificial life techniques for reporting cell planning in mobile computing", IEEE Trans. Parallel Distrib, Syst. 14(2), 142–153, 2003.