Assigning Codes in Wireless Networks: Bounds and Scaling Properties.

Roberto Battiti *, Alan A. Bertossi *, and Maurizio A. Bonuccelli †

October 10, 1996

Abstract

In the Code Division Multiple Access (CDMA) framework, collisions that can occur in wireless networks are eliminated by assigning orthogonal codes to stations, a problem equivalent to that of coloring graphs associated to the physical network.

In this paper we present new upper and lower bounds for two versions of the problem (hidden *and* primary collision avoidance – HP-CA – or hidden collision avoidance only – H-CA). In particular, optimal assignments for special topologies and heuristics for general topologies are proposed. The schemes show better average results with respect to existing alternatives. Furthermore, the gaps between the upper bound given by the heuristic solution, the lower bound obtained from the maximum-clique problem, and the optimal solution obtained by branch & bound are investigated in the different settings.

A scaling law is then proposed to explain the relations between the number of codes needed in Euclidean networks with different station densities and connection distances. The substantial difference between the two versions HP-CA and H-CA of the problem is investigated by studying the probabilistic distribution of connections as a function of the distance, and the asymptotic size of the maximum cliques.

1 Introduction

Wireless Networks (WN) were first demonstrated in 1969 at the University of Hawaii [1] and since then have greatly increased their presence and importance in the computer network scenario. In a WN, different computers are linked by radio frequencies and therefore equipped with radio transmitters and receivers to broadcast outgoing packets and to listen to incoming ones. One often deals with *multihop* networks, where the communication between two nodes not directly connected happens by having packets received and later retransmitted by intermediate stations before reaching their final destination, with a "store and forward" scheme.

Unconstrained transmission in broadcast media may lead to the time overlap of two or more packet receptions, called *collision*, resulting in damaged useless packets at the destination and therefore in increased delays and bandwidth usage caused by retransmissions. Collisions can be *direct (or primary)*, when they are due to the transmission of stations which can hear

*R. Battiti and A. Bertossi are with the Dipartimento di Matematica, Università di Trento, 38050 Povo (Trento), Italy. E-mail: battiti@science.unitn.it, bertossi@science.unitn.it

†M. Bonuccelli is with the Dipartimento di Informatica, Università di Pisa, Corso Italia 40, 56125 Pisa, Italy. E-mail: bonucce@di.unipi.it

each other, and *hidden (or secondary)*, when stations outside the hearing range of each other transmit to the same receiving stations.

CDMA protocols (*code division multiple access*) can be used to totally avoid collisions [16, 17, 9], by the use of spread spectrum communication techniques and by the proper assignment of *orthogonal codes*. CDMA protocols require that either transmitters or receivers are code-agile, i.e., able to communicate over a multitude of codes. These codes share the fixed channel capacity allocated to the network in the design stage. Thus, their number must not exceed a given bound, and their use has to be minimized.

In this paper, we investigate the problem of minimizing the codes needed to eliminate primary and/or hidden collisions in a Wireless Network with transmitter-oriented code assignment (TOCA) protocols [14, 11], where the transmitters are code-agile. In [14, 8] the problem is introduced and some two-hops assignments for regular networks and heuristics for general network topologies are proposed. The investigation of this paper builds upon the work in [4], where the problem is investigated from a graph-theoretic point of view, optimal algorithms for additional special network topologies, and fast suboptimal heuristic algorithms for general topologies are proposed and experimentally evaluated.

In order to highlight the important differences between the two versions of the problem (avoidance of only hidden, or both hidden and primary collisions) and the effects caused by the two-dimensional connectivity pattern, a simplified version of the "real-world" problem is considered, where stations are distributed randomly on a square, and connected if and only if their Euclidean distance is less than a fixed radius. While we are not claiming an immediate transfer of our results to complex real-world topologies, we argue that some qualitative differences in the problem solution and the derived scaling laws will be recognizable also in practical applications. In addition, the bounds given in the paper permit a performance evaluation of different code-assignment techniques in different contexts.

The paper is organized in the following parts. In Sec. 2 the problem is formulated and some relevant existing algorithms are summarized. In Sec. 3 new optimal code assignments for special topologies are presented. In Sec. 4 a new heuristic algorithm for general topologies is introduced and used to derive upper bounds. In addition, the gaps between these values, the maximum cliques lower bounds (Sec. 4.2), and the exact branch & bound solutions are investigated (Sec. 4.3). The spatial connectivity patterns, the scaling laws, and the notable differences between the two versions H-CA and HP-CA of the problem are further investigated in Sec. 5.

2 Formulation and previous results

Let us now describe how the physical network is mapped to an *associated graph*, so that the collision avoidance problem becomes equivalent to a *graph coloring* problem, and let us summarize some existing results that are relevant for the following discussion.

2.1 Problem Formulation

A WN can be modeled as an undirected graph $G_{WN} = (V, E)$, where the set of vertices (or nodes) $V = \{1, ..., n\}$ represents the set of stations, and the set of edges E the common channel property between pairs of stations. More precisely, there is a one-to-one mapping of the stations onto the vertices in V, and two vertices i and j in V are joined by an undirected edge $(i, j) \in E$ if and only if their corresponding stations can hear each other transmission. In such a case, the vertices (or, equivalently, the stations) are called *physically adjacent*. Thus, the graph G_{WN} represents the physical network topology. A path between the vertices i and j is a sequence $i = v_1, v_2, ..., v_h = j$ of vertices such that $[v_k, v_k + 1] \in E$ for $k = 1, 2, ..., h - 1$; its length is $h - 1$, namely the number of edges appearing in it. The distance d_{ij} between two vertices i and j of G_{WN} is the length of the shortest path between i and j; it equals the minimum number of hops that a packet must undergo in a communication between stations i and j.

Two vertices (stations) i and j can generate a primary collision if and only if $d_{ij} = 1$, a hidden collision if and only if they are two hops away, namely, when $d_{ij} = 2$.

Now, the collisions can be eliminated if i and j transmit on different orthogonal codes and, after equating codes with colors, the problems can be mapped to coloring problems on the associated graphs G_{HP} and G_H obtained as follows. The vertex set V of the two graphs is given by the set of stations, while the edges differ in the two cases:

- Problem H-CA (Hidden Collision Avoidance)
  Assign codes to vertices in V so that every pair of vertices at distance two is assigned a couple of different codes and the minimum number of different codes is used.

  The associated graph is $G_H = (V, E_H)$, where;

  $$(i, j) \in E_H \text{ iff } d_{ij} = 2$$

  where d is the distance in the physical graph G_{WN}

- Problem HP-CA (Hidden and Primary Collision Avoidance)
  Assign codes to vertices in V so that every pair of vertices at distance one or two is assigned a couple of different codes and the minimum number of different codes is used.

  The associated graph is $G_{HP} = (V, E_{HP})$, where;

  $$(i, j) \in E_{HP} \text{ iff } d_{ij} \leq 2$$

  where d is the distance in the physical graph G_{WN}

In the following, $N_G(i)$ will denote the neighborhood of node i in graph G, where $j \in N_G(i)$ iff $(i, j) \in E_G$, the set of edges in graph G.

2.2 Previous results

Let us briefly summarize some algorithms that are relevant for the current study. The focus of this paper is on bounds and therefore the distributed and dynamic realization of the algorithms is not discussed (see [8] for a study of these issues). Let us note that the number of codes assigned does not depend on the centralized or distributed implementation of the algorithm and therefore the centralized version can be used for studying the relative performance of different algorithms, when the number of used codes is considered.

Because the problem of finding the optimal code assignment is computationally intractable [8, 4], the proposed heuristics generate quickly (in polynomial time) code assignments which eliminate collisions but do not use the minimum number of codes. The extreme simplicity of the heuristics, however, makes them attractive for actual utilization.

The time-slot assignment problem (shown to be equivalent to the frequency assignment problem in [6]) is formulated as a graph coloring problem in [7] and solved through a distributed algorithm in [8]. The centralized version of the Chlamtac & Pinter algorithm (denoted as CP-CODE-ASSIGNMENT) is a coloring procedure, where nodes are examined in decreasing order of their identifiers (ID) and a given node i is assigned the smallest code not already assigned to nodes in its neighborhood in the associated graph. [8] considers the HP-CA version of the problem so that the associated graph is G_{HP}, but a similar algorithm can clearly be used also for the H-CA problem.

If the node ID's are attributed without any relation to the network characteristics, the results of this procedure are analogous to those obtained through a random choice of the next node to be colored at each step.

Without loss of generality, let us assume that the ID's are selected from $1, ..., n$ and, for a simpler comparison with the following algorithms, let the nodes be considered in *increasing* order of their ID's. Let us denote as $code[i]$ the code assigned to node i.

```
CP-CODE-ASSIGNMENT
    code[1] ← 1
    for ID ← 2 to n do
        code[ID] ← SMALLEST-NOT-ASSIGNED-IN ( N(ID) )
```

Figure 1: The CP-CODE-ASSIGNMENT algorithm

$N(ID)$ is the neighborhood in the appropriate associated graph (in the terminology of [8] for the HP-CA problem, the term *locality* is used for $N(ID)$). The SMALLEST-NOT-ASSIGNED-IN routine can be realized as illustrated in Fig. 2. ASSIGNED $[i]$ is a boolean variable, false at the beginning of the code-assignment algorithm, that is set to true as soon as node i is assigned a code.

```
SMALLEST-NOT-ASSIGNED-IN ( N )
    { N is a set of nodes , the function returns a valid code }
    AssignedCodeSet ← ∅
    for each j ∈ N do
        if ASSIGNED [j] then AssignedCodeSet ← AssignedCodeSet ∪ code[j]
    k ← 1
    while k ∈ AssignedCodeSet do
        k ← k + 1
    return k
```

Figure 2: The SMALLEST-NOT-ASSIGNED-IN routine

Simple centralized and distributed heuristic algorithms for the H-CA problem in general network topologies are proposed in [4], that investigates different *ordering schemes* of the nodes: random ordering, increasing or decreasing number of neighbors in the associated graph G_H or G_{HP}, and increasing or decreasing number of neighbors in the physical graph G_{WN}.

Let the n stations be named with ID's belonging to $1, 2, ..., n$, according to any specified criterion. Moreover, let $NS(i)$ be the set of neighboring stations j in the associated graph with index smaller than i. Depending on the two possible versions of the problem, the associated graph is G_H or G_{HP}.

In a similar way as in the CP-Code-Assignment algorithm, stations are considered sequentially. Station i is assigned the lowest code not assigned in $NS(i)$. The algorithm can be described in pseudo-language as follows:

```
BB-Code-Assignment
    Order the nodes according to the selected criterion
    code[1] ← 1
    for i ← 2 to n do
        code[i] ← Smallest-Not-Assigned-In ( NS(i) )
```

Figure 3: The centralized version of the BB-Code-Assignment algorithm

When the set $NS(i)$ is properly implemented, e.g. by means of a boolean array, the overall running time is $O(n^2)$. From the experiments in [4], the best ordering criterion is based on decreasing degree in the associated graph (called "D2 decreasing" in [4]). The explanation for the success of this choice is that nodes with higher degree tend to be more constrained in their choice of colors if they are colored in successive steps (because of the large neighborhood), with the danger that the number of used colors needs to be increased.

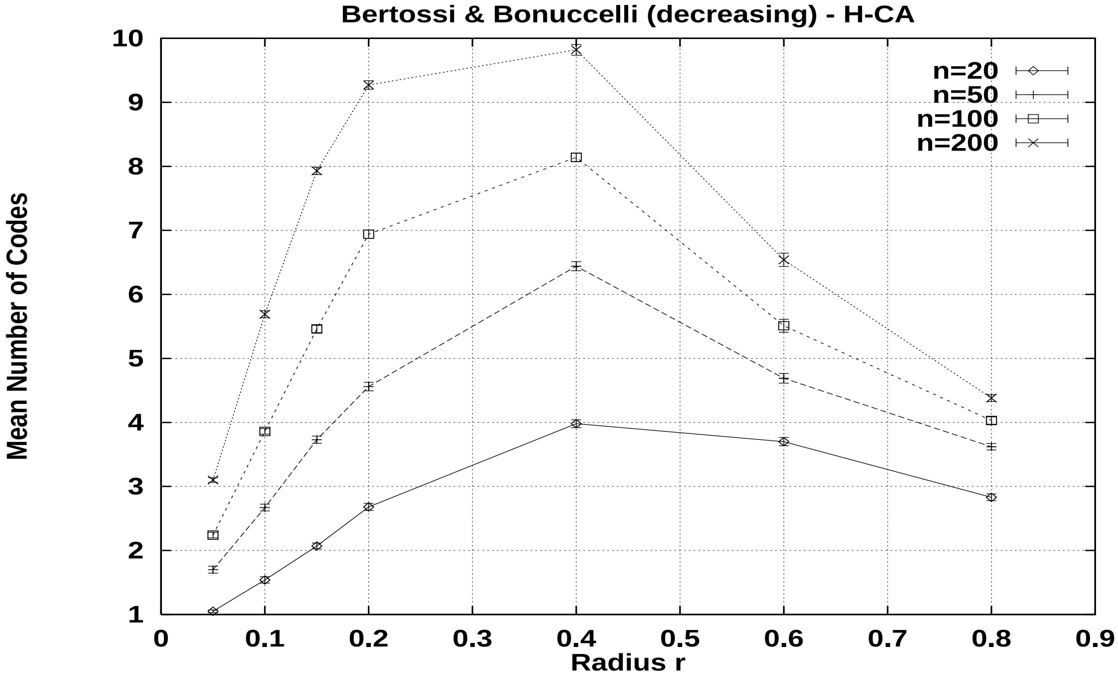
2.3 Benchmark instances

The current mathematical tools available for an average algorithm performance evaluation can be employed only for very simple and regular network topologies. Thus, the performance of different algorithms is assessed through the statistical analysis of simulation experiments.

The algorithms are run on finite random networks with varying connectivity patterns and number of stations. We consider n station Euclidean networks, with n = 20, 50, 100, and 200. The positions of the stations is given by n randomly generated points in the unit square $[0,1] \times [0,1]$. Each point is generated as a pair of random real numbers in the range $[0,1]$, which correspond to its Cartesian coordinates. For a given set of n points, the edges in the physical network G_{WN} are given by all pairs of points whose Euclidean distance is not larger than r (henceforth called "radius"). Several values of r are considered, namely 0.05, 0.10, 0.15, 0.2, 0.4, 0.6, and 0.8. From the physical network one then constructs the associated graph, including the corresponding neighborhoods.

For each value of n and r, 100 networks are generated, and the different algorithms are run.

The average results obtained on this benchmark by the BB-Code-Assignment algorithm with the best ordering (D2 decreasing) are reported in Fig. 4 for each pair of n and r. The results for the two variants of the problem H-CA and HP-CA are shown in the top and bottom graphs, respectively. Error bars equal to $\sigma/\sqrt{100}$, σ being the standard deviation, are also reported. Because of the small statistical errors on the measured average values, the bars are hardly visible on some graphs.



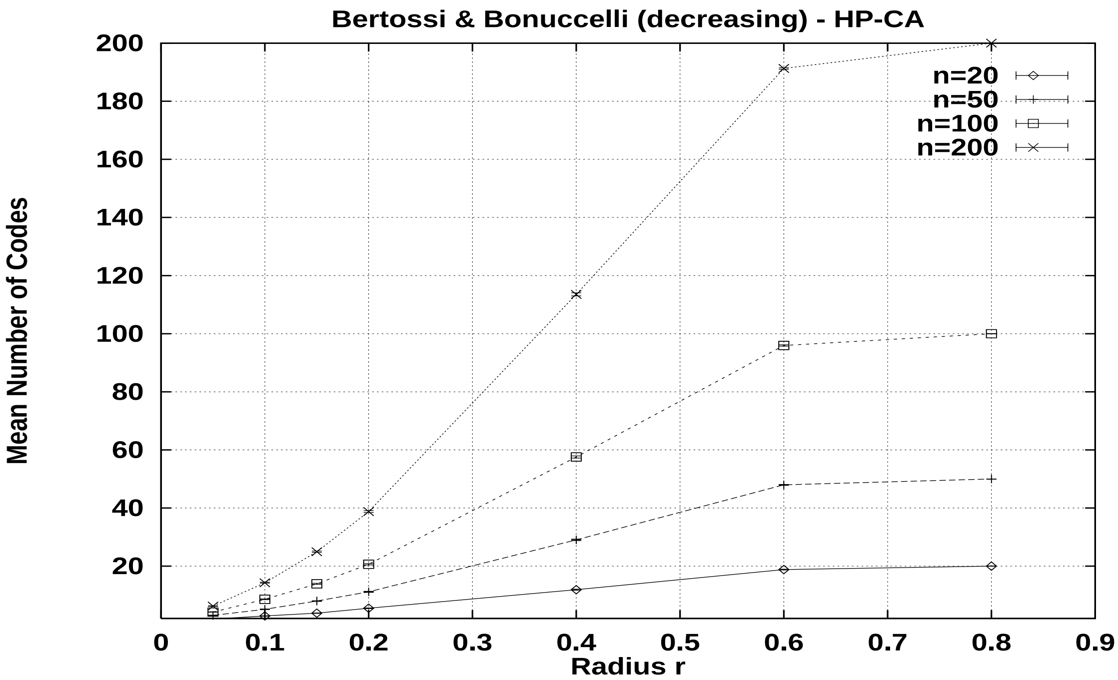


Figure 4: Bertossi - Bonuccelli (D2 decreasing). Problems H-CA (above) and HP-CA (below).

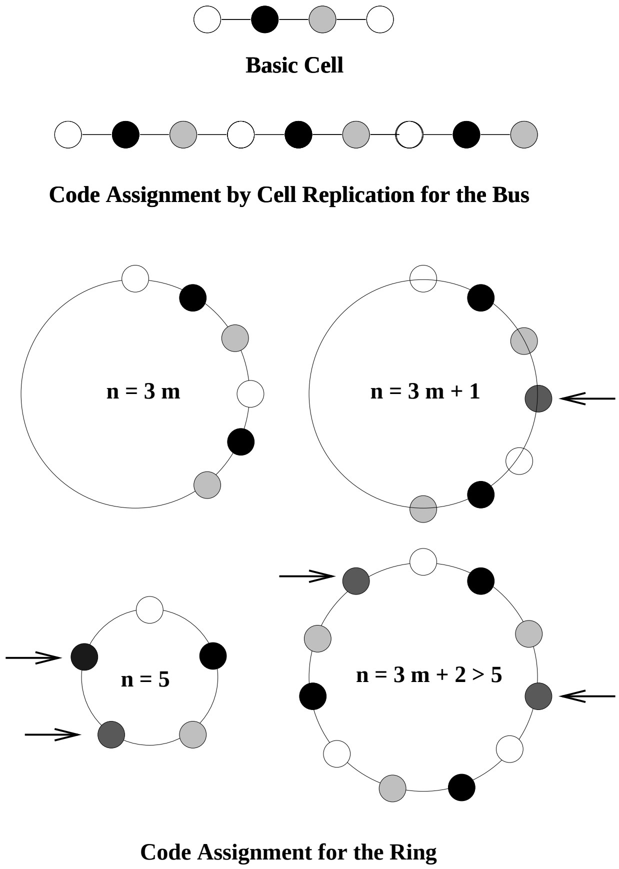


Figure 5: Optimal code assignment for the bus and ring networks.

3 Optimal algorithms for special topologies

In this section, new optimal algorithms for code assignment in special networks are presented. The role of *maximum clique* lower bounds is apparent in these regular topologies.

Let $G = (V, E)$ be an arbitrary undirected graph and $G(S) = (S, E \cap S \times S)$ *the subgraph induced by* S, where S is a subset of V. A graph $G = (V, E)$ is *complete* if all its vertices are pairwise adjacent, i.e. $\forall i, j \in V, (i, j) \in E$. A *clique* K is a subset of V such that $G(K)$ is complete. A clique of size k in the associated graph G_H or G_{HP} implies that at least k different codes have to be used for the given problem. In other words, the size of the maximum clique is a lower bound for the code assignment problem, as noted by Makansi [14].

This result allows optimal code assignments for some special kinds of regular network topologies to be derived for the H-CA problem. In particular, optimal assignments are proposed in [14] for busses, hexagonal, and grid topologies, and in [4] for rings and trees.

Let us now present optimal code assignments for the HP-CA problem. As in [14], a basic cell of the physical graph G_{WN} is identified for each special topology such that the complete code assignment can be obtained by replicating the code assignment for the basic cell.

Figures 5, 6, 7, and 8 show the basic cells and the corresponding code assignments for the bus and ring, grid, hexagonal grid, and hexagonal mesh. For all topologies apart from the ring, the number of codes used is equal to the maximum clique lower bound in the associated graph G_{HP}. Moreover, it is easy to see that in these cases the size of the maximum clique in

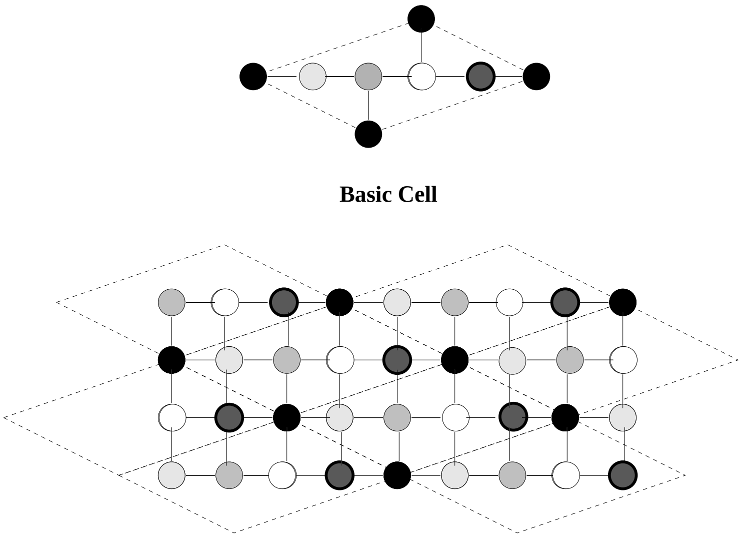


Figure 6: Optimal code assignment for the grid network.

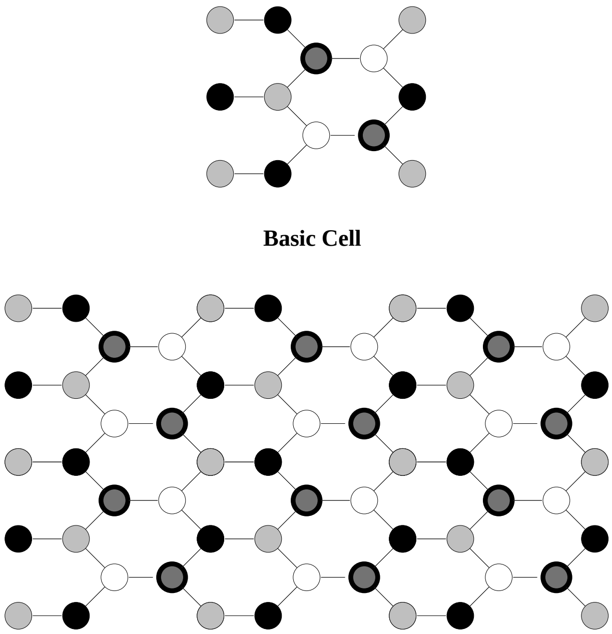


Figure 7: Optimal code assignment for the hexagonal grid network.

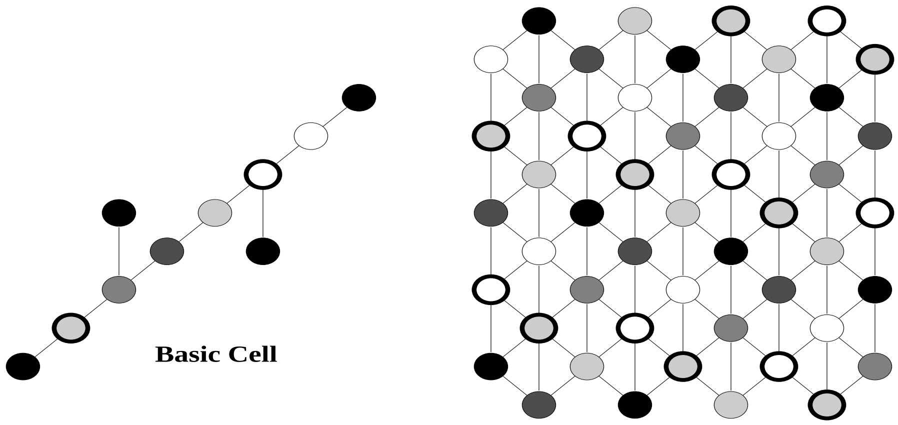


Figure 8: Optimal code assignment for the hexagonal mesh network.

G_{HP} is equal to one plus the maximum node degree of G_{WN}. The relevance and effectiveness of using maximum clique sizes for lower-bounding the number of codes will be confirmed in the analysis of the HP-CA problem in general topologies, shown in the next sections.

For the ring topology, the optimal code assignments are slightly more tricky because they depend on the number n of nodes. If n is a multiple of 3, the assignment is the same as that for the bus. Otherwise, an additional code is needed except for the particular case of $n = 5$, when 5 codes are necessary because the associated network is completely connected. In a similar way, different special cases are present for the multiring topology, depending on the number of nodes and the degree of the nodes. The extension to the multiring topology is straightforward and omitted for brevity.

4 Saturation degree code assignment

In this section new and better performing heuristics are proposed for the code assignment problem in general topologies, and used to obtain upper bounds on the number of codes needed. Let $MaxCodes$ be the maximum number of codes available. The basic design principles of the "saturation degree" coloring heuristic proposed by [5] is that the first nodes to be colored are those that have more colors already assigned to nodes in the neighborhood. The motivation is that these nodes have a more constrained choice and therefore a higher risk that all $MaxCodes$ colors will be present in the neighborhood in future steps, unless they are colored immediately. The heuristic is used in [5] to choose the next branching node in the branch & bound algorithm DSATUR. Korman [13] recommended choosing a node with highest degree in the uncolored subgraph and Kubale and Jackowski [12] validate the choice in their experiments.

4.1 The saturation degree heuristic

The SATURATION-DEGREE-CODE-ASSIGNMENT algorithm is illustrated in Fig 9. Let us denote as $N(i)$ the set of neighbors of node i in the associated graph (either G_H or G_{HP}, depending on the problem). As usual, $code[i]$ is the code assigned to node i, and the set $ToBeAssigned$ contains the yet-unassigned nodes. In addition, to each node i one associates

```
Saturation-Degree-Code-Assignment
1   Order nodes according to decreasing degree in associated graph
2   ToBeAssigned ← {1, 2, ..., n}
3   for i ← 1 to n do
4       NAssignedNeighbors[i] ← 0
5       NeighCodes[i] ← ∅
6   while ToBeAssigned ≠ ∅ do
7       MaxNeighCodes ← −1
8       MaxAssignedNeigbors ← −1
9       for each i ∈ ToBeAssigned do
10          if |NeighCodes[i]| > MaxNeighCodes then
11              MaxNeighCodes ← |NeighCodes[i]|
12              MaxAssignedNeigbors ← NAssignedNeighbors[i]
13              v* ← i
14          else if |NeighCodes[i]| = MaxNeighCodes then
15              if NAssignedNeighbors[i] > MaxAssignedNeigbors then v* ← i
16      code[v*] ← lowest code not in NeighCodes[i]
17      ToBeAssigned ← ToBeAssigned \ {v*}
18      for each j ∈ N(v*) ∩ ToBeAssigned do
19          NAssignedNeighbors[j] ← NAssignedNeighbors[j] + 1
20          NeighCodes[j] ← NeighCodes[j] ∪ code[v*]
```

Figure 9: The Saturation-Degree-Code-Assignment algorithm

a set $NeighCodes[i]$ containing the codes already assigned to nodes in the neighborhood, and an integer $NAssignedNeighbors[i]$ given by the number of neighbors that have been assigned codes. At each iteration, the node v^* that is assigned a code is one with the largest number of codes already assigned in the neighborhood (lines 7–13). Ties between nodes are broken by preferring the nodes with the largest number of assigned neighbors (lines 14–15).

As soon as a code is assigned to node v^*, all uncolored neighbors j must increase by one the number of colored neighbors $NAssignedNeighbors[j]$ (line 19) and update the set of neighboring codes $NeighCodes[j]$ (line 20).

With a proper implementation of the sets, the worst-case computational complexity of the above algorithm is $O(n^2)$.

The results obtained for the H-CA and HP-CA problems when the nodes are ordered according to decreasing degree in the associated graphs are illustrated in Fig. 10. If these results are compared with those obtained by the BB-Code-Assignment algorithm in Fig 4, a sizable reduction in the average number of codes used can be observed for the H-CA problem (e.g., for the n=200 instances the reduction is of 10.5% for r=0.1, 15.6% for r=0.2 and r= 0.4, 25.6% for r=0.6, 8.6% for r=0.8).

On the contrary, no appreciable reduction is obtained on the HP-CA problem: the average variation between the number of codes needed by Saturation-Degree-Code-Assignment and those needed by BB-Code-Assignment is always less than 1%. This result indicates a radical difference between the two versions HP-CA and H-CA of the code assignment problem, a difference that will be examined from different viewpoints in the following sections.

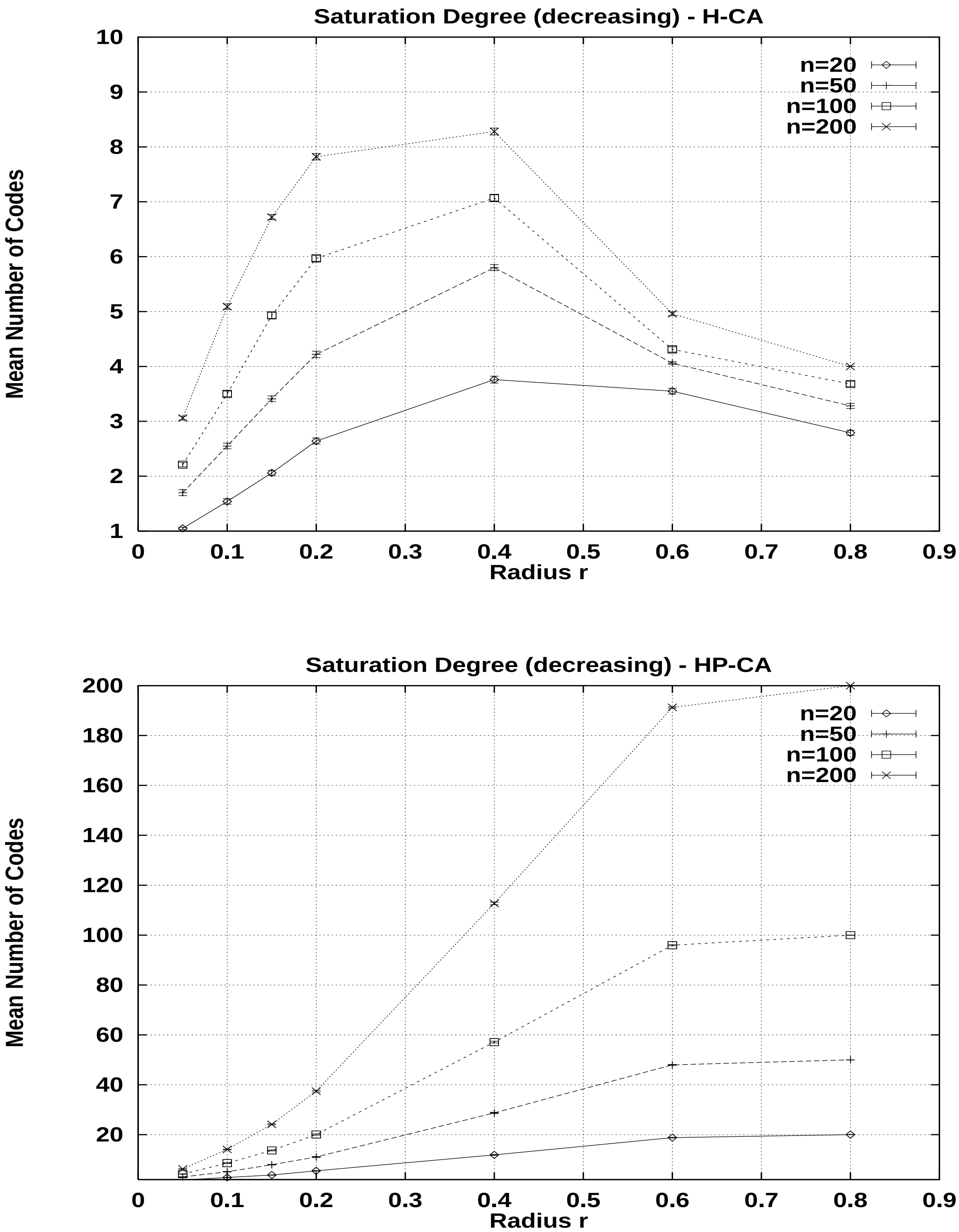


Figure 10: Saturation degree (D2 decreasing). Problems H-CA (above) and HP-CA (below).

4.2 Maximum-clique lower bound

In the previous subsection the algorithm SATURATION-DEGREE-CODE-ASSIGNMENT has been shown to perform better than BB-CODE-ASSIGNMENT , on the average. In addition to relative comparisons it is now of interest to compare its performance with lower bounds and, when possible, with the globally optimal solutions of the different instances.

The Maximum Clique (MC) problem asks for a clique of maximum cardinality. As noted in Sec. 3, the size of the maximum clique in the associated graph G_H or G_{HP} is a lower bound for the code assignment problem.

MC is an NP-hard problem, furthermore strong negative results have been shown about its approximation properties. In particular, if $P \neq NP$, no polynomial time algorithm can approximate the Maximum Clique problem within a factor $n^{\frac{1}{4}-\epsilon}$, where n is the number of nodes of the graph [3].

These theoretical results stimulated a research effort to design efficient heuristics for this problem [10]. In particular, a new *reactive* heuristic for the Maximum Clique problem has been proposed in [2]: Reactive Local Search (*RLS*). *RLS* complements local-neighborhood-search with *prohibition-based diversification* techniques, where the amount of diversification is determined in an automated way through a *feedback* scheme. *RLS* provides a significantly better performance than all competitive heuristics presented at the second DIMACS implementation challenge [10], considering both the obtained clique sizes and the CPU times utilized.

Let us now study how the performance of SATURATION-DEGREE-CODE-ASSIGNMENT compares with respect to the lower bound given by the size of the maximum clique in the associated graph. The average results obtained by the *RLS* algorithm on the H-CA and HP-CA problems are illustrated in Fig. 11. If the results of SATURATION-DEGREE-CODE-ASSIGNMENT (see Fig. 10) are compared with the maximum-clique lower bound, a relative gap of 30-40% can be observed for the H-CA problem, especially for instances with large n and small r, see for example the instances with $n = 200$, $r = 0.2$. In addition, no clique with a size larger than five has been observed in all tests: it appears that only small cliques (with up to five nodes) are present in the associated graph G_H. For each couple of n and r values the frequency with which a larger clique is generated (if it can be generated) appears to be less than about 1%.

The results are very different for the HP-CA problem, where the saturation degree algorithm performs almost in an optimal way. In this case the lower bound obtained by finding the maximum clique corresponds exactly to the number of colors found by the used heuristic, in most of the cases. In fact, the average values obtained in the two cases are hardly distinguishable (and the difference is always less than 1%).

Four implications can be derived for HP-CA:

- i) the algorithm *RLS* finds the optimal clique size in most cases;
- ii) the minimum number of different codes required for problem HP-CA tends to coincide with the size of the maximum clique, a result that agrees with those of Sec. 3;
- iii) simple algorithms like BB-CODE-ASSIGNMENT , provided that the "D2 decreasing" ordering is used, achieve results that are statistically undistinguishable from those of smarter heuristics like SATURATION-DEGREE-CODE-ASSIGNMENT ;
- iv) the minimum number of codes or a very close approximation thereof can be found in a polynomial number of iterations and, furthermore, in very small CPU times.

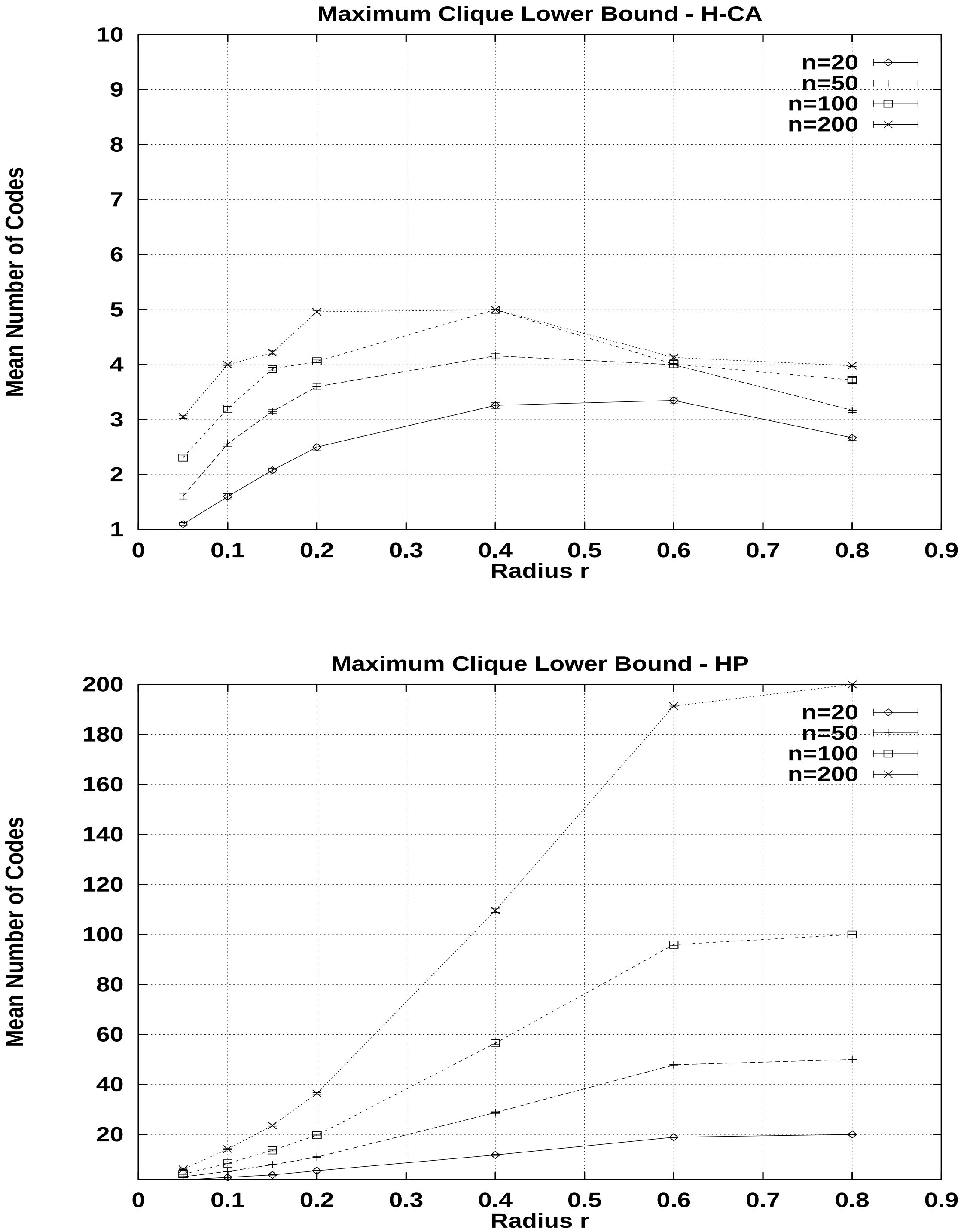


Figure 11: Maximum clique lower bound. Problems H-CA (above) and HP-CA (below)

The minimum number of codes or a very close approximation thereof can be found in a polynomial number of iterations and, furthermore, in very small CPU times.

4.3 Comparison with exact branch & bound

The coloring problems associated to the H-CA and HP-CA instances have been solved by using a branch and bound algorithm similar to DSATUR [5]. The C code corresponding to the algorithm has been obtained from M. A. Trick [1], implementation details are described in [15]. DSATUR works by dividing an instance into a series of subproblems, given by partial colorings of the graph. At each step an upper bound UB is available on the number of colors required. If a subproblem uses k colors, and k is at least UB, the subproblem can be fathomed. Otherwise, if every node in the graph is colored, and $k < UB$, then a better coloring has been found and UB is set to k. If the graph is not completely colored, and the current number of colors used is less than UB, new subproblems are created after choosing a node i for branching. For each feasible color for i (out of the k used in the subproblem), a new subproblem is created by assigning that color to i. In addition, a subproblem is created with i assigned color $k+1$. The algorithm terminates when no subproblems are left: UB then gives the coloring number of the graph. The rules for picking a branching node have already been described in Sec. 4.

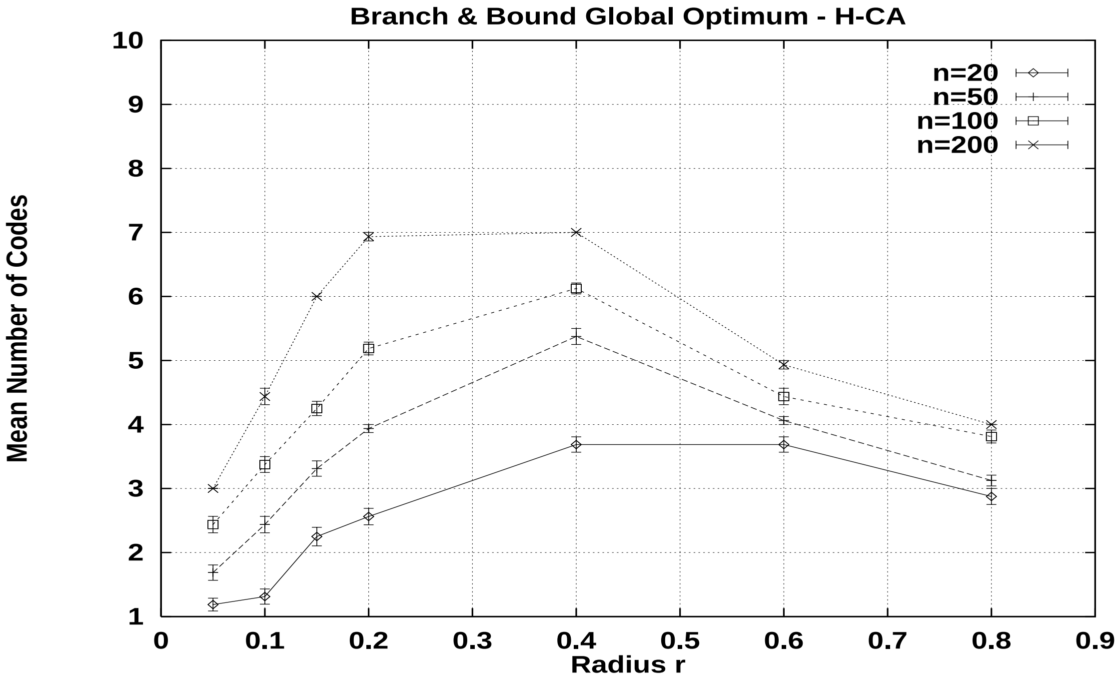
The results are illustrated in Fig. 12. For the H-CA problem, the unreliability of the maximum clique size as a tight lower-bounding measure is confirmed: for example, the relative distance between the lower bound and the optimal solution is of about 40% for the $n = 200$, $r = 0.2, 0.4$ instances. On the other hand, the relative performance gap between the branch & bound globally optimal solution and the SATURATION-DEGREE-CODE-ASSIGNMENT heuristic on the H-CA problem is not large: close approximations are found. As an example, the larger gaps for the $n = 200$, $r = 0.2, 0.4$ instances range between 13% and 18%. In absolute terms, the largest gaps are of about one additional code. Considering the enormous difference in computing times (seconds versus hours), the SATURATION-DEGREE-CODE-ASSIGNMENT heuristic is therefore suggested as a practical heuristic with a limited average performance loss. Let us note that $n = 200$ is close to the largest problem size that can be solved by branch & bound, in fact some problems had to be stopped after 24 hours of computing on a state-of-the-art workstation (and these suboptimal results are not included in the given averages).

The results for the HP-CA problem are completely different. As predicted by the closeness between the maximum clique lower bound and the solution provided by the different heuristics (for example SATURATION-DEGREE-CODE-ASSIGNMENT), the branch & bound solution tends to require the same number of codes as those of the heuristic (the difference is always less then 1%).

5 Explaining the difference between HP-CA and H-CA

The differences between the HP-CA and H-CA versions of the code assignment problem are now investigated in more detail, by studying the spatial distribution of connections and the expected size of the cliques in the graphs. A scaling law is then identified to relate characteristics of instances with different n and r values.

[1] Available on the World Wide Web at `http:/ /mat.gsia.cmu.edu/COLOR/solvers/trick.c`



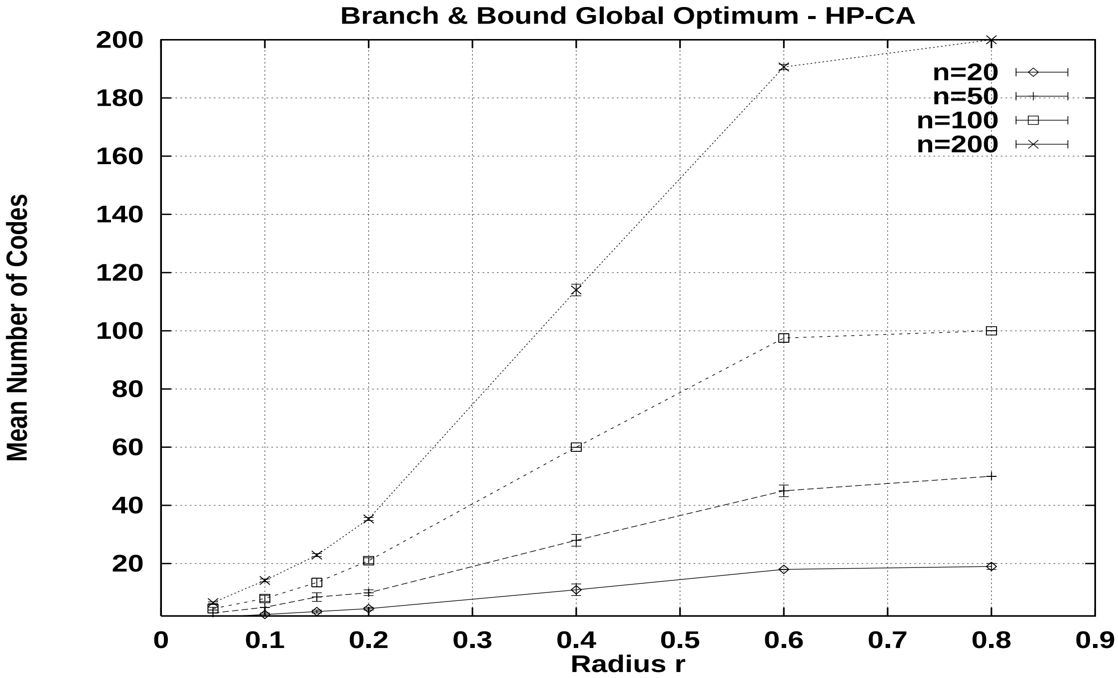


Figure 12: Comparison with exact branch & bound algorithm. Problems H-CA (above) and HP-CA (below)

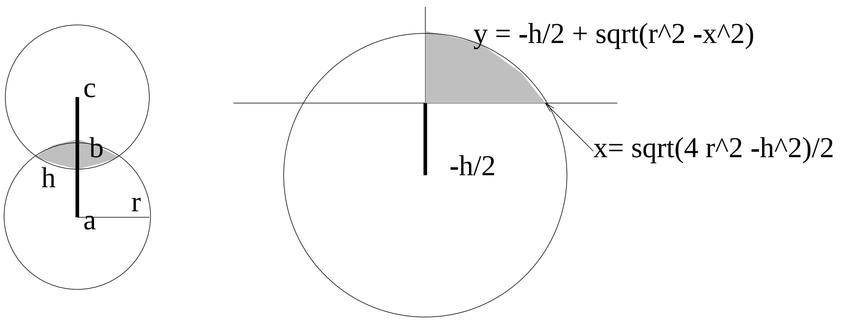


Figure 13: H-CA: calculating the probability of a two-hop connection as a function of h, with $r < h \leq 2\,r$.

5.1 Spatial distribution of connections

Because of the definition of the physical graphs, the connections in the associated graphs for the H-CA and HP-CA problems are localized in an Euclidean neighborhood of a given station. In particular, no connection is present in the associated graphs if the Euclidean distance between two stations is larger than $2\,r$. Let us now study in detail the spatial connection density for a given node as a function of the Euclidean distance of the connected stations. To simplify matters, let us neglect "border effects" and let us assume that the station is sufficiently far from the borders of the $[0,1] \times [0,1]$ square, or that r is sufficiently small, so that the circle of radius $2r$ centered at the station is completely contained in the square.

H-CA

Let us define as $p_H(a,c|h)$ the probability that two stations a and c are connected in G_H, given that their distance is h, i.e., the probability that they are not physically connected and that there is a third station b that is physically connected to both. Immediately one derives that $r < h \leq 2r$, otherwise either the stations are directly connected or they cannot be two-hops connected.

The probability $p_H(a,c|h)$ is derived in steps.

First, the probability p_b that a point b thrown onto the square with uniform probability achieves the required two-hops connection is equal to the shaded area in Fig. 13 (left), that is:

$$
\begin{aligned}
p_b &= 4\int_0^{\sqrt{4r^2-h^2}/2} \left(-\frac{h}{2} + \sqrt{r^2-x^2}\right) dx && (1)\\
&= \frac{-\left(h\,\sqrt{-h^2+4\,r^2}\right)}{2} + 2\,r^2\,\arctan\left(\frac{\sqrt{-h^2+4\,r^2}}{h}\right) && (2)\\
&= 2\,r^2\left(\arctan\left(\sqrt{\left(\frac{2r}{h}\right)^2-1}\right) - \left(\frac{h}{2r}\right)^2\sqrt{\left(\frac{2r}{h}\right)^2-1}\right) && (3)
\end{aligned}
$$

Let us now consider n uniformly distributed points (in the assumption that n is large the difference between n and $n-2$ is negligible). The probability $p_H(a,c|h)$ is the probability that at least one among the n points falls in the shaded area, i.e., one minus the probability that all points are not in the region:

$$
\begin{aligned}
p_H(a,c|h) &= 1-(1-p_b)^n \approx n p_b & (4)\\
&\approx n\,2\,r^2\left(\arctan\left(\sqrt{\left(\frac{2r}{h}\right)^2-1}\right)-\left(\frac{h}{2r}\right)^2\sqrt{\left(\frac{2r}{h}\right)^2-1}\right) & (5)
\end{aligned}
$$

where the last approximation is valid when $n\ p_b << 1$. The above equation holds for h such that $r < h \leq 2\ r$, the probability is zero otherwise. To obtain the connection probability density as a function of the distance h, the above quantity must be multiplied by $n2\pi h$ (on the average, $n2\pi h\ dh$ points fall in a strip at distance between h and $h+dh$).

Finally, after integrating between r and $2r$ one obtains the expected number of two-hops connections E_H:

$$
\begin{aligned}
E_H &= \int_r^{2r} n\,2\,\pi\,h\,p_H(a,c|h)\,dh & (6)\\
&\approx \int_r^{2r} n^2\,2\,\pi\,h\,2\,r^2\left(\arctan\left(\sqrt{\left(\frac{2r}{h}\right)^2-1}\right)-\left(\frac{h}{2r}\right)^2\sqrt{\left(\frac{2r}{h}\right)^2-1}\right)dh & (7)\\
&\approx 16\,\pi\,(n\,r^2)^2\int_{\frac{1}{2}}^{1} x\left(\arctan\left(\sqrt{\frac{1}{x^2}-1}\right)-x^2\left(\sqrt{\frac{1}{x^2}-1}\right)\right)dx & (8)\\
&\approx \frac{3\sqrt{3}}{4}\,\pi\,(n\,r^2)^2 & (9)
\end{aligned}
$$

Clearly, this quantity is equal to the expected degree of a node in G_H that is located in the interior of the square. If the assumptions adopted in the approximation of eqn. 4 are not valid, one could avoid the approximation and, for example, find an approximate value for the integral with standard numerical integration packages.

HP-CA

The probability $p_{HP}(a,c|h)$ can be easily obtained from the previous case. It is sufficient to consider that, if $0 \leq h \leq r$, the points a and c are directly connected, while if $r < h \leq 2\ r$ one has $p_{HP}(a,c|h) = p_H(a,c|h)$. Therefore one has:

$$
p_{HP}(a,c|h) = \begin{cases} 1 & \text{if } 0 \leq h \leq r \\ \approx n\,2\,r^2\left(\arctan\left(\sqrt{\left(\frac{2r}{h}\right)^2-1}\right)-\left(\frac{h}{2r}\right)^2\sqrt{\left(\frac{2r}{h}\right)^2-1}\right) & \text{if } r < h \leq 2r \end{cases} \quad (10)
$$

where the approximation is that already used in eqn. 5.

After integrating $n\ 2\ \pi\ h\ p_{HP}(a,c|h)$ between 0 and 2 r, one obtains that the expected degree E_{HP} of a node in G_{HP} is:

$$
\begin{aligned}
E_{GH} &= \int_0^{2r} n\,2\,\pi\,h\,p_{HP}(a,c|h)\,dh & (11)\\
&\approx \pi\,n\,r^2 + \frac{3\sqrt{3}}{4}\,\pi\,(n\,r^2)^2 & (12)
\end{aligned}
$$

In both versions of the problem the expected degree is a function of the quantity $n\ r^2$, in particular, a quadratic dependence in the above quantity is present in E_H, while a linear and a quadratic term are present in E_{HP}. Now, there is a simple physical interpretation of the above dependence. The number of physical neighbors of a station is equal to $n\ \pi\ r^2$, on the average. In other words, the above results tell us that the degree in the associated graph is a function of the number of physical neighbors. In addition, the detailed functional dependence provides us with scaling laws. As an example, if both the number of stations n and the radius r change in such a way that the above product is unchanged: $n\ r^2\ =\ n'\ r'^2$, the average connectivity pattern does not change, and therefore other related quantities like the number of codes needed does not change, on the average. This theoretical prediction is tested experimentally in Sec. 5.3.

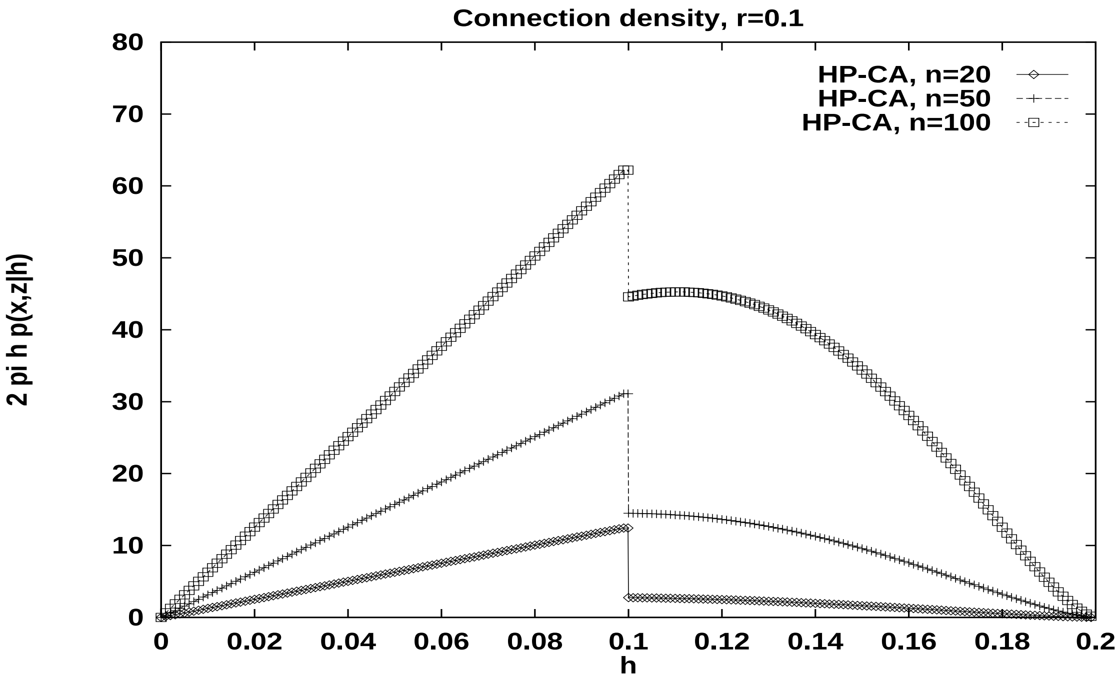


Figure 14: Connection probability density for HP-CA, $0 < h \leq 2\ r$, see text for details.

The connection probability density for the HP-CA problem as a function of h is shown in Fig. 14, for $r = 0.1$ and for values of n equal to 20, 50, and 100. The probability density is equal to $n\ 2\ \pi\ h\ p_{HP}(a, c|h)$, and the exact equation is used to calculate the $p_{HP}(a, c|h)$ values for the plot, see eqn. 4. It increases linearly for small h, then decreases with a discontinuity when h becomes larger than r, and therefore the associated edges derive only from two-hops connections. Let us note that the relative size of the jump in probability at the discontinuity decreases for larger n values. After that point direct connections do not exist, and the probability $p_{HP}(a, c|h)$ increases toward the asymptotic value of one for n tending to infinity ($\lim_{n\to\infty}(1 - p_b)^n = 0$ if $p_b > 0$). In other words, the jump tends to disappear for large n. The probability curves for H-CA can be easily derived: they are stuck at zero for $0 \leq h \leq r$ and they jump to the HP-CA values as soon as points can be connected by two hops (without being directly connected).

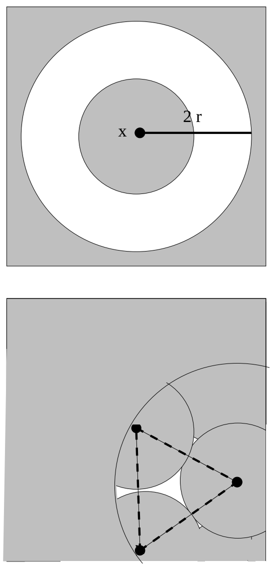


Figure 15: H-CA: limited size of maximum cliques.

5.2 Expected clique sizes

H-CA

The experimental results obtained in Sec. 4.2 (see Fig. 11, top) showed that the maximum clique size for the graphs associated to the H-CA problem tends to be very small. Apparently, no clique with a size larger than 5 was observed in all tests. In this section we show how the empirical findings are related to the geometrical structure of the problem.

A first result is that each clique in the G_H problem contains points that can always be included in a circle of radius $2\ r$ in the Euclidean space. This is immediately established: let us take a point x belonging to the clique. All points whose Euclidean distance from x is larger than $2\ r$ cannot be two-hops connected, and therefore they are not connected to it in G_H.

The situation is depicted in Fig. 15. The first image (top left) shows the admissible circular region (in white color) where the additional clique points can be placed. Let us now assume that a second clique point is placed as far as it can be from the point x, see the top right image, and that a third one is placed so that its distance from the first two is maximized. One obtains the situation illustrated in the bottom image of Fig. 15, where the clique edges in the associated graph are depicted with dashed lines. Now, the probability that two additional points (leading to a clique of size 5) fall in the admissible white region is already severely limited. The analysis of all possible configurations is quite complex but the given example clearly indicates the low reliability of clique sizes as an indicator of the number of codes needed in the H-CA problem. In other words the clique size, being limited to very small values, cannot be a tight lower bounds for the number of codes needed in a given H-CA task.

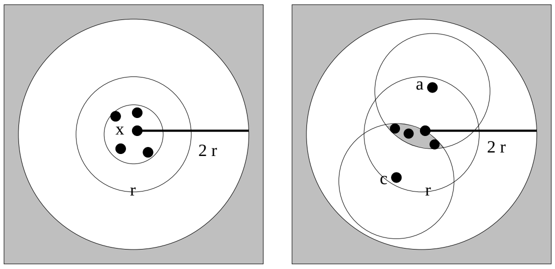


Figure 16: HP-CA: size of max cliques. Direct connections (left), two-hops connections (right).

HP-CA

The situation is completely different for the HP-CA problem. In particular one has the following theorems, that are valid in the assumptions that no "border effects" are present, i.e., that a circle of radius $2\,r$ around the node x does not meet the boundary of the $[0,1]\times[0,1]$ region. In fact smaller circles are sufficient in certain cases.

Theorem 5.1 *Given a node x, its physical neighbors at Euclidean distance not larger than $r/2$ form a clique in the associated graph G_{HP}, composed of points with direct physical connections. The expected size of the clique is $n\,\pi\,\left(\frac{r}{2}\right)^2$.*

The proof is as follows. Given two arbitrary points y and z in the above neighborhood, they are contained in a circle of diameter r, and therefore mutually connected by a one-hop connection. For the same reason x is connected to all such points. The expected size of the clique is given by the expected number of points that fall in a radius of length $r/2$ around x, i.e., $n\,\pi\,\left(\frac{r}{2}\right)^2$ $\square$.

The situation is illustrated in Fig. 16 (left).

Theorem 5.2 *Give a node x, its physical neighbors at Euclidean distance not larger than r form a clique in the associated graph G_{HP}, with a probability tending to one for n tending to infinity. The expected size of the clique is $n\,\pi\,r^2$.*

If two arbitrary points a and c fall in the given neighborhood, they have a probability larger than zero of being two-hops connected. In fact, the intersection of the two circles of radius r centered on them is not empty. Furthermore, from eqn. 4, the probability that a two-hops connection exists goes to one when n goes to infinity. In this limit, all such points will be two-hops connected. The expected size of the clique is given by the expected number of points falling in the neighborhood: $n\,\pi\,r^2$ $\square$.

The situation is illustrated in Fig. 16 (right).

Theorem 5.3 *Give a node $x \in G_{HP}$, the expected size of a clique that contains x is less than $n\,\pi\,(2\,r)^2$.*

The proof is trivial as soon as one notes that only the points at an Euclidean distance not larger that $2\,r$ can possibly be connected in G_{HP}. But the expected number of points in this area is $n\,\pi\,(2\,r)^2$ $\square$.

Let us comment the above theorems. First, let us note that they refer to the size of a single clique containing a node x, not to the maximum size of all cliques in the associated graph. This last quantity also reflects the fluctuations of the size around the mean value. Nonetheless, they are asymptotically correct estimates for large n, because the fluctuations are proportional to $\sqrt{n}$ so that their relative size becomes negligible.

If these results are compared with those for the H-CA problem, the situation is completely different. The size of the clique is not limited, on the contrary it grows in a manner proportional to $n\ r^2$. The experimental results tell us that the clique size is a very tight lower bound for the number of codes needed. Let us finally note that the results confirm the *scaling law*: the relevant quantities in the associated graphs (in this case the number of codes needed) are a function of $n\ r^2$.

5.3 A test of the scaling law

The scaling law stating that the number of codes needed is a function of the combination $n\ r^2$ is verified experimentally in fig. 17. Here the scaled curves obtained from the $n\ r^2$ functional relation are compared with the real curves. In detail, if $codes_n(r)$ gives the number of codes as a function of the radius r, other curves can be obtained by setting:

$$codes_{n'}(\sqrt{\frac{n}{n'}}\ r)\ =\ codes_n(r)$$

where the equation $n\ r^2\ =\ n'\ r'^2$ has been solved for r': $r' = \sqrt{\frac{n}{n'}}\ r$. In the above manner, the predicted curve for $codes_{100}(r)$ is derived from $codes_{50}(r)$, the predicted curve for $codes_{200}(r)$ is derived from $codes_{100}(r)$. As expected, the predicted curves are in close agreement with the real ones for small values of r (up to $r \approx 0.2$), while the predicted curves underestimate the number of codes required for large r values. In fact, when r is large, non-negligible border effects are present: more and more nodes are close to the border of the $[0,1] \times [0,1]$ region. In comparison with normal nodes, less connections are present on the border and less codes are therefore required.

6 Conclusions

The main purpose of this paper has been that of investigating performance bounds for the two versions H-CA and HP-CA of the code assignment problem. In addition, in the course of this work, the study of the BB-CODE-ASSIGNMENT algorithm of [4] has been continued, new code assignments for special topologies in the HP-CA problem have been presented, and a new algorithm (SATURATION-DEGREE-CODE-ASSIGNMENT) has been designed.

For the BB-CODE-ASSIGNMENT algorithm, and also for the new algorithm proposed, the *gaps* between the *upper bounds* given by the heuristic solution, the *lower bounds* obtained from the maximum-clique problem, and the exact solution obtained by *branch & bound* are investigated in the different settings. In particular, important differences are found between the H-CA and HP-CA problems. For example, the algorithm BB-CODE-ASSIGNMENT (with nodes sorted by decreasing degrees) is close to optimal for HP-CA, but inferior with respect to SATURATION-DEGREE-CODE-ASSIGNMENT for H-CA. In addition, maximum clique sizes are tight lower bounds for HP-CA but not for H-CA.

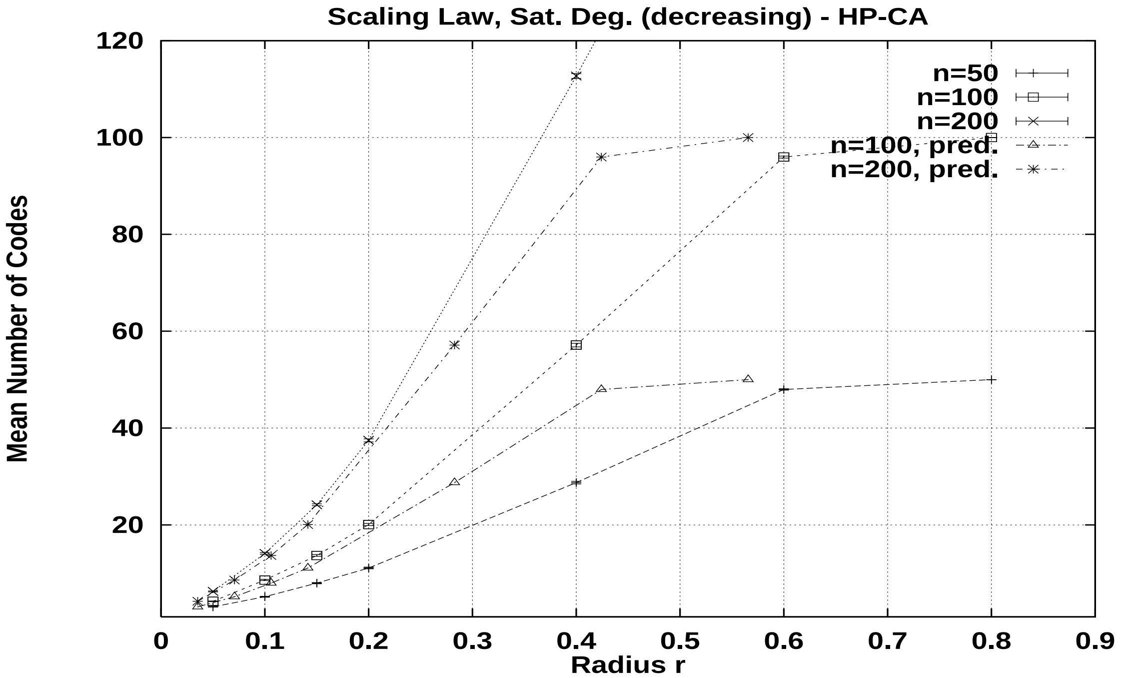


Figure 17: Scaling law: comparison between scaled curves and real curves (for $n = 100, 200$).

From the analysis of the probabilistic distribution of connections as a function of the Euclidean distance, and of the asymptotic size of the maximum cliques, a *scaling law* has been found from which the relations between the relevant properties of Euclidean networks with different station density and connection distance can be derived. In the assumption of negligible "border effects", the scaling law permits an immediate transfer of the results when the power of the transmitting stations (and therefore their connection radius r) increases, or when r is fixed but the number of stations n changes.

Acknowledgments

We would like to thank prof. Imrich Chlamtac, Bruce Hajek and Shivendra Panwar for helpful comments during the preparation of this paper. This work was supported in part by the Progetto Speciale 1995-96, Dip. di Matematica, Univ. di Trento.

References

[1] N. Abrahmson, The ALOHA system - Another alternative for computer communications, Proc. FJCC, pp. 281-285, 1970.

[2] R. Battiti and M. Protasi. Reactive local search for the maximum clique problem. Technical Report TR-95-052, ICSI, 1947 Center St.- Suite 600 - Berkeley, California, September 1995.

[3] M. Bellare, O. Goldreich, and M. Sudan, Free bits, PCPs and non-approximability. Toward tight results, Proc. 36-th Ann. Symp. on Foundations of Computer Science,IEEE Computer Society, pp. 422–431, 1995.

[4] A. A. Bertossi and M. A. Bonuccelli, Code Assignment for Hidden Terminal Interference Avoidance in Multihop Packet Radio Networks, IEEE Trans. on Networking, 3(4), pp. 441-449, 1995.

[5] D. Brélaz. New methods to color the vertices of a graph. Communications of the ACM, 22, pp. 251–256, 1979.

[6] I. Chlamtac and S. Kutten, On broadcasting in radio networks – Problem analysis and protocol design, IEEE Trans. Commun., Dec. 1985.

[7] I. Chlamtac and S. Kutten, A spatial reuse TDMA/FDMA for mobile multi-hop radio networks, in Proc. IEEE INFOCOM, Mar. 1985.

[8] I. Chlamtac and S. S. Pinter, Distributed Nodes Organization Algorithm for Channel Access in a Multihop Dynamic Radio Network, IEEE Trans. Computers 36, pp. 728-737, 1987.

[9] I. Cidon and M. Sidi, Distributed assignment algorithms for multihop packet radio networks, IEEE Trans. Computers 38, pp. 1353-1361, Oct. 1989.

[10] D. Johnson and M. Trick (Eds.), Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, in press.

[11] L. Hu, Distributed Code Assignments for CDMA Packet Radio Networks, IEEE Trans. on Networking 1(6), pp. 668–677, Dec. 1993.

[12] M. Kubale and B. Jackowski. A generalized implicit enumeration algorithm for graph coloring. Communications of the ACM, 28, pp. 412–418, 1985.

[13] S. M. Korman. The graph-coloring problem. In: N. Christophides, P. Toth, and C. Sandi, editors, Combinatorial Optimization, pp. 211-235, Wiley, New York, 1979.

[14] T. Makansi, Transmitted-oriented code assignment for multihop packet radio, IEEE Trans. Communications 35, pp. 1379-1382, Dec. 1987.

[15] A. Mehrotra and M. A. Trick, A column generation approach to graph coloring, Technical report series, Graduate School of Industrial Administration, Carnegie Mellon University, April 1995.

[16] R. Nelson and L. Kleinrock, Spatial TDMA: a collision-free multihop channel access protocol, IEEE Trans. Communications 33, pp. 934-944, Sept. 1985.

[17] M.J. Post, A.S. Kershenbaum, and P.E. Sarachik, Scheduling multihop CDMA networks in the presence of secondary conflicts, Algorithmica 4, pp. 365-394, Dec 1989.