Radio network optimization by means of neural networks *

Arnaud Berny

IRIN, Université de Nantes
2, rue de la Houssinière
BP 92208
44322 Nantes cedex 3 France

ECTIA
1, rue de la Noë
BP 92119
44321 Nantes France

Arnaud.Berny@ectia.ec-nantes.fr

Abstract

We propose new algorithms for designing and optimizing cellular radio networks. The aim is to cover a given area with cells whereas minimizing the cost of the network. Interference and traffic issues are also taken into account. One of the core ideas is that potential sites are considered as interacting particles. We show how sites are implicitly interacting through cells. The relation between antenna parameters and cells is learnt by radial basis function (RBF) neural networks. This relation is used to estimate antenna parameters whose cell best matches a cover hole around a site. The reduction of dimensionality of cell images is achieved with a convolutional neural network introduced in this paper.

1 Introduction

There has been an increasing demand on cellular radio networks for the last few years and this trend does not seem to slow down. This demand led operators to seek automated tools for the design and optimization of networks, in particular for frequency assignment. Now they are looking for automated cell planning tools which are able to manage many objectives and constraints at the same time. This radio network optimization problem shows both combinatorial and continuous sides, which makes it difficult.

The continuous part, i.e., the choice of antenna parameters, is addressed with radial basis function neural networks. It is viewed as an image matching problem which is to find a cell close to a given cover hole. Images are preprocessed by a convolutional neural network in order to reduce the complexity of subsequent stages.

The combinatorial part of the optimization is addressed with statistical physics tools used in spin-glass models or stochastic image restoration. Sites are seen as interacting particles and a collective behavior is sought in order to reach a good radio network.

*This work is funded by the EC under contract No. 23243 (ARNO project).

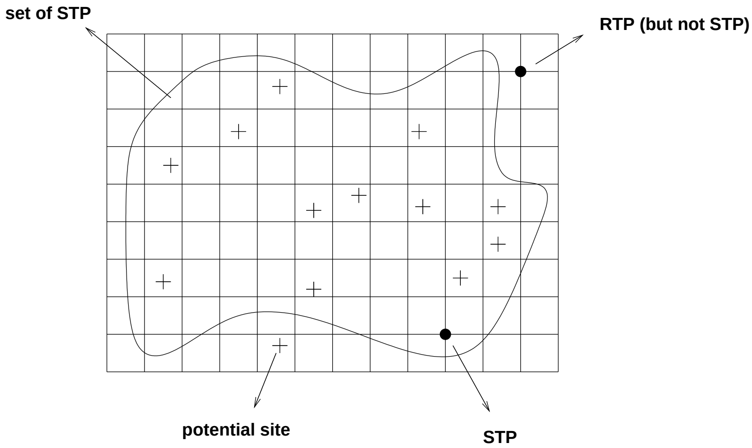

Figure 1: Basic sets of points in the area to cover.

The paper is organized as follows. Section 2 introduces the radio network optimization problem. Section 3 details the antenna parameter estimation, including the convolutional and RBF neural networks. Finally, Section 4 presents the stochastic algorithm which updates site configurations.

2 The radio network optimization problem

2.1 Basic sets

The area to be covered is sampled on a rectangular grid whose vertices are called reception test points (RTP). Only a subset of RTP needs to be covered; those points are called service test points (STP). Those sets are represented on Fig. 1.

The area contains a set of (fixed) potential sites, each of which can hold up to three base stations — a base station is an antenna plus one or more transmitters. There are two types of antenna: directional ones and omnidirectional ones. Those types are exclusive on the same site, i.e., a site can hold one omnidirectional antenna or one to three directional ones. There exists different kinds of directional antenna but we restrict ourselves to only one.

The state of a base station is completely described by three parameters: power of the transmitter, azimuth and tilt. The last two ones give the orientation of the antenna. The state of an omnidirectional one is completely described by only one parameter, the power of the transmitter.

2.2 Propagation model

The propagation model is restricted in the sense that the only information available is the loss from any site to any RTP and the angle of incidence under which any antenna sees any RTP. This information precisely synthesizes many electromagnetic phenomena such as reflexion, absorption or diffraction.

In this model, only amplitude of the electromagnetic field is considered, more precisely its logarithm (expressed in dB). Most computations on signals thus reduce to additions (gain) and substractions (loss).

Each STP receives a set of field strenghs coming from the stations in the radio network. Only fields above a sensibility threshold t_s are considered. Those fields are ordered in descending order $f_1 > f_2 > \cdots > f_n > t_s$. A quality threshold t_q is

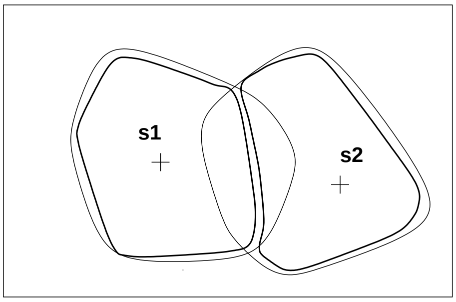

Figure 2: Difference between theoretical (thin line) and actual cells (thick line) for two stations on two neighbouring sites.

defined for each STP. It depends on the desired service (indoor, outdoor, car) and on the expected power of mobiles. If there exists at least one f_i above t_q the STP is covered. If $f_1 > t_q$ then the STP is assigned to the station that generated f_1 which is called "best server".

Each station may define a cell (provided its power is sufficient). There are indeed two different cells:

- a theoretical one, which is the set of STP where the signal is above the quality threshold t_q; it does not depend on any other station;
- an actual one, which is the set of STP where the signal is above t_q and also above any other signal; the station is the best server of this set. The actual cell depends on other stations.

The Fig. 2 illustrates this difference.

In order to ensure continuity of mobile communications across adjacent cells, radio networks provide a "handover" mechanism between adjacent cells so that an outgoing communication can be managed by at least one other station. In each cell there must exist "handover" STP belonging to at least two different (theoretical) cells, where the gap between field strenghs is lower than g_h. At most four fields may be candidate handover fields and the remaining ones are considered as interferer fields. In our notations handover fields are $f_2 \ldots f_5$ and interferer fields are $f_6 \ldots f_n$. We measure interference on an STP by the quantity $\sum_{i=6}^{n}(f_i - t_s)$.

2.3 Objectives, constraints and cost function

Traffic issues are central in the design of radio networks. We must ensure that traffic demand is served, both at a cell level and at a network level. Let us describe the way how traffic information is dealt with. A positive or null traffic quantum (expressed in Erlang) is assigned to each STP. The traffic covered by a cell is defined as the sum of traffic quanta over its STP. In order to serve this traffic demand, a station can hold one to seven transmitters — seven transmitters can serve a maximum of 43 Erlangs.

Another issue is cell topology. In order to limit handover between cells (management of the radio network) cells should consist in one connected component. This strong constraint is relaxed and "small" connected components are discarded.

The distinction between objectives and constraints is not definitive and depends on the priorities of the operator. Let us summerize the choices we have made in this model. For the constraints we have:

- cover: each STP must have a best server;

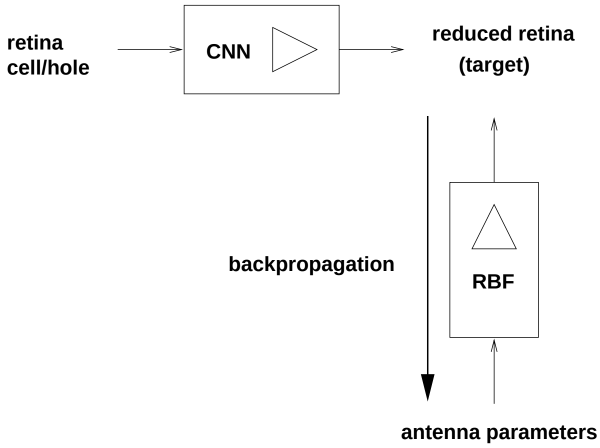

Figure 3: Architecture for antenna parameter estimation. CNN stands for convolutional neural network.

- the covered traffic of each cell must be under 43 Erlangs, i.e., the traffic capacity allowed by seven transmitters.
- each cell should consist in one connected component;

And for the objectives we have:

- minimize the number of used sites;
- minimize interference.

Only cover and traffic constraints are *explicitly* dealt with in our approach. Interference objective is taken into account during the parameter estimation stage. Nevertheless all constraints and objectives are combined in a linear cost function involved in the stochastic dynamics of site configurations.

3 Antenna parameter estimation

3.1 Introduction

Given a station and a cover hole, the aim of this stage is to estimate antenna parameters which lead to a theoretical cell close to the cover hole. This specification limits interference in the sense that cell overlap will be controlled.

The estimation is performed with two different kinds of neural networks. The first one reduces the complexity of images to be analyzed. It is a multilayer feed-forward linear neural network that extracts most of information in original images and reduces their dimension. Its output is still two-dimensional. It can be thought of as an alternative to principal component analysis (PCA) neural networks. It is called in this paper a convolutional neural network.

The second kind of neural network is a radial basis function (RBF) neural network — its output is a linear combination of Gaussians. Its inputs are antenna parameters and its outputs are the outputs of the convolutional neural network. The estimation is performed by backpropagation to the inputs.

Image analysis is restricted to a square centered on a site, called *retina*. The retina dimensions are 48×48 square RTP. The retina is the input to the convolutional neural network. We insist upon the fact that images are binary images. The overall architecture is charted on Fig. 3.

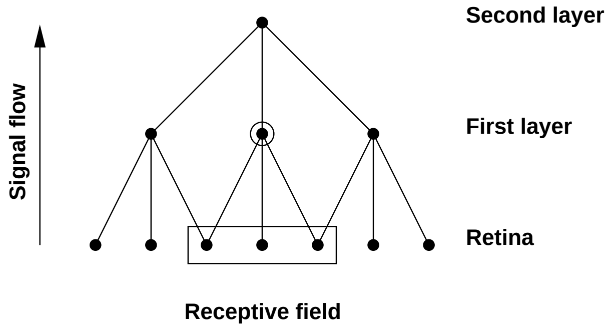

Figure 4: One-dimensional architecture of the convolution neural network.

3.2 Convolutional neural network

This neural network owes much to the work of Linsker [1]. Fig. 4 charts its architecture (one-dimensional for clarity). The neurons are organized in a pyramidal way. Each neuron takes its inputs from neurons situated in a small region in the previous layer (or the retina). This region is called a *receptive field*. One may link this architecture to multiresolution image analysis. An important parameter is the radius of the receptive field. Receptive fields may overlap.

Following [1], the main guideline in the design of the learning rule is that each neuron transmits most of information embodied in its receptive field. Let $\mathbf{x}$ be the receptive field vector of a neuron, y its scalar output and $\mathbf{w}$ its weight vector. We have $y = (\mathbf{x}|\mathbf{w})$ where $(\cdot|\cdot)$ is the scalar product. We write the learning rule as a differential equation:

$$\frac{d\mathbf{w}}{dt} = y(\mathbf{x} - y\mathbf{w}) \tag{1}$$

This rule is implemented as a discrete time stochastic algorithm. The weight vector $\mathbf{w}$ converges to the first principal component of the underlying probability distribution $p(\mathbf{x})$ and maximizes the variance $\langle y^2 \rangle$ of the output signal as a measure of the transmitted information.

Let us describe the learning protocol. Layers are trained successively, starting from the input one to the output one. At each iteration a retina is sampled from its distribution — theoretical cell images generated from the uniform distribution of antenna parameters. This retina is propagated to the trained layer and only weights in this layer are updated using Eq. 1.

Fig. 5 charts a retina and its processing by a convolutional neural network. One advantage of this device over PCA is its low complexity. Let us consider a convolutional neural network with two layers of dimensions 16×16 and 4×4 taking its input from a 48×48 retina. Set both radii to 2. The number of scalar weights is 6400 to be compared with the 36864 weights of a linear transform (such as PCA) from retinas to outputs.

A deeper analysis of convolutional neural networks in comparison with PCA needs to be carried out. Nevertheless, this architecture proved to be a good preprocessing stage for the estimation of antenna parameters.

3.3 Learning cells

The convolutional neural network provides reduced retinas and enables their learning by neural networks within a reasonable complexity. Returning to our goal of estimating antenna parameters, in order to fill any cover hole, we might think of

0.0032	-0.21	3.9	-0.14	-0.0031
0	3	6.4	5.4	2.1
0	0	8.1	6.6	-0.14
0	0	0	-0.27	0
0	0	0	0	0

Figure 5: Example of processing by a convolutional neural network.

learning the relation: (reduced) retinas $\mapsto$ parameters. The drawback is that the class of cover hole images is much wider than the class of cell images and harder to learn. If a neural network trained with cell images is presented with a cover hole (whose distribution differs from the cell distribution) its outputs might have no meaning at all in terms of antenna parameters. Moreover the learning of the direct relation is likely to be *ill-posed*, since a given cell image may map to many parameter vectors (which generate the cell).

The reverse relation: parameters $\mapsto$ (reduced) retinas is easier to learn (it is a function). Backpropagation to the inputs provides a way to change parameters in order to make the corresponding cell closer to the target cover hole.

The device that learns reduced cells is a radial basis function (RBF) neural network (see [2]). The chosen RBF is the scaled Gaussian kernel $\exp(-t^2/\sigma^2)$. Basically, a RBF neural network implements a linear combination of Gaussian kernels. Let F be the function to learn:

$$\begin{array}{rccc} F: & \mathbb{R}^m & \longrightarrow & \mathbb{R}^p \\ & x & \longmapsto & F(x) \end{array}$$

Let (z_j) be a familly of n reference vectors in $\mathbb{R}^m$. Each z_j is assigned a RBF r_j at scale σ_j. Let us introduce some linear algebra notations:

$$\begin{aligned} r_j(t) &= \exp(-t^2/2\sigma_j^2) \\ R &= (\ldots r_j(||x - z_j||) \ldots)^t \\ F &= (\ldots f_i(x) \ldots)^t \\ A &= [a_{ij}] \end{aligned}$$

where R is an n-vector and A a $p \times n$ matrix. $()^t$ denotes matrix transpose. The output of the neural network is $\tilde{F} = AR$.

The learning rule derives from the minimization of the mean square error and is an "on-line" stochastic algorithm. It expresses as:

$$\begin{aligned} A(t+1) &= A(t) + \eta(F - AR)R^t \\ z_j(t+1) &= z_j(t) + \eta \frac{(x - z_j)}{\sigma_j^2} r_j(||x - z_j||) \sum_i a_{ij}\Delta_i \\ \sigma_j^2(t+1) &= \sigma_j^2(t) + \eta \frac{||x - z_j||^2}{\sigma_j^4} r_j(||x - z_j||) \sum_i a_{ij}\Delta_i \end{aligned}$$

where $\Delta_i = f_i(x) - \sum_j a_{ij} r_j(||x - z_j||)$ is the ith output error, $\sum_i a_{ij}\Delta_i$ is the backpropagated error to the jth RBF and η is the learning rate. Those expressions closely follow those of backpropagation for multilayer perceptrons.

3.4 Estimating antenna parameters by means of inversion

Given a cover hole retina H, we want to find antenna parameters x that generate the closest cell to H. Let h be the reduced cover hole retina of H. The estimated reduced cell is $\tilde{F}(x)$ and the square error is simply $E = ||h - \tilde{F}(x)||^2$. After the training of the RBF, we have a differentiable model of F that is suitable for gradient descent techniques:

$$\frac{dx}{dt} = -\nabla_x E$$

With the notations of Sec.3.3, it leads to the equation

$$\frac{dx}{dt} = -BA^t \left[h - \tilde{F}(x) \right],$$

where B is an $m \times n$ matrix of elements

$$b_{kj} = \frac{x_k - z_{jk}}{\sigma_j^2} r_j(||x - z_j||),$$

and x_k and z_{jk} are the kth components of x and z_j. Since the model is analytically known, many second-order schemes are available (see [3]) and may result in faster convergence.

We show an example of such an "inversion" process in Fig. 6. The target is a random cell retina generated by a directional antenna. Initial conditions of parameters x are random. Subsequent images are generated by parameters x sampled from the inversion process. Sometimes the final estimated parameters do not generate the correct cell and the inversion gets stuck in some local minimum. In such cases, the solution is often to restart the inversion from other initial conditions. Those results have been obtained with 20 RBF, 30000 learning iterations and 1000 inversion iterations. RBF neural networks are site-dependant (in contrast to the convolutional neural network), which makes the learning time quite important. However, some generalization capacity has been observed and allows to start the training from a "generic" inital state in order to decrease the total learning time.

In the case of omnidirectional antennae, the function F to learn has only one parameter and RBF neural networks seem too heavy for the purpose of finding a best matching cell: very simple schemes such as the bissection method may suffice indeed.

In the case of many directional antennae on a site, an arbitrary initial solution is chosen, such as two antennae at 180° or three antennae at 120° and medium tilt and power. Then each antenna is configured on the basis of the cover hole it leaves when it is removed.

3.5 Enforcing traffic constraints

We recall that the traffic covered by an actual cell in the radio network must be lower than 43 Erlangs. The idea is to apply the same process as in cell inversion. First, an RBF neural network is trained to learn the relation: parameters $\mapsto$ traffic. If, after cell inversion, traffic exceeds 43 Erlangs, a backpropagation to the inputs (antenna parameters) causes the estimated traffic to converge towards 43 Erlangs. The drawback is that it may loose what the cell inversion had done. Both objectives (traffic and cover) are very often opposite and it is not clear whether the way traffic is corrected (priority given to power or azimuth) will be compatible with the main objective (cover).

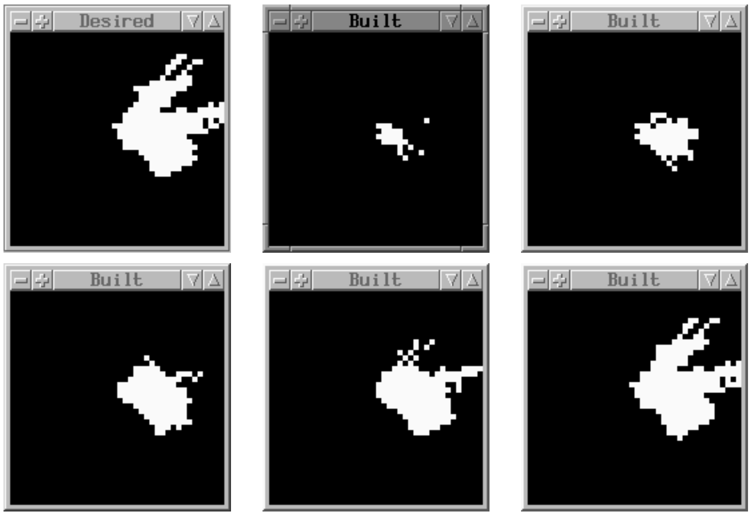

Figure 6: Inversion sequence. The first image is the target. Subsequent images are sampled from the inversion process.

4 A stochastic optimization algorithm

4.1 An analogy with statistical physics: interacting particle model

We have described a method for estimating antenna parameters but we still have to select a subset of potential sites and to decide for each of them how many antennae to put on it. For each site, the alternative reduces to:

- no antenna: the site is empty;
- one omnidirectional antenna;
- one directional antenna;
- two directional antennae;
- three directional antennae.

The choice of a site configuration leads (in a deterministic way) to antenna parameters and to a field strength distribution over the area. Sites are interacting through those overlapping field strength distributions: the shape of a cell reduces the free space available to other cells. This led us to consider sites as interacting particles taking discrete states (the continuous part has been studied in Sec. 3). The framework for studying large amounts of interacting particles is statistical physics. This framework has proved to be of great interest in the fields of image analysis and neural networks [4, 5].

We now refer to [4] and introduce the formalism of Markov random fields used in image restoration. Let Λ be the set of possible site configurations ($|\Lambda| = 5$) as described in the first paragraph and S denotes the finite set of potential sites. $\Omega = \Lambda^S$ is the set of all configurations over the whole radio network. Let $\omega = (x_s \in \Lambda, s \in S)$ be a configuration of all sites and $U(\omega)$ its cost as defined in Sec. 2.3, once antenna parameters have been estimated. Following the theory of Gibbs potential, we define a probability distribution π as

$$\pi(\omega) = \frac{1}{Z} e^{-U(\omega)/T},$$

where Z is a normalization constant (also called the partition function). T is a temperature as introduced by [6] and allows an annealing process during the algorithm. In the Gibbs potential theory, the cost function $U(\omega)$ is decomposed as $U(\omega) = \sum_{C \in \mathcal{C}} V_C(\omega)$, where $\mathcal{C}$ is the set of "cliques". A clique is a subset of sites S such that there exists an "interaction" between any pair of its elements. $V_C(\omega)$ only depends on configurations x_s of sites s in C. This property is at the heart of bayesian image restoration algorithms since it gives rise to great simplifications in the computation of conditional probability distributions.

The definition of cliques is straightforward for regular lattices in image processing but there is no such direct concept in radio networks. However there is a concept of electromagnetic range as modeled by the path loss (see Sec. 2.2).

4.2 Stochastic updating of site configurations

The algorithm builds a Markov chain over Ω: at each step, a site s is randomly selected and updated, i.e., a configuration for the selected site is sampled from the density $p_s(x)$. Let us explain how $p_s(x)$ is computed. Let ω be the current configuration and ω^x denotes the configuration ω where x_s at site s has been replaced by x. Then $p_s(x) = \pi(\omega^x) / \sum_{y \in \Lambda} \pi(\omega^y)$. For regular lattices, this expression simplifies and only involves cliques C such that site $s \in C$. Since cliques are not explicitly known in the case of radio networks, we have to evaluate the cost $U(\omega)$ of all possible configurations x for site s in order to compute $p_s(x)$ and then sample a configuration from it.

One notices that the algorithm updates sites *asynchronously*, much in the way Ising spin models or neural networks are simulated. There is no evidence of the existence of phenomenon such as "critical temperature" as observed and predicted for Ising spin models.

We claimed in the first paragraph that sites were selected randomly for updating. Actually, they may be selected from three different distributions, depending on the objective which may change during the algorithm:

- traffic distribution: high traffic regions should be served first;
- cover hole distribution: once high traffic regions have been covered, we turn to the remaining regions (with lower traffic demands but still to be covered);
- interference distribution: it is intended to localize "hard" regions.

5 Conclusion

We have presented new algorithms for the design and optimization of radio networks. Those algorithms rely on RBF neural networks for the continous part of the optimization, and on stochastic modeling, for the combinatorial part. RBF neural networks learn the relation between antenna parameters and reduced cells. Given a cover hole around a site, backpropagation to RBF inputs gives an estimation of antenna parameters whose cell best matches the cover hole. Reduced cells are computed by a convolutional neural network introduced in this paper. Finally, we have built a random field over the set of potential sites taking its values in the finite set of possible site configurations.

We have emphasized on an analogy with statistical physics and described sites as interacting particles: this analogy is fruitful in that it gives a good insight into the complex relationship (spatial, electromagnetic) between sites, without which no global optimization can be reached.

The optimization system has been developped in C++ language. Each of its components has been tested separately. Future work is well defined and includes full simulations of the model on real datasets of complexity ranging from 10 to 250 sites. Effort will especially bear upon the study of collective properties of the system and its optimization capacity.

References

[1] R. Linsker, "Self-organization in a perceptual network", *IEEE Computer*, vol. 21, no. 3, pp. 105–117, March 1988;

[2] T. Poggio and F. Girosi, "Networks for approximation and learning", *Proc. IEEE*, vol. 78, no. 9, pp. 1481–1497, September 1990;

[3] W. Press, S.A. Teukolsky, W.T. Vetterling and B.P. Flannery, *Numerical recipes in C*, second edition, Cambridge University Press, 1992;

[4] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the bayesian restoration of images", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 6, no. 6, pp. 721–741, November 1984;

[5] D.J. Amit, H. Gutfreund and H. Sompolinsky, "Spin-glass models of neural networks", *Physical Review A*, vol. 32, no. 2, pp. 1007–1018, August 1985;

[6] S. Kirkpatrick, C.D. Gelatt Jr. and M.P. Vecchi, "Optimization by simulated annealing", *Science*, 220, 1983.