Acyclic Edge Coloring Using Entropy Compression

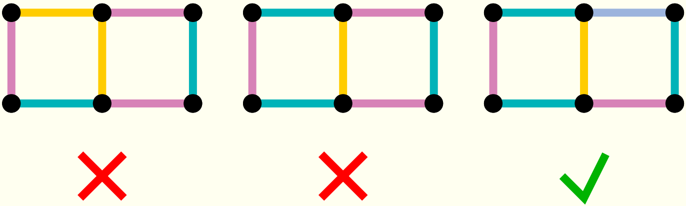
Louis Esperet (G-SCOP, Grenoble, France)
Aline Parreau (LIFL, Lille, France)

Bordeaux Graph Workshop, November 2012

Acyclic Edge Colorings of graphs

An acyclic edge coloring of a graph is a coloring of the edges such that:

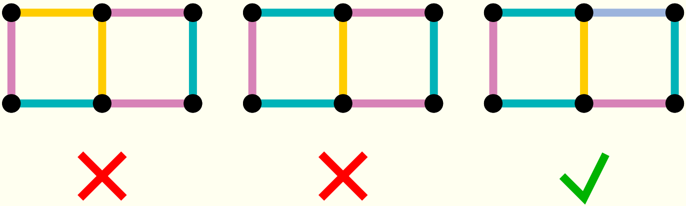
- two edges sharing a vertex have different color,
- there are no bicolored cycles.



Acyclic Edge Colorings of graphs

An acyclic edge coloring of a graph is a coloring of the edges such that:

- two edges sharing a vertex have different color,
- there are no bicolored cycles.



- $a'(G)$: minimum number of colors in an acyclic edge coloring of G.
- If G has maximum degree Δ:

$$a'(G) \geq \Delta.$$

Result

Conjecture Alon, Sudakov and Zaks, 2001

If G has maximum degree Δ, $a'(G) \leq \Delta + 2$.

Result

Conjecture Alon, Sudakov and Zaks, 2001

If G has maximum degree Δ, $a'(G) \leq \Delta + 2$.

Using the Lovász Local Lemma and variations:

- $a'(G) \leq 64\Delta$ (Alon, McDiarmid and Reed, 1991)
- $a'(G) \leq 16\Delta$ (Molloy and Reed, 1998)
- $a'(G) \leq 9.62\Delta$ (Ndreca, Procacci and Scoppola, 2012)

Result

Conjecture Alon, Sudakov and Zaks, 2001

If G has maximum degree Δ, $a'(G) \leq \Delta + 2$.

Using the Lovász Local Lemma and variations:

- $a'(G) \leq 64\Delta$ (Alon, McDiarmid and Reed, 1991)
- $a'(G) \leq 16\Delta$ (Molloy and Reed, 1998)
- $a'(G) \leq 9.62\Delta$ (Ndreca, Procacci and Scoppola, 2012)

Theorem Esperet and P., 2012

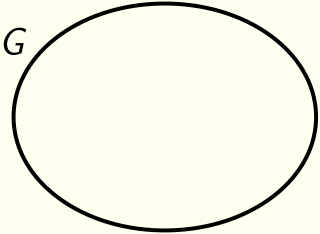
If G has maximum degree Δ, $a'(G) \leq 4\Delta$.

Method of "entropy compression" based on the proof by Moser and Tardos of LLL and extended by Grytczuk, Kozik and Micek.

Algorithm

Order the edge set.
While there is an uncolored edge:

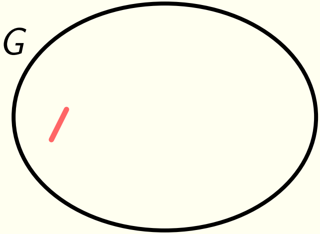
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

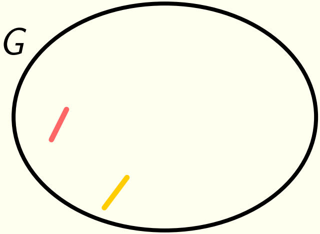
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

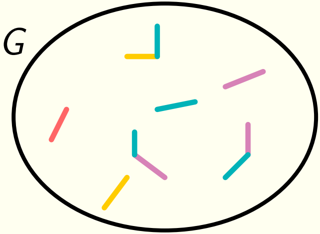
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

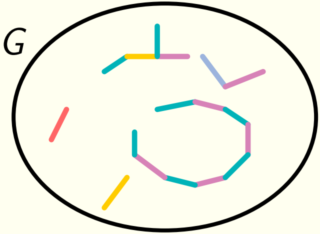
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

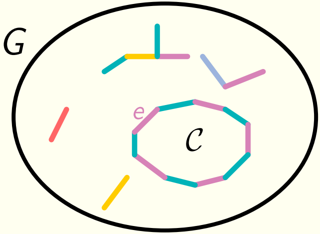
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

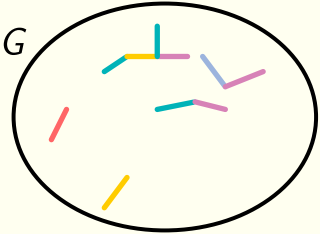
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

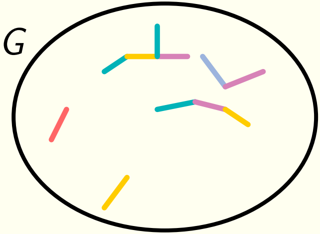
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

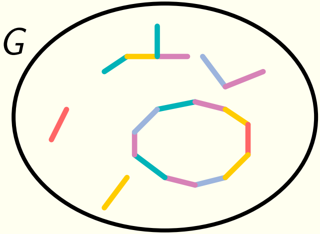
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

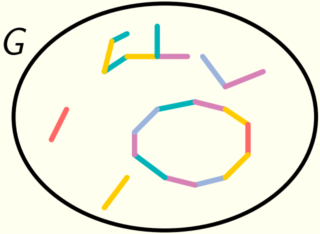
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

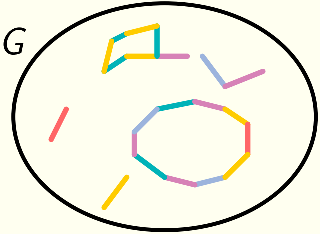
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

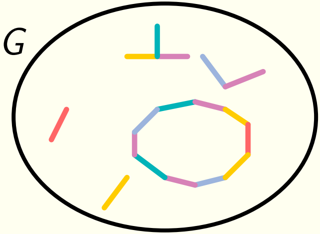
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

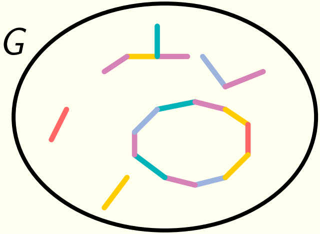
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

Order the edge set.
While there is an uncolored edge:

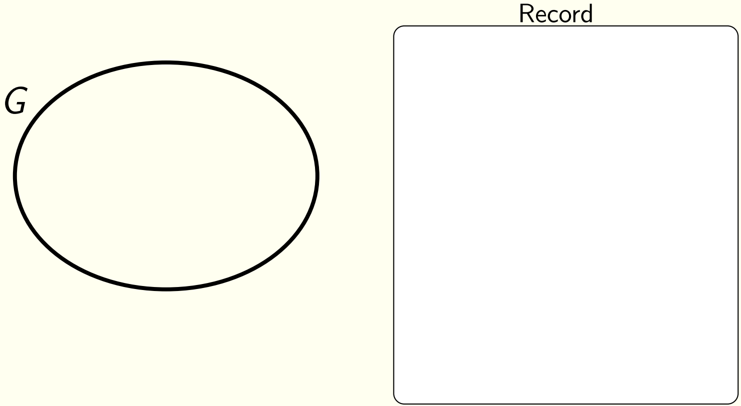
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



We prove that this algorithm ends with non zero probability.
$\Rightarrow$ Any graph has an acyclic edge coloring with 4Δ colors.

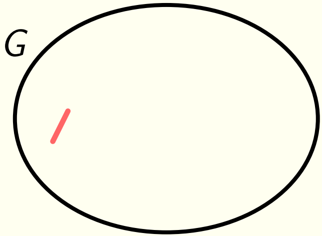
Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.



Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

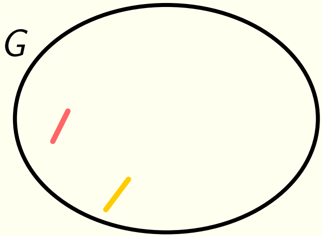


Record

```
1:-
```

Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

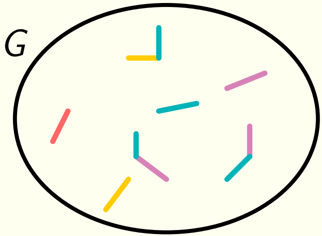


Record

```
1:-
2:-
```

Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

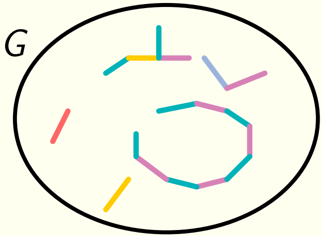


Record

```
1:-
2:-
...
```

Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

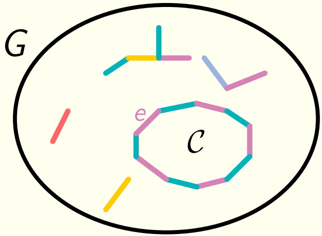


Record

```
1:-
2:-
...
17:-
```

Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

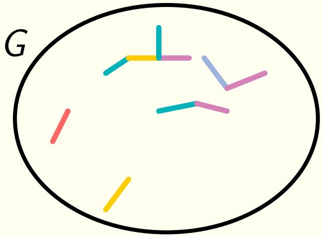


Record

```
1:-
2:-
...
17:-
18:Cycle C is uncolored
```

Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

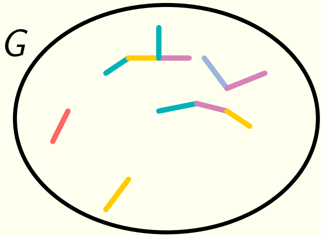


Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
```

Recording

- We assume the algorithm is still running after t steps.
  → bad scenario
- We record in a compact way what happens during the algorithm.

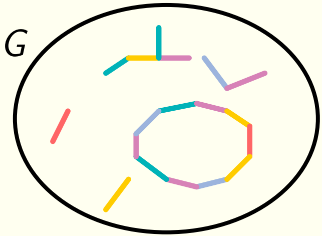


Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
19:-
```

Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

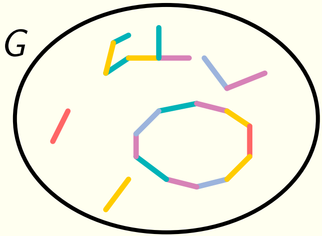


Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
19:-
...
```

Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

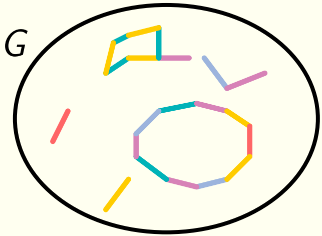


Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
19:-
...
276:-
```

Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

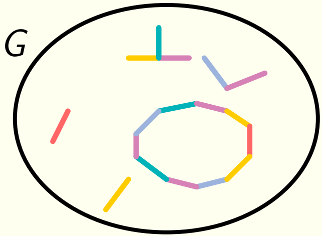


Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
19:-
...
276:-
277:Cycle 𝒞′ is uncolored
```

Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

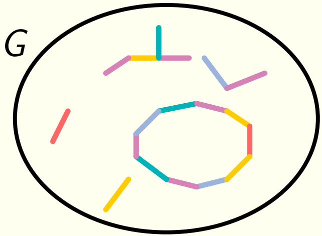


Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
19:-
...
276:-
277:Cycle 𝒞′ is uncolored
```

Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

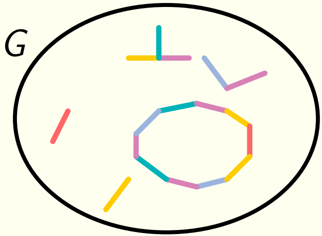


Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
19:-
...
276:-
277:Cycle 𝒞′ is uncolored
278:-
...
```

Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.



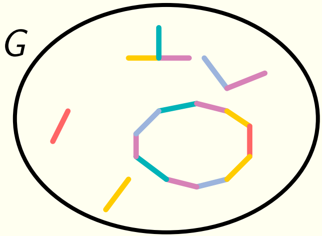
Final partial coloring Φ_t

Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
19:-
...
276:-
277:Cycle 𝒞′ is uncolored
278:-
...
t:-
```

Recording

- We assume the algorithm is still running after t steps.
  $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.



Final partial coloring Φ_t

Record

```
1:-
2:-
...
17:-
18:Cycle 𝒞 is uncolored
19:-
...
276:-
277:Cycle 𝒞′ is uncolored
278:-
...
t:-
```

1 record + 1 final partial coloring = 1 bad scenario

Rewrite the history

1. Top-down reading $\rightarrow$ set of colored edges at each step.

```
1:-
2:-
...
17:-
18:𝒞 is uncolored
19:-
...
276:-
277:𝒞′ is uncolored
278:-
...
t:-
```

1
Sets of colored edges

Rewrite the history

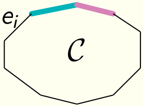
1. Top-down reading → set of colored edges at each step.
2. Down-top reading→ partial coloring at each step and scenario.

```
1:-
2:-
...
17:-
18:𝒞 is uncolored
19:-
...
276:-
277:𝒞′ is uncolored
278:-
...
t:-
```

1 Sets of colored edges

2 partial colorings and scenario

2



Step i

→

Rewrite the history

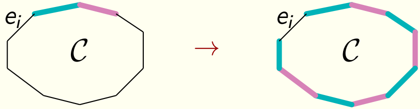
1. Top-down reading $\rightarrow$ set of colored edges at each step.
2. Down-top reading$\rightarrow$ partial coloring at each step and scenario.

```
1:-
2:-
...
17:-
18:𝒞 is uncolored
19:-
...
276:-
277:𝒞′ is uncolored
278:-
...
t:-
```

1
Sets of
colored edges

2
partial colorings
and scenario

2



Step i Step $i-1$

e_i gets

Rewrite the history

1. Top-down reading $\rightarrow$ set of colored edges at each step.
2. Down-top reading$\rightarrow$ partial coloring at each step and scenario.

1:-
2:-
...
17:-
18:$\mathcal{C}$ is uncolored
19:-
...
276:-
277:$\mathcal{C}'$ is uncolored
278:-
...
t:-

1
Sets of
colored edges

2
partial colorings
and scenario

$\Rightarrow$ 1 record + 1 final partial coloring = 1 bad scenario

Summary

1 record $+1$ partial coloring $=$ 1 bad scenario

Summary

$$1 \text{ record } + 1 \text{ partial coloring} = 1 \text{ bad scenario}$$

$$\leq (4\Delta + 1)^m$$

Summary

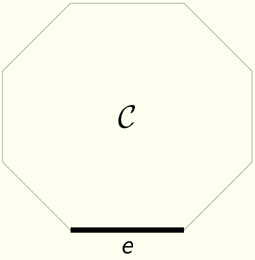
$$1 \text{ record } + 1 \text{ partial coloring} = 1 \text{ bad scenario}$$

? $\qquad \leq (4\Delta + 1)^m \qquad$?

How many possible records ?

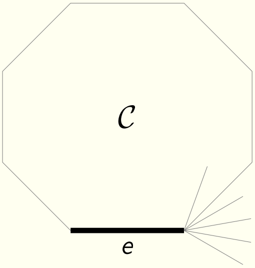
Compact records of cycles

- We know one edge e of $\mathcal{C}$.



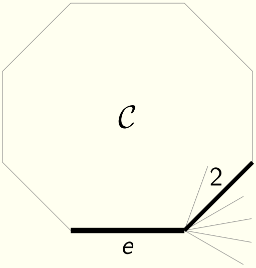
Compact records of cycles

- We know one edge e of $\mathcal{C}$.



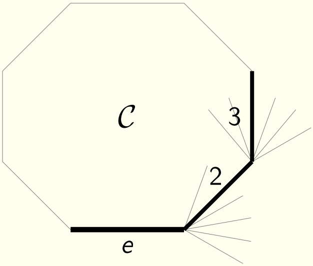
Compact records of cycles

- We know one edge e of $\mathcal{C}$.



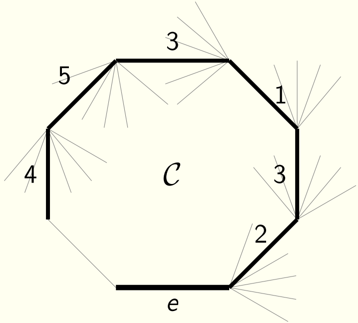
Compact records of cycles

- We know one edge e of $\mathcal{C}$.



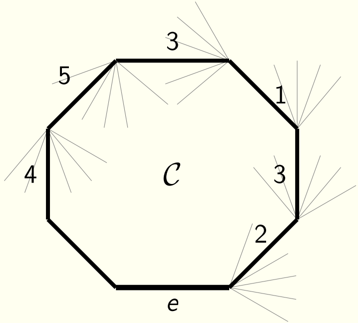
Compact records of cycles

- We know one edge e of $\mathcal{C}$.
- No choice for the last edge



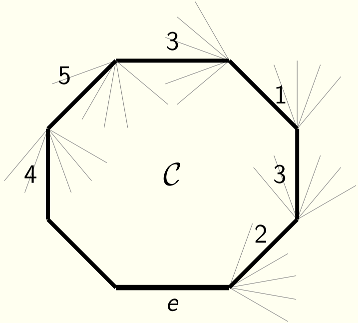
Compact records of cycles

- We know one edge e of $\mathcal{C}$.
- No choice for the last edge



Compact records of cycles

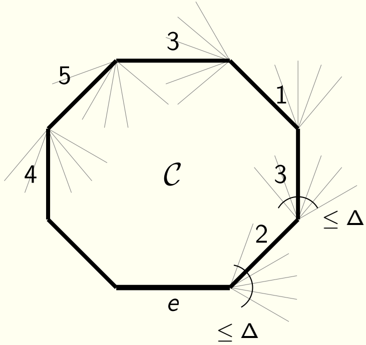
- We know one edge e of $\mathcal{C}$.
- No choice for the last edge



`i:`$\mathcal{C}$ `is uncolored` $\Leftrightarrow$ `i:231354`

Compact records of cycles

- We know one edge e of $\mathcal{C}$.
- No choice for the last edge



`i:`$\mathcal{C}$` is uncolored` $\Leftrightarrow$ `i:231354`

- Cycle coded by a word on $\{1, ..., \Delta\}^{2k-2}$ where $2k$ is the length of $\mathcal{C}$.

Number of records

Record $(- , - , ..., - , 231354 , - , ..., - , 4213 , - , ..., -)$

Number of records

$$
\begin{array}{lccccccccccccccccccccc}
\text{Record} & (& - & , & - & ,\ldots, & - & , & 231354 & , & - & ,\ldots, & - & , & 4213 & , & - & ,\ldots, & - &) \\
 & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & & \downarrow & \\
 & & 0 & & 0 & & 0 & & 0111111 & & 0 & & 0 & & 01111 & & 0 & & 0 &
\end{array}
$$

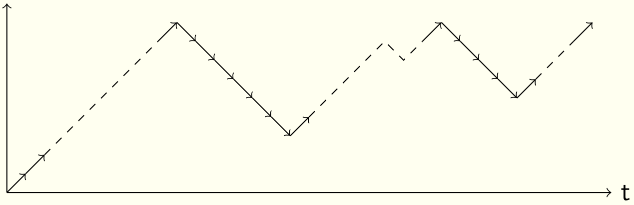
Number of records

Record	(− ,	− ,	…,	− ,	231354 ,	− ,	…,	− ,	4213 ,	− ,	…,	−)
	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$	$\downarrow$	$\downarrow$		$\downarrow$
	0	0		0	0111111	0		0	01111	0		0

Number of colored edges

$0 \leftrightarrow \nearrow$:
an edge is colored

$1 \leftrightarrow \searrow$:
an edge is uncolored



Partial Dyck word of length $\leq 2t$ and blocks of ones of even size.

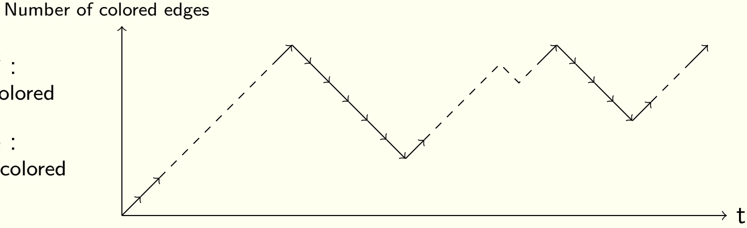
Number of records

Record $(- , - , \ldots, - , 231354 , - , \ldots, - , 4213 , - , \ldots, -)$

$\downarrow$ 0 $\quad$ 0 $\quad$ 0 $\quad$ 0111111 $\quad$ 0 $\quad$ 0 $\quad$ 01111 $\quad$ 0 $\quad$ 0

$0 \leftrightarrow \nearrow$:
an edge is colored

$1 \leftrightarrow \searrow$:
an edge is uncolored



Partial Dyck word of length $\leq 2t$ and blocks of ones of even size.

$\rightarrow$ Number of such words : $2^t/t^{3/2}$

Number of records

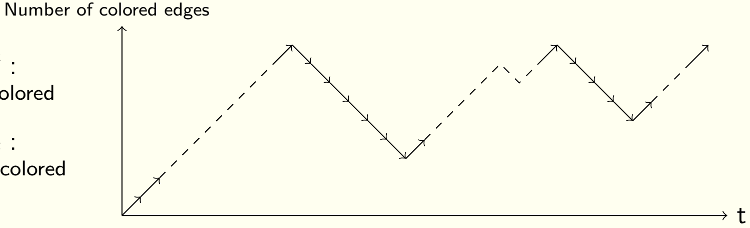
Record $(- , - , \ldots, - , 231354 , - , \ldots, - , 4213 , - , \ldots, -)$

$\downarrow$

$0 \quad 0 \quad 0 \quad 0111111 \quad 0 \quad 0 \quad 01111 \quad 0 \quad 0$

$0 \leftrightarrow \nearrow$:
an edge is colored

$1 \leftrightarrow \searrow$:
an edge is uncolored



Partial Dyck word of length $\leq 2t$ and blocks of ones of even size.

$\rightarrow$ Number of such words : $2^t/t^{3/2}$

$\rightarrow$ Number of records : $(2\Delta)^t/t^{3/2}$

End of the proof

$$1 \text{ record } + 1 \text{ partial coloring} = 1 \text{ bad scenario}$$

$$(4\Delta + 1)^m$$

End of the proof

1 record +1 partial coloring = 1 bad scenario

$(2\Delta)^t/t^{3/2}$

$(4\Delta+1)^m$

End of the proof

1 record $+1$ partial coloring $=$ 1 bad scenario

$(2\Delta)^t/t^{3/2}$

$(4\Delta+1)^m$

$\frac{(4\Delta+1)^m(2\Delta)^t}{t^{3/2}}$

End of the proof

$$1 \text{ record } + 1 \text{ partial coloring} = 1 \text{ bad scenario}$$

$(2\Delta)^t / t^{3/2}$ $\qquad (4\Delta + 1)^m \qquad \frac{(4\Delta+1)^m (2\Delta)^t}{t^{3/2}}$

- Number of scenarios: $(2\Delta)^t$
- Number of bad scenarios: $\frac{(4\Delta+1)^m(2\Delta)^t}{t^{3/2}} = o((2\Delta)^t)$

End of the proof

$$\underset{(2\Delta)^t/t^{3/2}}{1 \text{ record}} + \underset{(4\Delta+1)^m}{1 \text{ partial coloring}} = \underset{\frac{(4\Delta+1)^m(2\Delta)^t}{t^{3/2}}}{1 \text{ bad scenario}}$$

- Number of scenarios: $(2\Delta)^t$
- Number of bad scenarios: $\frac{(4\Delta+1)^m(2\Delta)^t}{t^{3/2}} = o((2\Delta)^t)$

$\Rightarrow$ For t large enough, there are good scenarios.

$\Leftrightarrow$ The algorithm stops with nonzero probability !

Conclusion

Theorem Esperet and P., 2012

If G has maximum degree Δ and girth g:

- $a'(G) \leq 4\Delta$;
- if $g \geq 7$, $a'(G) \leq 3.74\Delta$;
- if $g \geq 53$, $a'(G) \leq 3.14\Delta$;
- if $g \geq 220$, $a'(G) \leq 3.05\Delta$.

- Procedure in expected polynomial time using $(4+\epsilon)\Delta$ colors.
- Holds also for list coloring.
- Can be applied for any coloring with "forbidden" configurations.

Conclusion

Theorem Esperet and P., 2012

If G has maximum degree Δ and girth g:

- $a'(G) \leq 4\Delta$;
- if $g \geq 7$, $a'(G) \leq 3.74\Delta$;
- if $g \geq 53$, $a'(G) \leq 3.14\Delta$;
- if $g \geq 220$, $a'(G) \leq 3.05\Delta$.

- Procedure in expected polynomial time using $(4+\epsilon)\Delta$ colors.
- Holds also for list coloring.
- Can be applied for any coloring with "forbidden" configurations.

Thanks !