

# Improved bounds for the binary paint shop problem

J. Hančl, A. Kabela, M. Opler, J. Sosnovec, R. Šámal, P. Valtr

June 15, 2020

## Abstract

We improve bounds for the binary paint shop problem posed by Meunier and Neveu [Computing solutions of the paintshop-necklace problem. *Comput. Oper. Res.* 39, 11 (2012), 2666–2678]. In particular, we disprove their conjectured upper bound for the number of color changes.

## 1 Introduction

A *double occurrence word*  $w$  is a word (sequence of letters) in which each of its letters occurs exactly twice. *Legal 2-coloring* of  $w$  is a coloring of individual letters such that each letter occurs once red and once blue. We let  $W_n$  denote the set of all double occurrence words with letters  $A_1, \dots, A_n$ , each occurring twice. We denote the first, resp. the second, occurrence of a letter  $A$  by  $\underline{A}$ , resp  $\bar{A}$ .

Our goal is to find for a double occurrence word  $w$  a legal 2-coloring of  $w$  with minimal number of color changes – neighboring letters of different color. We use  $\gamma(w)$  to denote this quantity.

To state this more formally, a legal coloring of  $w = w_1w_2\dots w_{2n} \in W_n$  is a mapping  $c := \{1, \dots, 2n\} \rightarrow \{Red, Blue\}$  such that if  $1 \leq i < j \leq 2n$  and  $w_i = w_j$  then  $c(i) \neq c(j)$ . We define

$$\gamma(w) = \min_{\text{legal } c} |\{i < 2n : c(i) \neq c(i+1)\}|.$$

Throughout the paper we use binary colors  $0 = Red$  and  $1 = Blue$ .

The motivation for the definition (and for the “paint shop” in the title of the paper) is the following: Imagine a line of cars in front of a paint shop factory, suppose that there are just two cars of each type, and that we need to paint one of them blue and the other red. To decrease cost, we want to minimize the number of color changes. This problem was introduced by Epping et al. [4] under the name binary paint shop problem (non-binary case requires to use more than two colors).

Another guise of the same problem is necklace splitting: two thieves stole a necklace with  $2n$  gems stones, two of each of  $n$  types. As the price of each of the

gems is unknown, they want to cut the necklace, so that each of the thieves can get one of each type of the gems. Alon’s necklace-splitting theorem [1] gives an upper bound for this (and for much more general) version. Translated to our setting, this results says that  $\gamma(w) \leq n$  for every  $w \in W_n$ . Easily, this is tight for  $w = A_1 A_1 A_2 A_2 \dots A_n A_n$ .

It is known [4] that deciding whether  $\gamma(w) \leq k$  for a given  $w$  and  $k$  is NP-complete. Moreover,  $\gamma$  is APX-hard [3, 6]. Thus, the study of various heuristics is in order. A natural way to evaluate them is to look at the behaviour on random instances of  $W_n$ .

This motivates the following notion, which will be of our main interest: We let  $\gamma_n$  be the expectation of  $\gamma(w)$  when  $w$  is a random element of  $W_n$ ,

$$\gamma_n := \mathbb{E}_n \gamma$$

where the right-hand side of the above equation means  $\mathbb{E}_{w \in W_n} \gamma(w)$ . To exemplify,  $W_2$  consists of the following words:  $AABB$ ,  $ABAB$ ,  $ABBA$  (up to renaming the letters). As  $\gamma(AABB) = 2$ ,  $\gamma(ABAB) = \gamma(ABBA) = 1$ , we have  $\gamma_2 = 4/3$ .

Andres and Hochstättler [2] describe two heuristics, greedy and recursive greedy. Use  $g(w)$  and  $rg(w)$  for the resulting number of color changes. They prove that

$$\begin{aligned} \mathbb{E}_n g &= \frac{1}{2}n + O(1) \\ \mathbb{E}_n rg &= \frac{2}{5}n + \frac{7}{10} \end{aligned}$$

Consequently,  $\gamma_n \leq \frac{2}{5}n + \frac{7}{10}$ . Meunier and Neveu [5] say: “We were not able to propose an interesting lower bound on  $\mathbb{E}_n \gamma$ , but we conjecture that  $\mathbb{E}_n \gamma = o(n)$ .”

In the second section, we disprove this conjecture, proving  $\gamma_n \geq 0.214n - o(n)$ . In the third section we slightly improve the recursive greedy heuristics, getting an upper bound of  $(0.4 - \varepsilon)n$  for any  $\varepsilon < 10^{-6}$ . In the fourth section we describe a new simple heuristics “recursive star greedy” that probably satisfies  $\mathbb{E}_n rsg \leq 0.361n$ , although we are not able to prove that. Finally, in the fifth section we use Azuma inequality to show that the value of  $\gamma(w)$  is concentrated on a short interval.

## 2 Lower bounds

We could not find in the literature any mention of lower bounds for  $\gamma_n$ . As a warm-up, we present a very simple lower bound using random interval graphs. Scheinerman [7] introduced a model of random interval graphs that is closely related to our problem. He starts by choosing at random, uniformly and independently,  $x_i, y_i \in [0, 1]$  for  $i = 1, \dots, n$ . Then he creates an interval graph  $G$  of intervals  $[x_i, y_i]$ . Explicitly, the vertices of  $G$  are  $[n] = \{1, \dots, n\}$  and  $ij$  is an edge of  $G$  whenever intervals  $[x_i, y_i]$  and  $[x_j, y_j]$  intersect.

With probability 1, no two of the selected  $2n$  real number coincide. Thus, we may to our random selection of  $x_i$ 's and  $y_i$ 's also assign a double-occurrence word: we sort  $X = \{x_i, y_i : i \in [n]\}$  and replace both  $x_i$  and  $y_i$  by  $A_i$  (the  $i$ -th letter). If  $w$  is the resulting word, we call  $G$  the interval graph associated with  $w$  and use  $IG(w)$  to denote it.

It is easy to see that  $\gamma(w)$  is equal to the smallest  $k$  such that there is a set  $S \subset [0, 1]$  of size  $k$ , such that for every  $i$  the size of  $[x_i, y_i] \cap S$  is odd. This leads to an easy lower bound for  $\gamma(w)$  in terms of properties of random interval graphs.

**Claim 2.1.**  $\gamma(w) \geq \alpha(IG(w))$

As Scheinerman [7] proves that expectation of  $\alpha(G)$  is at least  $C\sqrt{n}$ , we have the following corollary.

**Corollary 2.2.** *There is a  $C > 0$  such that  $\gamma_n \geq C\sqrt{n}$ .*

Next, we provide a linear lower bound on  $\gamma_n$ , disproving the conjecture of Meunier and Neveu [5].

**Theorem 2.3.**  $\gamma_n \geq 0.214n - o(n)$

*Proof.* We let  $w$  be a uniformly random element of  $W_n$ . We will show, for an appropriate choice of  $k$  and  $p$ , that

$$\Pr[\gamma(w) \leq k] \leq p.$$

This will prove that

$$\gamma_n = \mathbb{E} \gamma(w) \geq (1 - p)k. \tag{1}$$

eq:goal

Let  $C_n^{\leq k}$  be the set of all possible binary colorings of  $1, \dots, 2n$  using first red  $n$  times and blue  $n$  times, that have at most  $k$  color changes. We use union bound and straightforward estimates:

$$\begin{aligned} \Pr[\gamma(w) \leq k] &= \Pr[w \text{ has a legal coloring in } C_n^{\leq k}] \\ &\leq \sum_{C \in C_n^{\leq k}} \Pr[C \text{ is legal for } w] \\ &= \sum_{C \in C_n^{\leq k}} \frac{n!^2}{(2n)!/2^n} \leq \sum_{l=0}^k \binom{2n-1}{l} \frac{2^n}{\binom{2n}{n}} \\ &\leq \frac{\sqrt{4n}}{2^n} \sum_{l=0}^k \binom{2n}{l} \leq \frac{\sqrt{4n}}{2^n} \left( \frac{e \cdot 2n}{k} \right)^k \end{aligned}$$

So we may let  $p$  be equal to the last line and we have proved (1).

Put  $a = k/n$ . Then  $p = \sqrt{4n} \left( \left( \frac{2e}{a} \right)^a \frac{1}{2} \right)^n$ . Thus if  $\left( \frac{2e}{a} \right)^a \frac{1}{2} < 1$ , then we have  $p = o(1)$  and, thus,

$$\mathbb{E} \gamma(w) \geq (1 - o(1))an.$$

Numerical computation shows, that this works whenever  $a < 0.214\dots$ . This finishes the proof.  $\square$

### 3 Small improvement of the upper bound

In [2] the *recursive greedy* (RG) heuristics was used to prove the upper bound  $\gamma_n \leq 0.4n + O(1)$ . We slightly improve that result adding a linear term. Main interest of this is showing that  $0.4n$  is not the final answer.

In the original proof the following recursive greedy heuristics was used. To obtain a coloring

$$c : [2n] = \{1, 2, \dots, 2n\} \rightarrow \{0, 1\}$$

of the double occurrence word  $w \in W_n$ , we first omit the both occurrences of the first letter of  $w$ , say  $A$ . Then we color the shorter word recursively, add letter  $A$  back and color it in the best possible way. Before we give more formal description, recall that we denote the first, resp. second, occurrence of a letter  $A$  by  $\underline{A}$ , resp.  $\bar{A}$ .

**Recursive greedy algorithm:** Let 1 and  $j$  be the position of  $\underline{A}$  and  $\bar{A}$  in a word  $w \in W_n$  and  $c' : [2n-2] \rightarrow \{0, 1\}$  be the coloring of  $w' \in W_{n-1}$ . We define the coloring  $c$  of  $w$  such that it coincides with  $c'$  on the letters of  $w'$ , hence only colors  $c(\underline{A}) = c(1)$  and  $c(\bar{A}) = c(j)$  are to be determined. Let  $N_{c'}(\bar{A}) = \{c'(j-2), c'(j-1)\}$  be the multiset of colors that  $c'$  uses in the neighborhood of  $\bar{A}$ .

- (a) If  $j = 2$  then  $c(\underline{A}) = 1 - c'(1)$ .
- (b) If  $j = 2n$  and  $n > 1$  then  $c(\underline{A}) = 1$ .
- (c) If  $j \in [3, 2n-1]$  then

$$c(\underline{A}) = \begin{cases} c'(1) & \text{if } N_{c'}(\bar{A}) \text{ contains } 1 - c'(1), \\ 1 - c'(1) & \text{if } N_{c'}(\bar{A}) = \{c'(1), c'(1)\}. \end{cases}$$

and color  $\bar{A}$  accordingly.

Note that in our description of RG we reversed the input of the original algorithm and colored first letter first (in [2] they colored last letter first). We use the following observation.

1:changes

**Lemma 3.1.** *Let  $n \geq 1$  and  $w \in W_n$  be a fixed word colored by RG. Then the number of neighboring pairs in  $w$  that are colored with color  $c$  is at least  $\lfloor (n-1)/2 \rfloor$ .*

*Proof.* We prove by induction that  $rg(w)$ , i.e. the number of color changes in the coloring of  $w$  obtained from RG, is at most  $n$ . This clearly holds for  $n = 1$  and in every recursive step RG adds two letters and at most one color change. Since every color colors the same number of letters we conclude that half of the remaining  $n-1$  neighboring pairs have the color  $c$ .  $\square$

As proved in [2], this construction results in  $\mathbb{E}_n rg \leq 0.4n + O(1)$ . However, we can improve the final coloring of RG. We find a set  $V$  of pairs of letters of  $w$  such that recoloring any pair of  $V$  decrease the number of color changes. Moreover, we prove that  $|V|$  is linear in  $n$ , hence we significantly (resp. linearly) improve the output of RG by switching the colors in  $V$ .

thm:epsilon

**Theorem 3.2.** For  $\varepsilon \approx 2 \times 10^{-6}$  and sufficiently large  $n$  we have

$$\gamma_n < \left(\frac{2}{5} - \varepsilon\right) n.$$

*Proof.* Let (for simplicity)  $n$  be even and  $w \in W_n$ . Denote the letters of  $w$  as  $A_1, A_2, \dots, A_n$  such that the first occurrence of  $A_j$  lies before the first occurrence of  $A_i$  whenever  $i < j$ , i.e. the letters are ordered in descending order by their first occurrences. We denote by  $\tau_k(w) \in W_k$  the word obtained from  $w$  by removing every letter  $A_i$  for  $i > k$ . We split RG into two stages. In the first stage we color the word  $\tau_{n/2}(w)$ , in the second stage we extend that coloring onto  $w$ ; in both stages we color  $n/2$  pairs of letters.

Let us fix  $w' \in W_{n/2}$  and denote by  $c' : [n] \rightarrow \{0, 1\}$  the coloring in the output of RG for  $w'$ . We set

$$T = \{t \in [n-1] : c'(t) = c'(t+1) = 0\}.$$

to be the set of positions of two consecutive letters colored by 0. We know that  $T$  contains at least  $\lfloor (n-2)/4 \rfloor$  elements due to Lemma 3.1. We refer to the elements of  $T$  as monochromatic pairs. We consider the probability space of all words  $w \in W_n$  that were build from  $w'$ ; that is  $\tau_{n/2}(w) = w'$ .

Let  $t \in T$ . We estimate the probability  $p = p(t)$  such that:

- S1. In the second stage of RG there are exactly two letters, say  $C$  and  $D$ , inserted in between the monochromatic pair  $t$ . Moreover, no letter is ever inserted around  $C$  and  $D$  (other than the one in the next step).
- S2. First occurrences  $C$  and  $D$  are colored so that there are two color changes around.

Let  $U_{i,j}$  be the set of words in  $W_n$  such that S1, S2 holds and letter  $C$  and  $D$  is the  $i$ -th and  $j$ -th added letter in stage two for indices  $i < j$ . We denote by  $p_{i,j}$  the probability that  $w \in U_{i,j}$  which allows us to express  $p$  as

$$p = \sum_{1 < i < j < n/2} \Pr[w \in U_{i,j}] = \sum_{1 < i < j < n/2} p_{i,j}.$$

We remark that if  $j = i+1$  then the first occurrences of  $C$  and  $D$  cannot have two changes around them and  $p_{i,j} = 0$ . Let us therefore assume that  $j - i \geq 2$ . We let  $U_{i,j}^k$  be the set of words of length  $n/2 + k$  that can be extended to some word  $w \in U_{i,j}$ , that is

$$U_{i,j}^k = \{\tau_{n/2+k}(w) \mid w \in U_{i,j}\}.$$

Observe that  $U_{i,j}^{n/2}$  is precisely the set  $U_{i,j}$  while  $U_{i,j}^0$  contains only the word  $w'$ . We define the probability that after adding  $k$  letters in the second stage we did not violate any of the desired properties conditioned by the fact that the same holds after adding  $k-1$  letters.

$$p_{i,j}^k = \Pr[\tau_{n/2+k}(w) \in U_{i,j}^k \mid \tau_{n/2+k-1}(w) \in U_{i,j}^{k-1}].$$

We can chain these probabilities and get

$$p_{i,j} = \prod_{k=1}^{n/2} p_{i,j}^k.$$

We proceed by providing lower bounds on  $p_{i,j}^k$  for every  $k$  between 1 and  $n/2$ . Let us denote the  $k$ -th added letter by  $A_k$ .

- (1) For  $k = i$ , we have to insert  $\bar{C}$  inside the pair  $t$  and thus  $p_{i,j}^i = \frac{1}{n+2i-1}$ .
- (2) For  $k = j$ , we have to insert  $\bar{D}$  next to  $\bar{C}$  and thus  $p_{i,j}^j = \frac{2}{n+2j-1}$ .
- (3) For  $k$  such that  $|k - i| = 1$  or  $|k - j| = 1$ , we have to guarantee that  $A_k$  receives the color opposite to the color of  $\bar{C}$  and  $\bar{D}$ . Due to Lemma 3.1, we have at least  $\lfloor ((n/2) + k - 2)/2 \rfloor \geq (n + 2k - 6)/4$  options and  $p_{i,j}^k \geq \frac{n+2k-6}{4(n+2k-1)} > \frac{n+2k-8}{4(n+2k-1)}$ .
- (4) Otherwise, we can insert  $A_k$  everywhere except in  $t$  or around  $\bar{C}$  and  $\bar{D}$ . There are at most 7 forbidden positions and thus  $p_{i,j}^k \geq \frac{n+2k-8}{n+2k-1}$ .

There can be either 3 or 4 positions for the case (3) depending on whether  $j - i = 2$ . However we can assume that there are 4 of them as we are aiming to obtain a lower bound. Putting it all together, we get

$$\begin{aligned} p_{i,j} &\geq \frac{1}{(n+2i-1)} \frac{2}{(n+2j-1)} \frac{1}{4^4} \prod_{\substack{k=1 \\ k \neq i,j}}^{n/2} \frac{n+2k-8}{n+2k-1} \\ &= \frac{1}{2^7(n+2i-8)(n+2j-8)} \prod_{k=1}^{n/2} \frac{n+2k-8}{n+2k-1} \\ &> \frac{1}{2^7(2n-3)^2} \left( \frac{n-7}{n} \right)^{n/2}. \end{aligned}$$

As we remarked, the lower bound does not hold whenever  $j - i = 1$ . Furthermore, we cannot guarantee the condition S2 if  $i = 1$ . Summing over all other choices of  $i$  and  $j$  we obtain

$$p > \left( \binom{n/2}{2} - n \right) \frac{1}{2^7(2n-3)^2} \left( \frac{n-7}{n} \right)^{n/2} \rightarrow \frac{e^{-7/2}}{2^{12}} > 7 \cdot 10^{-6}.$$

Thus there exists  $n_0$  such that for  $n \geq n_0$  and for fixed  $w' \in W_{n/2}$  the expected size of  $V \subset T$  for which S1 and S2 holds is

$$\mathbb{E}_n |V| = p \cdot |T| > \left\lfloor \frac{n-2}{4} \right\rfloor p > n \cdot 10^{-6}.$$

By recoloring both letters inserted between a color pair of  $V$  we decrease the number of color changes by two. Notice that the condition S2 guarantees that

letters inserted into different color pairs of  $V$  do not neighbor since their first occurrences are colored with 1. Therefore recoloring all inserted letters for all color pairs of  $V$  decreases the number of color changes by  $2|V|$  which implies

$$\gamma_n < \frac{2}{5}n + O(1) - 2\mathbb{E}_n |V| = \left(\frac{2}{5} - 2 \cdot 10^{-6}\right)n.$$

Setting  $\varepsilon = 2 \cdot 10^{-6}$  we get the bound. □

## 4 Recursive star greedy heuristics

In this section we describe new heuristic for binary paint shop problem called *recursive star greedy* (RSG) and discuss its mean output that appears to be approximately  $0.361n$ . That bound is better than the previously described RG heuristics; however, we are not able to prove that rigorously.

Let us start with a simple but crucial observation. In a legal coloring of a word  $w$  there might be a letter  $X$  such that when we flip the colors of  $\underline{X}$  and  $\bar{X}$  the total number of color changes remains the same. Then during the recursive coloring process, we might use this color flip to avoid introducing a new color change.

The *recursive star greedy* (RSG) heuristics is the following modification of RG. It introduces an additional color  $*$  which marks that the occurrences of a given letter can be colored in any of the two legal ways without changing the total number of color changes. RSG maintains the invariant that for any letter  $X$ , either both  $\underline{X}$  and  $\bar{X}$  are colored by stars, or they are colored by different binary colors, and that there are no two neighboring letters both colored by star. At the point when another letter is inserted next to a letter  $X$  colored by star, RSG recolors  $\underline{X}$  and  $\bar{X}$  with binary colors if it prevents increasing the number of color changes. The final output of RSG is then a binary coloring of  $w$  obtained by recoloring all stars with binary colors in an arbitrary (legal) way.

More precisely, let  $w \in W_n$ . We first color  $w$  with coloring  $c^*$  and then recolor stars to obtain binary coloring  $c$ ; that is,

$$c^* : [2n] \rightarrow C = \{0, 1, *\} \quad \text{and} \quad c : [2n] \rightarrow \{0, 1\}.$$

We proceed recursively with adding (both copies  $\underline{X}$   $\bar{X}$  of) the first letter  $A$  but keep in mind the positions of the last added letter  $B$ , resp. record the number of color changes around  $B$  to possibly recolor it by a star.

**Recursive star greedy:** Let 1 and  $j$  be the positions of a letter  $A$  in a word  $w \in W_n$ . Let  $w' \in W_{n-1}$  be the word that one obtains from  $w$  by deleting  $A$  and  $\bar{A}$ ,  $B$  be the first letter of  $w'$ , and  $c'$  be the  $C$ -coloring of  $w'$  from the recursion. Recall that  $N_{c'}(\bar{A}) = \{c'(j-2), c'(j-1)\}$  denotes the multiset of colors that  $c'$  uses in the neighborhood of  $\bar{A}$ . We define the coloring  $c^*$  of  $w$  by the case analysis below. Note that we use the notation  $c^*(A) = c^*(1)$  and  $c^*(\bar{A}) = c^*(j)$ .

- (A) If  $j = 2$  and  $n > 1$  we set  $c^*(A) = 1 - c'(1)$ .
- (B) If  $j = 2n$  we set  $c^*(A) = 1$ .
- (C) If  $N_{c'}(\bar{A}) = \{t, t\}$  for some  $t \in \{0, 1\}$  we set  $c^*(A) = 1 - t$ .
- (D) If  $N_{c'}(\bar{A}) = \{0, 1\}$  we set  $c^*(A) = c'(1)$ .
- (E) If  $N_{c'}(\bar{A}) = \{1 - c'(1), *\}$  we set  $c^*(A) = c'(1)$ .
- (F) If  $N_{c'}(\bar{A}) = \{c'(1), *\}$  we set  $c^*(A) = c'(1)$ . Moreover, we set  $c^*(X) = 1 - c'(1)$  and  $c(\bar{X}) = c'(1)$ , where  $X$  is the neighbor of  $\bar{A}$  with star color.
- (G) If  $B$  and  $\bar{B}$  are not neighboring, all their neighbors are colored with binary colors and swapping the colors of  $B$  and  $\bar{B}$  preserves the total number of color changes, we set  $c^*(B) = c^*(\bar{B}) = *$ .

Finally, we color  $c^*(\bar{A})$  accordingly. The output of RSG for  $w$  – the binary coloring  $c$  – is obtained by coloring every \*-colored letter  $X$  arbitrarily, say  $X$  with 0 and  $\bar{X}$  with 1.

An example of better performance of RSG over RG is for the word  $w = ABCBDCAD$ , for which one obtains  $rg(w) = 3$ , but  $rsg(w) = 2$ . We follow both algorithms in the following table.

tab:RSG\_ex1

algorithm	$ABCBDCAD$	$BCBDCD$	$CD$	$DD$
RG	1 0 1 1 1 0 0 0	0 1 1 1 0 0	1 1 0 0	1 0
RSG	0 0 0 1 1 1 1 0	0 * 1 1 * 0	1 1 0 0	1 0

On the other hand, there are words for which RG still performs better than RSG, as may be observed in the following table where word  $v = WXABCBDCAD$  is colored. Surprisingly, we can create  $v$  by adding two new letters to  $w$  in a way that they both introduce one new color change to RSG, but not to RG. Hence  $rg(v) = 3$  but  $rsg(v) = 4$ .

tab:RSG\_ex2

algorithm	$WXABCBDCAD$	$XABCBDCAD$
RG	1 1 1 0 1 1 1 0 0 0 0 0	1 1 0 1 1 1 0 0 0 0
RSG	0 1 * 0 0 1 1 1 1 * 0 0	1 * 0 0 1 1 1 * 0 0

Observe that the star color is introduced only to paint the letter B in case (G) and that the algorithm never colors two neighboring letters by star. Every star is then recolored back to a binary color either at the end with the transformation of  $C$ -coloring into a binary coloring, or in case (F) where it allows us to color  $A$  and  $\bar{A}$  without increasing the total number of changes. That is in contrast to RG, for which the number of color changes in such step might have increased. This adjustment seems to improve the final number of color changes as the following lines suggest.

Computer experiments suggest, that this modification leads to a significant saving in terms of color changes:



**Conjecture 4.1.**  $\mathbb{E}_n rsg = 0.361n - o(n)$

We were unable to prove this conjecture. However, we provide below some of the arguments we have tried. and that explain where the constant 0.361 comes from.

Let  $c^*$ , resp.  $c$ , be the  $C$ -coloring, resp. binary coloring, of a random word  $w \in W_n$  produced by RSG. Let  $s_n$  be the probability that a random letter  $A \in w$  is assigned the star color in the coloring  $c^*$ . Then  $ns_n$  is the expected number of pairs of stars (coupled by the letter they color) in  $c^*$ . Furthermore, let  $a_n/2$  be the probability that a random pair of neighbouring letters in  $w$  constitute a color change in the coloring  $c$ . Observe that  $\mathbb{E}_n rsg = (2n - 1)a_n/2$  is the value we are interested in.

One could express the probabilities of the individual cases (A) - (G) with the variables  $a_n$  and  $s_n$ , which leads to recurrence relations that allow one to compute  $\mathbb{E}_n rsg$  for fixed  $n$  in time that is polynomial in  $n$ . However, we were not able to solve these recurrences in general.

For a word  $w \in W_n$  and  $j \in [1, 2n - 1]$ , let  $N_c(w, j)$  be the pair of colors  $c(j)c(j+1)$ . Based on  $s_n$  and  $a_n$  we can count the probabilities of all variations of pairs of neighboring colors in  $c^*$ -coloring. They are displayed in the Pr-column of the Table 4. For example, in case (D2), the probability of a color with condition  $c(2) = 1$  is

$$\begin{aligned} \Pr_{w,j}[(D2)] &= \frac{1}{2} \left( \Pr_{w,j}[N_c(w, j) \in \{01, 10\}] - \frac{1}{2} \Pr_{w,j}[* \in N_{c^*}(w, j)] \right) + O(1/n) \\ &= \frac{1}{2} \left( \frac{a_n}{2} - s_n \right) = \frac{a_n}{4} - \frac{s_n}{2} + O(1/n), \end{aligned}$$

where the first fraction  $1/2$  refers to the condition  $c(2) = 1$  and the second fraction  $1/2$  is there because only half of the color changes with stars produce 01/10 color changes (others produce 00/11). Finally, the  $O(1/n)$  term includes the case when  $|N_c(w, j)| = 1$ , i.e. when  $\bar{A}$  gets inserted either at the beginning or at the end of the word. Other probabilities of Table 4 up to the error term  $O(1/n)$  can be calculated similarly. Set  $q_n = 1/4 - a_n/8 - s_n/4$  to be the probability of cases (C1)-(C4).

Observe that the average increase of color change in  $c$  is

$$(2n - 1) \frac{a_n}{2} - (2n - 3) \frac{a_{n-1}}{2} = a_{n+1} + (a_{n+1} - a_n) \left( n - \frac{1}{2} \right).$$

That increase happens only in cases (C1) and (C2); indeed, they are the only cases with  $+1$  in  $\Delta a_n$ -column and nonzero probability, we have

$$a_{n+1} + (a_{n+1} - a_n) \left( n - \frac{1}{2} \right) = \Pr \left[ (C1) \cup (C2) \right] = 2q_n + O(1/n).$$

By definition, the average increase of the number of stars is  $ns_n - (n-1)s_{n-1}$ . We may color  $A$  by a star in next step in cases (C1), (C2), (D1), (D2), (F1) and (F2), but only under the condition that the color of the next added letter

tab:RCG

Recursive star greedy							
Case	$w$	$N_{c^*}(A)$	$c(A)$	$c(12)$	$\Delta a_n$	$\Delta s_n$	Pr
(A1)	$\underline{A}A0\dots\dots$	$\{0\}$	1	10	+1		0
(A2)	$\underline{A}\bar{A}1\dots\dots$	$\{1\}$	0	01	+1		0
(B1)	$\underline{A}0\dots\dots 0\bar{A}$	$\{0\}$	1	10	+1		0
(B2)	$\underline{A}1\dots\dots 0\bar{A}$	$\{0\}$	1	11	=		0
(C1)	$\underline{A}0\dots 0\bar{A}0\dots$	$\{0,0\}$	1	10	+1	+	$q_n$
(C2)	$\underline{A}1\dots 1\bar{A}1\dots$	$\{1,1\}$	0	01	+1	+	$q_n$
(C3)	$\underline{A}1\dots 0\bar{A}0\dots$	$\{0,0\}$	1	11	=		$q_n$
(C4)	$\underline{A}0\dots 1\bar{A}1\dots$	$\{1,1\}$	0	00	=		$q_n$
(D1)	$\underline{A}0\dots 0\bar{A}1\dots$	$\{0,1\}$	0	00	=		$a_n/4 - s_n/2$
(D2)	$\underline{A}1\dots 0\bar{A}1\dots$	$\{0,1\}$	1	11	=		$a_n/4 - s_n/2$
(E1)	$\underline{A}1\dots 0\bar{A} * \dots$	$\{0,*\}$	1	11	=		$s_n/2$
(E2)	$\underline{A}0\dots 1\bar{A} * \dots$	$\{1,*\}$	0	00	=		$s_n/2$
(F1)	$\underline{A}0\dots 0\bar{A} * \dots$	$\{0,*\}$	0	00	=	−	$s_n/2$
(F2)	$\underline{A}1\dots 1\bar{A} * \dots$	$\{1,*\}$	1	11	=	−	$s_n/2$

Table 1: Case analysis of recursive star greedy heuristics:  $N_{c^*}(\bar{A})$  is the multiset of colors around  $\bar{A}$ ,  $c(A)$  the resulting color of  $A$ ,  $c(12) = c(1)c(2)$ ,  $\Delta a_n$ , resp.  $\Delta s_n$  are the increases of number of color changes of  $c$ , resp. pairs of stars, Pr is the probability for that case up to  $O(1/n)$  error term and  $q_n = 1/4 - a_n/8 - s_n/4$ .

$X$  will be opposite to the color of  $\underline{A}$ , which happens only in the cases (C1) and (C2). Here we assume that these events are independent. On the other hand, we can also decrease the number of stars by recoloring them to binary colors which happens in case (F). Hence

$$\begin{aligned}
& s_{n+1} + (s_{n+1} - s_n)n \\
&= \Pr \left[ (C1) \cup (C2) \right] \Pr \left[ (C1) \cup (C2) \cup (D) \cup (F) \right] - \Pr \left[ (F) \right] \\
&= 2q_n \left( 2q_{n-1} + \frac{a_{n-1}}{2} \right) - s_n + O(1/n).
\end{aligned}$$

Our goal is to solve the system of two recurrences above. However, we are not able to do that and we need additional assumptions. We assume that

$$a_n - a_{n-1} = o(1/n) \quad \text{and} \quad s_n - s_{n-1} = o(1/n). \quad (2)$$

assump

That allows us to substitute  $a_n$  by  $a_{n\pm 1}$  and  $s_n$  by  $s_{n\pm 1}$ . Hence we modify the latter system and obtain

$$\begin{aligned}
a_n &= \frac{1}{2} - \frac{a_n}{4} - \frac{s_n}{2} + o(1) \\
s_n &= \frac{3}{2}a_n^2 - s_n + o(1)
\end{aligned}$$

with the only relevant solution

$$a_n = 0.361 \quad \text{and} \quad s_n = 0.098$$

where  $0.361 = (\sqrt{37} - 5)/3$  is the root of  $3a_n^2 + 10a_n - 4 = 0$ . We were able to compute the values  $a_n$  and  $s_n$  for  $n$  up to 120. The obtained data suggest that our assumptions (2) are reasonable, and that both sequences  $(a_n)_{n \geq 1}$  and  $(s_n)_{n \geq 1}$  seems to be monotone and bounded.

Finally, we remark that the RSG heuristics actually uses two somewhat different types of stars since the pair of letters  $\bar{X}, \bar{X}$  colored by star can be either surrounded by four neighbors of the same color or both  $\bar{X}$  and  $\bar{X}$  have one neighbor colored with 0 and the other colored using 1. It might be more feasible to analyze a modified heuristic that only uses the second type of stars. The numerical data suggest that this heuristic still performs better than RG albeit worse than RSG. However, we were not able to rigorously show this either.

## 5 Concentration result for the optimal number of color changes

In this section, we show that the random variable  $\gamma(w)$  is strongly concentrated around its expected value  $\gamma_n$ . We will need the following version of the Azuma-Hoeffding inequality.

**Proposition 5.1.** *Let  $X_0, X_1, \dots, X_n$  be a martingale with  $\mathbb{E}X_n = X_0$  and let  $c$  be a real number such that  $|X_k - X_{k-1}| \leq c$  for every  $k \in \{1, \dots, n\}$ . Then,*

$$\Pr [|X_n - X_0| \geq t] \leq 2 \exp \left( \frac{-t^2}{2c^2n} \right).$$

We proceed with the concentration result.

thm:concentrationcc

**Theorem 5.2.** *Let  $w$  be a random element of  $W_n$ . Then*

$$\Pr \left[ |\gamma(w) - \gamma_n| \geq \sqrt{n \log n} \right] \leq 2n^{-1/8}.$$

*Proof.* For a word  $w \in W_n$  and a letter  $A_k$  with  $k \in \{1, \dots, n\}$ , let  $w(k) \in \binom{\{1, \dots, 2n\}}{2}$  denote the set of the two positions of the occurrences of  $A_k$  in  $w$ .

First, we describe a procedure of generating a uniformly random word  $w$  from  $W_n$ . In each step  $k = 1, \dots, n$ , we uniformly randomly choose a pair of two distinct positions  $i_k, j_k$  from  $\{1, \dots, 2n\} \setminus \{i_1, j_1, \dots, i_{k-1}, j_{k-1}\}$ . These are the positions where the letter  $A_k$  occurs in  $w$ , i.e., we set  $w(k) = \{i_k, j_k\}$ . Clearly, the resulting random word  $w$  is uniform from  $W_n$ . For  $k \in \{0, 1, \dots, n\}$ , let  $U_k \subseteq W_n$  be the (random) set of words  $w'$  such that the positions of the letters  $A_1, \dots, A_k$  agree with  $w$ , that is,  $w'(\ell) = w(\ell)$  for all  $\ell \in \{1, \dots, k\}$ . Note that  $U_0 = W_n$  and  $U_n = \{w\}$ .

We now introduce a random process  $X_0, X_1, \dots, X_n$  as follows. For  $k \in \{0, 1, \dots, n\}$ , let  $X_k$  be the random variable equal to the expectation of  $\gamma(w)$  after the positions of the letters  $A_1, \dots, A_k$  have been fixed. In other words,  $X_k$

is equal to the average of  $\gamma(w')$  taken over all words  $w' \in U_k$ . Clearly, we have  $X_0 = \gamma_n$  and  $X_n = \gamma(w)$ .

By its definition, the process  $X_0, X_1, \dots, X_n$  forms a martingale. Our goal now is to show that

$$|X_k - X_{k-1}| \leq 2 \quad (3)$$

eq:martingale1

holds for every  $k \in \{1, \dots, n\}$ . Fix the positions  $i_1, j_1, \dots, i_k, j_k$  of the generated word  $w$ . For a word  $u \in U_k$ , let  $U_{k-1}^u$  denote the set of all words  $u' \in U_{k-1}$  with the same relative ordering of all letters, except for  $A_k$ , as in  $u$ . That is, a word  $u'$  is in  $U_{k-1}^u$  if  $u' \in U_{k-1}$  and the induced sub-word obtained by removing both occurrences of the letter  $A_k$  is the same for  $u'$  and  $u$ .

We claim that for every  $u \in U_k$  and  $u' \in U_{k-1}^u$ ,

$$|\gamma(u) - \gamma(u')| \leq 2. \quad (4)$$

eq:martingale2

Fix an optimal colouring  $\varphi: \{1, \dots, n\} \rightarrow \{0, 1\}$  of  $u$ , where  $\varphi(\ell)$  is the colour assigned to the first occurrence of the letter  $A_\ell$  in  $u$ . Let  $\varphi'$  be the colouring that is equal to  $\varphi$  on  $\{1, \dots, n\} \setminus \{k\}$ , with the value of  $\varphi'(k)$  chosen as the color in  $\varphi$  of the letter preceding the first occurrence of  $A_k$  in  $u'$ . Thus, the colouring  $\varphi'$  applied to  $u'$  has at most two additional colour changes. By an analogous argument for the other inequality, we obtain (4). In fact, we will need the easy corollary of (4) that for every  $u \in U_k$ , we have

$$\left| \gamma(u) - \frac{1}{|U_{k-1}^u|} \sum_{u' \in U_{k-1}^u} \gamma(u') \right| \leq 2. \quad (5)$$

eq:martingale3

Observe that  $U_{k-1}$  is equal to the disjoint union of  $U_{k-1}^u$ , taken over all  $u \in U_k$ . Also, the size of  $U_{k-1}^u$  is the same for all  $u \in U_k$ . It follows that

$$X_{k-1} = \frac{1}{|U_{k-1}|} \sum_{u' \in U_{k-1}} \gamma(u') = \frac{1}{|U_k|} \sum_{u \in U_k} \frac{1}{|U_{k-1}^u|} \sum_{u' \in U_{k-1}^u} \gamma(u').$$

We obtain

$$|X_k - X_{k-1}| \leq \frac{1}{|U_k|} \sum_{u \in U_k} \left| \gamma(u) - \frac{1}{|U_{k-1}^u|} \sum_{u' \in U_{k-1}^u} \gamma(u') \right| \leq 2,$$

where the second inequality follows from (5). We have proved (3).

Finally, the Azuma-Hoeffding inequality applied to the martingale  $X_0, X_1, X_2, \dots, X_n$  with  $c = 4$  yields that

$$\Pr[|\gamma(w) - \gamma_n| \geq t] \leq 2 \exp\left(\frac{-t^2}{8n}\right).$$

Plugging in  $t = \sqrt{n \log n}$ ,

$$\Pr[|\gamma(w) - \gamma_n| \geq \sqrt{n \log n}] \leq 2 \exp\left(\frac{-\log n}{8}\right) = \frac{2}{n^{1/8}}.$$

This concludes the proof of Theorem 5.2.  $\square$

We remark that the function  $\sqrt{n \log n}$  in Theorem 5.2 could be replaced by  $f(n)\sqrt{n}$  for an arbitrary function  $f$  such that  $f(n) \rightarrow \infty$  for  $n \rightarrow \infty$ . We would still have the key corollary that for a random word  $w \in W_n$ ,

$$\Pr \left[ |\gamma(w) - \gamma_n| \geq f(n)\sqrt{n} \right] \rightarrow 0.$$

## Acknowledgements

Our attention to the paint shop problem was brought by a nice talk given by W. Hochstättler at Midsummer combinatorial workshop in Prague (MCW2017). We thank both to Winfried and to the organizers of the workshop. This research was started during workshop KAMAK 2017, we are grateful to its organizers. The research was supported by the grant SVV-2017-260452.

## References

- [Alon] [1] ALON, N. Splitting necklaces. *Adv. in Math.* 63, 3 (1987), 247–253.
- [AH] [2] ANDRES, S. D., AND HOCHSTÄTTLER, W. Some heuristics for the binary paint shop problem and their expected number of colour changes. *J. Discrete Algorithms* 9, 2 (2011), 203–211.
- [BEH] [3] BONSMMA, P., EPPING, T., AND HOCHSTÄTTLER, W. Complexity results on restricted instances of a paint shop problem for words. *Discrete Appl. Math.* 154, 9 (2006), 1335–1343.
- [EHO] [4] EPPING, T., HOCHSTÄTTLER, W., AND OERTEL, P. Complexity results on a paint shop problem. *Discrete Appl. Math.* 136, 2-3 (2004), 217–226. The 1st Cologne-Twente Workshop on Graphs and Combinatorial Optimization (CTW 2001).
- [MN] [5] MEUNIER, F., AND NEVEU, B. Computing solutions of the paintshop-necklace problem. *Comput. Oper. Res.* 39, 11 (2012), 2666–2678.
- [MS] [6] MEUNIER, F., AND SEBŐ, A. Paintshop, odd cycles and necklace splitting. *Discrete Appl. Math.* 157, 4 (2009), 780–793.
- [RIG] [7] SCHEINERMAN, E. R. Random interval graphs. *Combinatorica* 8, 4 (1988), 357–371.