A comparison of different optimisation search methodologies for self-optimisation in wireless cellular networks

J. Sánchez-González, O. Sallent, J. Pérez-Romero, R. Agustí
Department of Signal Theory and Communications (TSC)
Universitat Politècnica de Catalunya (UPC)
C/ Jordi Girona 1-3, Campus Nord, 08034, Barcelona, Spain
e-mail: juansanchez@tsc.upc.edu

***Abstract*—Self-Organising Networks (SON) concept is seen as a way to reduce costs by automating functionalities such as network optimisation usually performed manually with extensive human work time. This paper provides a general formulation of the self-optimisation problem in a cellular wireless network and a description of the optimisation search process by means of iterative algorithms. Different optimisation methodologies, namely simulated annealing, genetic and particle swarm algorithms have been considered. These methodologies have been implemented for the optimisation of the cell coverage and cell overlap using real measurements of a UMTS network deployed in a medium-size European city. These methodologies have been compared in terms of speed of convergence and performance of the solutions provided by the different proposed algorithms.**

I. INTRODUCTION

In the last years, a drastic increase in the mobile data traffic has been experienced since the widespread deployment of third-generation (3G) mobile communication systems, the introduction of HSPA (High Speed Packet Access) and the recent take-off of LTE (Long Term Evolution). Due to the increasing demand for wideband services in a competitive market, network operators are always investing large budgets to deploy and upgrade their networks. This process was traditionally done manually or semi-automatically. However, being wireless networks inherently dynamic and very sensitive to traffic and interference variations, which is particularly more relevant with the CDMA (Code Division Multiple Access) and OFDMA (Orthogonal Frequency Division Multiple Access) based radio interfaces of 3G and LTE systems, this approach can easily lead to inefficient operations, which will become more critical when broadband services are to be provided. On the other hand, the envisaged high density of small sites, e.g. with the introduction of femto-cells, and the pressure to reduce costs clearly indicate that deploying and running networks needs to be more cost-effective. The introduction of Self-Organising Networks (SON) functionalities, aiming to configure and optimize the network automatically, is seen as one of the promising areas for an operator to save operational expenditures (OPEX). For this reason, SON has received a lot of attention in recent years in different standardization fora, research projects as well as academic works [1]-[5].

Clearly, the SON concept at the largest extent of a totally automatic network, able to operate with minimum human intervention, is quite ambitious and challenging, so that it can be anticipated that SON will continue as a hot research topic in coming years requiring further research efforts to facilitate its practical implementation. In accordance with this, in [1] a roadmap is presented starting from the practical application of UMTS (Universal Mobile Telecommunications System) optimisation towards automated self-optimisation procedures in LTE, thus outlining a roadmap from current deployed networks (2G, 3G HSPA), mainly managed by centralized remote operations and maintenance (O&M) applications with intensive human intervention, to future SONs (LTE, HSPA+).

The self-optimisation process can be seen as the automatic determination of the most adequate values of several network configuration parameters to optimise the network performance in terms of specific optimisation targets defined by the network operator (such as the avoidance of coverage holes, reduction of cell overshooting effects, etc. [1]). In a real network, a large amount of different network configuration parameters can be tuned. Furthermore, due the inter-cell coupling effects, the changes done in one cell may influence on the performance observed in the area of another cell. Correspondingly, when considering large cellular networks consisting of hundreds of cells, the resulting number of possible network configurations increases dramatically. As a consequence, the use of self-optimisation algorithms becomes necessary since it is very hard for an engineer to cope manually with this level of complexity.

A wide range of possible strategies have been proposed in the literature to find automatically the optimum or at least some acceptable sub-optimum solution for different optimisation problems. The selection of the optimisation search strategy typically depends on the problem complexity, the time available to develop and implement the optimisation technique and the necessity to obtain solutions with an optimum objective value. Some simple optimisation methodologies such as greedy algorithms, local search techniques or Tabu Search have been commonly used for optimisation [6]-[8]. In turn, other methodologies (such as simulated annealing, genetic algorithms or particle swarm algorithms) provide in general better solutions at the expense of increasing the computation time and the algorithm complexity [9]-[11]. Several works that make use of these algorithms for the optimisation of mobile

communication networks are [12]-[14]. Most of the works that can be found in the literature related to cellular wireless networks optimisation focus on a particular optimisation methodology. However, the comparison of different optimisation techniques for the self-optimisation problem in cellular wireless networks is not usually addressed.

Within this context, the first contribution of this work is a general formulation of the optimisation problem in a cellular wireless network and a description of the optimisation search procedure. The second contribution of the paper is the description and comparison of different optimisation search methodologies (i.e. simulated annealing, genetic algorithm and particle swarm) in terms of algorithm convergence and performance of the solutions provided by the different algorithms. This comparison has been done for a particular case study using real measurements of a wireless cellular network in which the objective is the optimisation of the cell coverage and the cell overlap by adjusting the CPICH (Common Pilot CHannel) transmitted power of the different cells. The rest of the paper is organised as follows. Section II provides a general description of the optimisation problem and the general procedure for the network optimisation search. Section III provides a brief description of different optimisation techniques that are proposed for the network optimisation. Section IV presents the comparison of the different optimisation methodologies for the considered case study. Conclusions are summarized in Section V.

II. Optimisation procedure

A general scenario consisting of N cells with P tuneable parameters per cell is considered. The network configuration is represented by a $P \times N$ matrix $\boldsymbol{\psi}=[\psi_{p,n}]$ where the term $\psi_{p,n}$ denotes the value of the p-th tuneable parameter of the n-th cell. The set of possible values of parameter $\psi_{p,n}$ is the range $[V_{min,p},V_{max,p}]$ with resolution Δv_p. The proposed self-optimisation procedure is illustrated in Figure 1 and consists in a continuous loop that interacts with the real network based on *observations* and *actions* [1]. The objective of the network optimisation loop is to determine the most adequate network configuration $\boldsymbol{\psi}$ to simultaneously achieve M optimisation targets specified by the network operator. At the observation phase, certain measurements are collected from the network. The self-optimisation procedure runs a Network Performance Monitoring process that analyses the measurements to detect the situations where some of the optimisation targets are not properly fulfilled, so the network is behaving sub-optimally. When this is detected, the Optimisation Search process is triggered in order to find the adequate value of the network configuration parameters $\psi_{p,n}$ that solve the sub-optimal operation situation. In that case, the *action* consists in applying to the network the configuration $\boldsymbol{\psi}$ obtained by the Self-Optimisation process. In the following, the Network Performance Monitoring and Optimisation Search procedures are briefly described.

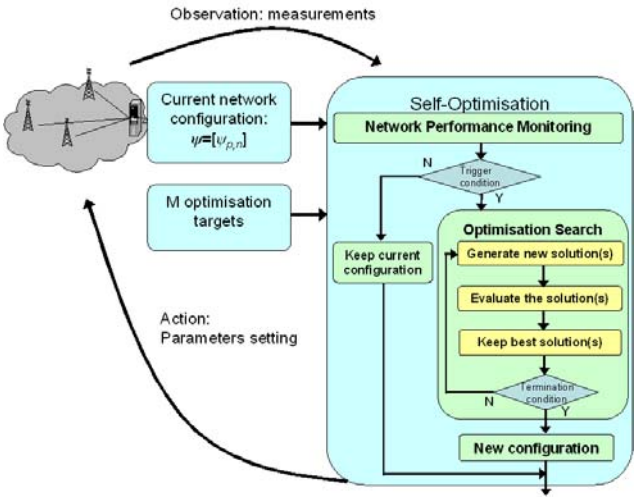

Figure 1.- Network optimisation loop.

A. Network Performance Monitoring

This stage is in charge of analysing the collected set of measurements in accordance with the M operator specific optimisation targets. This process is carried out on a cell-by-cell basis and the result will be the $M \times N$ performance matrix $\boldsymbol{S(\psi)}=[S_{m,n}(\boldsymbol{\psi})]$ in which the term $S_{m,n}(\boldsymbol{\psi})$ ($0 \le S_{m,n}(\boldsymbol{\psi}) \le 1$) reflects the performance obtained by the n-th cell in terms of the m-th optimisation target with the current network configuration $\boldsymbol{\psi}$. The higher the value of $S_{m,n}(\boldsymbol{\psi})$, the more likely that the m-th target is not sufficiently optimised in the n-th cell. Based on the elements of matrix $\boldsymbol{S(\psi)}$ a trigger condition will be evaluated to decide if the measured performance is sufficiently satisfactory or if the network needs to be further optimised thus triggering the optimisation search. An example of how the $S_{m,n}(\boldsymbol{\psi})$ metrics are evaluated is presented in [15].

B. Optimisation Search

The optimisation search problem can be formulated as the search of the network configuration parameters in matrix $\boldsymbol{\psi}=[\psi_{p,n}]$ that optimise the network performance given by matrix $\boldsymbol{S(\psi)}$. This is a multi-cell and multi-objective problem since it involves N cells and M optimisation targets. In general, the optimisation targets can be partly contradictory (e.g. an increase in the transmitted power devoted to the pilot channel may reduce the existence of coverage holes but may cause an increase in the cell overlap and interference [16]). For this reason, the network operator has to specify a trade-off criterion among the different optimisation goals. A usual approach is to define a joint objective or cost function as a linear combination of the different quality measures with certain weights β_m assigned to each optimisation target, given by:

$$C(\psi) = \sum_{m=1}^{M} \beta_m \sum_{n=1}^{N} S_{m,n}(\psi) \qquad (1)$$

The weights β_m $m=1,...,M$ allow the operator to give more relevance to certain targets in the optimisation process in front

of others. The optimum solution is thus given by the configuration matrix $\boldsymbol{\psi}^*$ that minimises the cost, that is:

$$\boldsymbol{\psi}^* = \arg\min_{\boldsymbol{\psi}} C(\boldsymbol{\psi}) \quad (2)$$

The process of finding the optimum solution $\boldsymbol{\psi}^*$ in a real wireless network is rather complex and challenging, particularly in real wireless networks consisting of tens, or even hundreds, of cells and several tuneable configuration parameters per cell. This requires the use of smart optimisation search methodologies able to efficiently explore the search space while keeping the computational complexity at acceptable levels. The methodologies considered in this paper are based on an iterative process. As shown in Figure 1, in the Optimisation Search phase, new candidate solution(s) are generated in each iteration. These candidate solutions are evaluated and the best ones in terms of cost are selected for the generation of new solution(s) in the subsequent iteration. Both the generation and the selection procedures are specified by the optimisation search methodology as it will be detailed in section III. In general, with this approach, the algorithm provides better solutions as the number of iterations is increased. The optimisation search algorithm is run until a termination condition is fulfilled that evaluates whether the cost has reached a given threshold or whether a maximum number iterations has been achieved. The best solution found after this iterative process becomes the output of the Self-Optimisation procedure.

III. OPTIMISATION SEARCH METHODOLOGIES DESCRIPTION

This section presents a brief description of the three optimisation search methodologies considered in this paper, namely simulated annealing, genetic algorithms and particle swarm.

A. Optimisation search based on simulated annealing (SA).

In the simulated annealing technique, a new candidate solution of the optimisation problem, i.e., a candidate network configuration $\boldsymbol{\psi}$, is generated in each iteration [9]. Figure 2 shows the pseudo-code of the simulated annealing methodology. First, the considered initial solution is the current configuration being used in the real network. The algorithm evaluates this solution $\boldsymbol{\psi}$ and calculates the corresponding cost $C(\boldsymbol{\psi})$ according to equation (1). Then, a new solution $\boldsymbol{\psi}'$ is generated by modifying part of the current one. In general, this is done by selecting randomly a particular p-th network configuration parameter in the n-th cell ($\psi_{p,n}$) and setting the new value for this parameter $\psi'_{p,n}$ either as an increase or a decrease (determined randomly with equal probability) in one resolution unit within the defined search space of the corresponding parameter, as shown in equation (3).

$$\psi'_{p,n} = \begin{cases} \min\left(\psi_{p,n} + \Delta v_p, V_{\max,p}\right) & with \quad prob = 0.5 \\ \max\left(\psi_{p,n} - \Delta v_p, V_{\min,p}\right) & with \quad prob = 0.5 \end{cases} \quad (3)$$

The new solution $\boldsymbol{\psi}'$ is evaluated and its cost $C(\boldsymbol{\psi}')$ is determined. If the cost of this new solution $\boldsymbol{\psi}'$ is lower than the cost of the previous one $\boldsymbol{\psi}$, (i.e. $C(\boldsymbol{\psi}')<C(\boldsymbol{\psi})$), then this new solution is accepted (i.e. the previous solution is substituted by the new one). Otherwise, even if the cost of the new solution is higher than the previous one, the solution may also be accepted according to an acceptance probability P_{accept} that decreases as a function of the number of iterations carried out. In general, a negative exponential function is considered which is controlled by a term called *Temperature T* according to equation (4).

$$P_{accept} = e^{-\frac{[C(\psi)-C(\psi')]}{T}} \quad (4)$$

In each iteration x, the temperature is decreased in steps of Δ_T according to the cooling function $T(x+1)=T(x)-\Delta_T$, so that, in the first iterations (the so-called *exploration phase*) a high acceptance probability is usually considered while in the last iterations (*convergence phase*) a lower acceptance probability is set [9]. The possibility to accept new solutions with higher cost than the previous ones avoids the algorithm to get stuck in local minima. The whole process is run iteratively until the termination condition is fulfilled, i.e. the cost of the current solution is below a threshold or the maximum number of iterations $I_{max,SA}$ has been reached.

```
x ← 0
ψ ← Initial solution;
T(x) ← Initial temperature T0
Repeat
        ψ' ← N(ψ); //derive a new solution
        If C(ψ')<C(ψ) then {
                ψ ← ψ';
        else
                Generate uniform random variable Z∈[0,1]
                If Z<P_accept then
                        ψ ← ψ';
                else    keep ψ as the best solution found.
        }
        T(x+1) ← T(x) - Δ_T;
        x ← x+1;
until termination condition is fulfilled;
```

Figure 2.- Simulated annealing algorithm.

B. Optimisation search based on genetic algorithms (GA).

Genetic algorithms are based on the analysis of a set of N_{POP} candidate solutions that constitute the so-called *population* of the algorithm [10]. Each solution corresponds to a network configuration matrix, denoted as $\boldsymbol{\psi}^i$ i=1,...,N_{POP}. These solutions are modified in the successive iterations (i.e. *generations*) following certain rules inspired by the natural evolution principles. Each of the possible solutions $\boldsymbol{\psi}^i$ is called *chromosome* or *individual* and each of the elements $\psi^i_{p,n}$ of the network configuration matrix $\boldsymbol{\psi}^i$ is called *gene*. Figure 3 shows the pseudo-code of the genetic algorithm. The algorithm starts with the initialisation of the N_{POP} individuals of the population corresponding to the first generation. The current network configuration is one of these individuals (i.e. $\boldsymbol{\psi}^1=\boldsymbol{\psi}$). For the rest of individuals i=2,...,N_{POP} the network configuration parameters are chosen randomly with uniform distribution within the range of each parameter $[V_{min,p}, V_{max,p}]$ with a resolution Δv_p. Each individual $\boldsymbol{\psi}^i$ is evaluated in terms of its

cost $C(\psi^i)$. Once all the individuals of a particular generation have been evaluated, the genetic algorithm proceeds with a new generation and creates N_{POP} new individuals making use of the selection, recombination and mutation operators. These operators model the evolution process using some evolution principles that help the algorithm to provide better solutions as the generations evolve [10]. In this paper, the considered selection process is the so-called "cost proportional selection" in which individuals are randomly selected with a probability that depends on their cost, so that individuals with lower cost are selected more often than those with higher cost. The recombination consists in making a combination of the different genes (i.e. the network configuration parameters) of the two individuals selected in the previous step (called *parents*) to generate two new individuals (called *children*). The recombination process considered here is the so-called "1-point crossover" [10]. Finally, the mutation process makes small random changes in the individuals after recombination. Mutation is responsible for introducing new gene material into the population. When the new generation of solutions is obtained after the selection, recombination and mutation processes, the corresponding costs $C(\psi^i{}')$ are determined. This iterative process is repeated until the termination condition is fulfilled, i.e. the minimum cost in the current generation is below a threshold or the maximum number of $I_{max,GA}$ iterations (generations) has been reached.

```
x ← 0
Initialise Population of N_POP solutions
Repeat
    Evaluate (Population)  // determine the cost C for each solution
    Repeat
        Select two solutions of the population according to the costs
        Recombine the selected solutions
        Mutation over the recombined solutions
    Until A new set (Population) of N_POP solutions are obtained
    x ← x+1;
Until termination condition is fulfilled
```

Figure 3.- Genetic algorithm.

C. Optimisation search based on particle swarm (PS).

Particle swarm optimisation is a robust technique inspired by the social behaviour of flocking organisms [11]. The algorithm considers a population (called *swarm*) of N_{part} *particles* that move inside a solution search space of $D=P\times N$ dimensions. At a given iteration x, the *position* of the i-th particle inside this space is given by the $P\times N$ components $\psi^i_{p,n}(x)$ of a candidate solution $\boldsymbol{\psi^i(x)}$ $i=1,...,N_{part}$. In each iteration of the algorithm, all the particles make a movement according to a certain *velocity* that determines their next positions (i.e. the next candidate solutions) in the subsequent iteration. The i-th particle velocity at the x-th iteration is represented by matrix $\boldsymbol{v^i(x)}=[v^i_{p,n}(x)]$ where the term $v^i_{p,n}(x)$ is the change step of the p-th tuneable parameter in the n-th cell between iterations x and $x+1$. The pseudo-code of the particle swarm optimisation algorithm is presented in Figure 4. First, the initial position and velocity of all the N_{part} particles is determined. The current configuration being used in the real network is considered as the initial position of the first particle ($i=1$). The initial position for the rest of the particles $i=2,...,N_{part}$ is set randomly with uniform distribution inside the solution space search. The particles' initial velocities $\boldsymbol{v^i(0)}$ are also determined randomly. At the x-th iteration, the algorithm evaluates all the possible network configurations $\boldsymbol{\psi^i(x)}$ for the different particles and determines their cost $C(\boldsymbol{\psi^i(x)})$. This cost is used in each iteration to determine (and update if necessary) the best historical configuration found for each particle $\boldsymbol{\psi^i_{best}(x)}$ (the so-called *individual component* that represents the solution with lowest cost found by the i-th particle during the execution of the algorithm up to iteration x) and the best historical configuration found for the entire swarm of particles $\boldsymbol{\psi_{social}(x)}$ (the so-called *social component* that represents the best solution found by the entire swarm of particles up to iteration x). Then, the new position and the new velocity of each particle are determined using the following expressions:

$$\psi^i_{p,n}(x+1)=\psi^i_{p,n}(x)+v^i_{p,n}(x) \quad (5)$$

$$v^i_{p,n}(x+1)=k_1 v^i_{p,n}(x)+k_2 r_2\left(\psi^i_{best}(x)-\psi^i_{p,n}(x)\right)+k_3 r_3\left(\psi_{social}(x)-\psi^i_{p,n}(x)\right) \quad (6)$$

Each particle movement comprises three contributions that depend on $\boldsymbol{v^i(x)}$, $\boldsymbol{\psi^i_{best}(x)}$ and $\boldsymbol{\psi_{social}(x)}$ as illustrated in Figure 5 and reflected in the velocity update of equation (6) where r_2 and r_3 are two random variables uniformly distributed between [0,1]. The weights k_1, k_2 and k_3 allow the algorithm to give more relevance to the individual or the social component. The whole process is repeated several iterations until a termination condition is fulfilled, i.e. the minimum cost among all the particles in a given iteration is below a threshold or the maximum number of iterations $I_{max,PS}$ has been reached.

```
x ← 0
Initialise positions and velocities of each of the N_part particles
Repeat
  Evaluate all the particles   //determine the cost C for each particle
  Update ψ^i_best and ψ_social
  For each particle i repeat {
    For each combination of n and p repeat {
      Determine new value of ψ^i_p,n according to equation (5)
      Determine new value of v^i_p,n according to equation (6)
    }
  }
  x ← x+1;
Until termination condition is fulfilled
```

Figure 4.- Particle swarm algorithm.

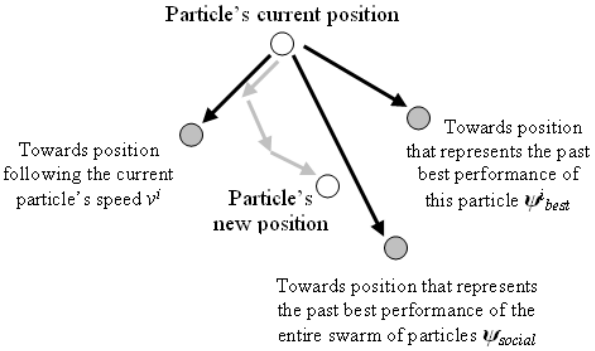

Figure 5.- Update of a particle position in the solution space.

IV. RESULTS

This section presents a case study to illustrate the performance of the proposed optimisation search methodologies using measurements of a real UMTS network. Drive test data was collected in an urban area of a medium-size European city consisting on 18 UMTS cells distributed in 6 Node-Bs as shown in Figure 6. The different cells are identified as *Cell_id* where *id* is a number that represents an identifier of each cell. Drive test measurements were carried out along certain streets as represented in Figure 6. The CPICH transmitted power is initially set to 30dBm for all the cells in the scenario.

The considered $M=2$ optimisation targets are cell coverage and cell overlap optimisation. The coverage of a cell is related to the ability to establish a communication in the cell's service area. In turn, cell overlap exists in the areas where access to the network is possible through multiple cells. A certain degree of cell overlap is useful to facilitate the handover process. However, a large overlapping may generate excessive interference and soft handover overheads.

The measurements analysed by the Network Performance Monitoring for these two optimisation targets are the CPICH received power, the Active Set lists and the User Equipment transmitted power. For details on how to combine these measurements to obtain the corresponding $S_{m,n}(\psi)$ values the reader is referred to [15]. The Network Performance Monitoring stage identifies some regions where coverage and overlap targets are not properly optimised. Figure 7 presents a map of the identified coverage and overlap problems detected with the current configuration being used in the network before applying the optimisation procedure.

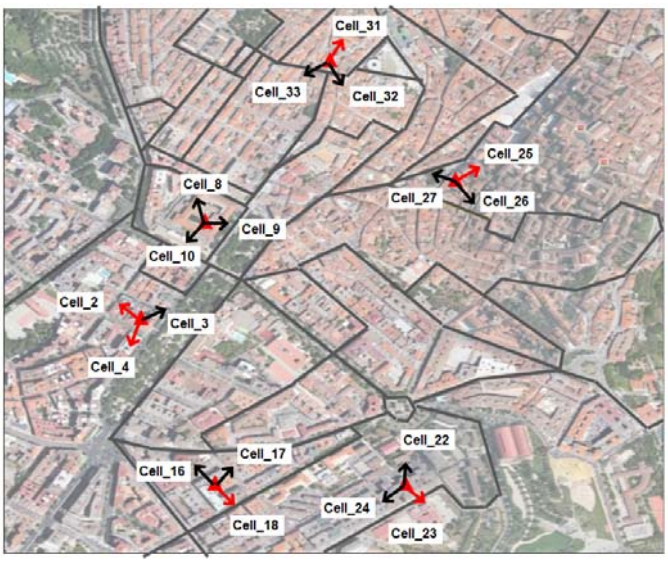

Figure 6.- Considered scenario.

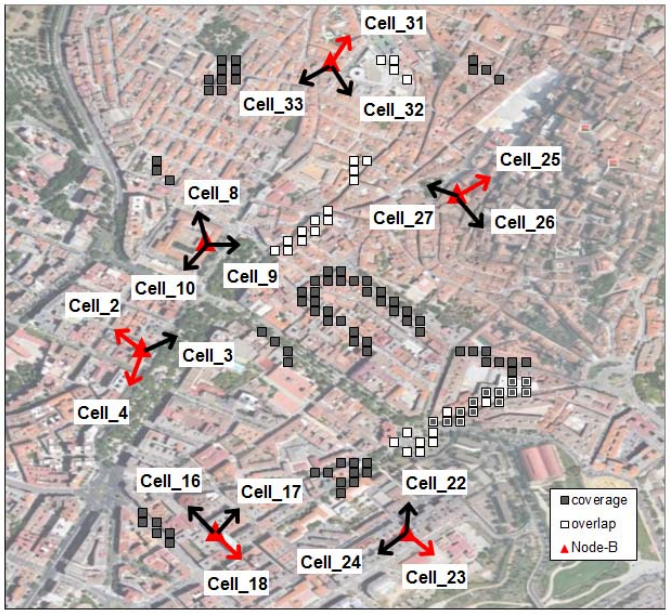

Figure 7.- Location of coverage and overlap problems.

The proposed optimization algorithms adjust the CPICH transmitted power for the $N=12$ cells marked in black colour in Figure 6. The range of variation is between $V_{min,l}=25dBm$ and $V_{max,l}=35dBm$ in steps of $\Delta v_l=1dB$. The weights for the calculation of the cost $C(\psi)$ are assumed to be the same for both optimisation targets (i.e. $\beta_1=\beta_2=1$). After several tests of the different methodologies in the considered optimisation problem, an adequate value for the different optimisation search parameters has been determined for each methodology. Table 1 shows a summary of the considered parameters for the different optimisation search techniques. For the sake of comparison, in all the optimisation methodologies 960 candidate configurations are analysed. This has been done by a proper setting of the maximum number of iterations for each algorithm. Specifically, for the SA, each iteration corresponds to the analysis of one candidate configuration, so $I_{max,SA}$ is set to 960. In turn, the population based algorithms (i.e. GA and PS) evaluate 40 configurations in each iteration, so the number of iterations is set to 24, so that a total of 960 solutions are also evaluated. In all the cases, and to focus the analysis on the convergence aspects, the termination condition according to the cost threshold has not been implemented.

Table 1.- Considered parameters.

	Parameter	Value
Simulated Annealing (SA) parameters	$I_{max,SA}$	960
	T_0 (Initial temperature)	0.4
	Δ_T	0.4/960
Genetic Algorithm (GA) parameters	N_{POP}	40
	$I_{max,GA}$	24
	Mutation probability	3%
Particle Swarm (PS) parameters	N_{part}	40
	$I_{max,PS}$	24
	k_1	1
	k_2	2
	k_3	2

In the following, a comparison in terms of algorithm convergence is presented for the three considered optimisation search techniques. With the available amount of measurements, the process of generation of the new candidate solution(s) to be evaluated is practically negligible (for the considered algorithms) with respect to the time to evaluate the proposed network configurations. For this reason, Figure 8 represents the cost of the best solution found by each algorithm as a function of the number of solutions that are evaluated which, in this sense, is used to represent the elapsed time. As shown, GA and PS algorithms provide faster convergence than SA. Note that in the first iterations, both GA and PS are able to find good solutions since both algorithms initially generate a group of random solutions spread all over the solution search space and, as a consequence, a broader search is done during the initial phase in comparison to SA. Note that the cost of the final solution found by SA is considerably higher than the final cost for GA and PS algorithms. In fact, GA and PS algorithms are able to find the same final solution thus leading to the same final cost for both methodologies as shown in Figure 8. Nevertheless, PS algorithm is able to find the best configuration after 200 solutions analysed in contrast to GA that needs to evaluate 920 solutions, which reflects the faster convergence provided by the PS methodology.

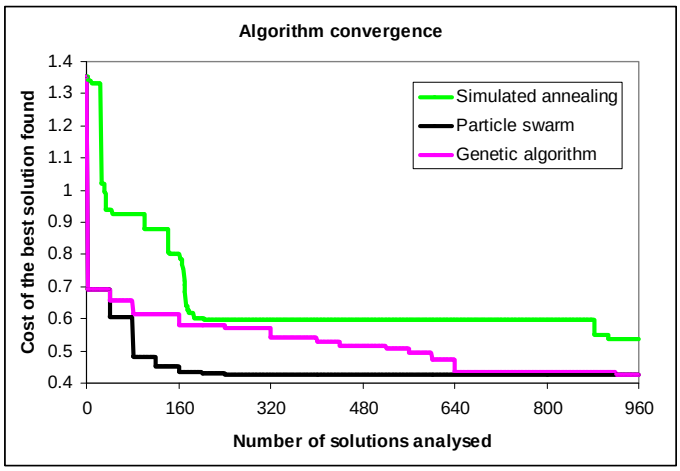

Figure 8.- Algorithm convergence.

The following results present a comparison in terms of network performance for the final solution found by the different optimisation methodologies proposed in this paper. These results have been obtained by doing an estimation of the network performance in terms of cell coverage and cell overlap that would be observed for the best configuration found by each optimisation technique. In particular, Figure 9 represents a map with the location of the coverage and the overlap problems that can be observed for the best configuration found with the SA algorithm. In turn, Figure 10 represents the performance of the configuration found by GA and PS algorithms (this configuration is the same for these two algorithms). Table 2 presents the values of $S_{m,n}$ for the $M=2$ optimisation targets and the $N=12$ considered cells. It compares the values obtained with the initial network configuration against the final result obtained by each algorithm. The term $S_{1,n}$ represents the degree of the coverage problems suffered in the n-th cell and the term $S_{2,n}$ corresponds to the degree of the overlap problems caused by the n-th cell. Note that, since $\beta_1=\beta_2=1$, the values of $S_{1,n}$ correspond to the contribution of the coverage to the total cost in equation (1) while the values $S_{2,n}$ correspond to the contribution of the overlap. Finally, Table 3 presents the initial network configuration in terms of the CPICH transmitted power configured before the optimisation procedure is run and the best configuration found by each optimisation algorithm. By comparing Figures 9 and 10 with Figure 7, it can be observed that the configurations provided by the optimisation methodologies are able to solve some coverage and overlap problems, especially the ones identified in the central region of the considered scenario. Some problems located at the borders could be addressed by adjusting the network configuration parameters of some neighbouring cells not considered in this analysis. As shown in Table 2, the pilot power adjustments carried out by SA mainly reduce the coverage problems with respect to the initial configuration (observe a total coverage cost of 1.018 in the initial configuration and a total coverage cost of 0.307 with the solution provided by SA). On the other hand, the configuration found by GA and PS (see Table 3) allows a further reduction of the cost with respect to SA (see Table 2). As shown in Figure 10, GA and PS remove some geographical regions with problems that are present with the configuration provided by the SA algorithm in Figure 9. Moreover, some regions remain with coverage and overlap problems with the configuration found by GA and PS, but with a lower level with respect to SA, which is represented with lower values of $S_{m,n}$ in Table 2. It is worth noting that the optimisation procedure only considers the adjustment of the CPICH power. Including other network configuration tuneable parameters (such as antenna tilt or azimuth) may improve the network performance.

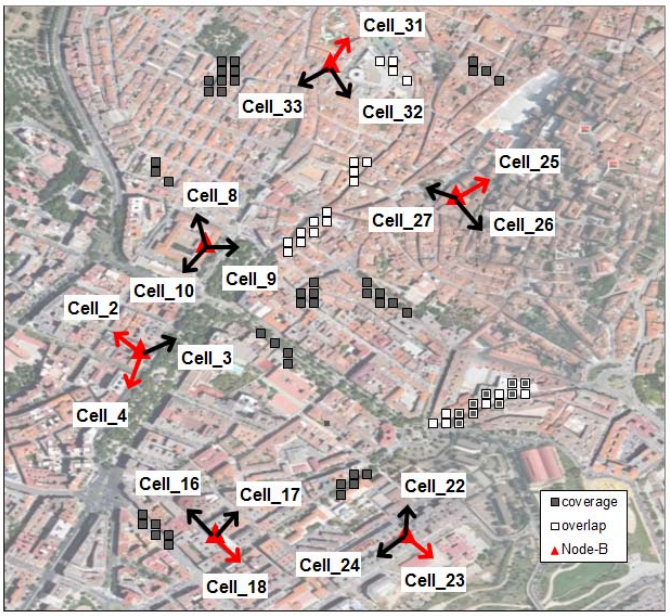

Figure 9.- Location of coverage and overlap problems for the best solution found by the simulated annealing algorithm.

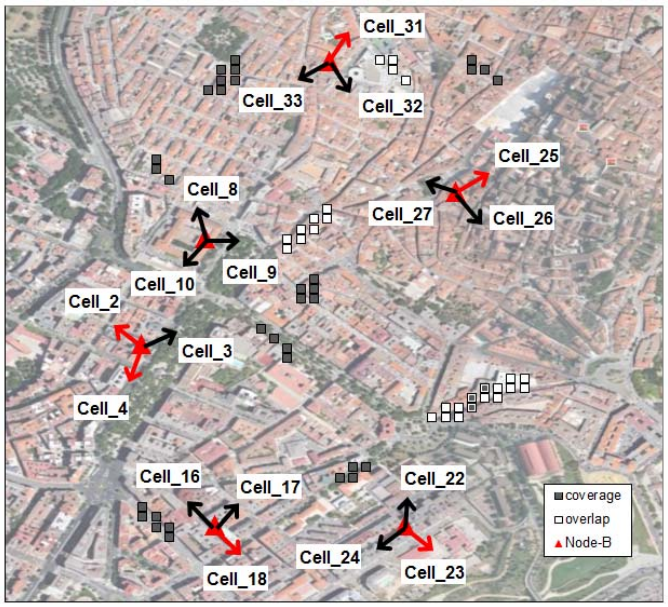

Figure 10.- Location of coverage and overlap problems for the best solution found by the genetic and particle swarm algorithms.

Table 2.- Values of $S_{m,n}$ for the coverage and overlap optimisation targets in the considered cells

	Coverage cost ($S_{1,n}$)			Overlap cost ($S_{2,n}$)		
Cell_id	Initial config.	SA	PS/GA	Initial config.	SA	PS/GA
Cell_3	0.021	0.021	0.021	0.152	0.096	0.046
Cell_8	0.011	0.011	0.011	0	0	0
Cell_9	0.095	0.041	0.023	0	0	0
Cell_10	0	0	0	0	0	0
Cell_16	0	0	0	0	0	0
Cell_17	0	0	0	0.144	0.112	0.112
Cell_22	0.156	0.049	0.020	0	0	0
Cell_24	0.120	0.120	0.093	0	0	0
Cell_26	0.550	0	0	0	0	0
Cell_27	0	0	0	0.035	0.017	0.035
Cell_32	0.014	0.014	0.014	0	0	0
Cell_33	0.051	0.051	0.048	0	0	0
Total	**1.018**	**0.307**	**0.230**	**0.331**	**0.225**	**0.193**

Table 3.- CPICH transmitted power (dBm) of each cell.

Cell_id	Initial configuration	SA	PS/GA
Cell_3	30	28	27
Cell_8	30	31	28
Cell_9	30	33	35
Cell_10	30	26	27
Cell_16	30	34	34
Cell_17	30	31	33
Cell_22	30	35	35
Cell_24	30	31	35
Cell_26	30	29	29
Cell_27	30	28	35
Cell_32	30	31	29
Cell_33	30	29	33

In the following, a comparison between the GA and the PS algorithm is provided in order to illustrate the way how each algorithm performs the solution search along the different iterations. For that purpose, the Euclidean distance between the different solutions evaluated by the algorithm and the final solution found is presented. The Euclidean distance $d(\psi, \psi')$ between two different network configurations ψ and ψ' is defined as:

$$d(\psi,\psi') = \sqrt{\sum_p \sum_n \left(\frac{\psi_{p,n} - \psi'_{p,n}}{\Delta v_p} \right)^2} \qquad (7)$$

This metric represents how similar the network configurations ψ and ψ' are. As an example, $d(\psi,\psi')=0$ indicates that all the tuneable parameters for configuration ψ and ψ' have exactly the same value.

Figure 11 shows the maximum, the minimum and the average value of the Euclidean distance between each of the $N_{POP}=N_{particles}=40$ solutions evaluated in each iteration and the final solution found by both PS and GA algorithms. As shown in Figure 11, the PS algorithm is able to locate its solutions closer to the optimum faster than GA. Note that the optimum solution is found by PS after 11 iterations while GA needs 24 iterations. On the other hand, observe in Figure 11 a higher dispersion of the Euclidean distance for GA than for PS, especially in the first iterations, which illustrates the higher dispersion of the evaluated solutions. This reflects that PS is able to search for solutions in the direction of the optimum faster than GA.

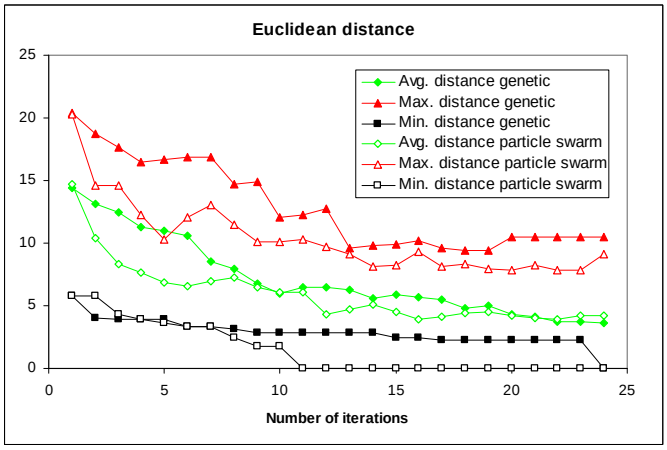

Figure 11.- Statistics of the Euclidean distance.

The following results illustrate how GA and PS algorithms adjust the tuneable configuration parameters of some cells in order to reach the final solution. For the sake of simplicity, the evolution of the pilot power in Cell_3, Cell_9 and Cell_16 corresponding to the best solution found in each iteration is analysed. Figure 12 and Figure 13 plot this evolution for PS and GA, respectively. As an example, it can be seen in Figure 12 that PS does not identify the most adequate value of the pilot power in Cell_16 (i.e. 34dBm) until iteration number 9, but the algorithm has already realized in the second iteration that a higher value of the pilot power is necessary for this cell,

thus reflecting that the performed changes tend to go in the proper direction from the initial iterations. On the contrary, as seen in Figure 13, the best solutions found by GA in the first iterations consider a too low value of the pilot power in Cell_16 (around 28 or 29dBm) and the algorithm needs 18 iterations in order to include the adequate pilot power value of 34dBm for this cell in the best solution found in this iteration.

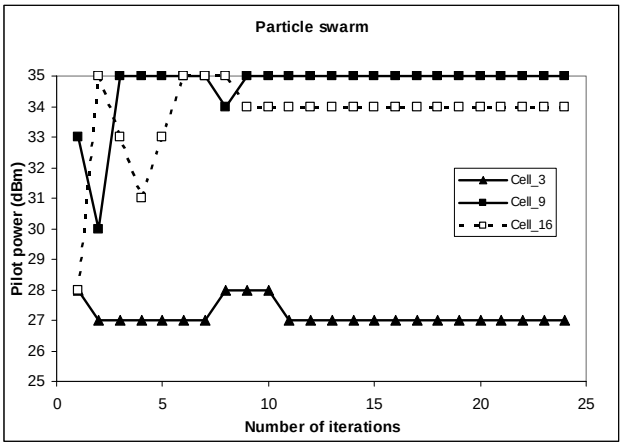

Figure 12.- Evolution of the pilot power for the best solution found with PS.

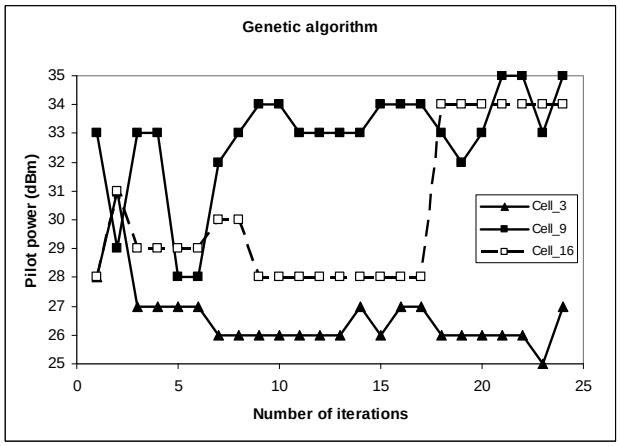

Figure 13.- Evolution of the pilot power for the best solution found with GA.

V. CONCLUSIONS

This paper has presented a general framework for the self-optimisation process in a wireless cellular network. It is composed of two main stages, namely the performance monitoring based on real measurements collected by the network and the optimisation search to identify the most adequate solution by configuring different tuneable network parameters. Different optimisation search methodologies, namely simulated annealing, genetic and particle swarm algorithms have been presented and evaluated. A case study using real data of a UMTS network deployed in a medium-size European city has been presented to evaluate and compare the different optimisation search methodologies in the context of the cell coverage and cell overlap optimisation. Results have revealed that the three methodologies are able to reduce considerably the cell coverage and cell overlap problems with respect to the initial network configuration before running the optimisation process. In turn, both GA and PS algorithms are able to find the same solution that provides a better performance than the one obtained by SA. In terms of speed of convergence, it has been observed that PS finds the final solution faster than GA thanks to a more efficient way of determining new candidate solutions to be evaluated in the subsequent iterations.

ACKNOWLEDGEMENTS

This work is supported by the Spanish Research Council and FEDER funds under ARCO grant (ref. TEC2010-15198). The authors would like to thank Telefónica for providing the measurements.

REFERENCES

[1] O. Sallent, J. Pérez-Romero, J. Sánchez-González, R. Agustí, M.A. Díaz-Guerra, D. Henche, D, Paul, "A Roadmap from UMTS Optimization to LTE Self-Optimization", *IEEE Communications Magazine*, Vol. 49, No.6 , June 2011, pp. 172-182.

[2] INFSO-ICT-216284 SOCRATES project, jan. 2008 - dec. 2010.

[3] NGMN Alliance Deliverable "NGMN Top OPE Recommendations", September 2010.

[4] 3GPP TR 36.902, "Evolved Universal Terrestrial Radio Access Network (E-UTRAN); Self-configuring and self-optimizing network use cases and solutions".

[5] H. Hu, X. Zheng, Y. Yang, P. Wu, "Self-configuration and self-optimization for LTE networks", *IEEE Communications Magazine*, Vol. 48, No.2, February, 2010, pp. 94-100.

[6] C. Y. Lee, H.G. Kang, "Cell planning with capacity expansion in mobile communications: a Tabu Search approach", *IEEE Transactions on Vehicular Technology*, Vol: 49, issue 5, pp 1678-1691, 2000.

[7] A. Molina, G.E. Athanasiadou, A.R. Nix, "The automatic location of base-stations for optimised cellular coverage: a new combinatorial approach", *IEEE Vehicular Technology Conference*, 1999.

[8] F. Gordejuela-Sanchez, A. Juttner, J. Zhang, "A multiobjective optimization framework for IEEE 802.16e network design and performance analysis", *IEEE Journal on Selected Areas in Communication*s, Vol 27, issue 2, pp 202-216, 2009.

[9] E. Aarts, J.H.M. Korst, *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons, 1989.

[10] L. Davis (ed.), *Handbook of genetic algorithms*, International Thomson Computer Press, 1996.

[11] M. Clerc, J. Kennedy, " The particle swarm: Explosion, stability and convergence in a multi-dimensional complex space", *IEEE Transactions on Evolutionary Computation*, vol. 6 pp. 58-73, February 2002.

[12] I. Siomina, P. Varbrand, D. Yuan, "Automated optimization of service coverage and base station antenna configuration in UMTS networks", *IEEE Wireless Communications*, Volume 13, Issue 6, 2006.

[13] L. Shaobo, P. Weijie, Y. Guanci, C. Linna, "Optimization of 3G Wireless Networks Using Genetic Programming", *International Symposium on Computational Intelligence and Desing*, ISCID'09, 2009.

[14] D. Tsilimantos, D. Kaklamani, G. Tsoulos, "Particle swarm optimization for UMTS WCDMA network planning", *International Symposium on Wireless Pervasive Computing* 2008, pp: 283-287, 2008.

[15] O. Sallent, J. Pérez-Romero, J. Sánchez-González, R. Agustí, M.A. Díaz-Guerra, D. Henche, D. Paul, "Automatic Detection of Sub-optimal performance in UMTS Networks based on Drive-test Measurements", *International Conference on Network and Service Management* (CNSM), Paris (France), 2011.

[16] J. Pérez-Romero, O. Sallent, R. Agustí, M. A. Díaz-Guerra, *Radio resource management strategies in UMTS*, John Wiley & Sons, 2005.