



# On the $\Delta(d)$ –chromatic number of a complete balanced multipartite graph

AP Burger\*      I Nieuwoudt†      JH van Vuuren‡

*Received: 26 October 2006; Revised: 29 December 2006; Accepted: 20 January 2007*

## Dedication to Emeritus Professor Gerhard Geldenhuys

*Dedicated with deep respect and appreciation to our friend, teacher and mentor, Gerhard Geldenhuys, who introduced us to the sheer beauty of graph theory in the early nineteen nineties, and who fostered in us an affinity for hard operations research.*

## Abstract

In this paper we solve (approximately) the problem of finding the minimum number of colours with which the vertices of a complete, balanced, multipartite graph  $G$  may be coloured such that the maximum degrees of all colour class induced subgraphs are at most some specified integer  $d \in \mathbb{N}$ . The minimum number of colours in such a colouring is referred to as the  $\Delta(d)$ –chromatic number of  $G$ . The problem of finding the  $\Delta(d)$ –chromatic number of a complete, balanced, multipartite graph has its roots in an open graph theoretic characterisation problem and has applications conforming to the generic scenario where users of a system are in conflict if they require access to some shared resource. These conflicts are represented by edges in a so-called resource access graph, where vertices represent the users. An efficient resource access schedule is an assignment of the users to a minimum number of groups (modelled by means of colour classes) where some threshold  $d$  of conflict may be tolerated in each group. If different colours are associated with different time periods in the schedule, then the minimum number of groupings in an optimal resource access schedule for the above set of users is given by the  $\Delta(d)$ –chromatic number of the resource access graph. A complete balanced multipartite resource access graph represents a situation of maximum conflict between members of different user groups of the system, but where no conflict occurs between members of the same user group (perhaps due to an allocation of diverse duties to the group members).

**Key words:** Graph/vertex colouring, chromatic number, maximum degree.

**AMS Classification:** 05C15.

---

\*Department of Logistics, University of Stellenbosch, Private Bag X1, Matieland, 7602, South Africa

†Applied Mathematics Division, Department of Mathematical Sciences, University of Stellenbosch, Private Bag X1, Matieland, 7602, South Africa

‡Corresponding author: Department of Logistics, University of Stellenbosch, Private Bag X1, Matieland, 7602, South Africa, email: [vuuren@sun.ac.za](mailto:vuuren@sun.ac.za)

## 1 Introduction

A *colouring* of a simple graph  $G$  is an assignment of colours to the vertices of  $G$ , one colour to a vertex, according to some rule  $\mathcal{R}$ . A colouring may therefore be thought of as a partition (induced by the rule  $\mathcal{R}$ ) of the vertex set of  $G$  into a number of subsets, called *colour classes*, such that each vertex of a particular colour class receives the same colour. For example, in the *classical graph colouring problem* the rule  $\mathcal{R}$  is that the colour classes should be independent sets (*i.e.* no two vertices in the same colour class should be adjacent). The smallest number of colours for which a classical colouring exists for a graph  $G$  is called the (classical) *chromatic number* of  $G$ , denoted by  $\chi(G)$ . The problem of determining  $\chi(G)$  for a general graph  $G$  is NP-hard [6, pp. 287–288], has been the focus of a large body of work [2, 3, 13, 16, 19] and has numerous applications. Perhaps the most prevalent operational research applications of this problem occur in timetabling and scheduling endeavours [9]. Storage problems in supply chains may also be resolved by means of graph colouring models [9, 12]. Graph colouring applications in the information technology sector include the so-called *register allocation problem* where the objective is to assign (program) variables to a limited number of computer registers (hardware devices) during execution [12, 15, 20], and the *problem of testing printed electronic circuit boards* for unintended short circuits (caused by stray soldering lines) [20].

In this paper we consider a different colouring rule  $\mathcal{R}$ , namely that the maximum degrees of all colour class induced subgraphs should be bounded from above by some specified integer, and we seek the minimum number of colour classes satisfying this requirement. We make this definition more precise and also outline an application of this colouring problem in the following section. This is followed by our main objective, namely seeking solutions to the above colouring problem for specifically the class of complete balanced multipartite graphs (because this class of graphs seems to play a central role in an open graph theoretic characterisation problem and because the particular colouring problem has natural applications when considering this graph class). In §3 a normalised colouring procedure according to our rule  $\mathcal{R}$  is described. This colouring strategy is then used in §4 to determine an ideal colouring procedure according to our rule  $\mathcal{R}$ , where fractions of vertices may be coloured. In the remainder of §4 this idealized situation is corrected by discretising the fractions of vertices that should be coloured. We conclude the paper in §5 with a summary of what has been achieved, as well as an indication of possible future work and open problems.

## 2 The notion of a $\Delta$ -colouring

If  $x$  colours are used in a colouring of a graph  $G$ , then the colouring is referred to as an  $x$ -colouring of  $G$ . We take the colouring rule  $\mathcal{R}$  to be the requirement that the induced subgraph of each colour class should have a maximum degree not exceeding some specified  $d \in \mathbb{N}$  and we refer to an  $x$ -colouring of a graph  $G$  satisfying this requirement as a  $\Delta(d, x)$ -colouring of  $G$ . The smallest value of  $x$  for which a  $\Delta(d, x)$ -colouring of a graph  $G$  exists (for some fixed value of  $d$ ) is called the  $\Delta(d)$ -chromatic number of  $G$ , denoted by  $\chi_d^\Delta(G)$ , and such an optimal colouring is called a  $\chi_d^\Delta$ -colouring of  $G$ .

The parameter  $\chi_d^\Delta(G)$  admits the classical chromatic number as special case, in the sense that  $\chi_0^\Delta(G) = \chi(G)$ , and the problem of finding the  $\Delta(d)$ -chromatic number of a graph was first considered independently and almost simultaneously by Andrews and Jacobson [1], Harary and Jones [14] and Cowen *et al.* [7] during the mid 1980s. Since then a substantial amount of work has been done on this parameter. For instance, the classical chromatic number of a graph  $G$  may be bounded in terms of the  $\Delta(d)$ -chromatic number of  $G$ . More specifically, Andrews and Jacobson [1] showed that

$$\chi_d^\Delta(G) \leq \chi(G) \leq (d+1)\chi_d^\Delta(G). \quad (1)$$

Furthermore, Frick [10] and Lovász [17] established the bounds

$$\left\lceil \frac{\omega(G)}{d+1} \right\rceil \leq \chi_d^\Delta(G) \leq \min \left\{ \left\lceil \frac{v(G)}{d+1} \right\rceil, \left\lceil \frac{\Delta(G)+1}{d+1} \right\rceil \right\} \quad (2)$$

on the  $\Delta(d)$ -chromatic number of a graph  $G$ , where  $\omega(G)$ ,  $v(G)$  and  $\Delta(G)$  denote respectively the clique number, the order and the maximum degree of  $G$  [6, pp 3, 16 & 284].

Cowen *et al.* [7] focussed on graphs embedded on various surfaces and gave a complete characterization of all  $d, x \in \mathbb{N}$  for which  $\Delta(d, x)$ -colourings exist for every planar and every outerplanar graph.

The  $\Delta(d)$ -chromatic numbers of a number of simple graph classes are summarised in the following theorem, the proof of which is omitted due to the simplicity of the underlying arguments [18, Chapter 4].

**Theorem 1** *Suppose  $d \in \mathbb{N}_0$  and  $m, n \in \mathbb{N}$ . Let  $T_n$ ,  $C_n$ ,  $K_n$  and  $K_{m,n}$  denote respectively a tree, a cycle, a complete graph of order  $n$  and a complete bipartite graph with partite sets of cardinalities  $m$  and  $n$  [6, pp 25, 27 & 62]. Then*

$$(a) \quad \chi_d^\Delta(T_n) = \begin{cases} 1 & \text{if } d \geq \Delta(T_n), \\ 2 & \text{if } d < \Delta(T_n). \end{cases}$$

$$(b) \quad \chi_d^\Delta(C_n) = \begin{cases} 1 & \text{if } d \geq 2, \\ 2 & \text{if } d = 0 \text{ and } n \text{ is even, or if } d = 1, \\ 3 & \text{if } d = 0 \text{ and } n \text{ is odd.} \end{cases}$$

$$(c) \quad \chi_d^\Delta(K_n) = \lceil n/(d+1) \rceil.$$

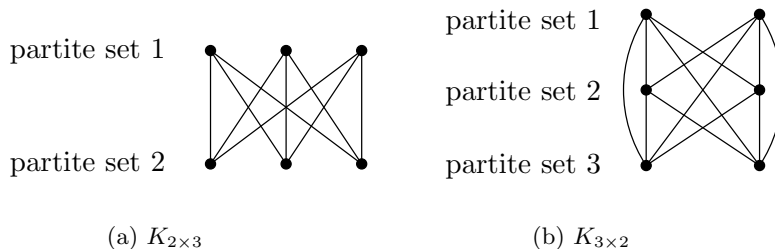
$$(d) \quad \chi_d^\Delta(K_{m,n}) = \begin{cases} 1 & \text{if } d \geq \max\{m, n\}, \\ 2 & \text{otherwise.} \end{cases} \quad \blacksquare$$

In this paper we are interested in determining the  $\Delta(d)$ -chromatic number for the class of complete balanced multipartite graphs. The reason for our interest in this graph class follows from an attempted characterisation of the sequence  $\{\chi_d^\Delta(G) : d \in \mathbb{N}_0\}$  for a general graph  $G$ , *i.e.* to determine which integral sequences are, in fact,  $\Delta(d)$ -chromatic.

Our interest in determining the  $\Delta(d)$ -chromatic number for the class of complete balanced multipartite graphs also stems from a scheduling application where users of a computer system are in conflict if they require access to one or more of the same data files [8]. These conflicts are represented by edges in a so-called file access graph, where vertices represent

the users. An efficient access schedule is an assignment of the users to a minimum number of groups (modelled by means of colour classes) where some threshold of conflict may be tolerated in each group. For example, users may find the slowdown incurred by the retrieval of data simultaneous with, say two, other (conflicting) users of the system acceptable, but the slowdown caused by more than two other (conflicting) users unacceptable. If different colours are associated with different time periods in the schedule, the file access schedule for the above set of users, may be found by means of a  $\Delta(2, x)$ -colouring of the file access graph. A complete, balanced, multipartite file access graph represents a situation of maximum conflict between members of different user groups of the computer system, but where no conflict occurs between members of the same user group (perhaps due to an allocation of diverse duties to the group members).

Let  $K_{k \times n}$  denote a complete balanced multipartite graph consisting of  $k$  partite sets, each of cardinality  $n$  (see, for example, Figure 1 for graphical representations of the two complete, balanced multipartite graphs  $K_{2 \times 3}$  and  $K_{3 \times 2}$ ).



**Figure 1:** Graphical representations of two complete balanced multipartite graphs.

Consider the *sequence of  $\Delta(d)$ -chromatic numbers* for  $K_{k \times n}$  as the colour class induced maximum degree bound  $d$  is increased, starting from zero, that is  $\chi_0^\Delta(K_{k \times n})$ ,  $\chi_1^\Delta(K_{k \times n})$ ,  $\chi_2^\Delta(K_{k \times n})$ ,  $\dots$ . Examples of such sequences are shown in Example 1.

**Example 1** *The sequences of  $\Delta(d)$ -chromatic numbers for  $K_{4 \times 4}$  and  $K_{4 \times 8}$  are given below.*

$d$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	...	23	24	...
$\chi_d^\Delta(K_{4 \times 4})$	4	4	4	3	2	2	2	2	2	2	2	2	1	1	...	1	1	...
$\chi_d^\Delta(K_{4 \times 8})$	4	4	4	4	4	4	3	3	2	2	2	2	2	2	...	2	1	...

*Notice that the number of repetitions of each entry in the sequence of  $\Delta(d)$ -chromatic numbers for  $K_{4 \times 8}$  are exactly double the number of repetitions in the sequence of  $\Delta(d)$ -chromatic numbers for  $K_{4 \times 4}$ . □*

The structure of the sequences in the above example is typical, in the sense that  $\chi_0^\Delta(K_{k \times n}) = \chi(K_{k \times n}) = k$  and the sequences are non-increasing, usually involving a large amount of repetition and ending in an infinite tail of ones. To determine the sequences in Example 1, one therefore only needs to know at which values of  $d$  the sequence values change. Hence, it makes sense to invert the  $\Delta(d)$ -chromatic problem by rather fixing the number

of colours,  $x$  say, that may be used and then seeking to minimise the colour class induced maximum degree (*i.e.* seeking the smallest value of  $d$  for which there exists a  $\Delta(d, x)$ -colouring of  $G$ , for some fixed value of  $x$ ). Denote the answer to this inverted problem by  $D_x^\Delta(G)$  and call such an optimal colouring a  $D_x^\Delta$ -colouring of  $G$ . Note that a  $\chi_d^\Delta$ -colouring of  $G$  is not necessarily a  $D_x^\Delta$ -colouring of  $G$ . The value of  $D_x^\Delta(G)$  gives the *position* in the maximum degree chromatic sequence where the number of colours  $x$  appears for the first time. Hence, the first sequence in Example 1 may be captured by the three values  $D_3^\Delta(K_{4 \times 4}) = 3$ ,  $D_2^\Delta(K_{4 \times 4}) = 4$  and  $D_1^\Delta(K_{4 \times 4}) = 12$ .

Returning to our file access application, if we use too many time slots (colours), the waiting period for each user until the next time slot during which (s)he may retrieve data may be too long, resulting in an unacceptable slowdown of the system. Therefore, one may also decide to fix the number of available time slots and schedule the users in such a way that the threshold of conflict per time slot is minimized. If  $x$  time slots are fixed for the file access graph  $G$ , the solution to this minimisation problem is  $D_x^\Delta(G)$ .

It is possible to establish the following relationship between  $\chi_d^\Delta(G)$  and  $D_x^\Delta(G)$ , by noting that if there exists a  $\Delta(d, x)$ -colouring of  $G$ , for some fixed values  $d, x \in \mathbb{N}$ , then the inequalities  $\chi_d^\Delta(G) \leq x$  and  $D_x^\Delta(G) \leq d$  both immediately follow.

**Proposition 1** For any graph  $G$ ,

- (a)  $\chi_d^\Delta(G) \leq x$  if and only if  $D_x^\Delta(G) \leq d$ ,
- (b)  $\chi_d^\Delta(G) = x$  if and only if  $D_x^\Delta(G) \leq d$  and  $D_{x-1}^\Delta(G) > d$ ,
- (c)  $D_x^\Delta(G) = d$  if and only if  $\chi_d^\Delta(G) \leq x$  and  $\chi_{d-1}^\Delta(G) > x$ .

**Proof:** (a) Suppose  $\chi_d^\Delta(G) \leq x$  for some graph  $G$ . Then there exists a  $\Delta(d, x)$ -colouring of  $G$ , implying that  $D_x^\Delta(G) \leq d$ . Conversely, suppose  $D_x^\Delta(G) \leq d$ . Then again there exists a  $\Delta(d, x)$ -colouring of  $G$ , implying that  $\chi_d^\Delta(G) \leq x$ .

(b)–(c) These results follow from (a) and its contra-positive. ■

For the  $\Delta(d)$ -chromatic number of a graph  $G$  and its related inversion number,  $D_x^\Delta(G)$ , the following growth properties hold.

**Proposition 2 (Growth properties)**

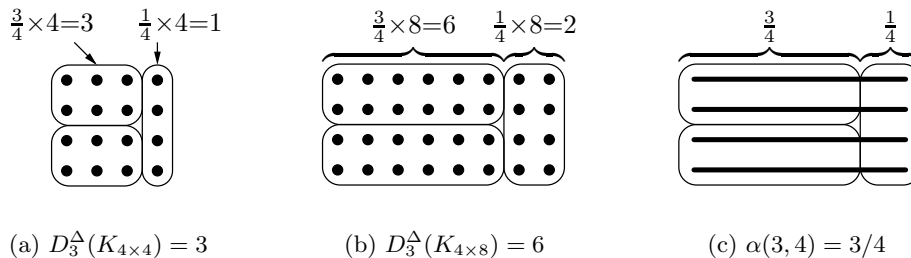
- (a)  $1 \leq \chi_{d+1}^\Delta(G) \leq \chi_d^\Delta(G) \leq \chi(G)$  for any graph  $G$  and any  $d \in \mathbb{N}_0$ .
- (b)  $0 \leq D_{x+1}^\Delta(G) \leq D_x^\Delta(G) \leq \Delta(G)$  for any graph  $G$  and any  $x \in \mathbb{N}$ .

**Proof:** (a) The lower bound follows from the non-emptiness of the vertex set in the definition of a simple graph and is attained when  $d \geq \Delta(G)$ . The upper bound corresponds to the special case where  $d = 0$ . Let  $\chi_d^\Delta(G) = x^*$ . Then any  $\Delta(d, x^*)$ -colouring of a graph  $G$  is also a  $\Delta(d+1, x^*)$ -colouring of  $G$ . Therefore  $\chi_{d+1}^\Delta(G) \leq x^*$ .

(b) The upper bound is attained when  $x = 1$ . The lower bound is the smallest possible value that the maximum degree of a graph may have. Let  $D_x^\Delta(G) = d^*$ . Then, for any  $\Delta(d^*, x)$ -colouring of a graph  $G$ , there exists a  $\Delta(d^*, x+1)$ -colouring of  $G$ . Hence  $D_{x+1}^\Delta(G) \leq d^*$ . ■

### 3 Normalised $k$ -partite colourings

The sequences in Example 1 are similar in the sense that the number of times that any specific entry in the first sequence appears is double that in the second sequence. Therefore, the sequence of numbers  $\{D_x^\Delta(\bullet)\}_{x=1,2,3,\dots}$  for  $K_{4 \times 8}$  is double the corresponding numbers for  $K_{4 \times 4}$  (e.g.  $D_3^\Delta(K_{4 \times 4}) = 3$  and  $D_3^\Delta(K_{4 \times 8}) = 6$ ). Graphical representations of  $D_3^\Delta$ -colourings of  $K_{4 \times 4}$  and  $K_{4 \times 8}$  are shown in Figure 2(a) and (b); notice that the colouring structures in Figures 2(a) and (b) are the same. From the figure it is clear that we are solving essentially the same problem.



**Figure 2:** Graphical representations of  $D_3^\Delta$ -colourings of (a)  $K_{4 \times 4}$  and (b)  $K_{4 \times 8}$  with the same colouring pattern. The dots in each row of (a) and (b) represent the vertices of a partite set of  $K_{4 \times n}$ , while a colour class is represented by means of a rectangular frame. In (c) a similarly structured normalised  $\Delta(\frac{3}{4}, 3)$  4-partite colouring is given where the partite sets are viewed as line segments of unit length.

It therefore makes sense to view the problem of determining  $D_x^\Delta(K_{k \times n})$  in a continuous setting where each partite set is replaced by a line segment of unit length; hence disregarding the actual number of vertices in each partite set. In such a continuous setting colour classes originally comprising discrete entities (graph vertices) are replaced by collections of (continuous) line subsegments, called *normalised colour classes*, representing the original colour classes in a normalised sense. In this continuous setting the notion of a colour class induced maximum degree is replaced by that of a *normalised colour class induced maximum degree* — the sum total of the lengths of all subsegments in a normalised colour class, except for the shortest subsegment in that class. If the lengths of all subsegments in a normalised colour class are the same, we shall refer to this length as the *normalised width* of the normalised colour class. If  $x$  colours are used and if the largest normalised colour class induced maximum degree is  $\tilde{d} \in \mathbb{R}$ , then the resulting colouring is referred to as a *normalised  $\Delta(\tilde{d}, x)$   $k$ -partite colouring*. A graphical representation of a normalised  $\Delta(\frac{3}{4}, 3)$  4-partite colouring is shown in Figure 2(c).

Let  $\alpha(x, k)$  denote the smallest value of  $\tilde{d}$  for which a normalised  $\Delta(\tilde{d}, x)$   $k$ -partite colouring exists; such an optimal colouring is called a *normalised  $k$ -partite  $D_x^\Delta$ -colouring*. We shall show in the following section how an upper bound on  $\alpha(x, k)$  may be used to produce a  $\Delta(d, x)$ -colouring of  $K_{k \times n}$  for any  $n \in \mathbb{N}$  via a discretisation process, where  $d$  is related to  $n$  and  $\tilde{d}$  in a manner that will be specified. However, in this section our focus will be on the continuous problem of determining  $\alpha(x, k)$ . This seems to be a hard problem that

has only been solved approximately. The following general bounds on  $\alpha(x, k)$  are known [5, Proposition 2].

**Proposition 3**  $k/x - 1 \leq \alpha(x, k) \leq \lceil k/x \rceil - 1$ . ■

From the above proposition it is clear that  $\alpha(x, k) = k/x - 1$  when  $x|k$ ; this is Case 1 in Table 1. However, based on the divisibility properties of  $k$  by  $x$ , a further three cases arise, as shown in Table 1; the results for Cases 2–4 in the table were proved in [5, Theorem 2]. Here we merely capture the essence of the four cases listed in Table 1.

	$s = 0$	$0 < s < \lceil \frac{k}{x} \rceil - 1$	$s = \lceil \frac{k}{x} \rceil - 1$
$x' = 0$	<b>Case 1 (Trivial)</b> $\alpha(x, k) = \frac{k}{x} - 1$	<b>Case 2 (Balancing)</b> $\alpha(x, k) = \frac{(k-1)(\lceil \frac{k}{x} \rceil - 1)}{\lceil \frac{k}{x} \rceil x - 2}$	<b>Case 3 (Reduction)</b> $\alpha(x, k) \leq \alpha(x-1, k-s)$
$x' > 0$	<b>Case 4 (Composition)</b> $\alpha(x, k) \leq \frac{1}{\frac{1}{\alpha(x-x', k)} + \frac{1}{\alpha(x', k)}}$		

**Table 1:** Values of and upper bounds on  $\alpha(x, k)$ , depending on the values of  $s \equiv k \pmod{\lceil \frac{k}{x} \rceil}$  and  $x' = x - \lceil k/\lceil \frac{k}{x} \rceil \rceil$ .

For all four cases we start by forming  $\lceil k/\lceil k/x \rceil \rceil$  normalised colour classes, each comprising  $\lceil k/x \rceil$  full partite sets, except possibly for the last utilised colour class, which comprises  $s \equiv k \pmod{\lceil k/x \rceil}$  full partite sets if  $x \nmid k$  (in the case where  $x|k$ , all colour classes comprise the same number of partite sets). Upon completion of this partition of the partite sets,  $x' = x - \lceil k/\lceil k/x \rceil \rceil$  colours remain unutilised. The values of  $s$  and  $x'$  are used as decision variables to distinguish between the cases in Table 1 in our recursive normalised  $\Delta(\vec{d}, x)$   $k$ -partite colouring strategy given in pseudo-code as Algorithm  $\bar{\alpha}(x, k)$ , which outputs an upper bound  $\bar{\alpha}(x, k)$  on  $\alpha(x, k)$ .

**Algorithm**  $\bar{\alpha}(x, k)$

**Input:**  $x, k \in \mathbb{N}$ .

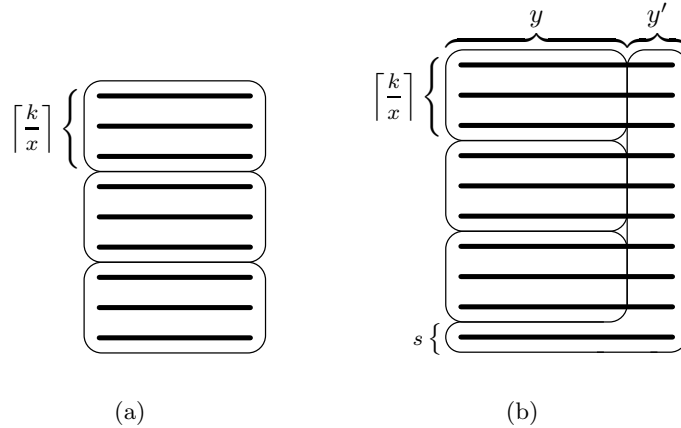
**Output:** An upper bound on  $\alpha(x, k)$ .

- 
1. **if**  $x \mid k$  **then output**  $\frac{k}{x} - 1$
  2.  $x' \leftarrow x - \lceil k/\lceil \frac{k}{x} \rceil \rceil$ ,  $s \leftarrow k \pmod{\lceil \frac{k}{x} \rceil}$
  - 3.1 **if**  $s = \lceil \frac{k}{x} \rceil - 1$  **then output**  $\bar{\alpha}(x-1, k-s)$
  - 3.2 **else if**  $x' > 0$  **then output**  $1/(\frac{1}{\bar{\alpha}(x-x', k)} + \frac{1}{\bar{\alpha}(x', k)})$
  - 3.3 **else output**  $\frac{(k-1)(\lceil \frac{k}{x} \rceil - 1)}{\lceil \frac{k}{x} \rceil x - 2}$
-

It should be clear from the pseudo-code listing that Algorithm  $\bar{\alpha}(x, k)$  functions directly according to the classification of cases in Table 1. We describe the colouring substrategy and hence motivate the values of  $\bar{\alpha}(x, k)$  attained in each case separately.

**Case 1 (Trivial)** The case where  $x' = 0$  and  $s = 0$  represents a terminating call of Algorithm  $\bar{\alpha}(x, k)$ . If this is also the first call (*i.e.* if  $x|k$ ), then a colouring of the form depicted in Figure 3(a) results. Otherwise a colouring of this structure will be a subconstruction of the overall colouring strategy involving line subsegments of normalised width  $y = \bar{\alpha}(x, k)/(\lceil k/x \rceil - 1)$ , because in this case the degree  $\bar{\alpha}(x, k)$  of each colour class is the normalised width  $y$  multiplied by the number of line subsegments included per colour class ( $\lceil k/x \rceil$ ), less one.

**Case 2 (Balancing)** In the case where  $x' = 0$  and  $0 < s < \lceil k/x \rceil - 1$  the normalised maximum degree of the last colour class is sufficiently smaller than that of the other colour classes to allow recolouring of a small proportion  $y'$  of each of the line segments of the other colour classes using the last colour, so as to increase the maximum degree of the last colour class and decrease the maximum degree of the other colour classes. The normalised width  $y'$  is determined such that the maximum degree of the last colour class is the same as the maximum degree of the other colour classes, *i.e.* such that  $ys + (k - 1)y' = \bar{\alpha}(x, k)$ . If  $y$  is calculated as in Case 1 and substituted in this expression, it follows that  $y' = \bar{\alpha}(x, k)(\lceil k/x \rceil - s - 1)/((\lceil k/x \rceil - 1)(k - 1))$ . This case is illustrated graphically in Figure 3(b), where the inverted, L-shaped frame at the bottom represents the last colour class after balancing.



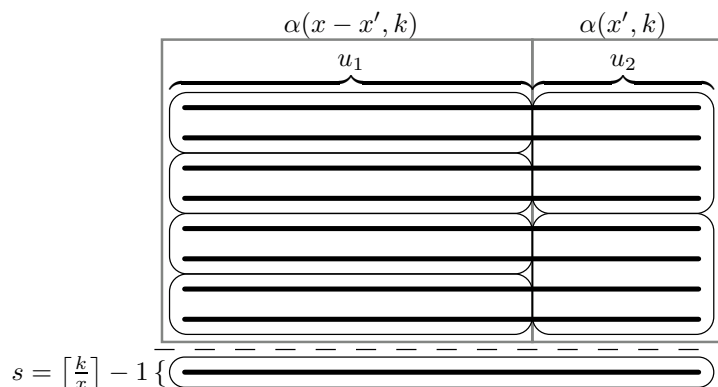
**Figure 3:** Graphical representation of the colouring approach in Algorithm  $\bar{\alpha}(x, k)$  for (a) Case 1 and (b) Case 2, where a balancing operation is performed on the last colour class. As before, the line segments represent normalised partite sets and colour classes are represented by means of rectangular or inverted, L-shaped frames.

**Case 3 (Reduction)** In the case where  $s = \lceil k/x \rceil - 1$  the last utilised colour class comprises exactly one partite set fewer than the other colour classes, and hence it is not possible to perform a balancing operation (as described in Case 2). In this case the partite sets comprising the last utilised colour class do not form part of future



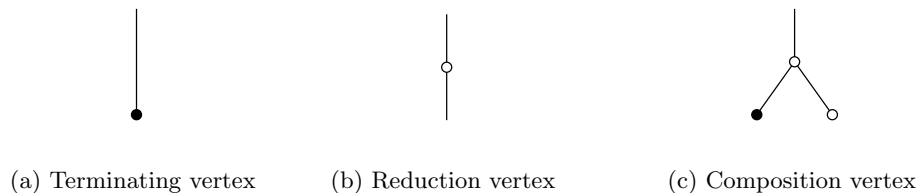
calls of the algorithm. This is equivalent to reducing the problem of determining  $\bar{\alpha}(x, k)$  for a smaller problem, namely with one colour fewer and with  $k - s$  partite sets. Notice that if  $x' = 0$ , the next recursive call of Algorithm  $\bar{\alpha}(x, k)$  will be the trivial case (Case 1). However, if  $x' > 0$ , then the next recursive call will be a composition (Case 4 described below). This case is illustrated graphically in conjunction with the case of composition in Figure 4.

**Case 4 (Composition)** If  $x' > 0$  and  $0 \leq s < \lceil \frac{k}{x} \rceil - 1$ , then we solve two problems separately and combine the results obtained by means of the composition formula given in Case 4 of Table 1. All partite sets are divided in two portions, one of length  $u_1$  and one of length  $u_2$ , such that  $u_1 + u_2 = 1$ . We then combine two normalised  $\Delta(\vec{d}, x)$   $k$ -partite colouring strategies obtained by using respectively  $x'$  and  $x - x'$  colours on two collections of shorter  $k$ -partite sets — the values of  $u_1$  and  $u_2$  are chosen so that  $u_1 \bar{\alpha}(x - x', k) = u_2 \bar{\alpha}(x', k)$ . Note that the one problem (resulting in the bound  $\bar{\alpha}(x - x', k)$ ) always terminates immediately either as the trivial or as the balancing case. The other problem (resulting in the bound  $\bar{\alpha}(x', k)$ ) may call any of the four sub-construction cases, and therefore may or may not terminate immediately. This case is illustrated graphically in conjunction with the case of reduction in Figure 4.



**Figure 4:** Graphical representation of the output of Algorithm  $\bar{\alpha}(x, k)$  if the first call is a reduction call (Case 3) and the second call is a composition call (Case 4) for which both recursive subcalls are terminating.

The progress of Algorithm  $\bar{\alpha}(x, k)$  may be captured by the construction of a so-called *progress tree* in which vertices represent algorithm calls. All vertices in this binary tree may be classified as one of three types, namely (a) a *terminating vertex* of degree 1 corresponding to a Case 1 or 2 terminating call in Table 1, (b) a *reduction vertex* of degree 2 corresponding to a Case 3 non-terminating call and (c) a *composition vertex* of degree 3 corresponding to a Case 4 non-terminating call. These types of vertices are shown graphically in Figure 5. Note that because the call  $\bar{\alpha}(x - x', k)$  following a Case 4 algorithmic call is necessarily a terminating call, the corresponding descendant of the composition vertex will be a terminating vertex — our convention will be to place this descendant on the left. Furthermore, the terminating vertices in the progress tree are numbered in increasing



**Figure 5:** Graphical representations of the types of vertices in the progress tree of Algorithm  $\bar{\alpha}(x, k)$ .

order first from the top level in the tree downwards and from the left to the right within each level.

Suppose a total of  $\ell \in \mathbb{N}$  terminating calls are made during the execution of Algorithm  $\bar{\alpha}(x, k)$  and denote the values of  $x$ ,  $x'$ ,  $k$  and  $s$  during the  $j$ -th terminating call by  $x_j$ ,  $x'_j$ ,  $k_j$  and  $s_j$  respectively ( $j = 1, \dots, \ell$ ). Furthermore, let

$$a_j = \left\lceil \frac{k_j}{x_j} \right\rceil - 1, \quad j = 1, \dots, \ell. \quad (3)$$

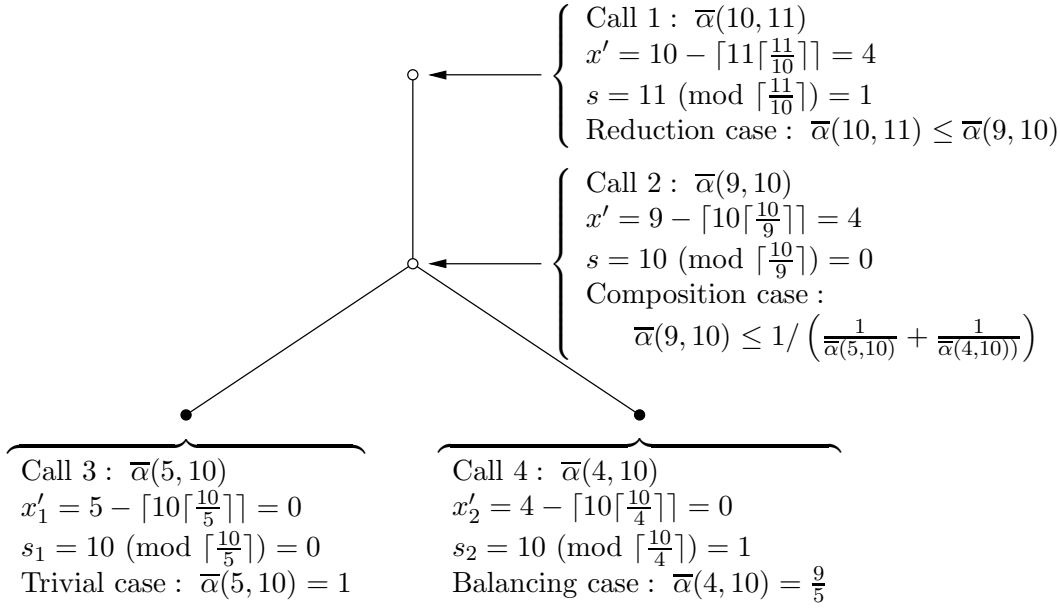
Then  $a_j + 1$  is the number of line subsegments in each colour class (except possibly for the last colour class) during the  $j$ -th terminating call of Algorithm  $\bar{\alpha}(x, k)$ . Finally, let  $y_j$  be the value of  $y$  at the  $j$ -th terminating call (as described in Cases 1 and 2), and if the  $j$ -th terminating call is a Case 2 (balancing) call, then let  $y'_j$  be the value of  $y'$  during that call. The working of Algorithm  $\bar{\alpha}(x, k)$  is illustrated by means of a simple example.

**Example 2** Suppose we seek a normalised 11-partite  $D_{10}^\Delta$ -colouring. The values of the variables  $x$ ,  $k$ ,  $s$  and  $x'$  during each recursive call of Algorithm  $\bar{\alpha}(x, k)$  are given in Figure 6.

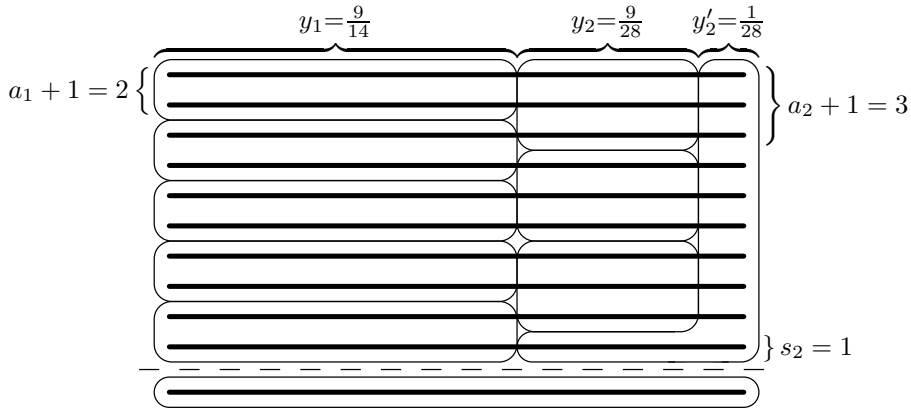
A total of four calls of Algorithm  $\bar{\alpha}(x, k)$  are made, two of which are terminal, as may be seen in Figure 6. We conclude that  $\alpha(10, 11) \leq \bar{\alpha}(10, 11) = 1/(1 + 1/(9/5)) = 9/14$ . A graphical representation of a normalised  $\Delta(\vec{d}, 10)$  11-partite colouring achieving this bound is shown in Figure 7.  $\square$

We believe Algorithm  $\bar{\alpha}(x, k)$  produces good upper bounds. These bounds are, in fact, exact when only one recursive call is made. However, the upper bound produced by the algorithm is not optimal in general — there are rare cases where it is possible to improve slightly upon the construction produced by the algorithm, as illustrated in the following example.

**Example 3** Suppose we seek a normalised 19-partite  $D_9^\Delta$ -colouring. The construction produced by Algorithm  $\bar{\alpha}(x, k)$ , which results in the bound  $\alpha(9, 19) \leq \bar{\alpha}(9, 19) = (\bar{\alpha}(7, 19)^{-1} + \bar{\alpha}(2, 19)^{-1})^{-1} = (19/36 + 1/9)^{-1} = 36/23$ , is shown in Figure 8(a). However, an alternative normalised  $\Delta(\vec{d}, 9)$  19-partite colouring corresponding to the bound  $\alpha(9, 19) \leq \bar{\alpha}(7, 15) = (\bar{\alpha}(5, 15)^{-1} + \bar{\alpha}(2, 15)^{-1})^{-1} = (1/2 + 1/7)^{-1} = 14/9$  is shown in Figure 8(b).



**Figure 6:** The progress tree of Algorithm  $\bar{\alpha}(x, k)$  for Example 2 where we seek a normalised 11-partite  $D_{10}^\Delta$ -colouring.

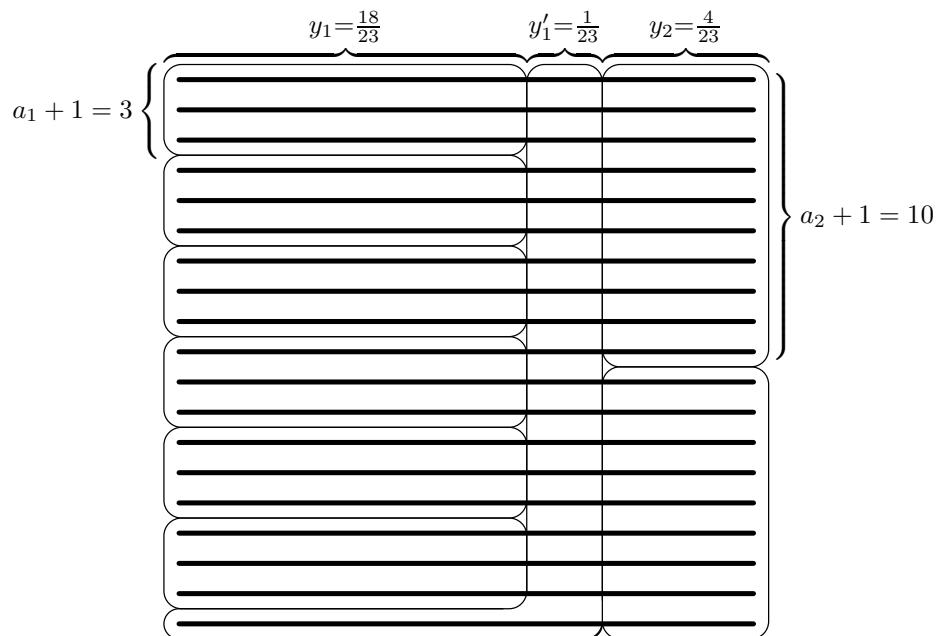


**Figure 7:** Graphical representation of a normalised  $\Delta(\tilde{d}, 10)$  11-partite colouring with a normalised colour class induced maximum degree of  $\bar{\alpha}(10, 11) = 9/14$ .

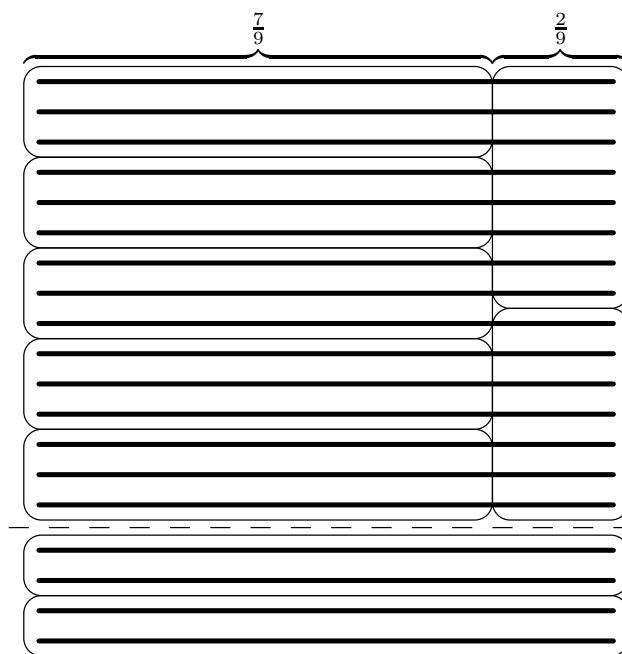
The latter colouring represents an improvement of  $2/207$  over the upper bound on  $\alpha(9, 19)$  established by the colouring produced by Algorithm  $\bar{\alpha}(x, k)$ .  $\square$

As is evident from the above example, it is sometimes more beneficial to perform a “double reduction” instead of performing a balancing operation as dictated by Algorithm  $\bar{\alpha}(x, k)$ . It is not clear exactly when this scenario occurs, but the smallest value of  $k$  for which this phenomenon occurs is  $(x, k) = (9, 19)$ . Other known cases are  $(7, 22)$ ,  $(10, 22)$  and  $(12, 25)$ .

In [5, Theorem 4] Algorithm  $\bar{\alpha}(x, k)$  was shown to exhibit a worst-case time complexity of  $\mathcal{O}(\log \log x)$ .



(a)



(b)

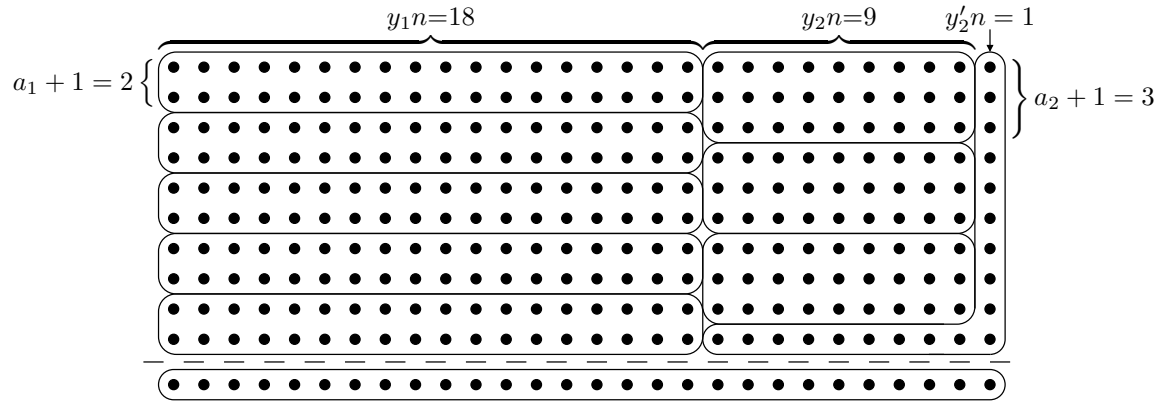
**Figure 8:** A graphical representation of one of the rare cases, where  $x = 9$  and  $k = 19$ , for which the colouring approach in Algorithm  $\bar{\alpha}(x, k)$  may be improved upon slightly. In (a) the colouring corresponding to Algorithm  $\bar{\alpha}(x, k)$  is given with  $\bar{\alpha}(9, 19) = 36/23$ . A different colouring construction where a “double reduction” is made instead of a balancing subconstruction, is given in (b) resulting in the bound  $\alpha(9, 19) \leq 14/9$ .

## 4 Discrete colourings for $K_{k \times n}$

In this section our objective is to establish an upper bound  $\overline{D}_x^\Delta(K_{k \times n})$  on  $D_x^\Delta(K_{k \times n})$  for various values of  $n$  in terms of the upper bound  $\overline{\alpha}(x, k)$  on  $\alpha(x, k)$  determined in the previous section. Note that, if  $y_j n$  and  $y'_j n$  are integers for all  $j = 1, \dots, \ell$ , then we may of course take  $\overline{\alpha}(x, k)n$  as an upper bound on  $D_x^\Delta(K_{k \times n})$ , as illustrated by means of a numerical example below. However, in other cases the (continuous) colouring strategy of the previous section has to be discretised in order to avoid the situation where fractions of vertices are coloured. In cases where fractions of vertices are coloured we refer to the colouring as an *ideal  $\Delta(d, x)$ -colouring* with an *ideal colour class induced maximum degree*.

**Example 4** In Example 2 we sought a normalised 11-partite  $D_{10}^\Delta$ -colouring. The bound  $\overline{\alpha}(10, 11) = 9/14$  was determined by means of Algorithm  $\overline{\alpha}(x, k)$ . It may be verified that  $y_1 = 9/14$ ,  $y_2 = 9/28$  and  $y'_2 = 1/28$  in this case.

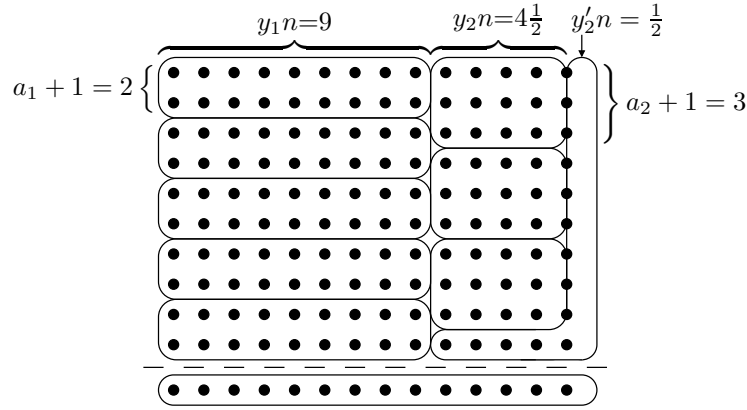
(a) Suppose  $n = 28$ . Then  $y_1 n = 18$ ,  $y_2 n = 9$  and  $y'_2 n = 1$  are integral. Thus,  $\overline{D}_{10}^\Delta(K_{11 \times 28}) = \overline{\alpha}(10, 11)n = 18$  is an appropriate upper bound on  $D_{10}^\Delta(K_{11 \times 28})$ , as witnessed by the  $\Delta(18, 10)$ -colouring of  $K_{11 \times 28}$  shown graphically in Figure 9.



**Figure 9:** Graphical representation of a  $\Delta(18, 10)$ -colouring of  $K_{11 \times 28}$ .

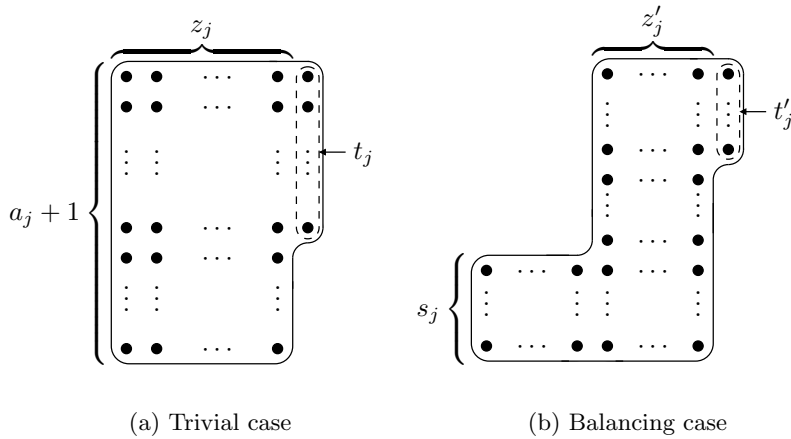
(b) Suppose now  $n = 14$ . Then  $y_1 n = 9$ ,  $y_2 n = 9/2$  and  $y'_2 n = 1/2$  are not all integral. Thus, an upper bound on  $D_{10}^\Delta(K_{11 \times 14})$  remains to be determined. However, a graphical representation of an ideal  $\Delta(9, 10)$ -colouring of  $K_{11 \times 14}$  may be found in Figure 10.  $\square$

The values of  $y_2 n$  and  $y'_2 n$  in Example 4(b) are not integral; hence fractions of vertices may be found in some colour classes if the graph were to be coloured according to the colour structure shown in Figure 10. When the colouring is discretised,  $\overline{\alpha}(x, k)n$  may not necessarily be an upper bound on  $D_x^\Delta(K_{k \times n})$ . One possible strategy to determine an upper bound on  $D_x^\Delta(K_{k \times n})$  would be to successively attempt achieving colour class induced maximum degrees  $\lceil \overline{\alpha}(x, k)n \rceil$ ,  $\lceil \overline{\alpha}(x, k)n \rceil + 1$ ,  $\lceil \overline{\alpha}(x, k)n \rceil + 2, \dots$  using  $x$  colours, until a value  $\overline{D}$  is found for which no colour class induced maximum degree exceeds  $\overline{D}$ . In the remainder of this section we shall show that if a value of  $\lceil \overline{\alpha}(x, k)n \rceil$  for  $\overline{D}$  does not suffice,  $\overline{D} = \lceil \overline{\alpha}(x, k)n \rceil + 1$  will certainly suffice. However,  $\overline{D} = \lceil \overline{\alpha}(x, k)n \rceil$  seems to be sufficient in most cases.



**Figure 10:** Graphical representation of an ideal  $\Delta(9,10)$ -colouring of  $K_{11 \times 14}$ .

In a bid to formalise the discretisation process we introduce some terminology and notation. We use the normalised width  $y_j$  and scale it with a factor  $n$  to arrive at the *ideal width*  $y_j n$  (where fractional vertex colouring is allowed). However, if we discretise the ideal colour class induced maximum degree,  $\bar{\alpha}(x, k)n$ , to  $\bar{D}$ , we potentially increase the ideal width associated with the degree of that colour class. We call this width the *ideal width associated with degree  $\bar{D}$* . Let  $z_j$  represent the integer part of the ideal width associated with degree  $\bar{D}$ . We may then add  $t_j$  vertices (at most one vertex per partite set) to each colour class formed during the  $j$ -th terminating call of Algorithm  $\bar{\alpha}(x, k)$  such that  $\bar{D} = a_j z_j + t_j$ , giving us the desired maximum degree for the colour class. See Figure 11(a) for a graphical illustration of the meaning of the parameters  $z_j$  and  $t_j$ . If balancing occurs, we denote the value of  $z_j$  for the last colour class by  $z'_j$ . The parameter  $t'_j$  has a similar definition, as illustrated in Figure 11(b).



**Figure 11:** Graphical representation of the discretisation process.

It is easy to verify that  $z_j = \lfloor \bar{D}/a_j \rfloor$ . Furthermore,  $t_j = \bar{D} - a_j \lfloor \bar{D}/a_j \rfloor$ . (Note that  $t_j$  cannot have the value  $a_j$ .) Let  $W_j$  denote the *average discretised width* of each colour class

at the  $j$ -th terminating call of Algorithm  $\bar{\alpha}(x, k)$ . Then  $W_j = \lfloor \bar{D}/a_j \rfloor + t_j/(a_j + 1) = (\lfloor \bar{D}/a_j \rfloor + \bar{D})/(a_j + 1)$  is an approximation of the ideal width  $\bar{D}/a_j$  associated with the degree  $\bar{D}$ . Let  $\tilde{e}_j$  denote the error in this approximation. It is easy to see that this error is at most  $1/(a_j + 1)$ , *i.e.*

$$0 \leq \underbrace{\frac{\bar{D}}{a_j} - W_j}_{\tilde{e}_j} \leq \frac{1}{a_j + 1}. \quad (4)$$

Let  $e_j$  denote the error incurred when we discretise the colour classes with ideal width  $y_j n$ , *i.e.*  $y_j n = W_j + e_j$ .

Similarly, for the last colour class formed during the  $j$ -th terminating call of Algorithm  $\bar{\alpha}(x, k)$  in the case where balancing is performed, hereafter referred to as a *balancing colour class* (see Figure 11(b)), we have  $z'_j = \lfloor (\bar{D} - s_j \lfloor \bar{D}/a_j \rfloor)/(k_j - 1) \rfloor$  and  $t'_j = \bar{D} - (z'_j s_j + z'_j (k_j - 1))$ . Following the same procedure as for the case above, we obtain the approximation

$$W'_j = \left\lfloor \frac{\bar{D} - s_j \lfloor \bar{D}/a_j \rfloor}{k_j - 1} \right\rfloor + \frac{t'_j}{k_j} = \frac{\left\lfloor \frac{\bar{D} - s_j \lfloor \bar{D}/a_j \rfloor}{k_j - 1} \right\rfloor + \bar{D} - s_j \lfloor \bar{D}/a_j \rfloor}{k_j}$$

for the ideal width  $(\bar{D} - s_j \lfloor \bar{D}/a_j \rfloor)/(k_j - 1)$  of the balancing colour class associated with the degree  $\bar{D} - s_j z_j$ . (Note that we subtract the vertices in the balancing class associated with the width  $y_j n$ .) In this case the error  $\tilde{e}'_j$  involved in the approximation satisfies

$$0 \leq \underbrace{\frac{\bar{D} - s_j \lfloor \bar{D}/a_j \rfloor}{k_j - 1} - W'_j}_{\tilde{e}'_j} \leq \frac{1}{k_j}. \quad (5)$$

Finally, let  $e'_j$  be the error incurred when we discretise the balancing colour class with ideal width  $y'_j n$ , *i.e.*  $y'_j n = W'_j + e'_j$ . Note  $W'_j = e'_j = y'_j = 0$  if there is no balancing during the  $j$ -th terminating call of Algorithm  $\bar{\alpha}(x, k)$ .

The following growth property of the sequence  $a_1, \dots, a_\ell$  computed in (3), for which a proof may be found in [5, Lemma 5], is used in Theorem 2.

**Lemma 1** *The sequence  $(a_j)_{j=1}^\ell$  generated in (3) satisfies the recursive relationship  $a_{j+1} \geq a_j(a_j + 1)$ , for all  $j = 1, \dots, \ell - 1$ .  $\blacksquare$*

The results above may now be used to establish bounds on  $D_x^\Delta(K_{k \times n})$ .

**Theorem 2**  $\lceil \alpha(x, k)n \rceil \leq \bar{D}_x^\Delta(K_{k \times n}) \leq \lceil \bar{\alpha}(x, k)n \rceil + 1$ .

**Proof:** The lower bound follows from the fact that  $\alpha(x, k)$  is directly proportional to the length of the partite sets, *i.e.* if the length of the partite sets is changed by a factor  $n$ , then  $\alpha(x, k)$  also changes by a factor  $n$ .

Let  $\bar{D} = \lceil \bar{\alpha}(x, k)n \rceil + 1$  and let  $W_j$  and  $W'_j$  be defined as before. To prove that all colour classes have degree at most  $\bar{D}$ , it is required to show that  $\sum_{j=1}^\ell (W_j + W'_j) \geq n$ . However,

$\sum_{j=1}^{\ell} (W_j + W'_j) = \sum_{j=1}^{\ell} (y_j n - e_j + y'_j n - e'_j) = n - \sum_{j=1}^{\ell} (e_j + e'_j)$ . So, to complete the proof, it need only be shown that  $\sum_{j=1}^{\ell} (e_j + e'_j) \leq 0$ .

To achieve this goal, it is first shown that  $e_j \leq 1/(a_j + 1) - 1/a_j$ . The difference between the ideal width and the ideal width associated with degree  $\bar{D}$ , is  $y_j n - \bar{D}/a_j = y_j n - (\lceil a_j y_j n \rceil + 1)/a_j \leq y_j n - (a_j y_j n + 1)/a_j = -1/a_j$ . If this inequality is added to (4) the result is  $e_j = y_j n - W_j \leq 1/(a_j + 1) - 1/a_j$ , as desired.

Next, the inequality  $e'_j \leq 1/k_\ell \leq 1/(2(a_\ell + 1))$  is proved. The first inequality follows from (5). The second inequality follows from the fact that at least three colour classes are required when balancing occurs. (If  $x = 2$ , the number of partite sets in each colour class will differ by at most 1). Thus,  $\sum_{j=1}^{\ell} (e_j + e'_j) \leq \sum_{j=1}^{\ell} (1/(a_j + 1) - 1/a_j + 1/(2(a_\ell + 1)))$ . For  $\ell \geq 2$  it is not difficult to verify that the above expression is negative, as desired. (The identity  $1/a_j - 1/(a_j + 1) = 1/(a_j(a_j + 1))$  and Lemma 1 may be used.) More care is needed for the case  $\ell = 1$ . It is sufficient to show that  $W_1 + W'_1 \geq n$  for the case where there is only one balancing terminating call, since without balancing,  $e_1 \leq 0$  and the theorem holds. In the remainder of the proof, all subscripts are omitted for the sake of convenience, since they are all 1. Note that  $W' = (\lfloor (\bar{D} - sn)/(k - s - 1) \rfloor + \bar{D} - sn)/(k - s)$ , because the discretisation of the last  $s$  partite sets is not required. Thus,

$$W + W' = \frac{\lfloor \frac{\bar{D}}{a} \rfloor + \bar{D}}{a + 1} + \frac{\lfloor \frac{\bar{D} - sn}{k - s - 1} \rfloor + \bar{D} - sn}{k - s} \geq n$$

must be shown to hold true. Writing  $k - s$  as  $(x - 1)(a + 1)$  and multiplying with  $(x - 1)(a + 1)$  we obtain (after simplification)

$$\left\lfloor \frac{\bar{D}}{a} \right\rfloor (x - 1) + \left\lfloor \frac{\bar{D} - sn}{k - s - 1} \right\rfloor + \bar{D} x \geq kn. \quad (6)$$

Because all  $x$  colour classes have normalised colour class induced maximum degree  $\alpha(x, k)$  in the normalised  $\Delta(\vec{d}, x)$   $k$ -partite colouring,  $\alpha(x, k)$  may be written as  $\alpha(x, k) = (k - y(x - 1) - y')/x$  or  $yn(x - 1) + y'n + \alpha(x, k)nx = kn$ . Substituting this into (6), we need to prove that

$$\left( \left\lfloor \frac{\bar{D}}{a} \right\rfloor - yn \right) (x - 1) + \left\lfloor \frac{\bar{D} - sn}{k - s - 1} \right\rfloor - y'n + (\bar{D} - \alpha(x, k)n)x \geq 0. \quad (7)$$

As before,  $\bar{D}/a - yn \geq 1/a$ . Therefore,  $\lfloor \bar{D}/a \rfloor - yn > -1$ . It is also known that  $\alpha(x, k) = y'(k - s - 1) + s = ay$ , so that  $((ay - s)n)/(k - s - 1) = y'n$ . This implies that  $(\bar{D} - sn)/(k - s - 1) > y'n$  or  $\lfloor (\bar{D} - sn)/(k - s - 1) \rfloor - y'n > -1$ . Also,  $\bar{D} - \alpha(x, k)n \geq 1$ . Substituting these three inequalities into (7) it follows that

$$\left( \left\lfloor \frac{\bar{D}}{a} \right\rfloor - yn \right) (x - 1) + \left\lfloor \frac{\bar{D} - sn}{k - s - 1} \right\rfloor - y'n + (\bar{D} - \alpha(x, k)n)x \geq (-1)(x - 1) + (-1) + (1)x = 0$$

which completes the proof. ■

The algorithm presented next (which should be thought of as a follow-up to Algorithm  $\bar{\alpha}(x, k)$  of §3), is capable of determining whether  $D_x^\Delta(K_{k \times n}) \leq \lceil \bar{\alpha}(x, k)n \rceil$ , or whether the upper bound in Theorem 2 is required.



**Algorithm**  $\overline{D}_x^\Delta(K_{k \times n})$ 


---

**Input:** The cardinality of each partite set,  $n$ , the upper bound,  $\overline{\alpha}(x, k)$ , the sequences  $k_1, \dots, k_\ell$  and  $s_1, \dots, s_\ell$  determined during the course of Algorithm  $\overline{\alpha}(x, k)$ , as well as the sequences  $a_1, \dots, a_\ell$ ,  $y_1, \dots, y_\ell$  and  $y'_1, \dots, y'_\ell$ .

**Output:** An upper bound on  $D_x^\Delta(K_{k \times n})$ .

---

1.  $\overline{D} \leftarrow \lceil \overline{\alpha} n \rceil$
  2. **if**  $y_j n, y'_j n \in \mathbb{N}$  for all  $j = 1, \dots, \ell$  **then return**  $\overline{D}$ , **stop**
  3. **for**  $j = 1$  **to**  $\ell$  **do**
    - 3.1  $W_j \leftarrow (\lfloor \overline{D}/a_j \rfloor + \overline{D})/(a_j + 1)$
    - 3.2 **if**  $\lfloor \overline{D}/a_j \rfloor = 0$  **then**  $W_j \leftarrow W_j + 1/(a_j + 1)$
    - 3.3 **if**  $0 < s_j < a_j$  **then**
      - 3.3.1  $W'_j \leftarrow (\lfloor (\overline{D} - \lfloor \overline{D}/a_j \rfloor)/(k_j - 1) \rfloor + \overline{D} - s_j \lfloor \overline{D}/a_j \rfloor)/k_j$
      - 3.3.2 **if**  $\lfloor (\overline{D} - \lfloor \overline{D}/a_j \rfloor)/(k_j - 1) \rfloor = 0$  **then**  $W'_j \leftarrow W'_j + 1/k_j$
    - 3.4 **else**  $W'_j \leftarrow 0$
  4. **if**  $\sum_{j=1}^{\ell} (W_j + W'_j) < n$  **then**  $\overline{D} \leftarrow \overline{D} + 1$
  5. **return**  $\overline{D}$
- 

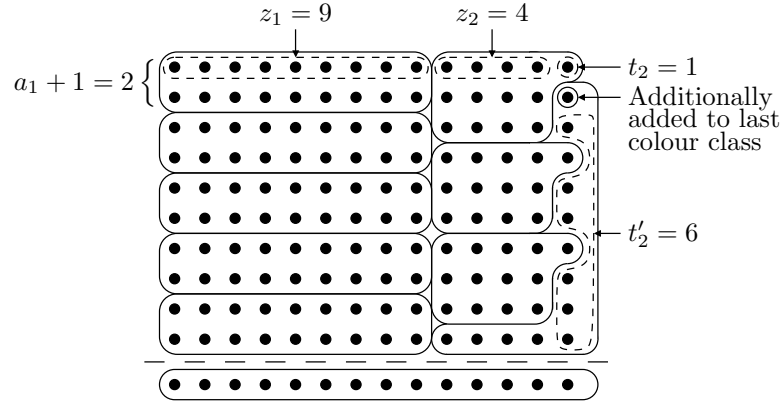
Algorithm  $\overline{D}_x^\Delta(K_{k \times n})$  commences by setting  $\overline{D}$  equal to  $\lceil \overline{\alpha}(x, k)n \rceil$  and then determines the ideal widths  $y_j n$  and  $y'_j n$  for all  $j = 1, \dots, \ell$ . If all these ideal widths are integral, as in Example 4(a), then no discretisation is required and the algorithm terminates, since a good  $\Delta(d, x)$ -colouring of  $K_{k \times n}$  has been achieved. On the other hand, if any of the above ideal widths is not integral, as in Example 4(b), discretisation is required, as is implicitly achieved by computation of the average discretised widths,  $W_j$  and  $W'_j$ , of the colour classes in Step 3 of the algorithm. Note that if  $\lfloor \overline{D}/a_j \rfloor = 0$ , then  $t_j$  should be incremented by one more in order to obtain the same maximum degree  $\overline{D}$  as that of the other colour classes, because in this case each of the  $t_j$  vertices is adjacent to  $\overline{D} - 1$  vertices in the same colour class. Similar arguments hold for  $t'_j$ . These are reflected in Steps 3.2 and 3.3.2 of the algorithm. If the sum of all the average discretised widths,  $\sum_{j=1}^{\ell} (W_j + W'_j)$ , is less than  $n$ , then all vertices cannot be coloured in such a way that the maximum degrees of all the colour class induced subgraphs are smaller than or equal to  $\lceil \overline{\alpha}(x, k)n \rceil$ . However, in this case it follows from Theorem 2 that all vertices may indeed be coloured such that all the colour class induced maximum degrees are smaller than or equal to  $\lceil \overline{\alpha}(x, k)n \rceil + 1$ .

Step 3 of Algorithm  $\overline{D}_x^\Delta(K_{k \times n})$  is performed a total of  $\ell$  times. Therefore Algorithm  $\overline{D}_x^\Delta(K_{k \times n})$  computes at most  $2\ell$  average discretised widths  $W_j$  and  $W'_j$ . Hence, Algorithm  $\overline{D}_x^\Delta(K_{k \times n})$  also has an  $\mathcal{O}(\log \log x)$  worst-case time complexity, similar to that of Algorithm  $\overline{\alpha}(x, k)$ .

**Example 5 (Continuation of Example 4(b))**

Recall that in Example 4(b) we sought to colour the graph  $K_{11 \times 14}$  with 10 colours. The values of the variables determined before were  $\overline{\alpha}(10, 11) = 9/14$ ,  $k_1 = 10$ ,  $k_2 = 10$ ,  $s_1 = 0$ ,  $s_2 = 1$ ,  $a_1 = 1$ ,  $a_2 = 2$ ,  $\overline{\alpha}(10, 11)n = 9$ ,  $y_1 n = 9$ ,  $y_2 n = 9/2$  and  $y'_2 n = 1/2$ . Since some of the  $y_j n$  and  $y'_j n$  values are not integral, the discretisation in Step 3 of Algorithm  $\overline{D}_x^\Delta(K_{k \times n})$

must be performed. The values of the variables in Step 3 of Algorithm  $\overline{D}_x^\Delta(K_{k \times n})$  are determined as  $W_1 = (9 + 9)/2 = 9$ ,  $W'_1 = 0$ ,  $W_2 = (4 + 9)/3 = 4.33$  and  $W'_2 = (0 + 9 - 4)/10 = 0.5$ . Since  $\sum_{j=1}^2 (W_j + W'_j) = 9 + 4.33 + 0.5 = 13.83 < 14$ , Algorithm  $\overline{D}_x^\Delta(K_{k \times n})$  terminates with  $\overline{D}_{10}^\Delta(K_{11 \times 14}) = 10$ . The colour classes of a good  $\Delta(10, 10)$ -colouring of  $K_{11 \times 14}$  are shown graphically in Figure 12.  $\square$



**Figure 12:** A graphical representation of the discretisation approach achieved in Algorithm  $\overline{D}_x^\Delta(K_{k \times n})$  for the graph  $K_{11 \times 14}$  considered in Examples 4(b) and 5. The corresponding ideal  $\Delta(9, 10)$ -colouring was represented graphically in Figure 10.

## 5 Concluding remarks

In this paper we determined a good upper bound on the value of the parameter  $D_x^\Delta(K_{k \times n})$  for the complete balanced multipartite graph  $K_{k \times n}$ . In doing so, we obtained a good  $\Delta(d, x)$ -colouring strategy for  $K_{k \times n}$ , and hence a good approximation to the true value of the  $\Delta(d)$ -chromatic number,  $\chi_d^\Delta(K_{k \times n})$ , for the class of complete balanced multipartite graphs. These bounds find application in the problem of scheduling grouped access to some shared resource in a way that (a) for a specific maximum toleration of conflict between users in the same group, the number of groups are minimised, or (b) for a fix number of groups, the threshold of conflict between users of the same group is minimised. For  $d = 0$  or  $d = 1$  (and  $k, n \geq 2$ ),  $\chi_d^\Delta(K_{k \times n}) = k$ . Values of the  $\Delta(d)$ -chromatic number of  $K_{k \times n}$  for other small values of  $d$  ( $2 \leq d \leq 7$ ), where  $3 \leq k \leq 10$  and  $n \geq 2$ , are presented in Table 2.

Finally, the work reported in this paper forms part of an open graph colouring problem, namely the characterisation of  $\Delta$ -chromatic sequences for graphs, where the  $\Delta$ -chromatic sequence of a graph  $G$  consists of the values of  $\chi_d^\Delta(G)$  as  $d$  increases from zero. Frick & Henning [11] provided necessary conditions for a sequence of positive integers to be the  $\Delta$ -chromatic sequence of some graph  $G$ , but no sufficient conditions are known to the present authors. It transpires that an ability to determine the value of  $\chi_d^\Delta(K_{k \times n})$  is expected to play a central role in establishing a sufficient condition for a sequence of positive integers to be the  $\Delta$ -chromatic sequence of a graph, as explained in [4, 18].

		n		
		2	3	4
k	3	2	3	3
	4	2	4	4
	5	3	5	5
	6	3	5	6
	7	4	6	7
	8	4	7	8
	9	5	8	9
	10	5	9	10

(a) Values of  $\chi_2^\Delta(K_{k \times n})$   
( $3 \leq k \leq 10$ ;  $n \geq 2$ )

		n						
		2	3	4	5	6	7	8
k	3	1	2	2	2	3	3	3
	4	2	2	2	2	3	4	4
	5	2	3	3	3	4	5	5
	6	2	3	3	3	4	5	6
	7	3	4	4	4	5	6	7
	8	3	4	4	4	6	7	8
	9	3	4	5	5	7	8	9
	10	4	5	5	5	7	9	10

(d) Values of  $\chi_5^\Delta(K_{k \times n})$   
( $3 \leq k \leq 10$ ;  $n \geq 2$ )

		n			
		2	3	4	5
k	3	2	2	3	3
	4	2	2	3	4
	5	3	3	4	5
	6	3	3	5	6
	7	4	4	6	7
	8	4	4	6	8
	9	5	5	7	9
	10	5	5	8	10

(b) Values of  $\chi_3^\Delta(K_{k \times n})$   
( $3 \leq k \leq 10$ ;  $n \geq 2$ )

		n								
		2	3	4	5	6	7	8	9	10
k	3	1	1	2	2	2	3	3	3	3
	4	1	2	2	2	2	3	3	4	4
	5	2	2	3	3	3	4	4	5	5
	6	2	2	3	3	3	4	5	5	6
	7	2	3	4	4	4	5	6	6	7
	8	2	3	4	4	4	6	6	7	8
	9	3	3	5	5	5	7	7	8	9
	10	3	4	5	5	5	7	8	9	10

(e) Values of  $\chi_6^\Delta(K_{k \times n})$   
( $3 \leq k \leq 10$ ;  $n \geq 2$ )

		n					
		2	3	4	5	6	7
k	3	1	2	2	3	3	3
	4	2	2	2	3	4	4
	5	2	3	3	4	5	5
	6	2	3	3	5	5	6
	7	3	4	4	6	6	7
	8	3	4	4	6	7	8
	9	3	5	5	7	8	9
	10	4	5	5	7	9	10

(c) Values of  $\chi_4^\Delta(K_{k \times n})$   
( $3 \leq k \leq 10$ ;  $n \geq 2$ )

		n									
		2	3	4	5	6	7	8	9	10	11
k	3	1	1	2	2	2	2	3	3	3	3
	4	1	2	2	2	2	2	3	3	4	4
	5	2	2	3	3	3	3	4	4	5	5
	6	2	2	3	3	3	3	4	5	5	6
	7	2	3	3	4	4	4	5	6	6	7
	8	2	3	4	4	4	4	5	6	7	8
	9	3	3	4	5	5	5	6	7	8	9
	10	3	4	5	5	5	5	7	8	9	10

(f) Values of  $\chi_7^\Delta(K_{k \times n})$   
( $3 \leq k \leq 10$ ;  $n \geq 2$ )

**Table 2:** Values of  $\chi_d^\Delta(K_{k \times n})$ ,  $3 \leq k \leq 10$ ,  $n \geq 2$  for different values of  $d$ . Note that once  $\chi_d^\Delta(K_{k \times n}) = k$  for some  $n = n_0$  then  $\chi_d^\Delta(K_{k \times n}) = k$  for all  $n \geq n_0$ .

## Acknowledgement

The authors are indebted to Dr Werner Gründlingh, who produced the graphics in this paper. Work towards this paper was supported financially by Research Subcommittee B of the University of Stellenbosch in the form of a post-doctoral fellowship for the first author and by the South African National Research Foundation in the form of a bursary for the second author's PhD study and a research grant (GUN 2072999) awarded to the third author.

## References

- [1] ANDREWS JA & JACOBSON MS, 1985, *On a generalization of the chromatic number*, *Congressus Numerantium*, **47**, pp. 33–48.
- [2] BRELAZ D, 1979, *New methods to colour the vertices of a graph*, *Communications of the Association for Computing Machinery*, **22**, pp. 251–256.
- [3] BROOKS RL, 1941, *On colouring the nodes of a network*, *Proceedings of the Cambridge Philosophical Society*, **37**, pp. 194–197.
- [4] BURGER AP & GROBLER PJP, 1993, *Aspects of  $(m,k)$ -colourings*, *Research Report 146/93(7)*, University of South Africa, Pretoria, pp. 40–49.
- [5] BURGER AP & VAN VUUREN JH, *On the placement of a number of strings in a collection of hats*, submitted.
- [6] CHARTRAND G & OELLERMANN O, 1993, *Applied and algorithmic graph theory*, McGraw-Hill, New York (NY).
- [7] COWEN L, COWEN R & WOODALL D, 1986, *Defective colourings of graphs in surfaces: Partitions into subgraphs of bounded valence*, *Journal of Graph Theory*, **10**, pp. 187–195.
- [8] COWEN L, GODDARD W & JESURUM CE, 1997, *Defective colouring revisited*, *Journal of Graph Theory*, **24**, pp. 205–219.
- [9] FOULDS LR, 1992, *Graph theory applications*, Springer–Verlag, Inc., New York (NY).
- [10] FRICK M, 1993, *A survey of  $(m,k)$ -colourings, quo vadis, graph theory?*, *Annals of Discrete Mathematics*, **55**, pp. 45–58.
- [11] FRICK M & HENNING MA, 1994, *Extremal results on defective colourings of graphs*, *Discrete Mathematics*, **126**, pp. 151–158.
- [12] GROSS JL & YELLEN J, 1999, *Graph theory and its applications*, CRC Press, Boca Raton (FL).
- [13] HARARY F, HEDETNIEMI ST & ROBINSON RW, 1969, *Uniquely colourable graphs*, *Journal of Combinatorial Theory*, **6**, pp. 264–270.
- [14] HARARY F & JONES K, 1985, *Conditional colourability II: Bipartite variations*, *Congressus Numerantium* **50**, pp. 205–218.
- [15] HOWE D, 2004, *The free on-line dictionary of computing*, [Online], [cited 2005, May 28], Available from: <http://foldoc.doc.ic.ac.uk/foldoc/contents.html>
- [16] JENSEN TR & TOFT B, 1995, *Graph coloring problems*, John Wiley & Sons, Inc., New York (NY).
- [17] LOVÁSZ L, 1966, *On decompositions of graphs*, *Studia Scientiarum Mathematicarum Hungarica*, **1**, pp. 237–238.

- [18] NIEUWOUDT I, 2007, *On the maximum degree chromatic number of a graph*, PhD Dissertation, University of Stellenbosch, Stellenbosch.
- [19] SZEKERES G & WILF HS, 1968, *An inequality for the chromatic number of a graph*, Journal of Combinatorial Theory, **4**, pp. 1–3.
- [20] TRICK MA, 1994, *Network resources for colouring a graph*, [Online], [cited 2003, May 28], Available from: <http://mat.gsia.cmu.edu/COLOR/color.html>

