Available online at www.sciencedirect.com
SCIENCE DIRECT
Discrete Applied Mathematics 154 (2006) 1335 – 1343
DISCRETE APPLIED MATHEMATICS
www.elsevier.com/locate/dam

Complexity results on restricted instances of a paint shop problem for words

P. Bonsma[a], Th. Epping[b], W. Hochstättler[c]

[a] *Department of Applied Mathematics, University of Twente, P.O. Box 217, 7500 AE Enschede, The Netherlands*
[b] *Department of Mathematics, BTU Cottbus, Germany*
[c] *Fern Universität in Hagen, D-58084 Hagen, Germany*

Received 15 September 2003; received in revised form 2 September 2004; accepted 6 May 2005
Available online 24 January 2006

Abstract

We study the following problem: an instance is a word with every letter occurring twice. A solution is a 2-coloring of its letters such that the two occurrences of every letter are colored with different colors. The goal is to minimize the number of color changes between adjacent letters.

This is a special case of the paint shop problem for words, which was previously shown to be $\mathcal{NP}$-complete. We show that this special case is also $\mathcal{NP}$-complete and even $\mathcal{APX}$-hard. Furthermore, derive lower bounds for this problem and discuss a transformation into matroid theory enabling us to solve some specific instances within polynomial time.

Keywords: $\mathcal{APX}$-hardness; Binary matroids; MaxFlow-MinCut; Paint shop; $\mathcal{NP}$-completeness; Sequencing

1. Introduction

An *alphabet* Σ is a set of letters. A *word* w is an ordered sequence of letters from an alphabet. The same letter can appear at multiple positions in a word. In [5], the following problem is studied: an instance consists of a word w, a set of colors $C = \{1, \ldots, c\}$, and a color requirement $r_{\sigma i} \in \mathbb{N}_{\geqslant 0}$ for every letter $\sigma \in \Sigma$ in the word and every color $i \in C$ which states that exactly $r_{\sigma i}$ occurrences of letter σ must be colored with color i. A feasible solution is a coloring of the letters satisfying the color requirement. If the letters σ and σ' appearing at position i and $i+1$ are colored with different colors, we say there is a *color change* between position i and $i+1$. If it is unambiguous, we will also say that there is a color change between the consecutive letters $\sigma\sigma'$. The goal is to find a feasible solution that minimizes the number of color changes.

This problem is called the *paint shop problem for words* (PPW), and is motivated by an application in car manufacturing [5]: different letters represent different car types. To avoid the need for large depots, cars are built to order. So in a given period, for every car type it is known how many cars of each color should be produced. The word represents the order in which the uncolored car bodies arrive at the paint shop of the production line. Changing colors here is a costly process and should be minimized. Even though in practice there may exist some possibilities to change the order

E-mail addresses: bonsma@math.utwente.nl (P. Bonsma), thomas.epping@ifbag.com (Th. Epping), Winfried.Hochstaettler@fernuni-hagen.de (W. Hochstättler).

0166-218X/$ - see front matter
doi:10.1016/j.dam.2005.05.033

of the cars, we assume that this order is determined by the objectives of other shops of the production line. Therefore, we will assume that the sequence of car bodies is fixed.

In [5], it is shown that the PPW is $\mathscr{NP}$-complete even when restricted to instances where only two different letters occur in the word, and also when restricted to instances where only two different colors are used. The main result of the present paper is the $\mathscr{APX}$-hardness of the PPW for an even smaller subset of its instances: instances with $|C|=2$ and $r_{\sigma i}=1$ for every $\sigma \in \Sigma$ and $i \in C$ (called 1-*regular* 2-*colored instances*). This answers one of the problems stated in [5]. Moreover, we show that the 1R2C-PPW is equivalent to the problem of finding a shortest circuit in a certain class of binary matroids. Consequences of this approach are polynomial time solution algorithms for specific instances and a duality theorem.

Problem 1 (*1R2C-PPW*). *Let w be a word of length $n=2|\Sigma|$ in which every letter of Σ occurs exactly twice, and let C be a set of colors with $|C|=2$. Find an assignment of the colors in C to the letters in w such that every letter of Σ is colored exactly once in each color of C and the number of color changes within w is minimized.*

2. Preliminaries

There is an insightful way (first introduced in [4]) to represent instances and solutions of the 1R2C-PPW, which is shown in Fig. 1. If the two occurrences of a letter $\sigma \in \Sigma$ appear at position x_σ and y_σ in the word, represent this letter pair with the interval $I_\sigma := [x_\sigma, y_\sigma]$ (drawn as a horizontal line segment). The set of all letter pair intervals of the instance is denoted by $\mathscr{I} = \{I_\sigma : \sigma \in \Sigma\}$. A solution is a set of real numbers corresponding to the color changes. Observe that a solution for the 1R2C-PPW is feasible if and only if there is an odd number of color changes within every interval I_σ.

In the remainder of this section, we collect the main definitions that are used within this paper. Our notation, however, is fairly standard.

2.1. Graphs

Let $G=(V, E)$ denote a simple, undirected graph. The *degree of a vertex* $v \in V$ is denoted by $d(v)$, and the *maximum degree of* G is defined by $\max\{d(v) : v \in V\}$. A *cubic graph* is a graph in which every vertex has degree 3. The edge incident with vertex i and j is denoted by ij or ji. Edges with a direction assigned to them are called *arcs*. An arc from i to j is denoted by (i, j). In this case, i is an *in-neighbor* of j and j is an *out-neighbor* of i. Directed graphs are denoted by $G=(V, A)$. If $G=(V, A)$ is a directed graph, $H=(V, E)$ is an undirected graph, $|A|=|E|$ and $(i, j) \in A \Rightarrow ij \in E$ then A is called an *orientation* of H. $G - v$ denotes the graph obtained from G by removing vertex v and all edges incident with v. $G + uv$ denotes the graph obtained by adding an edge between vertices $u, v \in V$. A *vertex cover* of a graph G is a set $S \subseteq V$ such that every edge is incident with at least one vertex of S.

2.2. Approximation algorithms

An algorithm for a minimization problem is called a ρ-*approximation algorithm* if for every instance with optimal solution value k it gives a solution with value at most ρk. Here, $\rho = 1 + \varepsilon$ with $\varepsilon > 0$ (for maximization problems the definition is similar). A *PTAS* for a minimization problem is a scheme to find polynomial time $(1+\varepsilon)$-approximation algorithms for *every* $\varepsilon > 0$. The problem class $\mathscr{APX}$ is the class of problems for which a polynomial time ρ-approximation algorithm exists for some $\rho > 1$ ($0 < \rho < 1$ for maximization problems). A problem is called $\mathscr{APX}$-hard if the existence of a PTAS for this problem would imply that for every problem in $\mathscr{APX}$ a PTAS exists. Moreover, Arora et al. [2] have shown that:

Fig. 1. The interval representation of a 1R2C-PPW instance and a solution.

Theorem 2. *If there exists a PTAS for an $\mathscr{APX}$-hard problem, then $\mathscr{P} = \mathscr{NP}$.*

3. Complexity results

In this section, the $\mathscr{APX}$-hardness of the 1R2C-PPW is proved using an L-reduction as introduced in [10]. We give a reduction from the following problem:

Problem 3 (*3GVC*). *Let $G = (V, E)$ be a cubic graph. Find a vertex cover $S \subseteq V$ of minimum cardinality.*

In [1], this problem is shown to be $\mathscr{APX}$-hard. First, we show that graphs with maximum degree 3 have an ordering of the vertices and an orientation of the edges satisfying the following criterion:

Criterion 4. *Let $G = (V, A)$ be a directed graph with a complete order $\prec$ on the vertices. We say that G and $\prec$ satisfy Criterion 4 if every vertex $v \in V$ has at most one in-neighbor u with $u \prec v$, and at most one in-neighbor w with $v \prec w$, and every vertex $v \in V$ with degree 2 has exactly one in-neighbor.*

Lemma 5. *Every graph G with maximum degree 3 has an ordering of the vertices and an orientation of the edges that satisfies Criterion 4. This vertex order and orientation can be found in polynomial time.*

Proof. We can assume $G=(V, E)$ to be connected. We proceed by induction on $|V|$. The assertion is true when $|V|=1$. Consider four cases:

(1) G has a leaf. This case is trivial.
(2) There is a degree 2 vertex v with neighbors u, w and $uw \notin E$. Now $G' = G - v + uw$ has an orientation and a vertex order as claimed. Assume w.l.o.g. $u \prec w$. Use this orientation and order for G, with v added in this order such that $u \prec v \prec w$, and the edges uv and vw oriented as a subdivision of the arc between u and w in G'. Now for u and w the situation in G is the same as in G' and v has one in-neighbor and degree 2, so this order and orientation satisfies Criterion 4.
(3) There is a degree 2 vertex v with neighbors u, w and $uw \in E$. The graph $G' = G - v$ has an orientation and a vertex order as claimed. Assume (u, w) is oriented this way. Then u has at most one in-neighbor x in G'. If there is such an x and $x \prec u$, insert v after u in this order, otherwise insert v before u in this order. Using the orientation of G' and adding (w, v) and (v, u) gives an orientation for G. Now w has one in-neighbor. u has two in-neighbors if and only if $d(u) = 3$, and in this case one in-neighbor appears before u in the order and one after. Otherwise u has one in-neighbor. v has one in-neighbor and degree 2. Therefore, this order and orientation satisfies Criterion 4.
(4) All vertices have degree 3. Choose a vertex v. Consider $G' = G - v$ and denote the neighbors of v by x, y and z. G' has an orientation and vertex order as required. In G', x, y and z all have degree 2. Let u be the in-neighbor of x and w be the in-neighbor of y. Observe that w.l.o.g. we can assume that either $u \prec x$ and $w \prec y$ or $x \prec u$ and $y \prec w$. In the first case, insert v in the order after x and y, and in the second case insert v before x and y. Using the orientation of G' and adding (v, x), (v, y) and (z, v) gives an orientation for G. Now x (y) has two in-neighbors, one ordered before x (y) and one after, and z and v both have one in-neighbor, so this order and orientation satisfies Criterion 4.

This describes a polynomial time algorithm to find a vertex order and orientation satisfying Criterion 4 for every graph with maximum degree 3. □

Theorem 6. *The 1R2C-PPW is $\mathscr{APX}$-hard.*

Proof. Let G be an instance of the 3GVC. Find an orientation of the edges and an order $\prec$ on the vertices that satisfies Criterion 4 (see Lemma 5). The resulting directed graph will also be denoted by $G = (V, A)$ with $V = \{1, \ldots, n\}$ and $u < v \Leftrightarrow u \prec v$. We use G to construct an instance for the 1R2C-PPW. This instance will consist of n blocks of letters,

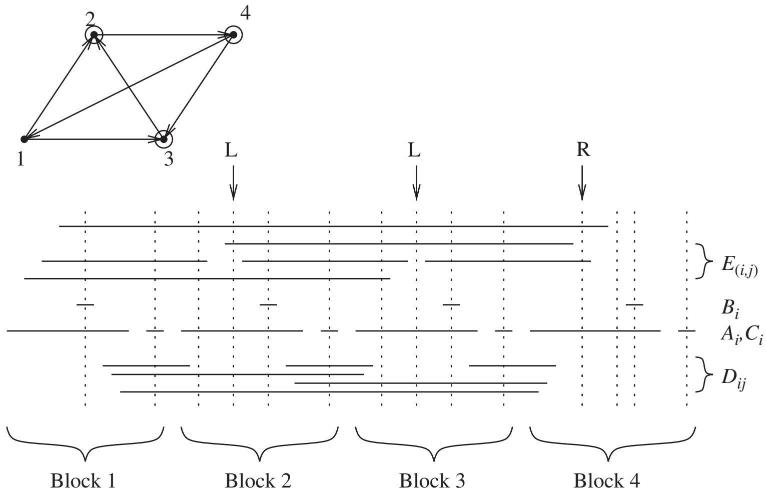

Fig. 2. A 3GVC instance and solution and the corresponding 1R2C-PPW instance and solution.

one block for every vertex. For every $i \in V$ we will introduce letters A_i, B_i and C_i that only appear in block i. For every arc $a \in A$ we will introduce letter E_a that appears once in both of the blocks corresponding to the end vertices. For every vertex pair $i, j \in V$ with $i < j$ we will introduce letter D_{ij}, that appears once in block i and once in block j.

Observe that because G satisfies Criterion 4, for every vertex v we can find a complete order $\prec_A$ on the arcs incident with v such that:

- if $a = (u, v)$ and $u < v$ then $a \prec_A b$ for every other arc b incident with v.
- if $a = (u, v)$ and $v < u$ then $b \prec_A a$ for every other arc b incident with v.

If vertex i is incident with arcs a, b and c, and the order $a \prec_A b \prec_A c$ satisfies the above properties, then we introduce the following block for v:

$$A_i D_{1i} D_{2i} \ldots D_{(i-1)i} E_a E_b E_c B_i B_i D_{i(i+1)} D_{i(i+2)} \ldots D_{in} A_i C_i C_i.$$

Now order the blocks left to right from 1 to n. Taken together this gives the word w for the 1R2C-PPW. See Fig. 2 for an example. In this example, in block 1, $E_{(1,3)}$ is placed before $E_{(1,2)}$. This is an arbitrary choice, both options give a valid construction. The same is true for the placement of $E_{(4,1)}$ and $E_{(4,3)}$ in block 4.

Claim 7. *If the 1R2C-PPW instance w has a solution with at most $2|V| + 2k$ color changes, then G has a vertex cover using at most k vertices.*

To prove this, we make the following observations that are true for every feasible solution of instance w.

(1) Because of the place of the B_i pair and the C_i pair in block i, in block i there are at least two color changes, for every i.
(2) If there is an arc $a \in A$ between i and j, then in any optimal solution of the 1R2C-PPW instance, either in block i or in block j there are at least four color changes.

We prove this last observation by contradiction. Suppose a is an arc between i and j, $i < j$ and in block i and in block j there are at most three color changes. Since one of the color changes in block i must be between $C_i C_i$, there are at most two color changes between the A_i pair. Since the number of color changes between a letter pair must be odd, there is only one color change between the A_i pair. The same holds for the A_j pair. This color change between the A_i (A_j) pair must be between the B_i (B_j) pair. In block i, E_a appears before the B_i pair, whereas D_{ij} appears after it. In block j, E_a and D_{ij} both appear before the B_j pair. This means that there is an odd number of color changes between the E_a pair if and only if there is an even number of color changes between the D_{ij} pair, a contradiction.

From the two observations above it follows that if there is a solution with at most $2|V| + 2k$ color changes, then G has a vertex cover with at most k vertices.

Claim 8. *If G has a vertex cover of cardinality k, there is a solution for the 1R2C-PPW instance w with* $2|V| + 2k$ *color changes.*

Let S be a vertex cover of G. If $i \notin S$, we apply only two color changes in block i: one between $B_i B_i$ and one between $C_i C_i$. If $i \in S$, we use four color changes in block i: between $B_i B_i$ and $C_i C_i$ but also between the consecutive letters $D_{(i-1)i} E_a$ and between the consecutive letters $E_c B_i$. If there is an arc $a \in A$ with end vertices $i \in S$ and $j \in S$ $(i < j)$, we have to move one of the latter two color changes. If a is directed towards i, then $E_a B_i$ is a consecutive letter combination in w and we move the color change one position to the left (between $E_b E_a$ for some b). If a is directed towards j, then $D_{(j-1)j} E_a$ is a consecutive letter combination in w, and we move the color change one position to the right (between $E_a E_b$ for some b). This ensures that for every a between i and j with $i < j$, there is either one color change in block i between E_a and B_i and no color change in block j between $D_{(j-1)j}$ and E_a or there is no color change in block i between E_a and B_i and one color change in block j between $D_{(j-1)j}$ and E_a. In Fig. 2 an example of such a 1R2C-PPW solution corresponding to a vertex cover is shown. The color changes marked with 'L' ('R') are the color changes moved to the left (right) in this last step. We prove that this method gives a feasible solution:

(1) Let $i < j$. In block i, there is one color change to the right of D_{ij}: this is the color change between $C_i C_i$. In block j, there is no color change to the left of D_{ij}. In every block between i and j there is an even number of color changes. There are no color changes between blocks. So there is an odd number of color changes between the two D_{ij} letters for every $i < j$.
(2) Let a be an arc between i and j, $i < j$. We know that either there is a color change in block i between E_a and the first B_i, or there is a color change in block j between D_{ij} and E_a, but not both. In addition, there is exactly one color change in block i between the first B_i and D_{ij} (this is the color change between $B_i B_i$). So there is an odd number of color changes between the E_a pair if there is an odd number of color changes between the D_{ij} pair. This was shown to be true, so for every a there is an odd number of color changes between the two E_a letters.
(3) Between every A_i pair, there are either one or three color changes.
(4) Between $B_i B_i$ and between $C_i C_i$ there is a color change.

We conclude that this is a feasible solution with exactly $2|V| + 2|S|$ color changes.

Now suppose a $(1 + \varepsilon)$-approximation algorithm for the 1R2C-PPW exists. We use this to construct a $(1 + 3\varepsilon)$-approximation algorithm for the 3GVC. Let $G = (V, E)$ be a 3GVC instance with minimum vertex cover S. We use the above transformation to construct a 1R2C-PPW instance w, with minimum number of color changes $m = 2|S| + 2|V|$. Use the approximation algorithm to find a solution with $k \leqslant (1 + \varepsilon)m$ color changes. This gives a vertex cover of cardinality $k' \leqslant k/2 - |V|$. Observe that since G is cubic, $|S| \geqslant \frac{1}{3}|E| = \frac{1}{2}|V|$, so $3|S| \geqslant |V| + |S|$. Therefore:

$$\frac{k' - |S|}{|S|} \leqslant \frac{k/2 - |V| - (m/2 - |V|)}{|S|} \leqslant 3\frac{k/2 - m/2}{|V| + |S|} = 3\frac{k - m}{m} \leqslant 3\varepsilon.$$

So a PTAS for the 1R2C-PPW would give a PTAS for the 3GVC. Therefore, the 1R2C-PPW is also $\mathscr{APX}$-hard. □

Observe that $\mathscr{NP}$-completeness for the corresponding decision problem follows from the $\mathscr{APX}$-hardness.

4. Lower and upper bounds

In this section, we will consider the interval representation $\mathscr{I}$ corresponding to an instance w. If $\mathscr{I}$ is a set of intervals, a subset of pairwise disjoint intervals in $\mathscr{I}$ is called an *independent set* of $\mathscr{I}$. A maximum independent set in the interval set $\mathscr{I}$ of a 1R2C-PPW instance w yields a straightforward lower bound on the minimum number of color changes for w. We will improve this lower bound in two ways. Given an instance $\mathscr{I} = \{I_\sigma : \sigma \in \Sigma\}$, we call the symmetric difference $S := I_1 \Delta I_2 \Delta \ldots \Delta I_{2k+1}$, $k \geqslant 0$ of an odd subset of the intervals an *odd row sum*. An *odd row sum packing* $\{S_1, \ldots, S_l\}$ is a set of pairwise disjoint odd row sums.

Proposition 9. *If $\{S_1, \ldots, S_l\}$ is an odd row sum packing of an instance $\mathscr{I}$ of 1R2C-PPW, then any feasible solution of 1R2C-PPW requires at least l color changes.*

Proof. As the row sums are pairwise disjoint it suffices to show that there is at least one color change in any odd row sum $I_1 \Delta I_2 \Delta \ldots \Delta I_{2k+1}$. Let c_σ denote the number of color changes in I_σ. In a feasible solution, c_σ is odd for any σ, and since the number of intervals is odd, $\sum_{\sigma=1}^{2k+1} c_\sigma$ is odd. Another way to count this sum is as follows: for every color change, add the number of intervals in $I_1, \ldots, I_{2k+1}$ that contain the color change. Now suppose there are no color changes in $I_1 \Delta I_2 \Delta \ldots \Delta I_{2k+1}$. Then all color changes are contained in an even number of intervals from this set, so the sum is even, a contradiction. □

For reasons that we will comment on later let us call the maximum size of a disjoint odd row sum packing the *MaxFlow-MinCut bound*. We do not know the computational complexity of this bound but we expect it to be hard.

In the following we introduce a slightly (but strictly) weaker lower bound that is computable in polynomial time.

Consider the partial order on $\mathscr{I}$ given by proper containment, and the corresponding comparability graph $G_{\mathscr{I}} = (\mathscr{I}, A)$, where $(I_\sigma, I_\tau) \in A \Leftrightarrow I_\tau \subset I_\sigma$. $N^+(I_\sigma)$ denotes the set of out-neighbors of I_σ in $G_{\mathscr{I}}$, which is the set of intervals contained in I_σ.

Lemma 10. *Let $I_1, \ldots, I_k$ be a set of intervals with the following properties*:

- *For every I_σ and I_τ: $I_\sigma \subset I_\tau$, $I_\tau \subset I_\sigma$ or $I_\sigma \cap I_\tau = \emptyset$.*
- *For every I_σ, $|N^+(I_\sigma)|$ is even.*

Then for an 1R2C-PPW instance containing these intervals, an odd row sum packing of size k exists, and therefore at least k color changes are needed.

Proof. Let $\mathscr{I} = \{I_1, \ldots, I_k\}$ be a set of intervals with these properties.

We first prove by induction over k that if there is one interval that contains all others, then for $\mathscr{I}$ an odd row sum packing of size k exists. So w.l.o.g. for every $2 \leqslant \tau \leqslant k$, $I_\tau \subset I_1$.

Now let J be the set of vertices in $G_{\mathscr{I}}$ with no in-neighbor other than I_1. By the first property, every interval not in J, other than I_1, is contained in exactly one interval of J. We may assume that $J = \{I_2, \ldots, I_l\}$. For every $I_\tau \in J$, by induction there exists an odd row sum packing of $N^+(I_\tau) \cup \{I_\tau\}$ of size $|N^+(I_\tau)| + 1$. Since $I_2, \ldots, I_l$ are pairwise disjoint, and every interval is contained in one of them, taken together this is an odd row sum packing of size $k - 1$. Since $|N^+(I_1)|$ is even, and for every $I_\tau \in J$, $|N^+(I_\tau)|$ is even, $|J|$ must be even too. So $I_1 \Delta I_2 \Delta \ldots \Delta I_l$ is also an odd row sum, disjoint with all other row sums. This is an odd row sum packing of $\mathscr{I}$ of size k.

If no interval contains all others, then w.l.o.g. $\{I_1, \ldots, I_l\}$ is the set of vertices in $G_{\mathscr{I}}$ with no in-neighbor. Now we have shown that for every $\tau \leqslant l$ there exists an odd row sum packing of $\{I_\tau\} \cup N^+(I_\tau)$ of size $|N^+(I_\tau)| + 1$. Taken together, this leads to an odd row sum packing of I of size k. □

We will call a set of intervals with the two properties from Lemma 10 a *non-overlapping set*.

Now we will construct a polynomial time algorithm that calculates the size of a maximum non-overlapping set for any 1R2C-PPW instance with interval set $\mathscr{I}$. The algorithm depicted in Fig. 3 assigns weights $w(I_\sigma)$ to each interval I_σ which correspond to the minimum number of color changes needed in I_σ as calculated by the above bound. A vertex $I_\sigma \in \mathscr{I}$ is called *processed* if a value $w(I_\sigma)$ is assigned to it. It is called *processable* if all out-neighbors of I_σ are processed. For any interval set $\mathscr{I}$, an easy linear time algorithm exists that finds a independent set in $\mathscr{I}$, such that the sum of weights of these intervals is maximized (see also [7]). The subroutine maxWIS($\mathscr{I}$) returns the weight of such a set for interval set $\mathscr{I}$. Instead of maxWIS($N^+(I_\sigma)$) we will write maxWIS(I_σ).

Theorem 11. *For input $\mathscr{I}$, the value calculated by the algorithm in Fig. 3 is equal to the size of a maximum non-overlapping set in $\mathscr{I}$, and the time complexity is $\mathscr{O}(|A|)$, where A is the arc set of graph $G_{\mathscr{I}}$.*

INPUT: The interval set $\mathcal{I}$ of a 1R2C-PPW instance w
OUTPUT: A lower bound on the minimum number of color changes for w

Construct the comparability graph $G_{\mathcal{I}} = (\mathcal{I}, A)$ of $\mathcal{I}$
Assign a weight $w(I_\sigma) = 1$ to all vertices $I_\sigma \in \mathcal{I}$ with no out-neighbor

While there exists an unprocessed vertex in $\mathcal{I}$
  Choose a processable vertex $I_\sigma \in \mathcal{I}$ and set $w(I_\sigma) = \texttt{maxWIS}(I_\sigma)$
  If $w(I_\sigma)$ is even, set $w(I_\sigma) \leftarrow w(I_\sigma) + 1$

Return $\texttt{maxWIS}(\mathcal{I})$

Fig. 3. An algorithm for the computation of a lower bound.

Proof. We prove by induction that the value $w(I_\sigma)$ is equal to the size of a maximum non-overlapping set in $\{I_\sigma\} \cup N^+(I_\sigma)$.

If I_σ has no out-neighbors in $G_{\mathscr{I}}$, then the statement is true. Otherwise, let $\{I_1, \ldots, I_k\}$ be a maximum non-overlapping subset of $N^+(I_\sigma)$ and w.l.o.g. let $\{I_1, \ldots, I_l\}$ again be the subset of intervals that are not contained in another interval of $\{I_1, \ldots, I_k\}$. $\texttt{maxWIS}(I_\sigma) \geqslant \sum_{\tau=1}^{l} w(I_\tau) = k$ by induction and choice of $\{I_1, \ldots, I_l\}$. Also, if $\{I_1, \ldots, I_l\}$ is an independent subset of $N^+(I_\sigma)$ such that $\texttt{maxWIS}(I_\sigma) = \sum_{\tau=1}^{l} w(I_\tau)$, then by induction a non-overlapping subset of $N^+(I_\sigma)$ can be found of this size. Finally, a non-overlapping subset of $N^+(I_\sigma) \cup \{I_\sigma\}$ of size $\texttt{maxWIS}(I_\sigma) + 1 = w(I_\sigma)$ can be found if and only if $\texttt{maxWIS}(I_\sigma)$ is even.

In the computation of this lower bound, the subroutine $\texttt{maxWIS}(I_\sigma)$ is of complexity $\mathcal{O}(|N^+(I_\sigma)|)$. If $I_\tau \in N^+(I_\sigma)$, this corresponds to arc $I_\sigma I_\tau$ of $G_{\mathscr{I}}$, so every arc of $G_{\mathscr{I}}$ is considered exactly once, and Algorithm 3 requires a time complexity of $\mathcal{O}(|A|)$. Note, however, that $|A| = \mathcal{O}(|\Sigma|^2)$ for some instances. □

By Lemma 10, the output of the algorithm is a lower bound for the number of color changes, and by Theorem 11 it is even the best possible lower bound of this type. It can be checked that the value of this lower bound is always less than twice the value of the trivial independent set bound.

The instance in Fig. 1 illustrates the differences between the two lower bounds. The MaxFlow-MinCut bound is tight with a packing of the four odd row sums I_B, I_E, $I_A \Delta I_B \Delta I_C$, $I_C \Delta I_D \Delta I_E$ but $\texttt{maxWIS}(r) = 3$.

There is an immediate general upper bound.

Theorem 12. *Each 1R2C-PPW instance using an alphabet Σ has a solution with at most $|\Sigma|$ color changes.*

Proof. This bound is guaranteed by the trivial algorithm that colors the letters from left to right, only switching colors if the second letter would receive the same color as the first. □

$|\Sigma|$ color changes are required for words $w = \sigma_1\sigma_1\sigma_2\sigma_2 \ldots \sigma_{|\Sigma|}\sigma_{|\Sigma|}$ which contain the two occurrences of each letter $\sigma_i \in \Sigma$ consecutively. Thus, there are cases where the bound is strict. On the other hand, this simple approach also requires $|\Sigma|$ color changes for the word

$$\tilde{w} = \sigma_1\sigma_2 \ldots \sigma_{|\Sigma|/2}\sigma_{|\Sigma|/2}\sigma_{|\Sigma|/2+1} \ldots \sigma_{|\Sigma|}\sigma_{|\Sigma|}\sigma_1\sigma_{|\Sigma|-1}\sigma_2 \ldots \sigma_{|\Sigma|/2-1}\sigma_{|\Sigma|/2+1},$$

where $|\Sigma|$ is required to be even and which could be colored with three color changes.

5. The matroid connection

Despite the complexity results of Section 3, known results from matroid theory imply that some instances of the 1R2C-PPW can be solved in polynomial time.

In the following we assume basic familiarity with matroid theory, see e.g. [9,13].

We will consider a certain binary matroid, i.e. a matroid having a representation $M = M[A]$ by a matrix A over GF(2). In our context this arises as follows:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \end{pmatrix} \begin{matrix} \mathrm{A} \\ \mathrm{B} \\ \mathrm{C} \\ \mathrm{D} \\ \mathrm{E} \end{matrix}$$

Fig. 4. A 1R2C-PPW instance and the associated matrix A.

Definition 13. Let w be a 1R2C-PPW instance. The $(|\Sigma| \times (n-1))$-matrix $A = (a_{\sigma i})$ is defined with respect to the interval set $\mathscr{I} = \{[x_\sigma, y_\sigma] : \sigma \in \Sigma\}$ of w by

$$a_{\sigma i} := \begin{cases} 1 & \text{if } x_\sigma \leqslant i < y_\sigma, \\ 0 & \text{otherwise.} \end{cases}$$

This way, every column of A corresponds to a possible position of a color change in w (see Fig. 4 for an example).

Recall that there must be an odd number of color changes between the two occurrences of any letter $\sigma \in \Sigma$ in w. Thus, any solution $x \in \{0,1\}^n$ of $(A, \vec{1})x \equiv \vec{0} \bmod 2$ with $x_n = 1$ corresponds to a feasible solution of the 1R2C-PPW (we denote by $\vec{1}$ ($\vec{0}$) the vector of all ones (zeros) of appropriate dimension).

Interpreting each column of $(A, \vec{1})$ as an element of a binary vector matroid $M[A, \vec{1}]$, we are faced with the problem of finding a shortest circuit in $M[A, \vec{1}]$ that contains the element $\vec{1}$. This leads to the following problem:

Problem 14. *Let $M = (E, \mathscr{I})$ be a binary matroid with a special element l. Find a minimum size circuit C of M containing l.*

Problem 14 is a simplified version of the *binary clutter problem*. Observe that it is $\mathscr{NP}$-hard, since we showed that 1R2C-PPW can be reduced to it. See also [13].

If $M = M[A]$ is a binary vector matroid represented by a matrix A over GF(2), then M is called *regular* if the entries of A can be signed so that A is represented by a totally unimodular matrix over $\mathbb{R}$. The matrix A resulting from any 1R2C-PPW instance is totally unimodular, because A has the consecutive ones property for rows (see [6]).

If the one-point extension of $M[A]$, $M[A, \vec{1}]$ is also regular, then our problem can be solved by linear programming [13]. Another characterization of regular matroids is the following by Tutte (see e.g. [9]): a binary matroid is regular if and only if it excludes the Fano matroid F_7 and its dual as minors. This shows that the following result is stronger: our problem can be solved by matroid decomposition if $M[A, \vec{1}]$ does not contain an F_7 or if it does not contain the F_7 dual [3,9,13]. Note that the existence of an F_7 minor or its dual within a matroid can be checked in polynomial time. Unfortunately, some enumeration indicates that even most instances using only 10 letters already have an F_7 as well as its dual as a minor.

Since regular matroids can be decomposed from graphic and co-graphic matroids and a particular extra matroid [12], we consider the cases where $M[A]$ is graphic resp. co-graphic.

For the graphic case, we use the following method: add an extra row of all zeros to A, and apply elementary row operations such that the resulting matrix is the incidence matrix of a graph G. These operations do not change the structure of the matroid. Now instead apply these operations to $(A, \vec{1})$. In the resulting matrix, the last column corresponds to a partition of the vertices in sets T and $\overline{T}$. A circuit containing the new element can be seen to correspond to the following problem: find a selection of edges M such that a vertex v is incident with an odd number of edges from M if and only if $v \in T$. This is the *T-join* problem for graphs, which can be solved in polynomial time [8].

If $M[A]$ is co-graphic, we can also transform our problem to a graph problem in a very similar way, but unfortunately the resulting problem contains the $\mathscr{NP}$-hard Max-Cut problem as a special case.

5.1. MaxFlow-MinCut duality

Our problem of finding a minimum circuit containing a given element l can be checked to have the following dual problem: find a maximum set of co-circuits that all contain l but are pairwise disjoint for the other elements (a disjoint packing of co-circuits containing l). In the dual matroid, these two problems correspond to two more well-known problems: minimum cocircuit (minimum cut) resp. maximum flow. The name MaxFlow-MinCut bound is derived from these problems.

By a famous result of Seymour [11] in some cases we have *strong duality* here, i.e. the MaxFlow-MinCut bound is tight, if $M[A, \vec{1}]$ does not contain an F_7-minor. We refer to [3,4] for details.

Theorem 15. *If $M[A, \vec{1}]$ has no F_7 minor, the minimum number of color changes for a 1R2C-PPW instance equals the value of the MaxFlow-MinCut bound.*

Example 16. The minimum number of color changes for the instance shown in Fig. 4 is 3, but $M[A, \vec{1}]$ contains an F_7 minor and a maximum disjoint odd row sum packing has size 2.

6. Summary and open problems

We have answered one open problem stated in [5]. It turns out that the 1R2C-PPW, a very restricted version of the PPW, is still $\mathcal{NP}$-complete, and even $\mathcal{APX}$-hard. This brings the next open question in [5] into focus: whether there is a constant factor approximation algorithm for the 1R2C-PPW. One possibility to answer this question to the affirmative would be if one could show, that the lower bound from Theorem 11 yields at least a constant fraction of the optimum value.

Furthermore, we have shown that the 1R2C-PPW is equivalent to the problem of finding a shortest circuit in a certain class of binary matroids, which allows the solution of some specific instances within polynomial time. In this context it would be interesting if, for instances without F_7-minor not only the value of a disjoint odd row sum packing, but also the disjoint odd row sum packing itself can be computed efficiently. We do not know how to do this even for regular matroids $M[A, \vec{1}]$.

References

[1] P. Alimonti, V. Kann, Some $\mathcal{APX}$-completeness results for cubic graphs, Theoret. Comput. Sci. 237 (1–2) (2000) 123–134.
[2] S. Arora, C. Lund, R. Motwani, M. Sudan, M. Szegedy, Proof verification and the hardness of approximation problems, J. ACM 45 (3) (1998) 501–555.
[3] Th. Epping, Color sequencing, Ph.D. Thesis, Brandenburgische Technische Universität, Cottbus, 2004.
[4] Th. Epping, W. Hochstättler, M.E. Lübbecke, Maxflow-mincut duality for a paint shop problem, in: U. Leopold-Wildburger, F. Rendl, G. Wäscher (Eds.), Operations Research Proceedings 2002, Springer, Berlin, 2003, pp. 377–382.
[5] Th. Epping, W. Hochstättler, P. Oertel, Complexity results on a paint shop problem, Discrete Appl. Math. 136 (2004) 217–226.
[6] R.L. Graham, M. Grötschel, L. Lovasz (Eds.), Handbook of Combinatorics, vol. II, Elsevier, Amsterdam, 1995.
[7] J.Y. Hsiao, C.Y. Tang, R.S. Chang, An efficient algorithm for finding a maximum weight 2-independent set on interval graphs, Inform. Process. Lett. 43 (1992) 229–235.
[8] L. Lovász, M.D. Plummer, Matching Theory, Annals of Discrete Mathematics, vol. 29, North-Holland, Amsterdam, 1986.
[9] J.G. Oxley, Matroid Theory, Oxford University Press, Oxford, 1997.
[10] C.H. Papadimitriou, M. Yannakakis, Optimization, approximation and complexity classes, J. Comput. System Sci. 43 (3) (1991) 425–440.
[11] P.D. Seymour, The matroids with the max-flow min-cut property, J. Combin. Theory B (23) (1977) 189–222.
[12] P.D. Seymour, Decomposition of regular matroids, J. Combin. Theory B 28 (1980) 305–359.
[13] K. Truemper, Matroid Decomposition, Academic Press, 1992, ⟨http://emis.math.metu.edu.tr/monographs/md/⟩.