A New Version of the Multiobjective Artificial Bee Colony Algorithm for Optimizing the Location Areas Planning in a Realistic Network

Víctor Berrocal-Plaza*, Miguel A. Vega-Rodríguez, and Juan M. Sánchez-Pérez

Dept. of Computers & Communications Technologies, University of Extremadura
Escuela Politécnica, Campus Universitario S/N, 10003, Cáceres, Spain
{vicberpla,mavega,sanperez}@unex.es

Abstract. In this paper, we present our version of the MultiObjective Artificial Bee Colony algorithm (a metaheuristic based on the foraging behavior of honey bees) to optimize the Location Areas Planning Problem. This bi-objective problem models one of the most important tasks in any Public Land Mobile Network: the mobile location management. In previous works of other authors, this management problem was simplified by using the linear aggregation of the objective functions. However, this technique has several drawbacks. That is the reason why we propose the use of multiobjective optimization. Furthermore, with the aim of studying a realistic mobile environment, we apply our algorithm to the mobile network developed by the Stanford University (a mobile network located in the San Francisco Bay, USA). Experimental results show that our proposal outperforms other algorithms published in the literature.

Keywords: Location Areas Planning Problem, Mobile Location Management, Multiobjective Optimization, Stanford University Mobile Activity Traces, Artificial Bee Colony Algorithm.

1 Introduction

The Public Land Mobile Networks (PLMNs) are the networks that provide mobile communications to the public. For it, the desired coverage area is divided into several smaller regions (known as network cells) with the aim of providing service to a huge number of mobile subscribers with few radioelectric resources [1]. In such networks, the mobile location management is a fundamental task whereby the mobile network tracks the subscribers' movement with the goal of redirecting the incoming calls to the callee users.

Commonly, the mobile location management consists of two main procedures: the subscriber location update (LU), and the paging (PA) [9]. The LU is initiated by the mobile station (MS, the subscriber terminal) according to a predefined method. There are several LU procedures published in the literature: never

* The work of Víctor Berrocal-Plaza has been developed under the Grant FPU-AP2010-5841 from the Spanish Government. This work was partially funded by the Spanish Ministry of Economy and Competitiveness and the ERDF (European Regional Development Fund), under the contract TIN2012-30685 (BIO project).

A.-H. Dediu et al. (Eds.): TPNC 2013, LNCS 8273, pp. 19–30, 2013.

update, whenever the MS moves to a new cell (or always update), whenever the MS moves to a new area (location areas), every certain number of visited cells (distance-based), every certain number of cell crossing (movement-based), etc. [16]. However, the location update based on location areas highlights as one of the most popular strategies in current mobile networks, and therefore, this is the *LU* procedure studied in this work. On the other hand, the *PA* procedure is initiated by the mobile network to know the exact cell in which the callee subscribers are located [13]. For it, several paging messages are broadcast around the last known location. This last procedure can be classified into two main groups: simultaneous paging and sequential paging. In the simultaneous paging, the network cells are simultaneously polled. And in the sequential paging, the network cells are grouped into paging areas that are sequentially polled. Due to the time constraint associated with this last procedure, the maximum number of paging cycles (and hence, the maximum number of paging areas) is restricted to be lower or equal to three [13].

In this work, we research the mobile location management based on location areas with the two-cycle sequential paging defined in [15]. This *LU* strategy groups the network cells into continuous and non-overlapped logical areas (or Location Areas, LAs [16]) with the aim of delimiting the signaling load associated with the *LU* and *PA* procedures. In this way, the mobile station only updates its location whenever it moves to a new Location Area (LA), and the paging procedure is only performed within the last updated LA. Therefore, the proper dimensioning of this location management strategy is an important engineering issue, in which the main goal is to find the configurations of LAs that simultaneously reduce the number of location updates and the number of paging messages. That is, the Location Areas Planning Problem (LAPP) is a multiobjective optimization problem with two objective functions (as we will see in Section 3).

With the aim of finding the best possible configurations of location areas, we propose a new version of the MultiObjective Artificial Bee Colony algorithm (MO-ABC, a swarm intelligence algorithm based on the foraging behavior of honey bees [14]). Furthermore, in order to study a realistic mobile environment, we solve the mobile network developed by the Stanford University [11], because it is well-validated against real data measured in the San Francisco Bay (USA).

The rest of the paper is organized as follows. Section 2 presents the related works. The Location Areas Planning Problem is defined in Section 3. The main features of a multiobjective optimization problem, the quality indicators used in this work, and a detailed explanation of our proposal are shown in Section 4. Experimental results and comparisons with other works are presented in Section 5. Finally, Section 6 discusses our conclusions and future work.

2 Related Work

The mobile location management based on location areas has been widely researched in the last decade due to the exponential increase in the number of mobile subscribers. There are several works in the literature in which different metaheuristics have been applied to optimize the Location Areas Planning Problem. P. R. L. Gondim in [10] was one of the first authors to define the LAPP as an NP-hard combinatorial optimization problem due to the huge size of the objective space. In his work, he proposed a Genetic Algorithm (GA) to find quasi-optimal configurations of Location Areas. P. Demestichas et al. in [8] proposed three metaheuristics (Simulated Annealing (SA), Tabu Search (TS), and GA) to study the LAPP in different mobile environments. R. Subrata and A. Y. Zomaya proposed a dynamic location update procedure based on location areas [15], which allows reducing the number of location updates, but not the total signaling load because a personalized configuration of location areas should be calculated, stored, and transmitted whenever a mobile station moves out of its current location area. In this work, R. Subrata and A. Y. Zomaya studied the real-time mobile activity trace developed by the Stanford University [11], a mobile activity trace that is well-validated against real data measured in the San Francisco Bay (USA). This mobile activity trace was also studied in [2,3], where S. M. Almeida-Luz et al. optimized the LAPP with the algorithms: Differential Evolution (DE, [2]) and Scatter Search (SS, [3]).

In all of these related works, the LAPP was solved by using different algorithms from the single-objective optimization field. For it, the two objective functions of the LAPP were linearly combined into a single objective function. This technique allows simplifying the optimization problem, but has several drawbacks (see Section 3). That is why our research focuses on optimizing this problem with different multiobjective metaheuristics. Recently, we have just published our versions of two well-known MultiObjective Evolutionary Algorithms (MOEAs): the Non-dominated Sorting Genetic Algorithm II (NSGAII [5]), and the Strength Pareto Evolutionary Algorithm 2 (SPEA2 [4]).

3 Location Areas Planning Problem

The location update procedure based on location areas (by definition, a location area is a continuous and non-overlapped group of network cells) is being widely used in current Public Land Mobile Networks. In this location update strategy, a mobile station is free to move inside a given location area without updating its location, and the paging procedure is only performed in the network cells within the last updated location area (because the network knows the subscriber location at a location area level). Therefore, the main challenge of the location areas scheme is to find the configurations of location areas that minimize signaling load associated with the location update procedure (LU_{cost}) and the paging load (PA_{cost}). In this work, we use a two-cycle sequential paging based on the last known location (last updated cell), the same paging procedure is used in [15,2,3,5]. Formally, these two objective functions can be expressed as Equation 1 and Equation 2 respectively, where: $\gamma_{t,i}$ is a binary variable that is equal to 1 when the mobile station i moves out of its current location area in the time t, otherwise $\gamma_{t,i}$ is equal to 0. $[T_{ini}, T_{fin}]$ is the time interval of the mobile activity trace. N_{user} is the number of mobile users. $\rho_{t,i}$ is a binary variable that is equal to 1 only when the subscriber i has an incoming call in the time t. $\alpha_{t,i}$ is a binary variable that is equal to 1 only when the mobile station i is located in its last updated cell in the time t. NA is a vector that stores the number of network cells of each location area. And LA_t is a vector that stores the location area in which every mobile subscriber is located in the time t.

$$f_1 = min\left\{LU_{cost} = \sum_{t=T_{ini}}^{T_{fin}} \sum_{i=1}^{N_{user}} \gamma_{t,i}\right\}, \tag{1}$$

$$f_2 = min\left\{PA_{cost} = \sum_{t=T_{ini}}^{T_{fin}} \sum_{i=1}^{N_{user}} \rho_{t,i}\left(\alpha_{t,i} + \overline{\alpha}_{t,i} \cdot NA\left[LA_t\left[i\right]\right]\right)\right\}. \tag{2}$$

It should be noted that these two objective functions are conflicting. That is, if we would reduce the signaling load of the location update procedure, we should increase the size of the location areas, which leads to an increment in the number of paging messages because a larger number of network cells have to be polled whenever a subscriber has an incoming call. And, on the other hand, if we reduce the paging cost with smaller location areas, we will have more location updates (increasing this other cost).

In previous works from other authors, this optimization problem was simplified by using the linear aggregation of the objective functions (see Equation 3). However, this technique has several drawbacks. Firstly, a very accurate knowledge of the problem is required to properly configure the weight coefficients ($\alpha, \beta \in \Re$). Secondly, different states of the signaling network might require of different values of α and β. And thirdly, a single-objective optimizer must perform an independent run for each combination of the weight coefficients. That is why we propose the use of multiobjective metaheuristics to optimize the LAPP. Furthermore, a multiobjective optimization algorithm provides a wide range of solutions among which the network operator could select the one that best adjusts to the real state of the signaling network.

$$f_{SOA}\left(\alpha, \beta\right) = \alpha \cdot f_1 + \beta \cdot f_2. \tag{3}$$

4 Multiobjective Optimization Paradigm

A Multiobjective Optimization Problem (MOP) can be defined as an optimization problem in which two or more conflicting objective functions have to be optimized simultaneously under certain constraints [6] (e.g. the Location Areas Planning Problem). In the following and without loss of generality, we assume a minimization MOP with two objective functions (as the LAPP). In this kind of problems, the main challenge consists in finding the best possible set of solutions, where each solution is related to a specific trade-off among objectives. These solutions are referred as non-dominated solutions, and the set of

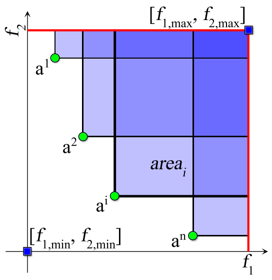


Fig. 1. Hypervolume for a minimization problem with two objectives

non-dominated solutions is commonly known as Pareto Front. For definition, a solution $\mathbf{x}^1$ is said to dominate the solution $\mathbf{x}^2$ (denoted as $\mathbf{x}^1 \prec \mathbf{x}^2$) when $\forall i \in [1,2], f_i\left(\mathbf{x}^1\right) \leq f_i\left(\mathbf{x}^2\right) \wedge \exists i \in [1,2] : f_i\left(\mathbf{x}^1\right) < f_i\left(\mathbf{x}^2\right)$.

In the literature, there are several indicators to measure the quality of a Pareto Front in the multiobjective context. The quality indicators used in this work are presented in Section 4.1 to Section 4.3. Section 4.4 presents a detailed explanation of our version of the MultiObjective Artificial Bee Colony algorithm.

4.1 Hypervolume: $I_H(A)$

In a bi-objective MOP, the Hypervolume indicator ($I_H(A)$) computes the area of the objective space that is dominated by the Pareto Front A, and is bounded by the reference points (points that are calculated by using the maximum and minimum values of each objective function) [6]. Due to the fact that the main goal of a multiobjective optimization algorithm is to find a wide range of non-dominated solutions, the Pareto Front A is said to be better than the Pareto Front B when $I_H(A) > I_H(B)$. Fig. 1 shows the $I_H(A)$ calculation, which can be formally described by Equation 4.

$$I_H(A) = \left\{ \bigcup_i area_i \mid \mathbf{a}^i \in A \right\}. \tag{4}$$

4.2 Set Coverage: $SC(A, B)$

The Set Coverage indicator ($SC(A,B)$) computes the proportion of solutions of the Pareto Front B that are weakly dominated by the solutions of the Pareto Front A [6]. For definition, a solution $\mathbf{x}^2$ is weakly dominated by another solution $\mathbf{x}^1$ (denoted as $\mathbf{x}^1 \preceq \mathbf{x}^2$) when $\forall i \in [1,2], f_i\left(\mathbf{x}^1\right) \leq f_i\left(\mathbf{x}^2\right)$. This indicator establishes that the Pareto Front A is better than the Pareto Front B when $SC(A,B) > SC(B,A)$. The $SC(A,B)$ can be expressed as Equation 5, where the operator $|\cdot|$ represents the number of solutions of a Pareto Front, or the number of solutions that satisfy a given condition.

$$SC\left(A,B\right)=\frac{\left|\left\{\mathbf{b}\in B;\exists\mathbf{a}\in A:\mathbf{a}\preceq\mathbf{b}\right\}\right|}{\left|B\right|}. \quad (5)$$

4.3 ϵ-Indicator: $I_\epsilon\left(A,B\right)$

The ϵ-Indicator ($I_\epsilon(A,B)$) calculates the minimum distance (ϵ) that the Pareto Front B must be translated in the objective space to be weakly dominated by the Pareto Front A [18]. With this indicator, the Pareto Front A will be better than the Pareto Front B when $I_\epsilon(A,B) < I_\epsilon(B,A)$. The $I_\epsilon(A,B)$ can be formally represented as Equation 6.

$$I_\epsilon(A,B)=min\left\{\epsilon\in\Re|\forall\mathbf{b}\in B\exists\mathbf{a}\in A:\forall i\in\left[1,2\right],f_i\left(\mathbf{a}\right)\leq\epsilon\cdot f_i\left(\mathbf{b}\right)\right\}. \quad (6)$$

4.4 Our Multiobjective Artificial Bee Colony Algorithm

In this work, we propose our version of the MultiObjective Artificial Bee Colony algorithm to optimize the Location Areas Planning Problem in a realistic mobile environment. This algorithm was proposed by A. Rubio-Largo et al. in [14] as a multiobjective adaptation of the original Artificial Bee Colony algorithm (ABC), which was proposed by D. Karaboga and B. Basturk in [12]. Basically, the MO-ABC is the original ABC with a *MOFitness* function (used to know the quality of a solution in the multiobjective context), and with the *fast non-dominated sorting* procedure of the well-known Non-dominated Sorting Genetic Algorithm II [7] (used to arrange the population at the end of each generation).

The ABC is a swarm intelligence algorithm based on the foraging behavior of honey bees. In this algorithm, we can distinguish three different entities: the food sources, the artificial colony, and the artificial hive. Every food source represents a possible solution of the problem, being its amount of nectar proportional to the solution quality. The artificial colony (with size N) is constituted by three kinds of bees: employed, onlooker, and scout bees. The employed bees (the first half of the artificial colony) search and exploit food sources around the artificial hive, and then, they share their knowledge (quality of the food sources) with the onlooker bees (the second half of the artificial colony), which wait in the dance area of the artificial hive. Subsequently, every onlooker bee selects an employed bee (with a probability proportional to the quality of its food source), and it becomes employed bee of such food source. And finally, a bee becomes a scout bee when its food source is overexploited. This kind of bees performs random search around the artificial hive in order to find new unexploited food sources.

Fig. 2 shows the task decomposition of our version of the MO-ABC algorithm. As we can see in this figure, the first step consists in initializing the first population of employed bees. Note that each bee exploits a food source (a possible solution of the problem), and therefore, a bee can be expressed as an encoded

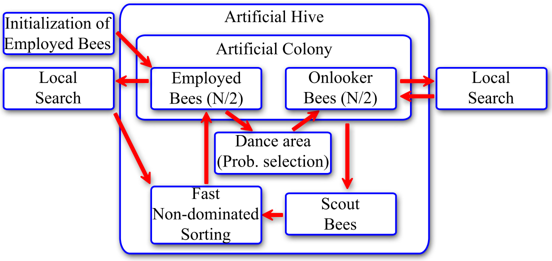
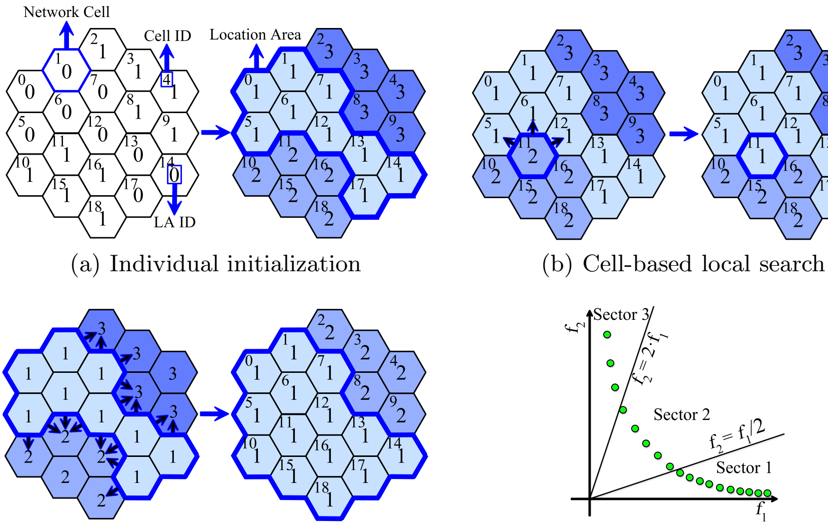


Fig. 2. Task decomposition of our version of the MO-ABC algorithm

solution of the problem. After the initialization process, an iterative method is used to improve this set of initial solutions. In the first step of this iterative method, for each employed bee ($\mathbf{x}^{EB}$), we perform a local search with the aim of finding better solutions ($\mathbf{x}^{newEB}$). Every initial solution is replaced by a new solution when $\mathbf{x}^{newEB} \prec \mathbf{x}^{EB}$. In the next step, the *fast non-dominated sorting* [7] is applied to arrange the employed bees according to their quality in the multiobjective context. Subsequently, every onlooker bee selects an employed bee (probabilistic selection of employed bees, $\mathbf{x}^{OB} \leftarrow \mathbf{x}^{EB}$) and performs a local search in order to improve the selected solution. In this case, the selected solution is replaced by a new solution when $(\mathbf{x}^{newOB} \prec \mathbf{x}^{OB}) \vee (\mathbf{x}^{newOB} \nprec \mathbf{x}^{OB} \wedge \mathbf{x}^{OB} \nprec \mathbf{x}^{newOB})$. With this last condition, we favor the search of new solutions, and even in the same front. The replacement criterion is one of the main differences with the MO-ABC presented in [14], where the *MOFitness* is used to determine if an old solution should be replaced by a new solution. With a criterion based on the dominance concept, we avoid the evaluation of the whole population whenever the local search process finds a new solution. The other differences are in the operators used in our MO-ABC version. The ABC algorithm defines a method to avoid its stagnation. This is done by using the scout bees. For it, every bee has a counter that is incremented whenever its solution is not improved (or replaced) after the local search process. In this way, a bee becomes a scout bee when its counter is higher or equal to a certain number of generations (*limit*). Every scout bee initializes its counter to 0. And finally, the last step of this iterative method consists in selecting the best bees (employed, onlookers, and scouts) as the employed bees of the next generation. For it, we use the *fast non-dominated sorting* to know the quality of every bee (or solution) in the multiobjective context. This iterative method is executed until the stop condition is reached. In this work we use the maximum number of generations as stop condition [2,3,5].

Individual Initialization. As mentioned above, a bee is an encoded solution of the problem. In this work, we use a vector representation in which every position of the vector is an integer that represents a network cell and stores the Location Area assigned to that cell. In order to generate the first employed bees, every vector is filled with a random pattern of 0s and 1s (integer numbers). Afterwards,



(a) Individual initialization (b) Cell-based local search

(c) LA-based local search (d) Probabilistic selection

Fig. 3. Operators of our MO-ABC version

this random pattern is used to determine the corresponding configuration of LAs. For it, and according to the definition of Location Area (a continuous and non-overlapped group of cells), every Location Area will be composed by a continuous group of network cells with the same value of vector. Fig. 3(a) shows an example of this procedure.

Local Searches. The local searches are used to explore the neighborhood of a solution (in the objective space) with the goal of finding new solutions of greater or equal quality. In this work, we propose two local searches: a cell-based local search, and a LA-based local search. The cell-based local search consists in merging a boundary cell (which is randomly selected) with one of its neighboring LAs. Fig. 3(b) shows an example of this local search. And in the LA-based local search, we merge a randomly selected LA with one of its neighboring LAs (see Fig. 3(c)). In these two procedures, we evaluate all the possibilities and then, we select the one that satisfies the replacement criterion (defined above).

Probabilistic Selection. The probabilistic selection is the method used by the onlooker bees to select a solution according to its quality in the multiobjective context. The LAPP has a very important feature, the density of feasible solutions in the objective space decreases when increasing the size of the Location Areas. And therefore, the solutions with high LU_{cost} (solutions with a high number of small location areas) will be selected with higher probability. With the aim of dealing with such inconvenience, we propose to decompose the normalized objective space in three sectors (see Fig. 3(d)). Thus, we can directly control the number of solutions selected in each sector. In this work, we define three kinds of onlooker bees, each one specializing in exploiting a specific sector. Furthermore, in order to cause the least possible impact in the search of the algorithm, the number of onlooker bees assigned to each sector is equal to the number of solutions (employed bees) in that sector. On the other hand, a solution within a sector S_i is selected with probability $p\left(\mathbf{x}^{EB,j}\right) = f_{fit}\left(\mathbf{x}^{EB,j}\right) / \sum_{k \subset S_i} f_{fit}\left(\mathbf{x}^{EB,k}\right)$. The function $f_{fit}(\mathbf{x})$ is a function that returns a real number, which is equal to the weighted sum of the *Ranking* procedure and the *Crowding distance* [7] ($f_{fit}(\mathbf{x}) = 10 \cdot Rank(\mathbf{x}) + Crowd(\mathbf{x})$, only valid for a normalized bi-objective problem). Note that $f_{fit}(\mathbf{x})$ is proportional to the quality of $\mathbf{x}$ in the multiobjective context.

Method for Generating the Scout Bees. The scout bees are the mechanism that the MO-ABC algorithm uses to avoid its stagnation. In this work, we propose the following way to generate a scout bee. Firstly, we select a sector (S_i) with a probability $p\left(S_i\right) = \left|S_i\right|^{-1} / \sum_j \left|S_j\right|^{-1}$, where $|S_i|$ represents the number of solutions in the sector S_i. Note that this probabilistic selection favors the exploitation of those zones of the objective space with lower number of solutions. Afterwards, we generate a scout bee ($\mathbf{x}^{SB}$) by using Equation 7 (where N_{cell} is the number of cells in the network), i.e. every scout bee is the average solution of the selected sector.

$$\mathbf{x}_j^{SB} = \frac{1}{|S_i|} \sum_{k \subset S_i} \mathbf{x}_j^k, \forall j \in \left[0, N_{cell} - 1\right]. \tag{7}$$

5 Experimental Results

With the aim of checking the quality of our algorithm in a realistic mobile environment, we study the real-time mobile activity trace developed by the Stanford University [11]. This mobile network is defined in Section 5.1. Furthermore, we compare our proposal with other algorithms published in the literature. This is shown in Section 5.2.

5.1 Standford University Mobile Activity Traces

The Stanford University Mobile Activity TRAces (SUMATRA) are a set of test networks that are available via the Internet [11]. In this work, we study the BALI-2 network, a mobile network with 90 cells and 66,550 subscribers. The main appeal of this mobile network is that it corresponds with a real-time mobile activity trace that is well-validated against real data measured in the San Francisco Bay (USA). And therefore, we will be able to analyze the behavior of our algorithm in a realistic mobile environment.

Table 1. Comparison with other works

Algorithm	#points	median (I_H)	iqr (I_H)	best $f_{SOA}(10,1)$
MO-ABC	958	93.90	3.53e-04	2,616,330
NSGAII[5]	772	93.75	1.79e-03	2,619,839
SPEA2[4]	752	92.79	2.02e-03	2,619,697
DE[2]	1	-	-	2,799,289
SS[3]	1	-	-	2,756,836
DBLA[15]	1	-	-	2,695,282

Table 2. Comparison among MOEAs: $SC\,(PF_A, PF_B)$

PF_A \ PF_B	MO-ABC	NSGAII[5]	SPEA2[4]
MO-ABC	-	66.06	63.43
NSGAII[5]	30.17	-	48.94
SPEA2[4]	26.10	38.21	-

Table 3. Comparison among MOEAs: $I_\epsilon\,(PF_A, PF_B)$

PF_A \ PF_B	MO-ABC	NSGAII[5]	SPEA2[4]
MO-ABC	-	1.03	1.01
NSGAII[5]	1.24	-	1.01
SPEA2[4]	1.32	1.33	-

5.2 Comparison with Other Works

In this section we present a comparison with other algorithms published in the literature [15,2,3,5,4], two of them are our previous multiobjective algorithms (NSGAII [5], and SPEA2 [4]). In order to perform a fair comparison, we have implemented the same local searches in all of our multiobjective algorithms. Furthermore, we use the same population size (300 individuals or bees) and the same stop condition (1000 generations) as the algorithms proposed in [2,3,5,4]. The other parameters of our MO-ABC algorithm have been configured by means of a parametric study of 31 independent runs per experiment. We have selected the parameter configuration that maximizes the I_H value: $limit = 15$, $P_{C-LS} = 0.9$, and $P_{LA-LS} = 0.1$, where P_{C-LS} is the probability of performing the cell-based local search and P_{LA-LS} is the probability of performing the LA-based local search. In this work, we have defined these two local searches in a way that a bee must only perform one local search per generation.

Table 1 gathers a summary of our comparative study (of 31 independent runs per experiment). In this table we compare our multiobjective algorithms by using statistical data of the I_H (median and interquartile range), and the number of points of the Pareto Front associated with the median I_H. In this paper, we use the same reference points as in [5,4]. This comparative study clarifies that our MO-ABC algorithm performs better than our previous multiobjective algorithms because it obtains a higher number of solutions and it achieves a higher I_H value with a lower interquartile range. The same conclusion can be reached if we compare our algorithms with the Set Coverage (see Table 2) and the ϵ-Indicator (see Table 3).

Furthermore, we present a comparison with other approaches published in the literature, all of them Single-objective Optimization Algorithms (SOAs) in which the fitness function is Equation 3 with $\alpha = 10$ and $\beta = 1$ [15,2,3]. Regrettably, the best solution found (of a set of runs) is the only data available in these papers, so we cannot perform a statistical comparison. In order to accomplish this comparison, we have searched in our median Pareto Front the solution that best meets the objective function used by these SOAs (Equation 3 with $\alpha = 10$ and $\beta = 1$). Table 1 reveals that our MO-ABC is very competitive because it outperforms the SOAs proposed in [2,3] and the Distance-Based Location Area (DBLA) algorithm presented in [15], which is far from trivial because these algorithms [15,2,3] are specializing in finding a single solution.

6 Conclusion and Future Work

In this work, we present our version of the MultiObjective Artificial Bee Colony algorithm (MO-ABC, an optimization algorithm based on the foraging behavior of honey bees) to optimize one of the most important planning problems in any Public Land Mobile Network (PLMN): the Location Areas Planning Problem. Furthermore, with the aim of checking the quality of our algorithm in a realistic mobile environment, we study the real-time mobile activity trace developed by the Stanford University [11]. The main contribution of our MO-ABC algorithm is the definition of specific local searches and the decomposition of the normalized objective space in three sectors in order to directly control the probabilistic selection, and hence the search of the algorithm. Experimental results show that our proposal is very promising because it outperforms other algorithms published in the literature.

As a future work, it would be interesting to study the effectiveness of each operator (cell-based local search, LA-based local search, etc.) independently. Furthermore, it would be a good challenge to study other multiobjective metaheuristics [17] and compare them with our MO-ABC algorithm.

References

1. Agrawal, D., Zeng, Q.: Introduction to Wireless and Mobile Systems. Cengage Learning (2010)
2. Almeida-Luz, S.M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Applying Differential Evolution to a realistic location area problem using SUMATRA. In: Proceedings of The Second International Conference on Advanced Engineering Computing and Applications in Sciences, pp. 170–175 (2008)
3. Almeida-Luz, S.M., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: Solving a realistic location area problem using SUMATRA networks with the Scatter Search algorithm. In: Proceedings of the Ninth International Conference on Intelligent Systems Design and Applications, pp. 689–694 (2009)
4. Berrocal-Plaza, V., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M., Gómez-Pulido, J.A.: Applying the Strength Pareto Evolutionary Algorithm to solve the location areas planning problem in realistic networks. In: 14th International Conference on Computer Aided Systems Theory, pp. 176–178 (2013)
5. Berrocal-Plaza, V., Vega-Rodríguez, M.A., Sánchez-Pérez, J.M., Gómez-Pulido, J.A.: Solving the location areas scheme in realistic networks by using a multiobjective algorithm. In: Esparcia-Alcázar, A.I. (ed.) EvoApplications 2013. LNCS, vol. 7835, pp. 72–81. Springer, Heidelberg (2013)

6. Coello, C.A.C., Lamont, G.B., Veldhuizen, D.A.V.: Evolutionary Algorithms for Solving Multi-Objective Problems (Genetic and Evolutionary Computation). Springer-Verlag New York, Inc., Secaucus (2006)
7. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multiobjective genetic algorithm: NSGA-II. IEEE Transactions on Evolutionary Computation 6(2), 182–197 (2002)
8. Demestichas, P., Georgantas, N., Tzifa, E., Demesticha, V., Striki, M., Kilanioti, M., Theologou, M.E.: Computationally efficient algorithms for location area planning in future cellular systems. Computer Communications 23(13), 1263–1280 (2000)
9. Garg, V.: Wireless Communications & Networking, 1st edn. Morgan Kaufmann Publishers Inc., San Francisco (2007)
10. Gondim, P.R.L.: Genetic algorithms and the location area partitioning problem in cellular networks. In: Procedings of the IEEE 46th Vehicular Technology Conference on Mobile Technology for the Human Race, vol. 3, pp. 1835–1838 (1996)
11. Jannink, J., Cui, Y.: Stanford university mobile activity traces (sumatra), `http://infolab.stanford.edu/sumatra` (accessed in 2013)
12. Karaboga, D., Basturk, B.: Artificial bee colony (ABC) optimization algorithm for solving constrained optimization problems. In: Melin, P., Castillo, O., Aguilar, L.T., Kacprzyk, J., Pedrycz, W. (eds.) IFSA 2007. LNCS (LNAI), vol. 4529, pp. 789–798. Springer, Heidelberg (2007)
13. Kyamakya, K., Jobmann, K.: Location management in cellular networks: classification of the most important paradigms, realistic simulation framework, and relative performance analysis. IEEE Transactions on Vehicular Technology 54(2), 687–708 (2005)
14. Rubio-Largo, A., Vega-Rodríguez, M.A., Gómez-Pulido, J.A., Sánchez-Pérez, J.M.: A multiobjective approach based on artificial bee colony for the static routing and wavelength assignment problem. Soft Comput. 17(2), 199–211 (2013)
15. Subrata, R., Zomaya, A.Y.: Dynamic location management for mobile computing. Telecommunication Systems 22(1-4), 169–187 (2003)
16. Wong, V.W.S., Leung, V.C.: Location management for next-generation personal communications networks. Netwrk. Mag. of Global Internetwkg. 14(5), 18–24 (2000)
17. Zhou, A., Qu, B.Y., Li, H., Zhao, S.Z., Suganthan, P.N., Zhang, Q.: Multiobjective evolutionary algorithms: A survey of the state of the art. Swarm and Evolutionary Computation 1(1), 32–49 (2011)
18. Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C., da Fonseca, V.: Performance assessment of multiobjective optimizers: an analysis and review. IEEE Transactions on Evolutionary Computation 7(2), 117–132 (2003)