Wireless Netw (2014) 20:1909–1924
DOI 10.1007/s11276-014-0718-x

Solving the location areas management problem with multi-objective evolutionary strategies

Víctor Berrocal-Plaza · Miguel A. Vega-Rodríguez · Juan M. Sánchez-Pérez

Published online: 8 April 2014

Abstract The location management is one of the most important tasks in current Public Land Mobile Networks because of the number of mobile subscribers has increased exponentially over the last decade. That is why systems that automatically optimize the operations involved in the location management (subscriber location update and paging) are becoming more necessary. There are several works in which different metaheuristics have been applied to optimize the location management tasks. In these works, the objective functions of the location update and paging were linearly combined into a single objective function with the goal of optimizing these two tasks by using Single-Objective Optimization Algorithms. In this paper, in order to avoid the drawbacks associated with the linear aggregation of the objective functions, we have adapted and modified two Multi-objective Evolutionary Algorithms: Non-dominated Sorting Genetic Algorithm II and Strength Pareto Evolutionary Algorithm 2. Furthermore, we have performed an in-depth analysis of the Location Areas scheme and its relation to the user's call and mobility patterns. This study concludes that the location areas are as small as possible due to the fast increase of the paging cost, and that the cells with higher mobile activity are located in the center of its location area. Moreover, results show that our algorithms outperform the single-objective optimization algorithms proposed by other authors in the two most complex test networks, as well as the advantages of using a multi-objective approach.

Keywords Evolutionary computation · Multi-objective optimization · Location Areas scheme · Mobility management · Mobile network

1 Introduction

There has been a huge increment in the number of mobile subscribers in the last decade due to services that are provided by next generation mobile networks, e.g. voice and data traffic (including real-time applications) regardless of the subscriber position and the moment at which they are requested. That is possible because the Public Land Mobile Networks (PLMNs) are characterized by a high level of capillarity and the systematic reuse of frequencies to serve a large traffic demand with limited radioelectric resources. For it, the PLMNs divide the geographical coverage area into several smaller areas known as cells [1–4]. Each cell is associated with a Base Station (BS, the network entity which provides access to the subscriber terminal). Therefore, and with the aim of properly redirecting the incoming calls, every mobile network must control the subscribers' movement across the network cells.

Commonly, the PLMNs are structured according to a three-level hierarchical scheme [1]. The first level, known as Mobile Subnet, is formed by all user terminals, or Mobile Station (MS). The second one, called Access Subnet or Radio Network, is constituted by all Base Station Systems (BSSs), that is, all BSs and its Base Station

V. Berrocal-Plaza (✉) · M. A. Vega-Rodríguez ·
J. M. Sánchez-Pérez
Department of Technologies of Computers and
Communications, University of Extremadura Escuela
Politécnica, Campus Universitario S/N, 10003 Cáceres, Spain
e-mail: vicberpla@unex.es

M. A. Vega-Rodríguez
e-mail: mavega@unex.es

J. M. Sánchez-Pérez
e-mail: sanperez@unex.es

Controllers (BSCs, the network entities that, in a centralized manner, perform the control tasks associated with several BSs). And the third level, or Core Network, includes the set of systems and registries related to network tasks, e.g. the Mobile Switching Centers (MSCs), the Home Location Registries (HLRs), and the Visitor Location Registries (VLRs) are the network entities that control the location management tasks. The MSC is the network entity specialized in performing tasks as mobile location, paging, and interoperability with other networks. The HLR is the registry in which the information related to the subscriber terminal is stored (identification code, subscriber number, user location, etc.). And the VLR is the register that stores information of the users served by its associated MSC.

All network entities interact through an advanced signaling system to make feasible the location update, paging, and handover (automatic switching of the radio-electric resources associated with a call-in process when the subscriber moves to another cell) [5]. The location update and paging are two of the most important management tasks in current Public Land Mobile Networks. In fact, Nowoswiat and Milliken show in [6] that the signaling traffic associated with the location update and paging in a large PLMN is more than 33 % of the total signaling load. That is the reason why the use of optimization techniques (such as metaheuristics) to reduce the signaling load of these two management tasks is an interesting research line. Usually, the location update and paging are treated jointly by the location management system to track the subscriber movement inside the mobile network. In the manual location update, or manual registration, every MS periodically measures the control channel associated with nearby BSs and it tunes in the control channel with the highest signal strength. Through this channel, the MS sends its identification information to the MSC, which stores the subscriber information (user location and user identification) in its VLR and its HLR associated. On the other hand, the paging procedure is performed to route the incoming calls. For this purpose, the network must find the cell in which the callee subscriber is located. In this procedure, the HLR is interrogated to know the MSC related to that MS and then, this MSC sends paging broadcast messages around the last geographical area associated with the MS in question [4]. Note that the location management tasks are highly dependent on the number of mobile subscribers, since an increment of subscribers involves a growth of the user mobility among cells and an increase of the traffic density.

The manual registration simplifies the location management tasks, but it leads to a huge signaling load associated with the location update, since it is performed each time the MS changes its cell. Therefore, other location management strategies have been developed. They can be classified into two main groups: static and dynamic location update methods [5, 7]. In dynamic schemes, different subscribers perceive a different logical topology of the network. These logical topologies are dependent on the user's call and mobility patterns. On the other hand, static schemes are characterized for providing the same logical topology to all subscribers. These last schemes are more popular than dynamic ones because they require fewer network capabilities. Examples of static strategies are: always update, never update, and location areas.

Furthermore, there are several strategies to manage the paging procedure, mainly classified in two groups: without delay constraint and with delay constraint [5]. In systems without delay constraint, the paging procedure is performed regardless of the execution time. Therefore, in systems with delay constraint, the paging procedure must be performed before the timer expires, known as *maximum paging delay*. These last systems have been widely researched because they are strategies used in real mobile networks. Examples of paging procedures with delay constraint are: blanket-polling, two-cycle sequential paging, and shortest distance.

This paper focuses on researching the Location Areas (LAs) scheme, because it is a strategy widely applied in current PLMNs [5] and it presents planning issues similar to those that we will find in next-generation networks (e.g. Tracking Areas in Long Term Evolution, LTE). The LAs strategy exploits the cell architecture of the mobile network by grouping cells into logical areas (or logical clusters). Thus, the subscriber location only is updated when the user moves to another location area. For it, every BS periodically broadcast its *cell global identification*, a packet which contains (among others) the Cell Id (CI) and the Location Area Code (LAC) [3]. The Location Areas scheme reduces the number of location updates with respect to the manual registration, but it complicates the paging procedure because the subscriber must be searched in the whole location area. Therefore, the main challenge of this location management strategy is to find the configurations of Location Areas that minimize the number of location updates and the number of paging messages.

In the literature, there are several works in which the LAs Planning Problem (LAPP) was optimized with different metaheuristics of the single-objective optimization field. For it, the objective functions of the LAPP were linearly combined into a single objective function. Gondim in [8] was one of the first authors to argue that the LAPP is an NP-hard combinatorial optimization problem due to the size of the objective space. That is why he proposed a Genetic Algorithm (GA) for finding quasi-optimal configurations of LAs. Demestichas et al. in [9] developed three Single-objective Optimization Algorithms (SOAs: Tabu

Search (TS), Simulated Annealing (SA), and GA) to solve the LAPP in different environments. Regrettably, a fair comparison with [8] and [9] cannot be conducted because the test networks used in these papers are not available. Subsequently, Taheri and Zomaya in [7, 10–13] provided a set of test networks in which the subscribers' call and mobility patterns are close to those that we can find in real world networks. In order to solve these test networks, they proposed several single-objective optimization algorithms: GA, SA, Hopfield Neural Networks (HNNs), and combinations of GA with HNNs (GA-HNNs). Recently, S. M. Almeida-Luz et al. applied in [14] the Differential Evolution algorithm (DE) to solve the test networks provided by Taheri and Zomaya in [7, 10–13].

However, the linear aggregation of the objective functions has several important drawbacks (see Sect. 2). That is why we propose the use of multi-objective metaheuristics. Furthermore, a multi-objective optimization algorithm (in contrast to single-objective optimization algorithms) provides a set of solutions among which the network operator could select the one that best adjusts to the network real state, e.g. when the signaling load associated with other network operations is considered. A very preliminary version of this work was presented at the conferences [15–17], these works have been considerably extended with: the improvement of our multi-objective algorithms (please, see for example Sect. 3.6), which leads to an improvement of our previous results, a complete statistical analysis (see Sect. 4, and particularly Sect. 4.1) comparing our versions of the Non-dominated Sorting Genetic Algorithm II (NSGAII) and the Strength Pareto Evolutionary Algorithm 2 (SPEA2), and an in-depth study of the Location Areas scheme and its relation to the user's call and mobility patterns (see Sect. 4.2, and the corresponding figures).

The paper is organized as follows. Section 2 defines the LAs scheme. The main features of a multi-objective optimization algorithm and our versions of NSGA-II and SPEA2 are presented in Sect. 3. Results, comparisons with other authors, and an analysis of the solutions obtained are discussed in Sect. 4. Finally, Sect. 5 summarizes our conclusions.

2 Location areas management scheme

The Location Areas (LAs) scheme is one of the most common strategies to solve the location management problem automatically. In this strategy, the network cells are grouped into logical areas (known as location areas) with the aim of reducing the signaling load associated with the subscriber location update. Thus, a location update is only performed when a subscriber moves to a new Location Area. Consequently, the network knows the location of its subscribers at a Location Area level, and hence, the paging is only carried out in the networks cells within the last updated Location Area [5]. Note that these two procedures (location update and paging) are conflicting. To reduce the number of location updates (or location update cost, $\mathrm{LU}_{\mathrm{cost}}$), the size of the location areas should be as large as possible, leading to an increase of the number of paging messages (or paging cost, $\mathrm{PA}_{\mathrm{cost}}$), since a higher number of cells must be paged. For example, in the hypothetical case of an only one location area (never update strategy), the $\mathrm{LU}_{\mathrm{cost}}$ is zero and the $\mathrm{PA}_{\mathrm{cost}}$ is maximum because the subscriber must be searched in the whole network. On the other hand, the $\mathrm{PA}_{\mathrm{cost}}$ is reduced to a minimum when each cell belongs to a different location area (always update strategy) because the network would know the cell associated with every subscriber at each moment. Nevertheless, this would increase the $\mathrm{LU}_{\mathrm{cost}}$. Then, we can conclude that a high $\mathrm{LU}_{\mathrm{cost}}$ involves a low $\mathrm{PA}_{\mathrm{cost}}$, and vice versa. That is, the Location Areas Planning Problem defines a multi-objective optimization problem in which the main challenge is to find the configuration of LAs that simultaneously minimize the number of location updates and the number of paging messages. In this paper, we study test networks that have been defined for LAs and the blanket-polling paging (i.e. all cells within the location area of the callee subscriber are polled simultaneously) [7, 10–13]. The objective functions related to the location update ($\boldsymbol{f}_1$) and paging procedure ($\boldsymbol{f}_2$) can be formulated as:

$$\boldsymbol{f}_1 = \min\left\{ LU_{cost} = \sum_{i=1}^{NCell} \sum_{j=1}^{NCell} \rho_{i,j} \times N_{i,j} \right\}, \tag{1}$$

$$\boldsymbol{f}_2 = \min\left\{ PA_{cost} = \sum_{k=1}^{NArea} \sum_{i \in CA_k} NIC_i \times NA_k \right\}, \tag{2}$$

subject to

$$NArea \leq NCell, \tag{3}$$

$$1 \leq NA_k \leq NCell, \tag{4}$$

$$\sum_{k=1}^{NArea} \mu_{i,k} = 1, \quad \forall i \in [1, Ncell], \tag{5}$$

where the involved variables are: $\mathrm{N}_{\mathrm{i,j}}$: Number of subscribers that move from cell i to cell j. $\rho_{\mathrm{i,j}}$: Binary variable that indicates whether the cell i and cell j belong to different location areas (if it is equal to 1) or not (if it is equal to 0). $\mathrm{NIC}_{\mathrm{i}}$: Number of incoming calls of the cell i. NA_{k}: Number of cells belonging to location area k. CA_{k}: Vector that stores the cells associated with the location area k. NCell: Number of cells. NArea: Number of location areas. $\mu_{\mathrm{i,k}}$: Binary variable that indicates whether the cell i belongs to location area k (if it is equal to 1) or not (if it is equal to 0).

Constraints (3) and (4) establish that the maximum number of location areas and the maximum size of a location area are limited to the number of cells. Constraint (5) establishes that the multilevel LAs scheme is not allowed [18], i.e. a cell must only belong to one location area. It should be noted that the total signaling traffic related to the mobile location management also involves other important factors (such as the cost of database management, the cost associated with the communication among network entities, the handover cost, and the packet delivery cost) [19–22]. However, in a non-cooperative mobile network, the signaling traffic generated in the Radio Network is considered to be sufficient to compare different location management strategies [19, 20].

In previous works, the LAPP was solved by using single-objective optimization algorithms [7–14]. For it, the $\mathrm{LU}_{\mathrm{cost}}$ and $\mathrm{PA}_{\mathrm{cost}}$ were combined into an only objective function equal to the linear sum of weighted costs. Equation (6) presents the objective function used by these Single-objective Optimization Algorithms (SOAs), where β is a ratio constant defined to assign a higher priority to one of the two costs, and it is commonly configured as 10.

$$f^{SOA}(\beta) = \min\{Cost_{tot}(\beta) = \beta \times LU_{cost} + PA_{cost}\}. \qquad (6)$$

Several drawbacks arise when the LAs scheme is tackled with single-objective optimization algorithms. Firstly, a very accurate knowledge of the problem is required, since the obtained solutions are very sensitive to the values of the weight coefficients. Secondly, it is possible that the proper value of β might be different depending on the load of the signaling network. And thirdly, a single-objective optimization algorithm must perform an independent run for each value of β. That is why, in this paper, we propose the use of multi-objective optimization algorithms to solve the Location Areas Planning Problem. By means of this technique, we obtain simultaneously a set of solutions (each one related to a β value) among which the network operator could select the one that best adjusts to the real state of the mobile network at the same time as the $\mathrm{LU}_{\mathrm{cost}}$ and the $\mathrm{PA}_{\mathrm{cost}}$ are reduced to a minimum.

3 Multi-objective Evolutionary Algorithms (MOEAs)

Many real-world problems involve two or more conflicting objectives [23], as the Location Areas Planning Problem in mobile location management. In this kind of problems, the search of a single solution (typical of Single-objective Optimization Problems, SOPs) is replaced by the search of a wide range of solutions, each one related to a specific trade-off among objectives. Such problems are commonly referred as Multi-objective Optimization Problems (MOP) and can be formally represented as a quadruple (X, Z, $\boldsymbol{f}$, *mode*), where X is defined as the decision space, Z denotes the objective space, $\boldsymbol{f} : X \rightarrow Z$ consists in several objective functions f_{i} which assign to each decision vector $\mathbf{x} \in X$ an objective vector $\mathbf{z} = \boldsymbol{f}(\mathbf{x}) \in Z$ under the constraints established by $e(\mathbf{x})$, and *mode* defines the optimization mode (maximization or minimization) for each f_{i} [24]. Thus, a MOP can be formulated as follows:

$$\begin{array}{ll} \textit{Optimize} & \mathbf{z} = \boldsymbol{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})), \\ \textit{subject to} & e(\mathbf{x}) = (e_1(\mathbf{x}), e_2(\mathbf{x}), \ldots, e_m(\mathbf{x})) \leq 0, \\ \textit{where} & \mathbf{x}^i = \left(\mathrm{x}_1^i, \mathrm{x}_2^i, \ldots, \mathrm{x}_n^i\right) \in X, \\ & \mathbf{z}^i = \left(\mathrm{z}_1^i, \mathrm{z}_2^i, \ldots, \mathrm{z}_k^i\right) \in Z. \end{array} \qquad (7)$$

The set of decision vectors that meet the constraints $e(\mathbf{x})$ defines the feasible decision space (X_{f}), and its image is known as feasible objective space ($Z_{\mathrm{f}} = \boldsymbol{f}(X_{\mathrm{f}})$). Therefore, the goal of a MOP is to find the set of feasible vectors $\mathbf{x}^* \in X_{\mathrm{f}}$ that optimize the function vector $\boldsymbol{f}(\mathbf{x})$. A fundamental concept in a MOP is the *Pareto Dominance*: if we assume a minimization problem (without loss of generality) the decision vector $\mathbf{x}^1$ is said to dominate the decision vector $\mathbf{x}^2$ (denoted by $\mathbf{x}^1 \prec \mathbf{x}^2$) if and only if $\forall i \in I = \{1, 2, \ldots, k\}$, $\mathrm{z}_i^1 \leq \mathrm{z}_i^2 \wedge \exists \mathrm{i} \in I : \mathrm{z}_i^1 < \mathrm{z}_i^2$. Thus, the set P^* of decision vectors that satisfy $\forall \mathbf{x}^* \in P^*$, $\nexists\, \mathbf{x} \in X_{\mathrm{f}} : \mathbf{x} \prec \mathbf{x}^*$ is defined as *Pareto Optimal Set* and its image is known as *Pareto Front* ($PF^* = \{\mathbf{z} = \boldsymbol{f}(\mathbf{x}^*) : \mathbf{x}^* \in P^*\}$). And hence, obtaining the best possible approximation set to the PF^* is the main goal of a multi-objective optimization algorithm, i.e. it should identify a set of solutions evenly distributed and as close as possible to the surface of the PF^* [25]. For notation, this set of solutions is denoted as P_{app}. However, the *Pareto Optimal Set* is unknown in many real-world engineering problems (as the problem addressed in this manuscript). In such cases, it is widely accepted the use of multi-objective indicators that do not need this information. These indicators can estimate the quality of a P_{app} [26, 27].

In this paper, we use two of the most popular indicators: the Hypervolume ($I_H(A)$) and the Set Coverage ($C(A,B)$). In the following, we assume the study of a MOP with two objectives, as the LAPP. The Hypervolume indicator is used to estimate the quality of the P_{app} obtained. This indicator ($I_H(A)$) can be defined as the area of the objective space (Z) that is dominated by the approximation set A and is bounded by the reference points. These reference points are calculated by means of the maximum and minimum values of each objective function. Taking into account that the target of a multi-objective optimization algorithm consists in finding a diverse set of non-dominated solutions, the approximation set $A \subseteq X_{\mathrm{f}}$ will be better than the approximation set $B \subseteq X_{\mathrm{f}}$ if $I_H(A) > I_H(B)$. Figure 1 shows an example of the Hypervolume calculation. And the Set Coverage indicator ($C(A,B)$) is used to know the proportion of solutions of the set $B \subseteq X_{\mathrm{f}}$ that are weakly

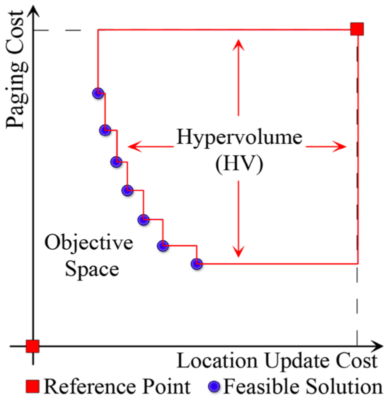


Fig. 1 Hypervolume for a minimization problem with two objectives

dominated by the solutions of the set $A \subseteq X_f$. Note that the solution or decision vector $\mathbf{x}^1$ is said to weakly dominate the decision vector $\mathbf{x}^2$ (denoted by $\mathbf{x}^1 \preccurlyeq \mathbf{x}^2$) when $\forall i \in I = \{1, 2, \ldots, k\}$, $z_i^1 \leq z_i^2$ [27].

Evolutionary Algorithms (EAs) are very suitable to solve multi-objective optimization problems, because they obtain a set of solutions simultaneously. An EA is a population-based metaheuristic optimization algorithm that uses the typical evolutionary operators (EVOPs) of biological systems (mutation, crossover, and natural selection) in order to improve a set of solutions by means of an iterative method [28]. Every individual of the population is an encoded solution of the problem ($\mathbf{x}^i$), and stores the individual's biological genotype. This genotype encodes the phenotype ($\mathbf{z}^i = f(\mathbf{x}^i)$) and is constituted by one or more chromosomes, where each chromosome is composed of several genes. The gene value ($x_j^i, \forall j \in \{1, 2, \ldots, n\}$) is known as allele and takes on values defined in a particular genetic alphabet [26], see Fig. 3. Mutation and crossover are two operations that modify the individual's genotype, and hence, they are defined in the decision space (X). On the other hand, the individual's survival depends on the individual's ability to adapt to the environment (i.e. the natural selection is dependent on the phenotype), so the natural selection is defined in the objective space (Z).

A Multi-Objective Evolutionary Algorithm (or MOEA) requires k + 1 functions: the fitness function and k objective functions. The k objective functions are those that the MOEA must to optimize: $\boldsymbol{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), \ldots, f_k(\mathbf{x})\}$. And the fitness function is the one which assigns a real-value to every objective vector $\mathbf{z}^i$. This value is proportional to the solution quality in the multi-objective context and allows discriminating among solutions.

As mentioned before, a MOEA finds the best possible P_{app} by means of an iterative method. In the first step, the individual initialization is carried out to obtain the initial population of parents. For it, a decision vector (and hence, an objective vector) is calculated for every individual. Then, these objective vectors are used to evaluate each individual by using the fitness function. The next step consists in finding new solutions by using the crossover (or recombination of parents) and the mutation operations. With the crossover operation, an offspring population is generated from the parent population, where each offspring has genetic information of its parents. For detailed information about the crossover operation, please see Sect. 3.4. On the other hand, the mutation operation consists in changing the value of one or more genes. Commonly, a chromosomal repair function is applied after the crossover and mutation operations, since the new solutions might be outside the feasible decision space (X_f), and consequently, outside the feasible objective space (Z_f). Finally, the offspring is evaluated and the best individuals found so far are selected as the new parent population. Figure 2 shows the task decomposition of a MOEA.

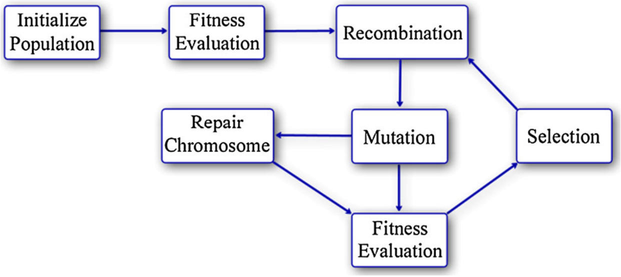


Fig. 2 Main tasks of a MOEA

In this paper we present our versions of two well-known MOEAs to solve the LAPP: the Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) and the Strength Pareto Evolutionary Algorithm 2 (SPEA2). We have chosen these algorithms because they are considered as standards in the multi-objective evolutionary optimization field. These two algorithms have been adapted to solve the LAPP (i.e. we use our evolutionary operators specific to solve the LAPP) and modified with the inclusion of a transformation function. This function allows us to redirect the search of the algorithms and improve the results of our previous works [15, 16]. These evolutionary operators and transformation function will be explained in the following subsections (from 3.3 to 3.6).

In order to obtain a fair comparison, both algorithms use the same EVOPs, with the exception of the selection procedure because each MOEA has its own fitness function. Sections 3.1 and 3.2 show a detailed explanation of our modified versions of NSGA-II and SPEA2 respectively. Section 3.3 presents the individual representation and the initialization operator. The crossover and mutation operators are discussed in Sects. 3.4 and 3.5, respectively. In addition to the evolutionary operators of any MOEA, we have implemented a transformation function which allows us to achieve a wide spread of P_{app}, and hence a higher I_{H} value. This function is described in Sect. 3.6. Note that, in this paper, all random variables are associated with a uniform probability density.

3.1 Non-Dominated Sorting Genetic Algorithm-II

The Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) is proposed by Deb et al. in [29] as an improved version of the original NSGA, defined by Srinivas and Deb in [30]. The main contributions of NSGA-II with respect to its predecessor are: less computational complexity and the use of elitism. The authors of NSGA-II have not explicitly defined a fitness function in their paper. However they propose a two step method for classifying a set of objective vectors. Firstly, the Pareto Dominance concept is used to arrange these objective vectors in groups (or fronts, function $\boldsymbol{R}(\mathbf{z}^{\mathbf{i}})$ in Eq. 8) in a way that the Pareto Dominance cannot be established among any pair of vectors of the same front. This procedure is commonly known as non-dominated sorting. And then, the density estimation is performed to measure the density of solutions surrounding a particular point in the feasible objective space (Z_{f}). The density value associated with each objective vector (also called crowding distance, function $\boldsymbol{C}(\mathbf{z}^{\mathbf{i}})$ in Eq. 8) is the sum of distances between the processed solution and its closest solution in each objective. An important criterion that has to be considered in the calculation of the crowding distance is the fact that solutions with the highest or lowest value of each objective are prioritized with the aim of favoring the spread of P_{app}. For detailed information of these procedures, please see [29]. Considering the definitions of the non-dominated sorting and the crowding distance, we propose the following fitness function for normalized values of the objective vectors:

$$f^{NSGA-II}_{fitness}\left(\overline{\mathbf{z}^{i}}\right) = 10^{\left(\left\lfloor \log_{10}\left(2\times N_{Obj}\right)\right\rfloor+1\right)} \times \left(N_F - \boldsymbol{R}\left(\overline{\mathbf{z}^{i}}\right)\right) + \boldsymbol{C}\left(\overline{\mathbf{z}^{i}}\right), \tag{8}$$

where $\mathrm{N}_{\mathrm{Obj}}$ is the number of objective functions, N_{F} is the number of fronts provided by the non-dominated sorting procedure, $\boldsymbol{R}(\mathbf{z}^{\mathrm{i}})$ is a function that returns the identifier of the front associated with the objective vector $\mathbf{z}^{\mathrm{i}}$ (where the fronts are arranged from the best to the worst), and $\boldsymbol{C}(\mathbf{z}^{\mathrm{i}})$ is a function that returns the crowding distance of the objective vector $\mathbf{z}^{\mathrm{i}}$, where we have considered that the infinite value is equal to the upper bound to all the objective functions multiplied by the number of objectives. In this work, we use normalization (in the range [0,1]) to ensure that the two objective functions have the same influence in the fitness function. It should be noted that this fitness function must be maximized.

Algorithm 1 Pseudo-code of our modified version of NSGA-II

```
% Initialization of the population (parent population)
Indv ← InitializationProcedure (Npop);
% Evaluation of the population
Indv ← ObjectiveFunctionsEvaluation(Indv);
Indv ← FitnessFunctionEvaluation (Indv);
% Main loop
while (Stop condition ≠ TRUE) {
 % Crossover or recombination of parents
 Offsp ← ElitistCrossover (Indv, P_C);
 % Mutation of the offspring
 Offsp ← Mutation (Offsp, P_M);
 % Evaluation of the offspring
 Offsp ← ObjetiveFunctionsEvaluation(Offsp);
 % Transformation function
 Offsp ← TransformationFunction(Offsp, Npop/2);
 % Evaluation of all population
 [Indv, Offsp] ← FitnessFunctionEvaluation(Indv, Offsp);
 % Selection of the fittest individuals
 Indv ← NaturalSelection(Indv, Offsp);
}
```

The pseudo-code of our modified version of NSGA-II is shown in Algorithm 1, where Npop is the population size, P_{C} is the crossover probability, and P_{M} is the mutation probability. This pseudo-code defines an iterative algorithm that is executed until the stop condition is reached. In this work, we use the same stop condition as other authors: the maximum number of generations.

3.2 Strength Pareto Evolutionary Algorithm 2

The Strength Pareto Evolutionary Algorithm 2 (SPEA2) is proposed by Zitzler et al. in [31]. SPEA2 presents two main differences with respect to its predecessor SPEA, defined by Zitzler and Thiele in [32]. First, it incorporates the density estimation into the fitness function, technique that allows redirecting the search of the algorithm to areas of the feasible objective space with less number of solutions. And second, it uses an enhanced archive truncation method. The main difference between SPEA2 and NSGA-II is that SPEA2 defines an archive of configurable size in which the best solutions found so far are stored. Furthermore, these solutions are used to generate the offspring population by means of an elitist crossover.

Zitzler et al. define in [31] the fitness function used by SPEA2. This fitness function has two main terms: the density estimation and the raw fitness. The raw fitness of a solution $\mathbf{z}^{\mathrm{i}}$ can be defined as the sum of "strengths" of its dominating solutions ($\mathbf{z}^{j} \prec \mathbf{z}^{i}$), where the "strength" of a solution $\mathbf{z}^{\mathrm{j}}$ (function $\boldsymbol{S}(\mathbf{z}^{\mathrm{j}})$ in Eq. 9) is the number of

Algorithm 2 Pseudo-code of our modified version of SPEA2

```
% Initialization of the population
Indv ← InitializationProcedure (Npop);
% Evaluation of the population
Indv ← ObjectiveFunctionsEvaluation(Indv);
%Create the archive
Arch ← CreateEmptyArchive(Narch);
% Main loop
while (Stop condition ≠ TRUE) {
  % Evaluation of all population
  [Indv, Arch] ← FitnessFunctionEvaluation(Indv,Arch);
  % Copy the fittest individuals in the archive (Natural Selection)
  Arch ← Copy(Arch, Indv);
  % Crossover or recombination of the archived individuals
  Indv ← ElitistCrossover (Arch, P_C);
  % Mutation of the offspring
  Indv ← Mutation (Indv, P_M);
  % Evaluation of the offspring
  Indv ← ObjetiveFunctionsEvaluation(Indv);
  % Transformation function
  Indv ← TransformationFunction(Indv, Npop/2);}
```

solutions dominated by $\mathbf{z}^j$. And the density estimation of solutions around the solution $\mathbf{z}^i$ is calculated by means of the Euclidean distance between $\mathbf{z}^i$ and its k-th nearest objective vector ($d(\mathbf{z}^i,\mathbf{z}^k)$), where k is commonly configured as the square root of the sample size (equal to the sum of the archive size and the population size). Equation (9) shows the SPEA2's fitness function. In this equation, P_t and P_t^{arch} are the set of individuals stored in the population and in the archive at the time t. It should be noted that this fitness function must be minimized, and we have defined it for normalized values of $\mathbf{z} \in Z_f$ in order to avoid prioritization among objectives.

$$f^{SPEA2}_{fitness}\left(\overline{\mathbf{z}^i}\right) = \frac{1}{2+d\left(\overline{\mathbf{z}^i},\overline{\mathbf{z}^k}\right)} + \sum_{j\in P_t+P_t^{arch},\,\overline{\mathbf{z}^j}\prec\overline{\mathbf{z}^i}} S\left(\overline{\mathbf{z}^j}\right), \qquad (9)$$

As in Sect. 3.1, the pseudo-code of our modified version of SPEA2 defines an iterative algorithm with a stop condition, which will be the same as the one used in NSGA-II (the maximum number of generations). This pseudo-code is presented in Algorithm 2, where Narch is the archive size.

3.3 Individual representation and initialization

As mentioned before, each individual encodes a solution of the problem. In this work, every individual is represented by a vector in which we store the location area associated with each network cell. Figure 3 shows the individual representation used in this paper. In order to obtain the first population, every vector is filled with a random pattern of 0 and 1 s, and then this pattern is used to determine the LAs configuration. For it, every location area is composed by a continuous and non-overlapped group of cells with the same value of vector. This is shown in Fig. 4.

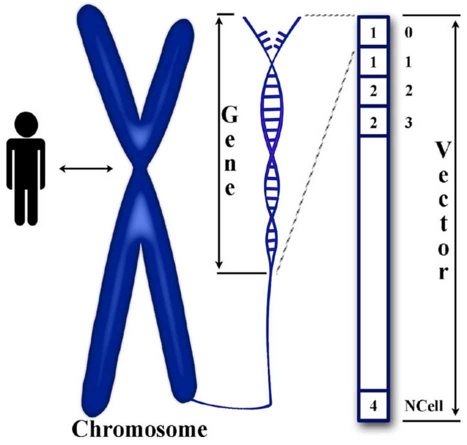


Fig. 3 Individual representation

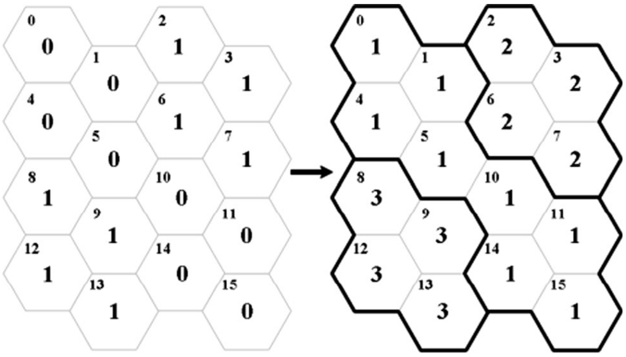


Fig. 4 Individual initialization. *Centre* number: *LA* identifier. *Upper left* number: Cell identifier

3.4 Crossover operation

The crossover operation is used by the EA to obtain a new set of solutions known as offspring population. This operation is performed according to the crossover probability (P_C) and it is mainly constituted by two procedures: selection of parents and recombination of selected parents. In this paper, we have implemented an elitist crossover, i.e. four individuals grouped in pairs are randomly selected, and then each parent is the best individual of each group. After that, the recombination procedure is performed to obtain the offspring. For it, each parent (or chromosome) is cut and recombined with pieces of the other parent. In this procedure, the number of crossover points and their positions are randomly determined in the range [1, 4] and [0,

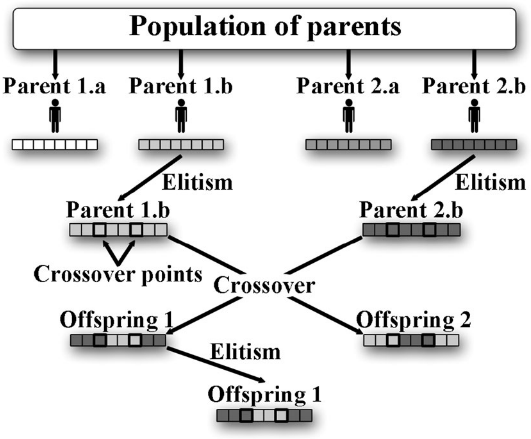


Fig. 5 Crossover operation

NCell-1], respectively. The crossover operation provides two offspring but we only store the best one in the offspring population. An example of the crossover operation is shown in Fig. 5.

3.5 Mutation operation

The previous crossover operation provides solutions with a high number of location areas. That is why we have defined two mutation operations which allow us to explore regions of Z_f with high PA_{cost}: Gene Mutation (GM) and Merge-LA Mutation (MLAM). The GM consists in changing the gene value of a boundary cell (a cell that is border among two or more location areas) by its neighboring location area with less number of cells, see Fig. 6. And the MLAM merges the smallest location area with its neighboring location area of lower size, see Fig. 7. Finally, in order to transform invalid solutions in feasible solutions, we apply a chromosomal repair function which deletes location areas constituted by separated groups of cells, see Fig. 4.

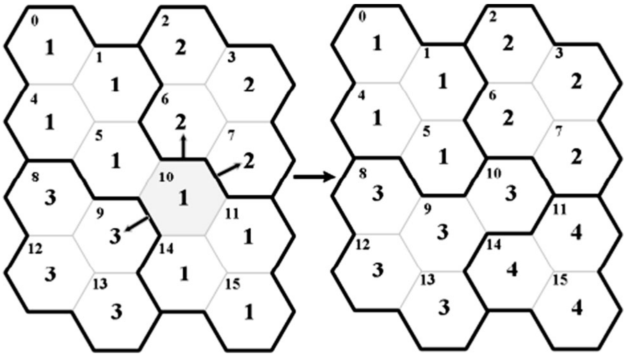


Fig. 6 Example of Gene Mutation. *Centre* number: *LA* identifier. *Upper left* number: Cell identifier

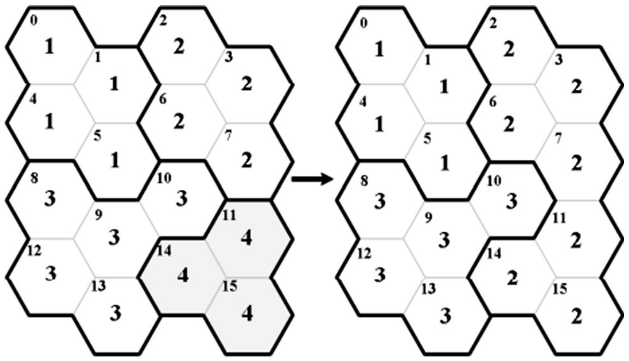


Fig. 7 Example of Merge-LA Mutation. *Centre* number: *LA* identifier. *Upper left* number: Cell identifier

3.6 Transformation function

In this work, we propose the use of a novel strategy that allows us to achieve a greater I_H value. We have implemented a function that transforms solutions with high LU_{cost} (configurations with a high number of small location areas) into solutions with high PA_{cost} (configurations with a small number of large location areas). For it, every location area which consists of only one network cell is merged with its neighboring location area of highest size. Therefore, this function transforms the always update strategy (where every network cell belongs to a different location area) into the never update strategy (where all the network cells belong to the same location area). This function is only applied to the first half of the population with the aim of exploring areas of Z_f with high LU_{cost} and PA_{cost} simultaneously.

4 Experimental results

With the purpose of determining the behavior of our proposals, we have studied five test networks of different complexity: LA25 (a test network with 5 × 5 cells) [10, 11], LA35 (a test network with 5 × 7 cells) [10, 11], LA49 (a test network with 7 × 7 cells) [10, 11], LA63 (a test network with 7 × 9 cells) [10, 11], and the BALI-2 network (a realistic mobile network of 90 cells and 66,550 subscribers) [34]. The test networks published in [10, 11] have got associated a mobile activity trace based on the subscriber's call and mobility patterns defined in [5]. These mobile activity traces are processed and compressed with the aim of deleting useless information. And the BALI-2 network is a mobile activity trace (proposed by the Stanford University [34]) that is well-validated against real-world data measured in the San Francisco Bay (USA). Therefore, if we assume that the user's call and mobility patterns are approximately known, the network operator could use our algorithms to find the configurations of Location Areas that best meet the real state of the signaling network considering the subscribers' behavior.

Furthermore, with the aim of knowing the quality of our solutions we have compared our results with those obtained by other authors, who developed other metaheuristics to solve the test networks proposed in [10, 11]: Hopfield Neural Network (HNN) [13], Simulated Annealing (SA) [11], Genetic Algorithm (GA) [10], combinations of GA with HNN (GA-HNN) [7], and Differential Evolution (DE) [14]. All of them optimize the LAPP by means of single-objective optimization strategies, where the objective function is Eq. (6) with β equal to 10. In contrast to a multi-objective approach, a single-objective optimization algorithm only provides a single solution, the one that best meets its objective function. Thus, to compare our multi-objective optimization algorithms with these single-objective optimization algorithms, we must search in each P_{app} the solution that minimizes Eq. (6).

On the other hand, we have calculated two indicators from the multi-objective optimization field that allow us to compare NSGA-II with SPEA2: the Hypervolume ($I_H(A)$) and the Set Coverage $C(A,B)$. Remember that these two indicators have been defined in Sect. 3. Note that NSGA-II and SPEA2 are stochastic algorithms, so in order to make decisions with a certain level of confidence, we have performed a statistical study of the results obtained [33]. Figure 8 shows the statistical study followed in this work. Firstly, we use the Shapiro–Wilk test to determine if the sample of an experiment follows a normal distribution. If this test is positive for every experiment, the Levene test is applied to check the homogeneity of the variances. Finally, when the Levene test is positive, the balanced one-way ANOVA analysis is performed to determine whether there is a significant difference among the means of the experiments. On the other hand, when the Shapiro–Wilk test or the Levene test are negatives, we apply the Wilcoxon rank sum test to check whether there is a significant difference among the medians of the experiments. All tests mentioned before are configured with a confidence level equal to 95 % (a significance level of 5 %), which means that the differences are unlikely to have occurred by chance with a probability of 95 %.

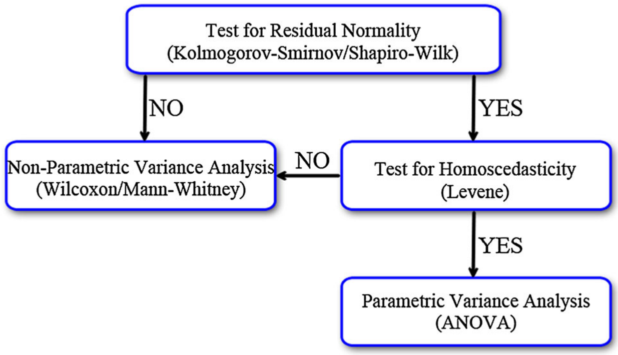

Test for Residual Normality
(Kolmogorov-Smirnov/Shapiro-Wilk)
NO
YES
Non-Parametric Variance Analysis
(Wilcoxon/Mann-Whitney)
NO
Test for Homoscedasticity
(Levene)
YES
Parametric Variance Analysis
(ANOVA)

Fig. 8 Statistical analysis methodology

A further novel contribution of this paper is the analysis of the LAs configuration for several values of β and the relation between a particular LAs configuration and the user's call and mobility patterns. Section 4.1 shows our results and comparisons with other authors. An analysis of the solutions obtained and its relation to the user's call and mobility patterns are discussed in Sect. 4.2. And Sect. 4.3 presents the study of a realistic mobile network.

4.1 Comparisons among algorithms

In this section, we present a comparison of our multi-objective optimization algorithms with the single-objective optimization algorithms proposed in [7, 10, 11, 13, 14]. For it, we have searched in the P_{app} of each multi-objective optimization algorithm (NSGA-II and SPEA2) the solution that minimizes Eq. (6) with β equal to 10. Moreover, we have performed a comparison between NSGA-II and SPEA2 by using the multi-objective indicators described in Sect. 3. These indicators allow us to compare the quality of the P_{app} obtained by each MOEA.

To perform a fair comparison, we use the same population size and the same stop condition (maximum number of generations) as in [7, 10, 11, 13, 14]. The other parameters of each MOEA have been tuned separately with the purpose of maximizing the Hypervolume value, because this is the indicator used in this manuscript to estimate the quality of the P_{app}s obtained. Furthermore, we have performed a statistical study in each experiment to ensure that the results obtained are statistically relevant.

Table 1 shows the parameter configuration of our MOEAs, where Npop is the population size, $\mathrm{P_C}$ is the crossover probability, $\mathrm{P_M}$ is the total mutation probability, and Narch is the archive size. The total mutation probability is configured such that $\mathrm{P_M} = 2\mathrm{P_{GM}} = 2\mathrm{P_{MLAM}}$, where $\mathrm{P_{GM}}$ is the Gene Mutation probability and $\mathrm{P_{MLAM}}$ is the Merge-LA Mutation probability. Table 2 presents the comparison of our multi-objective implementations with the single-objective optimization algorithms proposed in [7, 10, 11, 13, 14]. This table reveals that there is at least one solution

Table 1 Configuration of MOEAs

Algorithm	Parameter				
	Npop	Generations	$\mathrm{P_C}$	$\mathrm{P_M}$	Narch
NSGA-II	250	5,000	0.90	0.20	–
SPEA2	250	5,000	0.90	0.26	300

Table 2 Comparison with single-objective algorithms

Algorithm	Test network			
	LA25	LA35	LA49	LA63
NSGA-II	26,990	39,832	60,685	89,085
SPEA2	26,990	39,832	60,685	89,085
HNN[13]	27,249	39,832	63,516	92,493
SA[11]	26,990	42,750	60,694	90,506
GA[10]	28,299	40,085	61,938	90,318
GA-HNN1[7]	26,990	40,117	62,916	92,659
GA-HNN2[7]	26,990	39,832	62,253	91,916
GA-HNN3[7]	26,990	39,832	60,696	91,819
DE[14]	26,990	39,859	61,037	89,973

Eq. (6): $\text{Cost}_{\text{tot}}(10)$

Table 3 Multi-objective indicators

Multi-objective indicator	Test network			
	LA25	LA35	LA49	LA63
median $(I_H(P_{app}^{NSGA\text{-}II}))(\%)$	76.351	80.870	83.732	84.179
median $(I_H(P_{app}^{SPEA2}))(\%)$	76.309	80.813	83.608	84.055
iqr $(I_H(P_{app}^{NSGA\text{-}II}))(\%)$	0.006	0.020	0.037	0.046
iqr $(I_H(P_{app}^{SPEA2}))(\%)$	0.013	0.025	0.158	0.114
$C\left(\tilde{P}_{app}^{NSGA-II}, \tilde{P}_{app}^{SPEA2}\right)(\%)$	91.124	94.800	83.714	60.937
$C\left(\tilde{P}_{app}^{SPEA2}, \tilde{P}_{app}^{NSGA-II}\right)(\%)$	55.526	36.163	26.034	38.881

Table 4 Results from the statistical study

	Test network			
	LA25	LA35	LA49	LA63
p value	3.336e^{-11}	3.332e^{-11}	1.609e^{-10}	5.706e^{-9}

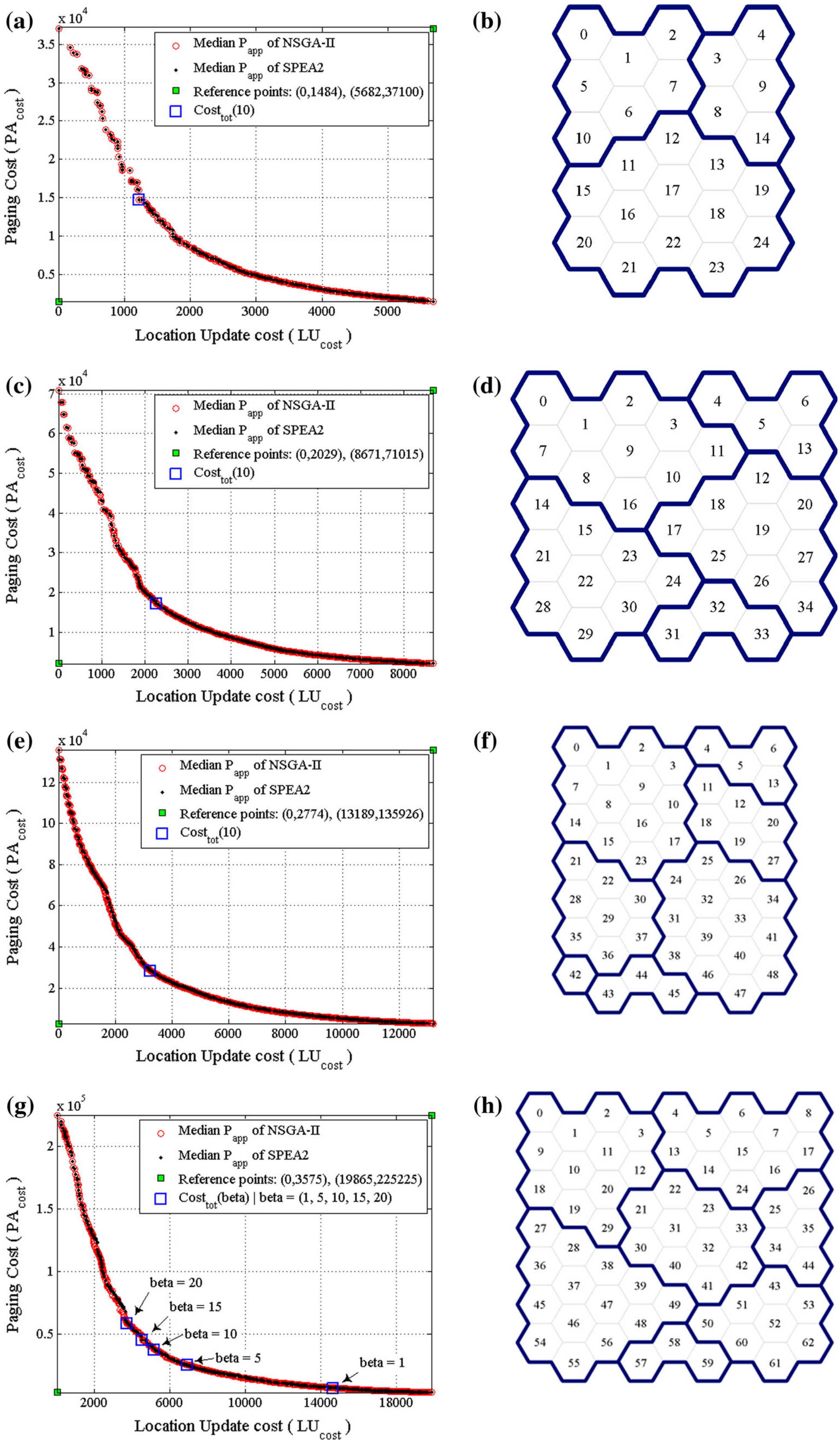
in each P_{app} that outperforms the results obtained by the single-objective approaches in the two most complex test networks: LA49 and LA63. Table 3 shows statistical data (median and interquartile range) of the Hypervolume obtained by our multi-objective optimization algorithms, where each experiment consists of 31 independent runs, as well as the Set Coverage ($C(A, B)$) of the median P_{app} of each MOEA. For notation, we define $\tilde{x}$ as the median value of the sample x. And in Table 4, we present a summary of the statistical study performed to compare the I_Hs of NSGA-II and SPEA2, which provides a coefficient known as *p* value (a *p* value lower than the significance level means that we have sufficient evidence to conclude that there are significant differences between the samples studied). These tables reflect that NSGA-II provides better and more stable results than SPEA2, since NSGA-II achieves a higher I_H value than SPEA2 in all test networks and with a lower interquartile range value. It should also be noted that, according to the Set Coverage indicator, SPEA2 is weakly dominated by NSGA-II in all test networks. Furthermore, although the I_H values of NSGA-II and SPEA2 are very similar at first glance, Table 4 clarifies that there is a significant difference between the I_Hs of NSGA-II and SPEA2 in all test networks, since the *p* value provided by the statistical study is lower than the desired significance level (0.05). That is due to the small value of the interquartile ranges. Finally, Fig. 9 shows a graphic representation of the median P_{app} obtained by NSGA-II and SPEA2 for each test network. This figure also shows the LAs configuration of our solutions in Table 2 and the position of each one in the corresponding P_{app}. In this figure, we can observe:

- A high spread of the P_{app}s obtained, which include the never-update and always-update strategies.
- A decrease of the P_{app} density when the size of the location areas increases (i.e. solutions with high PA_{cost}), because of there is a less number of feasible solutions.
- The paging cost increases faster than the location update cost.
- In Fig. 9(g), there is a huge number of LAs configurations between the solutions associated with $\text{Cost}_{\text{tot}}(1)$ and $\text{Cost}_{\text{tot}}(5)$. This indicates that β is a real number, which greatly complicates the configuration of this coefficient in single-objective optimization algorithms.

4.2 Analysis of solutions

In this section, we have performed an in-depth study of the LAs configuration for several values of β and its relation to the user's call and mobility patterns. This analysis is a novel contribution and will allow us to understand the behavior of the LAs scheme. Figure 11 shows the results of the study for the most complex test network (LA63). First, Fig. 10(a) presents the number of incoming and outgoing users per cell. Figure 10(b) shows the number of incoming calls per cell. From Fig. 11(a) to (j), the LAs configuration and the $\text{Cost}_{\text{tot}}(\beta)$ per cell for β equal to 1, 5, 10, 15, and 20 are presented. These figures provide us important information. Firstly, it should be noted that the LAs configuration has a huge number of location areas when the LU_{cost} and the PA_{cost} are equally treated ($\beta = 1$), although the number of incoming and outgoing users per cell is higher than the incoming calls per cell. This is because the PA_{cost} increases faster than the LU_{cost}, as was demonstrated in Sect. 4.1. That is why the $\text{Cost}_{\text{tot}}(1)$ per cell is very similar to the user's mobility pattern, Figs. 10(a),

Fig. 9 Median P_{app} of NSGA-II and SPEA2 and the LAs configuration of solutions in Table 2 for: **a**, **b** LA25, **c**, **d** LA35, **e**, **f** LA49, **g**, **h** LA63



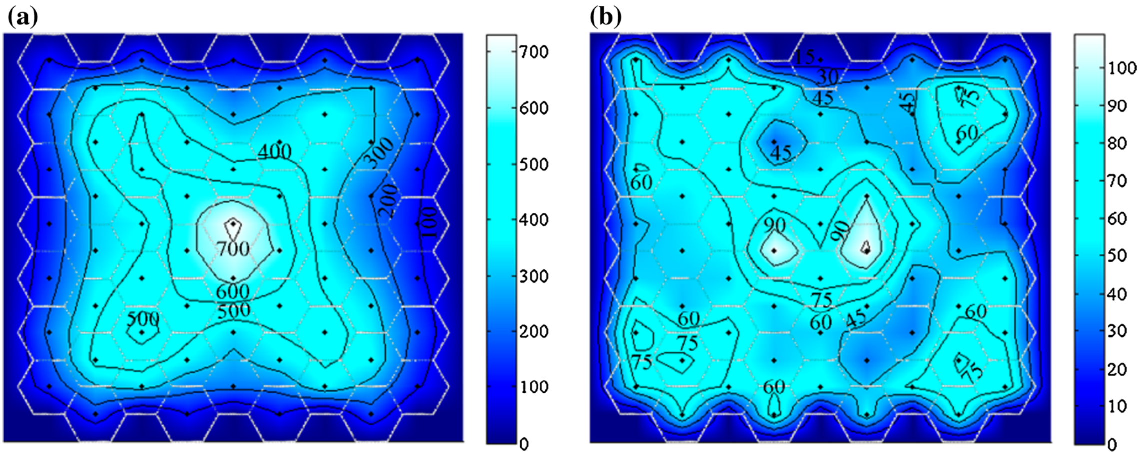


Fig. 10 **a** Number of incoming and outgoing users and **b** incoming calls per cell for LA63

11(b) respectively. Therefore, it is concluded firstly that cells with higher exchange of users are grouped in location areas as small as possible with the aim of reducing the LU_{cost} without greatly increasing the PA_{cost}. And secondly, an increment of β implies an increase of the LAs size in order to minimize the LU_{cost}. So doing, the number of boundary cells is reduced and cells with higher number of incoming and outgoing users are located in the centre of its location area, see cells 31 and 32 of Fig. 11(h). Furthermore, the maximum values of $Cost_{tot}(\beta)$ are located in boundary cells, as expected because LU_{cost} has a greater contribution in $Cost_{tot}(\beta)$ than PA_{cost}.

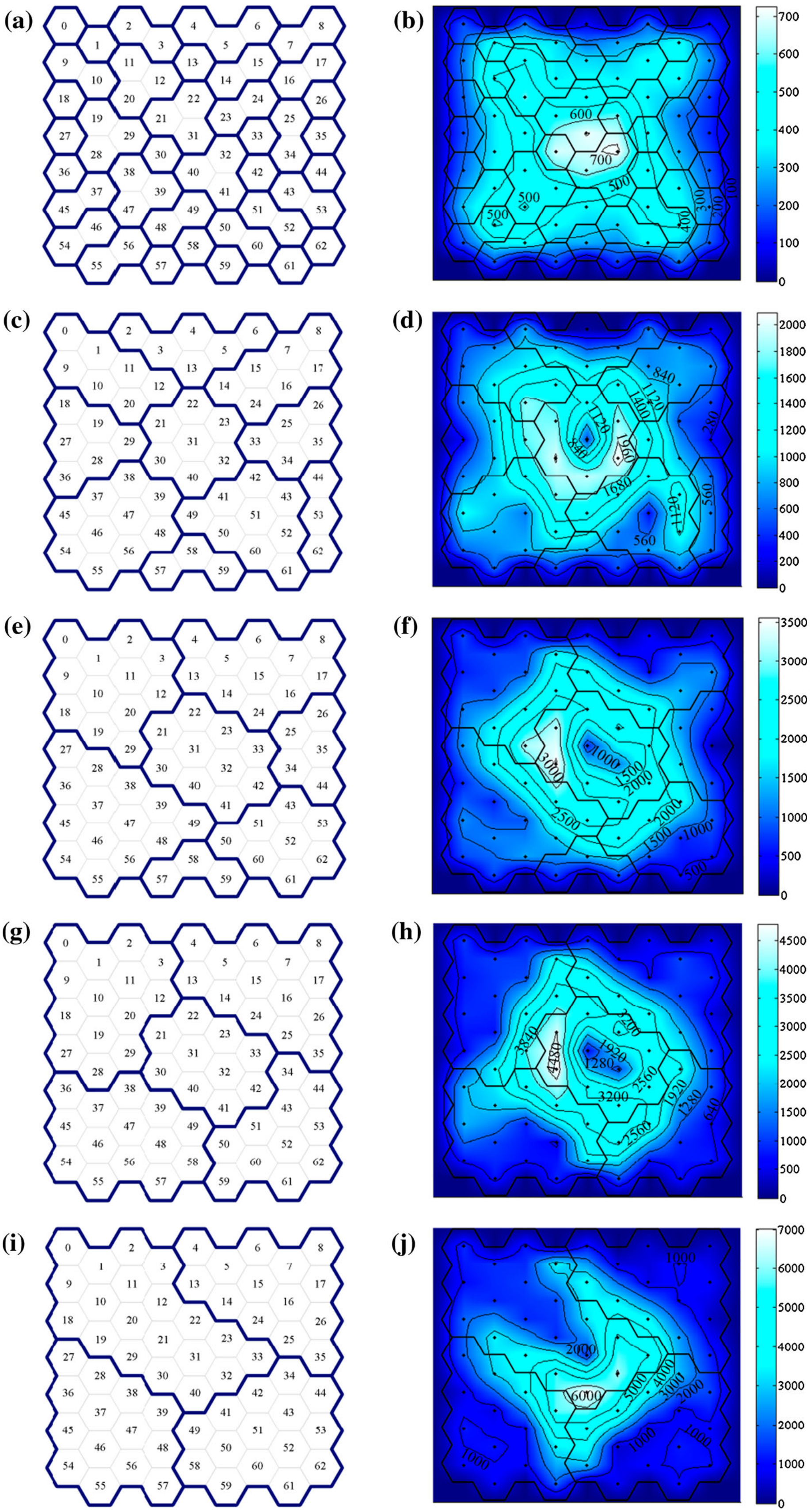
4.3 Study of a realistic mobile network

Once we have determined the goodness of the multi-objective optimization and that our proposals outperform other optimization techniques published in the recent literature, we optimize the Location Areas Planning Problem in a realistic mobile network [34]. This mobile network is called BALI-2 and consists of 90 cells and 66,550 subscribers. We have chosen this network because it is well-validated against real-world data measured in the San Francisco Bay (USA). Figure 12 shows a graphical representation of the BALI-2 network. This study will give us information about the behavior of our proposals when they optimize the LAPP in a realistic mobile environment. Figure 13 shows the median P_{app} obtained by our proposals in the BALI-2 network. And Table 5 gathers the results of this experimental study, where we have performed 31 independent runs per experiment. In this table, we present statistical data (median and interquartile range) of the Hypervolume, and also the Set Coverage ($C(A, B)$) of the median P_{app}s. The reference points used in Fig. 13 are: (0, 1570807) and (243951, 141372630). This figure reveals that our proposals also obtain good P_{app} in a realistic mobile network (and hence, in a more complex problem instance). Furthermore, if we analyze the results in Table 5 we can obtain several conclusions. Firstly, the Hypervolume indicator establishes that our version of NSGA-II obtains better and more stable results than our version of SPEA2. In this experimental study, the p value is equal to $1.402e^{-11}$, so we can conclude that the differences between our proposals are statistically significant. On the other hand, the Set Coverage indicator determines that our version of SPEA2 performs better than our version of NSGA-II. This is because SPEA2 obtains better results than NSGA-II in those regions of the objective space with higher location update cost. In such cases, a graphical representation could be of great assistance (see Fig. 13). Figure 13 clarifies that our proposals obtain very similar results in regions of the objective space with higher location update cost, and that NSGA-II performs better than SPEA2 in regions of the objective space with higher paging cost. Therefore, we could conclude that our version of NSGA-II in better than our version of SPEA2. The paging procedure used in this experimental study is the blanket polling paging (see Sect. 2).

5 Conclusions and future work

In this paper, we research one of the most important tasks in the Public Land Mobile Networks, the location management. There are several strategies to manage this task but we focus on the LAs scheme, because it is widely used in current terrestrial mobile networks. In this scheme, the location management task is defined as a multi-objective optimization problem with two costs that must be minimized: location update (LU_{cost}) and paging (PA_{cost}). Previously, the LAs scheme was solved by using single-objective optimization techniques, where the LU_{cost} and

Fig. 11 LAs configuration and $\mathrm{Cost_{tot}}(\beta)$ for LA63: **a**, **b** $\beta = 1$, **c**, **d** $\beta = 5$, **e**, **f** $\beta = 10$, **g**, **h** $\beta = 15$, **i**, **j** $\beta = 20$



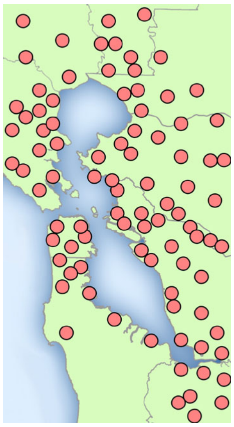


Fig. 12 Graphical representation of the BALI-2 network

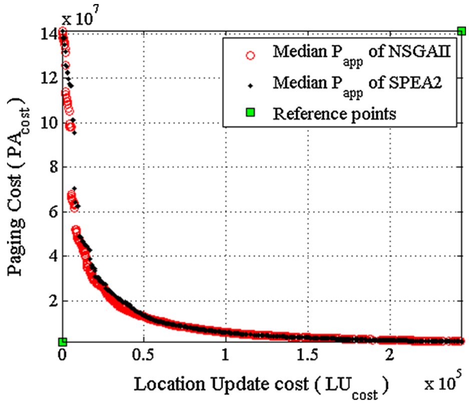


Fig. 13 Graphical representation of the median P_{app}s

Table 5 Multi-objective indicators

Multi-objective indicator	Algorithm	
	NSGA-II	SPEA2
$median(I_H(P_{app}))(\%)$	93.095	92.250
$iqr(I_H(P_{app}))(\%)$	0.001	0.004
$C\left(\tilde{P}_{app}^{NSGA-II}, \tilde{P}_{app}^{x}\right)(\%)$	–	29.926
$C\left(\tilde{P}_{app}^{SPEA2}, \tilde{P}_{app}^{x}\right)(\%)$	40.376	–

PA$_{\text{cost}}$ were combined in a weighted objective function, see Eq. (6). However, several drawbacks arise when the LA Planning Problem is tackled with single-objective approaches. Firstly, an accurate knowledge of the problem is required to properly configure the value of the weight coefficient (β), which is a real value (as was demonstrated in Sect. 4.1). Secondly, a temporal variation of the network state should be considered when setting this coefficient. And thirdly, a single-objective optimization algorithm must perform an independent run for each value of β. That is why we have proposed the use of multi-objective optimization algorithms to solve the LAPP. In contrast to a single-objective approach, a multi-objective optimization algorithm provides a set of solutions among which the network operator could select the one that best adjusts to the real state of the signaling network. Moreover, these solutions are obtained simultaneously and each one is associated with a value of the weight coefficient (β).

In this work, we propose our versions of two multi-objective optimization algorithms that could be considered as standards of the multi-objective evolutionary optimization field: Non-Dominated Sorting Genetic Algorithm-II (NSGA-II) and Strength Pareto Evolutionary Algorithm 2 (SPEA2). These two algorithms have been adapted (i.e. we use our evolutionary operators specific to solve the Location Areas Planning Problem) and modified with the inclusion of a transformation function. With the aim of checking the quality of our algorithms, we have solved four test networks of different complexity and we have compared our results with those obtained by other authors, who developed single-objective optimization algorithms to optimize the same instances (there are no other previous proposals that tackle the LAPP with a multi-objective approach). For it, we have searched in the set of solutions the one that minimizes the objective function used by these single-objective optimization algorithms. Results show that our multi-objective implementations outperform the results obtained by other authors in the two most complex test networks: LA49 and LA63.

On the other hand, we have calculated indicators of the multi-objective optimization field to compare our modified versions of NSGA-II and SPEA2. A statistical study has been performed to determine whether the differences between these multi-objective evolutionary algorithms are significant with a certain level of confidence. This study establishes that NSGA-II is slightly superior to SPEA2 in all test networks. Furthermore, we have analyzed the configurations of LAs for several values of β. This analysis is a novel contribution of this paper and clarifies that the cells with higher mobile activity are grouped in location areas as small as possible with the goal of reducing the LU$_{\text{cost}}$ without greatly increasing the PA$_{\text{cost}}$. Moreover, it should be noted that these cells are located in the centre of their location area. Finally, we have demonstrated that our proposals also obtain good results in a realistic mobile network. In future work, it would be interesting to implement other multi-objective optimization algorithms to solve these instances and compare them with our versions of NSGA-II and SPEA2. Furthermore, it would be a good challenge the study of other location management strategies. In particular, different mobile location management strategies based on mobility patterns and movement prediction could be implemented and evaluated in a multi-objective manner.

Acknowledgments This work has been partially funded by the Spanish Ministry of Economy and Competitiveness and the ERDF (European Regional Development Fund), under the Contract TIN2012-30685 (BIO project). Víctor Berrocal-Plaza is supported by the research Grant FPU-AP2010-5841 from the Spanish Goverment (Ministerio de Educación).

References

1. Lin, Y.-B., & Chlamtac, I. (2008). *Wireless and mobile network architectures*. New York: John Wiley India Pvt Limited.
2. Agrawal, D. P., & Zeng, Q. A. (2010). *Introduction to wireless and mobile systems* (3rd ed.). Stamford: Cengage Learning.
3. Garg, V. (2010). *Wireless communications & networking. The Morgan Kaufmann series in networking*. San Francisco: Elsevier Science.
4. Lescuyer, P., Lucidarme, T. (2008). *Evolved packet system (EPS): The LTE and SAE evolution of 3G UMTS*. New York: Wiley Publishing.
5. Kyamakya, K., & Jobmann, K. (2005). Location management in cellular networks: classification of the most important paradigms, realistic simulation framework, and relative performance analysis. *IEEE Transactions on Vehicular Technology, 54*, 687–708.
6. Nowoswiat, D., & Milliken, G. (2013). *Managing LTE core network signaling traffic*. Techzine: Alcatel-Lucent.
7. Taheri, J., & Zomaya, A. (2007). A combined genetic-neural algorithm for mobility management. *Journal of Mathematical Modeling and Algorithms, 6*, 481–507.
8. Gondim, P. R. L. (1996). Genetic algorithms and the location area partitioning problem in cellular networks. In *Proceedings of IEEE 46th vehicular technology conference, 'mobile technology for the human race'*, pp. 1835–1838.
9. Demestichas, P., Georgantas, N., Tzifa, E., Demesticha, V., Striki, M., Kilanioti, M., et al. (2000). Computationally efficient algorithms for location area planning in future cellular systems. *Computer Communications, 23*, 1263–1280.
10. Taheri, J., & Zomaya, Y. A. (2005). A genetic algorithm for finding optimal location area configurations for mobility management. In *Proceedings of the IEEE conference on local computer networks 30th anniversary*, pp. 568–577.
11. Taheri, J., & Zomaya, Y. A. (2005). A simulated annealing approach for mobile location management. In *Proceedings of the 19th IEEE international parallel and distributed processing symposium*, pp. 194–201.
12. Taheri, J., & Zomaya, Y. A. (2006). Clustering techniques for dynamic mobility management. In *Proceedings of the 4th ACM international workshop on mobility management and wireless access*, pp. 10–17.
13. Taheri, J., & Zomaya, Y. A. (2004). The use of a Hopfield neural network in solving the mobility management problem. In *Proceedings of the IEEE/ACS international conference of pervasive services*, pp. 141–150.
14. Almeida-Luz, S. M., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., & Sánchez-Pérez, J. M. (2011). Differential evolution for solving the mobile location management. *Applied Soft Computing, 11*, 410–427.
15. Berrocal-Plaza, V., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., & Sánchez-Pérez, J. M. (2012). A multi-objective approach to solve the location areas problem. In *1st International conference on the theory and practice of natural computing*, pp. 72–83.
16. Berrocal-Plaza, V., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., & Sánchez-Pérez, J. M. (2012). Solving the location areas problem with strength pareto evolutionary algorithm. In *Proceedings of the 13th IEEE international symposium on computational intelligence and informatics*, pp. 49–54.
17. Berrocal-Plaza, V., Vega-Rodríguez, M. A., Gómez-Pulido, J. A., & Sánchez-Pérez, J. M. (2013). Solving the location areas scheme in realistic networks by using a multi-objective algorithm. In *Proceedings of the 16th European conference on applications of evolutionary computation*, pp. 72–81.
18. Hu, L.-R., & Rappaport, S. S. (1997). An adaptive location management scheme for global personal communications. In *Proceedings of IEEE international conference on personal communications*, pp. 54–60.
19. Taheri, J., Zomaya, A. Y., & Iftikhar, M. (2011). Fuzzy online location management in mobile computing environments. *Journal of Parallel and Distributed Computing, 71*(8), 1142–1153.
20. Subrata, R., & Zomaya, A. Y. (2003). Evolving cellular automata for location management in mobile computing networks. *IEEE Transactions on Parallel and Distributed Systems, 14*(1), 13–26.
21. Zhao, W., & Xie, J. (2011). ReLoAD: resilient location area design for internet-based infrastructure wireless mesh networks. GLOBECOM, pp. 1–5.
22. Zhao, W., & Xie, J. (2013). DoMaIN: A novel dynamic location management solution for internet-based infrastructure wireless mesh networks. *IEEE Transactions on Parallel and Distributed Systems, 24*(8), 1514–1524.
23. Zitzler, E. (1999). *Evolutionary algorithms for multiobjective optimization: Methods and applications*. PhD thesis, ETH Zurich.
24. Knowles, J., Thiele, L., & Zitzler, E. (2006). *A tutorial on the performance assessment of stochastic multiobjective optimizers. Computer engineering and networks laboratory (TIK)*, TIK Report 214, ETH Zurich.
25. Deb, K. (2001). *MultiObjective optimization using evolutionary algorithms* (1st ed.). Chichester: Wiley.
26. Coello, C. A., Lamont, G. L., & Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems* (2nd ed.). Berlin: Springer.
27. Janssens, G. K., & Pangilinan, J. M. (2010). *Multiple criteria performance analysis of non-dominated sets obtained by multi-objective evolutionary algorithms for optimization*. AIAI, pp. 94–103.
28. Weise, T. (2009). *Global optimization algorithms: Theory and application [Online]*. http://www.it-weise.de/.
29. Deb, K., Agrawal, S., Pratap, A., & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation, 6*, 182–197.
30. Srinivas, N., & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation, 2*, 221–248.
31. Zitzler, E., Laumanns, M., & Thiele, L. (2002). *SPEA2: Improving the strength pareto evolutionary algorithm for multi-objective optimization. Evolutionary methods for design, optimisation and control with application to industrial problems*, pp. 95–100.
32. Zitzler, E., & Thiele, L. (1998). *An evolutionary algorithm for multiobjective optimization: The strength pareto approach. Computer engineering and communication networks lab (TIK), Tech. Rep. 43*, ETH Zürich.
33. Sheskin, D. J. (2011). *Handbook of parametric and nonparametric statistical procedures* (5th ed.). New York: Chapman & Hall/CRC Press.
34. Stanford University Mobile Activity TRAces (SUMATRA). Accessed in 2014 http://infolab.stanford.edu/sumatra.

Víctor Berrocal-Plaza received the Telecommunication Engineer degree in 2010 from the University of Sevilla (Spain), and received his MS degree in 2011 from the University of Extremadura (Spain). He is currently pursuing the Ph.D. degree with the University of Extremadura under the direction of Dr. Vega-Rodríguez. His main research interests include evolutionary and bioinspired algorithms, and their applications to optimization problems in cellular networks.

Miguel A. Vega-Rodríguez received his Ph.D. degree in Computer Engineering from the University of Extremadura, Spain, in 2003. He is currently an Associate Professor (Accredited as Full Professor) of computer architecture in the Department of Computer and Communications Technologies, University of Extremadura. He has authored or co-authored more than 530 publications including journal papers (more than 75 JCR impact factor journal papers), book chapters, and peer-reviewed conference proceedings. His main research interests include parallel and distributed computing, evolutionary computation, and reconfigurable computing. Dr. Vega-Rodríguez is an editor and reviewer of diverse international JCR impact factor journals. Furthermore, he has been guest editor in 9 special issues in different international JCR impact factor journals.

Juan M. Sánchez-Pérez received his Ph.D. degree in Physics from the Complutense University of Madrid, Spain, in 1976. He is Full Professor of computer architecture in the Department of Computer and Communications Technologies, University of Extremadura, Spain. He has authored or co-authored more than 410 publications including journal papers (more than 45 JCR impact factor journal papers), book chapters, and peer-reviewed conference proceedings. His main research interests include artificial intelligence, logic design, and modern computer architectures.