Control and Cybernetics

vol. **29** (2000) No. 3

Fixed channel assignment in cellular communication systems considering the whole set of packed patterns. An investigation based on metaheuristics

by

Pedro Castro Borges* and **René Victor Valqui Vidal****

*FCEE – Portuguese Catholic University
R. Diogo Botelho 1327, 4100 Porto, Portugal
E-mail: pmb@porto.ucp.pt

**Dept. of Mathematical Modelling, Technical University of Denmark
2800 Lyngby, Denmark
E-mail: vvv@imm.dtu.dk

Abstract: This paper addresses the problem of fixed channel assignment in cellular communication systems with nonuniform traffic distribution. The objective of the channel assignment is to minimise the average blocking probability.

Methods for finding a good allocatiȯn can be based on first building a number of sets of cochannel cells or allocation patterns and then assigning them to channels. This usually implies that only a subset of the feasible region is attainable. The approach suggested in this paper uses the concept of *packed pattern*, since all patterns in an optimal solution will be of that kind. With a constructive method, the entire set of packed patterns is built and used in the optimisation process.

The complexity (large-scale and nonlinearity) of the resulting problem suggested the use of general search procedures (local search, tabu search, simulated annealing, etc.), which have the further advantage of flexibility when dealing with extensions to the problem. A neighbouring structure was used, that facilitated the calculations while still allowing for the search in the entire solution space. A summary of extensive numerical experiments is presented. The outcome is an improvement over previous results.

Keywords: mobile communication systems, search methods, channel allocation, channel assignment, patterns, heuristics.

1. Introduction

One of the main physical constraints to radio-based communication systems is lead to regulation and allocation of different parts of the frequency spectrum to different purposes. The bandwidth demand associated with the proliferation of mobile telecommunication systems, namely mobile telephones, meant that some mechanism had to be used to allow meeting the demand more efficiently with the available bandwidth. This gave rise to the cellular concept.

In cellular systems, the same carrier frequency (channel) can be used simultaneously in several areas provided that those are distant enough from each other to avoid interference. The concept implies two operations: splitting a geographical area into cells and controlling the assignment of a number of channels to them. Both operations affect the efficiency of a mobile system. Cells splitting is modelled and performed by dividing a geographical area into regular polygons. Cell layout could be modelled by congruent polygons such as squares or triangles, however, hexagonal cells are the most commonly used for this purpose since they have advantages over other possible layouts (Macdonald, 1979), even if real networks do not have hexagonal grids.

For the operation of assigning channels to cells there are basically two philosophies: fixed channel assignment (FCA) and dynamic channel assignment (DCA). In FCA each channel is assigned to a fixed set of cells (and conversely each cell is allocated a permanent number of channels) in such a way that no cochannel interference occurs. In DCA there is no permanent allocation and the allocation is adapted on-the-fly, so that a channel that serves a call in a particular cell can later be used in a neighbouring cell.

Many DCA strategies are variations of FCA, extended with borrowing and locking mechanisms, allowing a channel assigned to a particular cell to be borrowed to neighbour cells where blocking would otherwise occur (Katzela and Naghshineh, 1996). This makes the initial assignment operation (FCA) a primary issue sometimes called nominal channel assignment. This is the problem discussed in this paper.

One of the criteria in performing nominal allocation could be the minimisation of the average probability of not being able to place a call (average blocking probability). If the traffic in a cellular grid is uniform, the optimal allocation is simple to find and will correspond to a uniform distribution of channels. However, if the traffic is nonuniform, that allocation will not be the best.

In this paper the search is performed using a subset of the feasible decision space that includes the optimal solution. To keep the optimisation approach general enough to accommodate extensions in the problem formulation, general search procedures that take advantage of the structure of the problem are used.

Section 2 presents mathematical models for the fixed channel assignment problem and discusses some of the approaches to deal with it. Section 3 defines the concept of packed patterns and suggests an approach to build the set of such patterns. Section 4 shows how a simple neighbourhood definition allows for efficient updating of the objective function. After experimenting with local results are presented in Section 6. Finally, Sections 7 and 8 point to further research and summarise the conclusions, respectively.

2. The fixed channel assignment problem

Given a cellular grid composed of N cells where each cell i has traffic demand λ_i and a number m_i of allocated channels, the call blocking probability in each cell is given by the Erlang B formula:

$$B_i = \frac{{\lambda_i}^{m_i}}{m_i!}\left[\sum_{k=0}^{m_i}\frac{{\lambda_i}^k}{k!}\right]^{-1}$$

The average weighted blocking probability is defined as:

$$AvgB = \frac{\sum_{i=1}^{N}\lambda_i B_i}{\sum_{i=1}^{N}\lambda_i}.$$

The existence of cochannel interference means that the same channel can not be assigned to two different cells that are close to each other. Such prohibition occurs if the distance between their centres is less than a minimum reuse distance d. Fig. 1 gives an illustration of the area of interference of a cell for $d = \sqrt{21}r$, where r is the cell radius.

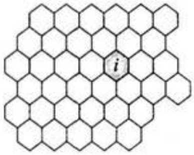


Figure 1. Use of channel j in cell i forbids its use in any of the shadowed cells

If the system has M channels available and we define binary variables a_{ij} as being equal to 1 if channel j is assigned to cell i and zero otherwise, the channel allocation model which minimises the average blocking probability is a non-linear binary programming problem and may be written as follows:

$$\min AvgB = \Big(\sum_{i=1}^{N}\lambda_i\Big)^{-1}\sum_{i=1}^{N}\left[\frac{{\lambda_i}^{m_i+1}}{m_i!}\Big(\sum_{k=0}^{m_i}\frac{{\lambda_i}^k}{k!}\Big)^{-1}\right]$$

subject to

$$m_i = \sum_{j=1}^{M} a_{ij}\ i = 1,\ldots,N$$

$$a_{sj} + a_{tj} \le 1,\ j = 1,\ldots,M,\ (s,t) \in \mathbb{I}$$

$$a_{ij} \in \{0,1\}$$

This formulation of the problem is difficult to handle because of its size. It will have large numbers of variables and constraints. The number of solutions is 2^{MN}, including infeasible ones (thus, for example, with $N = 49$ and $M = 70$ this number will be about 3.10^{1032}).

Taking advantage of the convexity of the objective function, Kim and Chang (1994) use piecewise linearisation to come to a mixed-integer formulation, which creates a larger model. They also conclude that the problem is NP-complete.

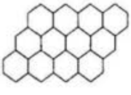
To reduce the size of the model one can regard the problem as the allocation of channels to grid patterns. A feasible pattern is a set of non-interfering cells. By using feasible patterns one avoids the need to check feasibility. In this way the check for cochannel interference feasibility is taken out of the optimisation phase, into a preliminary generation of feasible patterns.

Kim and Chang (1994) use the above mentioned mixed-integer formulation together with the concept of feasible pattern to arrive at a model allowing $(M+1)^P$ combinations of the main problem variables, where P represents the number of patterns. They use a lagrangean relaxation technique that allows obtaining information about "deviations from the true optimal solutions". This is correct if all patterns present in the optimal solution are considered. Otherwise, it is only an approximation, as the authors noted. How good this approximation is will be discussed below.

In the approach presented in this paper all the *packed patterns* are generated, as explained in Section 3, and then assigned to the available channels. If there are P patterns there will be P^M ways of assigning them to the available channels (including repeated solutions due to symmetry). This means that the search space will be smaller than $(M+1)^P$ if $P > M$ (for $P = 200$ and $M = 100$, $P^M \approx 10^{230}$ and $(M+1)^P \approx 7\cdot10^{400}$). By using search procedures in a way that allows easy calculation of the objective function it is possible to avoid having decision variables to represent the number of channels allocated to a cell (this will be detailed in Section 4). Furthermore, the only constraints left are those that correspond to variable's domains.

3. Pattern generation

The number of feasible patterns is less than 2^N. Consider the small grid presented in Fig. 2. The total number of patterns, including the empty one, is 2^{12}=4096, but, as will be shown, there are only 30 feasible combinations.



A connected grid of cells can be formed by joining two or more smaller connected grids. For example, the grid in Fig. 2 could be built by putting together three rows of four cells.

For a pattern to be feasible all its subpatterns must also be feasible. This allows to drastically reduce the number of patterns to be searched when generating all the feasible patterns. To generate the feasible patterns for the grid in Fig. 2 one can start with a single four cell row (4×1 grid), and after testing the 16 possible patterns conclude that only 6 satisfy the minimum reuse distance specified and thus are feasible (including the empty). By combining two 4×1 grids into a 4×2 grid, it is necessary to test $6.6 = 36$ patterns, to conclude that there are 15 feasible patterns in a 4×2 grid. The sets obtained for grids of dimensions 4×1 and 4×2 are then combined to obtain the feasible patterns corresponding to the 4×3 grid; the number of combinations is thus $15 \cdot 6 = 90$, of which 30 are feasible and non-empty. To build the feasible pattern set we tested feasibility of 142 patterns, many of which were smaller than the final grid (the sequence described is not the one requiring least tests).

The next step uses the fact that average blocking probability $AvgB$ is a monotonic decreasing function in m_i. This means that a feasible pattern to which one extra transmitter cell can be added without loss of feasibility can always be replaced by the latter with advantage. The only feasible patterns that are necessary to obtain an optimal solution are those in which no more cells can be inserted while maintaining feasibility. We call those relevant configurations *packed patterns*. For the 4×3 grid example, only 18 patterns have this characteristic. These are shown in Fig. 3.

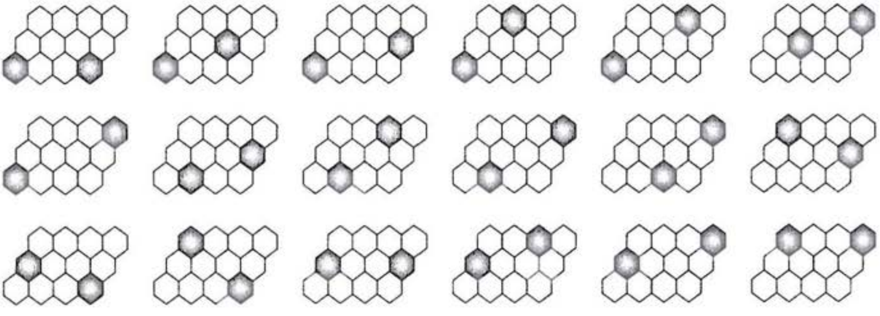


Figure 3. Complete set of packed patterns for a 4×3 grid

The elimination of feasible patterns that are non-packed might occur at the end of the generation of feasible patterns, but can be done before in respect to the areas of large patterns that will not interfere with cells outside them when building larger patterns. Fig. 4 shows a few examples of packed patterns for a mentioned above, one can generate all the packed patterns and discover that for the 7 × 7 grid there are exactly 16500 packed patterns. The number of cells in a packed pattern can vary considerably, as can be seen in Fig. 4. In the 7 × 7 case, the maximum and minimum numbers of cells are 4 and 9 cells, respectively.

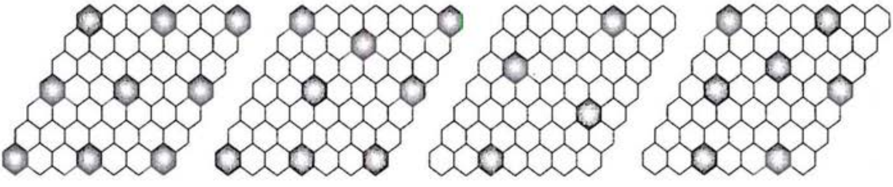


Figure 4. Examples of packed patterns for a 7 × 7 grid

Since $AvgB$ is a monotonic decreasing function in m_i, then any solution that assigns a channel to a feasible non-packed pattern can be improved by swapping that pattern with a packed one (whose set of cells is a superset of the first) yielding lower $AvgB$. Therefore, confining the solution space by only using packed patterns does not eliminate the optimal solution. We will thus restrict the search to using packed patterns.

Other reductions of the search space, like the one obtainable by considering only *compact patterns* (Zhang and Yum, 1991), may make the optimal solution of the original problem unattainable. Compact allocation patterns are patterns with minimum average distance between cochannel cells and are very easy to generate.

4. Calculating the objective function

We used general search procedures to solve this version of the channel assignment problem. All of them take an initial solution, which is modified through a succession of iterations, until a stopping criterion is met. When considering P packed patterns, M channels and N cells, a decision vector $\mathbf{x}$, representing a feasible solution to the problem, may be written as:

$$\mathbf{x} = \{x_1, \ldots, x_M\},\ x_i \in \{1, \ldots, P\}.$$

A solution is composed of M decision variables, each one representing which pattern is assigned to a particular channel. Since all the channels are equivalent, the same solution can be represented in more than one way. This redundancy could be avoided, for example, by forcing $x_i \geq x_{i+1}$. This is not done here because it would yield narrower neighbourhoods.

Since the demand values λ_i are known, the contribution to the average block-

These in turn may only take integer values from 0 to M, which makes it easy to calculate the weighted probabilities $WB_i(m_i)$ before the optimisation phase:

$$WB_i(m_i) = \Big(\sum_{i=1}^{N}\lambda_i\Big)^{-1}\frac{{\lambda_i}^{m_i+1}}{m_i!}\Big(\sum_{k=0}^{m_i}\frac{{\lambda_i}^k}{k!}\Big)^{-1},$$

$$AvgB = \sum_{i=1}^{N} WB_i(m_i).$$

Given a particular vector $\mathbf{x}$, the number of channels per cell m_i can be found by counting the number of patterns present in $\mathbf{x}$ that contain cell i. If an extra pattern is added to $\mathbf{x}$, the number of channels per cell can be updated by incrementing all the m_i whose index i is contained in the extra pattern. The update of the objective $AvgB$ is done by adding $D_i(m_i) = WB_i(m_{i+1}) - WB_i(m_i)$ for the same relevant cells i. The values $D_i(m_i)$ are also calculated beforehand. Updating the average blocking probability when removing or replacing one pattern in $\mathbf{x}$ can similarly be performed very efficiently. Since the updates of $AvgB$ are done by adding or subtracting the differences values D_i, numerical errors can accumulate and the objective value is therefore re-calculated from the values WB_i once in a while.

5. Search methods

General search procedures have some advantages over other techniques: transparency; easiness of implementation; suitability for experimentation; they always produce usable solutions; and are flexibile in that they still can be used if, for example, more constraints are included in the problem.

General search procedures like local descent methods require the definition of a neighbourhood structure. This is the set of solutions that can be reached in one single iteration from the current solution, and is determined by the mechanisms used to generate it. A very simple neighbourhood is used in this case: a neighbour of solution $\mathbf{x}$ is any solution that can be obtained by changing the packed pattern allocated to one of its channels. As seen above, modifying the value of a single variable x_i allows for very fast updating of the objective function. The size of the whole neighbourhood is $M \cdot (P-1)$. The order of generation is such that all channels are explored sequentially, starting in a random channel i (or alternatively in the channel next to the one where last change occurred) and proceeding in such a way that channel 1 comes after channel M. For each channel i, the current pattern x_i is replaced by another according to a random sequence including all packed patterns without repetition. The first pattern to be considered in each channel is chosen at random and the sequence order is changed regularly.

The problem used to test different search procedures is taken from Kim and

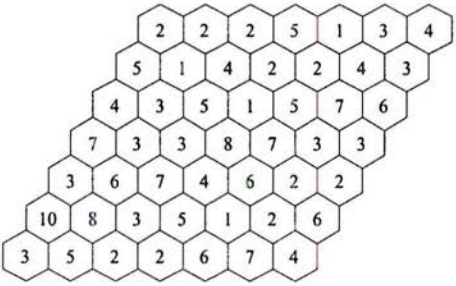


Figure 5. Cellular system with nonuniform traffic distribution (Kim and Chang, 1994)

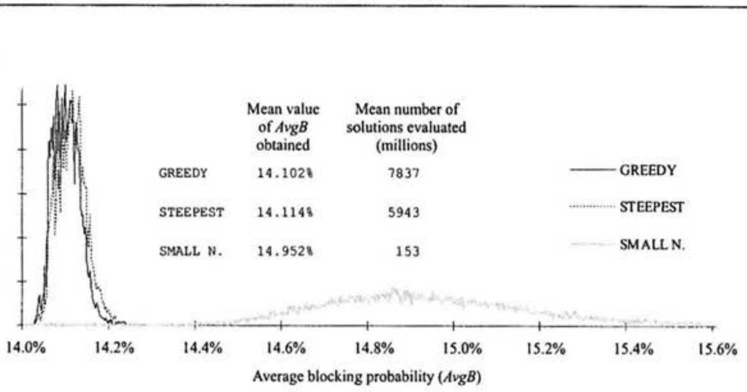
values. The tests were performed considering a percentage increase of 100% in traffic load.

Descent methods

To gain insight about the problem, it is a good idea to start by using the simplest descent methods because they are simple to implement and can be used as a benchmark. Several descent strategies were compared: a greedy version where the first neighbour that improves the current solution is immediately accepted (GREEDY); a steepest descent where the entire neighbourhood is searched before picking the best solution (STEEPEST) and; a steepest descent where the neighbourhood is smaller and chosen at random (SMALL N: 200 patterns are chosen at random for each of the 70 channels). All methods stop when no improvement is found in the current neighbourhood. A large number of initial solutions was generated (16000), and the three methods used them as initial points.

Fig. 6 summarises the results as frequency distributions for *AvgB*. The observed arithmetic means and the computational effort measured by the mean number of evaluations are also presented. The immediate conclusion is that the procedure with reduced neighbourhood is not as good as the others. Even if it is fifty times faster, the best result obtained with it is worse than the worst obtained with the greedy descent, out of all the runs. This neighbourhood could be made larger, but the analysis of the solution evolution suggests that closer to the final solutions, only a very small number of neighbours lead to improvement, which favours the conservative approach of allowing examining the entire neighbourhood. The greedy descent gives slightly better results than the steepest version, but it also requires more evaluations and therefore it is not clearly better if multiple runs are to be made.

Even if not conclusive, these results allow us to accept using greedy principles in more elaborate search procedures. That is the orientation maintained



	Mean value of *AvgB* obtained	Mean number of solutions evaluated (millions)
GREEDY	14.102%	7837
STEEPEST	14.114%	5943
SMALL N.	14.952%	153

Figure 6. Frequency distributions for $AvgB$ obtained with 3 descent methods

As a note one should mention that tests using intermediate greediness values resulted in distributions and computational effort between the greedy and steepest descents. Note that most solutions found by GREEDY and STEEPEST lie within 1% of the best $AvgB$ found.

Analysing the final solutions of the greedy and steepest descents together with an intermediate greediness descent, it was observed that only 356 out of the 16500 packed patterns are used in the final solutions (the total number of channels in those solutions is $3 \cdot 70 \cdot 16000 = 336 \cdot 10^4$). And out of those, only about 200 patterns appear more than 100 times. This means that for this particular problem there is a smaller set of interesting packed solutions. The other patterns were not dropped at this point, because they could become interesting if the problem conditions change.

The next step is the implementation of more elaborate search methods. In the following tabu search, two versions of simulated annealing, and again greedy descent are used. There is a concern in maintaining the number of evaluations identical for the sake of comparison.

Tabu search (TS)

The implemented tabu search acts like a greedy descent that is allowed to accept a degradation in the objective function if no improvement can be found in the current neighbourhood. To avoid that the search gets trapped in local minima or that it cycles, the generation of a recently visited neighbour is prohibited by a tabu rule. This can be done by forbidding the generation of the last visited solutions within a certain number of iterations, but that demands a computational effort that is too big. Instead one can keep track of the last patterns that entered or left the current solution. When pattern a replaces pattern b, both are placed in a tabu list and no moves involving a or b are allowed in the next few iterations. The idea behind this is that a pattern that enters does not leave immediately and vice-versa. The use of two different might be explained by pattern similarity; in this case there was a tendency for a pattern to be replaced in all its occurrences by a similar one; and later the reverse would occur.)

Preliminary tests showed that small tabu lists entail a higher probability that cycling occurs, which is helped by the existence of patterns very similar to each other. On the other hand, if the tabu list is too large, there is a tendency to have a tabu condition on the best patterns and worse results are obtained. The tabu size used in the test runs had 13 elements. The tabu list is emptied each time a new overall minimum is reached, to avoid missing minima that would be found by conventional descent methods. The search stops once a maximum number of evaluations is reached.

Simulated annealing (SA)

In simulated annealing the acceptance of a neighbour is sanctioned according to an acceptance rule that includes a temperature parameter T. Improvements are always accepted. Two acceptance rules are used in this paper, in which the probability of accepting a move to a neighbour producing a variation $\Delta AvgB$ in the objective is given by one of the following functions:

$$\begin{array}{c} \text{Rule A} \\ \text{(Exp)} \end{array} \begin{cases} 1, & \Delta AvgB < 0, \\ \exp\left(\frac{-\Delta AvgB}{T}\right), & \Delta AvgB \geq 0, \end{cases}$$

$$\begin{array}{c} \text{Rule B} \\ \text{(Taylor)} \end{array} \begin{cases} 1, & \Delta AvgB < 0, \\ \max\left\{0; 1 - \frac{-\Delta AvgB}{T}\right\}, & \Delta AvgB \geq 0. \end{cases}$$

The first acceptance rule is traditional for the simulated annealing literature while the second consists of the first terms of its Taylor expansion around zero. Rule B might be much faster to evaluate in many systems since it does not rely on the `exp()` function. It also causes earlier convergence at lower temperatures because it has no tail. The temperature T is decreased according to a geometric cooling schedule by being multiplied by a cooling factor α every thousand evaluations.

It is desirable to make the parametrisation of the simulated annealing (setting initial $T = T_0$ and α) somewhat adaptive to the problem even if no fine tuning is intended.

If a descent method is used to find a local minimum, the neighbourhood can be analysed and the local distribution of $F(\Delta AvgB)$ found. With the usual acceptance rule (Vidal, 1993) one can calculate temperature T_t such that the probability of accepting the pct_t'th percentile of $F()$ is equal to a value p_t. In other words, temperature T_t is such that pct percent of its neighbours have probability greater or equal to p_t of being accepted. This mechanism does not take all into account, since the whole shape of $F()$ influences the SA procedure. However, it can be used as a rationale for finding temperature values, even if

The initial temperature T_0 is set with this mechanism, given parameters p_0 and pct_0. Likewise, if after n_{01} thousand iterations, the probability of accepting the best pct_1 neighbours should be of at least p_1, that allows finding the value for the cooling factor α. In the test runs these parameters were given the following values: $p_0 = p_1 = 0.2$, $pct_0 = 0.3$, $pct_1 = 0.1$ and $n_{01} = 10^4$.

The SA search stops either when a maximum number of evaluations is reached or when it gets trapped in a local optimum at low temperature. In the last case a perturbation is introduced and the temperature is raised to a value T_2 according to two parameters p_2 and pct_2 (which were set to 0.2 and 0.1 respectively). This mechanism only acts if there is time enough to converge again, which is evaluated on the basis of the recent past. The perturbation mechanism is explained further ahead.

The initial temperature is not chosen high enough for all neighbours to have a very high probability of being selected. Test runs were made with much higher T_0 and did not yield better results, only longer runs. This could be due to the high connectivity of the solution space. More important is ensuring a slow annealing, especially at the lower temperature ranges. This is the reason why the raising of the temperature after perturbations is done to a relatively low value T_2.

Greedy descent and perturbations

The main shortcoming of the greedy descent is its inability to step out of local minima, while it takes a large number of iterations for the search to converge from a random solution to a local minimum (even if it is not as slow as SA). To avoid starting all over again from an arbitrary solution after a local extreme is found, only a perturbation is applied to the current solution. This is done by two mechanisms: setting one variable x_i at random to a pattern chosen also randomly, and always letting the neighbourhood generation start with the channel next to the one previously modified. This usually provokes a moderate increase in $AvgB$ and changes the patterns allocated to most channels.

6. Results

This section compares four heuristics whose implementation was based on the procedures described above:

- Tabu search (TS)
- Greedy descent (GREEDY)
- Simulated annealing with acceptance rule A (SA(EXP))
- Simulated annealing with acceptance rule B (SA(TAYLOR)).

The four heuristics were used a thousand times each, starting with the same random initial solutions. The maximum number of evaluations was set to $125.5 \cdot 10^7$, which is enough for all of them to come to at least one local minimum and algorithm in the runs reported in Fig. 6. The computational effort should be kept constant in the runs, for the sake of comparison.

With the greedy descent and SA implementations there is the risk that the maximum number of evaluations is reached and the run stopped before converging from a perturbation, yielding waste of resources. This must be avoided for comparison sake. Before each perturbation takes place, the procedures compare the effort used for the last convergence with the number of evaluations still left in the current run plus a safety margin. If it is found that convergence could be aborted before time, the procedure stops and the remaining evaluations are added to the maximum number of evaluations allowed in the next run, so as to keep their total number unchanged.

The results are summarised in Fig. 7 in the form of boxplots, where the mean is also indicated.

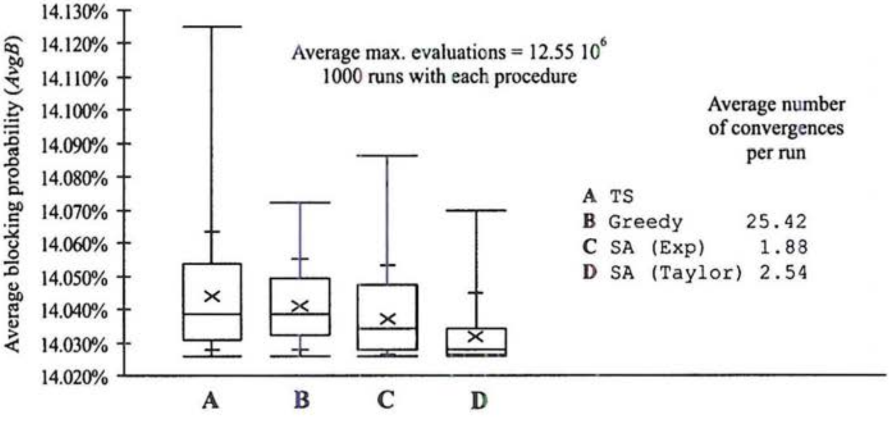


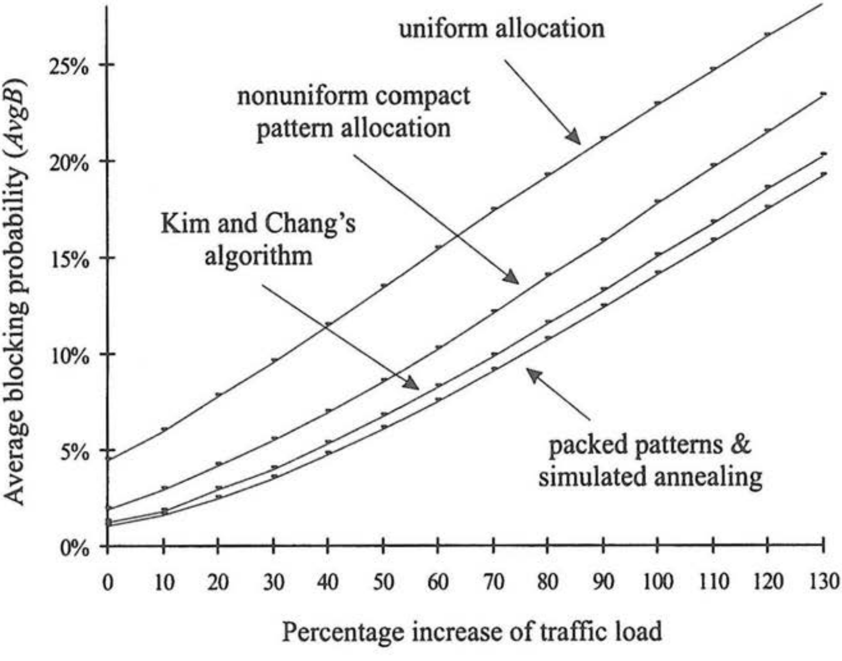
Figure 7. Boxplots of the observed distribution of *AvgB* with four procedures. Tenth and ninetieth percentiles are marked with a dash and the means by "×".

All the procedures managed to produce the same overall best objective value (14.0259%). This is slightly better than the value found by the greedy and steepest descents reported in Fig. 6 (14.0263%).

Figure 7 shows that the implemented tabu search was the least effective, what should be related with the issues mentioned when discussing the choice of tabu list size. If the list is too small — the search is trapped, if it is large — the search achieves poor results. A referee of this paper suggests that we might be observing an instance of the space/time principle (Glover and Laguna, 1997), which could perhaps be addressed by using more elaborate memory mechanisms. Also, no investigation was made as to the extent to which the performance of TS

Simulated annealing with the acceptance rule B gave the best results, attaining or beating the overall best *AvgB* from Fig. 6 in 35% of the runs. The fact that this simpler acceptance rule performed better than the traditional one (rule A) could be related to its faster convergence at very low temperatures, which allows the perturbation mechanism to be used more often. Finally, let us mention that all the 4000 final solutions found by all methods were within about 0.7% of the best found value, as the same picture shows.

This section ends with a comparison between simulated annealing (acceptance rule B) with the results reported in Kim and Chang, (1994). That paper plots a summary of results obtained with several channel allocation procedures for the grid of Fig. 5 when the traffic load is increased. Our results are represented in the same plot, shown in Fig. 8. The maximum number of iterations allowed for the simulated annealing was $251 \cdot 10^6$. The two uppermost series of points correspond to uniform allocation and to a non-uniform packed allocation algorithm by Zhang and Yum (1991). This algorithm is a greedy constructive heuristic that considers only compact allocation patterns. The third series of points was obtained by Kim and Chang's algorithm, which uses Lagrange relaxation where a Lagrangean heuristic (Beasley, 1993) must be used in each iteration to achieve feasibility. Their pattern generation procedures generated only 38 patterns.



The simulated annealing procedure described in this paper took advantage of using all the packed patterns and improved over the previous results by a few percent.

Even if none of these methods comes with a proof of optimality, it is clear that the patterns available and the use of the structure of the problem are more important than the technique used, as the relative success of simple descent methods showed in Section 5.

7. Further research

The improvement observed in Fig. 8 must be mainly due to the usage of the whole relevant set of patterns and thus not limiting the reachable solution space. However, we also confirmed that some patterns are more attractive than others. If for a particular problem those patterns were known or could be determined in advance, the search could be made more effective. Notice, however, that the now uninteresting patterns may become useful either when new restrictions are added to the problem or if other criteria are considered.

The advantage of search approaches such as the ones used herein is that they can be easily modified to incorporate additional constraints. For example, the consideration of adjacent channel interference (Gamst, 1982) would not add fundamental structural problems in this approach.

An extension to the problem was considered in Kim and Chang (1994) by including an extra constraint that limits the maximum blocking probability per individual cell. This corresponds to a required minimum service quality presented as an attempt to achieve some fairness.

Another perspective could be the explicit consideration of two objectives: minimising both the average blocking probability ($AvgB$) and the highest individual blocking probability ($maxB$). These objectives are highly correlated. Procedures can be developed to obtain approximations to the Pareto optimal set, or efficient frontier as in the examples from Ulungu and Teghem (1994), Hansen (1997), and Borges (2000). Given the relative success of descent methods in this problem and the high correlation of the criteria in the bi-criteria case, we used a procedure consisting of alternating greedy descents on both objectives. Fig. 9 was obtained with repeated usage of the following sequence:

```
repeat twice {
      greedy descent on the average blocking probability (AvgB)
      greedy descent on the highest individual blocking probability (maxB)   }
```

$AvgB$ is a quality measure that affects the money flows the cellular system generates. In this sense the efficient frontier can be used to analyse trade-offs

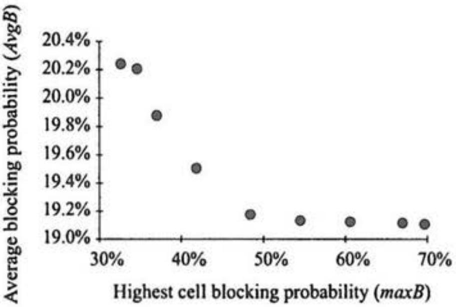


Figure 9. Approximation to the efficient frontier for 130% increase in traffic load

8. Conclusions

This paper uses a two-stage approach in which the problem of fixed channel assignment is first suitably formulated taking advantage of its structure and then optimisation is done and extensively tested using a whole family of search procedures.

It is shown that as far as individual patterns are concerned only packed patterns should be considered for the fixed channel assignment problem. The set of those patterns for a particular grid can be efficiently generated by constructive procedures. In addition, a model was used, with a relatively small solution space that in conjunction with a simple definition of neighbourhood allowed for very efficient evaluation of the objective function.

The use of the whole set of packed patterns in conjunction with general search procedures yields improvements over previous results without losing flexibility and still allowing for inclusion of other factors. The numerical tests indicate that for this problem, the appropriate formulation using the structure of the problem is more important than the optimisation method itself, even if some procedures performed slightly better than others.

References

BEASLEY, J.E. (1993) Lagrangean Relaxation. In: C.R. Reeves, ed., *Modern Heuristic Techniques for Combinatorial Problems*, chapter 6, pages 243–303. Blackwell.

BORGES, P.C. (2000) CHESS: Changing Horizon Efficient Set Search. To appear in *Journal of Heuristics.*

GAMST, A. (1982) Homogeneous distribution of frequencies in regular hexagonal cell system. *IEEE Transactions on Vehicular Technology*, **31**, 3, 132–144.

HANSEN, M.P. (1997) Tabu search for multiobjective optimization: MOTS. Paper presented at the 13th International Conference on Multiple Criteria Decision Making. University of Cape Town, January 6–10, 1997.

KATZELA, I. and NAGHSHINEH, M. (1996) Channel Assignment Schemes for Cellular Mobile Telecommunication Systems: A Comprehensive Survey. *IEEE Personal Communications*, pages 10–31, Jun. 1996.

KIM, S.H. and CHANG, K.N. (1994) Optimal Channel allocation for cellular mobile systems with nonuniform traffic distribution. *INFOR*, **32**, 3, 202–213.

MACDONALD, V.H. (1979) The cellular concept. *Bell Syst. Tech. J.*, 15–41.

PIRLOT, M. (1993) General Local Search Heuristics in Combinatorial Optimization: a tutorial. *Belgian Journal of Operations Research, Statistics and Computer Science*, **32**, 7–67.

PIRLOT, M. and VIDAL, R.V.V. (1996) Simulated Annealing: a tutorial. *Control and Cybernetics*, **25**, 1, 9–31.

ULUNGU, E.L. and TEGHEM, J. (1994) Multi-objective combinatorial optimization problems: a survey. *Journal of Multiple-criteria Decision Analysis*, **3**, 83–104.

VIDAL, R.V.V., ed. (1993) *Applied Simulated Annealing.* Lecture Notes in Economics and Mathematical Systems 396, Springer.

ZHANG, M. and YUM, T.S. (1989) Comparisons of Channel-Assignment Strategies in Cellular Mobile Telephone Systems. *IEEE Transactions on Vehicular Technology*, **38**, 4, 221–215.

ZHANG, M. and YUM, T.S. (1991) The Nonuniform compact pattern allocation algorithm for cellular mobile systems. *IEEE Transactions on Vehicular Technology*, **40**, 2, 387–391.