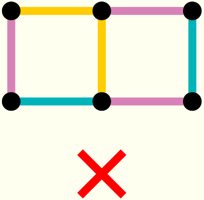
Coloration acyclique des arêtes d'un graphe en utilisant la compression d'entropie

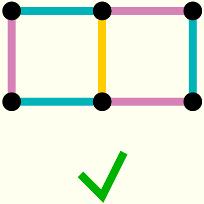
Louis Esperet (G-SCOP, Grenoble, France)
Aline Parreau (LIFL, Lille, France)

Séminaire Algorithmique Distribuée et Graphes du LIAFA
Mardi 18 décembre 2012

Proper Edge Colorings of graphs

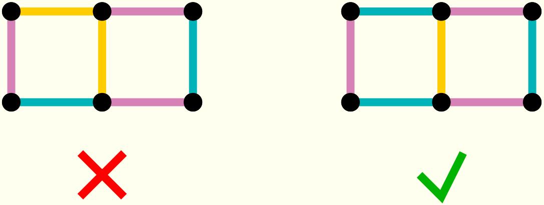
A proper edge coloring of a graph is a coloring of the edges such that two edges sharing a vertex have different colors.





Proper Edge Colorings of graphs

A proper edge coloring of a graph is a coloring of the edges such that two edges sharing a vertex have different colors.

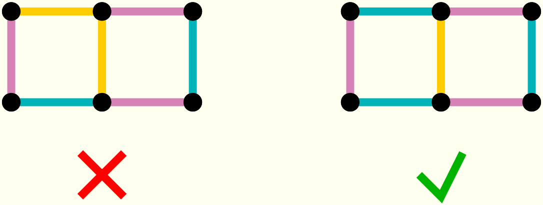


- $\chi'(G)$: minimum number of colors in a proper edge coloring of G.
- If G has maximum degree Δ:

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

Proper Edge Colorings of graphs

A proper edge coloring of a graph is a coloring of the edges such that two edges sharing a vertex have different colors.



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

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

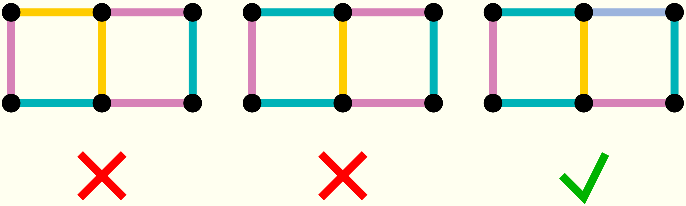
Theorem Vizing, 1964

If G has maximum degree Δ, $\chi'(G) \leq \Delta + 1$.

Acyclic edge coloring of graphs

An acyclic edge coloring of a graph is:

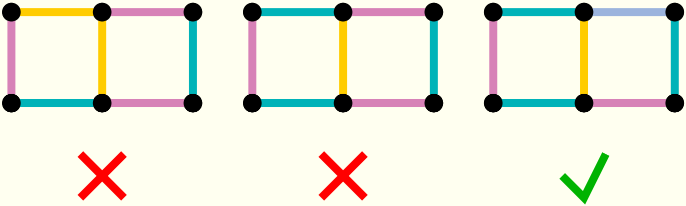
- a proper edge coloring,
- without bicolored cycles.



Acyclic edge coloring of graphs

An acyclic edge coloring of a graph is:

- a proper edge coloring,
- without bicolored cycles.

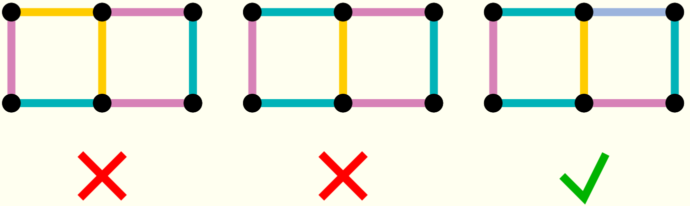


- $a'(G)$: minimum number of colors in an acyclic edge coloring of G.
- If G has maximum degree Δ, $a'(G) \geq \Delta$.

Acyclic edge coloring of graphs

An acyclic edge coloring of a graph is:

- a proper edge coloring,
- without bicolored cycles.



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

Conjecture Alon, Sudakov and Zaks, 2001

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

Lovász Local Lemma

Theorem Lovász Local Lemma

- $A_1, \ldots, A_k$ 'bad' events, each occurs with small probability,
- each event is independent of almost all the others,

$\Rightarrow$ with nonzero probability, no bad event occurs.

Lovász Local Lemma

Theorem Lovász Local Lemma

- $A_1,\ldots,A_k$ 'bad' events, each occurs with small probability,
- each event is independent of almost all the others,

$\Rightarrow$ with nonzero probability, no bad event occurs.

Acyclic edge coloring:

- Take a uniform random coloring with K colors.
- Bad event: a cycle is bicolored or two adjacent edges have the same color.
- Dependancy: one edge is not in 'too many' cycles.

Results

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)

Results

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)

Using entropy compression :

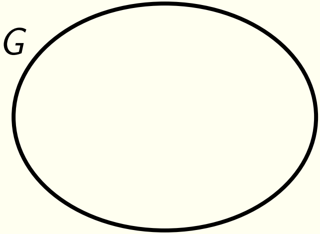
Theorem Esperet and P., 2012

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

Algorithm

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

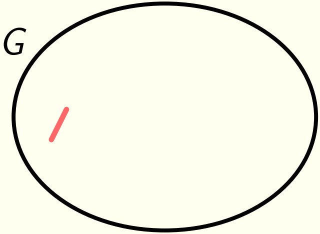
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

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

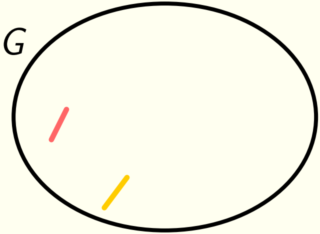
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

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

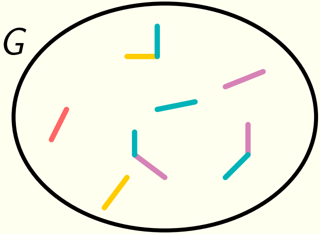
- Select the smallest uncolored edge e
- Give a random color in $\{1, \dots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

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

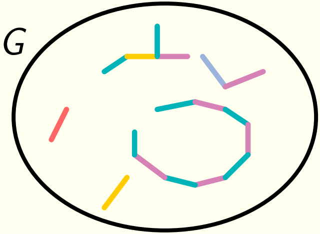
- Select the smallest uncolored edge e
- Give a random color in $\{1, \dots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

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

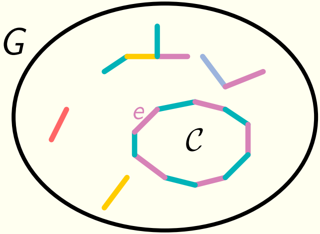
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

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

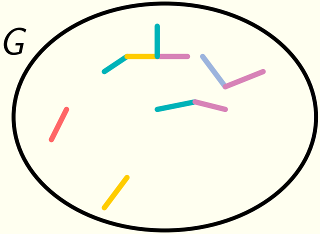
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

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

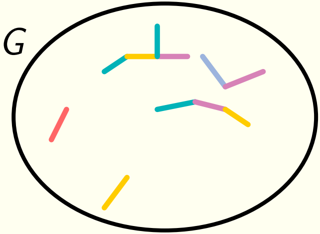
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

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

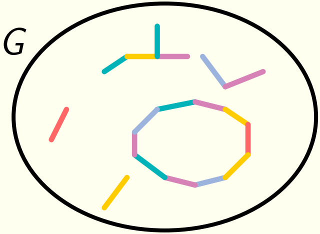
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

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

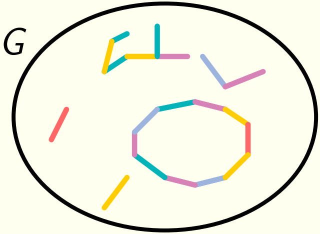
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

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

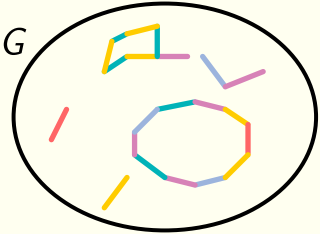
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

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

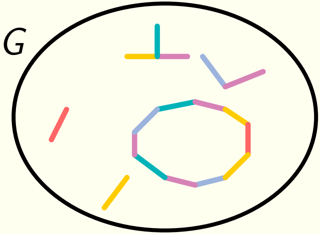
- Select the smallest uncolored edge e
- Give a random color in $\{1, ..., 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



Algorithm

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

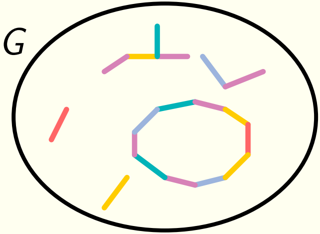
- Select the smallest uncolored edge e
- Give a random color in $\{1, \ldots, 4\Delta\}$ to e (not appearing in $N[e]$)
- If e lies in a bicolored cycle $\mathcal{C}$, uncolor e and all the other edges of $\mathcal{C}$, except two edges.



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.



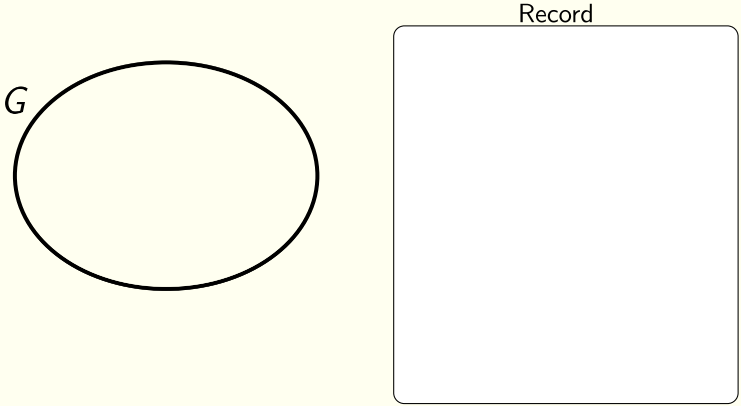
We prove that this algorithm ends with non zero probability.
$\Rightarrow$ Any graph has an acyclic edge coloring with 4Δ colors.

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario

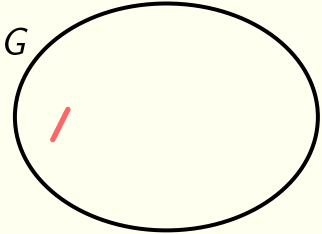
Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.



Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

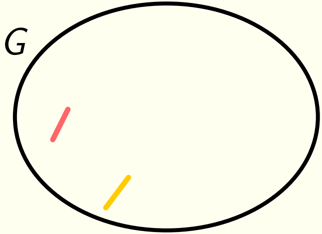


Record

```
1:-
```

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

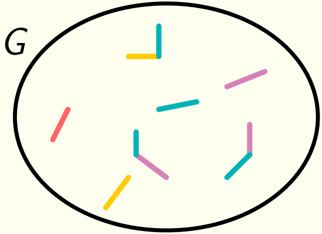


Record

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

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

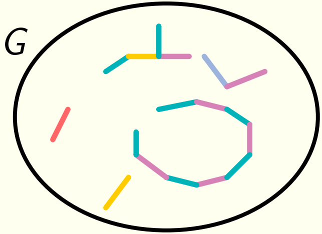


Record

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

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

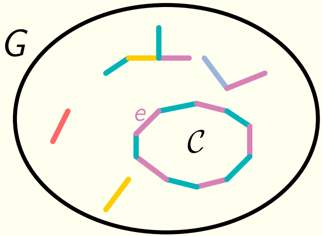


Record

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

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

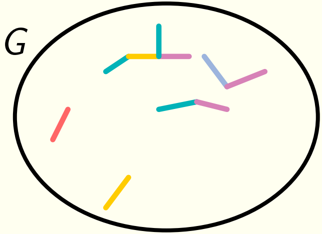


Record

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

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

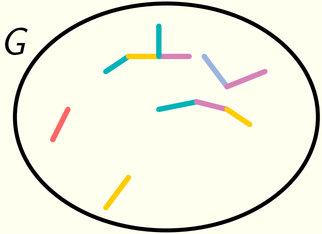


Record

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

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

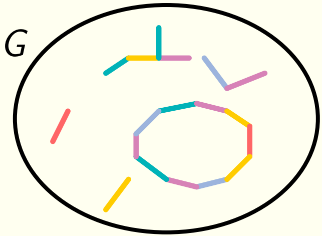


Record

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

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

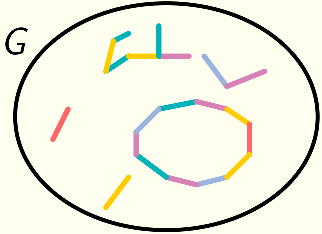


Record

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

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

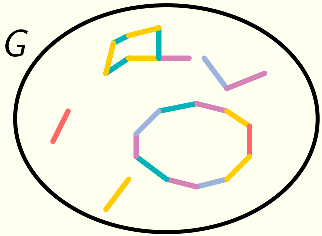


Record

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

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

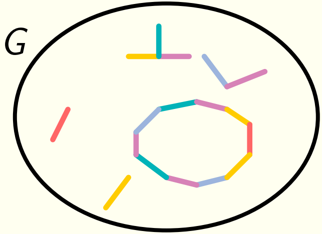


Record

1:-
2:-
...
17:-
18:Cycle $\mathcal{C}$ is uncolored
19:-
...
276:-
277:Cycle $\mathcal{C}'$ is uncolored

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

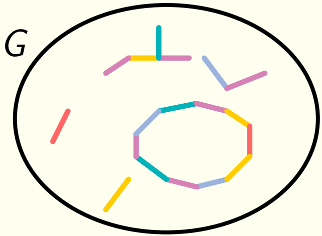


Record

1:-
2:-
...
17:-
18:Cycle $\mathcal{C}$ is uncolored
19:-
...
276:-
277:Cycle $\mathcal{C}'$ is uncolored

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.

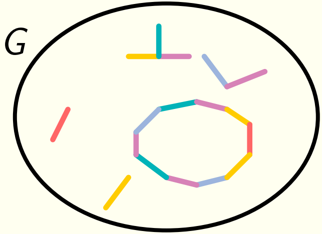


Record

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

Recording

- Execution determined by set of drawn colors : scenario
- Assume the algorithm is still running after t steps. $\rightarrow$ bad scenario
- We record in a compact way what happens during the algorithm.



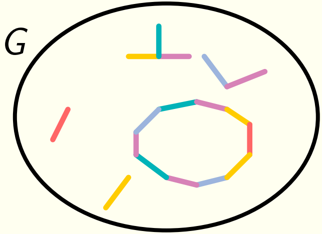
Final partial coloring Φ_t

Record

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

Recording

- Execution determined by set of drawn colors : scenario
- 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 I

- X_i: set of uncolored edges after step i
- reading of the record to get X_i:
    - $X_0 = V$

Rewrite the history I

- X_i: set of uncolored edges after step i
- reading of the record to get X_i:
  - $X_0 = V$
  - Step i:

2 cases

- `i:-` $\rightarrow$ $X_{i+1} = X_i - \{\text{smallest edge of } X_i\}$
- `i:`$\mathcal{C}$ `is uncolored` $\rightarrow$ $X_{i+1} = X_i + \{\mathcal{C} \text{ except two edges}\}$

Rewrite the history I

- X_i: set of uncolored edges after step i
- reading of the record to get X_i:
  - $X_0 = V$
  - Step i:

2 cases

`i:-` $\rightarrow \quad X_{i+1} = X_i - \{\text{smallest edge of } X_i\}$

`i:`$\mathcal{C}$ `is uncolored` $\rightarrow \quad X_{i+1} = X_i + \{\mathcal{C} \text{ except two edges}\}$

With the record, we can find the edge e_i which is colored at step i.

Rewrite the history II: partial colorings

- Φ_i: partial coloring after step i
- Inverse reading of the record to get Φ_i:
  - Φ_t is known

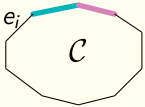
Rewrite the history II: partial colorings

- Φ_i: partial coloring after step i
- Inverse reading of the record to get Φ_i:
  - Φ_t is known
  - Step i:

2 cases

`i:-` $\rightarrow$ $\Phi_{i-1} = \Phi_i$ with e_i uncolored

`i:`$\mathcal{C}$` is uncolored` $\rightarrow$ $\Phi_{i-1} = \Phi_i$ with $\mathcal{C}$ recolored and e_i uncolored



Step i

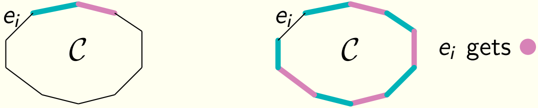
Rewrite the history II: partial colorings

- Φ_i: partial coloring after step i
- Inverse reading of the record to get Φ_i:
  - Φ_t is known
  - Step i:

2 cases

`i:-` $\rightarrow$ $\Phi_{i-1} = \Phi_i$ with e_i uncolored

`i:`$\mathcal{C}$ `is uncolored` $\rightarrow$ $\Phi_{i-1} = \Phi_i$ with $\mathcal{C}$ recolored and e_i uncolored



Step i Step $i-1$

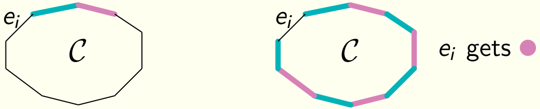
Rewrite the history II: partial colorings

- Φ_i: partial coloring after step i
- Inverse reading of the record to get Φ_i:
  - Φ_t is known
  - Step i:

2 cases

`i:-` $\rightarrow$ $\Phi_{i-1} = \Phi_i$ with e_i uncolored

`i:`$\mathcal{C}$ `is uncolored` $\rightarrow$ $\Phi_{i-1} = \Phi_i$ with $\mathcal{C}$ recolored and e_i uncolored



Step i Step $i-1$

With Φ_t and the record, we can find the partial colorings and the scenario.

Rewrite the history - Summary

1. Top-down reading → set of colored edges at each step.

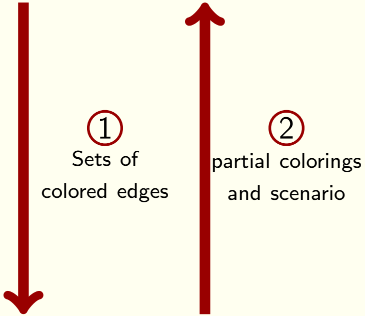
1:-
2:-
...
17:-
18:$\mathcal{C}$ is uncolored
19:-
...
276:-
277:$\mathcal{C}'$ is uncolored
278:-
...
t:-

1
Sets of colored edges

Rewrite the history - Summary

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:-



Rewrite the history - Summary

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

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

Summary

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

Summary

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

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

Summary

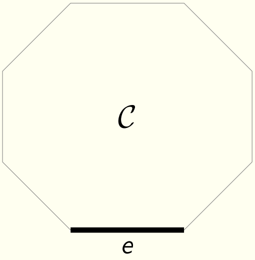
$$1 \text{ record } + 1 \text{ partial coloring} = 1 \text{ bad scenario}$$

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

How many possible records ?

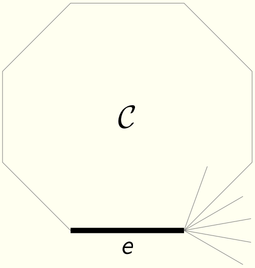
Compact records of cycles

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



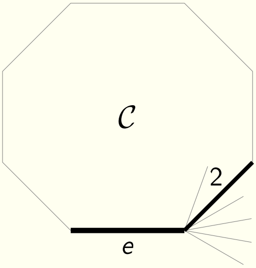
Compact records of cycles

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



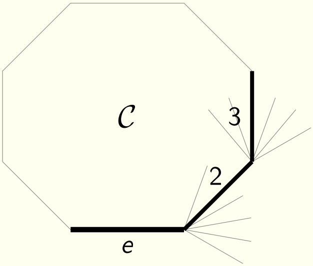
Compact records of cycles

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



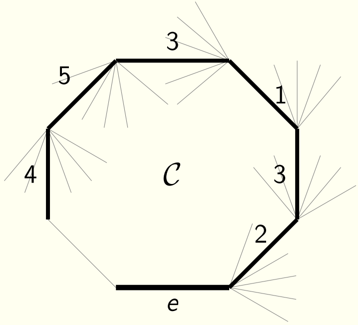
Compact records of cycles

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



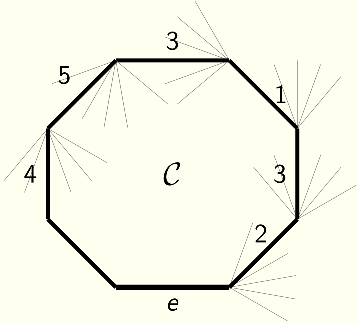
Compact records of cycles

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



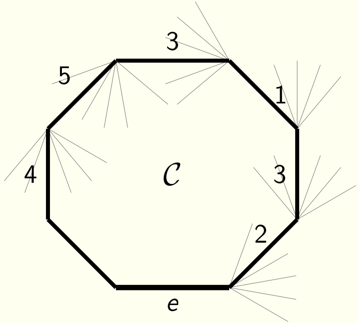
Compact records of cycles

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



Compact records of cycles

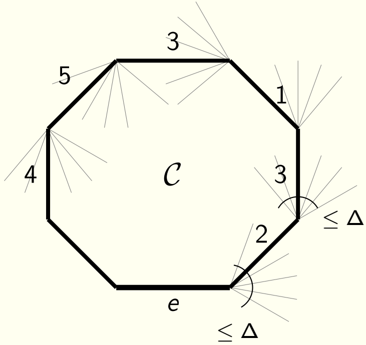
- We know one edge e of $\mathcal{C}$.
- No choice for the last edge



`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

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

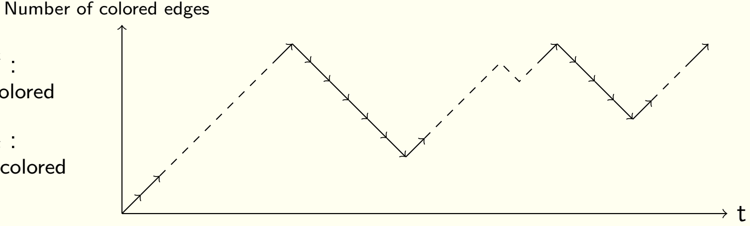
Number of records

$$\text{Record} \qquad (\ -\ ,\ -\ ,\ldots,\ -\ ,\ 231354\ ,\ -\ ,\ldots,\ -\ ,\ 4213\ ,\ -\ ,\ldots,\ -\ )$$

$$\begin{array}{ccccccccc} \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow & \downarrow \\ 0 & 0 & 0 & 0111111 & 0 & 0 & 01111 & 0 & 0 \end{array}$$

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

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



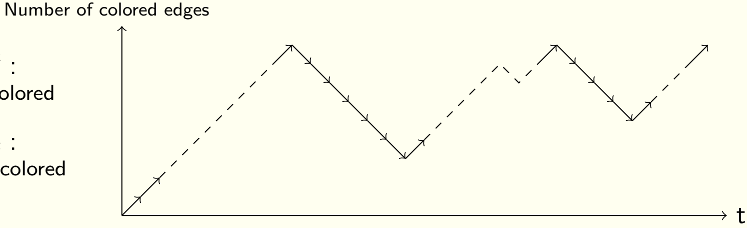
Number of records

Record (− , − , ..., − , 231354 , − , ..., − , 4213 , − , ..., −)

0 0 0 0111111 0 0 01111 0 0

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

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



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

Number of records

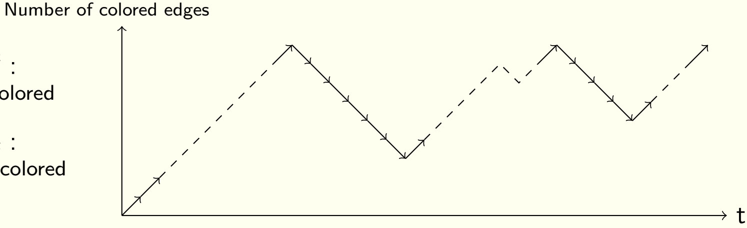
Record $(- , - , \ldots, - , 231354 , - , \ldots, - , 4213 , - , \ldots, -)$

$\downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow \quad \downarrow$

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

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

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



Partial Dyck word of length $\leq 2t$ and descents of even size > 2.

Number of records

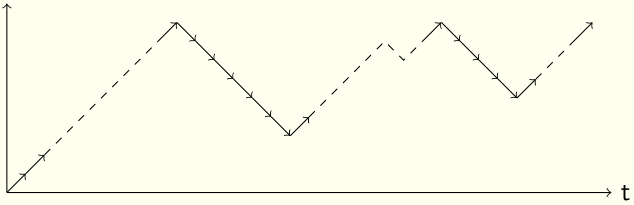
Record $(- , - , ..., - , 231354 , - , ..., - , 4213 , - , ..., -)$

$\downarrow$ 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 descents of even size > 2.

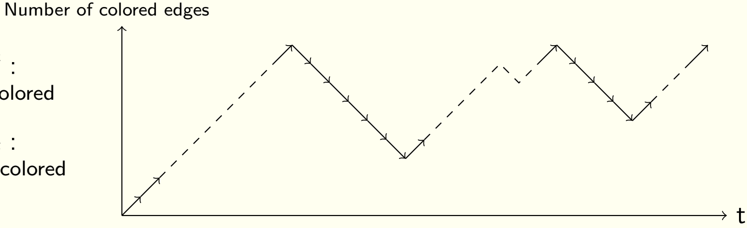
$\rightarrow$ Number of such words : $2^t/t^{3/2}$

Number of records

Record	($-$,	$-$, ...,	$-$,	231354 ,	$-$, ...,	$-$,	4213 ,	$-$, ...,	$-$)
	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$	$\downarrow$
	0	0	0	0111111	0	0	01111	0	0

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

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



Partial Dyck word of length $\leq 2t$ and descents of even size > 2.

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

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

End of the proof

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

$(4\Delta + 1)^m$

End of the proof

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

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

$(4\Delta + 1)^m$

End of the proof

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

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

End of the proof

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

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

- 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

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

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

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

$\Leftrightarrow$ The algorithm stops with nonzero probability !
$\Leftrightarrow$ There is a coloring in 4Δ colors.

Algorithmic aspect

- To have a small propability of a bad event in t steps, we should have:

$$\frac{\text{bad scenarios}}{\text{all scenarios}} = \frac{(4\Delta+1)^m (2\Delta)^t / t^{3/2}}{(2\Delta)^t} < \delta$$

Equivalently:

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

$\rightarrow$ t can be exponential in m.

Algorithmic aspect

- To have a small propability of a bad event in t steps, we should have:

$$\frac{\text{bad scenarios}}{\text{all scenarios}} = \frac{(4\Delta + 1)^m (2\Delta)^t / t^{3/2}}{(2\Delta)^t} < \delta$$

  Equivalently:

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

$\rightarrow$ t can be exponential in m.

- But if we have $(4 + \epsilon)\Delta$ colors :

$$\frac{\text{bad scenarios}}{\text{all scenarios}} = \frac{(4 + \epsilon)\Delta + 1)^m (2\Delta)^t / t^{3/2}}{((2 + \epsilon)\Delta)^t} < \delta$$

Algorithmic aspect

- To have a small propability of a bad event in t steps, we should have:

$$\frac{\text{bad scenarios}}{\text{all scenarios}} = \frac{(4\Delta + 1)^m (2\Delta)^t / t^{3/2}}{(2\Delta)^t} < \delta$$

Equivalently:

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

$\rightarrow$ t can be exponential in m.

- But if we have $(4 + \epsilon)\Delta$ colors :

$$\frac{\text{bad scenarios}}{\text{all scenarios}} = \frac{(4 + \epsilon)\Delta + 1)^m (2\Delta)^t / t^{3/2}}{((2 + \epsilon)\Delta)^t} < \delta$$

$\rightarrow$ t is polynomial in m.

With larger girth

Girth: size of the smallest cycle in G.

Girth $\geq \ell$

$\Leftrightarrow$ All the uncolored cycles have size at least ℓ

$\Leftrightarrow$ All the descents in the Dyck word have size $2k$ for some $k \geq \ell/2$

With larger girth

Girth: size of the smallest cycle in G.

Girth $\geq \ell$

$\Leftrightarrow$ All the uncolored cycles have size at least ℓ

$\Leftrightarrow$ All the descents in the Dyck word have size $2k$ for some $k \geq \ell/2$

There are fewer Dyck words !

Counting Dyck Words

- E: set of integers
- $\mathcal{D}_{t,E}$: Dyck words of length $2t$ with descents in E
- Couting $\mathcal{D}_{t,E} \Leftrightarrow$ counting plane rooted trees on $t+1$ vertices where each vertex as a number of children in E.

Generating function $f(x) = \sum_t \mathcal{D}_{t,E} x^t$:

$$f(x) = x + x \sum_{i \in E} f(x)^i$$

$\Rightarrow$ Using analytic combinatorics (Flageolet and Sedgwick, 2009), the asymptotic behaviour of $\mathcal{D}_{t,E}$ is $\gamma_E^t t^{-3/2}$.

Some results

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 $g \geq 220$, $a'(G) \leq 3.05\Delta$.

History of entropy compression

- Moser in 2009 and Moser,Tardos in 2010: constructive proof of LLL.
  $\rightarrow$ Example of SAT with small intersections between clauses.
- Same ideas applied to
  $\rightarrow$ nonrepetitive sequences (Grytczuk, Kozik, Micek, 2012)
  $\rightarrow$ nonrepetitive coloring (Dujmović, Joret, Kozik, Wood, 2012)

History of entropy compression

- Moser in 2009 and Moser,Tardos in 2010: constructive proof of LLL.
  $\rightarrow$ Example of SAT with small intersections between clauses.
- Same ideas applied to
  $\rightarrow$ nonrepetitive sequences (Grytczuk, Kozik, Micek, 2012)
  $\rightarrow$ nonrepetitive coloring (Dujmović, Joret, Kozik, Wood, 2012)

Entropy compression ?

- Input: large random vector
- Output: smaller record

History of entropy compression

- Moser in 2009 and Moser,Tardos in 2010: constructive proof of LLL.
  $\rightarrow$ Example of SAT with small intersections between clauses.
- Same ideas applied to
  $\rightarrow$ nonrepetitive sequences (Grytczuk, Kozik, Micek, 2012)
  $\rightarrow$ nonrepetitive coloring (Dujmović, Joret, Kozik, Wood, 2012)

Entropy compression ?

- Input: large random vector
- Output: smaller record

Works well since :

- we can remove a lot of colors ($\rightarrow$ add entropy);
- while being able to recover the coloring (give compact record).

Generalization

Coloring with *forbidden configuration* H_i:

- H_i graph with a specific coloring c_i;
- For any vertex v of H_i, there are k_i fixed vertices that determines c_i,
- $\ell_i = |E(H_i)| - k_i$ (number of vertices that will be uncolored), $E = \{\ell_i\}$
- d_ℓ : max number of configurations H_i, $\ell_i = \ell$, containing a vertex ;
- Bound :

$$\gamma_E \times \sup d_\ell^{1/\ell}$$

Example: star coloring, bound in $3\sqrt{2}\Delta^{3/2}$.

Generalization

Coloring with *forbidden configuration H_i*:

- H_i graph with a specific coloring c_i;
- For any vertex v of H_i, there are k_i fixed vertices that determines c_i,
- $\ell_i = |E(H_i)| - k_i$ (number of vertices that will be uncolored),
  $E = \{\ell_i\}$
- d_ℓ : max number of configurations H_i, $\ell_i = \ell$, containing a vertex ;
- Bound :

$$\gamma_E \times \sup d_\ell^{1/\ell}$$

Example: star coloring, bound in $3\sqrt{2}\Delta^{3/2}$.

Thanks !