Channel Assignment for Cellular Radio Using Simulated Annealing

Manuel Duque-Antón, Dietmar Kunz, and Bernhard Rüber

Abstract— **The channel assignment problem (CAP), i.e., the task to assign the channels to the radio base stations in a spectrum efficient way, is an NP-complete optimization problem occurring during design of cellular radio systems. So far, this problem has been solved by graph coloring algorithms. An alternative approach is presented in this paper. The problem is solved using simulated annealing, which is a general approach to combinatorial optimization. The algorithm has been successfully applied to practical radio network planning situations. One major benefit of the approach consists in the enhanced flexibility it gives to the engineer.**

I. Introduction

Radio network planning consists in a sequence of tasks starting from coverage analysis and ending in channel assignment. Therein, the operator has to assign channels (or frequencies) to base stations, in a way that quality of service is guaranteed; i.e., the probability that for an incoming call request an idle channel can be found is sufficiently high and the probability that the signal-to-interference-ratio (S/I) falls below a predefined value is sufficiently low. Calculating the radio frequency (RF) propagation from topography and morphostructure, and using it together with the spatial density of the expected traffic, leads to compatibility constraints stating which base stations may use the same channel or adjacent channels. The traffic can also be used to calculate, for each base station, the number of required channels. Furthermore, there may be technical or legal restrictions arising, e.g., from combiner constraints and reflecting the assigned (respectively forbidden) frequency bands.

Besides these more basic requirements, in practice there often appear additional engineering "desires." They may, e.g., stem from the task of extending an existing network, i.e., the desire to preserve as many as possible of the already installed channels, or on the other hand, in building up a new network, to leave some freedom in order to adapt to future traffic changes. In order to be prepared for future traffic growth one may wish to estimate the number of not needed channels within the allocated frequency band. In particular this necessitates the ability to compute the minimal number of consecutively assigned channels. A quite common task in any case is the necessity to trade-off all these design objectives against each other while respecting their individual importance.

In this form, mathematically speaking, channel assignment appears as a combinatorial optimization problem closely related to graph coloring, and is as such known to be NP-complete [12], [22], an exact search for the best solution is practically impossible due to an exponentially growing calculation time. The classical graph theoretic methods [2], [7], [10], [12], [22], [25] which traditionally have been used besides the regular hexagon schemes have provided acceptable results in most cases. But they suffer from the following major drawbacks.

1) The graph theoretic approach needs as input a hard interference decision indicating whether the use of the same channel by two radio cells is allowed or not. However, such a hard decision is questionable since it is based on a high amount of uncertainty (e.g., spatial distribution of the traffic). Therefore, methods taking into account gradual differences in interference probabilities would be more appropriate for cellular radio planning.
2) A trade-off between the different requirements is impossible, i.e., if a channel assignment fulfilling all constraints cannot be found, the engineer has no way to influence which ones he prefers to violate. This is particularly annoying in the cases one wishes to rank different solutions according to low priority criteria as e.g., the retention of preset channels.
3) The graph theoretic approach does not quite hit the actual problem as it only aims for minimization of the used spectrum, whereas in practice there might exist more appropriate ways to use the assigned bandwidth, e.g. by locally reserving some channels to provide for future network growth.

To address the above points, formulating CAP as a cost function optimization problem obviously is superior to graph coloring [4], [5], [19], [20]. Reformulating the channel assignment problem in this way makes it accessible to general discrete optimization methods. The neural network approach of Hopfield and Tank [14], [4], [19] was shown to be inappropriate for CAP [20], as it yields bad solutions, even in simple cases, i.e., it favors suboptimum channel assignments. Simulated annealing, which does not suffer from this "illness," has been successfully applied to different practical planning situations [5].

II. Simulated Annealing

Simulated annealing which is a general method for the approximate solution of difficult (i.e., NP-complete) combi-

Manuscript received April 14, 1992; revised June 22, 1992.
The authors are with the Philips Research Laboratory, Aachen, Germany.
IEEE Log Number 9205018.

natorial optimization problems was originally proposed by Kirkpatrick *et al.* [18] and Černy [3]. Since then it has been applied in such diverse areas as computer aided design of integrated circuits, image processing, code design, etc.

The algorithm can be considered as a generalization of the well known iterative improvement scheme (local search). On the other hand, it can be viewed as a simulation of the physical annealing processes found in nature, e.g., the settling of a solid to its ground state, i.e., the state with minimum energy.

Generally, a combinatorial optimization problem consists of a set S of configurations or solutions and a cost function C which determines for each configuration s the cost $C(s)$, i.e., a real number. Furthermore, for performing a local search one needs to know the neighbors s' of each solution s, i.e., one has to define a neighborhood structure N on S. $N(s)$ determines for each solution s a set of possible transitions which can be proposed by s.

For local search, starting from an arbitrary solution s, in each step of iterative improvement a neighbor s' of s is proposed at random. s is only then replaced by s' if cost does not rise, i.e., $C(s') \leq C(s)$. Obviously, this procedure terminates in a local minimum, i.e., in a configuration whose neighbors do not offer any improvement in cost. Unfortunately, such a local minimum may have a substantially higher cost than the global one.

To avoid this trapping in poor local optima, now, simulated annealing occasionally allows "uphill moves" to solutions of higher cost according to the so-called Metropolis criterion [21]. More precisely, if s and $s' \in N(s)$ are the two configurations to choose from, then the algorithm continues with configuration s' with a probability given by $min\{1, \exp(-(C(s') - C(s))/t)\}$ (acceptance probability), where t is a positive control parameter, which is gradually decreased to zero during the execution of the algorithm. t is the analogue of the temperature in the physical annealing process. Note that the acceptance probability decreases for increasing values of $C(s') - C(s)$ and for decreasing values of t and that cost-decreasing transitions are always accepted (see Fig. 1).

Mathematically, simulated annealing can be modeled as an inhomogeneous Markov process, consisting of a sequence of homogenous chains at each temperature level t. Its transition matrices are then given by $P(t) = G(t)A(t)$, where $g_{ss'}(t)$ denotes the probability to propose configuration s' while being in configuration s (generation probability) and $a_{ss'}(t)$ denotes the corresponding acceptance probability.

Using this mapping to a Markov process it is shown in literature [1], [11] that, under very general conditions, there exist two modi for the stochastic convergence of simulated annealing to the globally minimal configurations. Firstly (homogenous case), asymptotic convergence to a global minimum is guaranteed, if the temperature is lowered in any (unrestricted) way to 0, and if the homogenous chains are extended to infinite length to establish the stationary distribution on each level. Secondly (inhomogenous case), convergence is guaranteed, irrespective of the length of the homogenous chains, if the temperature t_k approaches 0 logarithmically slow, i.e., if $t_k \geq t_0/\ln(1+k)$ with t_0 being a sufficiently high start temperature and k the temperature level count.

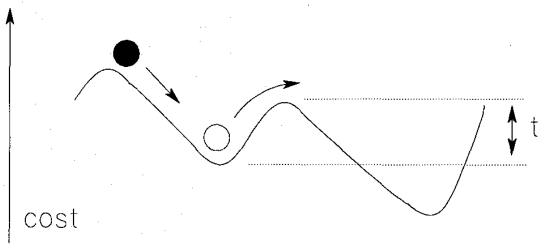


Fig. 1. Allowing uphill moves up to a cost proportional to the instantaneous temperature t.

As both convergence modi call for infinite computing time, and as simple enumeration of the configuration space has an exponential time complexity, in practice, some polynomial-time approximation is used, while preserving as much as possible of the flavor of the convergence theory. The formal procedure is to choose a so-called cooling schedule, i.e., to decide for:

- the start condition, i.e., the initial temperature t_0;
- the decrement rule for the temperature;
- the equilibrium condition, i.e., the length L_k of the Markov chains;
- the stop condition, i.e., the final temperature.

In doing so, most cooling schedules lean on the homogenous variant of simulated annealing and try to establish and maintain equilibrium on each temperature level by appropriately adjusting the length of the homogenous Markov chains and the cooling speed. Of course, any polynomial-time approximation can at best achieve a pseudo-equilibrium.

In summary, using simulated annealing to solve a discrete optimization problem means the following.

At first, one has to express the problem as a cost function optimization problem by defining the configuration space S, the cost function C, the neighborhood structure N. Next, one chooses a cooling schedule, and finally one performs the annealing process. The following pseudo-code program describes the general approach for the simulated annealing procedure.

```
1) Initialize $(s_{\text{start}}, t_0)$.
2) $k := 0$.
3) $s := s_{\text{start}}$
4) Until *Equilibrium reached* do:
   4.1 Generate $s'$ from $N(s)$.
   4.2 Metropolis test:
      4.2.1 if $\min\{1, \exp(-(C(s') - C(s))/t_k)\} >$
            random$[0,1)$ then continue with $s := s'$,
      4.2.2 else continue with the old $s$.
5) If *stop criterion valid,* stop.
6) $k := k + 1$.
7) Calculate $t_k$.
8) Goto 4.
```

Initially, one starts with an arbitrary configuration s_{start}. The corresponding initial temperature has to be chosen high enough in order to allow that most of the proposed transitions pass the Metropolis criterion. Hence, at the start of the algorithm, a free search in configuration space is intended.

As t decreases, ever fewer proposed transitions are accepted, and finally, at very small values of t, now being in a local minimum, no proposed transition is accepted at all. Having arrived at this point there is no use in further continuing the annealing process, i.e., the algorithm may stop if there are no expectations for a further substantial improvement in cost. Thus the annealing process converges to a final configuration s_{final}, which could be interpreted as a solution of the discrete optimization problem.

III. Application

In order to apply simulated annealing to CAP, we have to formulate CAP as a discrete optimization problem. For that reason, we have to define the corresponding discrete configuration space S, the cost function C and the neighborhood structure N. On a first approach this could be done in a more or less straightforward manner.

A. Configuration Space

We assume to have a mobile radio network of n radio cells, each of them capable to carry any of the m channels that are available for the whole system. m is given either by the available radio spectrum (reserved for building up the network at hand) or can be estimated by using graph theoretic methods to compute lower bounds for it [8], [9]. In a configuration one wishes to represent the current state of the channel assignment. So, a natural choice is given by a binary matrix (s_{ij}) of dimension $m \times n$ with the following interpretation of the solution entries:

$$s_{ij} = \begin{cases} 0 \\ & \text{if channel } i \text{ is } \begin{Bmatrix} \text{not used} \\ \text{used} \end{Bmatrix} \text{ at radio cell } j. \\ 1 \end{cases} \tag{1}$$

B. Cost Function

Basic requirements for a mobile radio system are the avoidance of interference and the ability to serve the expected traffic. Thereby, the interference compatibility matrix and the channel demand can be calculated using standard radio planning tools, like, e.g., GRAND [8]. The compatibility matrix $(c_{jj'})$ is used to model interference: possibly interfering base stations j and j' (i.e., $c_{jj'} \geq 0$) use different channels i and i' (i.e., $|i - i'| > c_{jj'}$). Analysis of traffic data yields a number of required channels traf_j for each base station j. Thus a generic choice for a cost function is

$$C(s) = \frac{1}{2} A \cdot \sum_{\substack{(i,j),(i',j') \\ (i,j) \neq (i',j') \\ |i-i'| \leq c_{jj'}}} s_{ij} s_{i'j'} + \frac{1}{2} B \cdot \sum_j \left(\sum_i s_{ij} - \text{traf}_j \right)^2 \tag{2}$$

which penalizes the violation of each such constraint in the following way.

The first term becomes positive if two interfering cells j and j' are assigned two channels i and i' within their interference bandwidth of $c_{jj'}$, whereas the second term penalizes traffic violations, i.e., it becomes positive if the number of channels momentarily employed at cell j (which is $\sum_i s_{ij}$, differs from its demand traf_j.

Thus $C(s)$ reaches its minimum of zero if all constraints are satisfied.

C. Neighborhood Structure

Simple choices for the neighborhood of a configuration s are produced by performing the following transitions:

- a *single flip,* i.e., just switching on or off one channel i in one cell j,
- a *flip-flop,* i.e., replacing at cell j one used channel i_1 with one unused i_2.

We use a generation probability which proposes new configurations equally in $N(s)$.

Obviously, the flip-flop is designed to preserve the number of channels used at each base station. Consequently, the configuration space has to be restricted to channel assignments with the required channel numbers and the traffic term in the cost function (second term in (2)) renders itself superfluous. This demonstrates the quite close interplay between the different ingredients of simulated annealing.

IV. Improving Algorithm Performance

Now, the hard work of applying simulated annealing to a definite problem consists mostly in the fine tuning of the algorithm's components in order to optimize run time efficiency. As for the cooling schedule, this work can be done in a consistent way for a substantial class of problems. The results in this paper have been obtained by implementation of a mixture of different cooling schedules (see [1], [15], [16], [24]) which show a polynomial-time approximation behavior.

But the larger part to be gained lies in optimizing the neighborhood structures with respect to the definite problem, in conjunction with the configuration spaces and cost functions. In the case of the channel assignment task this gives us enough freedom to combine the simulated annealing procedure with graph coloring heuristics.

In the following, after describing the cooling schedule, we exemplify two of our approaches, used for regular networks on the one hand, and for inhomogeneous ones on the other hand (cf. Section VI).

A. Cooling Schedule

According to the general rules outlined in Section II, the following mixture of cooling schedules from literature [1], [15], [16], [24] has been employed.

For the *initial temperature* it is sometimes useful to just set it by hand [16]. This gives some room for analytical precalculations as well as for free experimentation. But as a standard scheme we employed the automatic setting according to Aarts and Korst [1]. There, the initial temperature is determined in a way as to assure a user specified acceptance ratio χ of accepted to proposed transitions. For that, first, t is set to 0, and then it is iteratively changed until the desired acceptance ratio is reached. According to the above general considerations for a cooling schedule, we have used this procedure for acceptance ratios between 0.7 and 0.9, where it worked very well.

Maintaining equilibrium, or better, not disturbing it too much, is interpreted by Huang *et al.* [15], [24] as the following restriction to *temperature decrement*: The decrease ΔC in average cost between two subsequent temperature levels t and t' should be less than the standard deviation of the cost (on level t):

$$\Delta C \overset{\text{def}}{=} \langle C\rangle(t') - \langle C\rangle(t) = -\lambda\sigma, \text{ with } \lambda \in (0,1). \qquad (3)$$

Then, using the statistical definition of temperature [23] and some calculus, [15], [24] transform (3) into the following temperature decrement rule:

$$t' = t \cdot \exp\left(-\frac{\lambda t}{\sigma}\right). \qquad (4)$$

The *establishment of equilibrium* at a specific temperature means that the cost distribution of the homogenous Markov chains gets stationary. In practice, testing for such stationarity would cause an unacceptable monitoring overhead. Therefore, Huang *et al.,* [15], [24] approximate this check in two respects. Firstly, they assume a Gaussian form for the equilibrium distribution, of which the average and standard deviation are estimated from the Markov chain itself. Secondly, they regard it as stationary if the ratio of the number of accepted transitions, their costs being in the interval $[\langle C\rangle - \delta, \langle C\rangle + \delta]$, to the total number of accepted transitions reaches a stable value $\operatorname{erf}(\delta/\sigma)$, where $\operatorname{erf}(x)$ denotes the error function of x [6]. A typical value for (δ/σ) is 0.5. Additionally, in order to guarantee the polynomial-time behavior of the schedule, one needs an upper limit to the length of the homogenous chains. Thus in case that the criterion for stationarity could not be reached, the length of the chain is bounded by a maximum number which is proportional to the number of configurations which can be reached in one transition.

The *final temperature* is reached if a substantial improvement in cost can no more be expected. Huang *et al.,* [15], [24] monitor that by comparing the difference between the maximum and minimum costs encountered on a temperature level with the maximum single change in cost on that level. If they are the same, the annealing is assumed to be trapped in the basin of attraction of a local minimum, and the algorithm is stopped.

B. Dense Packing

In regular hexagons the optimal channel assignments are produced by repeating a basic pattern characterized by the fact that in this way the same channel is reused as closely as possible. In order to imitate this behavior for the simulated annealing algorithm we have modified the basic flip-flop transitions in the following way.

1) Choose at random a cell j.
2) Find all the nearest cells j' to j that may share the same channel with j (see Fig. 2 for the 3 cell cluster).

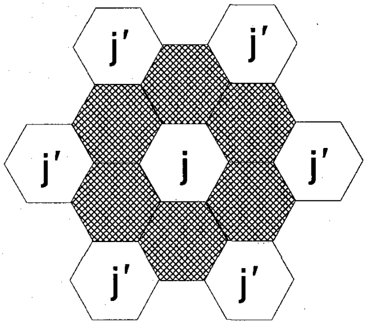


Fig. 2. "Nearest reusable" cells j' to cell j in the 3 cell cluster.

3) From all the channels i currently not used at j switch on that one which is used most of all within cells j'. (This introduces the preference of densely packed channel structures as found in the hexagons.)
4) Switch off at random one of the channels previously used at j.

Out of theoretical reasons [1] these dense packing transitions cannot be used solely, but have to be mixed with the basic flip-flops. More precisely, this means to combine the different generation probabilities in the following way:

$$g_{ss'} := a \cdot g_{ss'}^{\text{flipflop}}(t) + b \cdot g_{ss'}^{\text{dense}}(t)$$

with

$$a + b = 1 \wedge 0 \leq a, b \leq 1.$$

The parameter setting $a = b = 0.5$ yielded excellent results for the hexagons (cf. Section VI).

C. "Extended" Neighborhood Schemes

A serious problem with the inhomiogeneous networks was to be trapped in an overwhelming number of local minima. Also, it proved impossible to concentrate computing time on "relevant regions of the cooling curve." We just nearly always encountered the following behavior: once being trapped in a poor minimum (by a misplaced transition), the chance to randomly get out of it, being just one of the numerous other possible transitions, is much too low.

Simple approaches to cure the situation are tuning the neighborhoods to prefer flip-flops which resolve existing interference or preset violations and to disadvantage those (additionally to the penalty in cost) that introduce new ones. By this, some improvements in the algorithm's performance could be reached.

Of course, these observations are quite typical for simulated annealing as it is just the existence of this kind of minima which causes its worst case behavior of logarithmically slow cooling [11]. It is known from literature [26], [24] that one can preserve a fast cooling schedule if the algorithm is allowed to occasionally propose arbitrary long jumps. These long jumps open up the possibility to detrap from any minimum in a single transition, without being questioned by a maybe long chain of acceptance decisions.

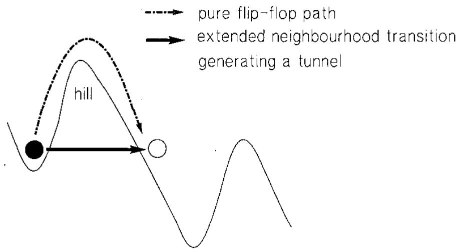


Fig. 3. "Tunneling" within the cost function landscape.

A simple method for producing a long jump is to extend the basic transitions, in our case the flip-flops, to a chain of consecutive ones. This additionally gives one the freedom to select the individual chain members according to some quality criteria, and to stop chain generation if no further improvement is expected [17]. Hence, we employed a sequence of flip-flops each member being chosen as to systematically resolve the existing interferences while introducing the smallest possible number of new ones (see above).

By properly adjusting the chain length, this allows to tunnel through a hill of the cost function landscape in one single jump, instead of painfully working to its top just to fall down into the next valley (see Fig. 3).

V. Customizing the Channel Assignment

In practice, besides the basic requirements of avoiding interference and serving the traffic, there often arise additional engineering needs. As examples we mention the need to take into account gradual differences in interference probabilities as well as the task of extending an already existing network. Optionally, one may wish to minimize the used spectrum. Such additional "constraints" can be treated in the same manner as the basic ones, i.e., by appropriately designing the simulated annealing components. Of course, in doing this, care has to be taken that basic and additional requirements are treated with their appropriate weights. To illustrate what is meant by that in detail we present the following examples.

A. Soft Interferences

The compatibility matrix $(c_{jj'})$treats all interference relations between base stations with equal strength. This ignores the practical situation that some cells strongly interfere whereas others only conflict in some small boundary areas. Moreover, interference prediction is a quite delicate task, depending, e.g., on the details of the traffic assumptions. Therefore, sometimes it is more appropriate to treat them in a soft way, i.e., just as desires. If a channel assignment respecting all interference relations (or more general fulfilling all constraints) cannot be found, it should be possible to rank their violation according to their individual probabilities. An appropriate modification of the interference term in the cost function is given, e.g., by

$$C_{\text{int}} = \frac{1}{2} A \cdot \sum_{\substack{(i,j),(i',j') \\ (i,j)\neq(i',j')}} \tilde{c}_{jj'ii'} s_{ij} s_{i'j'} \qquad (5)$$

where $\tilde{c}_{jj'ii'}$ models the interference strength between channels i and i' at cells j and j'.

B. Hard and Soft Presets

In extending existing networks it may be useful to preserve the already installed radio channels in order to avoid hardware replacements. The simplest way to achieve these channel retentions is to restrict the configuration space to assignments respecting the presets, which we call the case of hard presets. By this one gains the additional advantage of reducing problem size.

But often it is more appropriate to treat the presets in the same manner as the soft interferences, i.e., just as a desire to preserve as many channels as possible without violating the basic interference and traffic constraints "too much." A straightforward way for achieving this consists in adding a soft preset term to the cost function which punishes preset violation:

$$C_{\text{pres}} = \frac{1}{2} C \cdot \sum_j \sum_i p_{ij}(1 - s_{ij}) \qquad (6)$$

where the strength p_{ij} determines the weight with which preset channel i is demanded at base station j.

C. Minimized Spectrum Demand

If one wishes to be prepared for future traffic growth, one needs to know how much channel capacity is left unused, for the network to be built up at the moment. In general, the unused part of the whole spectrum could be distributed among the whole spectrum in various ways. The simplest case consists in minimizing the whole spectrum, in particular leading to a minimum number of consecutively assigned channels. In this case, we need an upper bound m for the number of required channels. In the case of only co-channel interference being present, m is simply $d_{\max} + 1$, where $d_{\max}$ is the maximal degree of the expanded interference graph. The case of adjacent channel interference is slightly different, see [4].

Now, we have to add the following term to the cost function which handles the minimization aspect:

$$C_{\min} = \frac{1}{2} D \cdot \sum_i \sum_{i'} \frac{|i - i'|}{m} s_i s_{i'}$$

$$\text{where} \qquad s_i = \max_j s_{ij}. \qquad (7)$$

$C_{\min}$ controls the part of the spectrum to be used. It favors states which minimize the used spectrum minimizing the distances between used channels.

VI. Results

To access the capabilities of simulated annealing in network planning, several cases of channel assignments have been extensively studied. To highlight our results, here, we present two representative examples:

1) the *7-cell cluster* as a standard test case for regular cell assemblies;
2) a *typical inhomogeneous network* as found, e.g., in hilly terrains.

TABLE I
MAIN PERFORMANCE FIGURES FOR THE DIFFERENT NEIGHBORHOOD SCHEMES (see text)

	Hit rate		Final Cost		Computing time (on T800)	
Problem	pure flip-flop	soph. scheme	pure flip-flop	soph. scheme	pure flip-flop	soph. scheme
7-cell cluster	16%	52%	48 ± 22	28 ± 29	(3.5±1) h	(3±2) h
inhom.network	4%	32%	2.24 ± 0.45	1.28 ± 0.44	(64±1) h	(51±7) min

The 7-cell cluster consists of 14 × 14 cells arranged in a doubly periodic array (a torus) where interference extends to the second ring of neighboring cells. There, the channel assignment task is to equip each cell with 2 out of 14 channels.

In the inhomogeneous example there are 239 cells and 38 channels (or more exactly: frequency groups). Interference is inhomogeneously distributed as caused by a typical hilly terrain with the number of interference partners of a cell ranging from about 20 to 70. Additional complexity is added by the desire to preserve as much as possible of an already existing channel assignment, which means that each cell typically has a preset of 1 and a demand of 1 to 3 frequency groups.

In order to appreciate the following results one additional word has to be said concerning the statistics. As mentioned in the previous sections there are several components of simulated annealing depending on random decisions, thus rendering it a stochastic algorithm. Accordingly, the details of a single run and especially its final configuration are to some extent arbitrary. We have made the experience that for the channel assignment application approximately 25 random number generator settings should be collected to come to statistically significant results as required for a sound analysis of the method. But, as you will see from the following data, high quality results can already be achieved with much fewer repetitions.

The rest of this section now is structured as follows. First, the implementation environment is described. Then, concentration is drawn to the two—what we think are the most important—characteristics of the simulated annealing algorithm: the benefits of sophisticated neighborhood structures and the quality-versus-effort trade-offs.

A. Implementation

The algorithm was implemented in C both on a VAX environment running under VMS and on an attached transputer network using the 3L software. For the transputers, the routines were not parallelized but the network is only used as a processor farm, i.e., each transputer runs its own simulated annealing program evaluating one start point out of an ensemble. By this, the statistics for the evaluation of the stochastic algorithm are supplied in parallel.

B. Simple Versus Sophisticated Neighborhood Structures

Table I presents a comparison of the main performance figures for the simulated annealing algorithms using simple resp. sophisticated neighborhood schemes. For the final costs and the computing times, their averages and spreads are computed over the set of 25 runs. The hit rate refers to the percentage of runs that reach a final configuration with zero violated interferences. This corresponds for the 7-cell cluster to the global minimum and for the inhomogeneous network it is at least a near optimum, as the presets are weighted ten times less than the interferences.

To obtain the results of Table I we used an optimum selection for the sophisticated neighborhood schemes which was for the 7-cell cluster a mixture of 50% dense packing to the flip-flops and for the inhomogeneous net 2% extended neighborhood structure and 98% flip-flops (cf. Section IV). Of course, the elaboration of these mixtures required numerous searches into parameter space. But we have made the experience that the values found are stable over a broad range of channel assignments which means that they are a reasonably good choice also for new problems.

Computing time for the run set of Table I generally was selected as to result in a reasonable trade-off between effort, i.e., run time and expected solution quality (cf. Section C). But as this has produced a hit rate of 0 for the pure flip-flops in the case of the inhomogeneous network (final cost $= 3.07 \pm 0.61$, computing time $= (202 \pm 21)$ min) we have tried to reach a zero interference solution by going up to the power limits of our computing environment. Indeed, then one out of the 25 runs found such a near optimum. But, as is quite clear from the average and spread of the final cost, this run has to be interpreted only as a really lucky runaway.

More detailed information on the statistical distribution of the different simulated annealing runs with respect to the finally reached cost can be found in Fig. 4. Figs. 5 and 6 show the typical convergence behavior of the average cost, i.e., the average of all cost values encountered on one temperature level is plotted against temperature. It is obvious that the sophisticated schemes provide a much higher solution quality on average and strive much more directly for the final low cost states.

C. Quality Versus Effort

In Fig. 7 one finds the quality-versus-effort trade-off of the simulated annealing algorithm. This is illustrated for the case of the inhomogeneous network using the above mentioned sophisticated neighborhood scheme. Computing time then was controlled by the cooling speed and the length of the homogenous Markov chains (at each temperature level).

In detail, in Fig. 7 the distance of average cost to supposed optimum is plotted against the average computing time. Again, the averages are taken over a set of 25 runs each differing

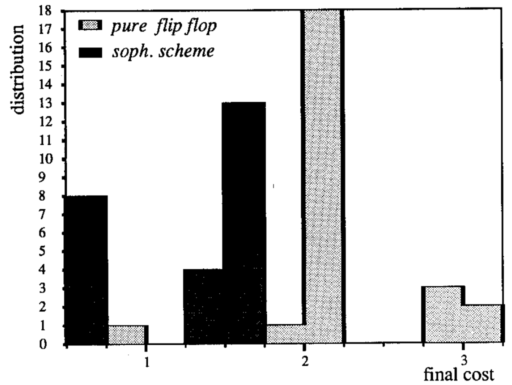


Fig. 4. Distribution of single runs with respect to solution quality for the inhomogenous network.

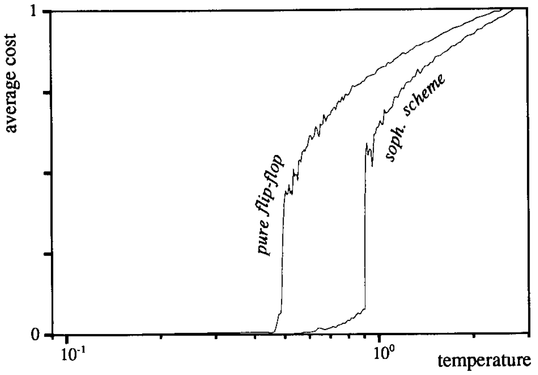


Fig. 5. Convergence histories of (normalized) average cost for both neighborhood schemes on the 7 cell cluster.

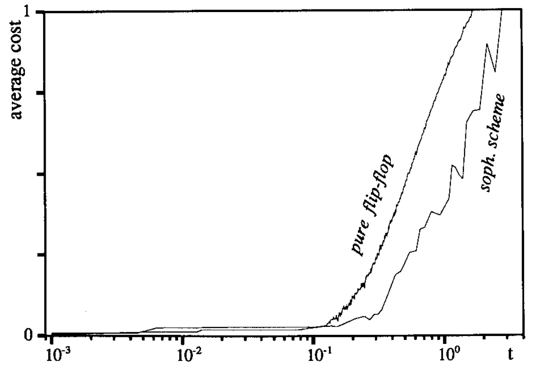


Fig. 6. Convergence histories of (normalized) average cost for both neighborhood schemes on the inhomogenous network.

only in their settings of the random number generator. For the cost of the supposed optimum the best value ever found was assumed (but note that this value only acts as a displacement for the ordinate axis of the figure).

The existence of such a quality curve together with the fact that already the very first coarse runs reveal the main characteristics of the problem at hand, as, e.g., the shape of the average cost convergence curve, may in practice be used in the following way. In tackling a new channel assignment first start with rapid cooling and short (standard) Markov chain lengths. By this, gain an overview over its specialities and improve solution quality by slowing down the cooling and lengthening the Markov chains. In such a way, within an acceptable amount of time, a solution is reached the good quality of which one can be reasonably convinced of.

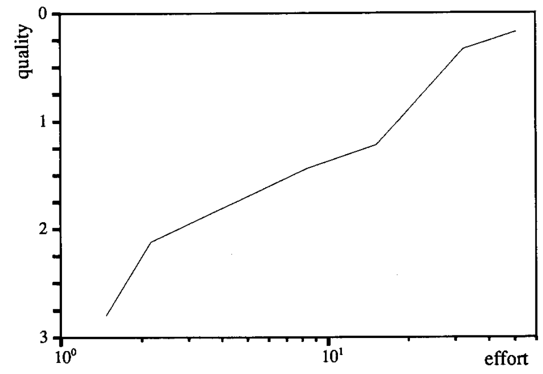


Fig. 7. Quality versus effort of the simulated annealing algorithm for the sophisticated neighborhood scheme on the inhomogeneous network where quality means distance in cost to supposed optimum and effort is measured in CPU minutes. (Please note the half-logarithmic scale of the figure.)

VII. Conclusions

Simulated annealing as a general algorithm for combinatorial optimization has been systematically applied to some forms of the channel assignment task of radio network planning. After a relative straightforward reformulation of channel assignment, problems were encountered concerning run time efficiency and solution quality. But these could be cured by the careful design of appropriate neighborhood structures. This design of specialized components for simulated annealing can be further exploited for the treatment of additional engineering desires within the basic channel assignment task.

After all, simulated annealing appears to be a quite valuable approach for practical radio network design, worthwhile to be studied further on (cf. also [13]).

Acknowledgment

The authors appreciate the assistance of R. Simon from Philips Kommunikations Industrie AG (PKI), Nuremberg, Germany for supplying actual planning data, and they thank him also for attracting their interests to the engineer's point of view to channel assignment.

References

[1] E. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines.* New York: Wiley, 1989.

[2] F. Box, "A heuristic technique for assigning frequencies to mobile radio nets," *IEEE Trans. Veh. Technol.,* vol. VT-27, pp. 57–64, May 1977.

[3] V. Černy, "Thermodynamical approach to the travelling salesman problem: an efficient simulation algorithm," *J. Opt. Theory Appl.,* vol. 45, pp. 41–51, 1985.

[4] M. Duque-Antón and D. Kunz, "Channel assignment based on neural network algorithms," in *Proc. DMR IV,* Oslo, Norway, June 26–28, 1990, pp. 5.4.1–5.4.9.

[5] M. Duque-Antón, D. Kunz, and B. Rüber, "Channel assignment using simulated annealing," in *Proc. MRC'91,* Nica, France, Nov. 13–15, 1991.

[6] W. Feller, *An Introduction to Probability Theory and Applications.* New York: Wiley, 1970, 3rd ed.

[7] A. Gamst and W. Rave, "On frequency assignment in mobile automatic telephone systems," in *Proc. GLOBECOM,* 1982, pp. 309–315.

[8] A. Gamst, R. Beck, R. Simon, and E.-G. Zinn, "An integrated approach to cellular radio network planning," in *Proc. 35th IEEE Veh. Technol. Conf.,* Boulder, CO, May 21–23, 1985, pp. 21–25.

[9] A. Gamst, "Some lower bounds for a class of frequency assignment problems," *IEEE Trans. Veh. Technol.,* vol. VT-35, pp. 8–14, 1986.

[10] A. Gamst, "A resource allocation technique for FDMA systems," *Alta Frequenza,* vol. LVII, pp. 89–96, Feb.–Mar. 1988.

[11] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions and the Bayesian Restoration of Images," *IEEE Trans. Patt. Anal. Mach. Intell.,* vol. 6, pp. 721–741, 1984.

[12] W.K. Hale, "Frequency assignment: theory and applications," *Proc. IEEE,* vol. 68, pp. 1497–1514, 1980.

[13] K. Hegedüs, Z. Zsuffa, and I. Bozsóki, "A computer-aided method for solving of frequency assignment," presented at the EMC, Wroclaw, Hungary, June 1990.

[14] J.J. Hopfield and D.W. Tank, "'Neural' computation of decisions in optimization problems," *Biol. Cybern.,* vol. 52, pp. 141–152, 1985.

[15] M.D. Huang, F. Romeo, and A. Sangiovanni-Vincentelli, "An efficient general cooling schedule for simulated annealing," in *Proc. IEEE ICCAD-86,* Santa Clara, CA, Nov. 11–13, 1986, pp. 381–384.

[16] D.S. Johnson, C.R. Aragon, L.A. McGeoch, and C. Schevon, "Optimization by simulated annealing: an experimental evaluation, pt. II (graph coloring and number partitioning)," to be published.

[17] B.W. Kernighan and S. Lin, An efficient heuristic procedure for partitioning graphs," *Bell Syst. Tech. J.,* pp. 291–307, Feb. 1970.

[18] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi, "Optimization by simulated annealing," *Science,* vol. 220, no. 4598, pp. 671–680, May 13, 1983.

[19] D. Kunz, "Channel assignment for cellular radio using neural networks," *IEEE Trans. Veh. Technol.,* vol. 40, pp. 188–193, Feb. 1991.

[20] D. Kunz, "Suboptimum solutions obtained by the Hopfield-Tank neural network algorithm," *Biol. Cybern.,* vol. 65, pp. 129–133, 1991.

[21] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.,* vol. 21, pp. 1087–1092, 1953.

[22] R.J. Pennotti, "Channel assignment in cellular mobile telecommunication systems," Masters thesis, Polytechnic Inst. New York, Brooklyn, 1976.

[23] F. Reif, *Statistical and Thermal Physics.* New York: McGraw Hill, 1965, p. 213.

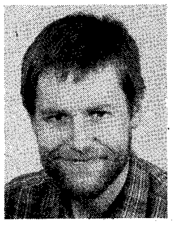
[24] F.I. Romeo, "Simulated annealing: Theory and applications to layout problems," Memo UCB/ERL M89/29, Univ. California, Berkeley, Mar. 1989.

[25] K.N. Sivarajan, R.J. McEliece, and J.W. Ketchum, "Channel assignment in mobile radio," in *Proc. 39th IEEE Veh. Technol. Soc. Conf.,* San Francisco, CA, May 1–3, 1989, pp. 846–850.

[26] H. Sue and R. Hartley, "Fast simulated annealing," *Phys. Lett. A,* vol. 122, no. 3–4, pp. 157–162, June 1987.

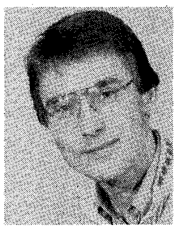
Manuel Duque-Antón was born in Zaragoza, Spain on December 10, 1958. He received the Diploma degree in computer science from the University of Aachen in 1984.

In 1985 he joined the Philips Research Laboratory in Hamburg, Germany. For four years, he has been working in the area of formal specification and verification of concurrent systems. Since 1989, he has been engaged in mobile radio communication research. His main interests are in the field of radio network management.



Dietmar Kunz was born in Wiesbaden, Germany on July 2, 1956. He received the Diplom degree in mathematics from the University of Mainz in 1980. From 1980 to 1983, he worked at the Zoological Institute of the University of Mainz, where he received the Ph.D. degree (Dr. rer. nat.) in mathematics for studies on information processing in the visual system.

In 1983 he joined the Philips Research Laboratory in Hamburg, Germany. For five years, he worked in projects on in vivo magnetic resonance imaging and spectroscopy. Since 1988, he is engaged in mobile radio communication research. His main interests are in the field of network optimization.



Bernhard Rüber was born in Mayen, Germany on November 22, 1959. He received the Diplom degree in physics from the University of Bonn in 1985.

In 1985 he joined the Philips Research Laboratory in Aachen, Germany. For four years, he worked on the computer simulation of gas discharge and incandescent lamps. Since 1990, he has been engaged in mobile radio communication research. His main interests are in the fields of simulation and optimization.