Matching techniques ride to rescue OLED displays

Andreas Karrenbauer*

Institute of Mathematics, EPFL, Lausanne, Switzerland
andreas.karrenbauer@epfl.ch

Abstract. Combinatorial optimization problems have recently emerged in the design of controllers for OLED displays. The objective is to decompose an image into subframes minimizing the addressing time and thereby also the amplitude of the electrical current through the diodes, which has a direct impact on the lifetime of such a display. To this end, we model this problem as an integer linear program. Subsequently, we refine this formulation by exploiting the combinatorial structure of the problem. We propose a fully combinatorial separation routine for the LP-relaxation based on matching techniques. It can be used as an oracle in various frameworks to derive approximation algorithms or heuristics. We establish NP-hardness and hardness of approximation. Nevertheless, we are able to work around this issue by only focusing on a subsets of the variables and provide experimental evidence that they are sufficient to come up with near optimal solutions in practice. On this basis, one can derive custom-tailored solutions adapting to technical constraints such as memory requirements. By allowing the addressing of distributed doublelines, we improve the addressing time in cases where previous approaches fall short due to their restriction to consecutive doublelines.

1 Introduction

Organic **L**ight **E**mitting **D**iodes (**OLED**s) have been a hot topic on the display market in the last years as the sizes of commercially available displays increased significantly. Moreover, they provide many advantages over current technology, such as Liquid Crystal Display (LCD).The image and video displayed has a very high contrast and a viewing angle of nearly 180 degrees. It reacts within 10 microseconds, which is much faster than the human eye can catch and is therefore well suited for video applications. Moreover, the display is physically flexible.

There are two different OLED technologies called *active matrix* (AM) and *passive matrix* (PM). The former is more expensive but offers a longer lifetime than the latter. Their limited lifetime is one major reason why there are only small-sized displays on the mass market. For mobile phones or digital cameras, large state of the art OLED displays are either too expensive or suffer from insufficient lifetime.

While a lot of research is conducted on the material science side, the so-called *Consecutive Multiline Addressing Scheme* for passive matrix OLED displays [1] tackles the lifetime-problem from an algorithmic point of view. It is based on the fact that equal rows can be displayed simultaneously with a lower electrical current than in

* Supported by Deutsche Forschungsgemeinschaft (DFG) within Priority Programme 1307 "Algorithm Engineering"

a serial manner [2, 3]. Here we restrict ourselves to an informal description for self-containment.

Fig. 1. Schematic electrical circuit of a display

A (passive matrix) OLED display has a matrix structure with n rows and m columns as depicted in Figure 1. At any crossover between a row and a column there is a vertical diode which works as a pixel. The image itself is given as an integral non-negative $n \times m$ matrix. For the sake of simplicity, we first consider the case of binary matrices, i.e. black/white images, and generalize to greyscale and colored images later on.

Consider the contacts for the rows and columns as switches. For the time the switch of row i and column j is closed, an electrical current flows through the diode of pixel (i, j) and it shines. Hence, we can not control each pixel directly. Therefore, such passive matrix displays are traditionally driven row by row in a round-robin fashion. At a sufficiently high framerate, say 50 Hz, the human eye perceives for each pixel only the average over time. Since, we may skip rows that are completely dark, the addressing time varies from to image to image. To maintain the brightness at the same level, we lower the amplitude of electrical current that is sourced into the columns. This procedure has two desired side effects: The power consumption is reduced and their lifetime is extended since high amplitudes of the electrical current are the major issues with respect to the lifetime of the diodes [4]. We can even save more time per frame, if we drive two rows simultaneously. However, this only works, if their content is equal as in the following example. As one can see, we need 5 units of time to display the image in the traditional way, i.e. row by row, whereas 3 units of time are sufficient with so-called *Distributed Doubleline Addressing* (DDA).

$$\begin{pmatrix} 0&1&1&0 \\ 1&0&0&1 \\ 1&1&1&1 \\ 1&0&0&1 \\ 1&0&0&1 \end{pmatrix} = \begin{pmatrix} 0&1&1&0 \\ 0&0&0&0 \\ 0&1&1&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{pmatrix} + \begin{pmatrix} 0&0&0&0 \\ 1&0&0&1 \\ 1&0&0&1 \\ 0&0&0&0 \\ 0&0&0&0 \end{pmatrix} + \begin{pmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 1&0&0&1 \\ 1&0&0&1 \end{pmatrix}$$

We thereby gain 40% of the time to display this image and as said before we may decrease the amplitude of the electrical current by that amount. Hence, it remains to find an algorithm that computes such a decomposition to benefit from Distributed Doubleline Addressing (DDA). We use the term *distributed* to distinguish from previous work where it is only allowed to combine *consecutive* lines. That is, we have to display the first matrix on the right-hand side of our example in two steps, which therefore only permits a reduction of the electrical current by 20% in that case.

To benefit from this decomposition in practice, we should adhere to the following design criteria. Since the algorithm has to be implemented on a driver chip attached to the display, it must have low hardware complexity allowing small production costs. Consequently it has to rely only on a small amount of memory and it should be *fully combinatorial*, i.e. only additions, subtractions, and comparisons are used. Though it

has to solve or approximate the optimization problem, which is formally described in Section 2, in real-time. We do not fulfill all these requirements in one shot. We rather apply an algorithm engineering process that approaches these goals in several iterations.

Previous Work

Algorithmic questions on the restriction to *Consecutive Doubleline Addressing* (CDA) have been discussed by Eisenbrand et al. [2, 5]. In these papers, the authors also considered to combine more than two lines simultaneously, but only consecutive ones. Other approaches based on *Non-negative Matrix Factorization* [6, 7] have been outlined by Smith et al. [8] and Smith [9].

Contributions of this paper

We describe an algorithm engineering process to develop efficient solutions for a real-world problem. That is, we first model the matrix decomposition problem of *Distributed Doubleline Addressing* as an integer program. On this basis, we improve the formulation by exploiting its combinatorial structure until we achieve a solution which is applicable in practice. On the theory side, we prove that computing optimal decompositions is NP-complete and also hard to approximate within a certain constant factor. To this end, we introduce the *Matchable Subset Problem* as a special case of our real-world problem. Though the complexity results sound discouraging, they give useful insight into the structure of the problem and also hints to the applicable methods. That is, we adopt approximation techniques such as LP-rounding to come up with a promising method in practice. We derive two LP formulations of our problem: A concise one and one with exponentially many constraints. Though the former is of polynomial size, it is impractical and inferior to the latter. This interesting and on the first glance counter-intuitive behavior is due to the fact that we apply techniques known from b-matching to develop an efficient fully combinatorial algorithm for the separation problem of the exponentially many constraints. Finally, we propose parameterized heuristics to achieve a solution, which is applicable in practice. We conclude with a presentation of some computational results showing the improvement with respect to previous work. We thereby show that methods from combinatorial optimization are well-suited to tackle algorithmic challenges in the design of flat panel display drivers.

2 A Linear Programming Formulation

In this section, we will briefly introduce a linear programming model for DDA. The interested reader is referred to [2] for a more profound elaboration on the technical details. For the sake of simplicity, we restrict ourselves to the special case of black/white images given by binary matrices $R = (r_{ij}) \in \{0,1\}^{n \times m}$ for time being. Let the binary variables $f_j(i,k) \in \{0,1\}$ denote whether the switch for column j is closed while the switches of the rows i and k are closed. Note that if i equals k, then the corresponding variables represent a single line. Moreover, $f_j(i,k)$ and $f_j(k,i)$ represent the same

switches and hence we implicitly require $f_j(i,k) = f_j(k,i)$ in the following. To get a lossless decomposition of the image R, the following constraints must hold.

$$\sum_{k=1}^{n} f_j(i,k) = r_{ij} \qquad \text{for all } i,j$$

Recall that our objective is to minimize the addressing time for each given image. Clearly, if we have for some pair $\{i,k\}$ that $f_j(i,k) = 0$ for all $j \in \{1,\ldots,m\}$, we can skip this doubleline (or singleline if i equals k). Hence, the total number of subframes is given by

$$\sum_{i \le k} \max\{f_j(i,k) : j = 1,\ldots,m\}.$$

We apply the standard trick to derive a linear programming formulation by replacing each maximum by an auxiliary variable $u(i,k)$. This yields

$$\begin{aligned} \min \quad & \sum_{i \le k} u(i,k) & & \\ \text{s.t.} \quad & \sum_{k=1}^{n} f_j(i,k) = r_{ij} & & \text{for all } i,j \\ & 0 \le f_j(i,k) \le u(i,k) & & \text{for all } i,j,k. \end{aligned} \tag{1}$$

This LP formulation is not integral in general, which can be verified by an example with three rows and a single column with $R = (1,1,1)^T$. The fractional optimum of $\frac{3}{2}$ is attained at $u(1,2) = u(2,3) = u(1,3) = \frac{1}{2}$, whereas the integer optimum is 2. This comes to no surprise regarding the hardness results of Section 4 for integer DDA. However, we stick to the LP formulation as it can be used with well-known approximation techniques, e.g. randomized rounding. We can also work around this issue on the technical side. To this end our display controller must work with a higher precision than the input data. Let us assume for now that it has an arbitrary precision. Then $u(i,k)$ denotes the fraction of time for which the row-switches i and k are simultaneously closed. Moreover, $f_j(i,k)$ is the fraction of time for which the column-switch j is closed while the switches for rows i,k are simultaneously closed. Now the generalization to greyscale images becomes straightforward by taking $R \in [0,1]^{n \times m}$. Since the images usually have a fixed resolution, we may assume that the input data is scaled to integers in $\{0,\ldots,\varrho\}$ for some integer ϱ, e.g. $\varrho = 255$ for 8-bit resolution. Mathematically, there is no difference between greyscale and colored images. In the latter case, we just have differently colored OLEDs at the respective pixels.

Experimental Evaluation

This basic LP formulation permits a first evaluation of the DDA approach using standard software. Though it is clear that we will not be able to implement a general purpose LP-solver on a chip that drives such a display, we can use the LP-solutions as a benchmark for our further algorithms and heuristics. Although the formulation (1) is concise in the theoretical sense as only a polynomial number (with respect to the input size) of

variables and constraints are used, the programs get huge for typical OLED displays. To give the reader a figure, we present the numbers for QQVGA resolution, e.g. sub-displays for mobile phones. There, we have $n = 120$ rows and $m = 3 \cdot 160 = 480$ columns. This means that we have $(m+1)n(n+1)/2 \approx 3.5 \cdot 10^6$ variables and about the same number of constraints in our LP. Hence, it comes to no surprise that CPLEX 10.0 takes for a much smaller LP of 30 rows already about 4 minutes on a 2.8 GHz Dualcore AMD Opteron with 16 GB RAM and the run for QQVGA did not finish within 300 hours. Clearly, we must have a deeper look at the theoretical properties of our problem to make any progress. To this end, we refine the formulation of our problem by exploiting its combinatorial structure.

3 Combinatorial Refinement

A closer look at the LP formulation (1) reveals that the objective only depends on the u-variables. Moreover, if those variables were fixed, then the problem would decompose into m independent parts. Hence, we wish to have an efficient method to solve the separation problem for the u-variables. That is, given an assignment to the u-variables, to decide whether all the independent parts are feasible, and if not, to return a violated inequality.

To this end, we introduce a combinatorial formulation of our problem. It is straightforward to consider an undirected graph $G = (V, E)$ where each vertex $i \in V$ corresponds to a row of the display and the edgeset E represents the pairs of row-switches. Note that we allow self-loops in G to model the singlelines. If no further restrictions are given, then G is the complete graph on n nodes.

In the following, we consider the column vectors of an image R as functions $r_j : V \to \mathbb{Z}_{\geq 0}$. A lossless decomposition is then considered as a *perfect r_j-matching problem* for each column $j = 1, \ldots, m$. That is, the set of feasible timings for column j is given by the polyhedron

$$P_j := \{f \in \mathbb{R}^{|E|}_{\geq 0} : f(\delta(i)) = r_j(i) \quad \forall i \in V\}$$

where $\delta(\cdot)$ denotes the set of incident edges and $f(\delta(\cdot))$ means the sum over variables of these edges.

Recall that the timings for the row-switches is determined by the maxima over all columns. That is, we have a variable $u(e)$ for each edge $e \in E$. A row-timing $u : E \to \mathbb{Z}_{\geq 0}$ is feasible, if and only if for each column $j \in \{1, \ldots, m\}$ there is a feasible matching $f_j \in P_j$ with $f_j \leq u$. Hence, a row-timing u is feasible for a column j if and only if it is contained in the up-hull of P_j, i.e. $u \in P_j^{\uparrow} := P_j + \mathbb{R}^{|E|}_{\geq 0}$. Thus, the set of feasible row-timings is given by the polyhedron

$$P := \bigcap_{j=1}^{m} P_j^{\uparrow}.$$

The problem can now be divided into two parts and understood as follows.

1. Find a row-timing $u \in P$ that minimizes the sum $u(E)$.

2. For each column $j = 1, \ldots, m$, compute a u-capacitated perfect r_j-matching f_j representing the timings for the corresponding column switches.

Note that in the second step, the columns become independent and the matching problems could be solved in parallel. Moreover, there are several combinatorial algorithms known from literature to solve this task.

It remains to find a good characterization of P, e.g. by an efficient combinatorial algorithm for the separation problem. That is, given a vector u' determine an inequality that is valid for all elements of P but violated for u', or assert that no such inequality exists and hence u' is contained in P.

Theorem 1. *The polyhedron P is determined by the inequalities*

$$2u(E[X]) + u(\delta(X) \setminus \delta(Y)) \geq r_j(X) - r_j(Y) \tag{2}$$

for all $X \subseteq V$, $Y \subset N(X)$, $j = 1, \ldots, m$ where $E[X]$ denotes the inner edges of X excluding the self-loops, $\delta(\cdot)$ also contains the self-loops of X, and $N(X) \subset V$ is the neighborhood of X in G.

We sketch the proof in the following, since it reveals our implementation of the separation routine for these inequalities. We show first that each of the Inequalities (2) is valid for P. To this end, arbitrarily fix $u \in P$, $X \subseteq V$, $Y \subset N(X)$, and $j \in \{1, \ldots, m\}$. Since $u \in P$, there is a $f \in P_j$ with $f \leq u$. Hence,

$$\begin{aligned} r_j(X) - r_j(Y) &= 2f(E[X]) + f(\delta(X)) - 2f(E[Y]) - f(\delta(Y)) \\ &\leq 2u(E[X]) + u(\delta(X) \setminus \delta(Y)). \end{aligned}$$

To prove sufficiency, we transform the problem to the uncapacitated case. To this end, we split each edge $e = \{i, k\} \in E$ by two new nodes e_1, e_2 such that we get a new graph $G' = (V', E')$ with

$$\begin{aligned} V' &:= V \quad \dot\cup \quad \{e_1, e_2 : e \in E\} \\ E' &:= \big\{\{i, e_1\}, \{e_1, e_2\}, \{e_2, k\} : e = \{i, k\} \in E\big\}. \end{aligned}$$

Note that the self-loops transform into 3-cycles as depicted in Figure 2.

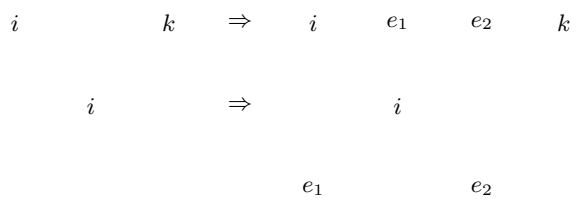

Fig. 2. Transformation to the uncapacitated case

Consider j to be fixed for time being. We define $b : V' \to \mathbb{Z}_{\geq 0}$ with $b(i) := r_j(i)$ for all $i \in V$, $b(e_1) := b(e_2) := u(e)$ for all $e \in E$ that are no self-loops, and

$b(e_1) := b(e_2) := u(e)/2$ for all self-loops e. It is easy to verify that a fractional perfect b-matching in G' corresponds to a fractional u-capacitated perfect r_j-matching in G and vice versa. The value attributed to the middle segment in G' determines the slack of the corresponding edge in G. The transformation to the uncapacitated case allows us to use a well-known characterization of the existence of perfect $2b$-matchings (cf. Corollary 31.5a in [10]) that we can state in our context as follows.

Lemma 1. *There is a fractional perfect b-matching for G' if and only if*

$$b(N'(S)) \geq b(S)$$

for each stable set S of G' where N' denotes the neighborhood in G'.

Let S be a stable set in G'. Define the set $X := S \cap V$, i.e. the nodes of S that correspond to nodes in the original graph G. Moreover, let $Y := N'(S \setminus X) \cap V$ and $F \subseteq \delta(Y)$ such that its edges correspond to the nodes of $S \setminus X$ in G'. Hence,

$$b(S) = b(X) + b(S \setminus X) = r_j(X) + u(F)$$

and

$$b(N'(S)) = 2u(E[X]) + u(\delta(X) \setminus F) + u(F) + r_j(Y).$$

It follows that $b(N'(S)) \geq b(S)$ is equivalent to

$$2u(E[X]) + u(\delta(X) \setminus \delta(Y)) \geq r_j(X) - r_j(Y).$$

It remains to show that it is sufficient to consider only $Y \subseteq N(X)$ in the original graph. However, this is easy to see since any $y \in Y \setminus N(X)$ would weaken the right-hand side and would leave the left-hand side unchanged.

Algorithmically, a violated inequality for a given assignment $u : E \to \mathbb{Z}_{\geq 0}$ can be found by a further transformation of the uncapacitated perfect b-matching problem to a transportation problem. That is, we construct a bipartite graph G'' such that each part of the bipartition consists of a copy of V', say $V'' := V_1' \dot\cup V_2'$, and for each edge $\{v, w\} \in E'$ we have the two edges $\{v_1, w_2\}$ and $\{v_2, w_1\}$ in E''. By directing the edges from V_1 to V_2 and considering the nodes of V_1 as supplies and the nodes of V_2 as demands, the separation problem becomes a transshipment problem, which can be solved by a maximum flow computation. If and only if the value of the maximum flow equals $b(V')$, then there exist a fractional perfect b-matching in G'. If the value of the maximum flow, or by duality the minimum cut, is smaller, then the nodes of G'' constituting a minimum cut also represent a vertex cover $y : V'' \to \{0, 1\}$ of the same weight. By setting $z(v) := y(v_1) + y(v_2)$, we get a 2-vertex cover of G' with

$$\sum_{v \in V'} b(v)z(v) < b(V').$$

Moreover, the set $S := \{v \in V' : z(v) = 0\}$ yields a stable set, while $N'(S) = \{v \in V' : z(v) = 2\}$. Hence,

$$b(V') > 2b(N'(S)) + b(V' \setminus N'(S) \setminus S) = b(N'(S)) + b(V' \setminus S),$$

which gives the equivalent violated inequality $b(S) > b(N'(S))$.

3.1 Further Improvements

In principle, we could start the separation with the zero-vector. However, it is much more efficient to provide a starting set of valid inequalities which is of moderate size and easy to solve but still yields a dual solution which is not too far from the optimum.

It is straightforward to select the inequalities arising from the sets $X = \{i\}$ and $Y = \emptyset$ for each $i \in V$. This means, we simply bound the variables in the perfect matching constraints by the corresponding capacities and get

$$u(\delta(i)) \geq \overline{r}_i \tag{3}$$

where $\overline{r}_i := \max\{r_{ij} : j = 1, \ldots, m\}$.

For our application, it is easy to see that the optimal objective value of this partial solution is at least half of the optimum of the whole problem, since $\overline{u}(i,i) := \overline{r}_i$ and $u(i,k) := 0$ for all $i \neq k$ is a feasible solution and

$$2u(E) \geq \sum_{i=1}^{n} \overline{r}_i$$

holds by summing up the inequalities (3) and the non-negativity constraints for $u(i,i)$.

Note that these inequalities together with the non-negativity constraints determine the fractional $\overline{r}$-edge cover polytope. This has the following two consequences for us: Firstly, there is a fully combinatorial algorithm to compute a minimum fractional b-edge cover. Secondly, the integer $\overline{r}$-edge cover polyhedron is contained in the integer hull P_I of P. Hence, we may also add the valid inequalities

$$u(E[X] \cup \delta(X)) \geq \left\lceil \frac{\overline{r}(X)}{2} \right\rceil \qquad \text{for all } X \subseteq V \tag{4}$$

to the description of P_I. Recall that we made the temporary assumption that our display controller works with arbitrary precision. But in the real world, this is hardly possible since our digital circuit shall work with a fixed clock frequency. Hence, there is a minimum amount of time for which switches can be closed and opened again. Thus, we only have fixed precision, which is equivalent to require the variables to be integer by appropriate scaling.

Nevertheless, the separation routine has not become useless since we already get a very simple approximation algorithm by simply rounding each fractional variable to the next greater integer. Note that then the obtained integer solution has an objective value within an additive error of $|E|$. Moreover, the separation routine can be used within the framework of [2] to come up with fully combinatorial heuristics.

It is natural to ask whether there is a completely different approach to solve the problem exactly in polynomial time. But there is little hope because of the NP-completeness result of Section 4. Before we come to this section, we give a brief overview of the experimental results with respect to the combinatorial separation.

3.2 Experimental Evaluation

We implemented the combinatorial separation using the LEDA 6.1 library to solve the transportation problem by the built-in MAXFLOW routine. We can use this separation

to speed up the solution time of CPLEX. For example, the instance with 30 rows mentionend before is now solved in 14 seconds instead of 240 seconds. However, it is still not possible to solve the LP relaxation of a full QQVGA instance in timely manner.

4 Hardness Results

We show in this section that already the restriction to the black/white case, i.e. binary matrices $R \in \{0,1\}^{n \times m}$, is NP-complete and also hard to approximate. To this end, we define the *Matchable Subset Problem* and analyze its complexity.

Definition 1 (Matchable Subset Problem). *Given an undirected graph $G = (V, E)$ and m subsets of the nodes $V_1, \ldots, V_m \subseteq V$, find an edgeset $\tilde{E} \subset E$ of minimum cardinality such that for each $j \in \{1, \ldots, m\}$ the set V_j is matchable in $\tilde{G} = (V, \tilde{E})$, i.e. there is a perfect matching in the subgraph of $\tilde{G}$ induced by V_j.*

Theorem 2. *The Matchable Subset Problem is NP-complete, even when restricted to complete graphs (with or without self-loops).*

Proof. Clearly, the problem is in NP. We show hardness by a reduction from vertex cover. Given an undirected graph $G = (V, E)$, we construct an undirected graph $G' = (V', E')$ as follows. Let

$$\begin{aligned} V' &:= \quad \{s\} \quad \dot\cup \quad V \quad \dot\cup \quad \{t_e : e \in E\} \\ E' &:= \quad \big\{\{s,u\} : u \in V\big\} \quad \dot\cup \quad \big\{\{u,t_e\},\{v,t_e\} : e = \{u,v\} \in E\big\} \end{aligned}$$

and let the matchable subsets be induced by the nodes $\{s, u, v, t_e\}$ for each original edge $e = \{u, v\} \in E$. An illustration is given in the left of Figure 3.

Fig. 3. The construction for the hardness-proof is illustrated on the left and the one for the decomposition problem on the right.

Given an edgeset $\tilde{E} \subseteq E'$ such that for every $e = \{u, v\} \in E$ the set $\{s, u, v, t_e\}$ is matchable in the graph $(V', \tilde{E})$, we define the nodeset $C := \{u \in V : \{s, u\} \in \tilde{E}\}$. By construction, we have that $|C| = |\tilde{E}| - |E|$. We show next that C is a vertex cover in G. Let $e = \{u, v\}$ be an arbitrary edge. This implies that $\big\{\{s, u\}, \{s, v\}\big\} \cap \tilde{E} \neq \emptyset$, since $\{s, u, v, t_e\}$ has a perfect matching within $\tilde{E}$. Hence, $\{u, v\} \cap C \neq \emptyset$, which proves that C is a vertex cover in G.

Conversely, if C is a vertex cover in G, then we define $\tilde{E} \subseteq E'$ as follows. We distinguish the two cases if an edge of G is covered by one or two vertices in C. If $e = \{u, v\} \in E$ is covered by exactly one vertex in C, say $C \cup e = \{u\}$, we include the edges $\{s, u\}$ and $\{v, t_e\}$ in $\tilde{E}$. If both of ends of an edge $e = \{u, v\}$ are contained in the cover C, then we include $\{s, u\}$, $\{s, v\}$, and an arbitrary edge of $\{u, t_e\}$ and $\{v, t_e\}$. This yields $|\tilde{E}| = |C| + |E|$ and moreover for each edge $e = \{u, v\} \in E$ the set $\{s, u, v, t_e\}$ has a perfect matching within $\tilde{E}$.

Including also the edges $\{u, v\}$ and $\{s, t_e\}$ in E' does not help as they are only contained in one induced subgraph. Moreover, it is easy to see that self-loops are not suited to improve the solution. □

The relation to DDA is as follows. Consider the complete graph and subsets $V_1, \ldots, V_m$ of its nodes. For each $j = 1, \ldots, m$, let r_j be the characteristic vector of V_j. The shortest DDA timing for the matrix R made up by the column vectors r_j is then equal to the minimum-cardinality edgeset solving the Matchable Subset Problem. Hence, the vertex cover problem for a graph $G = (V, E)$ can be solved as follows. From the node-edge-incidence matrix $A \in \{0, 1\}^{|V| \times |E|}$, we construct the image $R \in \{0, 1\}^{(1+|V|+|E|) \times |E|}$ as shown on the right of Figure 3. The optimum number of timesteps to display the image using DDA is equal to the minimum size of a vertex cover of G plus the number of edges in G. Note that the constructed graph G' does not contain odd cycles and thus constraining the input to bipartite graphs is not a restriction. Furthermore, it does not make the problem easier if we also consider fractional perfect matchings.

Note that vertex cover is hard to approximate within a factor $\alpha > 1.36$ [11]. Moreover, hardness of approximation also holds for graphs with bounded degree [12]. Thus, we get the following theorem.

Theorem 3. *There is a constant β such that it is NP-hard to approximate the Matchable Subset Problem within a factor of β.*

Proof. An approximation algorithm for the Matchable Subset Problem with guarantee β yields an approximation algorithm for vertex cover with

$$|C| = |\tilde{E}| - |E| \leq \beta(OPT_{vc} + |E|) - |E| \leq [(\Delta + 1)\beta - \Delta]OPT_{vc}$$

where Δ denotes the maximum degree of G. Hence, if it is hard to approximate vertex cover on a graph with maximum degree Δ within an approximation factor $\alpha(\Delta)$, then it is hard to approximate the matchable subset problem within an approximation factor

$$\beta = \frac{\alpha(\Delta) + \Delta}{\Delta + 1}$$

□

We get $\beta \geq \frac{261}{260} = 1.00385$ by the numbers from [12] and graphs with maximum degree 4. By the previous considerations, this also holds for the shortest DDA timing.

5 Cutting the Bandwidth

In the previous Section, we have seen that the problem is hard for complete graphs. Moreover, the experimental evaluations have shown that the large number of variables prevents ourselves from solving even the LP-relaxation in a timely manner. Hence, it is natural to dismiss some (or most) of them, i.e. to consider a suitable subgraph. From previous work [5], we know that Consecutive Doubleline Addressing (CDA) works pretty well in practice. A close look reveals that CDA is a special case of DDA, when we consider a path with n nodes (and self-loops at every node) as the corresponding graph. It is easy to see that this graph has bandwidth 1. If we take the square of this graph, i.e. inserting edges that skip one node on the path, we get bandwidth 2. The third power yields bandwith 3, and so on. Note that the n-th power, i.e. bandwidth n, is again the complete graph. We use the same testset of images as in [2] and [5] in QQVGA resolution (i.e. $n = 120$ and $m = 3 \cdot 160 = 480$) to compare the different bandwidths $b = 1, 2, 3, 4$. We do so with a small uncertainty by taking the mean of the LP-relaxation and the objective value after the naive rounding. This is justified since the error is negligible and the running time for solving the integer linear program is much higher, e.g. 27 seconds compared to 51.5 minutes. In fact, the error is so small that the error bars would not exceed the symbol size in Figure 4.

Fig. 4. Comparison between CDA ($b = 1$) and DDA with $b = 2, 3, 4$. The line marks the break even. The lower a symbol the better performs the algorithm.

The reduction factor of the worst instance for CDA dropped from 63% via 58% and 54% to 52% for $b = 2, 3, 4$, respectively. As one can see in Figure 4, there is a saturation with growing bandwidth for real-world instances. However, for artificial images like icons, text, wallpapers for mobile displays, etc. where the mean reduction by CDA is only 63% [5], we strongly believe that DDA with small bandwidth, e.g. $b = 3$, will be of great interest. Moreover, it will decrease the number of bad instances.

Hence, future work includes the development of an efficient hardware implementation specialized for graphs with bounded bandwidth. To this end, it is usefull to investigate the black/white case. While the problem is NP-complete for general graphs as shown in Section 4, is is solvable in polynomial time for $b = 1$ [5]. Since the theoretical research to come up with the polynomial time algorithm for $b = 1$ has lead to an efficient approximation algorithm in practice, the complexity of the Machable Subset Problem on graphs with bounded bandwidth $b > 1$ is a relevant open problem. A long term goal is to combine more than two distributed rows, but this is also more demanding from the technical point of view.

Since CDA has recently entered the market as part of Dialog Semiconductor's SMARTXTEND™ technology, which is included a 3" W-QVGA OLED displays of TDK, and the work presented in this paper further improves the addressing time, we strongly believe that DDA will follow with the next version of the display driver.

References

1. Xu, C., Wahl, J., Eisenbrand, F., Karrenbauer, A., Soh, K.M., Hitzelberger, C.: Verfahren zur Ansteuerung von Matrixanzeigen. Patent 10 2005 063 159, Germany (2005)
2. Eisenbrand, F., Karrenbauer, A., Skutella, M., Xu, C.: Multiline Addressing by Network Flow. In Erlebach, T., Azar, Y., eds.: ESA 2006. Volume 4168 of LNCS., Springer (2006) 744–755
3. Xu, C., Karrenbauer, A., Soh, K.M., Wahl, J.: A New Addressing Scheme for PM OLED Display. In Morreale, J., ed.: SID 2007 International Symposium Digest of Technical Papers. Volume XXXVIII., Society for Information Display (May 2007) 97–100
4. Soh, K.M., Xu, C., Hitzelberger, C.: Dependence of OLED Display Degradation on Driving Conditions. Proceedings of SID Mid Europe Chapter Fall Meeting (2006)
5. Eisenbrand, F., Karrenbauer, A., Xu, C.: Algorithms for longer OLED Lifetime. In Demetrescu, C., ed.: WEA 2007. Volume 4525 of LNCS., Springer (2007) 338–351
6. Paatero, P., Tapper, U.: Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. Environmetrics **5** (1994) 111–126
7. Lee, D.D., Seung, H.S.: Algorithms for non-negative matrix factorization. Advances in Neural Information Processing Systems **13** (2001)
8. Smith, E., Routley, P., Foden, C.: Processing digital data using non-negative matrix factorization. Patent GB 2421604A (2005)
9. Smith, E.C.: Total Matrix Addressing (TMA^{TM}). In Morreale, J., ed.: SID 2007 International Symposium Digest of Technical Papers. Volume XXXVIII., Long Beach, USA, Society for Information Display (May 2007) 93–96
10. Schrijver, A.: Combinatorial Optimization - Polyhedra and Efficiency. Volume 24 of Algorithms and Combinatorics. Springer Verlag (2003)
11. Dinur, I., Safra, S.: The importance of being biased. In: STOC. (2002) 33–42
12. M. Chlebík and J. Chlebíková: Inapproximability Results for Bounded Variants of Optimization Problems. In: FCT 2003. Volume 2751 of LNCS., Springer Berlin / Heidelberg (2003) 123 – 145