The binary paint shop problem

Stephan Dominique Andres

FernUniversität in Hagen
Lehrgebiet Diskrete Mathematik und Optimierung

MCW 2012, Prague, 30 July – 3 August 2012
1. The binary paint shop problem and its complexity status

2. Greedy heuristic

3. Problems
Instances of the *binary paint shop problem* PPW(2,1) are double occurrence words, i.e. words in which every letter occurs exactly twice, and every letter must be colored red once and blue once, so that the number of color changes is minimized.
The binary paint shop problem

Definition

Instances of the binary paint shop problem PPW(2,1) are double occurrence words, i.e. words in which every letter occurs exactly twice, and every letter must be colored red once and blue once, so that the number of color changes is minimized.

given word: ADEBAFCBCDEF
The binary paint shop problem

Definition

Instances of the *binary paint shop problem* PPW(2,1) are *double occurrence words*, i.e. words in which every letter occurs exactly twice, and every letter must be colored *red once* and *blue once*, so that the number of color changes is minimized.

Given word: ADEBAFCBCDEF

ADEBAFCBCDEF 4 color changes
The binary paint shop problem

Definition

Instances of the **binary paint shop problem** PPW(2,1) are **double occurrence words**, i.e. words in which **every letter occurs exactly twice**, and **every letter must be colored red once and blue once**, so that the number of color changes is minimized.

given word: ADEBAFCBCDDEF

<table>
<thead>
<tr>
<th>Word</th>
<th>Color Changes</th>
</tr>
</thead>
<tbody>
<tr>
<td>ADEBAFCBCDDEF</td>
<td>4</td>
</tr>
<tr>
<td>ADEBAFCBCDDEF</td>
<td>4</td>
</tr>
</tbody>
</table>
The binary paint shop problem

Definition

Instances of the binary paint shop problem PPW(2,1) are double occurrence words, i.e. words in which every letter occurs exactly twice, and every letter must be colored red once and blue once, so that the number of color changes is minimized.

given word: ADEBAFCBCDEF

ADEBAFCBCDEF  4 color changes
ADEBAFCBCDEF  4 color changes
ADEBAFCBCDEF  2 color changes
Interval representation

Every interval must be cut an odd number of times.
Every interval must be cut an odd number of times.
Definition

Interval representation

Every interval must be cut an odd number of times.
interval representation

Every interval must be cut an odd number of times.
The binary paint shop problem
The binary paint shop problem and its complexity status
Complexity status

The complexity of the binary paint shop problem

Theorem (Bonsma, Epping, Hochstättler (06); Meunier, Sebő (09))

The \textit{binary paint shop problem} is $\text{APX}$-hard.
The complexity of the binary paint shop problem

Theorem (Bonsma, Epping, Hochstättler (06); Meunier, Sebő (09))

The binary paint shop problem is \( \text{APX} \)-hard.

Corollary

The binary paint shop decision problem is \( \mathcal{NP} \)-complete.
The binary paint shop problem

The complexity of the binary paint shop problem

**Theorem (Bonsma, Epping, Hochstättler (06); Meunier, Sebő (09))**

The binary paint shop problem is $\text{APX}$-hard.

**Corollary**

The binary paint shop decision problem is $\text{NP}$-complete.

**Problem**

Is the binary paint shop problem in $\text{APX}$, i.e. is there a (polynomial) constant factor approximation?
Color the first letter red.

Scan the word from left to right, as long possible use the same color.

ABBCDECFGDFHJIKLHKAJIELG
Color the first letter red.

Scan the word from left to right, as long possible use the same color.

AB BCDECFGDFHIJKLMNOPQRSTUVWXYZ
Color the first letter red.

Scan the word from left to right, as long possible use the same color.

ABBCDE CFGDFHIJKLHKAJIELG
The binary paint shop problem

Greedy heuristic

Greedy heuristic

Color the first letter red.

Scan the word from left to right, as long possible use the same color.

ABBCDECFGD FHIJKLHKAJIELG
Greedy heuristic

Color the first letter red.

Scan the word from left to right, as long possible use the same color.

ABBCDECFGFHIJKL HKAJIELG
Greedy heuristic

Color the first letter red.

Scan the word from left to right, as long possible use the same color.

ABBCDECFGDHFIJKLMNOP

AJIELG
Color the first letter red.

Scan the word from left to right, as long possible use the same color.

ABBCDECFDFHIJKLHKA JIELG
Greedy heuristic

Color the first letter red.

Scan the word from left to right, as long possible use the same color.

ABBCDECFGDFHIJKLHKAJIEL  G
Color the first letter red.

Scan the word from left to right, as long possible use the same color.

ABBCDECFDFHIJKLMNOPAELG
Observation

The greedy heuristic is optimal on instances that do not contain subwords of the form ABBA.
Optimality of the greedy heuristic

Observation

The greedy heuristic is optimal on instances that do not contain subwords of the form ABBA.
Optimality of the greedy heuristic

Observation

*The greedy heuristic is optimal on instances that do not contain subwords of the form ABBA.*
Observation

The greedy heuristic is optimal on instances that do not contain subwords of the form ABBA.
Optimality of the greedy heuristic

**Observation**

The greedy heuristic is optimal on instances that do not contain subwords of the form $ABBA$. 

![Graph representing the greedy heuristic and its optimality conditions.](image-url)
Optimality and perfectness of the greedy heuristic

Theorem (Amini, Meunier, Michel, Mohajeri (2010))

The greedy heuristic is optimal on instances that do not contain subwords of the form $ABACCB$ or $ABBCAC$. 
The binary paint shop problem
Greedy heuristic
Greedy heuristic

Optimality and perfectness of the greedy heuristic

Theorem (Amini, Meunier, Michel, Mohajeri (2010))

The greedy heuristic is optimal on instances that do not contain subwords of the form $ABACCB$ or $ABBCAC$.

Theorem (Rautenbach, Szigeti (2012))

The greedy heuristic is optimal on every subword of a word $w$ if and only if $w$ does not contain subwords of the form $ABACCB$ or $ADDBCCAB$ or $ADDCBCAB$. 
The binary paint shop problem
Greedy heuristic
Greedy heuristic

Expected number of color changes for the greedy

Theorem (Amini, Meunier, Michel, Mohajeri (2010))

The expected number of color changes for the greedy heuristic on uniformly distributed double occurrence words of the length $2n$ with $n$ characters is

$$\mathbb{E}_n(g) \leq \frac{2}{3} n.$$
The binary paint shop problem
Greedy heuristic
Greedy heuristic

Expected number of color changes for the greedy

Theorem (Amini, Meunier, Michel, Mohajeri (2010))

The expected number of color changes for the greedy heuristic on uniformly distributed double occurrence words of the length $2n$ with $n$ characters is

$$E_n(g) \leq \frac{2}{3}n.$$

Conjecture (Amini, Meunier, Michel, Mohajeri (2010))

$$\lim_{n \to \infty} \frac{1}{n} E_n(g) = \frac{1}{2}.$$
On the proof of the conjecture I

**Idea:** The greedy heuristic ignores the first occurrence of a letter.

ABCZDBEDAECZ
Idea: The greedy heuristic ignores the first occurrence of a letter. Consider the last letter $Z$ of a word.

\text{ABCZDBEDAE CZ}
On the proof of the conjecture I

Idea: The greedy heuristic ignores the first occurrence of a letter. Consider the last letter $Z$ of a word. If we delete both occurrences of $Z$, then greedy colors the rest of the word in the same way.
On the proof of the conjecture I

**Idea:** The greedy heuristic ignores the first occurrence of a letter. Consider the last letter $Z$ of a word. If we delete both occurrences of $Z$, then greedy colors the rest of the word in the same way.

Only the second occurrence of $Z$ can create an additional color change.
On the proof of the conjecture I

Idea: The greedy heuristic ignores the first occurrence of a letter. Consider the last letter $Z$ of a word. If we delete both occurrences of $Z$, then greedy colors the rest of the word in the same way. Only the second occurrence of $Z$ can create an additional color change. This happens in about a half of the cases.
Lemma

The greedy heuristic colors the first occurrence of Z red with probability \( \frac{n}{2n-1} \) resp. blue with probability \( \frac{n-1}{2n-1} \).
On the proof of the conjecture II

Lemma

The greedy heuristic colors the first occurrence of $Z$ red with probability $\frac{n}{2n-1}$ resp. blue with probability $\frac{n-1}{2n-1}$.

Proof. First occurrence of $Z$

- at the beginning: $Z$ red — 1 case
- after red letter: $Z$ red — $n - 1$ cases
- after blue letter: $Z$ blue — $n - 1$ cases
First $Z$ red $\frac{n}{2n-1}$, first $Z$ blue $\frac{n-1}{2n-1}$

**Theorem (A, Hochstättler (2011))** The expected number of color changes for the greedy heuristic on uniformly distributed double occurrence words of length $2n$ with $n$ letters is

$$
\mathbb{E}_n(g) = \sum_{k=0}^{n-1} \frac{2k^2 - 1}{4k^2 - 1}.
$$
The binary paint shop problem
Greedy heuristic
Greedy heuristic

First $Z$ red $\frac{n}{2n-1}$, first $Z$ blue $\frac{n-1}{2n-1}$

**Theorem (A, Hochstättler (2011))** The expected number of color changes for the greedy heuristic on uniformly distributed double occurrence words of length $2n$ with $n$ letters is

$$E_n(g) = E_{n-1}(g) +$$
First $Z$ red $\frac{n}{2n-1}$, first $Z$ blue $\frac{n-1}{2n-1}$

**Theorem (A, Hochstättler (2011))** The expected number of color changes for the greedy heuristic on uniformly distributed double occurrence words of length $2n$ with $n$ letters is

$$
\mathbb{E}_n(g) = \mathbb{E}_{n-1}(g) + \frac{n}{2n-1} \cdot \frac{n-2}{2n-3} +
$$
First Z red \( \frac{n}{2n-1} \), first Z blue \( \frac{n-1}{2n-1} \)

**Theorem (A, Hochstättler (2011))** The expected number of color changes for the greedy heuristic on uniformly distributed double occurrence words of length \( 2n \) with \( n \) letters is

\[
E_n(g) = E_{n-1}(g) + \frac{n}{2n-1} \cdot \frac{n-2}{2n-3} + \frac{n-1}{2n-1} \cdot \frac{n-1}{2n-3}
\]
The binary paint shop problem
Greedy heuristic
Greedy heuristic

First $Z$ red $\frac{n}{2n-1}$, first $Z$ blue $\frac{n-1}{2n-1}$

Theorem (A, Hochstättler (2011)) The expected number of color changes for the greedy heuristic on uniformly distributed double occurrence words of length $2n$ with $n$ letters is

$$
\mathbb{E}_n(g) = \mathbb{E}_{n-1}(g) + \frac{n}{2n-1} \cdot \frac{n-2}{2n-3} + \frac{n-1}{2n-1} \cdot \frac{n-1}{2n-3}
$$

i.h.

$$
\sum_{k=0}^{n-2} \frac{2k^2 - 1}{4k^2 - 1} + \frac{n^2 - 2n}{4n^2 - 8n + 3} + \frac{n^2 - 2n + 1}{4n^2 - 8n + 3}
$$
First $Z$ red $\frac{n}{2n-1}$, first $Z$ blue $\frac{n-1}{2n-1}$

**Theorem (A, Hochstättler (2011))** The expected number of color changes for the greedy heuristic on uniformly distributed double occurrence words of length $2n$ with $n$ letters is

$$
\mathbb{E}_n(g) = \mathbb{E}_{n-1}(g) + \frac{n}{2n-1} \cdot \frac{n-2}{2n-3} + \frac{n-1}{2n-1} \cdot \frac{n-1}{2n-3}
$$

i.h. \quad \sum_{k=0}^{n-2} \frac{2k^2 - 1}{4k^2 - 1} + \frac{n^2 - 2n}{4n^2 - 8n + 3} + \frac{n^2 - 2n + 1}{4n^2 - 8n + 3}

$$
= \sum_{k=0}^{n-2} \frac{2k^2 - 1}{4k^2 - 1} + \frac{2(n-1)^2 - 1}{4(n-1)^2 - 1} = \sum_{k=0}^{n-1} \frac{2k^2 - 1}{4k^2 - 1}.
$$
On the performance of the greedy heuristic

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]
On the performance of the greedy heuristic

\[ A_1 \ldots A_k \ A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]
On the performance of the greedy heuristic

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} \ A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]
On the performance of the greedy heuristic

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} \ A_1 A_{k+1} A_{2k} A_{k+2} \ldots A_{k-1} A_{2k-1} \]
On the performance of the greedy heuristic

\[ A_1 \ldots A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]
On the performance of the greedy heuristic

\[ A_1 \ldots A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]
On the performance of the greedy heuristic

\[ A_1 \ldots A_k A_{k+1} \ldots A_{2k} A_k A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]

\[ A_1 \ldots A_k A_{k+1} \ldots A_{2k} A_{k+1} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]
On the performance of the greedy heuristic

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_k + 1 \ldots A_{k-1} A_{2k-1} \]

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_k + 2 \ldots A_{k-1} A_{2k-1} \]

Greedy: \( n \) color changes; Optimal: 3 color changes
On the performance of the greedy heuristic

\[ A_1 \ldots A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]

\[ A_1 \ldots A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]

Greedy: \( n \) color changes; Optimal: 3 color changes

\[ \Rightarrow \text{Greedy is not a constant factor approximation} \]
The Red-first heuristic and its performance

Color the first occurrence of every letter red

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]
The Red-first heuristic and its performance

Color the first occurrence of every letter red

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]
The Red-first heuristic and its performance

Color the first occurrence of every letter red

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]
The Red-first heuristic and its performance

Color the first occurrence of every letter red

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]
The Red-first heuristic and its performance

Color the first occurrence of every letter red

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]
The Red-first heuristic and its performance

Color the first occurrence of every letter red

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]

on the worst-case instance of greedy red-first colors optimally!
The Red-first heuristic and its performance

Color the first occurrence of every letter red

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]

on the worst-case instance of greedy red-first colors optimally!

\[ B_1 B_2 \ldots B_k B_k B_{k+1} B_{2k} B_{k+2} \ldots B_k B_{2k} B_{k+1} B_{k+2} \ldots B_{2k} \]
The Red-first heuristic and its performance

Color the first occurrence of every letter red

\[ A_1 \ldots A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]

on the worst-case instance of greedy red-first colors optimally!

\[ B_1 B_2 \ldots B_k B_{k+1} B_2 B_{k+2} \ldots B_k B_{2k} B_{k+1} B_{k+2} \ldots B_{2k} \]
The Red-first heuristic and its performance

Color the first occurrence of every letter red

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]

on the worst-case instance of greedy red-first colors optimally!

\[ B_1 B_2 \ldots B_k B_1 B_{k+1} B_2 B_{k+2} \ldots B_k B_{2k} B_{k+1} B_{k+2} \ldots B_{2k} \]
The Red-first heuristic and its performance

Color the first occurrence of every letter red

$$A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1}$$

on the worst-case instance of greedy red-first colors optimally!

$$B_1 B_2 \ldots B_k B_1 B_{k+1} B_2 B_{k+2} \ldots B_k B_{2k} B_{k+1} B_{k+2} \ldots B_{2k}$$
The Red-first heuristic and its performance

Color the first occurrence of every letter red

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]

on the worst-case instance of greedy red-first colors optimally!

\[ B_1 B_2 \ldots B_k B_1 B_{k+1} B_2 B_{k+2} \ldots B_k B_{2k} B_k B_{k+1} B_k B_{k+2} \ldots B_{2k} \]
The Red-first heuristic and its performance

Color the first occurrence of every letter red

\[ A_1 \ldots A_k A_k A_{k+1} \ldots A_{2k} A_{2k} A_1 A_{k+1} A_2 A_{k+2} \ldots A_{k-1} A_{2k-1} \]

on the worst-case instance of greedy red-first colors optimally!

\[ B_1 B_2 \ldots B_k B_1 B_{k+1} B_2 B_{k+2} \ldots B_k B_{2k} B_{k+1} B_{k+2} \ldots B_{2k} \]

red-first needs \( n + 1 \) color changes; greedy/optimal coloring needs 2

\[ \rightarrow \text{Red-first is neither a constant factor approximation} \]
The binary paint shop problem
Greedy heuristic
Red-first heuristic

Result for the red-first heuristic

Theorem (A, Hochstättler (2011))

The expected number of color changes for the red-first heuristic on an instance of length $2n$ is

$$\mathbb{E}_n(rf) = \frac{2n + 1}{3}.$$
Idea to improve the greedy heuristic

AZABBBZ
Idea to improve the greedy heuristic

AZ ABBZ
Idea to improve the greedy heuristic

AZ AB BZ
Idea to improve the greedy heuristic

AZABB Z
Idea to improve the greedy heuristic

AZABBZ
The binary paint shop problem
Greedy heuristic
Recursive greedy heuristic

Idea to improve the greedy heuristic

Recursive greedy heuristic:

\textcolor{red}{AZABBZ}

\textcolor{red}{AZABBZ}
Idea to improve the greedy heuristic

\[ AZABBZ \]

Recursive greedy heuristic:

\[ \rightarrow \text{delete both occurrences of the last letter } Z \]

\[ AABB \]
Idea to improve the greedy heuristic

AZABBZ

Recursive greedy heuristic:

→ delete both occurrences of the last letter $Z$

→ color the rest recursively
Idea to improve the greedy heuristic

$AZABBZ$

Recursive greedy heuristic:

$\rightarrow$ delete both occurrences of the last letter $Z$

$\rightarrow$ color the rest recursively

$\rightarrow$ if the first $Z$ is in a color change: color in such a way that no additional color changes is created; otherwise color according to greedy

$A A$
Idea to improve the greedy heuristic

\[ AZABBZ \]

Recursive greedy heuristic:

\[ \rightarrow \text{delete both occurrences of the last letter } Z \]

\[ \rightarrow \text{color the rest recursively} \]

\[ \rightarrow \text{if the first } Z \text{ is in a color change: color in such a way that no additional color changes is created; otherwise color according to greedy} \]

\[ A \ AB \]
Idea to improve the greedy heuristic

Recursive greedy heuristic:

⟶ delete both occurrences of the last letter \( Z \)

⟶ color the rest recursively

⟶ if the first \( Z \) is in a color change: color in such a way that no additional color changes is created; otherwise color according to greedy

\( AZABBZ \)
The binary paint shop problem
Greedy heuristic
Recursive greedy heuristic

Result for the recursive greedy heuristic

Theorem (A, Hochstättler (2011))

For all $n \geq 1$, the expected number $\mathbb{E}_n(rg)$ of color changes for the recursive greedy heuristic is bounded by

$$\frac{2}{5}n + \frac{8}{15} \leq \mathbb{E}_n(rg) \leq \frac{2}{5}n + \frac{7}{10}.$$
Summary

red-first heuristic \( \lim_{n \to \infty} \frac{1}{n} \mathbb{E}_n(rf) = \frac{2}{3} \)

greedy heuristic \( \lim_{n \to \infty} \frac{1}{n} \mathbb{E}_n(g) = \frac{1}{2} \)

recursive greedy heuristic \( \lim_{n \to \infty} \frac{1}{n} \mathbb{E}_n(rg) = \frac{2}{5} \)
Summary

red-first heuristic \[ \lim_{n \to \infty} \frac{1}{n} \mathbb{E}_n(rf) = \frac{2}{3} \]

greedy heuristic \[ \lim_{n \to \infty} \frac{1}{n} \mathbb{E}_n(g) = \frac{1}{2} \]

recursive greedy heuristic \[ \lim_{n \to \infty} \frac{1}{n} \mathbb{E}_n(rg) = \frac{2}{5} \]

Problem

*Find better heuristics (with expected number of color changes \( \leq 2n/5 \)).*
Summary

red-first heuristic \[ \lim_{n \to \infty} \frac{1}{n} E_n(rf) = \frac{2}{3} \]

greedy heuristic \[ \lim_{n \to \infty} \frac{1}{n} E_n(g) = \frac{1}{2} \]

recursive greedy heuristic \[ \lim_{n \to \infty} \frac{1}{n} E_n(rg) = \frac{2}{5} \]

Problem

Find better heuristics (with expected number of color changes \( \leq 2n/5 \)).

Problem

Characterize the instances where the recursive greedy is optimal.
Optimal coloring

Problem

Determine the expected number of color changes for optimal coloring.
### Problem

Determine the expected number of color changes for optimal coloring.

<table>
<thead>
<tr>
<th>$n$</th>
<th>$\mathbb{E}(\text{opt})$</th>
<th>$\frac{\mathbb{E}(\text{opt})}{n}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$\frac{4}{3}$</td>
<td>0.6667</td>
</tr>
<tr>
<td>2</td>
<td>$\frac{26}{15}$</td>
<td>0.5778</td>
</tr>
<tr>
<td>3</td>
<td>$\frac{223}{105}$</td>
<td>0.5310</td>
</tr>
<tr>
<td>4</td>
<td>$\frac{2355}{945}$</td>
<td>0.4984</td>
</tr>
<tr>
<td>5</td>
<td>$\frac{29541}{10395}$</td>
<td>0.4736</td>
</tr>
<tr>
<td>6</td>
<td>$\frac{429677}{135135}$</td>
<td>0.4542</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Optimal coloring

**Problem**

*Determine the expected number of color changes for optimal coloring.*

<table>
<thead>
<tr>
<th>( n )</th>
<th>( \mathbb{E}(\text{opt}) )</th>
<th>( \mathbb{E}(\text{opt}) / n )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>( \frac{4}{3} )</td>
<td>0.6667</td>
</tr>
<tr>
<td>3</td>
<td>( \frac{26}{15} )</td>
<td>0.5778</td>
</tr>
<tr>
<td>4</td>
<td>( \frac{223}{105} )</td>
<td>0.5310</td>
</tr>
<tr>
<td>5</td>
<td>( \frac{2355}{945} )</td>
<td>0.4984</td>
</tr>
<tr>
<td>6</td>
<td>( \frac{29541}{10395} )</td>
<td>0.4736</td>
</tr>
<tr>
<td>7</td>
<td>( \frac{429677}{135135} )</td>
<td>0.4542</td>
</tr>
</tbody>
</table>

**Conjecture (Meunier, Neveu (2012))**

*This number is sublinear in \( n \).*
In case

- there is a constant factor approximation and
- the expected number of color changes for optimal coloring is sublinear

every constant factor approximation for the binary paint shop problem is (in expectation) a better heuristic than the recursive greedy.
The necklace splitting problem

**Given:** open necklace of length $n$ with $t$ types of beads, every type $i$ occurs $q_a_i$ times, $q$ thieves want to cut the necklace in a fair way, so that everyone receives exactly $a_i$ beads of every type $i$. 
The necklace splitting problem

Given: open necklace of length $n$ with $t$ types of beads, every type $i$ occurs $q a_i$ times, $q$ thieves want to cut the necklace in a fair way, so that everyone receives exactly $a_i$ beads of every type $i$.

(binary paint shop: $t = \frac{n}{2}, q = 2, a_i = 1$.)

![Diagram of necklace splitting problem]
The necklace splitting problem

**Given:** open necklace of length $n$ with $t$ types of beads, every type $i$ occurs $qa_i$ times, $q$ thieves want to cut the necklace in a fair way, so that everyone receives exactly $a_i$ beads of every type $i$.  

(binary paint shop: $t = \frac{n}{2}, q = 2, a_i = 1$.)

**Theorem (Alon (1987))**

*There is a solution with at most $(q - 1)t$ cuts.*
The necklace splitting problem

**Given:** open necklace of length \( n \) with \( t \) types of beads, every type \( i \) occurs \( qa_i \) times, \( q \) thieves want to cut the necklace in a fair way, so that everyone receives exactly \( a_i \) beads of every type \( i \).

(binary paint shop: \( t = \frac{n}{2}, q = 2, a_i = 1. \))

**Theorem (Alon (1987))**

*There is a solution with at most \((q - 1)t\) cuts.*

This is best possible.
The necklace splitting problem

**Given:** open necklace of length $n$ with $t$ types of beads, every type $i$ occurs $qa_i$ times, $q$ thieves want to cut the necklace in a fair way, so that everyone receives exactly $a_i$ beads of every type $i$.

(binary paint shop: $t = \frac{n}{2}$, $q = 2$, $a_i = 1$.)

**Theorem (Alon (1987))**

There is a solution with at most $(q - 1)t$ cuts.

This is best possible.

**Problem**

Is there a polynomial algorithm to determine these cuts?
The necklace splitting problem

**Given:** open necklace of length \( n \) with \( t \) types of beads, every type \( i \) occurs \( qa_i \) times, \( q \) thieves want to cut the necklace in a fair way, so that everyone receives exactly \( a_i \) beads of every type \( i \).

(binary paint shop: \( t = \frac{n}{2}, q = 2, a_i = 1 \).)

**Theorem (Alon (1987))**

There is a solution with at most \((q - 1)t\) cuts.

This is best possible.

**Problem (Meunier, Neveu (2012))**

Is the necklace splitting problem PPAD-complete for \( q = 2 \)?

**Problem**

Is there a polynomial algorithm to determine these cuts?
Thank you!