A Randomized Saturation Degree Heuristic for Channel Assignment in Cellular Radio Networks

Roberto Battiti, Alan Bertossi, and Daniela Cavallaro

Dipartimento di Matematica Università di Trento,
Via Sommarive 14, 38050 Trento, Italy
battiti,bertossi@science.unitn.it

Abstract In this paper we investigate the channel assignment problem, that is the problem of assigning channels (codes) to the cells of a cellular radio network so as to avoid interference and minimize the number of channels used. The problem is formulated as a generalization of the *graph coloring problem*. We consider the *Saturation Degree* (SD) heuristic, first proposed as a technique for solving the *graph coloring problem*, which was already successfully used for code assignment in Packet Radio Networks. We give a new version of this heuristic technique for cellular radio networks, called *Randomized Saturation Degree* (RSD), based on node ordering and randomization. Furthermore we improve the solution given by RSD by means of a local search technique. Experimental results show the effectiveness of the heuristic both in terms of solution quality and computing times.

1 Introduction and Problem Formulation

In a cellular mobile network, the covered area is divided into a discrete number of cells. A set of channels is assigned to each cell of the network. Calls generated in a cell i may cause interference with calls generated in a cell j. If the geographical distance between two cells is larger than a fixed value (for instance 2), the same frequency channel (A) can be reused in both cells at the same time without any interference (*co-channel cells*). We define a cellular network in the following way.

1 a set of n distinct **cells**;
2 a **demand vector m=**(m_i), $1 \le i \le n$;
3 a **frequency separation matrix** or **interference matrix C=** $(c_{ij})_{n\mathrm{x}n}$;
4 a **frequency assignment** f_{ij}, $1 \le i \le n$, $1 \le j \le m_i$, where each frequency f_{ij} is represented by a positive integer (code);
5 a set of **frequency separation constraints**:

$$\mid f_{ik} - f_{jl} \mid \ge c_{ij} \qquad \forall\, i, k, j, l \quad (k \ne l)\ .$$

We consider the following frequency separation constraints:

- **co-channel constraint:** $c_{ij} = 1$, no frequency reuse is possible in cells i and j;
- **adjacent channel constraint:** $c_{ij} \ge 2$, no two adjacent channels may be assigned to cells i and j;
- **co-site constraint:** c_{ii} represents the required frequency separation between two channels assigned to the same cell i.

The *Channel Assignment Problem* (CAP) consists of finding a channel assignment, i.e. the f_{ik}'s, for the cellular network such that the system *bandwidth*, that is $\max_{ik} f_{ik}$ is minimized.

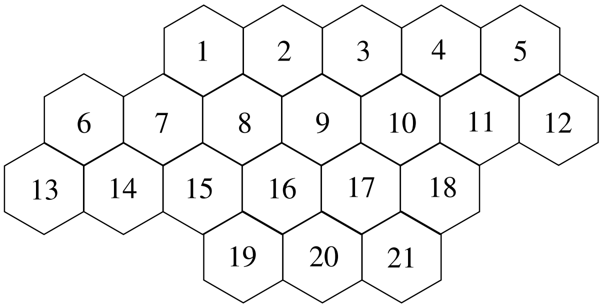

Figure1. A 21-cell system (each cell number is inside the cell itself).

The CAP problem can be formulated as the generalization of a graph coloring problem. We consider the *adjacency graph* (or *cellular graph*) formulation [6] defined as follows. Each node represents one cell and there is an edge between two nodes if the corresponding cells are adjacent in the network.

The CAP problem reduces to the problem of finding an assignment for the nodes of the cellular graph such that:

- exactly m_i codes are assigned to each node i;
- $\mid f_{ik} - f_{jl} \mid \geq c_{ij}$ for all i, k, j, l ;
- $\max_{ik} f_{ik}$ is minimized.

Clearly the graph coloring problem is the CAP problem where all entries of the demand vector are equal to 1 (only one code for each node) and where the interference matrix is a binary matrix (with meaningless diagonal entries). Therefore the CAP problem is NP-complete.

In this paper, we propose a new heuristic, called *Randomized Saturation Degree* (RSD), which is a generalization of *Saturation Degree* (SD) for graph coloring [3]. It is based on node ordering and on randomization of choices. Unlike other existing heuristics, ordering and coloring are carried out simultaneously: the first nodes to be colored are those with the greatest number of colors in the neighborhood. The RSD performance is experimentally tested on the benchmark problems proposed in [4, 5, 7, 8]. On these benchmarks, RSD often performs better than local search and provides good starting solutions for local search techniques. By combining RSD with a version of local search, obtained by giving more diversification to the CAP3 choices, we obtain the best results on most benchmark graphs. This extended abstract summarizes the algorithm and the results, while all details can be found in the extended version of the paper [2].

2 Benchmark Instances

We consider ten benchmark CAP instances, taken from [4, 7, 8], denoted as $A1, \ldots, A10$. Problems $A1, \ldots, A9$ are formulated on the 21-cells system of Figure 1, while $A10$ is formulated on a 25-cell system.

Two channel requirements for these problems are defined in Figure 2 (Cases 1 and 2 respectively).
We consider also 9 benchmark instances from [5]. The instances are formulated on the 7x7 network shown in Figure 3, and are denoted as $K1, \ldots, K9$.

The interferences extend up to the second ring of neighboring cells. The frequency separation between each pair of non-co-channel cells is 1 (absence of

			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
		node i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21
m	Case 1	m_i	8	25	8	8	8	15	18	52	77	28	13	15	31	15	36	57	28	8	10	13	8
	Case 2	m_i	5	5	5	8	12	25	30	25	30	40	40	45	20	30	25	15	15	30	20	20	25

Figure2. *Channel Requirements* for problems $A1, \ldots, A9$.

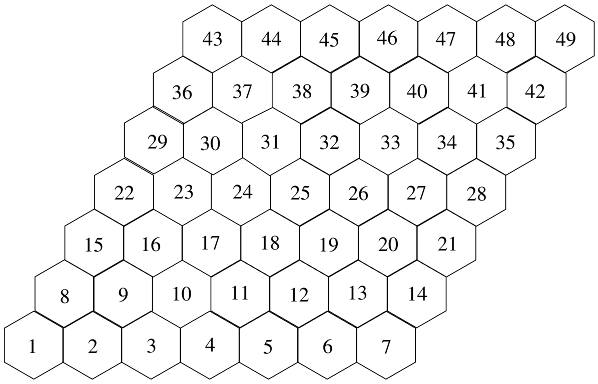

Figure3. A 49-cell system.

adjacent constraint), 2 or 3, and 3 or 4. The demand vector is generated by means of a distribution function U(X,Y), uniform over the interval [X,Y].

2.1 Local Search (LS) Algorithms

Until now, the best bandwidths for the previously presented benchmarks were achieved by local search algorithms [8].

Local search is based on a search space $\mathcal{R}$ and an objective function $\mathcal{F}$. For each point $x_p \in \mathcal{R}$ a set of neighbors $N(x_p) \subset \mathcal{R}$ is defined. A local search algorithm explores $N(x_p)$ by looking for an $x_{p+1} \in N(x_p)$ which is better than x_p, that is, an x_{p+1} that improves the value of $\mathcal{F}$.

The CAP problem is solved by a local search technique in [8]. Let us consider a cellular network with n cells, a demand vector **m** and an interference matrix **C**. Then $\mathcal{R}$, $\mathcal{F}$, $x_p \in \mathcal{R}$ and $N(x_p)$ are:

- $\mathcal{R}$: the set of all possible ordered lists of calls;
- $\mathcal{F}$: the system bandwidth;
- x_p is the current solution, namely an ordered list of calls;
- $N(x_p)$ is the neighborhood of x_p, defined as

$$N(x_p) = \{\, x'_p \,|\, H(x_p, x'_p) \leq d \,\}$$

where $H(x_p, x'_p)$ is the number of components in which x_p and x'_p differ. As in [8], we consider $d = 2$.

A local search algorithm for the CAP problem tries to find an $x'_p \in N(x_p)$ which decreases the system bandwidth (i.e. with $\mathcal{F}(x'_p) < \mathcal{F}(x_p)$). If this configuration exists, then x'_p becomes the new current solution (x_{p+1}) and the search is iterated; otherwise x_p is a local optimum in $N(x_p)$ and the search ends.

The local-search based CAP3 algorithm [8], which gives the best results for all of the above mentioned benchmarks, consists of three phases: an initialization phase and two search phases that explore the configurations in $N(x_p)$.

3 The Randomized Saturation Degree (RSD) Heuristic

We present a new CAP heuristic, which is a generalization of *Saturation Degree* (SD) proposed by Brelasz [3] for graph coloring. SD has been successfully applied by Battiti, Bertossi, and Bonuccelli [1] to Packet Radio Networks.

The basic idea of the SD heuristic is to color first that node having the largest number of colors already assigned to the neighbors.

If there is a tie between two or more nodes, the winning nodes are inserted into a set of *candidates*. The node v^* to be colored is then chosen randomly among the nodes in the set of candidates. By iterating this randomization technique, many legal colorings of the same graph can be found. Obviously, the assignment that achieves the smallest bandwidth is then chosen.

To adapt SD to cellular networks, some additional changes have been introduced. First, each node i has to be assigned exactly m_i codes. Second, codes (colors) must satisfy the constraints imposed by the interference matrix (*co-site constraint*, *adjacent channel constraint* and *co-channel constraint*).

It is worth observing that, in this heuristic, ordering and coloring of the calls are carried out simultaneously.

ToBeAssigned is the set of uncolored nodes. *NeighCodes*[i] contains the codes assigned to the neighbors that are at most k far from i, where k is the greatest distance for which an interference occurs. *NAssignedNeighbors*[i] is the number of neighbors of i, up to distance k, which have already been assigned a code.

At each step, a node v^* is chosen (lines 9-21). If there are ties (same coloring priority) the nodes are stored in a set of *Candidates* (lines 12-20), and then a random node is chosen (line 21).

Procedure ASSIGN-CODES assignes m_{v^*} colors to v^*, according to the constraints imposed by all nodes that are at most k edges far from v^* and according to the *frequency exhaustive strategy*. Then *NAssignedNeighbors*[i] and *NeighCodes*[i] are updated (lines 24-30).

3.1 Time Complexity

The two nested loops (lines 7 and 11) require $O(n^2)$ time. Procedure ASSIGN-CODES is carried out in $O(k^2 m_{max})$ time.

Consequently, the time complexity of the heuristic is $O(max(n^2, nk^2 m_{max}))$, where k is a small integer.

4 New Experimental Results

The execution of RSD obtains optimal solutions for 8 of the problems $A1, \ldots, A10$. For two harder problems neither the lower bound nor the CAP3 upper bound was reached. The results are shown in Figure 5 (third line).

```
RANDOMIZED-SATURATION-DEGREE
    ToBeAssigned ← {1, 2, ..., n}
    for i ← 1 to n do
      begin
         NAssignedNeighbors[i] ← 0
         NeighCodes[i] ← ∅
      end
    while ToBeAssigned ≠ ∅ do
    begin
          MaxNeighCodes ← −1
          MaxAssignedNeighbors ← −1
          for each i ∈ ToBeAssigned do
              if |NeighCodes[i]| > MaxNeighCodes then
              begin
                  MaxNeighCodes ← |NeighCodes[i]|
                  MaxAssignedNeighbors ← NAssignedNeighbors[i]
                  Candidates ← {i}
              end
              else if |NeighCodes[i]| = MaxNeighCodes then
                  if NAssignedNeighbors[i] ≥ MaxAssignedNeighbors then
                      Candidates ← Candidates ∪ {i}
          v* ← random(Candidates)
          ASSIGN-CODES(m_{v*},v*)
          ToBeAssigned ← ToBeAssigned \{v*}
          for h ← 1 to k do
              for each j ∈ Neigh(v*)[h] ∩ ToBeAssigned do
              begin
                  for s ← 1 to m_{v*} do
                      NeighCodes[j] ← NeighCodes[j] ∪ {f_{v* s}}
                  NAssignedNeighbors[j] ← NAssignedNeighbors[j] + m_{v*}
              end
     end
```

Figure4. *Randomized Saturation Degree*(RSD).

4.1 RSD plus Local Search

To improve the found solutions, a local search technique can be combined with the RSD heuristic. Starting from the ordered list of calls given by RSD, the idea is to search for an ordered list of calls that leads to a smaller bandwidth, by introducing local search (LS). To do this, the local search CAP3 algorithm, described in Section 2.1, can be modified as follows.

Initialization The starting calls list (x_0) is that given by *Randomized Saturation Degree.*

Search CAP3 selects the call a_{ik} which is assigned the maximum frequency channel in the current solution x_p. Then a_{ik} will remain the same until a new current configuration x_{p+1} will be found. This choice forces the algorithm to explore a small subset of $N(x_p)$. To introduce more diversification a_{jl} call is still chosen in a random way, while the a_{ik} call is that to which the maximum frequency is assigned in the coloring associated to the last ordering visited is $N(x_p)$. A new neighbor $x'_p \in N(x_p)$ is obtained from x_p by swapping a_{ik} and a_{jl}. In this way,

	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10
Lower Bound	533	533	381	414	309	309	221	229	529	73
CAP3	533	533	381	433	309	309	221	263	529	73
	<1s	<1s	<1s	110/170s	<1s	<1s	<1s	110/170s	<1s	<1s
RSD	533	533	381	463	309	309	221	275	529	73
	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s	<1s
RSD + LS	533	533	381	427	309	309	221	254	529	73
				<30 s				<35 s		

Figure5. Performance of *Randomized Saturation Degree* for $A1, \ldots, A10$.

		K1	K2	K3	K4	K5	K6	K7	K8	K9
SPCAP		96	241	337	121	331	415	166	455	604
RSD criterium 1	*best*	91	255	356	108	306	436	148	421	576
	average	97.16	269.03	376.1	120.47	340	474.7	169.07	479.4	670.27
RSD criterium 2	*best*	91	255	355	109	309	432	150	423	584
	average	98.1	269.07	376.2	121.33	340.4	475.8	169.23	481.7	674.06
RSD criterium 3	*best*	90	264	361	109	313	438	149	418	587
	average	95.8	275.367	383.2	120.43	340.03	473.43	168.7	472.96	658
RSD criterium 4	*best*	91	262	362	111	312	435	146	411	579
	average	96.03	274.83	382.03	120.6	339.36	473.4	168.5	469.7	655.23

Figure6. Performance of *Randomized Saturation Degree* for $K1, \ldots, K9$.

the search phase has more diversification in its choices and exploits the neighbors x_p while they are visited.

Further results The last line in Figure 3 shows the bandwidths obtained by executing the local search technique described above starting from the solutions found by RSD. The bandwidths obtained for the CAP benchmarks $K1, \ldots, K9$ by RSD are illustrated in Figure 6. For each problem the best solution obtained is shown together with the average over 30 iterations. The results of the SPCAP algorithm [8], derived from CAP3, are improved for 6 out of 9 problems. The local search, as described in Section 4.1, was also applied to problems $K1, \ldots, K9$. The resulting bandwidths are shown in Figure 7. The results and the four above mentioned criteria are studied in details in the extended version of the paper [2].

This work was supported by the Special Project "Laboratory of Experimental Algorithmics", University of Trento, under a grant.

References

1. R. Battiti, A.A. Bertossi & M.A. Bonuccelli. Assigning Codes in Wireless Networks: Bounds and Scaling Properties. *Wireless Networks* vol. 5, 195-209, 1999.
2. R. Battiti, A. Bertossi & D. Cavallaro. A Randomized Saturation Degree Heuristic for Channel Assignment in Cellular Radio Networks. *IEEE Transactions on Vehicular Technology*, submitted, 1999.
3. D. Brélasz. New methods to color the vertices of a graph. *Communications of the ACM* 22, 251-256, 1979.

		K1	K2	K3	K4	K5	K6	K7	K8	K9
SPCAP		96	(241)	(337)	121	331	(415)	166	455	604
RSD + LS criterium 1	*best*	(89)	249	352	(108)	(299)	419	(145)	405	563
	average	95.03	265.43	371.67	118.67	332.2	463.83	165.13	466.8	649.86
RSD + LS criterium 2	*best*	(89)	250	351	(108)	301	424	(145)	408	566
	average	95.2	265.4	371.83	118.6	332.77	466.86	165.33	466.2	651.9
RSD + LS criterium 3	*best*	90	255	351	(108)	301	421	(145)	405	572
	average	94.93	268.6	373.66	118.7	330.1	460.53	165.23	455.43	635.76
RSD + LS criterium 4	*best*	(89)	254	353	(108)	302	416	(145)	(392)	(558)
	average	95	267.36	373.73	118.7	329.7	458.83	165.2	454.06	634.13

Figure7. Performance of *Randomized Saturation Degree* and local search for $K1, \ldots, K9$.

4. N. Funabiki, Y. Takefuji. A Neural Network Algorithm for Channel Assignments in Cellular Radio Networks. *IEEE Transactions on Vehicular Technology* vol. 40, 14-21, Feb 1991.
5. S. Kim and S.L. Kim. A Two-Phase Algorithm for Frequency Assignment in Cellular Mobile Systems. *IEEE Transactions on Vehicular Technology* vol. 43, no. 3, 542-548, Aug. 1994.
6. A. Sen, T. Roxborough, S. Medidi. Upper and Lower Bounds of a Class of Channel Assignment Problems in Cellular Networks. Technical Report, Arizona State University, 1997. *http://www.eas.asu.edu/~csedept/people/faculty/sen.html*
7. K.N. Sivarajan, R.J. McEliece, J.W. Ketchum. Channel Assignment in Cellular Radio. *Proc. 39th IEEE Vehicular Technology Conf.* 846-850, May 1989.
8. W. Wang, C.K. Rushforth. Local Search for Channel Assignment in Cellular Mobile Networks. *DIMACS Series in Discrete Mathematics and Theoretical Computer Science* vol. 35, 689-709, 1997.