Alcom-FT Technical Report Series
ALCOMFT-TR-03-169

# Algorithms and Experiments on Colouring Squares of Planar Graphs

M.I. Andreou, S. E. Nikoletseas and P.G. Spirakis
Computer Technology Institute (CTI) and Patras University, Greece
Riga Fereou 61, 26221 Patras, Greece. Fax: +30-2610-222086
emails: {mandreou,nikole, spirakis}@cti.gr *

**Abstract**

In this work we study the important problem of *colouring squares of planar graphs* (SQPG). We design and implement *two new algorithms* that colour in a different way SQPG. We call these algorithms $MDsatur$ and $RC$. We have also implemented and experimentally evaluated the performance of most of the known approximation colouring algorithms for SQPG [14, 6, 4, 10]. We compare the quality of the colourings achieved by these algorithms, with the colourings obtained by our algorithms and with the results obtained from two well-known *greedy colouring heuristics*. The heuristics are mainly used for comparison reasons and unexpectedly give very good results. Our algorithm $MDsatur$ outperforms the known algorithms as shown by the extensive experiments we have carried out.

The planar graph instances whose squares are used in our experiments are "non-extremal" graphs obtained by LEDA and hard colourable graph instances that we construct.

The most interesting conclusions of our experimental study are:

1) all colouring algorithms considered here have *almost optimal* performance on the squares of "*non-extremal*" planar graphs. 2) all known colouring algorithms especially designed for colouring SQPG, give *significantly better* results, even on hard to colour graphs, when the vertices of the input graph are *randomly named*. On the other hand, the performance of our algorithm, $MDsatur$, becomes worse in this case, however it still has the best performance compared to the others. $MDsatur$ colours the tested graphs with $1.1\,OPT$ colours in most of

*This work has been partially supported by the EU IST/FET projects ALCOM-FT and CRESCCO

the cases, even on hard instances, where $OPT$ denotes the number of colours in an optimal colouring. 3) we *construct worst case instances* for the algorithm of Fotakis el al.[6], which show that *its theoretical analysis is tight.*

# 1 Introduction

Communication in wireless systems, such as radio networks and ad-hoc mobile networks, is accomplished through exploitation of a limited range of frequency spectrum. One of the mechanisms utilised is to reuse frequencies where this does not result to unacceptable levels of signal interference. In graph theoretic terms, the interference between transmitters is usually modelled by the *interference graph* $G = (V, E)$, where $V$ corresponds to the set of transmitters and $E$ represents distance constrains (e.g. if two adjacent vertices in $G$ get the same or nearby frequencies, then this causes unacceptable levels of interference).

In most real systems, the network topology has some special properties, e.g. $G$ is a lattice network or $G$ is a planar graph. Planar graphs are the object of study in this work, both because of their importance in real networks and also because of their independent combinatorial interest.

The *Frequency Assignment Problem* (FAP) is usually modelled by variations of the *vertex graph colouring* problem. The set of colours represents the available frequencies. In addition, in an acceptable assignment the colour of each vertex of $G$ gets an integer value which has to satisfy certain inequalities compared to the colours of its nearby vertices in $G$ (frequency-distance constraints). The FAP problem has been considered e.g. in [7, 8, 11].

**Definition 1.1** ($k$-colouring) Given a graph $G = (V, E)$ let $F : V \rightarrow \{1, ..., \infty\}$ be a function, called $k$-colouring of $G$, such that for each $u, v \in V$ it is $|F_u - F_v| \geq x$ for $x = 0, 1, ..., k$ if $D(u, v) \leq k - x + 1$. $D(u, v)$ denotes the distance between $u$ and $v$ in $G$, i.e. the length of the shortest path joining them. The number of colours used by $F$ is called *order* and is denoted by $\lambda = |F(V)|$. The range of them is denoted by $s = max_{v \in V} F(v) - min_{u \in V} F(u) + 1$ and is called *span*.

**Definition 1.2** (Radiocolouring, RCP) When $k = 2$ then the function $F$ is called a *radiocolouring* of $G$ of order $\lambda$ and span $s$. The *radiochromatic number* of $G$ is the least order for which there exists a radiocolouring of $G$ and is denoted by $X_{order}(G)$. The least span is denoted as $X_{span}(G)$, respectively.

A variation of RCP useful in practice is the following:

**Definition 1.3** The *min order min span radiocolouring* on a graph $G$ is a function that radiocolours $G$ with the least number of distinct colours while the range of colours used is the least possible.

It can be easily seen, that the radiochromatic number of any graph $G$ ($X_{order}(G)$) is equal to the chromatic number of its *square*, i.e. $\chi(G^2)$ [8].

**Definition 1.4** The *square* of a graph $G$ is denoted by $G^2$ and is a graph on the same vertex set as $G$ with an edge between any pair of vertices of distance at most two in $G$. The graph $G$ is a *square root* of graph $G^2$.

In this work we experimentally study the colouring problem on the square of any planar graph (i.e. the min order RCP) and the min order min span RCP on planar graphs. Ramanathan and Loyd proved that the first one of the above problems is NP-complete [15]. Fotakis et al. proved the NP-completeness of RCP on planar graphs [6]. Thus, efficient solutions for these problems necessarily rely on the choice of heuristics and approximation algorithms. To our knowledge, for the colouring of SQPG, there is an 1.66-approximation algorithm due to Molloy and Salavatipour [14] that currently gets the best approximation ratio. We are aware of another three colouring approximation algorithms designed for these graphs [10, 6, 4]. The first two of them have approximation ratio 2 and the third one has ratio 1.8.

Towards a global picture of the problem our work studies a variety of algorithms that have *distinct methodologies*. In this direction we have first implemented a new algorithm that we propose for colouring SQPG. We call this algorithm $MDsatur$, because it is a *non-trivial* modification of the well-known colouring algorithm $Dsatur$ of Brélaz [5]. In our opinion, our approach is novel, because all known algorithms for this problem colour the uncoloured vertices of a given graph mainly based on the number of neighbours of each of these vertices either in graph $G$, or in the subgraph of graph $G^2$ that has already been coloured. On the other hand, our algorithm colours the uncoloured vertices mainly based on the number of distinct colours that have already been assigned to their neighbours.

We have also implemented the algorithm which is presented and theoretically analysed in the work of Fotakis et al. [6] and we call it $FNPS$. We also considered the algorithm of Agnarsson and Halldórsson presented in [4] and we call it $AH$. This algorithm actually concerns planar graphs of large maximum degree $\Delta \geq 749$ only. It is proved in [4] that the upper bound achieved by $AH$ is tight.

To make our experimental study on colouring SQPG more complete we have also implemented two well known *greedy* colouring heuristics, which colour any graph. The first one is the *Randomised First Fit* (RFF). The second, which we call it $MaxIScover$, covers the given graph by maximal independent sets.

Finally, we also propose and implement a new min order min span radiocolouring algorithm, which we call $RadioColouring$ ($RC$). $RC$ min order min span radiocolours any planar graph $G$ given a colouring on $G^2$. Based on its results we interestingly *conjecture* that $X_{span}(G) - X_{order}(G)$ is bounded from above by a small constant. So any efficient algorithm that solves the min order min span problem on planar graphs can be used as an efficient algorithm for the colouring problem on SQPG and vice-versa.

We use three sets of planar graph instances to evaluate the performance of the above algorithms. The first one, $S1$, has graphs which are obtained from the graphs generation library of LEDA [12]. These graphs are "*non-extremal*", because of the randomness in their generation. In the second set, $S2$, we create graphs that are modifications of the graph instances of the first set. We expected that the graphs of set S2 would be *harder* to colour graph instances for the colouring algorithms studied. The last set, $S3$, has hard colourable graph instances at least for our algorithm $MDsatur$ and for algorithm $AH$.

From the experimental results we observe that each one of the colouring algorithms considered in this work colours the square of any planar graphs from set $S1$ with $\Delta + c$ colours, where $c$ is a small constant (i.e. 4). It can be easily seen, that $\Delta + 1$ colours are necessary. The results of these algorithms are similar and on the graphs of set $S2$ as well. On the other hand, the results on the harder to colour graph instances of set $S3$ demonstrate the differences on the performance of each one of the algorithms considered. In particular, we remark that $MDsatur$ has the best *experimental performance* ($EP$) in all the cases. We measure the EP as the ratio of the number of colours the algorithm uses over the lower bound value on the chromatic number of the square of any graph $G$, that is $\Delta(G) + 1$. Our algorithm has $EP$ less than 1.1 in most of the cases, even on hard to colour instances, while its theoretically claimed approximation ratio is 1.5 [2]. Heuristic $MaxIScover$ has the second best EP. Heuristic $RFF$ also gives very good results. There are no theoretical results for these heuristics. Thus, we find their experimental results very interesting. The other algorithms give more or less expected results, much better than their theoretical upper bounds. For example, $FNPS$ has $EP$ ranging from 1.1 to 1.4. However, we present in Section 3 a planar graph $G$ and an order of its vertices such that if algorithm $FNPS$ selects the vertices according to this order, then it needs $2\Delta$ colours to colour $G^2$. This proves that the theoretical upper bound of this algorithm is tight.

All our implementations are written in the C programming language. The programmes run on a Sun Enterprise Solaris 2.6 system with 4 processors at 300MHz and 256 MB RAM. The source codes and the planar graph instances used are available at: http://students.ceid.upatras.gr/ mandreou/.

## 2 Description of the Algorithms

**The Algorithm MDsatur**

We here present our algorithm $MDsatur$. As we have said in the introduction, this algorithm is based on a novel approach for the colouring problem on SQPG and is inspired by algorithm *Dsatur* of Brélaz [5]. *Dsatur* colours very well and optimally graphs whose chromatic number is strongly related to their clique number, i.e. random graphs and $k$-colourable random graphs [17, 16]. The chromatic number of SQPG is also strongly related to their clique number, because $\chi(G^2) \leq 1.66\Delta(G) + 24$ ([14]). Thus, $\chi(G^2) \leq 1.66\omega(G^2) + 24$ (where $G$ is a planar graph and $\omega(G^2)$ is the size of the maximum clique in $G^2$).

Algorithm *Dsatur* greedily colours a given graph $G$. At each point it colours a vertex that has the maximum value on its *degree of saturation* with the smallest allowable colour. The *degree of saturation*, $Ds$, of a vertex is the number of distinct colours already assigned to its neighbours.

Algorithm $MDsatur$ colours the vertices of a given graph $G^2$ based on a different to that of *Dsatur* order, $O$, which is augmented by vertices that satisfy *more restrictions*. More precisely, let $i$ be the current point of $MDsatur$ in its application on $G^2$. Then, the current uncoloured vertex which manages to become the vertex in position $i$ of $O$, let it be $v$, satisfies the following requirements in turn: R1) has the maximum value on its degree of saturation in the current point of this algorithm (as *Dsatur*), R2) has the oldest coloured neighbour (i.e. sits in the leftmost position in $O$ among the oldest neighbours of all the uncoloured vertices) and R3) is the 'closer' neighbour of vertex $O_{i-1}$ (namely the last vertex coloured). We below provide a pseudo-code description of $MDsatur$.

**Algorithm:** *MDsatur*
*Input*: a graph $G = (V, E)$. *Output*: a colouring of $G$.
**Begin**

1. (a) Set each vertex of $G$ as uncoloured, and set a dynamic order on them, called $O$, as empty.

   (b) Select any vertex of $G$.
   Add it in the beginning of $O$ (in the position 1 of $O$), and colour it with colour 1.

2. **For** $i = 2$ **to** $n = |V|$ **do**

   (a) Set $X$ to be the set of the current uncoloured vertices which have the maximum value on their *degree of saturation* **(Requirement R1)**.

   (b) **For** each vertex $v \in X$ **do**
   Find the first neighbour of $v$ coloured (i.e. its oldest coloured neighbour) **(Requirement R2)**.

   (c) Select from $X$ the vertex $v$ which has the oldest coloured neighbour, denoted by $ON$, and also has the maximum number of common neighbours with the last coloured vertex (namely $O_{i-1}$) **(Requirement R3)**.
   If more than one of the vertices of $X$ satisfy these requirements, then select at random one of them with equal probability.
   If none of the uncoloured neighbours of $ON$, which are in $X$, has common neighbours with vertex $O_{i-1}$, then select one of them at random.
   Set $O_i = v$.

   (d) Colour $v$ with the smallest of its allowable colours, i.e. are not assigned to its neighbours.
   If none of the colours used is allowable for this vertex, insert a new colour.

3. Return the colour of each vertex.

**End**

The most '*closer*' neighbour of a vertex $u$ is this one whose colouring is

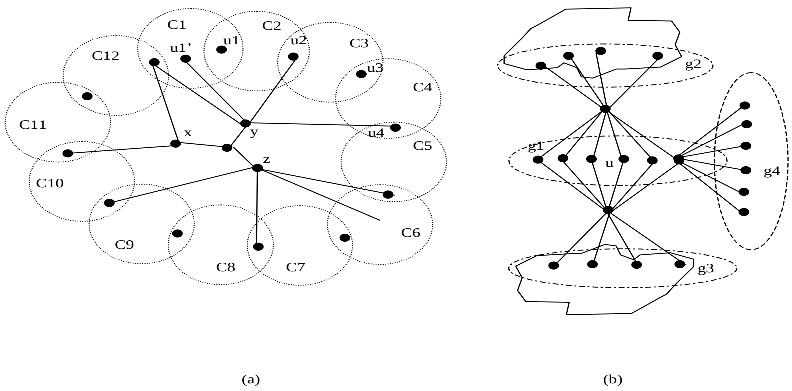


Figure 1: The graphs that are used in order to explain why $MDsatur$ colours a given graph more locally and compact than $Dsatur$.

determined at most by the colours of the other neighbours of $u$. In most cases this neighbour of $u$, lies near to it in a planar embedding of $G$. Note that we achieve to well-approximate the nearby vertices in a planar embedding of $G$, without actually finding any such embedding. The effect of this is that our algorithm can colour any graph, so its results are more interesting.

At this point, let us just provide an intuitive explanation of why this algorithm gives better results than the original algorithm $Dsatur$ on SQPG. The main reason is that $MDsatur$ colours the given graph *more locally* based on the last coloured vertex. Also, the subgraph of the given graph that has already been coloured at any point of the application of $MDsatur$ on this graph is *more compact* (dense) than the corresponding subgraph obtained from $Dsatur$. We explain how these properties of our algorithm are achieved using an example and we mention their consequences on the resulting colouring.

Let $G$ be a planar graph and suppose that $MDsatur$ is applied on $G^2$. Also suppose that in an intermediate point of this application the following hold: Let $\{C1, C2, C3, C4, ...\}$ be a sequence of partially coloured cliques and $\{u1, u2, u3, u4, ...\}$, be a sequence of uncoloured vertices. Each vertex $uj$ ($j = \{1, 2, 3, ...\}$), belongs in both cliques $Cj$ and $C(j+1)$ (see Fig. 1a). We assume that the values of the degree of saturation of all these vertices are the same. Then it would be possible to colour all these vertices sequentially before the colouring of other vertices which belong to the above cliques.

This fact, quite possibly yields an increase on the number of colours used, because it is possible to have vertices (like vertices $x, y, z$ in the figure), such that their degree of saturation, when they will be coloured, is mainly computed based on the colours of non-adjacent vertices.

To avoid the above situation, we focus our effort on the guidance of algorithm $Dsatur$, so that to colour as compact as possibly a given graph. In this direction we add requirement R2 in our algorithm, $MDsatur$. This has to be satisfied by the next vertex to be coloured, in each point of this algorithm. The behaviour of our algorithm on the above example is explained in the following. Let as call $ON_{u1}$ ($ON_{u2}$) the oldest coloured neighbour of vertex $u1$ ($u2$). If $u1$ is the next vertex to be coloured, then this means that vertex $ON_{u1}$ was coloured before vertex $ON_{u2}$ (due to req. R2). If $ON_{u1} \in C1$ (observe that $u1' \in C1$), then according to this algorithm the next vertex to be coloured cannot be vertex $u2$, because the oldest neighbour of $u1'$ is at least as old coloured as vertex $ON_{u1}$. Thus the next vertex to be coloured is vertex $u1'$, which extends only the same clique as the predecessor vertex coloured (namely $u1$). So, the first of our goals is achieved on the colouring obtained by $MDsatur$. However, if $ON_{u1} \notin C1$, then unfortunately it is possible to colour vertex $u2$ before vertex $u1'$.

Finally, requirement R3 is used in order to achieve to colour in turn nearby vertices (in an embedding of graph $G$) of the same clique. Namely we try to locally colour these vertices. As we have said before, the most 'closer' neighbour of the last vertex coloured lies near to its location in a planar embedding of graph $G$. Fig. 1b is representative of this case. Let $u$ be the next vertex to be coloured. Observe that the colouring of this vertex influences the value of the degree of saturation of each other vertex in the square of this graph. Suppose that all these vertices have the same value on their degree of saturation. By requirement R3 our algorithm colours after vertex $u$ a vertex from set $g1$ because only vertices of this set (i.e. neighbours of $u$ in the square of this graph) have the maximum number of common neighbours with the last vertex coloured ($u$).

**Lemma 2.1** (**Connectivity Property**) Suppose that algorithm $MDsatur$ is applied on a connected graph $G$. If $S$ is the subgraph of $G$ that has already been coloured at any point of $MDsatur$ during its application on $G$, then $S$ is a connected graph.

**Proof:** See full version [1].

■

**Remark.** Among the algorithms designed for colouring SQPG, [4, 6, 14, 2], the only one which has the Connectivity Property is algorithm $MDsatur$. This property of $MDsatur$ mainly differentiates it from the others and is one of the keypoints where the analysis of its performance is based on [2].

**The FNPS Algorithm**

$FNPS$ was first presented in [6]. It is a 2-approximation algorithm on SQPG.

**Lemma 2.2** [10] Let $G$ be a planar graph. Then there exists a vertex $v$ in $G$ with $k$ neighbours, let them $v_1,\ v_2,\ ...,\ v_k$, with $d(v_1) \leq\ d(v_2) \leq\ ...,\ d(v_k)$ such that one of the following is true:
(i) $k \leq 2$; (ii) $k = 3$ with $d(v_1) \leq 11$; (iii) $k = 4$ with $d(v_1) \leq 7$ and $d(v_2) \leq 11$; (iii) $k = 5$ with $d(v_1) \leq 6$, $d(v_2) \leq 7$ and $d(v_3) \leq 11$, where $d(v)$ is the degree of vertex $v$.

**Definition 2.1** Let $G$ be a graph and $e = uv$ be one of its edges. Set as $G' = G/e$ the graph that is produced from $G$ be deleting the vertex $u$ from it and joining each one its neighbours to the vertex $v$.

**FNPS Algorithm**
*Input*: a planar graph $G$ and its square $G^2$. *Output*: a colouring of $G^2$.
**Begin**

1. Sort the vertices of $G$ by their degree.

2. **If** $\Delta \leq 12$ **then** follows Procedure 1 below:
   **Procedure 1**: Every planar graph $G$ has at least one vertex of degree $\leq 5$. Now, inductively assume that any proper (in vertices) subgraph of $G^2$ can be coloured by 66 colours. Consider a vertex $v$ in $G$ with $degree(v) \leq 5$. Delete $v$ from $G$ to get $G'$. Now recursively colour $G'^2$ with 66 colours. The number of colours that $v$ has to avoid is at most $5\Delta + 5$. Thus, there is one free colour for $v$.

3. If $\Delta \geq 13$ then
   (a) Find a vertex $v$ and a neighbour $v_1$ of it, as described in Lemma 2.2, and set $e = vv_1$.
   (b) Form $G' = G/e$ and modify the sorted list of vertices according to their new degrees.
   (c) $F(G') = FNPS(G', G^{'2})$ that is the colouring which is produced by the recursive call of this function on $G'$ and $G^{'2}$.
   (d) Extend $F(G')$ to a valid colouring of $G^2$ as follows:
   Colour $v$ with one of the proper colours used in the colouring $F$ of $G^{'2}$. If no colour used is allowable for vertex $v$ insert a new one.

**End**

**The Algorithm $AH$**

We note that algorithm $AH$ colours the square of a planar graph $G$ if $\Delta(G) \geq 749$.

**AH Algorithm**
*Input*: a graph $G^2$ and the maximum degree of $G$, $\Delta$.
*Output*: a colouring of $G^2$.
**Begin**

1. $inductiveness = 1.8$;
2. **While** there are uncoloured vertices **do**
   (a) find any uncoloured vertex with degree less than $inductiveness * \Delta$ in the current subgraph of $G^2$ that has already been coloured i.e. vertex $v$
   (b) colour $v$ with the smallest allowable colour
   (c) delete $v$ from $G$
3. Return the colour of each vertex.

**End**

**The Heuristic RFF**

Let $G$ be a graph and $O$ a random order on its vertices. $RFF$ assigns to each vertex the first colour that is not assigned to any one of its previous neighbours (i.e. are before the vertex $v$ in order $O$).

**The Heuristic MaxIScover**

This heuristic finds a cover of the given graph $G$ by maximal independent sets (MISs). Namely, it uses an algorithm that finds a MIS in $G$. Then, it colours the vertices of this set with a new colour and it deletes them from the current graph $G$. This process is done recursively on until the graph $G$ becomes empty. Here we use heuristic $MaxIS$ [3] to find a maximal independent set of the current graph $G$. It is clear that instead of this algorithm any algorithm which finds MISs in SQPG can be used.

**The RadioColouring Algorithm**

We propose an algorithm that finds a min order min span radiocolouring. Since our experiments seem to suggest that in planar graphs the difference between $X_{span}(G)$ and $X_{order}(G)$ is a small constant (e.g. 4), the algorithm below can be also used to evaluate the chromatic number of the square of a planar graph.

**Algorithm RadioColouring (RC)**
*Input* : a graph $G$ and a colouring of its square, let it be $Col(G^2)$
*Output* : a min order min span radiocolouring of $G$, let it be $rad(G)$.
**Begin**

1. Initialise the radiocolouring of $G$ to be the colouring of its square (i.e. $rad(G) := Col(G^2)$).
2. Sort the vertices of $G$ in decreasing order by their degrees. Let $O$ be the resulting permutation.
3. Take into account the vertices of $G$ based on their rank in $O$. Radiocolour each one of them which has a wrong radiocolour, let it be $v$, as follows:

   $rad[v] :=$ smallest proper radiocolour, based on the radiocolours of the neighbours of $v$ in $G^2$, which have bigger degree than its degree (i.e. are to the left of vertex $v$ in $O$).
4. Return the radiocolour of each vertex.

**End**

# 3 Hard Instances

In this section we present the planar graphs whose squares are hard to colour by most of the algorithms considered. Let $G$ be any such graph. Agnarsson and Halldórsson in [4] characterise the graphs $G^2$ where their algorithm's performance bounds are tight. These graphs are worst case graph instances for $AH$. Andreou and Spirakis in [2] claim which the hard graph instance for our algorithm $MDsatur$ are. Here we identify a worst case instance for algorithm $FNPS$ and based on this we prove that its theoretical analysis is tight.

Based on Agnarsson et al, Andreou et al. [2, 4] we present here the potential structures of $G$. We also present an intuitive explanation of why the corresponding graphs $G^2$ are hard to colour graphs.

## 3.1 Hard Graph Instances for Algorithms $AH$ and $MDsatur$

Let $G'$ be a planar $r$-regular graph. The graph $G$ is obtained from $G'$ by replacing each of its edges, i.e. $e$, with a set of vertices and joining them

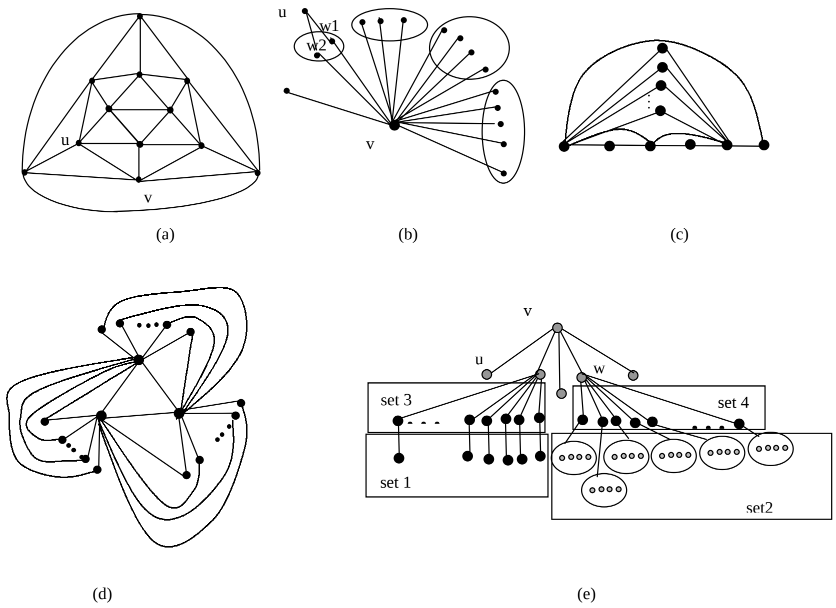


Figure 2: a) an example of a 5-regular planar graph, the icosahedron. b) the neighbours of vertex $v$ of icosahedron in the hard to colour graph obtained from the icosahedron when $k1 = 1, k2 = 2, k3 = 3, k4 = 4, k5 = 5$. c) a planar square root of a clique when there is a cycle of size six in its exterior in a planar embedding. d) a square root of a clique of size more than $\Delta + 3$. e) the graph that leads to the proof that the theoretical analysis on the performance of algorithm $FNPS$ is tight.

with the endpoints of the corresponding edge. We remark that $r \leq 5$ in the case of planar graphs [1]. An example of a 5-regular planar graph is shown in Fig. 2a.

Observe that each vertex in $G^2$ may belong in two cliques of maximum size. This happens when the degree of each vertex of $G'$ becomes $\Delta(G)$.

**About Algorithm $AH$:**

If $G'$ is a 5-regular planar graph, then the graph $G$ obtained from this graph as described above is one of the graphs, which makes the performance of algorithm $AH$ tight. This is true, because if all the edges of graph $G'$ are replaced by sets of vertices of the same size, i.e. $\Delta/5$, then each vertex has

$1.8\Delta$ neighbours in graph $G^2$. We recall that at each point of the application of this algorithm on $G^2$ it chooses a vertex of degree at most $1.8\Delta$ in the current subgraph of $G^2$ that has already been coloured and it colours it with the first allowable colour. Let $v$ be a vertex of $G$ and $H$ be the subgraph of $G$ on vertex $v$ and its neighbours (see Fig. 2a). Observe that $H^2$ is a clique in $G^2$ of size $\Delta + 1$. Thus, if all neighbours of each vertex of $H$ in $G^2 - H$ have already been coloured using a distinct colours from $\{1, 2, 3, ..., 0.8\Delta\}$, then $AH$ has to assign a distinct new colour to each of them. Thus, $AH$ uses $1.8\Delta$ colours to colour $G^2$.

**About Algorithm** ***MDsatur*****:**

The graph $G$ that is obtained from a planar $r$-regular graph $G'$ as described above also has a square that is a hard to colour graph by our algorithm $MDsatur$. Andreou and Spirakis claim this in [2]. The basic tools used in that work is the Connectivity Property of $MDsatur$ and the fact that no planar graph has as subgraph the graph $K_{3,3}$ [9]. Here we briefly present the basic steps of the proof of the assertion following [2]. Let $MDsatur$ apply on a graphs $G^2$.

1. Let $H^2$ be a currently uncoloured subgraph of $G^2$, then $MDsatur$ will colour its vertices with new colours iff the coloured neighbours of each of them have all the colours currently used. For example if the graph $H$ is the subgraph of $G$ on the vertex $v$ with its neighbour (like in Fig. 2b), then in order to colour graph $H^2$ with as many new colours as possible, it is implied that the vertices of graph $G^2 - H^2$ have to be already coloured. In [2] it is proved that at most half of the vertices of $H$ can have neighbours coloured with $\Delta$ distinct colours before their colouring.

2. In that work it is also proved that the structure of graph $H$ that leads to significantly hard to colour instances for $MDsatur$ is when $H$ is the star graph (observe that in this case $H^2$ is a clique). Otherwise, if $H^2$ is a clique and $H$ is not a star graph, then some of its vertices lie inside a cycle $C$ (in an embedding of $H$ in the plane) and by the planarity of $G$ it is impossible to have neighbours vertices of $G-H$ that lie outside cycle $C$ (see Fig. 2c). Thus, because of the connectivity property of $MDsatur$ it is proved in [2] that the graphs obtained under these conditions are not harder to colour, than the graphs obtained when $H$ is a star graph. It is also proved there, that the insertion of new vertices between any pair of vertices similar to $(w1, w2)$ in Fig.2b does not affect the performance of this algorithm, even if these new

vertices are $\Delta$. This is correct, because by the connectivity property it is impossible to colour the new vertices after the colouring of at least six vertices of $H$. This is true, because each group of new vertices (one for each pair of vertices $(w1, w2)$) are lie inside distinct bounded faces. Hence, their colouring affects only the degree of saturation of vertices of $H$.

3. If $H$ is the star graph, it is also proved in [2], that each set of at least four vertices of $H$ cannot belong to more than two cliques of size at least eight with at least three distinct vertices each in $G^2$ (because of $K_{3,3}$). Based on this it is proved in [2] that when $H$ is a star graph, $G^2$ is very hard to colour by $MDsatur$. By the argument used at the end of the above step we conclude that even if vertices of $H$ belong to more than two cliques of maximum size in $H^2$ this does not affect the performance of our algorithm.

4. Finally, if the graph $H^2$ is a clique of size greater than $\Delta + 3$, then $H$ has the structure of the graph shown in Fig. 2d as it is claimed in [2]. So again by the connectivity property it is proved in that work that the graph $G^2$ obtained under this condition is not hard to be coloured graph for $MDsatur$.

### 3.2 Worst Case Instance for Algorithm *FNPS*

In this subsection we prove that the graph $G$ properly obtained from the graph presented in Fig. 2e shows that the theoretical analysis of $FNPS$ is tight.

**Lemma 3.1** *The approximation ratio of algorithm $FNPS$ is tight.*

**Proof:** Let $G$ the graph obtained by the graph shown in Fig. 2e as follows: replace each supervertex of *set 2* by $\Delta - 1$ new vertices and joining each of the new vertices with the neighbour of the corresponding supervertex. To prove this Lemma it is enough to show that $FNPS$ colours vertex $x$ with colour $2\Delta$.

Suppose that algorithm $FNPS$ firstly selects the vertex $v$ (see in the pseudo-code step 3) to be a vertex from *set 1*. Observe that the degree of vertex $v$ in this case is two in the current graph $G$. Hence it is not needed to find a vertex that will satisfy the conditions of vertex $v1$ (see also the pseudo-code). Subsequently, $FNPS$ sequentially selects all the vertices of this set and all the vertices from *set 2* and then it deletes them from $G$. Observe that all these vertices had degree two in $G$, so their selection was allowable. Subsequently $FNPS$ can choose sequentially all the vertices of *set 3* and all the vertices of *set 4*. This is an allowable action, because the vertices of *set 2* have already been deleted from $G$. So the vertices of *set 3* had degree one at their selection time. Finally we assume that $FNPS$ selects in turn the vertices $u, w$ and $x$.

$FNPS$ is a recursive algorithm. Let $G'$ be the current graph $G$ in iteration $i$ and let $v$ be the vertex that is selected by $FNPS$ at this iteration. Then, from the way that $FNPS$ colours the vertices of $G$ we conclude that it colours vertex $v$ after the colouring of the graph $G - \{v\}$. Thus, $FNPS$ colours the vertices in the order that they are deleted from $G$. Hence, the vertices of *set 1* are first coloured by colour 1. The vertices of *set 2* which have replaced the same super vertex of the graph of Fig. 2e have the colours $\{1, 2, ..., \Delta-1\}$. To the vertices of *set 3* are assigned the colours $\{2, ..., \Delta-1\}$. Hence, the vertices of *set 4* have the colours $\{\Delta, \Delta+1, \Delta+2, ..., 2\Delta-2\}$. Finally vertex $u$ receives colour 1, vertex $w$ receives colour $2\Delta-1$ and vertex $x$ the colour $2\Delta$. ■

## 4 Graph Generation

The planar graph instances, which give the squares used as input to the colouring algorithms, that are experimentally evaluated in this work, are split in three sets. In the first one, we call it S1, are graphs created using the graphs generation library of LEDA [12]. More specific, we used the following LEDA procedures:

Class C1: random_planar_graph(G, n, m = 1.5*n); which produces a random maximal planar graph $G$ with $n-1$ vertices. Then it adds a new vertex into a random selected face $F$ and it joins this vertex with all the vertices of face $F$. An appropriate number of edges of graph $G$ are deleted randomly in order to have only $m$ edges, in the resulting graph. We mention here that $m \leq 3n-6$ in the case of planar graphs [9]. The graphs of Class C3 are maximal planar graphs and they have almost the maximum number of allowable edges. So, in this case we prefer an intermediate value on $m$ and we choose $m = 1.5n$

Class C2: triangulated_planar_graph(G, n); which produces planar graphs without any cycle of size at least four.

Class C3: maximal_planar_graph(G, n); which produces a maximal planar graph with $n$ vertices.

The planar graphs produced by the procedures of both classes C2 and C3 are initially generated the procedure of class C1, i.e. are random-planar graphs, and then these procedures appropriately add edges to these graphs to become triangulated planar and maximal planar, respectively.

In the second set, we call it S2, we take each of the graphs obtained from LEDA in classes C1, C2, C3 and we create a new graph modifying the original graph in order to get harder to colour graph instances for the considered algorithms. Their hardness has the characteristics of the hard graph instances which are discussed in Section 3. Hence, if $G$ is a planar graph from set S1 and $k$ is a positive integer, then the planar graph $G'$ obtained from $G$ by replacing each one of its edges, let it be $e$, with a set of new vertices of size $k$ and joining each one of the new vertices with the endpoints of edge $e$.

In the last set, S3, we create graphs obtained from the icosahedron ($20dron$) (see Fig. 2(a)) by replacing each of its edges with a set of new vertices. The size of these sets has one of the following values: $K = \{k1, k2, ..., k5\}$. We remark here that the icosahedron is a 5-regular planar graph with 30 edges. We also mention that the edges incident to each of its vertices have to be replaced by sets of vertices for each of the sizes $k1$ to $k5$. With this condition we achieve the degree of each vertex of the icosahedron to become $\Delta$ in the resulting graph (so each vertex of $G$ belongs in two cliques of maximum size in $G^2$. This is a hard case as it is explained in Section 3).

To specify the size of the set of vertices which replaces each edge of the $20dron$ we implement an exhaustive algorithm which edge-colours the $20dron$ with five colours and returns all possible edge colourings of this graph [1]. This algorithm returns 780 different edge colourings. Having an edge colouring we properly replace each edge of colour $i$ with a set of new vertices of size $ki$ ($1 \leq i \leq 5$).

The size $n$ of the planar graph instances of set S1 is 1024. In most past experimental works the size of the input graphs does not exceed 1024 [13]. We have done experiments on smaller (e.g. n = 256) and bigger graphs (e.g. n = 2048) and we observed that the asymptotic behaviours of the algorithms considered remain the same. So, in this sense, our experiments can be considered large enough and representative.

For graphs of set S2 we performed tests on the squares of the graphs obtained from the graphs of set S1 with size $n = 256$ and $k = 6$. In most cases the value of $n'$, that is the number of vertices of the resulting graph, is $3n * 6 \geq 4000$. As shown in Tables 1 and 2 the graphs of set S2, which are expected to be harder to colour instances than the corresponding graphs of set S1, do not affect the behaviour of the algorithms w.r.t. their behaviour on the squares of the graphs of set S1. Thus, we conclude that our experiments are representative.

We also performed some special experiments on the graphs of set S3. If $k1 = k2 = k3 = k4 = k5$, then we denote by $k$ this unique value. We produce graphs with $k = 20$ and $k = 100$. $20dron$ has only 30 edges, so it can be handled even with such a large value on $k$. From our results we conclude that the number of colours used by each algorithm considered is proportional to the increase of the value of $k$, while their performance remains the same. Thus, the results on these values of $k$ are representative.

In the case where the value of each $ki$ $(1 \leq i \leq 5)$ is distinct, we have done experiments for the case where $k1 = x, k2 = 2x, k3 = 3x, k4 = 4x, k5 = 5x$, for $x = 5, 15$. These results have greater interest on our algorithm, $MDsatur$ in contrast to the results of the previous case, which are more interesting for the rest of the algorithm and especially for $FNPS$.

# 5 Experimental Results

We split this section in two parts. In the first one, we present the experimental results of the colouring algorithms $RFF$, $MaxIScover$, $MDsatur$ and $FNPS$, which colour squares of planar graphs. In the second part, we present the result of the radiocolouring algorithm $RC$, when it has as input a planar graph $G$ and a colouring of $G^2$ obtained from each one of the above algorithms. In the following tables are depicted: a) the average number of colours used from each of the above algorithms on the squares of 10 planar graphs from each of the classes $C1, C2, C3$. b) the average maximum degrees of the corresponding planar graphs. We need this information in order to be able to make some comparisons between a good estimation of the optimal colouring (of each of these graphs, i.e. $\Delta + 1$) and the solutions produced from each one of the above algorithms. c) the value of $k$ or the value of $x$ depends on whether $ki$'s have the same or different values $(1 \leq i \leq 5)$.

## 5.1 Colouring the Square of a Planar Graph

The results of the application of the colouring algorithms, which are considered here, on SQPG are depicted in Tables 1-3. In Table 1 are shown the results on the squares of the graphs of set S1, in Table 2 the results on the squares of the graphs of set S2. In Table 3 we considered the squares of the graphs of set S3 and in Table 4 are shown the $ER$ of each of the algorithm considered on these graphs. The most important observation on

| planar graph class | max degree | Greedy | FindMIS | MDsatur | FNPS |
|---|---|---|---|---|---|
| random_planar_graph | 60,4 | 63 | 61 | 60,4 | 61 |
| triangulated_planar_graph | 24,1 | 26 | 28 | 24,5 | 25 |
| maximal_planar_graph | 120,1 | 121 | 120,5 | 120,5 | 121 |

**Table1: 1**

| planar graph class | max degree | Greedy | FindMIS | MDsatur | FNPS |
|---|---|---|---|---|---|
| random_planar_graph | 188,5 | 188,5 | 188,5 | 188,5 | 189 |
| triangulated_planar_graph | 99,4 | 102 | 101,2 | 99,8 | 100,5 |
| maximal_planar_graph | 295 | 295,5 | 295 | 295 | 296 |

**Table1: 2**

| | max | Greedy | FindMIS | MDsatur | FNPS |
|---|---|---|---|---|---|
| | degree | min % max% | min % max % | min % max% | min% max% |
| k=20 | 100 | 115 ~ 100% | 105 – 115 ~ 100% | 102,104 90% 120 10% | 135- 140 ~100% |
| k=100 | 500 | 540 – 560 ~ 100% | 510 – 550 ~ 100% | 502–505~90% 600 ~10% | 690–705 ~100% |
| x = 5 | 75 | 85-96 80% 100~20% | 80-85 ~100% | 77-82 50% 90-94 0.06% | 101 ~100% |
| x =15 | 225 | 247 – 260 ~ 100% | 240-260 ~100% | 227-247 50% 270-281 0.06% | 300 ~100% |

**Table1: 3**

| | max | Greedy | FindMIS | MDsatur | FNPS |
|---|---|---|---|---|---|
| | deg. | min % max% | min % max % | min % max% | min% max% |
| k=20 | 100 | 1.15 ~ 100% | 1.05–1.15 ~ 100% | 1.02-1.04 90% 1.20~10% | 1.35- 1.40 ~100% |
| k=100 | 500 | 1.1 ~ 100% | 1.02–1.10 ~ 100% | 1.004–1.01~90% 1.20~10% | 1.38–1.41 ~100% |
| x = 5 | 75 | 1.13-1.28 80%1.3 20% | 1.06,1.13%~100% | 1.01-1.09 50%1.2-1.25 0.06% | 1.35 ~ 100% |
| x =15 | 225 | 1.09 – 1.15 ~ 100% | 1.08-1.15 ~ 100% | 1.02-1.1 50% 1.2-1.25 0 .06% | 1.33 ~ 100% |

**Table1: 4**

Figure 3: Table 1: In this table are depicted the results, on average, of the colouring algorithms: $RFF$, $MaxIScover$, $MDsatur$ and $FNPS$. on 10 planar graphs from set S1 with 1024 vertices each. In Table 2 are shown the results, on average, of the above colouring algorithms on 10 planar graphs from set S2 with 256 vertices each and $k = 6$. In Table 3 are depicted the results, on average, of these algorithms on the icosahedron when $k = 20, 100$ and $x = 5, 15$. Finally in Table 4 are displayed the experimental ratios (ER) of each of the above algorithm on the results depicted in Table 3.

the results depicted in Table 1 is that each one of the considered algorithms uses $\Delta + \ constant$ colours to colour the square of any planar graph which belongs in set S1. Actually this constant term is very small, i.e. 4. This very good result yields the first indication that it is maybe possible to get colouring algorithms that colour the square of each planar graph with less than $1.66\Delta$ colours (that currently is the best known result). In order to further support this assumption we get results on harder to colour squares of planar graphs. I.e. the squares of the graphs of sets S2, S3.

In Table 2 are depicted the results on the squares of the graphs of set S2. From these results we conclude that the colouring algorithms we consider still colour very well the squares of the graph instances of set S2. There is no significant difference in the number of colours used from each one of the algorithms.

As we mention in the introduction the graphs obtained from LEDA are expected to be "non-extremal" graphs, because of the randomness in the way they are constructed (see Section 4). From the results we conclude that there does not exist a clique of size greater than $\Delta+3$ in these graphs, because all the algorithms considered colour them with $\Delta+$ *constant* colours. Because of this fact the results obtained on these graphs are reasonable (EP about 1), because as we can see from Tables 3, 4 almost all the algorithms considered have very good $EP$ even on hard colourable graph instances.

The last experiments have been done on the squares of graphs obtained from an $20dron$. I.e. the squares of the graphs of set S3. Our algorithm $MDsatur$ in the case where $k = 20$ uses $1.02\Delta$ to $1.04\Delta$ colours to colour the 90% from the 780 graphs tested. We recall here that we create one graph for each edge colouring of $20dron$. In only 10% of these squares it uses $1.20\Delta$ colours. When $k = 100$ the behaviour of our algorithm is improved. It uses at most $1.004\Delta$ to $1.01\Delta$ colours to colour 90% of the graphs tested. This is reasonable, because $MDsatur$ actually uses a constant number of colours with value bigger than $\Delta + 1$, so as greater is the value of $\Delta$ so smaller is the ratio among them (i.e. $\frac{\Delta+c}{\Delta+1}$).

Algorithm $FNPS$ uses in almost all cases $1.35\Delta$ to $1.40\Delta$ colours in order to colour the 780 graphs tested in the case where $k = 20$. In the case where $k = 100$ it uses slightly more colours, i.e. $1.38\Delta$ to $1.41\Delta$.

When $ki$'s have different values we examine the cases where $x = 5$ and $x = 15$. In these cases observe that the performance of our algorithm gets worse, but it is still better from the performance of the other algorithms. Algorithm $MDsatur$ gives similar results as in the case where all $ki$'s have the same value in only 50% of the tested graphs (i.e. it uses $1.01\Delta$ to $1.09\Delta$ colours). We observe that it uses at most $1.25\Delta$ colours to colour 0.06% of the graphs tested. The rest of these graphs are coloured with $1.1\Delta$ to $1.19\Delta$ colours by $MDsatur$. The colouring of each of these graphs is uniformly distributed in this range. On the other hand, the performance of $FNPS$

becomes better in this case.

The results presented in Tables 3 and 4 show the worst behaviour of algorithms $MDsatur$ and $FNPS$ on the squares of the graphs of set S3. For our algorithm these results are obtained when we named the vertices of $G$ randomly. On the other hand, when we label these vertices sequentially (namely nearby vertices in an embedding of $G$ receive consecutive integers as index), then it colours with $\Delta + c$ colours almost all (about 100%) of the graphs of set S3. The behaviour of algorithm $FNPS$ is exactly opposite. $FNPS$ colours almost all the squares of the graphs of set S3 with at most $1.2\Delta$ colours, when the vertices are randomly named, in contrast to $1.4\Delta$ in the other case.

Finally, on the squares of the graphs of set S3 both greedy heuristics considered ($RFF$,$MaxIScover$) colour the square of each such planar graph with $ER$ at most 1.15 in almost all the cases. The first of this heuristics, $RFF$, uses the same number of colours, that is $1.15\Delta(G)$, in almost each of the 780 graphs tested (for each value of $k$). The second heuristic $MaxIScover$ uses $1.02\Delta$ to $1.15\Delta$ colours uniformly spread in the above range on the squares of these graphs.

For the case where the values of $ki$'s are not the same, the performance of $RFF$ gets worse. Especially, when $x = 5$. In this case it uses $1.3\Delta$ colours in 20% of the graphs tested. Observe that its performance is better when $x = 15$. The heuristic $MaxIScover$ has almost the same results as in the above case.

At this point we discuss the experimental performance of algorithm $AH$. This algorithm colours the square of a planar graph with maximum degree at least 749. Because of this condition it is obvious why we do not present results from this algorithm. On the other hand, in order to study its behaviour and not the exact number of colours that it uses, we did experiments increasing the value of the variable *inductiveness* (see the pseudo code). From these results we can conclude that it has similar behaviour as algorithm $FNPS$. Namely, when it selects the next vertex to be coloured at random (beyond the vertices which satisfy the proper condition i.e. have less than $1.8\Delta$ neighbours in the already coloured subgraph of $G^2$), then it colours even the modified icosahedron with $k = \Delta/5$ with $1.2\Delta$ colours in almost all cases. Observe that exactly this graph gives the tightness on its performance as we explain in Section 3.

### 5.2 RadioColouring a Planar Graph

In this paragraph we notice that our algorithm *RadioColouring* min order min span radiocolours each of the planar graphs, $G$, used in the experiments shown in Tables 1-3 using at most four new extra colours in addition to the colours used by any of the algorithms: $RFF$, $MaxIScover$, $MDsatur$ and $FNPS$, to colour $G^2$.

The similarity in its behaviour on each of the set of graphs used and on each of the colourings of their squares allow as to make the following conjecture:

$X_{span}(G) - X_{order}(G) \leq c$, *where* $c$ *is a small constant.*

## References


[1] Maria Andreou, Sotiris Nikoletseas and Paul Spirakis: Algorithms and Experiments on Colouring Squares of Planar Graphs, TR2003/03 Computer Technology Institute, Greece, 2003. http://students.ceid.upatras.gr/m̃andreou/

[2] Maria Andreou and Paul Spirakis: Efficient Colouring of Squares of Planar Graphs, TR2002/11/01 Computer Technology Institute, Greece, 2002.

[3] Maria Andreou and Paul Spirakis: Planar Graphs, Hellenic Conference on Informatics, EPY, 2002.

[4] Geir Agnarsson, Magnus M.Hallorsson: Coloring Powers of Planar Graphs, ACM Symposium on Discrete Algorithms (SODA).

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

[6] D.A. Fotakis, S.E. Nikoletseas, V.G. Papadopoulou and P.G. Spirakis: $NP$-completeness Results and Efficient Approximations for Radiocoloring in Planar Graphs, In the Proceedings of the 25th International Symposium on Mathematical Foundations of Computer Science (MFCS), Editors Mogens Nielsen, Branislav Rovan, LNCS 1893, pp 363-372, 2000.

[7] D. Fotakis, G. Pantziou, G. Pentaris and P. Spirakis: Frequency Assignment in Mobile and Radio Networks. Networks in Distributed

Computing, DIMACS Series in Discrete Mathematics and Theoretical Computer Science, 45, American Mathematical Society (1999) 73-90.

[8] D. Fotakis and P. Spirakis: Assignment of Reusable and Non-Reusable Frequencies, International Conference on Combinatorial and Global Optimization (1998).

[9] Harary: Graph Theory, Addison-Wesley, 1972

[10] J. Van D. Heuvel and S. McGuiness: Colouring the Square of a Planar Graph, CDAM Research Report Series, July (1999).

[11] Katsela I. and M. Nagshineh: Channel assignment schemes for cellular mobile telecommunication system, IEEE Personal Communication Complexity, 1070, 1996.

[12] K. Mehlhorn and S. Naher: The LEDA Platform of Combinatorial and Geometric Computing, Cambridge University Press, 1999.

[13] D. Johnson, C. Aragon, L. Mcgeoch, C. Schevon: Optimization by simulated annealing: an experimental evaluation; Part II, Graph Coloring and Number Partitioning, Operating Research, Vol. 39, No. 3, 1991.

[14] M. Molloy and M.R. Salavatipour: Frequency Channel Assignment on Planar Networks , To appear in: Proceedings of 10th European Symposium on Algorithms, ESA 2002. The journal version is : A Bound on the Chromatic Number of the Square of a Planar Graph", submitted. ESA 2002

[15] S. Ramanathan, E. R. Loyd: The complexity of distance2-coloring, 4th International Conference of Computing and information, (1992) 71-74.

[16] J.S. Turner: Almost all k-colorable graphs are easy to color: Journal of Algorithms, 9, pp. 217-222, 1988.

[17] G. Tinhofer, E. Mayr, H. Noltemeier, M.M. Syslo (eds) in cooreration with R. Albrecht: Computational Graph Theory, Springer-Verlag/Wien, 1990 chapter "heuristics for graph colouring".