Cluttered Orderings for the Complete Bipartite Graph

Meinard Müller [a] , Tomoko Adachi [b] , and Masakazu Jimbo [c]

[a] *University of Bonn, Department of Computer Science, Römerstr. 164, 53113 Bonn, Germany, meinard@cs.uni-bonn.de*

[b] *Toho University, Department of Information Sciences, 2-2-1 Miyama, Funabashi, Chiba 274-8510, Japan, adachi@is.sci.toho-u.ac.jp*

[c] *Graduate School of Information Science, Nagoya University, Furo-cho, Chikusa-ku, 464-8601, Japan, jimbo@is.nagoya-u.ac.jp*

Abstract

To minimize the access cost in large disk arrays (RAID) Cohen et al. [5–7] introduced (d, f)-cluttered orderings of various set systems, $d, f \in \mathbb{N}$. In case of a graph this amounts to an ordering of the edge set such that the number of points contained in any d consecutive edges is bounded by the number f. For the complete graph, Cohen et al. gave some optimal solution for small parameters d [5] and introduced some general construction principle based on wrapped Δ-labellings [7]. In this paper, we investigate cluttered orderings for the complete bipartite graph. We adapt the concept of a wrapped Δ-labelling to the bipartite case and introduce the notion of a (d, f)-movement for subgraphs. From this we get a general existence theorem for cluttered orderings. The main result of this paper is the explicit construction of several infinite families of wrapped Δ-labellings leading to cluttered orderings for the corresponding bipartite graphs.

Key words: cluttered ordering, wrapped Δ-labelling, bipartite graph, RAID
PACS:

1 Introduction

We begin by introducing the concept of a (d, f)-cluttered ordering for arbitrary set systems generalizing the definition of [7]. A *set system* is a pair $\mathcal{S} = (X, \mathcal{B})$ consisting of a finite set X and a finite set $\mathcal{B} = \{B_0, B_1, \ldots B_{n-1}\}$, $n \in \mathbb{N}$, of subsets of X. The elements of X are called *points* and the ones of $\mathcal{B}$ *blocks*. Furthermore, let d be a positive integer, $1 \leq d \leq n$, called a *window* of $\mathcal{S}$, and let π be a permutation on the index set $\{0, 1, \cdots, n-1\}$, called a *block* *ordering* of $\mathcal{S}$. Define $X_i^{\pi,d} := \bigcup_{j=0}^{d-1} B_{\pi(i+j)}$ for $0 \leq i \leq n-1$, where indices are considered modulo n. For some given $f \in \mathbb{N}$, π is called a *cyclic* (d, f)*-cluttered ordering*, or simply (d, f)*-cluttered ordering*, if $\max_i |X_i^{\pi,d}| \leq f$.

In view of the RAID application described in the next section one is interested in minimizing the parameter f for some given d. Let $f_{\min}(\mathcal{S}, d)$ denote the minimum value of f over all block orderings π of $\mathcal{S}$ for some fixed d and, similarly, $d_{\max}(\mathcal{S}, f)$ the maximum value of d for some fixed f. Typical questions are, for example, to find lower and upper bound for $f_{\min}(\mathcal{S}, d)$ and $d_{\max}(\mathcal{S}, f)$. Furthermore, one is interested in explicit constructions of such orderings for certain subclasses of set systems.

Cluttered orderings were introduced and studied by Cohen et al. for Steiner triple systems [5] and the complete graph [6,7]. In case of the complete graph, Cohen et al. [6] constructed for the parameters $d = 3$ and $d = 4$ optimal cluttered orderings, in this case also denoted as *ladder ordering*, for all complete graphs K_n, $n \in \mathbb{N}$, except for the parameters $n \in \{15, 18, 22\}$. Furthermore, they also introduced in [7] some general construction principle of cluttered orderings for the complete graph based on wrapped ρ-labellings. However, in this construction they presuppose the existence of such wrapped ρ-labellings and give only a finite number of explicit examples computed by some backtracking algorithm. Furthermore, injectivity of the ρ-labelling is not needed in their construction. Dropping this condition we call the resulting labelling a Δ-labelling as will be defined later.

The rest of this paper is organized as follows. In Section 3, we adapt the notion of a Δ-labelling to the bipartite case, introduce the notion of a (d, f)-movement of certain subgraphs, and derive a general existence theorem for cluttered orderings of the complete bipartite graph. The main contribution of this paper can be found in Section 4 where we apply this general existence theorem to obtain several infinite families of cluttered orderings. First, we introduce, similar to [7] in the complete case, a class of bipartite graphs which allow suitable (d, f)-movements. To achieve the necessary consistency condition of such movements we adapt the concept of wrapped Δ-labellings to the bipartite case. Furthermore, we also give three explicit constructions of different infinite families of such wrapped Δ-labellings leading to cluttered orderings for the corresponding complete bipartite graphs. Among others, we get optimal cluttered orderings for $d = 3$, $d = 4$, $d = 5$, and $d = 6$ for all complete bipartite graphs whose number of vertices is divisible by three. In Section 5, we give a short summary of the results and conclude with some final remarks on lower bounds. As motivation for the combinatorial problems discussed in this paper we summarize the main ideas of the underlying RAID application in Section 2, where one also finds further links to the literature. However, this section may be skipped since the result are not needed in the rest of this paper.

2 RAID Application

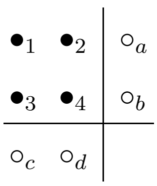
The desire to speed up secondary storage systems has led to the development of *disk arrays* which achieve performance through disk parallelism. While performance improves with increasing numbers of disks the chance of data loss coming from *catastrophic failures*, such as head crashes and failures of the disk controller electronics, also increases. To avoid high rates of data loss in large disk arrays one includes redundant information stored on additional disks — also called *check disks* — which allows the reconstruction of the original data — stored on the so-called *information disks* — even in the presence of disk failures. These disk array architectures are known as *redundant arrays of independent disks* (RAID) [4].

Hellerstein et al. [8] introduced data redundancy in form of erasure-resilient codes. Let $n, c \in \mathbb{N}$ and let $\mathrm{GF}(2)$ denote the field with two elements. Then an erasure-resilient code is defined by a linear injection $\gamma : \mathrm{GF}(2)^n \to \mathrm{GF}(2)^{n+c}$ such that an information $u \in \mathrm{GF}(2)^n$ appears unchanged in the first n bits – the so-called *information bits* – of the corresponding code vector $\gamma(u)$. The remaining c bits are referred to as *check bits* which can be computed as the parity of subsets of information bits. Each such code can be defined in terms of a $c \times (n + c)$*-parity check matrix*, $H = [C|I]$, where I denotes the $c \times c$ identity matrix and C is a $c \times n$ matrix. The codewords in the code are the vectors $v \in \mathrm{GF}(2)^{n+c}$ satisfying the equation $Hv = 0$. Note, that the first n columns of H correspond to the information bits and the last c columns to the check bits (see also Fig. 1).

An unreadable bit of a code vector is called an *erasure*. It is a well known fact that a code can correct a set of t erasures iff the corresponding t columns of H are linearly independent considered as vectors over $\mathrm{GF}(2)$. An erasure-resilient code which can correct any t erasures will be abbreviated as t-ERC. In view of the RAID-application there are the following two important metrics in ERCs. One metric is the *update penalty*, which is the number of check disks whose content must be changed if an information disk is changed. In terms of the matrix H it can be defined as the maximum over the weights of the columns of H. It follows easily that the update penalty of a t-ERC is at least t. Another metric is the *check bit overhead*, which is the ratio c/n of the number of check bits to information bits. Good erasure codes have high erasure correcting capabilities, whereas the update penalty as well as the check bit overhead is low. (See [8] for further details.)

From a set theoretic point of view, one can model the problem as follows. Let $\mathcal{S} = (X, \mathcal{B})$ be a set system as in the introduction and set $c := |X|$. The elements of X are identified with the check disks and the elements of $\mathcal{B}$ with the information disks. The bits of some check disk $x \in X$ are computed as the parity of the corresponding bits of the check disks $B \in \mathcal{B}$ with $x \in B$. In other words, the incidence matrix of the set system coincides with the matrix C of the parity check matrix $H = [C|I]$ of the corresponding code. For a detailed treatment of the connection between ERCs and combinatorial design theory we refer to the nice paper by Chee et al. [3].

In particular, we are interested in the two-dimensional parity code which is defined as follows. Let $n = \ell^2$ for some $\ell \in \mathbb{N}$, so that the n information bits can be arranged in a two dimensional array. Associate to each row and each column a check bit containing the parity of that row or column, i.e., $c = 2\ell$. The so defined code is called 2-*dimensional parity code* which is a 2-ERC with update penalty 2 and check bit overhead $\frac{c}{n} = \frac{4}{c}$. It is obvious that any 2-ERC with update penalty 2 cannot correct any 3-erasure consisting of an information bit and its two associated check bits. Such erasures are also called *bad* 3-*erasure*. In [8] it is shown that the 2-dimensional parity code can correct all 3-erasures except for the bad 3-erasures and has – with respect to this even higher erasure-correcting capability – optimal check disk overhead among all such codes. Fig. 1 shows the case $\ell = 2$. For example, information disk 1 is associated to the check disks a and c.



$$\begin{array}{c} \begin{array}{cccccccc} 1 & 2 & 3 & 4 & a & b & c & d \end{array} \\ H = \left[\begin{array}{cccc|cccc} 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 0 & 0 & 0 & 1 \end{array}\right] \begin{array}{c} a \\ b \\ c \\ d \end{array} \end{array}$$

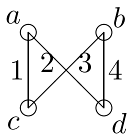


Fig. 1. 2-dimensional parity code for $\ell = 2$, its parity check matrix H, and the corresponding complete bipartite graph $K_{2,2}$.

In any erasure-resilient code with update penalty 2, where each information bit is associated with exactly two check bits, the corresponding set system of such a code is a graph: the check bits correspond to the vertices and the information bits to the edges. Now, it is not hard to see that the 2-dimensional parity code with $n = \ell^2$ information bits corresponds to the complete bipartite graph $K_{\ell,\ell}$ with 2ℓ vertices and ℓ^2 edges (see also Fig. 1). This explains our interest in the complete bipartite graphs in this paper.

In a RAID system disk writes are expensive operations and should therefore be minimized. When writing on a single information disk one also has to recompute the parity information and change the contents of all check disks involved. This overhead is expressed by the update penalty. In many applications there are writes on a small fraction of consecutive disks – say d disks – where d is small in comparison to n, the number of information disks. In this case a write can be implemented as an efficient *read-modify-write* which is described by Cohen et al. [5] as follows. Firstly, the d information disks are read followed by all of their associated check disks. In the case when check disks overlap the physical read only takes place once. Secondly, all of the new parity is computed and then this new parity and the new information is written back to the disks. Once again, the shared check disks are only physically written once. Therefore, to minimize the number of operations when writing to d consecutive information disks one has to minimize the number of check disks – say f – associated to the d information disks. In other words, the order of the information disks – or the order of the corresponding columns of the parity check matrix – plays a crucial role for the efficiency of the RAID system. In terms of set systems this leads exactly to the definition of a (d, f)-cluttered ordering as given in the introduction.

3 A General Existence Theorem for Cluttered Orderings

Let $\ell \in \mathbb{N}$ be a natural number and let $K_{\ell,\ell}$ denote the complete bipartite graph with 2ℓ vertices and ℓ^2 edges. In the following, we identify the vertex set of $K_{\ell,\ell}$ with $\mathbb{Z}_\ell \times \mathbb{Z}_2$ where two vertices are connected by an edge iff they have different second components in $\mathbb{Z}_\ell \times \mathbb{Z}_2$. The construction of (d, f)-cluttered orderings for $K_{\ell,\ell}$ with small $f \in \mathbb{N}$ is based on two fundamental concepts: Δ-*labellings* and (d, f)*-movements*. A suitable Δ-labelling is used to decompose $K_{\ell,\ell}$ into isomorphic copies of some subgraph. Then, based on a suitable (d, f)-movement, "local" edge orderings can be defined on each of these copies. When the (d, f)-movement in question is *consistent* with some translation operator the locally defined edge orderings can be composed to form a "globally" defined cluttered (d, f)-ordering for $K_{\ell,\ell}$. This construction principle, which was implicitly used in [7] in case of the complete graph, leads to some general existence theorem (Theorem 3.4).

In the following, $H = (U, E)$ denotes a bipartite graph with vertex set U partitioned into two subsets denoted by V and W. Any edge of the edge set E contains exactly one point of V and W, respectively. Let $\ell = |E|$, then a *difference labelleling* or, for short, a Δ-*labelling* of H with respect to V and W is defined to be a map $\delta : U \to \mathbb{Z}_\ell \times \mathbb{Z}_2$ with $\delta(V) \subset \mathbb{Z}_\ell \times \{0\}$ and $\delta(W) \subset \mathbb{Z}_\ell \times \{1\}$ such that each element of $\mathbb{Z}_\ell$ occurs exactly once in the difference list

$$\Delta(E) := \Big(\pi_1(\delta(v) - \delta(w))\Big| v \in V, w \in W, \{v, w\} \in E\Big). \qquad (1)$$

Here, $\pi_1 : \mathbb{Z}_\ell \times \mathbb{Z}_2 \to \mathbb{Z}_\ell$ denotes the projection onto the first component. Let H^δ denote the graph obtained from H by identifying the vertices which have the same image under δ. Of course, $H = H^\delta$ where δ is injective. However, we do not presume a Δ-labelling to be an injection. In any case, Condition

(1) implies that H^δ does not have multiple edges. In general, vertex labellings are well-known tools for the decomposition of graphs into subgraphs (see, e.g., [2]). In this context a decomposition is understood to be a partition of the edge set of the graph. In the case of the complete bipartite graph one has the following proposition.

Proposition 3.1 *Let $H = (U, E)$ be a bipartite graph, $\ell = |E|$, and δ a Δ-labelling of H. Then there is a decomposition of the complete bipartite graph $K_{\ell,\ell}$ into isomorphic copies of H^δ.*

Proof: Take ℓ copies of $H = (U, E)$, where each object of the ith copy is tagged by the superscript (i), e.g., $H^{(i)} = (U^{(i)}, E^{(i)})$, $i \in \mathbb{Z}_\ell$. Provide the ith graph $H^{(i)}$ with the labelling $\delta^{(i)}$ defined by $\delta^{(i)}(u^{(i)}) := \delta(u) + (i, 0)$, $u \in U$. Now, take the disjoint union of the graphs $H^{(i)}$, $i \in \mathbb{Z}_\ell$, and identify the vertices with the same label. In other words, $x \in U^{(i)}$ is identified with $y \in U^{(j)}$ iff $\delta^{(i)}(x) = \delta^{(j)}(y)$. It is not difficult to show that the resulting graph is isomorphic to the complete bipartite graph $K_{\ell,\ell}$ with vertex set $\mathbb{Z}_\ell \times \mathbb{Z}_2$. Here the vertices of $K_{\ell,\ell}$ correspond to the labels, i.e., a vertex $u \in U^{(i)}$ is identified according to its label $\delta^{(i)}(u)$ with the corresponding vertex in the vertex set $\mathbb{Z}_\ell \times \mathbb{Z}_2$ of $K_{\ell,\ell}$. One then just checks that each of the ℓ^2 edges of $K_{\ell,\ell}$ — having one vertex in $\mathbb{Z}_\ell \times \{0\}$ and the other in $\mathbb{Z}_\ell \times \{1\}$ — appears as an edge in some $H^{(i)}$ with the corresponding labels of the vertices. $\square$

For example, Fig. 3 in the next section shows a Δ-labelling δ of a graph $H = H(1; 5)$ with 15 edges leading to a decomposition of $K_{15,15}$ into isomorphic copies of H^δ. Next, we define the concept of a (d, f)-movement for arbitrary set systems.

Definition 3.2 Let $\mathcal{S} = (X, \mathcal{B})$ be a set system with finite point set X and block set $\mathcal{B} = \{B_0, B_1, \ldots B_{n-1}\}$, $n \in \mathbb{N}$. Furthermore, let $\Sigma_0, \Sigma_1 \subset \mathcal{B}$ with $d := |\Sigma_0| = |\Sigma_1| > 0$. For a permutation σ on $\{0, 1, \cdots, n-1\}$ define $X_i^{\sigma,d} := \bigcup_{j=0}^{d-1} B_{\sigma(i+j)}$ for $0 \le i \le n-d$ where indices are considered modulo n. Then, for some given $f \in \mathbb{N}$, σ is called a *(d, f)-movement from Σ_0 to Σ_1* if

$$\Sigma_0 = X_0^{\sigma,d}, \quad \Sigma_1 = X_{n-d}^{\sigma,d}, \quad \text{and} \quad \max_{0 \le i \le n-d} |X_i^{\sigma,d}| \le f. \tag{2}$$

Furthermore, let $\varphi : \Sigma_0 \to \Sigma_1$ be any bijection, then a (d, f)-movement σ from Σ_0 to Σ_1 is called *consistent* with φ if

$$\varphi(B_{\sigma(j)}) = B_{\sigma(n-d+j)}, \quad \text{for } j = 0, 1, \ldots, d-1. \tag{3}$$

With these definitions at hand it is not difficult to prove a general existence theorem for (d, f)-cluttered orderings of the complete bipartite graph. As before, let $H = (U, E)$ denote a bipartite graph, δ a Δ-labelling of H,

$\Sigma_0 \subset E$ a subset of order $d = |\Sigma_0|$, and $\ell = |E|$. By Proposition 3.1 we get a decomposition of $K_{\ell,\ell}$ into ℓ isomorphic copies of H^δ. We denote these copies by $H^{(i)} = (U^{(i)}, E^{(i)})$ and the corresponding subsets by ${\Sigma_0}^{(i)} \subset E^{(i)}$, $i \in \mathbb{Z}_\ell$. (Note that in the proof of Proposition 3.1 the $H^{(i)}$ denoted copies of H whereas we now regard these graphs as subgraphs of $K_{\ell,\ell}$ being isomorphic to H^δ.) For each $j \in \mathbb{Z}_\ell$ one obtains an automorphism τ_j of the bipartite graph $K_{\ell,\ell}$ via a cyclic translation of the vertex set:

$$\tau_j : \mathbb{Z}_\ell \times \mathbb{Z}_2 \to \mathbb{Z}_\ell \times \mathbb{Z}_2, \quad \tau_j((u,b)) := (u+j, b) \tag{4}$$

for $(u,b) \in \mathbb{Z}_\ell \times \mathbb{Z}_2$. τ_j induces in a natural way an automorphism of the edge set of $K_{\ell,\ell}$ also denoted by τ_j. Clearly, $\tau_j(E^{(i)}) = E^{(i+j)}$ and $\tau_j({\Sigma_0}^{(i)}) = {\Sigma_0}^{(i+j)}$, $i \in \mathbb{Z}_\ell$. Let κ be an integer coprime to ℓ which will be referred to as *translation parameter*. We define a subgraph $G^{(0)} \subset K_{\ell,\ell}$ by specifying its edge set $E(G^{(0)}) := E^{(0)} \cup {\Sigma_0}^{(\kappa)}$. The cyclic translation τ_κ induces in a canonical way a bijection ${\Sigma_0}^{(0)} \to {\Sigma_0}^{(\kappa)}$ which will be denoted by $\varphi_\kappa^{(0)}$.

Definition 3.3 With above notation, a (d,f)-movement of $G^{(0)}$ from ${\Sigma_0}^{(0)}$ to ${\Sigma_0}^{(\kappa)}$ consistent with $\varphi_\kappa^{(0)}$ will be called a (d,f)*-movement from* ${\Sigma_0}^{(0)}$ *consistent w.r.t. the translation parameter* κ.

Let σ be such a (d,f)-movement of $G^{(0)}$ from ${\Sigma_0}^{(0)}$ consistent w.r.t. κ. Then σ defines an edge ordering of $G^{(0)}$ such that the first d edges are the edges of ${\Sigma_0}^{(0)}$, the last d edges are the edges of ${\Sigma_0}^{(\kappa)}$, and the first ℓ edges are the edges of $E^{(0)}$. By means of the translation operator τ_κ the edge ordering of $G^{(0)}$ can be transferred to an edge ordering of $G^{(\kappa)} := \tau_\kappa(G^{(0)})$. Obviously, $E(G^{(0)}) \cap E(G^{(\kappa)}) = {\Sigma_0}^{(\kappa)}$ which are just the last d edges of $G^{(0)}$ and the first d edges of $G^{(\kappa)}$. Since σ is consistent w.r.t. κ, the last d edges of $G^{(0)}$ are ordered in the same way as the first d edges of $G^{(\kappa)}$. In the same way, one can now order the edges of $G^{(2\kappa)}$, $G^{(3\kappa)}$, and so on. These orderings will be referred to as *local orderings*. Altogether, we define a *global cyclic ordering* on the edge set of $K_{\ell,\ell}$ by first cyclically arranging the edge sets $E^{(i)}$, $i \in \mathbb{Z}_\ell$, of the decomposition along

$$E^{(0)}, E^{(\kappa)}, E^{(2\kappa)}, \ldots, E^{((\ell-1)\kappa)}$$

and then ordering the edges within each subgraph according to σ. Note that in this list all edge sets $E^{(i)}$, $i \in \mathbb{Z}_\ell$, appear exactly once since ℓ and κ are coprime. Furthermore, any d consecutive edges w.r.t. this global cyclic ordering lie in some $G^{(i)}$ and are also consecutive w.r.t. to the corresponding local ordering. Since the global ordering restricted to $G^{(i)}$ coincides with the local ordering defined by σ, the global cyclic ordering defines indeed a (d,f)-cluttered ordering of $K_{\ell,\ell}$. We summarize the result in the following theorem.

Theorem 3.4 *Let* $H = (U,E)$ *be a bipartite graph with* Δ*-labelling* δ *and*

$\kappa \in \mathbb{N}$ a translation parameter coprime to $\ell := |E|$. Let $\Sigma_0 \subset E$ with $d := |\Sigma_0|$ and $f \in \mathbb{N}$. If there is a (d, f)-movement from Σ_0 consistent w.r.t. κ, then there also is a (d, f)-cluttered ordering for the complete bipartite graph $K_{\ell,\ell}$.

As shown in Proposition 3.1 a Δ-labelling δ of some bipartite graph H with ℓ edges leads to a decomposition of $K_{\ell,\ell}$ into isomorphic copies of H^δ. However, in general one has no control of how these copies are embedded in $K_{\ell,\ell}$. Therefore, it is difficult to find good (d, f)-movements which are consistent with some translation operator κ. In the next section, we will deal with this problem.

4 Explicit Constructions

In this section we define an infinite family of bipartite graphs which allow (d, f)-movements with small f. In order to ensure that these (d, f)-movements are consistent with some translation parameter κ, we impose an additional condition on the Δ-labellings also referred to as *wrapped-condition*. Then, we construct three infinite families of such wrapped Δ-labellings. By applying Theorem 3.4 we get explicit (d, f)-cluttered orderings of the corresponding bipartite graphs.

4.1 Construction of Consistent (d, f)-Movements of $H(h; t)$

For each parameter $h \in \mathbb{N}$ and $t \in \mathbb{N}$ we define a bipartite graph denoted by $H(h; t) = (U, E)$. Its vertex set U is partitioned into $U = V \cup W$ and consists of the following $2h(t + 1)$ vertices:

$$\begin{aligned} V &:= \{v_i \mid 0 \le i < h(t+1)\}, \\ W &:= \{w_i \mid 0 \le i < h(t+1)\}. \end{aligned} \tag{5}$$

The edge set E is partitioned into subsets E_s, $0 \le s < t$, defined by

$$\begin{aligned} E'_s &:= \{\{v_i, w_j\} \mid s \cdot h \le i, j < s \cdot h + h\}, \\ E''_s &:= \{\{v_i, w_{h+j}\} \mid s \cdot h \le j \le i < s \cdot h + h\}, \\ E'''_s &:= \{\{v_{h+i}, w_j\} \mid s \cdot h \le i \le j < s \cdot h + h\}, \\ E_s &:= E'_s \cup E''_s \cup E'''_s, \quad \text{for } 0 \le s < t, \\ E &:= \bigcup_{s=0}^{t-1} E_s. \end{aligned} \tag{6}$$

Fig. 2 shows the edge partition of $H(2;1)$. Further examples can be found in Fig. 3, Fig. 4, and Fig. 5.

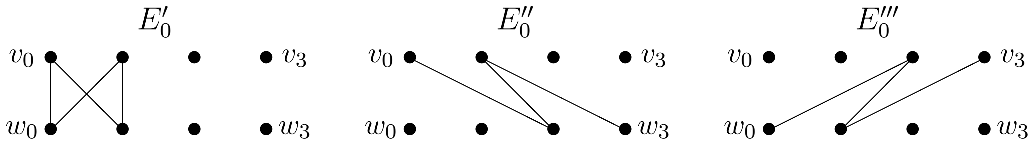


Fig. 2. Partition of the edge set of $H(2;1)$.

For the number $|E|$ of edges one obtains the equation $|E| = t \cdot (h^2 + \frac{h(h+1)}{2} + \frac{h(h+1)}{2}) = th(2h+1)$. The t subgraphs defined by the edge sets E_s, $0 \le s < t$, and its respective underlying vertex sets are isomorphic to $H(h;1)$. Intuitively speaking, the bipartite graph $H(h;t)$ consists of t "consecutive" copies of $H(h;1)$ where the last h vertices of V and W, respectively, of one copy are identified with the first h vertices of V and W, respectively, of the next copy. Traversing these copies with increasing s will define a (d,f)-movement of $H(h;t)$ with small parameter f as is shown in the next proposition.

Proposition 4.1 *There is a (d,f)-movement of $H(h;t)$ from E_0 to E_{t-1} with $d = h(2h+1)$ and $f = 4h$ for $h, t \in \mathbb{N}$, $t \ge 2$.*

Proof: From the definition, it follows that $d = h(2h+1) = |E_0| = |E_{t-1}|$ and that $f = 4h$ is the number of vertices contained in the edges of E_0. We define the edge ordering of E in such a way that all edges of E_r precede the ones of E_s whenever $0 \le r < s < h$. For any $0 \le s < h$, the edges within each E_s are ordered in the same way so that it suffices to specify the edge ordering on $E_0 = E'_0 \cup E''_0 \cup E'''_0$. First, arrange the edges of E_0 such that the edges of E'_0 precede the ones of E''_0 which in turn precede the ones of E'''_0. Then, order the edges of E'_0 arbitrarily. Order the edges of E''_0 "from left to right" meaning that the edge $\{v_i, w_{h+j}\}$ precedes $\{v_k, w_{h+\ell}\}$ whenever $i < k$ or ($i = k$ and $j < \ell$). Similarly, order the edges of E'''_0 such that $\{v_{h+i}, w_j\}$ precedes $\{v_{h+k}, w_\ell\}$ whenever $i < k$ or ($i = k$ and $j < \ell$). Altogether this defines an edge ordering in the sense of Definition 3.2 which clearly satisfies that the first d edges with respect to this ordering constitute E_0 and the last d edges constitute E_{t-1}.

It is left to show that any d consecutive edges contain at most f vertices. We start with the d edges of E_0, which clearly satisfy this condition, and "move" the edges successively along the specified edge ordering. The first h^2 moves amount to moving an edge from E'_0 to the corresponding edge of E'_1. Since the vertex set of E'_1 is contained in the one of E_0, any d consecutive edges along these moves contain at most f vertices. The next $\frac{h(h+1)}{2}$ moves amount to moving an edge from E''_0 to the corresponding edge of E''_1. Suppose that in this process some edge e_0 of E''_0 is replaced by the corresponding edge e_1 of E''_1, i.e., there are exactly $d-1$ edges between e_0 and e_1 with respect to the edge ordering. Here, note that whenever one gets a "new" vertex w_{2h+j} for some $1 \leq j \leq h$ which is contained in the edge $e_1 \in E_1''$ but not contained in the $d-1$ edges lying between e_0 and e_1, the vertex v_j contained in e_0 is no longer contained in any of the edges following e_0. So again, any d consecutive edges along these moves contain at most f vertices. Similarly, the same holds for the next $\frac{h(h+1)}{2}$ moves from E_0''' to the corresponding edge of E_1'''. Altogether we have shown that along the first d moves any d consecutive edges contain at most f vertices. Note that the edge set after d moves is E_1. Therefore, one can proceed exactly the same way to move from E_1 to E_2 and so on until getting to E_{t-1}. □

By Proposition 3.1 a Δ-labelling δ of the graph $H(h;t)$ will lead to a decomposition of the complete bipartite graph $K_{\ell,\ell}$ into ℓ isomorphic copies of $H(h;t)^\delta$, where $\ell = th(2h+1)$. However, in general there is no (d,f)-movement consistent with some translation parameter κ as needed in Theorem 3.4. To this means, we impose an additional condition on the Δ-labelling. The following definition generalizes and adapts the notion of a *wrapped* Δ*-labelling* to the bipartite case, which was introduced in [7] for certain subgraphs of the complete graph.

Definition 4.2 Let $H = (U,E)$, $\ell = |E|$, denote a bipartite graph and let $X, Y \subset U$ with $|X| = |Y|$. A Δ-labelling δ is called a *wrapped* Δ*-labelling* of H relative to X and Y if there exists a $\kappa \in \mathbb{Z}$ coprime to ℓ such that

$$\delta(Y) = \delta(X) + (\kappa, 0) \tag{7}$$

as multisets in $\mathbb{Z}_\ell \times \mathbb{Z}_2$. The parameter κ is also referred to as *translation parameter* of the wrapped Δ-labelling.

For the graphs $H = H(h;t)$, we define $X := \{v_i, w_i \mid 0 \leq i < h\}$ and $Y := \{v_i, w_i \mid ht \leq i < h(t+1)\}$. Furthermore, in the following we consider only wrapped Δ-labellings relative to X and Y for which the stronger condition

$$\delta(v_{i+ht}) = \delta(v_i) + (\kappa, 0) \quad \text{and} \quad \delta(w_{i+ht}) = \delta(w_i) + (\kappa, 0), \tag{8}$$

hold for $0 \leq i < h$. Suppose we have such a labelling δ satisfying Condition (8). Using the notation of Section 3, we denote ℓ the isomorphic copies of $H(h;t)^\delta$ by $H^{(i)}$ and its edge sets by $E^{(i)}$, $i \in \mathbb{Z}_\ell$. Obviously, the (d,f)-movements as described in Proposition 4.1 also define (d,f)-movements for $H(h;t)^\delta$ and hence for $H^{(i)}$, $i \in \mathbb{Z}_\ell$. Let ${\Sigma_0}^{(i)} \subset E^{(i)}$ consist of the first $d = h(2h+1)$ edges. Then from condition (8), it follows that the edge set $E(G^{(0)}) := E^{(0)} \cup {\Sigma_0}^{(\kappa)}$ of the graph $G^{(0)} \subset K_{\ell,\ell}$ can be identified with the edge set of $H(h;t+1)$ (even though the graphs $G^{(0)}$ and $H(h;t+1)$ are not isomorphic if the vertex labelling δ is not injective). Anyway, it is easy to check that the (d,f)-movement of

$H(h;t+1)$ from Proposition 4.1 induces a (d,f)-movement of $G^{(0)}$ from ${\Sigma_0}^{(0)}$ consistent w.r.t. κ. By applying Theorem 3.4 we get the following result.

Theorem 4.3 *From any wrapped Δ-labelling of $H(h;t)$, $t,h \in \mathbb{N}$, satisfying condition (8), one obtains a (d,f)-cluttered ordering of the complete bipartite graph $K_{\ell,\ell}$ with $\ell = th(2h+1)$, $d = h(2h+1)$, and $f = 4h$.*

Actually, one can show that Theorem 4.3 even holds for any wrapped Δ-labelling of $H(h;t)$. One just has to modify the (d,f)-movement described in Proposition 4.1 in some suitable manner, where one still has a lot of freedom concerning the edge ordering within each E_s. However, in the next three subsection we construct wrapped Δ-labellings which all clearly satisfy Condition (8).

4.2 Construction of Wrapped Δ-Labellings for $H(1;t)$

In this subsection, we define a wrapped Δ-labelling of $H(1;t)$ for any $t \in \mathbb{N}$. $H(1,t) = (U,E)$ has $2(t+1)$ vertices and $3t$ edges. For a fixed $t \in \mathbb{N}$, we define $\delta : U \to \mathbb{Z}_{3t} \times \mathbb{Z}_2$ on the vertex set $U = V \cup W$ as follows:

$$\delta(v_j) = \begin{cases} (jt,0), & \text{for } 0 \le j \le t-1, \\ (t^2+1,0), & \text{for } j = t, \end{cases}$$

$$\delta(w_j) = \begin{cases} (j(t-1),1), & \text{for } 0 \le j \le t-1, \\ (t^2+1,1), & \text{for } j = t, \end{cases}$$

where the integers in the first components are considered modulo $3t$. Fig. 3 shows δ for the case $t = 5$ indicating only the first components of $\mathbb{Z}_\ell \times \mathbb{Z}_2$. Note that in this example δ is not injective.

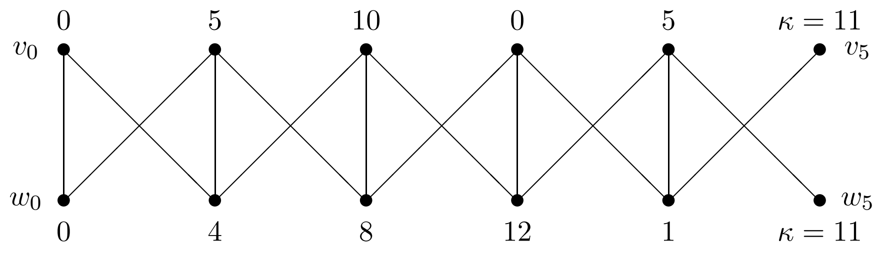


Fig. 3. Some wrapped Δ-labelling of $H(1;5)$, $|E| = 15$, $|V| = 12$, $\kappa = 11$.

We now compute the difference list $\Delta(E)$ of δ defined as in Equation (1):

$$
\begin{aligned}
\Delta(\textstyle\bigcup_{j=0}^{t-1} E'_j) &= (jt - j(t-1) \mid 0 \le j \le t-1) &&= (0, \dots, t-1),\\
\Delta(\textstyle\bigcup_{j=0}^{t-2} E''_j) &= (jt - (j+1)(t-1) \mid 0 \le j \le t-2) &&= (2t+1, \dots, 3t-1),\\
\Delta(\textstyle\bigcup_{j=0}^{t-2} E'''_j) &= ((j+1)t - j(t-1) \mid 0 \le j \le t-2) &&= (t, \dots, 2t-2),\\
\Delta(E''_{t-1} \cup E'''_{t-1}) &= ((t-1)t - (t^2+1), t^2+1-(t-1)^2) &&= (2t-1, 2t).
\end{aligned}
$$

Hence each element of $\mathbb{Z}_{3t}$ appears exactly once in $\Delta(E)$ and the difference condition holds. Obviously, the wrapped-condition (8) relative to $X = \{v_0, w_0\}$ and $Y = \{v_t, w_t\}$ holds as well and the translation parameter $\kappa = t^2 + 1$ is coprime to $3t$ for any $t \in \mathbb{N}$. Therefore, δ defines the desired wrapped Δ-labelling of $H(1, t)$ and, by applying Theorem 4.3, we get the following result.

Theorem 4.4 *For all $t \in \mathbb{N}$ there is a (d, f)-cluttered ordering of the complete bipartite graph $K_{3t,3t}$ with $d = 3$ and $f = 4$.*

Enlarging the window d provides a further useful result. Using the same edge ordering of the edge set of $K_{3t,3t}$ as before one easily checks the following theorem.

Theorem 4.5 *For all $t \in \mathbb{N}$ there is a (d, f)-cluttered ordering of the complete bipartite graph $K_{3t,3t}$ with $d = 3s + r$ and $f = 2(s + 1) + r$, $s > 0$, $r = 0, 1, 2$.*

Note that from Theorem 4.5 only for small d one obtains "good" cluttered orderings in the sense that f is not much bigger than $f_{\min}(K_{3t,3t}, d)$. For example, one obtains a $(3, 4)$-, $(4, 5)$-, or $(6, 6)$-cluttered ordering of $K_{3t,3t}$, $t \in \mathbb{N}$. Theorem 4.5 also gives a $(30, 22)$-cluttered ordering or a $(36, 26)$-cluttered ordering of $K_{3t,3t}$. For some of these graphs, these results can be improved considerably by using the constructions of the next sections.

4.3 Construction of Wrapped Δ-Labellings for $H(2; t)$

In this section, we define a wrapped Δ-labelling of $H(2; t)$ for any $t \in \mathbb{N}$. $H(2; t) = (U, E)$ has $4(t + 1)$ vertices and $10t$ edges. For a fixed $t \in \mathbb{N}$, a labelling δ is a map $\delta : U \to \mathbb{Z}_{10t} \times \mathbb{Z}_2$ on the vertex set $U = V \cup W$. We specify the second component of δ on the vertices $V = (v_0, v_1, \dots, v_{2t+1})$ sequentially by the following list of $2t + 2$ numbers:

$$c_0, c_0 + a, c_1, c_1 + a, \dots, c_j, c_j + a, \dots, c_{t-1}, c_{t-1} + a, c_0 + \kappa, c_0 + a + \kappa,$$

and similarly on the vertices $W = (w_0, w_1, \dots, w_{2t+1})$ by

$$d_0, d_0 + b, d_1, d_1 + b, \dots, d_j, d_j + b, \dots, d_{t-1}, d_{t-1} + b, d_0 + \kappa, d_0 + b + \kappa,$$

where we set

$$
\begin{aligned}
a &:= 6t-1, \quad c_j := 2jt, \qquad\quad j = 0,1,\ldots,t-1,\\
b &:= 6t-2, \quad d_j := 2j(t-1), \quad j = 0,1,\ldots,t-1,\\
\kappa &:= 2t^2+1.
\end{aligned}
$$

All integers are considered modulo $10t$. Note that $|E| = 10t$ and $\kappa = 2t^2 + 1$ are coprime for all $t \in \mathbb{N}$ and that the wrapped-condition (8) is obviously fulfilled. Fig. 4 illustrates the definition for the case $t = 2$.

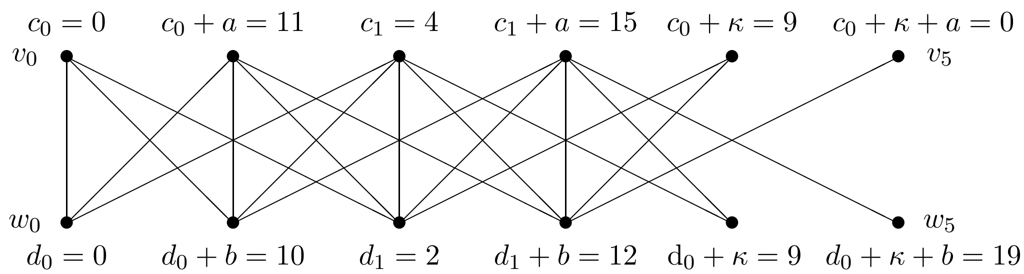


Fig. 4. Some wrapped Δ-labelling of $H(2;2)$, $|E| = 20$, $|V| = 12$, $\kappa = 9$.

We now compute the differences of δ using the notation in (1):

$$
\begin{aligned}
\Delta(E'_j) &= (c_j - d_j, c_j - d_j + (a-b), c_j - d_j + a, c_j - d_j - b)\\
&= (2j, 2j+1, 2j+6t-1, 2j+4t+2) \quad \text{for } j = 0,1,\ldots,t-1,\\
\Delta(E''_{j-1}) &= (c_{j-1} - d_j, c_{j-1} - d_j + a, c_{j-1} - d_j + (a-b))\\
&= (2j+8t, 2j+4t-1, 2j+8t+1) \quad \text{for } j = 1,2,\ldots,t-1,\\
\Delta(E'''_{j-1}) &= (c_j - d_{j-1}, c_j - d_{j-1} - b, c_j - d_{j-1} + (a-b))\\
&= (2j+2t-2, 2j+6t, 2j+2t-1) \quad \text{for } j = 1,2,\ldots,t-1,\\
\Delta(E''_{t-1}) &= (c_{t-1} - d_0 - \kappa, c_{t-1} - d_0 - \kappa + a, c_{t-1} - d_0 - \kappa + (a-b))\\
&= (8t-1, 4t-2, 8t),
\end{aligned}
$$

$$
\begin{aligned}
\Delta(E'''_{t-1}) &= (c_0 + \kappa - d_{t-1}, c_0 + \kappa - d_{t-1} - b, c_0 + \kappa - d_{t-1} + (a-b))\\
&= (4t-1, 8t+1, 4t).
\end{aligned}
$$

From this one easily checks that the first three lists cover all numbers in $\mathbb{Z}_{10t} \setminus \{4t-2, 4t-1, 4t, 8t-1, 8t, 8t+1\}$ exactly once. The missing values are exactly covered by $\Delta(E''_{t-1})$ and $\Delta(E'''_{t-1})$. Thus, δ defines a wrapped Δ-labelling and, by applying Theorem 4.3, we get the following result.

Theorem 4.6 *For all $t \in \mathbb{N}$ there is a (d, f)-cluttered ordering of the complete* *bipartite graph $K_{10t,10t}$ with $d = 10$ and $f = 8$.*

Using the same edge ordering of $K_{10t,10t}$ one obtains the following theorem by enlarging the window d.

Theorem 4.7 *For all $t \in \mathbb{N}$ there is a (d, f)-cluttered ordering of the complete bipartite graph $K_{10t,10t}$ with $d = 10s + r$ and $f = 4(s + 1) + \min(r, 4)$, $s > 0$, $r = 0, 1, \ldots, 9$.*

From this, for example, we get a $(30, 16)$-cluttered ordering of $K_{10t,10t}$. For the graphs $K_{30t,30t}$, $t \in \mathbb{N}$, this is a much better ordering than the $(30, 22)$-cluttered ordering from Theorem 4.5.

4.4 Construction of Wrapped Δ-Labellings for $H(h; 1)$

Next, we define in this section a wrapped Δ-labelling for $H(h; 1)$ for any $h \in \mathbb{N}$. $H(h; 1) = (U, E)$ has $4h$ vertices and $h(2h + 1)$ edges. We define the Δ-labelling $\delta : U \to \mathbb{Z}_{h(2h+1)} \times \mathbb{Z}_2$ on the vertex set $U = V \cup W$ by specifying the first component of δ on the vertices $V = (v_0, v_1, \ldots, v_{2h-1})$ sequentially by the following list of $2h$ numbers:

$$a_0, a_1 \ldots, a_{h-1}, a_0 + \kappa, a_1 + \kappa, \ldots, a_{h-1} + \kappa,$$

and similarly on the vertices $W = (w_0, w_1, \ldots, w_{2h-1})$ by

$$b_0, b_1 \ldots, b_{h-1}, b_0 + \kappa, b_1 + \kappa, \ldots, b_{h-1} + \kappa,$$

where we set

$$\begin{aligned}
a_0 &:= 0, \quad a_i := 2i - (2h + 1), && i = 1, 2, \ldots, h - 1,\\
b_0 &:= 0, \quad b_j := -j(2h + 1) - 1, && j = 1, 2, \ldots, h - 1,\\
\kappa &:= -1.
\end{aligned}$$

All integers are considered modulo $h(2h + 1)$. Obviously, $|E| = h(2h + 1)$ and κ are coprime for any $h \in \mathbb{N}$ and the wrapped-condition (8) is fulfilled. Fig. 5 illustrates the definition for the case $h = 3$.

We now show that δ indeed defines a wrapped Δ-labelling for $H(h; 1)$. To this means we compute the differences of δ using a partition of the edge set of $H(h; 1)$ as shown in Fig. 6. Using the notation of Equation (6), the edges (1) to (4) account for the edge set E_0', the edges (5) to (7) for E_0'' and the edges (8) to (10) for E_0'''.

Note that the following computations are considered modulo $h(2h + 1)$.

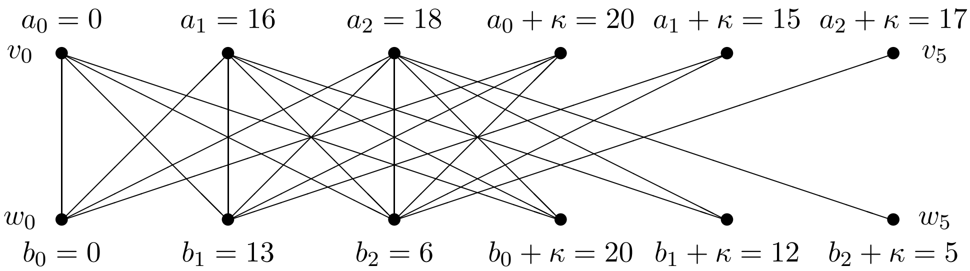


Fig. 5. Some wrapped Δ-labelling of $H(3;1)$, $|E| = 21$, $|V| = 12$, $\kappa = -1$.

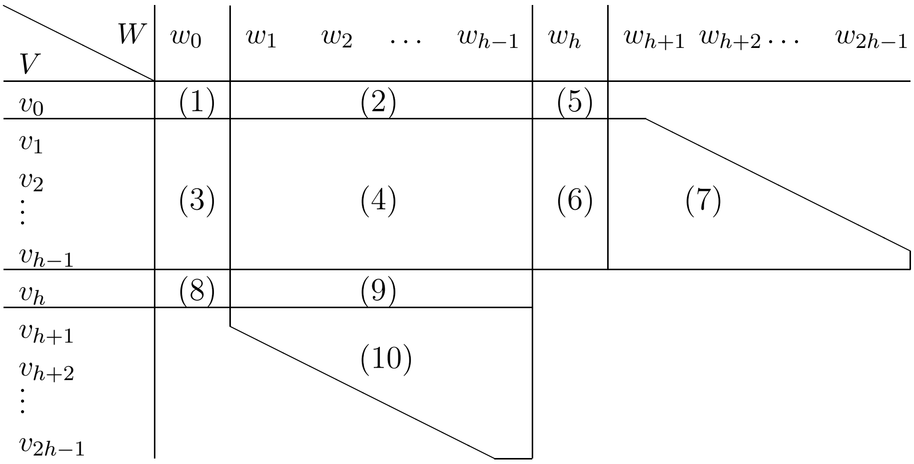


Fig. 6. Edge partition of $H(h;1)$.

$$
\begin{array}{lll}
(1) & a_0 - b_0 & = 0 \\
(2)\ 1 \le j \le h-1: & a_0 - b_j & = j(2h+1) + 1 \\
(3)\ 1 \le i \le h-1: & a_i - b_0 & = (h-1)(2h+1) + 2i \\
(4)\ 1 \le i, j \le h-1: & a_i - b_j & = (j-1)(2h+1) + 2i + 1 \\
(5) & a_0 - (b_0 + \kappa) & = 1 \\
(6)\ 1 \le i \le h-1: & a_i - (b_0 + \kappa) & = (h-1)(2h+1) + 2i + 1 \\
(7)\ 1 \le j \le i \le h-1: & a_i - (b_j + \kappa) & = (j-1)(2h+1) + 2i + 2 \\
(8) & a_0 + \kappa - b_0 & = -1 \\
(9)\ 1 \le j \le h-1: & a_0 + \kappa - b_j & = j(2h+1) \\
(10)\ 1 \le i \le j \le h-1: & a_i + \kappa - b_j & = (j-1)(2h+1) + 2(i-1) + 2
\end{array}
$$

Now, we split up the list (10) and get

$$
\begin{array}{ll}
(10a)\ 1 \le i < j \le h-1: & (j-1)(2h+1) + 2i + 2 \\
(10b)\ 1 \le j \le h-1: & (j-1)(2h+1) + 2
\end{array}
$$

Forming suitable unions of the lists above one obtains the following lists which are also schematically shown in Fig. 7.

$$
\begin{aligned}
(4) + (7) + (10\text{a}) &: 0 \le \ell \le h-2, 3 \le k \le 2h : && \ell(2h+1)+k \\
(1) + (9) &: 0 \le \ell \le h-1 : && \ell(2h+1) \\
(2) + (5) &: 0 \le \ell \le h-1 : && \ell(2h+1)+1 \\
(10\text{b}) &: 0 \le \ell \le h-2 : && \ell(2h+1)+2 \\
(3) + (6) + (8) &: 2 \le k \le 2h : && (h-1)(2h+1)+k
\end{aligned}
$$

<table>
<tr><th>ℓ \ k</th><th>0</th><th>1</th><th>2</th><th>3 … $2h$</th></tr>
<tr><td>0
1
⋮
$h-2$</td><td rowspan="2">(1) + (9)</td><td rowspan="2">(2) + (5)</td><td>(10b)</td><td>(4) + (7) + (10a)</td></tr>
<tr><td>$h-1$</td><td colspan="2">(3) + (6) + (8)</td></tr>
</table>

Fig. 7. Differences $\ell(2h+1)+k$, $0 \le \ell < h$, $0 \le k < 2h+1$, of δ in $\mathbb{Z}_{h(2h+1)}$.

From these lists one can easily read off that all numbers in $\mathbb{Z}_{h(2h+1)}$ appear exactly once as difference of δ which hence defines a wrapped Δ-labelling. Applying Theorem 4.3 we get the following result.

Theorem 4.8 *For all $h \in \mathbb{N}$ there is a (d, f)-cluttered ordering of the complete bipartite graph $K_{h(2h+1),h(2h+1)}$ with $d = h(2h+1)$ and $f = 4h$.*

From Theorem 4.8 one obtains, for example, a $(36, 16)$-cluttered ordering of $K_{36,36}$, which is much better then the $(36, 26)$-cluttered ordering from Theorem 4.5. Or one obtains a $(210, 40)$-cluttered ordering of $K_{210,210}$ in comparison to the $(210, 88)$-cluttered ordering one obtains from Theorem 4.7.

5 Final Remarks

In this paper, we investigated (d, f)-cluttered orderings in the case of complete bipartite graphs based on some strategy suggested by Cohen et al. [7] in the case of complete graphs. We introduced the concept of a (d, f)-movement and formulated a general existence theorem for (d, f)-cluttered orderings. Then, we addressed ourselves to the explicit construction of such orderings. To this means we defined a family of bipartite graphs which allow (d, f)-movements with relatively small parameters f. To achieve the necessary consistency of these movements we generalized and adapted the notion of a wrapped Δ-labellings of bipartite graphs which was first introduced for complete graphs in [7]. However, in general the construction of wrapped Δ-labellings is a difficult combinatorial problem. For the complete graph, Cohen et al. [7] gave a couple of examples of such labellings which were found by means of a computer, but no systematic construction was given. In case of bipartite graphs we gave three explicit constructions of wrapped Δ-labellings for some infinite family of graphs respectively leading to different (d, f)-cluttered orderings.

As formulated in the introduction, one interesting problem is to ask for the minimal number $f_{\min}(\mathcal{S}, d)$ for some set system $\mathcal{S}$. Giving explicit constructions based on some different technique, Cohen et al. [6] solved this problem for $d = 3$ and $d = 4$ for all complete graphs K_n, $n \in \mathbb{N}$, except for the parameters $n \in \{15, 18, 22\}$. In the case of complete bipartite graphs one can show that $f_{\min}(K_{\ell,\ell}, 3) \geq 4$, $f_{\min}(K_{\ell,\ell}, 4) \geq 5$, $f_{\min}(K_{\ell,\ell}, 5) \geq 6$, $f_{\min}(K_{\ell,\ell}, 6) \geq 6$, and $f_{\min}(K_{\ell,\ell}, 10) \geq 8$, for $\ell > 2$. Hence from Theorem 4.5 and Theorem 4.7 one obtains the following corollary.

Corollary 5.1 *For any $\ell \in \mathbb{N}$ which is divisible by three there are optimal $(3, 4)$-, $(4, 5)$-, $(5, 6)$-, and $(6, 6)$-cluttered orderings of $K_{\ell,\ell}$. For any $\ell \in \mathbb{N}$ which is divisible by ten there is an optimal $(10, 8)$-cluttered ordering of $K_{\ell,\ell}$.*

However, further increasing the size d of the window the cluttered orderings from Theorem 4.5 and Theorem 4.7 are far from being optimal. On the other hand, by Theorem 4.8 one obtains (d, f)-cluttered orderings only for a small class of complete bipartite graphs $K_{\ell,\ell}$, $\ell = h(2h + 1)$, $h \in \mathbb{N}$, with $d = \ell$ and $f = 4h$. In this cases, the value $f = 4h$ is comparatively close to $f_{\min}(K_{\ell,\ell}, d)$. Actually, any bipartite graph with $d = h(2h + 1)$ edges contains at least $2\sqrt{2}h$ vertices and hence $f_{\min}(K_{\ell,\ell}, h(2h + 1)) \geq 2\sqrt{2}h$. Therefore, one has $f \leq \sqrt{2} f_{\min}(K_{\ell,\ell}, h(2h + 1))$ for the (d, f)-cluttered ordering of Theorem 4.8.

The explicit construction of good cluttered orderings as well as the specification of good lower and upper bounds for $f_{\min}(\mathcal{S}, d)$ in the case of complete graphs, complete bipartite graphs, or even more general set systems constitute a difficult combinatorial problem and is the content of an ongoing research project of the authors.

Acknowledgement

The first author was supported by a scholarship of the Japanese Government (MEXT). The authors thank the referees for their helpful comments.

References

[1] Beth, T., Jungnickel, D., and Lenz, H.: *Design Theory.* Cambridge University Press, 1999.

[2] Bosák, J.: *Decompositions of Graphs.* Kluwer Academic Publishers, Dordrecht, 1990.

[3] Chee, Y. M., Colbourn, C. J., and Ling, A.: Asymptotically optimal erasure-resilient codes for large disk arrays. *Discrete Applied Mathematics* **102** (2000), 3–36.

[4] Chen, P., Lee, E., Gibson, G., Katz, R., and Patterson, D.: RAID: High-performance, reliable secondary storage. *ACM Computing Surveys* **26** (1994), 145–185.

[5] Cohen, M. and Colbourn, C.: Optimal and Pessimal Orderings of Steiner Triple Systems in Disk Arrays. *Theoretical Computer Science* **297**, Issues 1–3, March 2003, 103–117.

[6] Cohen, M. and Colbourn, C.: Ladder orderings of pairs and RAID performance. *Discrete Applied Mathematics* **138**, Issues 1-2, March 2004, 35–46.

[7] Cohen, M., Colbourn, C., and Froncek, D.: Cluttered orderings for the complete graph. LNCS **2108**, Springer (2001), 420–431.

[8] Hellerstein, L., Gibson, G., Karp, R., Katz, R., and Patterson, D.: Coding techniques for handling failures in large disk arrays. *Algorithmica* **12** (1994), 182–208.