Automatic equivalent model generation and evolution for small cell networks

Davide Cherubini*, Michele Portolan†

*Bell Laboratories, Alcatel-Lucent Ireland, Dublin, Ireland
†Bell Laboratories, Alcatel-Lucent France, Nozay, France
{*name.surname*}@alcatel-lucent.com

***Abstract*—Automated Self-optimizing Networks (SON) algorithms have been proposed to address and solve the issues related to optimization in small cell networks. However, automatic optimization approaches require precise knowledge of the deployment environment and users behaviors. This information is generally difficult, expensive to obtain and presents significant computational requirements. In this paper we introduce a method that, based on available measurements, enables the automatic generation of an abstract equivalent model and its adaptation to the environment in which the network is deployed. This model can be a key component to mitigate the computational burden and to speed up the convergence of self-learning and self-evolving coverage optimization algorithms.**

***Keywords*—Design Optimization**

I. Introduction

Microcell, picocell, femtocell and lately attocell. There is a current trend in mobile operators to extend the existing cellular systems with targeted cellular deployments commonly referred to as small cell networks. Recently, significant research efforts have focused on the area of Self-optimizing Networks (SON) to resolve issues of traditional manual optimization in small cell networks [1][2] [3].

Manual optimization, in fact, is time and cost expensive and it is not compatible with the "plug-and-play" business model used in small cell networks. As mentioned a possible solution is to use pre-defined SON algorithms designed for general cases. However these approaches may produce sub-optimal solutions due to the high variability of the scenarios, particularly if implemented in a distributed fashion. Automated algorithm design approaches, where the self-management algorithm itself is adapted to the actual environment, can solve these issues. One approach for automatic algorithm design is genetic programming (GP). For instance, GP has been proven to be an effective method to automatically create algorithms for distributed coverage optimization [4].

A schematic representation of such process is presented in Figure 1.

Generally, GP evolution steps can be summarized as follows:

1) Initialize the population with random algorithms.
2) Evaluate the performance of each program in the population using a given fitness function.

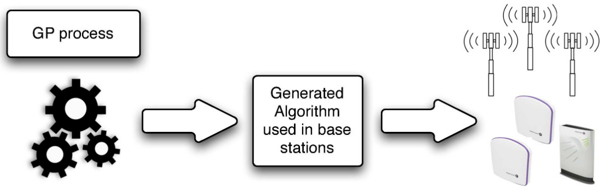

Fig. 1: Overview of application of GP process

3) Populate the next generation with offspring by applying mutation and crossover operators on individuals with high fitness.
4) Repeat steps 2 and 3 until stop conditions are met.

The evaluation of point 2 implies the simulation of the scenario over an important period, in order to take into account the impact of the moving users and varying traffic demands. A major limitation of this approach is the *environment model*: in order to obtain optimal results for a specific deployment scenario, a high accuracy degree of the environment is required. Such information is difficult and time expensive to collect. For instance, in order to obtain an estimate of the coverage overlap of neighboring cells, in addition to the base station configuration, one needs:

- a precise map of the area;
- the knowledge of all the construction materials and relative propagation behavior;
- the accurate wave propagation model.

Furthermore, user mobility is another determining factor to be taken into account during the modeling process: the same geographical scenario (for instance, a building) can have different optimal solutions depending on the number of users and of their mobility patterns.

Anyway, GP has a peculiarity: at each evolution step the approach is not really interested at the absolute value of the fitness function F, but rather at its gradient, i.e. the difference between different fitness values. Among the mutation set $\mathcal{M}$, the individual x_{new} with the highest fitness value is chosen as the new candidate::

$$x_{new} = \left\{x_i \in \mathcal{M} : F(x_i) = \max_{x \in \mathcal{M}}[F(x_1), F(x_2), \cdots]\right\} \quad (1)$$

This observation is the key motivator for this paper: we propose an abstract model that has no pretension to give a precise estimation for the absolute value of the fitness, but only focuses on modeling the fitness gradient in order to guide the GP evolution in the right direction with minimal computational requirements.

II. State of the Art

The common scheme used in all adaptive approaches (e.g., incremental, heuristic, evolutionary, machine learning) is to compute the value of a given objective function, which estimates the goodness of a provided solution in terms of effectiveness of the new adaptation step. The objective functions are usually computed by an analysis of the scenario, both the physical and mobility aspects are taken into account, so the precision of the model is of paramount importance. The usual adopted solution is to develop accurate models of environment and users behaviors, which can reliably replicate reality [5][6]. This is generally done through two main elements:

- physical model;
- mobility model.

The *physical model* involves analysis and tracking of wave propagation in the environment in order to understand signal characteristics (e.g., integrity, interference, Signal-to-Noise Ratio). This requires the understanding of the physics involved, of the radio signaling protocol and of the RF equipment. Moreover, and most importantly, a deep knowledge of the environment is required: the exact position of all obstacles (e.g., walls, trees, cars), their electromagnetic characteristics (e.g., interference, reflection, attenuation, dielectric constants), the exact position of all signal sources, an so on and so forth. Similarly, the *mobility model* concerns modeling and tracking users moving into the environment and interacting with the base stations. It requires the knowledge of the exact position and behavior of each user, which can rise up to hundreds or thousands in an outdoor scenario. All this wealth of information is the main problem of such approach:

1) it is really difficult to obtain;
2) it is time and economically expensive to realize;
3) it makes simulation complexity huge;
4) it is intrinsically centralized.

Point 1 and 2 are especially problematic for the small cell deployments business model, which is based on “plug-and-play” of off-the-shelf components by the final customer, with little or no intervention from the operator. Moreover, Points 3 and 4 go against the solution given to the previous two points (1 and 2), which is based on autonomous adaptation of the cells after their deployment. This is not feasible if the required computational resources are too big (it would have too big an impact on the final cost), and if centralization requires specific additions to the infrastructure, once more with a negative impact on costs.

III. Abstract Graph Representation

The key point of the adaptive approaches presented in the previous section is to compare the performances (i.e. *fitness*) of different solutions in order to identify the best one. To reach this goal, precise simulations are done to obtain a reliable estimation of the absolute value of fitness of each solution, which is then used as the comparison metric.

In the present paper we propose a radically different approach: to reduce the overall complexity of the problem by developing an abstract equivalent model based on the concept of *equivalent distance*. This concept, explained in details in the following section, estimates the distance two base stations should be at in an ideal free-space propagation environment to observe the same signal attenuation they have in reality. This value can be easily calculated from direct measurements obtained during the normal operations of small cells in a real environment. Free-space propagation is extremely simple, and most problems are reduced to simple Euclidean geometry. More formally, the use of an equivalent distance allows the representation of the real scenario by mean of a directed graph $\mathcal{G}(N, A)$ composed of N nodes and A arcs.
Each node $i \in N$ represents a small cell and is defined by one or more of its global attributes, such as:

- name ID;
- real space coordinates;
- equivalent space coordinates;
- forward and backward stars;

and by the combination of one or more problem related (local context) attributes, like for example:

- number of connected users;
- number of arriving/leaving users;
- transmission power;
- number of detected drop calls;
- traffic characteristics (e.g., number of handled calls, amount of traffic).

One or more directed arcs, or links, can connect the nodes. Each arc $(i, j) \in A$ is defined by its global attributes, such as:

- name ID;
- tail end node i ;
- head end node j ;
- equivalent distance d_e ;
- neighbors list;

and by mean of local context attributes, like for example:

- real distance d (if available);
- number of users “moving” along the arc;
- overlapping area along the direction defined by the arc.

The flexibility of such simple graph representation further allows the definition of additional customized global and local context attributes for the nodes and the arcs depending on the problem at hand.

IV. Creating The Equivalent Model

The graph representation described above can be used to model a variety of parameters such as cells area overlap or user mobility characteristics as required for the automatic evolution of self-X algorithms. This allows on the one hand a bigger number of iterations (therefore the error introduced by the simplification can be reduced) and on the other hand leaves enough computational resources to evolve the model itself, so that over time it becomes more precise.

A. The Equivalent Distance

We propose the following technique to generate the equivalent distance: during the start-up process a small cell, identified by mean of a particular identifier (e.g., the cell ID), emits a pulse at maximum transmission power for a predefined time t_{init}. A neighbor cell sensing the pulse measures the received power and, using the free-space path loss equation, calculates the distance that would be between the two base stations to have the same attenuation in an ideal free space propagation scenario (i.e., the equivalent distance), as detailed in the example of paragraph IV-B.

Such value can be internally stored, exchanged within the small cell network, or reported back to a given central optimization system. The same procedure can be applied to calculate the equivalent distance between a small cell and a mobile phone or the one existing for a pair of mobile phones. In general the value of the equivalent distance can be evaluated when a pair of transmitting/receiving entities exists.

B. Overlap Estimation

In this paragraph the equivalent model is used to estimate the overlapping area of two neighboring cells. The overlapping area (or *overlap*) is a term used to evaluate the objective function in the coverage optimization process [4]. However, the estimated coverage overlap requires burdensome computation over a highly precise model. In a traditional simulation scenario, in fact, the simulator must model the propagation patterns of each cell, and identify the areas where signal interference is higher than a given threshold. With our equivalent model approach, cell interference on a real deployment is modeled using the free space propagation laws. In such an ideal scenario, as depicted in Figure 2, for a given cell *A* transmitting a pulse of power P_A at frequency f, the power P_{AB} received at a given point *B* is inversely proportional to the square of the distance d [7]:

$$P_{AB} = k^2 \frac{P_A}{d^2} \tag{2}$$

where $k = c_0/(4\pi f)$, with c_0 being the speed of light in vacuum.
Once the minimum power threshold P_{TH} to achieve coverage is defined, the points where interference with *A* is possible are therefore contained in a circle of ray r and centered in *A*, as depicted in Figure 2. The ray r is defined as:

$$r = k\sqrt{\frac{P_A}{P_{TH}}} \tag{3}$$

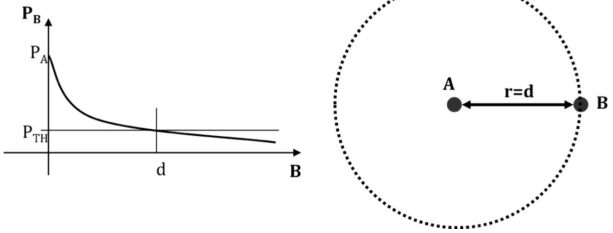

Fig. 2: Free space propagation

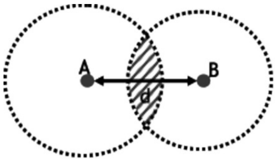

Fig. 3: Overlap in free space propagation

The computation of overlap between two cells is therefore reduced to a mere geometrical problem, as depicted in Figure 3. If $d \leq (r_A + r_B)$, the overlap area (shaded in Figure 3 can be computed based only on the transmit powers P_A, P_B, the threshold P_{TH} and the distance d. [8]

It also means that for a given cell *A* transmitting at power P_A and for every point *B* closer than r, it is possible to compute their mutual distance $d(A, B)$ just from the received power P_{AB}. Of course, the real interference area is not a perfect circle due to the presence of obstacles, multi-path effects, etc. Figure 4 roughly represents such scenario: the wall between *A* and *B* modifies the attenuation factor, and the overlap area (shaded) is smaller than it would be in free space. As said, the propagation pattern of Figure 4 is only indicative and in reality it is much more complicated. The equivalent model can be built by computing the equivalent distance d_e for each couple of cell. Its formal definition can be the following:

- let *A* be a cell transmitting at power P_A;
- let *B* be a second cell, receiving a signal with power P_{AB} from *A*;
- let $d(A, B)$ be the real distance between cells *A* and *B*;

then the equivalent distance $d_e(A, B)$ can be defined as the distance *A* and *B* would be in free space, so that the transmitted power P_A gets attenuated to P_{AB}:

$$d_e(A, B) = k\sqrt{\frac{P_A}{P_{AB}}} \tag{4}$$

Figure 5 shows a graphical representation of $d_e(A, B)$. When the equivalent distance is computed for both transmitting cells, the *equivalent* overlapping area can be defined, as shown in Figure 6. This area can be easily obtained through Euclidean geometry [8], making the computation much lighter.
Moreover, the computation of equivalent distance only requires the knowledge of transmitted and received powers, two values

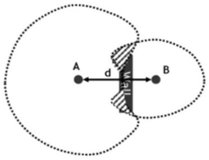

Fig. 4: Overlap in presence of an obstacle

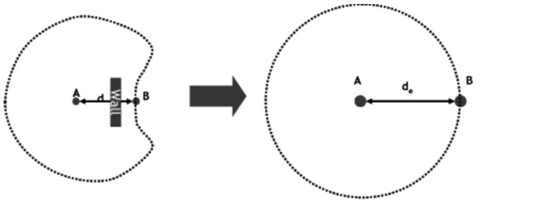

Fig. 5: Definition of equivalent distance

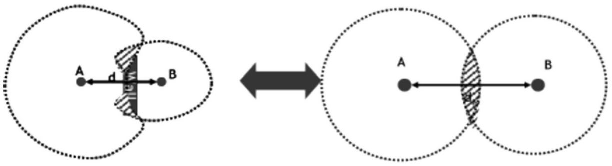

Fig. 6: Definition of equivalent overlap area

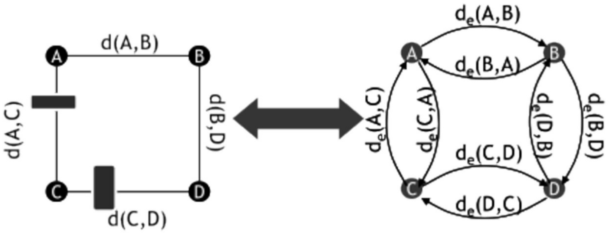

Fig. 7: Equivalent model generation

easily obtained through direct measurements. The equivalent model has two important properties:

- it may be NOT symmetric because the propagation is not always symmetrical, i.e. $d_e(A,B)$ can be different from $d_e(B,A)$;
- it does NOT conserve topology.

For instance, if nodes *A*, *B* and *C* form a triangle with sides $\overline{AB}, \overline{BC}, \overline{CA}$, then the three equivalent distances $d_e(A,B)$, $d_e(B,C)$ and $d_e(C,A)$ may not form a triangle.

The resulting equivalent model is therefore a weighted directed graph roughly representing the system, where one of the metrics associated to the arcs is the equivalent distance, as shown in Figure 7. Optimization algorithms can directly be applied to this graph, and resulting solutions implemented in the real system. Note that the values computed from the equivalent model will generally have a high error margin and their absolute value can arbitrarily be far from the real one. Nevertheless, the important factor is that the equivalent model and the real scenario show the same trend, in other words:

- given two successive states of the system, S_1 and S_2;
- let $Obj(S_1)$ and $Obj(S_2)$ be the corresponding real system objective values (e.g., the overlap area);
- for adaptation, the difference between these two objectives can be calculated as $\Delta = Obj_{eq}(S_2) - Obj_{eq}(S_1)$;
- if $\Delta \geq 0$, the adaptation is kept, otherwise it is discarded;
- similarly, the corresponding values $Obj_{eq}(S_1)$, $Obj_{eq}(S_2)$ and Δ_{eq} can be calculated in the equivalent model.

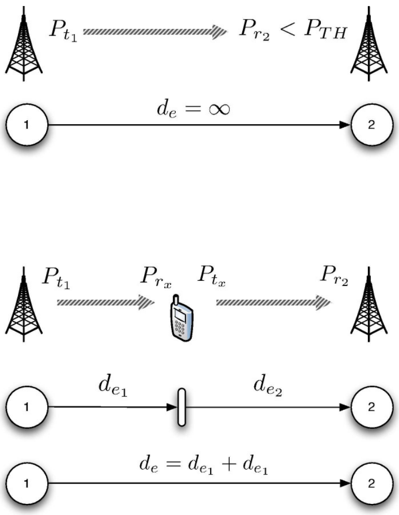

Fig. 8: Equivalent distance update

Most likely $Obj_{eq}(S_1) \neq Obj(S_1)$ and $Obj_{eq}(S_2) \neq Obj(S_2)$, but the sign of Δ_{eq} should be most of the time the same as for Δ. As a result, the adaptation can be driven in the same direction, but based on the much simpler computations on the equivalent model. Early results of the goodness of such statement are shown in Section V.

C. Model Adaptation

The accuracy of the model can be improved using information derivable from feedback generated by end users. For example, this feedback could be:

- statistics indicating a large number of dropped calls in a area supposedly well covered;
- users reporting high interference in areas with little overlap.

The model can be adapted in many ways to take into account this feedback. One particular case is where the received power P_{RX} on a given cell is smaller than a threshold P_{TH}. In such a case the equivalent distance d_e can not be calculated and the likely existing overlapping area can not be "detected" by the equivalent model. However, when the user walks in the actually existing overlapping area, he will be able to communicate with both cells. Therefore the knowledge of transmitted and received powers of both cells and of the mobile handset, allows the calculation of d_e as shown in Figure 8.

D. Model Extension

In the previous paragraphs the equivalent model was computed based on a single parameter, that is the equivalent distance. To further enrich the model and make the equivalent objective function closer to the real objective function, some additional parameters can be introduced, such as:

1) number of observed handovers from one cell to another;
2) number of users registered in a cell;
3) degree of a cell (number of discovered neighbors).

In this case, (1) would be a weight on a connecting arc, while (2) and (3) are scalar values on the node that could be used in a varieties of ways in the objective function. Metric (1) for instance, could help determine the mobility patterns of the user, and identify potential bottlenecks, while metric (2) could be used to adjust the coverage area of the cell, enlarging it (transmitting at higher power) if the user count is low and shrinking it if the cell is overloading. Finally, parameter (3) could be used to identify the probability of handover of each node in order to create a statistical mobility pattern. All these extension metrics can be subject to adaptation and refinement over time and additional parameters can also be derived from the real system.

E. Mobility Modeling

A special case of model extension is the modeling of user mobility. As aforementioned, the behavior of the users can be a determining factor for reaching an optimal solution. Traditional schemes use "a-priori" mobility models for the users, which are made to move inside the physical scenario, and then used to measure the parameters needed for the objective function.

In a real small cells deployment there is scanty knowledge of the environment and so it can be difficult to determine which a-priori mobility model matches the problem better. On the other hand, the cells have the capability to measure a number of parameters directly linked to users' mobility, such as:

- the load of each cell (i.e., number of users in a certain area)
- inbound handover events (users entering an area);
- outbound handover events (users leaving an area);
- "directional handovers" (as correlation of the previous two parameters: users leaving an area to enter another one).

In the equivalent model, these parameters can be considered to build a mobility model adapted to the real scenario. Many techniques can be used (e.g., statistical analysis, machine learning, pattern recognition) depending on the available performances and on the required precision.

One example to build the parameter listed above is to estimate $P_{HO} = (A, B, t)$, the probability of having a handover from cell *A* to cell *B* at a time *t*. In the simplest case, this can be obtained by collecting handovers/drop calls history data over a time measuring interval T and extracting the relative probability distribution (occurrence, variance). This will create very basic user behavioral patterns.

In addition the information on the degree of each cell (number of incoming/outgoing arcs) can be used to calculate the handover probability distribution of all possible next cells, identifying the most probable one.

The combination of both the mobility model and the coverage estimation allows the development of more accurate handover prediction algorithms. In more advanced approaches, in fact, P_{HO} can be obtained using a self learning process based on the evolution of the model and on the combination of location, direction, speed and node degree history data. An accurate prediction model is extremely useful to optimize the node power consumption, to minimize handover latency, and to improve the QoE (Quality of Experience) preconfiguring the resources needed for a given service. Therefore this second use complements the coverage estimation and makes the Equivalent Model rich enough to be used as benchmark within an optimization process by estimating the goodness of new mutations generated in a genetic approach, before applying them to the real system. Finally, the mobility model extraction process is not intrinsically centralized, so it could easily be used in distributed approaches, where each cell estimates the transition probability to/from its neighbors and uses it for its independent evolution process.

V. PRELIMINARY RESULTS

In the first experiment, the equivalent model has been used to estimate the overlap areas in the small cells office simulation scenario in [4]. As part of the fitness function this value is used to drive the genetic approach in the coverage optimization process. The obtained solutions can be compared with those resulting from available measured values of the real scenario, in order to validate the effectiveness of the proposed approach. In Figure 9 the evolution choice comparison between the equivalent model and the real overlap is shown. At each generation of the genetic process, the best individual (best algorithms in this particular implementation) is chosen to generate new children. As shown in Figure 9, the choices made based on the equivalent model match by around 90% those made based on the real scenario. For the same experiment, in Figure 10 and Figure 11 the maximum and mean fitness comparison over 100 generations are shown. Figure 10 reveals that the trend of the maximum fitness related to the equivalent model (solid line) follows that of the real scenario (dashed line). Similarly, Figure 11 shows that for each generation, the sign of the first derivative of the average fitness calculated using the equivalent model largely matches that related to the real scenario (i.e., the mean of the fitness function shows the same behavior over time).

A. Future Work

More sophisticated approaches, which would be worth of further investigation, involve a more radical modification of the graph itself, like for instance:

- introduction of "discontinuity nodes", which could be used for example to model walls;
- additional parameters to balance the influence of cells in the computation of the objective function, which can be used in highly asymmetrical situations (i.e., when $P_A \gg P_B$);

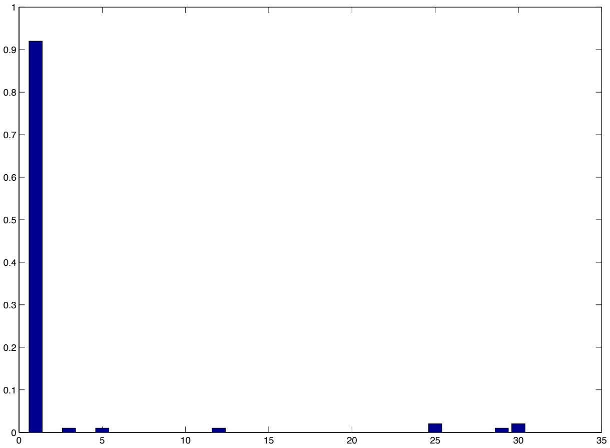

Fig. 9: Mutation choice comparison between equivalent model and real overlap

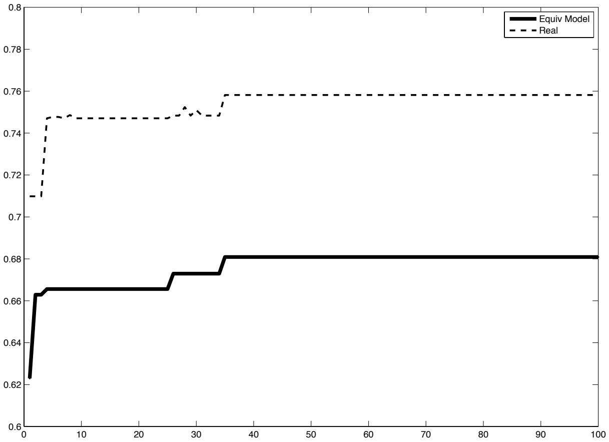

Fig. 10: Maximum fitness comparison over 100 generations

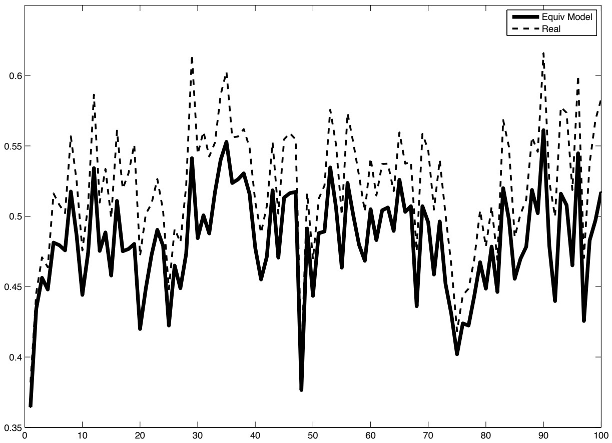

Fig. 11: Mean fitness comparison over 100 generations

- integration of users as elements of the graph (e.g., as "mobile" nodes).

VI. Conclusions

This paper introduces the generation and the automatic adaptation of a simplified Equivalent Model (EM) that can be used to speed up the convergence of self-X optimization algorithms. These methods, in fact, do not require a detailed knowledge of the target environment, but only enough information to compute and compare the fitness function values of the different mutations. The Equivalent Model is tailored for such needs because:

- it takes as input only values that can effectively be read on the small cells: *power levels* (both in transmission and reception) and *network statistics* (cell load, handovers, dropped calls);
- it tracks and simulates only values used as inputs for the fitness function: *network statistics* (cell load, handovers, dropped calls).

The fitness functions computed from these values will suffer from a lot of simplification errors, and their absolute value will not be reliable. On the other hand, if GP is used then the absolute value has no meaning: what is important is the gradient of the function, i.e. which mutation has the highest fitness value. All mutations should suffer from roughly the same errors derived from the EM, so the gradient should be respected. Unfortunately, the high variability of the scenarios makes a formal demonstration almost impossible: an important deliverable of this paper has been the validation of this assumption, as detailed in section V.

References

[1] F. J. Mullany, L. T. W. Ho, L. G. Samuel, and H. Claussen, "Self-deployment, self-configuration: critical future paradigms for wireless access networks," in *Proceedings of the First international IFIP conference on Autonomic Communication*, ser. WAC'04, 2005, pp. 58–68.

[2] H. Claussen, L. Ho, and L. Samuel, "Self-optimization of coverage for femtocell deployments," in *Wireless Telecommunications Symposium, 2008. WTS 2008*, april 2008, pp. 278 –285.

[3] O. Blume, H. Eckhardt, S. Klein, E. Kuehn, and W. M. Wajda, "Energy savings in mobile networks based on adaptation to traffic statistics," *Bell Labs Technical Journal*, vol. 15, no. 2, pp. 77–94, 2010. [Online]. Available: http://dx.doi.org/10.1002/bltj.20442

[4] L. Ho, I. Ashraf, and H. Claussen, "Evolving femtocell coverage optimization algorithms using genetic programming," in *Personal, Indoor and Mobile Radio Communications, 2009 IEEE 20th International Symposium on*, sept. 2009, pp. 2132 –2136.

[5] S. Fortune, D. Gay, B. Kernighan, O. Landron, R. Valenzuela, and M. Wright, "WISE design of indoor wireless systems: practical computation and optimization," *Computational Science Engineering, IEEE*, vol. 2, no. 1, pp. 58 –68, spring 1995.

[6] A. Feki and V. Capdevielle, "Autonomous resource allocation for dense lte networks: A multi armed bandit formulation," in *Personal Indoor and Mobile Radio Communications (PIMRC), 2011 IEEE 22nd International Symposium on*, sept. 2011, pp. 66 –70.

[7] C. A. Balanis, *Antenna Theory: Analysis and Design.* Wiley-Interscience, 2005.

[8] E. Weisstein, "Circle-circle intersection," 2011. [Online]. Available: http://mathworld.wolfram.com/Circle-CircleIntersection.html