This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2014.2363670, IEEE Communications Letters

A Rule-Based Solution Search Methodology for Self-Optimization in Cellular Networks

J. Sánchez-González, J. Pérez-Romero, O. Sallent

***Abstract*—This letter presents a Rule-Based Solution Search (RBSS) algorithm for self-optimization in a cellular network. It improves the solution search process of classical optimization methodologies by including a set of rules that capture the knowledge of how a given problem can be solved. The results obtained in a scenario using measurements from a real network reveal that including the RBSS algorithm achieves reductions of up to 60% and 90% in the convergence time of Particle Swarm and Genetic Algorithms, respectively.**

***Index Terms*—Self-optimization, Self-Organizing Networks (SON), cellular networks, cell coverage, cell overlap.**

I. INTRODUCTION

THE introduction of new wireless communication technologies and services has increased the complexity of wireless networks in recent years. The solution to address this complexity is the automation of many network procedures with the objective of creating an autonomously managed network. In this context, the concept of Self-Organizing Networks (SON) is seen as a way to reduce both operational and capital expenditures. For these reasons, in the last few years, there has been intense research activity in this field [1]-[4]. The self-organization of cellular networks includes self-configuration, self-optimization and self-healing. The self-optimization process is in charge of automatically finding the most appropriate values of the network configuration parameters to optimize the network performance in terms of specific performance targets. Due to the large number of network parameters that may be tuned in a large cellular network and the existence of coupling effects among different cells, the use of automatic optimization methodologies becomes fundamental because it is very hard for an engineer to cope manually with the associated complexity. In this context, several strategies have been proposed in the literature [5]-[9].

Optimization strategies are usually based on an iterative process in which new solutions are proposed and evaluated to find better solutions as the number of iterations increases. One of the critical aspects is the so-called *solution search* process that determines how to generate the new solutions to be evaluated at each new iteration. This process is usually based on making random changes to the current solution(s) to obtain the new solution(s) to be evaluated, as in algorithms such as Simulated Annealing (SA) [7], Genetic Algorithm (GA) [8] and Particle Swarm (PS) [9]. Several works propose the inclusion of specific functionalities that guide the optimization algorithm in the direction of better solutions to speed up the algorithm convergence, mainly by reducing the degree of randomness in the exploration of the solution search space, which tends to slow down the convergence [10]. In [10], a quadratic approximation of the objective function is used to perform deterministic mutation in evolutionary algorithms. However, this approximation may not be valid in many problems where the objective function cannot easily be approximated to a known function, as in the optimization problem in this letter. In [11], a guided mutation process based on a probability model is proposed to assist a genetic algorithm under the assumption that the variables in the search space take binary values. The main drawback is the excessive computational cost.

Within this context, this work proposes a new algorithm called Rule Based Solution Search (RBSS) that allows the optimization methodology to move toward better solutions inside the search space, thus increasing the algorithm convergence. The RBSS algorithm uses rules based on the knowledge of how each specific problem may be solved. To the best of the authors' knowledge, this approach is new in the field of cellular networks optimisation. The proposed RBSS approach is integrated with the general self-optimization framework presented in [12] and particularized for the joint optimization of cell coverage and overlap. The benefits provided by RBSS are evaluated using measurements from a real network. The rest of the letter is organized as follows. Sections II and III present the self-optimization framework and the RBSS algorithm, respectively. Section IV particularizes it for the optimization of cell coverage and overlap. Section V presents the performance assessment. Section VI summarizes the conclusions.

II. RBSS SELF-OPTIMIZATION FRAMEWORK

Let us consider a general cellular network that consists of N cells with P tunable parameters per cell. The network configuration is represented by a $P \times N$ matrix $\psi = [\psi_{p,n}]$, where $\psi_{p,n}$ is the value of the p-th tunable parameter of the n-th cell. The set of possible values of $\psi_{p,n}$ is the range $[V_{min,p}, V_{max,p}]$ with resolution Δ_p. The self-optimization procedure consists of a continuous loop that interacts with the real network based on *observations* and *actions* [1], as illustrated in Figure 1. At the observation phase, certain measurements are collected from the different cells. The Network Performance Monitoring (NPM) process analyses these measurements to detect situations where some of the M optimization targets specified by the operator are not properly fulfilled. The result of the NPM will be the $M \times N$ performance matrix $S(\psi) = [S_{m,n}(\psi)]$. The term $S_{m,n}(\psi)$ ($0 \leq S_{m,n}(\psi) \leq 1$) reflects the performance of the m-th optimization target in the n-th cell with the current configuration ψ. An excessively high value of $S_{m,n}(\psi)$ reflects that the m-th target is not sufficiently optimized in the n-th cell. Based on the elements of $S(\psi)$, a trigger condition will be evaluated to decide whether the performance is satisfactory

Manuscript received July 28, 2014; accepted October 5, 2014. This work has been supported by the Spanish Research Council and FEDER funds under ARCO (ref. TEC2010-15198) and RAMSES (ref. TEC2013-41698-R) grants.

The authors are with the Universitat Politècnica de Catalunya (UPC). C/ Jordi Girona, 1-3, Campus Nord D4, Barcelona, Spain, 08034 (e-mail: juansanchez@tsc.upc.edu)

1089-7798 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2014.2363670, IEEE Communications Letters

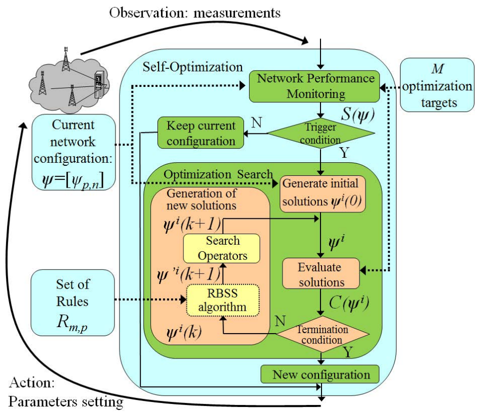

Fig. 1. Network optimization loop.

or if the network needs further optimization. In the latter case, the Optimization Search is triggered to find suitable values of ψ that optimize the network performance given by $S(\psi)$.

The Optimization Search is a multi-objective problem, as it involves M optimization targets, which in general can be partly contradictory. Therefore, some trade-off criterion among targets must be specified by the network operator. A usual approach is to define a joint cost function $C(\psi)$ that combines the contribution of the different targets by means of weights $\beta_m, m = 1, ..., M$, that make it possible to give more relevance to certain targets than others. The optimal solution is thus given by the configuration matrix ψ^* that minimizes $C(\psi)$:

$$\psi^* = \arg\min_{\psi} C(\psi) = \arg\min_{\psi} \left(\sum_{m=1}^{M} \beta_m \sum_{n=1}^{N} S_{m,n}(\psi) \right) \quad (1)$$

The Optimization Search process requires smart methodologies that efficiently explore the search space while maintaining low computational complexity. This letter assumes an iterative process in which new candidate solutions $\psi^i(k)$ are generated and evaluated in each iteration k, as seen in Figure 1. The process begins by initializing a set of N_{POP} candidate solutions $\psi^i(0)$ $i = 1, ..., N_{POP}$. One of them is the current network configuration, and the others are determined randomly. Each solution $\psi^i(0)$ is evaluated in a similar way as in the NPM process, estimating the matrix $S(\psi^i)$ and the cost $C(\psi^i)$ that would exist in the network if this solution was applied.

In iteration $k + 1$, the algorithm generates new N_{POP} solutions $\psi^i(k + 1)$ based on the solutions of the previous iteration $\psi^i(k)$ in two steps, as shown in Figure 1. First, the proposed RBSS algorithm explained in Section III is executed to obtain an intermediate set of new solutions denoted as $\psi'^i(k + 1)$ that are generated using deterministic rules based on a priori knowledge of how a change in a given parameter affects each optimization target. In this way, the new solutions will be directed to improve the performance for the different targets. Then, at the second stage, search operators based on classical optimization methodologies (e.g., GA, PS) are applied over the outputs $\psi'^i(k + 1)$ provided by the RBSS to obtain the solutions $\psi^i(k + 1)$ to be evaluated in the next iteration. The specific search operators depend on the optimization methodology (e.g., for the conventional GA, selection, mutation and recombination [8]). This second stage introduces the required randomness to explore the search space starting from a set of solutions that has been smartly selected by the RBSS. The joint operation of RBSS and search operators will lead a more efficient generation of the new solutions in each iteration than if only the search operators of conventional techniques were used. The new solutions $\psi^i(k+1)$ are evaluated, and the best solution found among the solutions analyzed in all the iterations is retained. The process is repeated until a termination condition is fulfilled, i.e., the cost of the best solution found is below a threshold, or until reaching a maximum number of iterations. The best solution found specifies the new values of the configuration parameters.

III. RBSS ALGORITHM

The RBSS takes as input a solution ψ^i and smartly modifies its parameters following a set of rules. A rule $R_{m,p}(\cdot)$ is a function that specifies how to modify the parameter p of a particular cell to improve the performance of the m target. In detail, for each candidate solution ψ^i, one parameter is modified to generate an intermediate solution ψ'^i as follows:

1) Randomly select a cell n and an optimization target m to be improved in this cell with probability:

$$P_{m,n} = \frac{S_{m,n}(\psi^i(k))}{\sum_{n=1}^{N} \sum_{m=1}^{M} S_{m,n}(\psi^i(k))} \quad (2)$$

Note that, according to (2), targets and cells with higher values of $S_{m,n}(\psi^i)$ are chosen with higher probability.

2) Randomly select a tunable parameter p to be adjusted. All parameters are selected with equal probability.

3) Randomly select the cell n^* (where the parameter p will be modified) with equal probability among the cell n and any other cell that has influence on the target m of cell n.

4) The new value of the parameter p in cell n^* is obtained by applying rule $R_{m,p}(\cdot)$, that is $\psi'^i_{p,n^*} = R_{m,p}(\psi^i_{p,n^*})$

The rules $R_{m,p}(\cdot)$ depend on the optimization targets and the parameters to be set, as explained in section IV. Note that the RBSS algorithm includes both random components (steps 1, 2 and 3) and deterministic rules (step 4). Random components allow consideration of the multiple possible options when deciding which parameters and cells should be modified to obtain improved solutions. In turn, the deterministic rules ensure that the modification of the selected parameter will make the new solution better than the previous one.

IV. RBSS FOR CELL COVERAGE/OVERLAP OPTIMIZATION

This section presents a particularization of the RBSS algorithm for optimizing cell coverage and overlap in a UMTS (Universal Mobile Telecommunications System) network. Cell coverage optimization should guarantee that communication is possible in the planned service area of a cell by avoiding coverage holes. Cell overlap optimization aims to avoid areas where access to the network is possible through too many cells, which may generate excessive interference and soft handover overheads [13]. Details on the measurements and the NPM procedure can be found in [12], while this section presents details of the RBSS. The $P = 3$ tunable parameters are the CPICH (Common Pilot Channel) transmitted power $\psi_{1,n}$,

1089-7798 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/LCOMM.2014.2363670, IEEE Communications Letters

TABLE I
CONSIDERED RULES FOR CELL COVERAGE AND CELL OVERLAP OPTIMIZATION.

	Cell Coverage optimization $(m = 1)$	Cell Overlap optimization $(m = 2)$
Tx. power $(p = 1)$	$R_{1,1}$: $\psi'_{1,n^*} = min(\psi_{1,n^*} + \Delta_1, V_{max,1})$	$R_{2,1}$: $\psi'_{1,n^*} = \begin{cases} \min(\psi_{1,n^*} + \Delta_1, V_{max,1}) & \text{if } n^* = v \\ \max(\psi_{1,n^*} - \Delta_1, V_{min,1}) & \text{if } n^* = n \end{cases}$
Tilt $(p = 2)$	$R_{1,2}$: $\psi'_{2,n^*} = \begin{cases} \max(\psi_{2,n^*} - \Delta_2, V_{min,2}) & \text{if } \psi_{2,n^*} > \gamma_{n^*} \\ \min(\psi_{2,n^*} + \Delta_2, V_{max,2}) & \text{if } \psi_{2,n^*} < \gamma_{n^*} \end{cases}$	$R_{2,2}$: $\psi'_{2,n^*} = \begin{cases} \max(\psi_{2,n^*} - \Delta_2, V_{min,2}) & \text{if } n^* = v \text{ and } \psi_{2,n^*} > \gamma_{n^*} \\ & \text{or } n^* = n \text{ and } \psi_{2,n^*} < \gamma_{n^*} \\ \min(\psi_{2,n^*} + \Delta_2, V_{max,2}) & \text{if } n^* = n \text{ and } \psi_{2,n^*} > \gamma_{n^*} \\ & \text{or } n^* = v \text{ and } \psi_{2,n^*} < \gamma_{n^*} \end{cases}$
Azimuth $(p = 3)$	$R_{1,3}$: $\psi'_{3,n^*} = \begin{cases} \max(\psi_{3,n^*} - \Delta_3, V_{min,3}) & \text{if } \psi_{3,n^*} > \mu_{n^*} \\ \min(\psi_{3,n^*} + \Delta_3, V_{max,3}) & \text{if } \psi_{3,n^*} < \mu_{n^*} \end{cases}$	$R_{2,3}$: $\psi'_{3,n^*} = \begin{cases} \max(\psi_{3,n^*} - \Delta_3, V_{min,3}) & \text{if } n^* = v \text{ and } \psi_{3,n^*} > \mu_{n^*} \\ & \text{or } n^* = n \text{ and } \psi_{3,n^*} < \mu_{n^*} \\ \min(\psi_{3,n^*} + \Delta_3, V_{max,3}) & \text{if } n^* = n \text{ and } \psi_{3,n^*} > \mu_{n^*} \\ & \text{or } n^* = v \text{ and } \psi_{3,n^*} < \mu_{n^*} \end{cases}$

the antenna tilt $\psi_{2,n}$ and the antenna azimuth $\psi_{3,n}$. They are the most common tunable parameters considered in the literature for cell coverage and overlap optimization [2][5][13]. According to the RBSS algorithm of Section III, and after selecting the cell n, the optimization target m and the tunable parameter p in steps 1 and 2, the cell n^* to be adjusted in step 3 is selected depending on the target m:

- *Cell coverage optimization* $(m = 1)$: The cell n^* is selected randomly with equal probability among the set $\{n, neigh(n)\}$, where $neigh(n)$ is the list of neighbor cells of cell n consisting of first ring of cells that surround cell n and have their antenna pointing in the direction of cell n.

- *Cell overlap optimization* $(m = 2)$: The cell n^* is selected randomly with equal probability among the set $\{n, v\}$ where v is the cell suffering the overlap generated by the n-th cell.

Table 1 presents the rules applied in step 4 over the selected cell n^*. In the case of a CPICH transmitted power change $(p = 1)$, rule $R_{1,1}$ increases by Δ_1 dB the CPICH transmitted power of the selected cell n^*. In turn, rule $R_{2,1}$ either increases by Δ_1 dB the CPICH transmitted power of cell $n^* = v$ that suffers the overlap or decreases by Δ_1 dB the CPICH power of the cell $n^* = n$ that generates the overlap. When adjusting the antenna tilt $(p = 2)$ or the antenna azimuth $(p = 3)$, the rules $R_{m,p}$ must consider that the effect of this change is not the same at all geographical locations (e.g., an increase in the antenna tilt of a cell may improve the cell coverage near the cell, but it may degrade the coverage at the cell edge). For this purpose, the NPM procedure first identifies the set of geographical regions where there are coverage $(m = 1)$ or overlap $(m = 2)$ problems associated with cell n. Then, the RBSS algorithm randomly selects one of these regions in such a way that regions with higher contributions to $S_{m,n}(\psi^i)$ are selected with higher probability. The selected region is characterized by the geographical position Ω of its centroid. Then, the rules applied by RBSS will tend to improve the coverage or overlap conditions at position Ω by computing as a reference the tilt γ_{n^*} and azimuth μ_{n^*} angles corresponding to the direction between cell n^* and position Ω (see Figure 2). Then, for the coverage case, rules $R_{1,2}$ and $R_{1,3}$ (see Table 1) adjust the antenna tilt/azimuth of cell n^* to increase the antenna gain in the direction γ_{n^*}, μ_{n^*} of position Ω. Specifically, for rule $R_{1,2}$, if the current tilt ψ_{2,n^*} is higher than γ_{n^*} (see Figure 2a), the rule decreases the tilt by Δ_2 degrees. Otherwise, the rule increases the tilt. A similar decision is made for the azimuth ψ_{3,n^*} and angle μ_{n^*} for rule

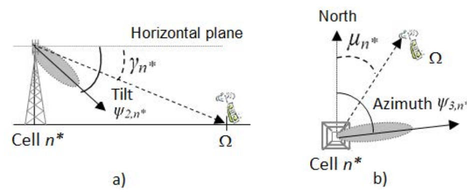

Fig. 2. Adjustment of antenna tilt (a) and azimuth (b).

$R_{1,3}$. In turn, rules $R_{2,2}$ and $R_{2,3}$ make a distinction depending on whether cell n^* is the one generating the overlap (i.e., $n^* = n$) or the one suffering it (i.e., $n^* = v$). In the first case, the tilt/azimuth is adjusted to reduce the antenna gain in the direction γ_{n^*}, μ_{n^*} of position Ω. In the second case, the adjustment is made to increase the antenna gain.

V. PERFORMANCE EVALUATION

The proposed RBSS methodology is evaluated using drive test measurements obtained from a real network in a medium-size European city. Figure 3 plots the location of the available drive test measurements and the regions with cell coverage and overlap problems detected by the NPM before applying the optimization procedure. The self-optimization algorithm adjusts the parameters of the $N = 12$ cells marked with black arrows. This number of cells allows considering sufficient neighboring cells to draw conclusions about how the cell coverage and overlap problems can be mitigated. The CPICH transmitted power $\psi_{1,n}$ varies in the range [25,35] dBm with resolution $\Delta_1 = 1$ dB, the antenna tilt $\psi_{2,n}$ in the range $[0°, 10°]$ in steps of $\Delta_2 = 1°$ and the antenna azimuth $\psi_{3,n}$ in the range $[a - 25°, a + 25°]$ in steps of $\Delta_3 = 5°$, where a is the current antenna azimuth existing in the network.

Evaluation is performed for different Optimization Problems (OP) defined in Table 2 depending on the number of selected targets and tunable parameters. The RBSS operating jointly with GA or PS is compared against the conventional GA and PS schemes without RBSS support. The parameters of PS and GA are the same as in [14]. Performance is evaluated in terms of the number of iterations required by each methodology to converge to a quasi-optimal solution (see Table 2). As the optimal solution cannot be known a priori in this complex scenario (see e.g., in Table 2 the huge number of possible combinations N_C for each OP), we assume convergence when a methodology finds a solution with a cost lower than $C_{best}(\psi^*) + 5\%$, where $C_{best}(\psi^*)$ is the cost of the best solution found for a given OP after an initial and extensive run of all the methodologies considered here. Table 2 reveals that

1089-7798 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.

TABLE II
NUMBER OF ITERATIONS TO REACH CONVERGENCE FOR EACH OP

Considered Methodology	OP1: Cell coverage only Tuned Parameter: $\psi_{1,n}$ $N_C = 11^{12} \approx 3.13 \cdot 10^{12}$	OP2: Cell coverage and overlap Tuned Parameter: $\psi_{1,n}$ $N_C = 11^{12} \approx 3.13 \cdot 10^{12}$	OP3: Cell coverage and overlap Tuned Parameters: $\psi_{1,n}$,$\psi_{2,n}$ $N_C = 11^{24} \approx 9.84 \cdot 10^{24}$	OP4: Cell coverage and overlap Tuned Parameters: $\psi_{1,n}$,$\psi_{2,n}$,$\psi_{3,n}$ $N_C = 11^{36} \approx 3.09 \cdot 10^{37}$
RBSS-PS	5 [37.50%]	7 [36.36%]	30 [56.52%]	39 [60.20%]
Conventional PS	8	11	69	98
RBSS-GA	25 [77.27%]	58 [87.47%]	66 [89.47%]	195 [90.89%]
Conventional GA	110	463	627	2141

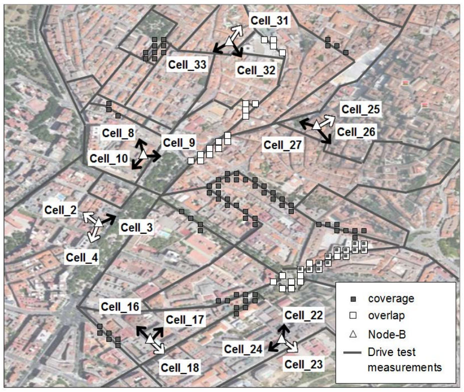

Fig. 3. Location of problems with the initial configuration.

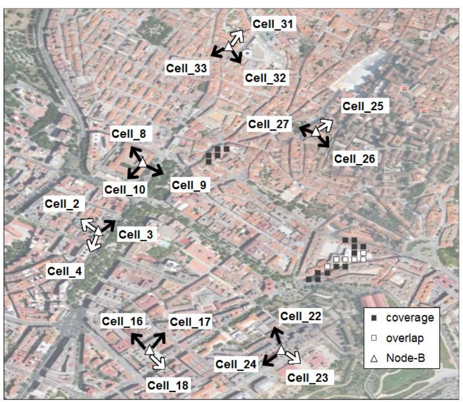

Fig. 4. Location of problems for the RBSS-PS.

RBSS reduces the convergence time very significantly with respect to the conventional PS and GA. The reductions (shown in brackets in Table 2) are higher for GA than for PS. It is also remarkable that the reduction produced by RBSS is much higher when considering more complex problems that include more tunable parameters and thus the search space contains more possible combinations N_C. With higher number of tuned parameters, the search space becomes larger, so it is more relevant to have smart mechanisms, such as RBSS, that narrow the search by quickly identifying the best combinations. The results also show that PS provides faster convergence than GA. Note also that the reduction in the number of iterations produced by RBSS has a direct impact on the reduction of the computational complexity which is mainly limited by the solution evaluation stage, while the additional computational cost incurred by the four simple steps of the RBSS algorithm is marginal. Figure 4 shows the location of the coverage and overlap problems after applying the RBSS-PS for OP4. The solution found by RBSS-PS achieves significant improvements with respect to the initial configuration before running the optimization algorithm (see Figures 3 and 4). Almost all the detected problems are removed, except certain coverage problems in Cell_9 and Cell_22 and some overlap generated by Cell_17 over Cell_22, which are nonetheless significantly reduced. The cost of the initial configuration is 1.349, while the cost of RBSS-PS is 0.083, so the cost reduction is 93.84% that represents a reduction of 82.25% in the number of geographical samples where coverage and/or overlap problems are identified. Although, due to the nature of the GA and PS algorithms, it cannot be proved that the global optimum is found, the results reflect that the solution is good in practice.

VI. CONCLUSION

This letter proposes a new RBSS self-optimization framework for cellular networks that improves the speed of convergence of the optimization search process by including in the solution search process a set of rules that reflect a priori knowledge about how a configuration parameter affects a given optimization target. This framework has been particularized for optimizing cell coverage and overlap and has been evaluated using measurements obtained from a real network. The results reveal that the inclusion of the RBSS algorithm reduces the convergence speed of conventional PS and GA by up to 60% and 90%, respectively. This reduction is more remarkable in optimization problems with large solution search spaces.

REFERENCES

[1] O. Sallent, J. Pérez-Romero, J. Sánchez-González, R. Agustí, ”A Roadmap from UMTS Optimization to LTE Self-Optimization”, *IEEE Comm. Magazine*, Vol. 49, No.6, June 2011, pp. 172-182.
[2] J. Ramiro, K. Hamied, *Self-Organizing Networks: Self-Planning, Self-Optimization and Self-Healing for GSM, UMTS and LTE*, Wiley, 2012.
[3] NGMN Deliverable ”NGMN Top OPE Recommendations”, Sept 2010.
[4] 3GPP TR 36.902, v9.3.1 (2011-03).
[5] 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.
[6] F. Gordejuela-Sanchez, A. Juttner, J. Zhang, ”A multiobjective optimization framework for IEEE 802.16e network design and performance analysis”, *IEEE JSAC*, Vol 27, issue 2, pp 202-216, 2009.
[7] E. Aarts, J.H.M. Korst, *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons, 1989.
[8] D. E. Goldbert, *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley, 1989.
[9] 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.
[10] A. Kanarachos, D. Koulocheris, ”Evolutionary algorithms with deterministic mutation operators used for the optimisation of the trajectory of a four-bar mechanism”, *Journal of Mathematics and Computers in Simulation*, Vol. 63, issue 6, pp: 483-492, November 2003.
[11] Q. Zhang, J. Sun, E. Tsang, ”An evolutionary algorithm with guided mutation for the maximum clique problem”, *IEEE Transactions on Evolutionary Computation*, Vol. 9, No. 2, pp: 192-200, April, 2005.
[12] J. Sánchez-González, O. Sallent, J. Pérez-Romero, R. Agustí, ”A multi-cell multi-objective self-optimisation methodology based on genetic algorithms for wireless cellular networks”, *International Journal of Network Managment*, Vol. 23, issue 4, pp: 287-307, July 2013.
[13] C. Chevalier, C. Brunner, et al., *WCDMA Deployment Handbook: Planning and Optimization Aspects*, John Wiley & Sons, 2006.
[14] J. Sánchez-González, O. Sallent, J. Pérez-Romero, R. Agustí, ”A comparison of different optimisation search methodologies for self-optimisation in wireless cellular networks”, *IFIP Wireless and Mobile Networking Conference*, Dubai, April 2013.

1089-7798 (c) 2013 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.