COMBINATORICA
Bolyai Society – Springer-Verlag

COMBINATORICA **20** (1) (2000) 87–102

ADAPTIVE MULTICOLOURING*

JEANNETTE JANSSEN, KYRIAKOS KILAKOS

Received November 14, 1996

The problem of adaptive multicolouring is finding a multicolouring for a graph and each of a sequence of changing weight vectors. Recolouring is either not allowed at all, or is allowed only in limited amounts. The aim is to minimize the number of colours used, subject to certain restrictions on the weight vectors. We establish the number of colours needed for adaptive multicolouring for the class of k-colourable graphs, both for the case when no recolouring is allowed, and for a case where limited recolouring is allowed.

1. Introduction

A multicolouring of a graph G and an integral weight vector c indexed by the vertices of G is an assignment of c_v colours to each vertex v such that the sets of colours assigned to adjacent vertices are disjoint. In adaptive multicolouring, the weight vector is constantly undergoing discrete changes, and at each change the multicolouring must be adapted to fit the new weights. Recolouring is either not allowed at all, or is allowed only within limited distance from the vertex where the weight has changed. In this paper our interest lies in the minimum number of colours needed for adaptive multicolouring when the successive weight vectors are all m-colourable for some fixed integer m (meaning that G and each of these successive weight vectors c can, with complete recolouring, be multicoloured using at most m colours).

Mathematics Subject Classification (1991): 05C15

* This work was partially financed by the FCAR-BNR-CRSNG project “Méthodes mathématiques pour la synthèse des systèmes informatiques.”

The interplay between changing weights and colouring strategy can be represented as a game between players A (adversary) and S (strategy). Given are a graph G, integers m and n, and an indication of the amount of recolouring that is allowed. In each move, player A changes one coordinate of the weight vector, restricted by the requirement that each weight vector must be m-colourable. (The parameter m should be seen here exclusively as a device to compare adaptive, interactive multicolouring with the best possible multicolouring that could otherwise have been achieved. Note that without m, S would always win and the game would be pointless.) If A drops the weight on a vertex, she may choose the colours that disappear from this vertex. Player S must adapt the previous colouring to the new weight, using only colours from $\{1,\dots,n\}$. She may do this by assigning new colours if the weight of a vertex has increased, and, if recolouring is allowed, by reassigning colours at the vertices that are within the recolouring distance of the vertex where the weight has changed (a definition of recolouring distance follows later).

Player S looses if she cannot find an adaptive multicolouring for the weights prescribed by A using only the given colours; she wins if she can find adaptive multicolourings for all possible moves of A. Note that the game is finite, because from the number of possible weight vectors and the number of possible colourings with colours $1,\dots,n$ for those weight vectors, a number t can be derived such that after t moves the same colouring must have been repeated, thus showing that player S can counter all moves of A. The representation of adaptive multicolouring as a game between players A and S is used repeatedly in this paper.

Our first result concerns adaptive multicolouring without recolouring. For a graph G and integer m, $\tilde{\chi}(G,m)$ is the minimum n for which player S has a winning strategy when she is allowed only to choose the colour when the weight on a vertex increases, but is not allowed to recolour, or choose which colour disappears when the weight decreases. For a class of graphs $\mathcal{G}$, $\tilde{\chi}(\mathcal{G},m)$ is the largest such n for any graph in the class. The result is given by the following theorem. Let $\mathcal{G}_k$ denote the class of all k-colourable graphs.

Theorem 1.1. *For all $k,m\geq 1$,*

$$\tilde{\chi}(\mathcal{G}_k,m)=k(m-1)+1.$$

Our second result concerns the situation when the adaptive multicolouring strategy (or player S in the game representation) is allowed limited recolouring. A recolouring distance of (p,q) means that recolouring is allowed within graph distance p from a vertex whose weight has increased and within distance q of a vertex whose weight has decreased. For a graph G and integer m, $\tilde{\chi}_{(p,q)}(G,m)$ is the minimum integer n for which player S has a winning strategy, when after a change by player A to the weight of vertex v, player S is allowed to reassign colours to all vertices at graph distance p or less from v if the change was an increase, and at graph distance q or less from v if the change was a decrease. For a class of graphs $\mathcal{G}$, $\tilde{\chi}_{(p,q)}(\mathcal{G},m)$ is the largest such n for any graph in the class. Obviously, when $p' \geq p$ and $q' \geq q$, $\tilde{\chi}_{(p',q')}(G,m) \leq \tilde{\chi}_{(p,q)}(G,m)$ for all G and m. Note that recolouring distance $(0,0)$ does allow recolouring at the vertices where the weight has changed, so in general, $\tilde{\chi}_{(0,0)}(G,m)$ is not the same as $\tilde{\chi}(G,m)$. The result is given by the following theorem.

Theorem 1.2. *For all $k \geq 2$ and $m \geq 1$,*

$$\tilde{\chi}_{(0,0)}(\mathcal{G}_k,m) = \tilde{\chi}_{(1,0)}(\mathcal{G}_k,m) = \begin{cases} \frac{1}{2}km & \text{if } m \text{ is even,} \\ \frac{1}{2}k(m-1)+1 & \text{if } m \text{ is odd.} \end{cases}$$

For both Theorem 1.1 and Theorem 1.2, there exist general multicolouring algorithms that attain the given bounds for all graphs in the class. For Theorem 1.1 this is an algorithm which can be considered ‘folklore’ in the telecommunications literature, and for Theorem 1.2 it is an algorithm developed in [12] and [5]. The crux of the proofs of the theorems lies in establishing that no possible algorithm can do better.

The multicolouring problem originates from cellular telephone network design. Users in a cellular network use a certain frequency to communicate with the base station in their cell. Frequencies in use in one cell can be reused in another cell, as long as the reuse constraints are not violated. These reuse constraints, which indicate in what pairs of cells the same frequency cannot be used simultaneously, can be modeled as the edges of a so called interference graph. For a given number of users in each cell, finding an allocation of the given frequencies that does not violate any reuse constraints is equivalent to finding a multicolouring of the interference graph. Whenever the number of users per cell changes, a new frequency assignment must be found. If no reassignment of frequencies is possible, this corresponds to adaptive multicolouring without recolouring (note that a decrease in weight corresponds to the termination of a phone call, so the frequency assigner has no control over which ‘colour’ disappears.) However, limited reassignment of frequencies (in other words, recolouring) has become a technical possibility. For more details, we refer to [5, 12]. Recent work on the adaptive multicolouring problem for restricted classes of graphs, especially socalled *hexagon graphs*, can be found in [11], [6] and [3].

The multicolouring game between players A and S fits in the context of graph colouring games (see [7] for an overview of such games). One such game is the game which determines the game-chromatic number (see for example [9]). There, a graph is given, and two players choose in turn a vertex and colour it. Player I tries to colour the whole graph with the given colours, and player II moves to prevent this. Another related game is the colouring extension game between a "mapmaker" and an "explorer"(see e.g. [1]). The explorer presents different k-colourable graph extensions to a set of independent vertices. The mapmaker has to colour the extensions, and looses if in the course of the game he is forced to use all possible colourings of the set of independent vertices. Even though the multicolouring game is of a slightly different nature, the flavour of the results and the proofs presented here puts it clearly in the same category.

An adaptive multicolouring strategy has to adapt to unpredictably changing weight vectors, so such a strategy is an on-line algorithm, and the problem is an on-line colouring problem. The corresponding off-line problem is the problem of finding an optimal multicolouring when a graph and a weight vector are given. It makes sense to compare on-line and off-line versions of the problem by the performance ratio. For multicolouring, this is the ratio between $\tilde{\chi}(G,m)$, the number of colours needed for adaptive, on-line multicolouring, and m, by definition the number of colours needed to colour the graph and any of the sequence of weight vectors off- line. Still, what is usually known as on-line graph colouring is of a slightly different nature. There, an unknown graph is given vertex by vertex. Each given vertex must be coloured before the next vertex is presented. For an overview of results on on-line colouring, see [7], [8], or [4].

2. Preliminaries

A *(vertex) colouring* of a graph is an assignment of colours to its vertices so that no two adjacent vertices receive the same colour. The minimum number of colours that is needed to colour a graph G is called the *chromatic number* of G and denoted by $\chi(G)$. A graph that can be coloured with k colours is called *k-colourable.* A *k-coloured graph* is a k-colourable graph with a colouring of its vertices with k colours.

A *weighted graph* G_c is a pair (G,c) where G is a graph with vertex set $V=\{v_1,\dots,v_{|V|}\}$ and c is a positive integer weight vector $(c_1,\dots,c_{|V|})$. A *multicolouring* of a weighted graph G_c is an assignment of c_i colours to each vertex $v_i \in V(G)$, such that no two adjacent vertices have any colours in common. Hence a colouring of a graph G is a multicolouring of $G_{\mathbf{1}}$, where $\mathbf{1}$

is the all-one vector. The chromatic number of a weighted graph G_c, $\chi(G_c)$, is the minimal number of colours needed for a multicolouring of G_c.

A *c-sequence* for a graph G is a sequence of weight vectors c^i for G, such that c^j and c^{j+1} differ at most in one coordinate. The chromatic number of a c-sequence for a graph G is the maximum of $\chi(G_{c^i})$, taken over all vectors c^i in the sequence. A c-sequence of chromatic number m is called m-colourable.

3. Without recolouring

In this section we consider adaptive multicolouring when no recolouring is allowed. So in the game between players A and S (given a graph G and integers m and n), player A changes the weight at a vertex, subject to the condition that the new weight vector remains m-colourable (in other words, if the new weight vector is denoted c, then it must be that $\chi(G_c) \leq m$). Player A also chooses the colours to be removed when the change is a decrease. When the change is an increase, player S chooses colours from $\{1, \ldots n\}$ and assigns them to the vertex such that the result is a multicolouring for the new weights. When the change is a decrease, player S skips a turn. We will prove the following theorem by proving first a lower bound on $\tilde{\chi}$ for k-colourable graphs, and then exhibiting a strategy which achieves this bound.

Theorem 1.1. *For all $k, m \geq 1$,*

$$\tilde{\chi}(\mathcal{G}_k, m) = k(m-1) + 1.$$

The first lemma establishes a lower bound on $\tilde{\chi}(\mathcal{G}_k, m)$ for all $m, k \geq 1$.

Lemma 3.1. *For all integers $m, k \geq 1$, there exists a k-coloured graph G with the property that for any adaptive multicolouring algorithm that does not recolour there exists an m-colourable c-sequence that will force this algorithm to colour $k(m-1)+1$ independent vertices of weight 1 with $k(m-1)+1$ different colours.*

Proof. Fix m. For simplicity, we adopt the notation $N_k = k(m-1)+1$ We will construct a graph G with the required property by induction on k. We will then prove that, when the multicolouring game is played on G with m and $n \geq N_k$, then player A can force player S to use $k(m-1)+1$ colours on $k(m-1)+1$ independent vertices of G, all of which have weight 1 (and thus contain only one colour). The strategy that player A follows to do this represents the required c-sequence.

For $k = 1$, let G be the graph that consists of m independent vertices. Player A starts by putting weight m (successively) on all vertices. Obviously, player S must use at least m colours to satisfy the given weights, and $N_1 = m$. Next, player A reduces the weight of all vertices (successively) to 1, and removes colours so that $m = N_1$ distinct colours, one on each vertex, are left.

For the induction step, fix k, and let G' be the graph that has the desired properties for $k-1$, obtained from the induction hypothesis. We construct G as follows. First we form G'' by taking m disjoint copies of G'. Then, for each set of N_{k-1} independent vertices that belong to the same copy of G' in G'', we attach $m-1$ new independent vertices to each vertex of this set. We will call these vertices the *new neighbours* of the set. Clearly, G is k-colourable, because G' (and thus also G'') is $(k-1)$-colourable and the new vertices form an independent set.

Player A starts by forcing player S to use N_{k-1} different colours on N_{k-1} independent vertices of weight 1 of each copy of G'. The induction hypothesis insures that she can do this, regardless of the strategy that player S follows (note that the colours that player S uses could be the same on each copy of G', or they could be completely different). After forcing these colours, player A drops the weight of all other vertices to zero; in other words, the only vertices that now have non-zero weight are N_{k-1} independent vertices in each copy of G'. The colours used on any copy of G' are distinct, by induction, but across the copies colours may be repeated.

The next move of player A depends on the colours that player S has used on these vertices of weight 1. Suppose first that two disjoint sets of N_{k-1} of these vertices exist such that the same N_{k-1} distinct colours occur on both sets. Call these sets V_1 and V_2. Player A then increases the weight of the $m-1$ new neighbours of V_1 from 0 to $m-1$. To colour these new neighbours, player S will need at least $m-1$ colours that are different from the N_{k-1} colours on V_1. By dropping the weight to 1 on each of the new neighbours of V_1 and choosing the colours that disappear appropriately, player A can guarantee $m-1$ new colours occur on $m-1$ new neighbours. The new neighbours of V_1 form an independent set and are not connected to V_2. Together with the vertices of V_2, they form a set of $N_{k-1} + (m-1) = N_k$ independent vertices of weight 1, coloured with N_k distinct colours.

If sets V_1 and V_2 with the required property cannot be found, then player A has already reached the objective of the lemma. In this case, no N_{k-1} colours occur twice on the vertices of non-zero weight. Now, let N be the number of distinct colours that are present on the vertices of non-zero weight. At most $N_{k-1}-1$ colours occur more than once, and they occur each at most m times (once in each copy of G'). There are mN_{k-1} vertices of non-zero weight, so $N \geq mN_{k-1} - (m-1)(N_{k-1}-1) = N_{k-1} + m - 1 = N_k$.

Fixed Allocation, an adaptive multicolouring strategy well- known in the field of radiocommunications (see for example [10] or [2]) achieves the bound of the preceeding lemma. It is based on a given colouring of G, which will be called the *base colouring*. Note that Fixed Allocation does not use any recolouring.

ALGORITHM. Fixed Allocation (FA).

DESCRIPTION. *For a given k-coloured graph G and an integer $n \geq k$, the n available colours are divided into k subsets of order $\lfloor \frac{n}{k} \rfloor$, and a rest set of order $n - k\lfloor \frac{n}{k} \rfloor$. Each colour class of the base colouring is associated with one of the sets of order $\lfloor \frac{n}{k} \rfloor$. For any vector c, the vertices of colour class i are coloured with the colours of the associated set of colours. When one of the weights c_i of a vector c exceeds $\lfloor \frac{n}{k} \rfloor$, the colours from the rest set are assigned greedily.*

Since the sets assigned by FA to adjacent vertices are disjoint, a conflict will never occur, no matter what history the c- sequence follows. Since the colours of the rest set are assigned dynamically, hardly any guarantee can be given about their use. These two considerations explain the following lemma.

Lemma 3.2. *For any $m, k \geq 1$, FA using $n = k(m-1)+1$ colours can find adaptive multicolourings for any k-coloured graph and every m-colourable c-sequence for such a graph.*

Proof. Fix k, and let G be any k-coloured graph. Consider the FA algorithm with $n = mk - (k-1)$ colours. As defined, each vertex will be assigned $m-1$ colours according to its base colour, and the one remaining colour will be assigned greedily. It is clear that for each m-colourable weight vector c, $c_i \leq m$ for all i. Each vertex v_i with $c_i \leq m-1$ can be coloured with the $m-1$ colours associated with its base colour. Moreover, if $c_i = m$, then $c_j = 0$ for each vertex v_j adjacent to v_i, so the colour that is assigned greedily can be used in addition to the $m-1$ colours assigned to v_i to cover this weight. Thus FA with n colours can adaptively multicolour any m-colourable weight vector. ∎

The two lemmas given in this section are sufficient to prove Theorem 1.1.

4. With recolouring

In this section we consider adaptive multicolouring when limited recolouring is allowed. We recall the definition of recolouring distance. If a recolouring distance of (p,q) is agreed, for some non-negative integers p and q, then the game between players A and S is the following. Given are a graph G and integers m and n. At each turn, player A changes the weight on a vertex v, such that the resulting weight vector remains m-colourable. Player S finds a multicolouring to fit the new weights. She is hereby allowed to assign new colours to all vertices that are at graph distance p or less from v, if the change was an increase, or at graph distance q or less from v, if the change was a decrease.

We prove the following theorem, by first giving a lower bound on $\tilde{\chi}_{(1,0)}$ for the class $\mathcal{G}_k$ for each $k\geq 2$, and then exhibiting an algorithm that attains this bound.

Theorem 1.2. *For all $k\geq 2$ and $m\geq 1$,*

$$\tilde{\chi}_{(0,0)}(\mathcal{G}_k,m)=\tilde{\chi}_{(1,0)}(\mathcal{G}_k,m)=\begin{cases}\frac{1}{2}km & \text{if } m \text{ is even,}\\ \frac{1}{2}k(m-1)+1 & \text{if } m \text{ is odd.}\end{cases}$$

First, a lemma will be stated and proven which gives as a corollary a lower bound on $\tilde{\chi}_{(1,0)}(\mathcal{G}_k,m)$ for all $k\geq 2$ and $m\geq 1$. The statement of the lemma is somewhat stronger than needed for the lower bound, to facilitate the inductive proof. Some definitions are needed.

We will consider graphs whose vertex set can be partitioned into equal-sized sets of independent vertices. We call these sets *supervertices*. Two supervertices are *colour disjoint* if none of the colours on any of the vertices in one of the supervertices occurs at any of the vertices in the other super vertex. Supervertices are called *independent* when there are no edges that connect any of their vertices. A supervertex is said to have *weight* x if each of its vertices has weight x.

The *uniform* chromatic number of a c-sequence for a graph whose vertex set is partitioned into supervertices is the smallest number t such that for every vector c^i of the c-sequence, G_{c^i} admits a multicolouring of t colours which is consistent on the supervertices, i.e. which assigns all vertices within a supervertex the same colour sets. Obviously, for any c-sequence the uniform chromatic number is at least as large as the chromatic number.

As in the proof of Lemma 3.1, we will construct graphs by taking several copies of the graph given by a previous induction step, and connecting these copies with some independent vertices. To avoid clutter, we define beforehand two parameters, one that is related to the number of copies that has to be taken at each induction step, and one that indicates the number of supervertices of the graphs constructed in the proof.

The *copy number* $C_{T,k,x}$ is defined as the number of ways to form k mutually disjoint sets $V_1,\dots,V_k$ such that $|V_i|\geq x$ and $V_i\subseteq\{1,\dots,T\}$ for all

$i \le k$. In the proof of the next lemma, the value of x will be fixed, so for clarity, we will drop the parameter x from this notation.

The *supervertex numbers* $N_{T,k}$ are defined recursively as follows, for every T and $k \ge 2$:

$$\begin{aligned} N_{T,2} &= 2(T+1), \\ N_{T,k+1} &= (N_{T,k})^{kC_{T,k}}(N_{T,k}kC_{T,k}+1)(T(k+1)+1). \end{aligned}$$

Lemma 4.3. *Fix an integer $x \ge 1$. For all integers $k \ge 2$, $M \ge 1$, $T \ge kx$, there exists a k-colourable graph $G(k,M,T)$ whose vertex set can be partitioned into $N_{T,k}$ supervertices of size M, which has the following property. For any adaptive multicolouring algorithm that uses recolouring distance at most* $(1,0)$, *there is a c-sequence for $G(k,M,T)$ which forces this algorithm to either use more than T colours, or to colour such that the resulting multicolouring creates k pairwise colour disjoint, independent supervertices of weight x.*

We will prove this lemma by induction on k. Since the base case, $k=2$, exhibits a structure that will also prove useful for the induction step, we handle this case as a separate lemma.

Definition. An (M,T,p)-*separator graph* is a bipartite graph on a vertex set with bipartition (U,V), where

$$\begin{aligned} U &= \{u_{ij} \mid 1 \le i \le T+p, 1 \le j \le M\}, \\ V &= \{v_{ij} \mid 1 \le i \le T+p, 1 \le j \le M\}, \end{aligned}$$

and u_{ij} is adjacent to $v_{i'j'}$ precisely when $i \ne i'$. The sets $U_i = \{u_{ij} | 1 \le j \le M\}$ and $V_i = \{v_{ij} | 1 \le j \le M\}$ are the supervertices of this separator graph.

Lemma 4.4. *Fix integers $x \ge 1$, and M,p,T. Let G be an (M,T,p)-separator graph. Let c be the vector that places weight x on all vertices of G. Then any multicolouring of G_c has the property that either more than T colours are used, or there are at least p pairs of independent supervertices (U_i,V_i) that are colour disjoint.*

Proof. Let G and c be as in the statement of the lemma. Assume a multicolouring of G_c. If any colour occurs on both members of a pair of supervertices (U_i,V_i), then this colour cannot be assigned to any other $U_{i'}$ or $V_{i'}$, $i' \ne i$, since all vertices of any of the $V_{i'}$ or $U_{i'}$ with $i' \ne i$ are adjacent to all vertices of U_i or V_i, respectively.

If less than p supervertex pairs (U_i,V_i) are colour disjoint, then at least $(T+p)-(p-1)=T+1$ supervertex pairs have some colours in common. No two supervertex pairs can have the same common colour, so at least $T+1$ distinct colours have been used.

Proof of Lemma 4.3. Fix x. We construct $G(k,M,T)$ for each M and each $T \geq kx$, by induction on k. We will then show the required c-sequence as a strategy of player A in the multicolouring game played on $G(k,M,T)$ with $m=2x$ and $n=T$, with allowed recolouring distance $(1,0)$, and with the additional requirement that for each move of player A, the resulting weight vector must be *uniformly* m-colourable. Also, we will add to the induction the requirement that no component of any vector of the c-sequence every exceeds x. We will show that player A can force player S to either use more than T colours, or to colour such that she creates k pairwise colour disjoint, independent supervertices of weight x.

The base case follows from Lemma 4.4: Fix M and $T \geq 2x$; $G(2,M,T)$ is the $(M,T,1)$-separator graph, which has $2(T+1)=N_{T,2}$ supervertices of size M. The strategy of player A is to raise the weight on each vertex to x. The fact that the player A so forces player S to either use more than T colours, or to force 2 independent colour disjoint supervertices, follows directly from the lemma.

Since $G(2,M,T)$ is bipartite, and each supervertex belongs entirely to one side of the bipartition, $G(2,M,T)_c$, where c is the vector that assigns weight x to each vertex, admits a colouring of $2x$ colours which is consistent on the supervertices: just put x colours on each side of the bipartition. So the c-sequence that represents the moves of player A has uniform chromatic number $2x$.

For the induction, suppose the statement of our lemma is true for k, for all $T \geq kx$ and for all $M \geq 1$. Fix M and $T \geq (k{+}1)x$. We form $G(T,M,k{+}1)$ as follows. Let $p=kT{+}1$, and let $M'=(N_{T,k})^{kC_{T,k}}M(T{+}p)$. Take $kC_{T,k}$ copies of $G(T,M',k)$, obtained by the induction hypothesis. Its supervertices, of size M', will be called the old supervertices. Partition the (old) supervertices of each copy of $G(T,M',k)$ into $(N_{T,k})^{kC_{T,k}}$ supersets of size $M(T+p)$, and label these supersets $1,2,\ldots,N_{T,k})^{kC_{T,k}}$. These supersets will contain our new supervertices.

Each of the $kC_{T,k}$ copies of $G(T,m',k)$ has $N_{T,k}$ (old) supervertices, so there are $N_{T,k})^{kC_{T,k}}$ ways to choose one supervertex from each copy. Label all these different combinations also from 1 to $N_{T,k})^{kC_{T,k}}$.

Now for each i $(1 \leq i \leq N_{T,k})^{kC_{T,k}})$, a new superset is attached to the graph in the following way. Take the collection of $kC_{T,k}$ old supervertices (of size M'), one from each different copy of $G(T,M',k)$, that has label i. Choose the superset with label i in each of the old supervertices of the collection. Create a new superset of $M(T+p)$ independent vertices, and join this superset to each of the supersets of the collection by way of an (M,T,p)-separator graph, such that the supervertices of the separator graph correspond to the new supervertices in the collection. The new supersets, which are not part of any of the copies of $G(k,M',T)$, are called the *added* supersets, and their (super) vertices the *added* (super) vertices. Figure 1. shows schematically the formation of $G(k+1,M,T)$ from copies of $G(k,M',T)$.

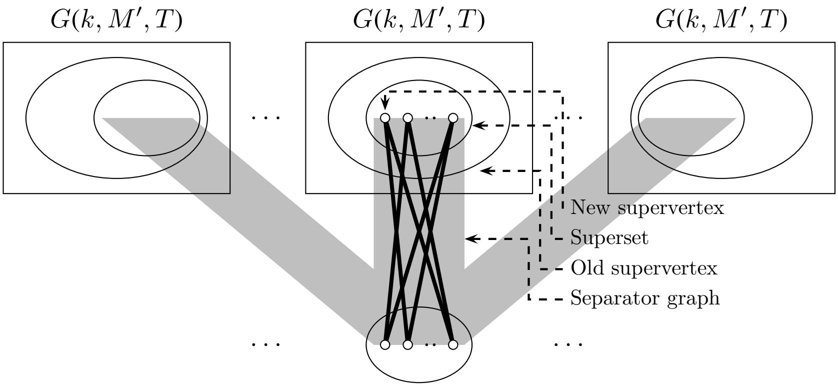

Fig. 1. Formation of $G(k+1,M,T)$

We now show that the new graph indeed has the required properties. First we count the number of supervertices. Each of the old supervertices in each of the $kC_{T,k}$ copies of $G(k,M',T)$ is divided into $(N_{T,k})^{kC_{T,k}}$ supersets, each containing $T+p = T(k+1)+1$ new supervertices. This makes for $(N_{T,k})^{kC_{T,k}} N_{T,k} kC_{T,k}(T(k+1)+1)$ new supervertices. On top of this we have the $(N_{T,k})^{kC_{T,k}}$ added supersets, each also containing $T+p=T(k+1)+1$ new supervertices. In total, then, the new graph $G(k+1,M,T)$ contains $(N_{T,k})^{kC_{T,k}} N_{T,k} kC_{T,k}(T(k+1)+1)+(N_{T,k})^{kC_{T,k}}(T(k+1)+1)=N_{T,k+1}$ new supervertices of size M, as required.

Since all added vertices form an independent set, and $G(k,M',T)$ is k-colourable, $G(k+1,M,T)$ is $k+1$-colourable.

We now show the strategy of player A which forces S to use more than T colours, or to create $k+1$ pairwise colour disjoint, independent new supervertices of weight x. First, player A increases to x the weight of all added vertices. Player S colours them. Then player A follows the strategy, given by the induction, that forces more than T colours or creates k independent colour disjoint old supervertices, in the first copy of $G(k,M',T)$, which we will call G_1. Player S is allowed to recolour within distance 1 of the vertices of G_1 where player A increases the weight during this step. This implies that player S can recolour some of the vertices in the added supersets. However, this does not interfere with the game on G_1.

If player S responds by colouring with more than T colours, then player A has established her objectives and the game is over. Suppose instead that player S creates k colour disjoint (old) supervertices in G_1. Player A reacts by dropping to 0 the weight of all vertices of G_1 that are not part of these supervertices. Player S is not allowed to recolour, since the weight has decreased.

Next player A follows the induction strategy (the one that forces player S to use more than T colours or create k independent colour disjoint old supervertices) on G_2, the second copy of $G(k,M',T)$. Again, player S is allowed to recolour some of the vertices in the added supersets. But the vertices that were retained from G_1 have distance at least 2 from any vertex in G_2, so the colours on these vertices will remain the same. We assume again that player S does not use more than T colours, and thus creates k independent colour disjoint old supervertices in G_2. Again player A drops to 0 the weight of all vertices of G_2 that are not part of those supervertices. This process is repeated for each further copy of $G(k,M',T)$.

Note that it follows from the strategy and the induction hypothesis that during this game, no vertex ever receives weight more than x.

To complete our proof, we have to check the following two conditions. Firstly, we have to check that player A achieved her objectives, in other words, we must show that if player S coloured none of the copies of $G(k,M',T)$ with more than T colours, then the multicoloured graph $G(k+1,M,T)$ that results should have $k+1$ independent pairwise colour disjoint supervertices of size M. Secondly, we have to show that player A didn't violate the rules of the game. In other words, the c-sequence that resulted from the moves of player A should have uniform chromatic number at most $2x$.

We first show the second condition.

By induction, weight vectors that result from the moves of player A on G_1 have uniform chromatic number at most $2x$ on G_1. In other words, for each vector c that results from the moves of player A on G_1, there is a multicolouring of $(G_1)_c$ which uses at most $2x$ colours and which colours all vertices in an old supervertex with the same colours. Each of those multicolourings can be easily extended to a multicolouring that includes the added vertices with assigned weight x, which still uses only $2x$ colours and is consistent on the new supervertices. This can be done since any of the added supersets, and hence any of the added new supervertices, is connected to only one of the old supervertices of G_1, and since no vertex ever receives weight more than x. So the weight vectors that result from the moves of player A of first raising the weight of the added vertices to x, and then using induction on G_1, also have uniform chromatic number $2x$.

When player A plays on G_2, again the corresponding c- sequence when restricted to G_2 has uniform chromatic number at most $2x$. Now the graph that consists of G_2 and the vertices that have non-zero weight when player A is playing G_2 is in fact G_2 itself with bipartite graphs attached 'sideways' to each old supervertex. This is because the supervertices of G_1 whose weight is not dropped to zero are independent. Hence again, for any vector c that results from the moves of player A on G_2, any multicolouring of $(G_2)_c$ with $2x$ colours which is consistent on the old supervertices of G_2 can be extended to a multicolouring which meets the same conditions of $G(k+1,M,T)$ with the same weights on G_2, and weight x on the added vertices and on some independent supervertices of G_1. So the weight vectors that reflect the moves of player A when she plays on G_2 have uniform chromatic number $2x$. The same is true, for exactly the same reasons, for the weight vectors that represents the moves of player A on all the subsequent copies of $G(k,M',T)$.

Finally we show that player A indeed forces player S to use more than T colours, or to create $k+1$ independent pairwise colour disjoint new supervertices. We assume that, in the game between player A and player S as described above, player S coloured all copies of $G(k,M',T)$ with not more than T colours (if this is not the case then we are done). We show that the resulting colouring must contain $k+1$ independent new supervertices that are pairwise colour disjoint.

At the end of the game, if player S coloured each copy with at most T colours, then the number of copies of $G(k,M',T)$ that we used, $kC_{T,k}$, and the definition of the copy number $C_{T,k}$, ensure that there are at least k copies of $G(k,M',T)$ whose k colour disjoint old supervertices contain exactly the same k sets of colours. Let these copies of $G(k,M',T)$ be called $G_1,\dots,G_k$, and let the k sets of colours that occur on the colour disjoint supervertices of each these G_i be called $Q_1,\dots,Q_k$. (Note that the Q_i are pairwise disjoint.)

For $i=1,\dots k$, let V_i be an old supervertex in G_i which contains x colours from Q_i. From the construction of $G(k+1,M,T)$ it follows that there is an added superset which, for each $1\le i\le k$, is connected by an (M,T,p)-separator graph to a superset in V_i.

Since $p=kT+1$, and since in each (M,T,p)-separator graph there are at most T pairs of (new) super vertices of size M that have some colours in common (Lemma 4.4), there must exist supervertex in the added superset which is part of a colour disjoint supervertex pair in each of these separator graphs. Hence this supervertex in the added superset, and its non-neighbours in each of the separator graphs, together form $k+1$ new supervertices that are pairwise colour disjoint. ∎

It may be clear from the construction in the preceeding proof that the size of $G(k,M,T)$ grows rapidly. Using the rather crude bounds

$$C_{T,k,x} \leq (k+1)^T$$

and

$$N_{T,k+1} \leq (N_{T,k})^{2k(k+1)^T} \leq (N_{T,2})^{\prod_{\ell=2}^{k} 2\ell(\ell+1)^T},$$

we obtain a bound on the size of $G(k,M,T)$ of $M(2T+2)^{\left(\frac{2k}{e}\right)^k \left(\frac{k+1}{e}\right)^{(k+1)T}}$. We will see in the Corrollary below that for the proof of Theorem 1.2 we take $T \leq kx+1$ and $M=1$, so our graph will have size $\mathcal{O}((kx)^{k^{4k^2 x}})$.

Corollary 4.1. *For all integers $m \geq 1$, $k \geq 2$,*

$$\tilde{\chi}_{(1,0)}(\mathcal{G}_k, m) \geq \begin{cases} \frac{1}{2}km & \text{if } m \text{ is even,} \\ \frac{1}{2}k(m-1)+1 & \text{if } m \text{ is odd.} \end{cases}$$

Proof. Fix $k \geq 2$ and $m \geq 1$. Suppose that m is even, $m = 2x$, say. Let $T = kx$, $M = 1$. By Lemma 4.3, there exists a graph, $G = G(M,T,k)$, and for each multicolouring algorithm that is allowed recolouring distance $(1,0)$ there exists a c-sequence for G which forces the algorithm to use at least T colours. Hence $\tilde{\chi}_{(1,0)}(\mathcal{G}_k, m) \geq kx = \frac{1}{2}km$.

If m is odd, say $m = 2x+1$, then we need an adapted version of Lemma 4.3. We construct $G(M,T,k)$ (here $T = kx+1$) as in this lemma, but use a slightly altered c-sequence. In the first move player A will raise the weight of the new vertices to $x+1$ instead of to x. This means that the resulting c-sequence can have chromatic number $2x+1 = m$. For the rest the proof is identical, and it tells us that for any adaptive multicolouring algorithm with recolouring distance $(1,0)$ there exists a c-sequence for $G(M,T,k)$ of chromatic number at most $2x+1$ that forces the algorithm to use at least $kx+1$ colours. Hence $\tilde{\chi}_{(1,0)} \geq kx+1 = \frac{1}{2}k(m-1)+1$. ∎

Fixed Preference Allocation is an algorithm that uses limited recolouring. To be more precise, the algorithm decides which colour is dropped when the weight at a vertex decreases, so it requires recolouring distance $(0,0)$: recolouring allowed only at the vertices where the weight changes. Fixed Preference Allocation is developed in [5], after an initial idea presented by Paul-André Raymond in [12].

ALGORITHM. Fixed Preference Allocation (FPA).

DESCRIPTION. *For a k-coloured graph and a fixed integer $n \geq k$, k different ordered lists of the n available colours are used to find a multicolouring. Each of these ordered lists is associated with one of the colour classes of the base colouring. Each vertex where the weight increases uses the colour of lowest rank in the ordering of the colour class of the base colour of this vertex, which is not used by itself or its neighbour. When the weight decreases at a vertex, the colour at the vertex that has highest rank in the ordering associated to the vertex, is dropped.*

It is clear that the performance of FPA depends on the ordered lists that are used. From results shown in [5] it can be derived that with ordered lists chosen such that the number of colours used is minimal, the following can be achieved.

Proposition 4.1. *For all integers $k \geq 2$, $m \geq 1$, FPA using $\frac{1}{2}km$ colours if m is even, and $\frac{1}{2}k(m-1)+1$ colours if m is odd, can find adaptive multicolourings for any k-coloured graph and every m-colourable c-sequence for such a graph.*

Since for all k and m, $\tilde{\chi}_{(0,0)}(\mathcal{G}_k, m) \geq \tilde{\chi}_{(1,0)}(\mathcal{G}_k, m)$, Corollary 4.1 and Lemma 4.1 together prove Theorem 1.2. (The proof that $\tilde{\chi}_{(0,0)}(\mathcal{G}_k, m)$ needs the number of colours that FPA can achieve is considerable shorter and uses less complex graphs than the proof of Lemma 4.1, which implies the equivalent statement for $\tilde{\chi}_{(1,0)}(\mathcal{G}_k, m)$)

We have thus established that for adaptive multicolouring with an allowed recolouring distance up to $(0,1)$, FPA gives the best possible performance for each class $\mathcal{G}_k$. It would be interesting to know whether, for larger recolouring distances, there exist algorithms that use fewer colours than the bound given by Theorem 1.2.

In conclusion we note that, if no knowledge of the chromatic number or of a colouring that achieves it is assumed, in other words if no base colouring can be found, the only adaptive multicolouring algorithm known is the greedy algorithm, known among engineers as Dynamic Allocation. Dynamic Allocation, like the greedy algorithm for on-line graph colouring, can do very poorly. In particular, a bipartite graph can be given in which player A can force a player following the Dynamic Allocation strategy to use arbitrarily many colours.

Finally, it might be of interest to note that the graphs constructed in the proof of Lemma 4.3 have the property that their fractional chromatic number is equal to their integral chromatic number, while their girth is at least 6.

Acknowledgements. We thank Demetrios Achlioptas for helpful discussions on on-line algorithms.

References

[1] R. Beigel and W. I. Gasarch: The mapmakers dilemma, *Discrete Appl. Math.*, **34** (1991), 37–48.

[2] D. D. Dimitrijević and J. Vučetić: Design and performance analysis of the algorithms for channel allocation in cellular networks, *IEEE Trans. Veh. Technol.*, **42** (4) (1993), 526–534.

[3] Tomás Feder and Sunil M. Shende: Online channel allocation in fdma networks with reuse constraints, *Inform. Process. Lett.*, **67**(6) (1998), 295–302.

[4] A. Gyárfás and J. Lehel: On-line and first fit colorings of graphs, *J. Graph Theory*, **12** (1988), 217–227.

[5] J. Janssen, K. Kilakos, and O. Marcotte: Fixed preference frequency allocation for cellular telephone systems, *IEEE Trans. Veh. Techn.*, **48**(2) (March 1999), 533–541.

[6] J. Janssen, D. Krizanc, L. Narayanan, and S. Shende: Distributed online frequency assignment in cellular networks, in: *Proceedings of STACS'98*, Berlin, 1998. Springer-Verlag.

[7] T. R. Jensen and B. Toft: *Graph Coloring Problems*, Interscience Series in Discrete Mathematics and Optimization, Wiley, 1995.

[8] H. A. Kierstead and W. A. Trotter: On-line graph coloring, in: L. A. McGeoch and D. D. Sleator, eds., *On-Line Algorithms: Proceedings DIMACS workshop, February 11–13, 1991*, pages 85–92. American Mathematical Society, 1992.

[9] H. A. Kierstead and W. A. Trotter: Planar graph coloring with an uncooperative partner, in: W. T. Trotter, ed., *Planar Graphs*, volume 9 of *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, pages 85–93. AMS, 1993.

[10] R. J. McEliece and K. N. Sivarajan: Performance limits for channelized cellular telephone systems, *IEEE Trans. Inform. Theory*, **40**(1), 1994.

[11] L. Narayanan and S. Shende: Static frequency assignment in cellular networks, in: *Proceedings of SIROCCO 97*, pages 215–227, Carleton Scientific Press, 1997, to appear in *Algorithmica*.

[12] P-A. Raymond: Performance analysis of cellular networks, *IEEE Trans. Commun.*, **39**(12) (1991), 1787–1793.

Jeannette Janssen

Dept. of Mathematics & Statistics,
Dalhousie University,
Halifax, B3H 3J5, Nova Scotia, Canada
janssen@mscs.dal.ca

Kyriakos Kilakos

Centre for Discrete and
Applicable Mathematics,
LSE, Houghton Street,
London WC2A 2AE, U.K.
kyri@cdam.lse.ac.uk