New Research in Nature Inspired Algorithms for Mobility Management in GSM Networks

Enrique Alba[1], José García-Nieto[1], Javid Taheri[2], and Albert Zomaya[2]

[1] Dept. de Lenguajes y Ciencias de la Computación, University of Málaga, ETSI Informática, Campus de Teatinos, Málaga - 29071, Spain
{eat,jnieto}@lcc.uma.es

[2] School of Information Technologies, University of Sydney, Sydney, NSW 2006, Australia
{javidt,zomaya}@it.usyd.edu.au

Abstract. Mobile Location Management (MLM) is an important and complex telecommunication problem found in mobile cellular GSM networks. Basically, this problem consists in optimizing the number and location of paging cells to find the lowest location management cost. There is a need to develop techniques capable of operating with this complexity and used to solve a wide range of location management scenarios. Nature inspired algorithms are useful in this context since they have proved to be able to manage large combinatorial search spaces efficiently. The aim of this study is to assess the performance of two different nature inspired algorithms when tackling this problem. The first technique is a recent version of Particle Swarm Optimization based on geometric ideas. This approach is customized for the MLM problem by using the concept of Hamming spaces. The second algorithm consists of a combination of the Hopfield Neural Network coupled with a Ball Dropping technique. The location management cost of a network is embedded into the parameters of the Hopfield Neural Network. Both algorithms are evaluated and compared using a series of test instances based on realistic scenarios. The results are very encouraging for current applications, and show that the proposed techniques outperform existing methods in the literature.

Keywords: Mobile Location Management, GSM Cellular Networks, Geometric Particle Swarm Optimization, Hopfield Neural Network.

1 Introduction

Mobility Management becomes a crucial issue when designing infrastructure for wireless mobile networks. In order to route incoming calls to appropriate mobile terminals, the network must keep track of the location of each mobile terminal. Mobility management requests are often initiated either by a mobile terminal movement (crossing a cell boundary) or by deterioration of the quality of a received signal in a currently allocated channel. Due to the expected increase in the usage of wireless services in the future, the next generation of mobile networks should be able to support a huge number of users and their bandwidth requirements [1,4].

Several strategies for Mobility Management have been used in the literature being the location area (LA) scheme one of the most popular [6,11]. An analogous strategy is the *Reporting Cells* (RC) scheme suggested in [3]. In RC, a subset of cells in the network is designated as reporting cells. Each mobile terminal performs a location update only when it enters one of these reporting cells. When a call arrives, the search is confined to the reporting cell the user last reported and the neighboring bounded nonreporting cells. It was shown in [3] that finding an optimal set of reporting cells, such that the location management cost is minimized, is an NP-complete problem. For this reason, bioinspired algorithms have been commonly used to solve this problem [7,10].

In this work, we use two nature inspired algorithms to assign the reporting cells of a network following the RC scheme. The first algorithm, called Geometric Particle Swarm Optimization (GPSO), is a generalization of the Particle Swarm Optimization for virtually any solution representation, which works according to a geometric framework. The second technique combines a Hopfield Neural Network with a Ball Dropping (HNN+BD) mechanism. Our contributions are both to perform better with respect to existing works and to introduce the GPSO algorithm for solving Telecommunications problems. In addition, these two techniques are experimentally assessed and compared from different points of view such as quality of the solutions, the robustness and design issues.

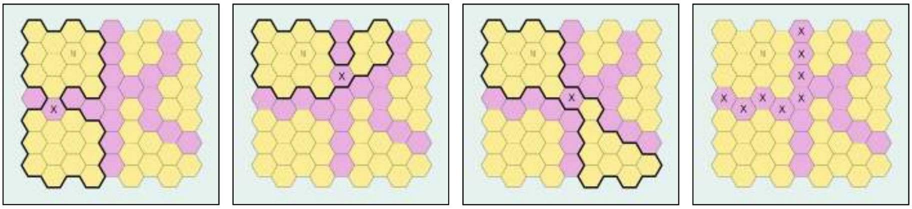
The remaining of the paper is organized as follows: Section 2 briefly explains the Mobility Management problem. The two algorithms, GPSO and HNN+BD, are described in sections 3 and 4 respectively. After that, Section 5 presents a number of experiments and results that show the applicability of the proposed approaches to this problem. Finally, conclusions are drawn in Section 6.

2 The Mobility Management Problem

Basically, the Mobility (location) Management problem consists in reducing the total cost of managing a mobile cellular network. Two factors take part when calculating the total cost: the updating cost and the paging cost. The updating cost is the portion of the total cost due to location updates performed by roaming mobile terminals in the network. The paging cost is caused by the network during a location inquiry when the network tries to locate a user[1].

According to the reporting cells scheme, there are two types of cells: reporting cells (RC) and non-reporting cells (nRC). A neighborhood is assigned to each reporting cell, which consists of all nRC that must also page the user in case of an incoming call. For both RC and nRC, a *vicinity* factor is calculated representing the maximum number of reporting neighbors for each cell that must page the user (including the cell itself) in case of an incoming call. Obviously, the vicinity factor of each RC is the number of neighbors it has (see Fig. 1).

[1] Other costs like the cost of database management to register user's locations or the cost of the wired network (backbone) that connects the base stations to each other were not considered here, since these costs are assumed to be the same for all location management strategies and hence aren't contemplated in comparisons.



(a) (b) (c) (d)

Fig. 1. Cells marked as 'N' belong to the neighborhoods of at least three RCs (grey cells). For example, the number of neighbors for cell 'X' is 25, 17, and 22 for (a), (b) and (c) respectively (25 to consider the worst case). However, if a nRC belongs to more than two neighborhoods the calculation must be done for all of them, and then, the maximum number is considered as the vicinity factor for this nRC. For example, the nRC marked as 'N' is a part of the neighborhood of all cells marked as 'X' in (d).

For nRC, the vicinity factor is calculated based on the fact that each nRC might be in the neighborhood of more than one RC, the maximum number of paging neighbors that contains such a cell is considered its vicinity factor.

Therefore, to calculate the total cost of the network location management, the general cost function is formulated as:

$$Cost = \beta \times \sum_{i \epsilon S} N_{LU}(i) + \sum_{i=0}^{N} N_P(i) \times V(i) \tag{1}$$

where, $N_{LU}(i)$ is the number of location updates for reporting cell number i, $N_P(i)$ is the number of arrived calls for cell i, $V(i)$ is the vicinity factor for cell i, S is the set of cells defined as reporting cells, and N is the total number of cells in the network. β is a constant representing the cost ratio of a location update to a paging transaction in the network (typically $\beta = 10$). This function is used either as *fitness function* by the GPSO or *energy function* by the HNN.

3 Geometric Particle Swarm Optimization

The recent Geometric Particle Swarm Optimization (GPSO) [5,2], enables us to generalize PSO to virtually any solution representation in a natural and straightforward way, extending the search to richer spaces, such as combinatorial ones. This property was demonstrated for the cases of Euclidean, Manhattan and Hamming spaces in the referenced work.

The key issue in this approach consists of using a multi-parental recombination of particles which leads to the generalization of a *mask-based crossover* operation, proving that it respects four requirements for being a *convex combination* in a certain space (see [5] for a complete explanation). This way, the mask-based crossover operation substitutes the classical *movement* in PSO, based on the *velocity* and *position update* operations, only suited for continuous spaces.

For Hamming spaces, which is the focus of this work, a *three-parent mask-based crossover* (3PMBCX) was defined in a straightforward way:

Definition 1. *Given three parents* a*,* b *and* c *in* $\{0, 1\}^n$*, generate randomly a crossover mask of length* n *with symbols from the alphabet* $\{a, b, c\}$*. Build the offspring* o *filling each position with the bit from the parent appearing in the crossover mask at the position.*

In a convex combination, the weights w_a, w_b and w_c indicate for each position in the crossover mask the probability of having the symbols a, b or c.

The pseudocode of the GPSO algorithm for Hamming spaces is illustrated in Algorithm 1. For a given particle i, three parents take part in the 3PMBCX operator (line 13): the current position x_i, the social best position g_i and the historical best position found h_i (of this particle). The weight values w_a, w_b and w_c indicate for each element in the crossover mask the probability of having values from the parents x_i, g_i or h_i respectively. A constriction of the geometric crossover forces w_a, w_b and w_c to be non-negative and add up to one.

Algorithm 1. GPSO for Hamming spaces

1: $S \leftarrow SwarmInitialization()$
2: **while** not stop condition **do**
3:    **for** each particle x_i of the swarm S **do**
4:       evaluate(x_i)
5:       **if** $fitness(x_i)$ is better than $fitness(h_i)$ **then**
6:          $h_i \leftarrow x_i$
7:       **end if**
8:       **if** $fitness(h_i)$ is better than $fitness(g_i)$ **then**
9:          $g_i \leftarrow h_i$
10:       **end if**
11:    **end for**
12:    **for** each particle x_i of the swarm S **do**
13:       $x_i \leftarrow 3PMBCX((x_i, w_a), (g_i, w_b), (h_i, w_c))$
14:       mutate(x_i)
15:    **end for**
16: **end while**
17: **Output:** best solution found

Since the GPSO for Mobility Management was developed for Hamming space, each particle i of the swarm consists of a binary vector $x_i = (x_{i1}, x_{i2}, ..., x_{in})$ representing a reporting cell configuration, where each element x_{ij} represents a cell of the network; x_{ij} can have a value of either "0", representing a nRC, or "1", representing a RC. For example, in an 6×6 network, the particle position will have a length (n) of 36.

4 Hopfield Neural Network with Ball Dropping

In this approach, the Ball Dropping technique is used as the backbone of the algorithm that employs the HNN as its optimizer, and is inspired by the natural behavior of individual balls when they are dropped onto a non-even plate (a plate with troughs and crests). As can be expected, the balls will spontaneously move to the concave areas of the plate, and in a natural process, find the minimum of the plate. A predefined number of balls are dropped onto several random positions on the plate, which is equivalent to the random addition of a predefined number of paging cells to the current paging cell configuration of the network.

As a result, after dropping a number of balls on the plate the energy value of the network increases suddenly and the HNN optimizer tries to reduce it by moving the balls around. The following procedure summarizes the basic form of this algorithm.

Algorithm 2. Ball Dropping Mechanism

1: Drop a predefined number of balls onto random positions
2: **repeat**
3: Shake the plate
4: Remove unnecessary balls
5: **until** location of balls does not lead to any better configuration
6: **Output:** best solution found

In relation to Equation 1, the state vector of the HNN, 'X', is considered to have two different components for location updates and call arrival as follows:

$$X = [x_0 \;\; x_1 \;\; \wedge \;\; x_{N-1} \;\; x_N \;\; x_{N+1} \;\; \wedge \;\; x_{2N-1}]^T \tag{2}$$

where x_0 to x_{N-1} is the location updates part, x_N to x_{2N-1} is the call arrival part and 'N' is the total number of cells in the network. This HNN model is designed to represents a RC configuration network, and then, tries to modify its RCs in order to reduce the total cost gradually. To summarize this explanation, we refer the reader to [8] where other aspects like generating a initial solution generation, definition of function to modify the state vector and reduction of the number of variations are given completely.

5 Simulation Results

In this section we present the experiments conducted to evaluate and compare the proposed GPSO and HNN+BD. We firstly give some details of the test network instances used. The experiments with both algorithms are presented and analyzed afterwards. We have made 10 independent runs for each algorithm and instance. Comparisons are made from different points of view such as the performance, robustness, quality of solutions and even design issues concerning the two algorithms. Finally, comparisons with other optimizers found in the literature are encouraging since our algorithms obtain competitive solutions which even beat traditional metaheuristic techniques in the previous state of the art.

5.1 Test GSM Network Instances

In almost all of the previous research in the literature, the cell attributes of the network are generated randomly. In general, two independent attributes for each cell are considered: the number of call arrivals (NP) and the number of location updates (NLU), which are set at random according to a normal distribution. However, these numbers are highly correlated in real world scenarios. Therefore, in this work, a more robust and realistic approach is used to seed the initial solutions, and consequently, the network attributes of each cell [9]. This makes the configuration of the solutions obtained in this work to be more realistic.

Therefore, a benchmark of twelve different instances were generated here to be used for testing GPSO and HNN+BD. The numeric values shaping the test networks configurations are given in tables below[2] for future reproduction of our results.

Test-Network 4			Test-Network 5			Test-Network 6		
Cell	*NLU*	*NP*	Cell	*NLU*	*NP*	Cell	*NLU*	*NP*
0	335	97	0	373	86	0	859	659
1	944	155	1	958	155	1	1561	621
2	588	103	2	264	99	2	450	93
3	1478	500	3	571	119	3	599	98
4	897	545	4	431	132	4	535	151
5	793	495	5	451	97	5	425	138
6	646	127	6	693	153	6	1219	590
7	1159	119	7	1258	149	7	1638	137
8	1184	115	8	847	112	8	991	114
9	854	95	9	1412	173	9	646	72
10	1503	529	10	1350	163	10	587	97
11	753	140	11	711	135	11	361	94
12	744	120	12	356	81	12	559	101
13	819	103	13	951	171	13	787	110
14	542	61	14	2282	1016	14	1738	191
15	476	103	15	2276	1067	15	1433	165
16	937	117	16	1217	139	16	562	87
17	603	69	17	341	96	17	404	63
18	617	90	18	337	87	18	342	79
19	888	102	19	1210	121	19	595	97
20	452	53	20	2228	979	20	1312	164
21	581	86	21	1104	171	21	1129	92
22	773	86	22	718	99	22	884	102
23	741	125	23	362	113	23	630	138
24	693	131	24	669	119	24	306	80
25	1535	576	25	1189	158	25	593	87
26	921	128	26	1032	157	26	603	82
27	1225	73	27	620	93	27	977	136
28	1199	133	28	893	140	28	1354	122
29	710	139	29	596	112	29	1225	641
30	782	464	30	367	74	30	421	158
31	879	477	31	389	108	31	594	163
32	1553	532	32	418	120	32	689	99
33	613	68	33	220	102	33	569	115
34	1044	121	34	799	120	34	1554	631
35	400	148	35	344	117	35	733	534

Test-Network 7			Test-Network 8			Test-Network 9		
Cell	*NLU*	*NP*	Cell	*NLU*	*NP*	Cell	*NLU*	*NP*
0	354	160	0	293	88	0	225	85
1	819	198	1	651	134	1	692	128
2	214	75	2	239	53	2	471	124
3	394	147	3	470	73	3	776	104
4	238	135	4	379	69	4	478	106
5	505	99	5	1089	435	5	1034	152
6	433	134	6	690	435	6	931	678
7	397	134	7	615	416	7	890	807
8	588	164	8	509	137	8	445	124
9	895	121	9	557	68	9	866	137
10	658	129	10	472	68	10	1068	136
11	636	121	11	481	80	11	699	112
12	462	104	12	678	100	12	737	108
13	925	134	13	860	124	13	796	120
14	1017	163	14	1229	446	14	1569	706
15	339	86	15	851	401	15	520	117
16	398	122	16	328	71	16	324	93
17	657	95	17	527	77	17	651	94
18	945	122	18	551	86	18	754	75
19	1088	161	19	708	64	19	582	83
20	828	148	20	626	109	20	552	99
21	995	130	21	640	69	21	570	98
22	687	128	22	924	108	22	809	103
23	295	114	23	507	86	23	384	92
24	324	101	24	334	74	24	330	85
25	652	153	25	1187	171	25	588	89
26	1130	142	26	868	74	26	652	117
27	2558	912	27	1324	512	27	584	89
28	1445	191	28	666	86	28	570	107
29	959	151	29	775	87	29	540	84
30	602	133	30	842	60	30	620	88
31	314	92	31	358	50	31	298	85
32	311	123	32	366	75	32	376	102
33	632	127	33	1545	149	33	659	140
34	1250	155	34	1148	92	34	604	98
35	2470	991	35	1239	420	35	577	100
36	2299	847	36	1406	469	36	522	77
37	1051	188	37	1088	104	37	558	88
38	602	140	38	1203	154	38	615	101
39	350	124	39	304	76	39	336	88
40	282	81	40	646	56	40	381	112
41	796	135	41	1215	92	41	763	129
42	1226	147	42	758	91	42	639	99
43	1076	149	43	646	103	43	565	103
44	1301	172	44	885	101	44	567	117
45	909	128	45	780	78	45	765	104
46	622	128	46	1024	169	46	641	119
47	413	105	47	307	74	47	345	96
48	367	115	48	937	477	48	566	148
49	1125	143	49	1308	544	49	1579	716
50	1053	127	50	879	110	50	852	149
51	585	126	51	682	87	51	876	104
52	701	118	52	533	62	52	789	144
53	722	109	53	527	70	53	1126	126
54	856	96	54	602	69	54	948	164
55	646	184	55	454	123	55	485	134
56	422	136	56	666	463	56	905	756
57	426	122	57	703	454	57	1000	744
58	568	142	58	1118	465	58	1100	179
59	264	138	59	353	133	59	429	83
60	480	143	60	474	67	60	902	109
61	223	92	61	258	54	61	536	114
62	734	114	62	629	131	62	706	113
63	341	153	63	273	102	63	253	102

Test-Network 1			Test-Network 2			Test-Network 3		
Cell	*NLU*	*NP*	Cell	*NLU*	*NP*	Cell	*NLU*	*NP*
0	452	484	0	280	353	0	488	455
1	767	377	1	762	438	1	765	290
2	360	284	2	686	599	2	271	201
3	548	518	3	617	503	3	626	475
4	591	365	4	447	403	4	550	247
5	1451	1355	5	978	560	5	1572	1479
6	816	438	6	1349	648	6	1010	377
7	574	415	7	562	431	7	635	300
8	647	366	8	608	412	8	526	240
9	989	435	9	1305	681	9	962	422
10	1105	510	10	966	508	10	1643	1545
11	736	501	11	466	408	11	642	274
12	529	470	12	664	503	12	570	485
13	423	376	13	710	530	13	249	196
14	1058	569	14	746	473	14	842	354
15	434	361	15	282	336	15	516	488

Test-Network 10			Test-Network 11			Test-Network 12		
Cell	*NLU*	*NP*	Cell	*NLU*	*NP*	Cell	*NLU*	*NP*
0	144	83	0	461	619	0	392	562
1	304	98	1	665	584	1	551	509
2	201	66	2	534	554	2	440	466
3	266	85	3	449	89	3	441	83
4	137	100	4	172	91	4	200	49
5	206	80	5	339	84	5	430	45
6	127	79	6	201	93	6	280	90
7	393	112	7	438	89	7	347	84
8	162	46	8	186	63	8	109	30
9	187	116	9	144	64	9	98	43
10	265	82	10	542	553	10	452	502
11	552	99	11	803	515	11	723	467
12	565	83	12	884	528	12	813	440
13	467	95	13	552	75	13	721	99
14	277	114	14	388	62	14	572	60
15	444	109	15	384	68	15	643	82
16	387	95	16	417	77	16	600	92
17	752	83	17	559	95	17	547	95
18	457	76	18	403	90	18	289	77
19	271	84	19	247	60	19	205	74
20	249	80	20	233	79	20	544	441
21	468	90	21	408	90	21	842	446
22	469	74	22	550	83	22	1008	417
23	612	103	23	538	93	23	683	88
24	571	114	24	431	57	24	614	69
25	1335	678	25	604	99	25	501	85
26	802	112	26	347	65	26	702	123
27	656	87	27	404	91	27	644	95
28	731	124	28	539	75	28	469	77
29	274	86	29	290	69	29	296	64
30	367	104	30	248	103	30	617	457
31	533	125	31	540	107	31	911	412
32	429	84	32	423	76	32	989	365
33	542	83	33	526	74	33	472	69
34	1306	708	34	840	107	34	428	65
35	1308	615	35	822	152	35	306	70
36	773	120	36	404	52	36	421	76
37	468	107	37	413	68	37	482	75
38	597	81	38	501	71	38	441	67
39	374	99	39	376	113	39	276	68
40	866	780	40	608	434	40	387	74
41	1050	697	41	1120	586	41	586	82
42	523	105	42	581	90	42	591	94
43	588	113	43	449	62	43	357	67
44	687	113	44	489	70	44	321	66
45	735	132	45	489	97	45	289	47
46	634	97	46	516	96	46	318	66
47	449	99	47	592	86	47	453	58
48	595	133	48	600	67	48	454	77
49	852	699	49	703	496	49	278	81
50	852	768	50	705	573	50	294	80
51	595	97	51	693	110	51	477	83
52	507	86	52	573	99	52	514	90
53	687	101	53	525	93	53	309	48
54	728	123	54	503	86	54	265	51
55	825	154	55	503	71	55	325	73
56	628	109	56	522	78	56	348	64
57	528	91	57	642	91	57	595	102
58	1097	667	58	1076	589	58	569	80
59	894	735	59	639	490	59	383	100
60	374	82	60	380	83	60	278	66
61	523	94	61	577	100	61	455	69
62	468	73	62	466	88	62	540	81
63	891	130	63	415	94	63	438	79
64	1414	692	64	790	115	64	310	63
65	1368	669	65	841	123	65	429	82
66	653	115	66	590	81	66	473	83
67	445	88	67	437	49	67	1070	450
68	590	99	68	481	92	68	901	414
69	385	100	69	249	94	69	659	483
70	309	74	70	267	60	70	288	53
71	647	104	71	555	109	71	481	97
72	717	96	72	426	58	72	705	125
73	878	104	73	422	60	73	675	127
74	1367	653	74	640	91	74	476	47
75	602	128	75	502	75	75	629	70
76	709	100	76	535	90	76	757	90
77	603	91	77	571	95	77	1041	434
78	530	99	78	403	81	78	912	395
79	288	72	79	239	85	79	596	499
80	317	93	80	276	80	80	190	37
81	462	82	81	403	84	81	306	69
82	793	116	82	575	71	82	558	120
83	430	105	83	460	77	83	579	102
84	455	117	84	385	69	84	668	99
85	294	94	85	385	77	85	544	68
86	526	108	86	585	98	86	743	88
87	619	120	87	881	492	87	815	490
88	580	101	88	751	408	88	736	440
89	261	72	89	496	566	89	517	587
90	169	98	90	150	79	90	113	41
91	178	99	91	169	70	91	140	59
92	378	91	92	394	100	92	342	81
93	118	89	93	199	99	93	256	64
94	214	77	94	357	93	94	461	70
95	123	79	95	212	84	95	212	57
96	264	67	96	477	83	96	484	76
97	232	115	97	573	585	97	470	470
98	344	87	98	639	570	98	542	419
99	162	82	99	450	615	99	374	459

[2] Four groups of Test-Network (TN) instances: (1)TN1-2-3 with 4×4 cells; (2)TN4-5-6 with 6×6 cells; (3)TN7-8-9 with 8×8 cells; (4)TN10-11-12 with 10×10 cells. TN files are available in URL `http://oplink.lcc.uma.es/problems/mmp.html`.

5.2 Experimental Results

We have conducted different experiments with several configurations of GPSO and HNN+BD depending on the test network used. Since the two algorithms perform quite different operations, we have set the parameters (Table 1) after preliminary executions of the two algorithms (with each instance) where the computational effort in terms of time and number of evaluations was balanced.

Table 1. Parameter settings for HNN+BD and GPSO. The columns indicate: the number of dropping balls (*N.DroppBalls*) and the number of trials (*N.Trials*) for HNN+BD. For GPSO are reported: the number of particles (*N.Particles*), the crossover probability (P_{cross}), the mutation probability (P_{mut}) and the weighted values (w_a, w_b and w_c).

Test Network	HNN+BD		GPSO			
Dim.	*N.DroppBalls*	*N.Trials*	*N.Particles*	P_{cross}	P_{mut}	$w_a + w_b + w_c$
(4×4)	7	3	20	0.9	0.1	0.33+0.33+0.33
(6×6)	10	5	50			
(8×8)	15	5	100			
(10×10)	15	5	120			

After the initial experimentation, several results were obtained; they are shown in Table 2. The first column contains the number and dimension (in parenthesis) of each test network. Three values are presented for each evaluated algorithm: the best cost (out of 10 runs), the average cost (*Aver.*) of all the solutions, and the deviation (*Dev.*) percentage from the best cost.

As it can be seen from the results, the two algorithms have similar performance in almost all of the instances, although there are a few differences for the large test networks. For example, GPSO obtains better solutions in Test-Network 7 and 10, while, HNN+BD obtains a better solution in Test-Network 11. In addition, it can be noticed that the deviation percentage from the best cost is generally lower in GPSO than in HNN+BD, specially for the smaller test networks. This behavior leads us to believe that the GPSO approach is more robust than HNN+BD, but just slightly.

Table 2. Results for Test Networks obtained by HNN+BD and GPSO

Test Network	HNN+BD			GPSO		
No.(Dim.)	*Best*	*Aver.*	*Dev.*	*Best*	*Aver.*	*Dev.*
1 (4×4)	98,535	98,627	0.09%	98,535	98,535	0.00%
2 (4×4)	97,156	97,655	0.51%	97,156	97,156	0.00%
3 (4×4)	95,038	95,751	0.75%	95,038	95,038	0.00%
4 (6×6)	173,701	174,690	0.56%	173,701	174,090	0.22%
5 (6×6)	182,331	182,430	0.05%	182,331	182,331	0.00%
6 (6×6)	174,519	176,050	0.87%	174,519	175,080	0.32%
7 (8×8)	308,929	311,351	0.78%	308,401	310,062	0.53%
8 (8×8)	287,149	287,149	0.00%	287,149	287,805	0.22%
9 (8×8)	264,204	264,695	0.18%	264,204	264,475	0.10%
10 (10×10)	386,351	387,820	0.38%	385,972	387,825	0.48%
11 (10×10)	358,167	359,036	0.24%	359,191	359,928	0.20%
12 (10×10)	370,868	374,205	0.89%	370,868	373,722	0.76%

Another obvious difference between HNN+BD and GPSO lies in the behavior of each algorithm. This can be observed in Fig. 2, where we show a graphical representation of algorithm runs for the different evaluated networks. Each graph, corresponding to one of the twelve test networks, plots a representative trace of the execution of each algorithm tracking the best solution obtained versus the number of iterations. On the one hand, GPSO shows a typical behavior in evolutionary metaheuristics, that is, it tends to converge from the solutions in the initial population to an optimal reporting cell arrangement. Graphically, the GPSO operation is represented by a monotonous decreasing (minimization) curve. On the other hand, HNN+BD carries out a different searching strategy, as from the initialization, it provokes frequent shaking scenarios in the population with the purpose of gradually diversifying and intensifying the search. These "shakes" are carried out by means of the Ball Dropping technique (Section 4) when no improvement in the overall condition of the network is detected, so the frequency of this operation is variable.

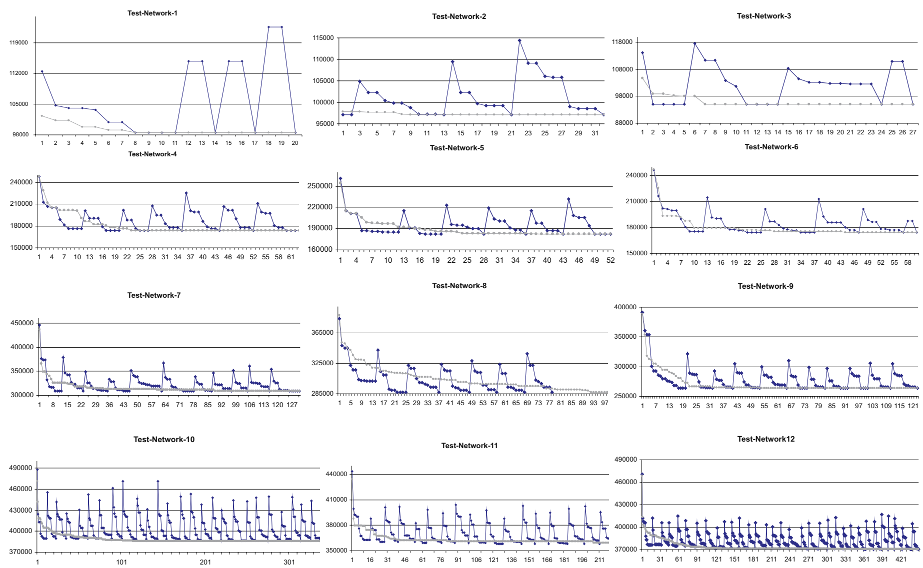


Fig. 2. Cost values level (Y axis) versus iterations (X axis) of all the test networks. Each graphic plots the energy level obtained, we track the evolution of the HNN+BD algorithm (black line with peaks and valleys), and the fitness level in the evolution of the GPSO algorithm (concave grey curve).

Evidently, as Fig. 2 shows, the number of drops in larger test networks is higher than in smaller ones, since the number of iterations required here to converge is also higher. Graphically, this behavior produces intermittent peaks and valleys in the evolution line.

From the point of view of the quality of solutions, as expected, optimal reporting cell configurations for all test networks split the network into smaller sub-networks by clustering the full area. This property can be seen in the large instances in a much clearer way than in the short ones (Fig. 3).

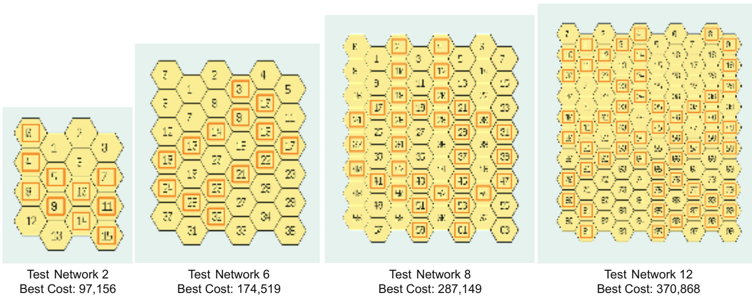


Fig. 3. Paging Cells (with squares) configurations obtained as solutions by the two algorithms (the same solutions) in Test Network 2, Test Network 6, Test Network 8 and Test Network 12. Neighborhood area clusters are easily visible in larger instances. All the legends show the Best Cost found by both algorithms.

5.3 Comparison with Other Optimizers

To the best of our knowledge a Genetic Algorithm (GA) is the only algorithm that can be compared against in this work. The modeling of the problem, the quality of the initial population, and the number of iterations are the main design issues that can affect the performance of the GA. When comparing the proposed approaches with a GA implementation given in [7], one can observe two advantages in terms of convergence and quality of solution in our two new approaches.

Despite the general good behavior of the GA, our two approaches generate a better solution when solving the Test-Network-2 (6×6 instance provided in [7]) in additional experiments. The energy value obtained by the GA is 229,556 with a total of 26 paging cells in the network, while, the cost obtained by HNN+BD in this work is 211,278 with 24 paging cells, and the GPSO obtained a cost of 214,313 with 23 paging cells. With respect to HNN+BD, a reasonable explanation for this difference could be due to the setup parameters used for the GA in [7]. However, our GPSO uses a similar setup parameters compared to the GA, providing a better solution with a smaller number of paging cells.

6 Conclusions

This paper addresses the use of two nature inspired approaches to solve the Mobile Location Management problem found in telecommunications: a new binary Particle Swarm Optimization algorithm called GPSO, and an algorithm based on a Hopfield Neural Network hybridized with the Balls Dropping Technique.

The problem is described and tackled following the Reporting Cells Scheme. In addition, the design and operation of HNN+BD and GPSO are discussed. Twelve test networks of different dimensions, generated following realistic scenarios of mobile networks, were for the first time used in this work. In addition, a comparison of the algorithms is carried out focusing on the performance, robustness, and design issues.

In conclusion, simulation results are very encouraging and show that the proposed algorithms outperform existing methods. Both approaches prove themselves as very powerful optimizers providing fast and good quality solutions.

This work has been carried out as a continuation of previous works where metaheuristics techniques were applied to solve the Mobile Location Management problem. For further work, we are interested in evaluating new test networks under different conditions of topology and dimension. In addition, new experiments will be carried out using different location area schemes.

References

1. Agrawal, D.P., Zeng, Q.-A.: Introduction to Wireless and Mobile Systems. Thomson Brooks/Cole Inc. (2003)
2. Alba, E., García-Nieto, J., Jourdan, L., Talbi, E.-G.: Gene Selection in Cancer Classification using PSO/SVM and GA/SVM Hybrid Algorithms. In: IEEE Congress on Evolutionary Computation CEC-2007, Singapore (September 2007)
3. Bar-Noy, A., Kessler, I.: Tracking mobile users in wireless communication networks. In: INFOCOM (3), pp. 1232–1239 (1993)
4. Lin, Y.-B., Chlamatac, I.: Wireless and Mobile Network Architecture. John Wiley and Sons, Chichester (2001)
5. Moraglio, A., Di Chio, C., Poli, R.: Geometric Particle Swarm Optimization. In: Ebner, M., O'Neill, M., Ekárt, A., Vanneschi, L., Esparcia-Alcázar, A.I. (eds.) EuroGP 2007. LNCS, vol. 4445, Springer, Heidelberg (2007)
6. Subrata, R., Zomaya, A.Y.: Location management in mobile computing. In: ACS/IEEE International Conference on Computer Systems and Applications, pp. 287–289 (2001)
7. Subrata, R., Zomaya, A.Y.: A comparison of three artificial life techniques for reporting cell planning in mobile computing. IEEE Trans. Parallel Distrib. Syst. 14(2), 142–153 (2003)
8. Taheri, J., Zomaya, A.Y.: The use of a hopfield neural network in solving the mobility management problem. In: IEEE/ACS International Conference on Pervasive Services, ICPS 2004, pp. 141–150 (July 2004)
9. Taheri, J., Zomaya, A.Y.: Realistic simulations for studying mobility management problems. Int. Journal of Wireless and Mobile Computing 1(8) (2005)
10. Taheri, J., Zomaya, A.Y.: A genetic algorithm for finding optimal location area configurations for mobility management. In: LCN 2005: Proceedings of the The IEEE Conference on Local Computer Networks 30th Anniversary, Washington, DC, USA, pp. 568–577. IEEE Computer Society, Los Alamitos (2005)
11. Wu, H.-K., Jin, M.-H., Horng, J.-T., Ke, C.-Y.: Personal paging area design based on mobile's moving behaviors. In: Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, INFOCOM 2001, vol. 1, pp. 21–30 (2001)