Forbidden structures for efficient First-Fit chain partitioning

Bartłomiej Bosek Tomasz Krawczyk (speaker)
and Grzegorz Matecki

Theoretical Computer Science
Jagiellonian University

{bosek, krawczyk, matecki}@tcs.uj.edu.pl

2012 SIAM Conference on Discrete Mathematics
Halifax, 18-21 June, 2012

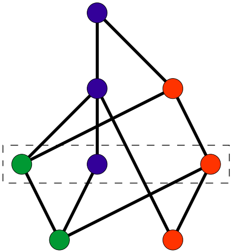
chain partitioning

- $P = (X, \leqslant)$
- antichain, $\mathrm{w}(P)$
- chain
- a chain partition of $P = (X, \leqslant)$ is a family of disjoint chains $C_1, \ldots, C_k$ so that $C_1 \cup \ldots \cup C_k = X$

Theorem (Dilworth 1950)

Every poset P can be partitioned into $\mathrm{w}(P)$ chains.

Example



on-line chain partitioning

- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:
  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning

- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:

  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning

- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:

  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning

- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:
  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning

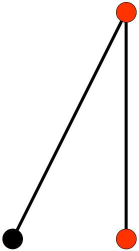
- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:

  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning



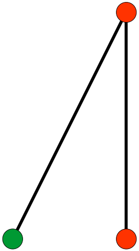
- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:

  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning



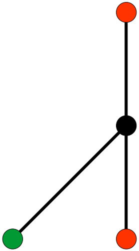
- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:

  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning



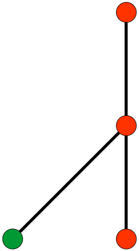
- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:

  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning

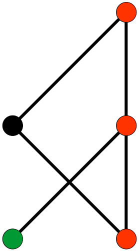


- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:
  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning

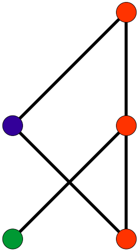


- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:
  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning

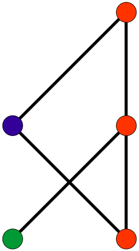


- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:
  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning



- on-line result = 3 colors

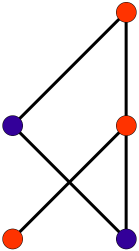
- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:

  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning



- on-line result = 3 colors
- off-line result = 2 colors

- two players: Spoiler and Algorithm,
- the game is played in rounds,

  during each round:
  - Spoiler introduces a new point with his comparability status to already presented points,
  - Algorithm assigns this new point to a chain.

- Algorithm tries to use as small number of colors as possible,
- Spoiler tries to force Algorithm to use as many colors as possible.

on-line chain partitioning

Estimate:

ocp(w) – the minimum n such that Algorithm has a strategy to partition any poset of width w into at most n chains

on-line chain partitioning

Estimate:

ocp(w) – the minimum n such that Algorithm has a strategy to partition any poset of width w into at most n chains

Theorem (Szemerédi 1981; Kierstead 1981)

$$\binom{w+1}{2} \leqslant \mathrm{ocp}(w) \leqslant \frac{5^w - 1}{4}$$

on-line chain partitioning

Estimate:

ocp(w) – the minimum n such that Algorithm has a strategy to partition any poset of width w into at most n chains

Theorem (Szemerédi 1981; Kierstead 1981)

$$\binom{w+1}{2} \leqslant \mathrm{ocp}(w) \leqslant \frac{5^w - 1}{4}$$

$$\mathrm{ocp}(w) \leqslant \mathit{poly}(w)?$$

first-fit

first-fit – on-line algorithm that always uses the lowest possible number

first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2.

first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2*.*

first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

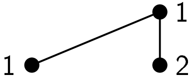
There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2.

first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2*.*

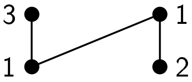


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2*.*

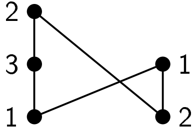


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2.

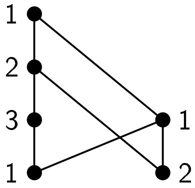


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2.

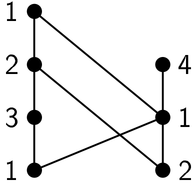


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2*.*

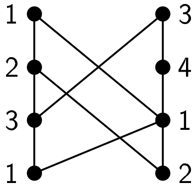


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2*.*

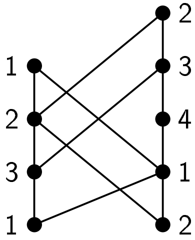


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2*.*

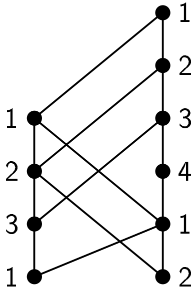


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2.

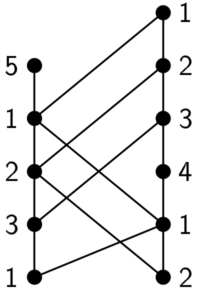


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2*.*

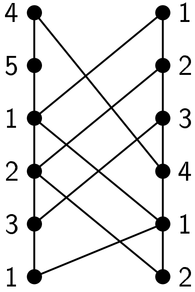


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2*.*

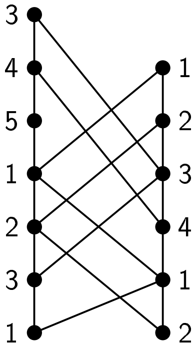


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2*.*

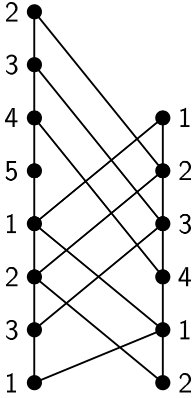


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2*.*

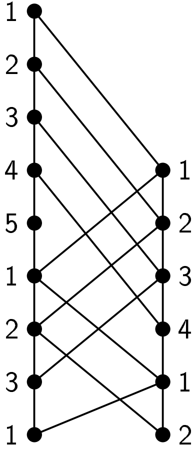


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2*.*

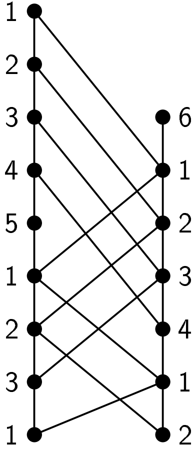


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2.

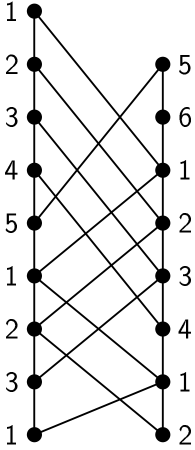


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2.

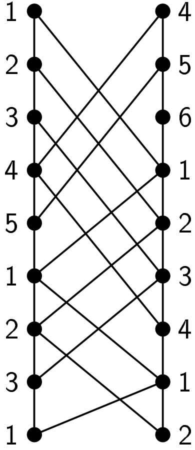


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2.

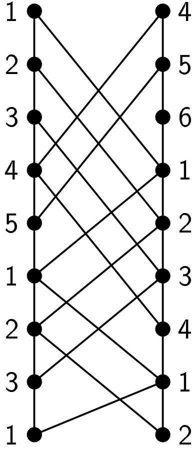


first-fit

first-fit – on-line algorithm that always uses the lowest possible number

Theorem (Kierstead 1986)

There exists a strategy for Spoiler that forces First-Fit to use infinitely many colors even if the game is played on orders of width 2.



Remarks

- *First-Fit works well for some classes of posets (e.g. interval posets)*
- *is used as a subroutine in on-line algorithms that give subexponential bounds on* ocp(w).

first-fit

$f_Q(w)$ – the maximum number of chains used by first-fit on $Q-$free posets of width w

on-line chain partitioning

Theorem (Bosek, Krawczyk 2009)

$$\mathrm{ocp}(w) \leqslant w^{13 \log w}$$

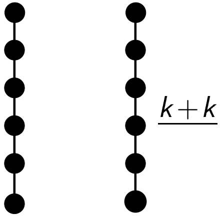
on-line chain partitioning

Theorem (Bosek, Krawczyk 2009)

$$\operatorname{ocp}(w) \leqslant w^{13 \log w}$$

Remarks

- *First-Fit on $\underline{2w+2w}-$free posets is used as a subroutine*
- *First-Fit works efficiently on $\underline{k+k}-$free posets!*



First-Fit for $\underline{k+k}$-free posets

Upper bounds on $f_{\underline{k+k}}(w)$:

- $3kw^2$, Bosek, Krawczyk, Szczypka 2008
- $8k^2w$, Joret, Milans 2010
- $16kw$, Dujmović, Joret, Wood 2011 (G. Joret talk).

Lower bounds on $f_{\underline{k+k}}(w)$:

- $(k-1)(w-1)$, Dujmović, Joret, Wood 2011.

on-line chain partitioning

Theorem (2011)

$$\operatorname{ocp}(w) \leqslant w^{3+6.5\log w}$$

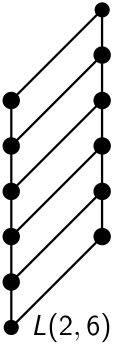
on-line chain partitioning

Theorem (2011)

$$\mathrm{ocp}(w) \leqslant w^{3+6.5\log w}$$

Remark

First-Fit on $L(2, 2w^2)-$*free posets is used as a subroutine*



on-line chain partitioning

Theorem (2011)

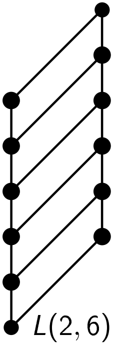
$$\mathrm{ocp}(w) \leqslant w^{3+6.5\log w}$$

Remark

First-Fit on $L(2, 2w^2)$−free posets is used as a subroutine

Proof:

- $\mathrm{ocp}(w) \leqslant w \cdot f_{L(2,2w^2)}(w)$,
  Bosek, Krawczyk 2009
- $f_{L(2,m)}(w) \leqslant w^{2.5\log w + 2\log m}$,
  Kierstead, M.Smith 2011 (M. Smith talk)
- subexponential bound on $\mathrm{ocp}(w)$ is best possible in this approach, Bosek, Matecki 2012



$L(2, 6)$

question

Question (Joret, Milans 2010)

For which posets Q there is a function f_Q such that First-Fit partitions $Q-$free posets of width w into at most $f_Q(w)$ chains.

question

Question (Joret, Milans 2010)

For which posets Q there is a function f_Q such that First-Fit partitions $Q-$free posets of width w into at most $f_Q(w)$ chains.

Remarks

- *Q must be a poset of width 2,*

question

Question (Joret, Milans 2010)

For which posets Q there is a function f_Q such that First-Fit partitions $Q-$free posets of width w into at most $f_Q(w)$ chains.

Remarks

- *Q must be a poset of width 2,*
- *f_Q exists for:*

question

Question (Joret, Milans 2010)

For which posets Q there is a function f_Q such that First-Fit partitions $Q-$free posets of width w into at most $f_Q(w)$ chains.

Remarks

- *Q must be a poset of width 2,*
- *f_Q exists for:*
  - $\underline{k+k}$: $f_{\underline{k+k}}(w) \leqslant 16kw$,

question

Question (Joret, Milans 2010)

For which posets Q there is a function f_Q such that First-Fit partitions $Q-$free posets of width w into at most $f_Q(w)$ chains.

Remarks

- *Q must be a poset of width 2,*
- *f_Q exists for:*
  - $\underline{k+k}$: $f_{\underline{k+k}}(w) \leqslant 16kw$,
  - $L(2,m)$: $f_{L(2,m)} \leqslant w^{2.5 \log w + 2 \log m}$,

question

Question (Joret, Milans 2010)

For which posets Q there is a function f_Q such that First-Fit partitions $Q-$free posets of width w into at most $f_Q(w)$ chains.

Remarks

- *Q must be a poset of width 2,*
- *f_Q exists for:*
  - $\underline{k+k}$: $f_{\underline{k+k}}(w) \leqslant 16kw$,
  - $L(2,m)$: $f_{L(2,m)} \leqslant w^{2.5 \log w + 2 \log m}$,
  - $L(2,2)$: $f_{L(2,2)}(w) = w^2$ *(tight result: Matecki 2011; Kierstead, M.Smith 2011)*

main result

Theorem (Bosek, Matecki, Krawczyk 2010)

Let Q be a poset. There exists a function f_Q such that First-Fit partitions Q-free posets of width w into at most $f_Q(w)$ chains iff Q is a poset of width 2.

main result

Theorem (Bosek, Matecki, Krawczyk 2010)

Let Q be a poset. There exists a function f_Q such that First-Fit partitions Q-free posets of width w into at most $f_Q(w)$ chains iff Q is a poset of width 2.

Remark

If First-Fit uses unlimited number of chains on a (countable) poset P of width w then P contains all width 2 posets.

idea of the proof - ladders (1)

Ladders:

- universal posets of width 2, parameterized by two variables

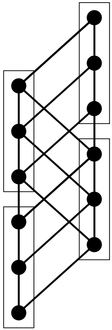
idea of the proof - ladders (1)

Ladders:

- universal posets of width 2, parameterized by two variables

Lemma

For any poset Q of width 2 there exist sufficiently large s, t such that Q is contained in $L(s, t)$.

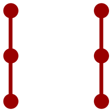


$L(4, 3)$

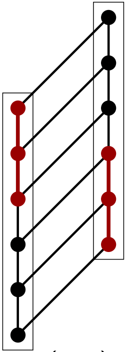
idea of the proof - ladders (2)

Lemma

For any poset Q of width 2 there exist sufficiently large s, t such that Q is contained in $L(s, t)$.



$3 + 3$

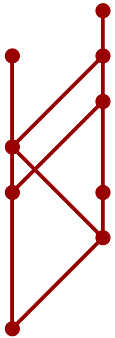


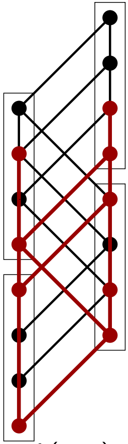
$L(2, 6)$

idea of the proof - ladders (2)

Lemma

For any poset Q of width 2 there exist sufficiently large s, t such that Q is contained in $L(s, t)$.



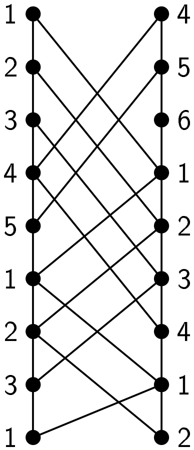


$L(4, 4)$

the Kierstead's example

Remark

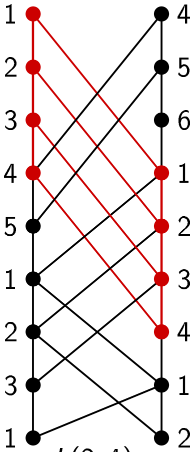
The poset from the Kierstead's example must contain all posets of width 2.



the Kierstead's example

Remark

The poset from the Kierstead's example must contain all posets of width 2.

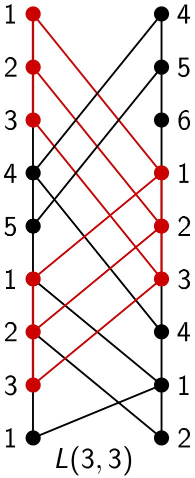


$L(2,4)$

the Kierstead's example

Remark

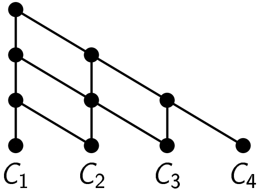
The poset from the Kierstead's example must contain all posets of width 2.



idea of the proof - wall

Definition

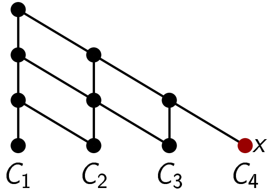
*Let $(P, \leqslant)$ be a poset. An ordered chain partition $C_1, C_2, \ldots, C_k$ of $(P, \leqslant)$ is a **wall** of size k if for any $x \in C_i$ and for any chain C_j, $j < i$, there is a point in C_j that is incomparable to x.*



idea of the proof - wall

Definition

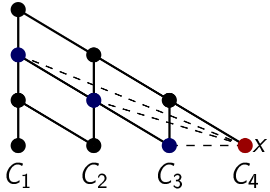
*Let $(P, \leqslant)$ be a poset. An ordered chain partition $C_1, C_2, \ldots, C_k$ of $(P, \leqslant)$ is a **wall** of size k if for any $x \in C_i$ and for any chain C_j, $j < i$, there is a point in C_j that is incomparable to x.*



idea of the proof - wall

Definition

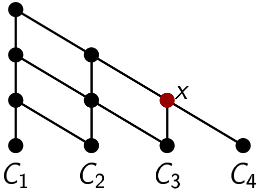
*Let $(P, \leqslant)$ be a poset. An ordered chain partition $C_1, C_2, \ldots, C_k$ of $(P, \leqslant)$ is a **wall** of size k if for any $x \in C_i$ and for any chain C_j, $j < i$, there is a point in C_j that is incomparable to x.*



idea of the proof - wall

Definition

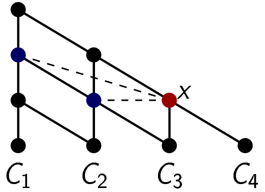
*Let $(P, \leqslant)$ be a poset. An ordered chain partition $C_1, C_2, \dots, C_k$ of $(P, \leqslant)$ is a **wall** of size k if for any $x \in C_i$ and for any chain C_j, $j < i$, there is a point in C_j that is incomparable to x.*



idea of the proof - wall

Definition

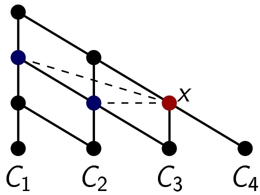
*Let $(P, \leqslant)$ be a poset. An ordered chain partition $C_1, C_2, \ldots, C_k$ of $(P, \leqslant)$ is a **wall** of size k if for any $x \in C_i$ and for any chain C_j, $j < i$, there is a point in C_j that is incomparable to x.*



idea of the proof - wall

Definition

*Let $(P, \leqslant)$ be a poset. An ordered chain partition $C_1, C_2, \ldots, C_k$ of $(P, \leqslant)$ is a **wall** of size k if for any $x \in C_i$ and for any chain C_j, $j < i$, there is a point in C_j that is incomparable to x.*



Remark

The maximum number of colors used by First-Fit on $(P, \leqslant)$ is equal to the maximum size of a wall in $(P, \leqslant)$.

- *First-Fit coloring of $(P, \leqslant)$ in the order $C_1 < C_2 < \ldots < C_k$ uses k colors,*
- *a partition $C_1, \ldots, C_k$ produced by first-fit is a wall in $(P, \leqslant)$.*

idea of the proof

We show that for every $s, t > 1$ the following holds:

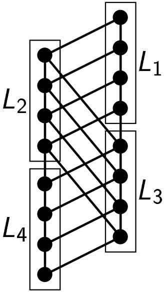
- any wall of width $w \geq 2$ and 'sufficiently large' size (i.e. first-fit uses a lot of colors) contains an $(s, t)-$ladder.

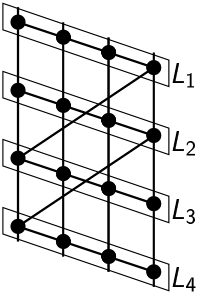
idea of the proof

We show that for every $s, t > 1$ the following holds:

- any wall of width $w \geq 2$ and 'sufficiently large' size (i.e. first-fit uses a lot of colors) contains an $(s, t)-$ladder.

- we 'are looking' for the 'wall-style' ladders



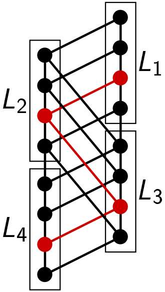


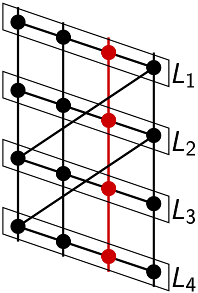
idea of the proof

We show that for every $s, t > 1$ the following holds:

- any wall of width $w \geq 2$ and 'sufficiently large' size (i.e. first-fit uses a lot of colors) contains an $(s, t)-$ladder.

- we 'are looking' for the 'wall-style' ladders





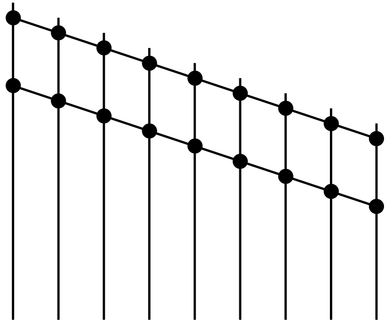
idea of the proof

First part:

idea of the proof

First part:

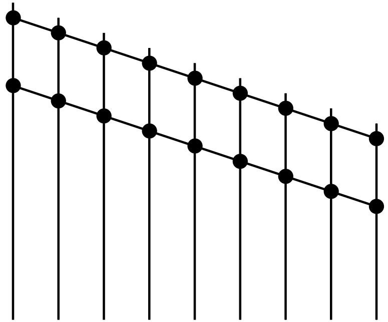
- every 'sufficiently large' wall of width $w \geq 2$ contains $L(2, t)$ ladder (M. Smith talk)



idea of the proof

First part:

- every ‘sufficiently large’ wall of width $w \geq 2$ contains $L(2, t)$ ladder (M. Smith talk)



Second part:

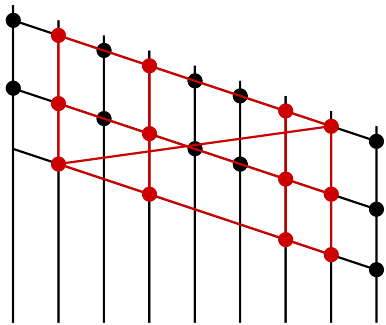
idea of the proof

First part:

- every 'sufficiently large' wall of width $w \geq 2$ contains $L(2,t)$ ladder (M. Smith talk)

Second part:

- for every s, if $L(2,t)$ is large enough, then we may attach another level to $L(2,t)$ to get $L(3,s)$



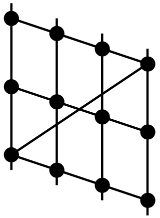
idea of the proof

First part:

- every 'sufficiently large' wall of width $w \geq 2$ contains $L(2, t)$ ladder (M. Smith talk)

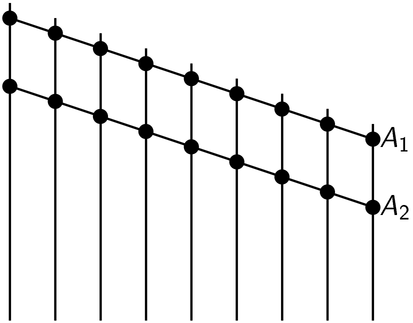
Second part:

- for every s, if $L(2, t)$ is large enough, then we may attach another level to $L(2, t)$ to get $L(3, s)$

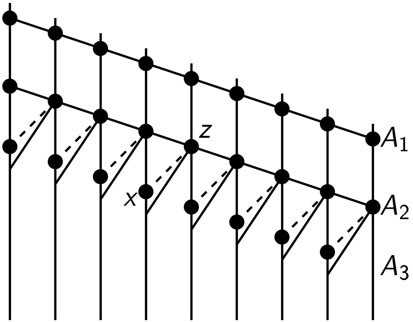


We focus on $L(3, 4)$.

idea of the proof



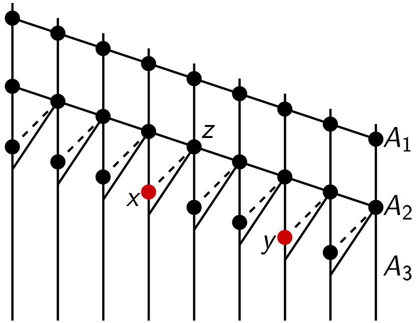
idea of the proof



idea of the proof

- x, y - two points from A_3, x is to the left of y:

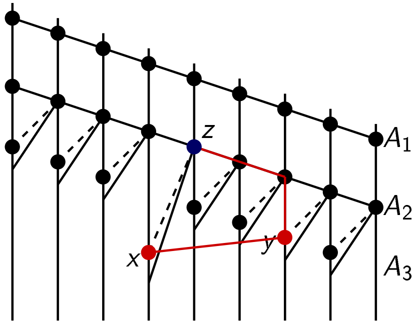
$$x || y \text{ or } x \geq y$$



idea of the proof

- x, y - two points from A_3, x is to the left of y:

$$x || y \text{ or } x \geq y$$

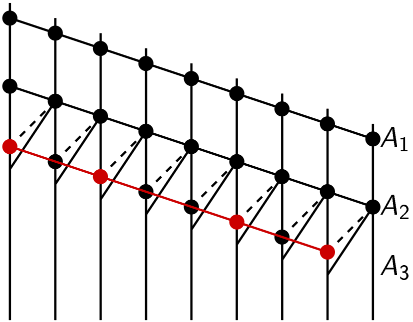


idea of the proof

- x, y - two points from A_3, x is to the left of y:

$$x || y \text{ or } x \geq y$$

- there is a large decreasing chain C in A_3

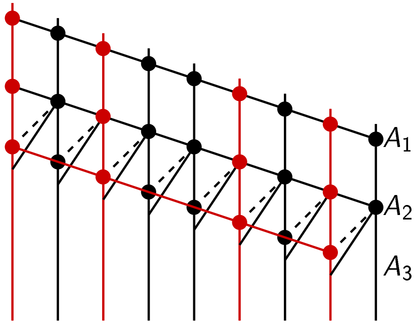


idea of the proof

- x, y - two points from A_3,
  x is to the left of y:

  $$x || y \text{ or } x \geq y$$

- there is a large decreasing chain C in A_3
- choose the chains from the wall that have some elements in C

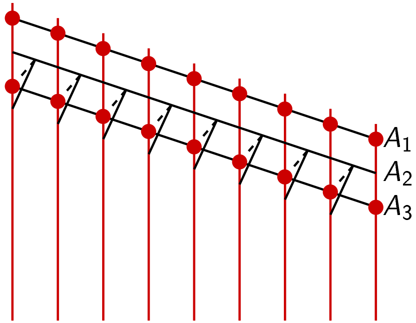


idea of the proof

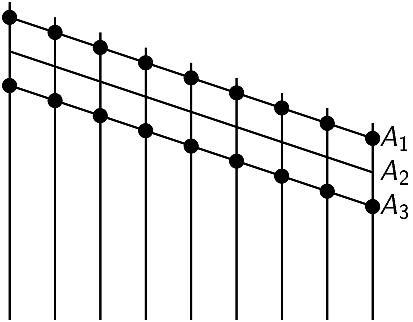
- x, y - two points from A_3, x is to the left of y:

$$x || y \text{ or } x \geq y$$

- there is a large decreasing chain C in A_3
- choose the chains from the wall that have some elements in C
- C can be arbitrarily large!



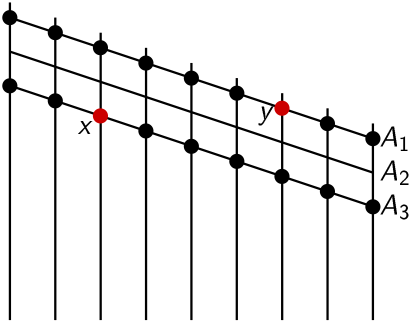
idea of the proof



idea of the proof

- $x \in A_3$, $y \in A_1$, x is 'far' to the left of y:

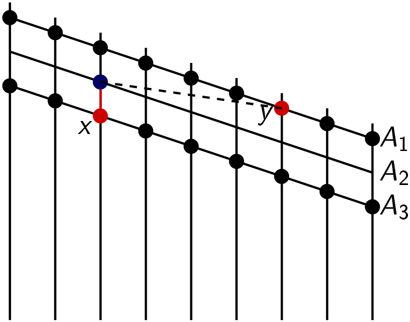
$$x || y$$



idea of the proof

- $x \in A_3$, $y \in A_1$, x is 'far' to the left of y:

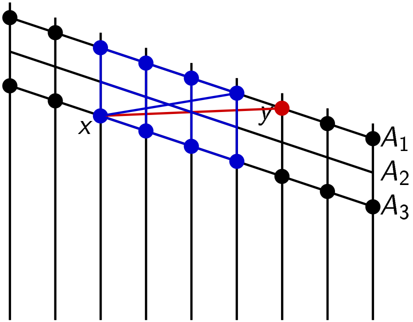
$$x || y$$



idea of the proof

- $x \in A_3$, $y \in A_1$, x is ‘far’ to the left of y:

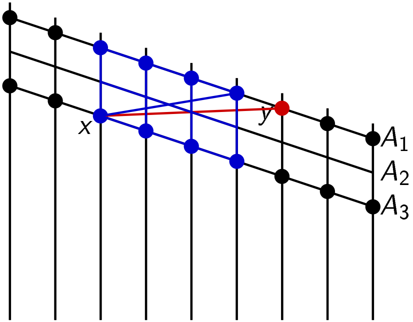
$$x || y$$



idea of the proof

- $x \in A_3$, $y \in A_1$, x is 'far' to the left of y:

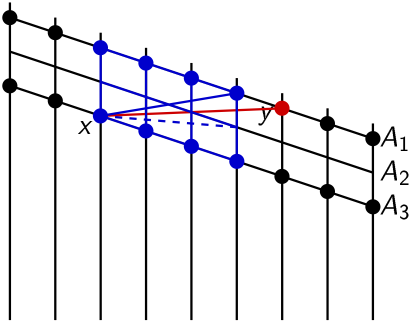
$$x || y$$



idea of the proof

- $x \in A_3$, $y \in A_1$, x is 'far' to the left of y:

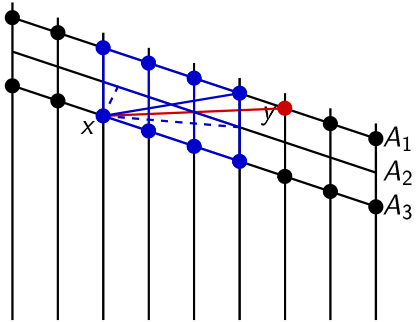
$$x || y$$



idea of the proof

- $x \in A_3$, $y \in A_1$, x is 'far' to the left of y:

$$x || y$$

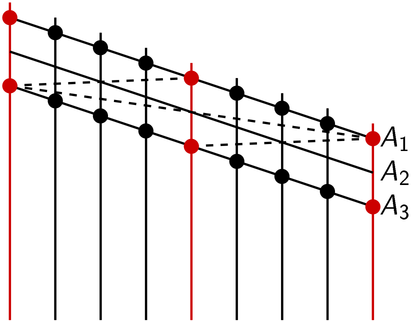


idea of the proof

- $x \in A_3$, $y \in A_1$, x is 'far' to the left of y:

$$x || y$$

- choose every 4−th chain of the wall

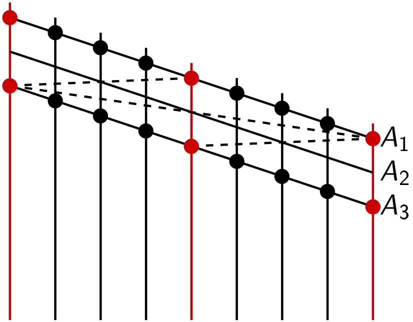


idea of the proof

- $x \in A_3$, $y \in A_1$, x is ‘far’ to the left of y:

$$x || y$$

- choose every 4−th chain of the wall
- chosen points from A_1 and A_3 form $L(2, *)$

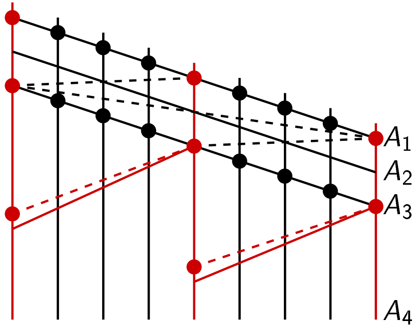


idea of the proof

- $x \in A_3$, $y \in A_1$, x is 'far' to the left of y:

$$x || y$$

- choose every 4−th chain of the wall
- chosen points from A_1 and A_3 form $L(2,*)$
- choose the set A_4

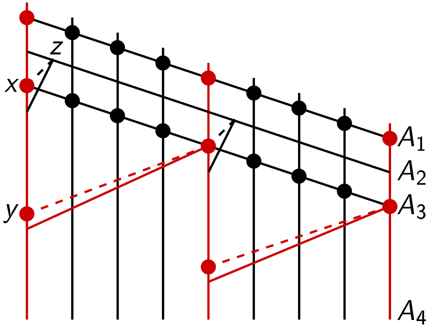


idea of the proof

- $x \in A_3$, $y \in A_1$, x is 'far' to the left of y:

$$x || y$$

- choose every 4−th chain of the wall
- chosen points from A_1 and A_3 form $L(2, *)$
- choose the set A_4
- each element in A_4 is below some element in A_2

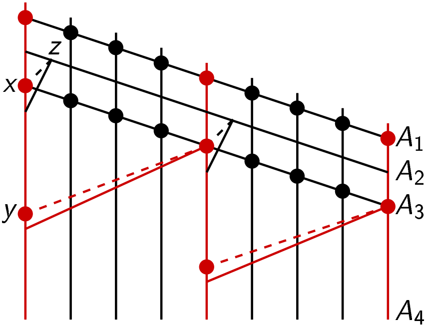


idea of the proof

- $x \in A_3$, $y \in A_1$, x is 'far' to the left of y:

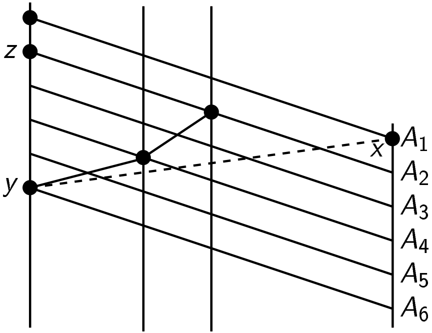
$$x || y$$

- choose every 4−th chain of the wall
- chosen points from A_1 and A_3 form $L(2, *)$
- choose the set A_4
- each element in A_4 is below some element in A_2
- we construct the sets A_3, A_4, A_5, A_6, e.t.c.



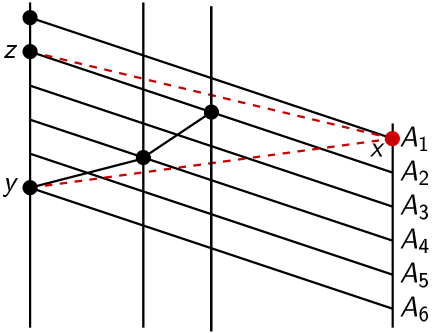
idea of the proof

- consider $L(2,*)$ formed by A_1 and A_6



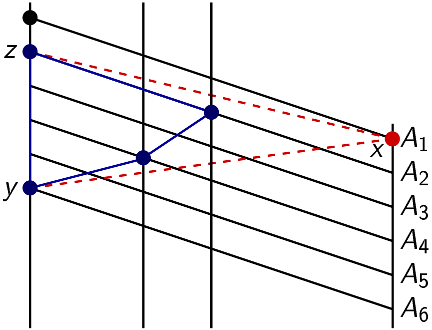
idea of the proof

- consider $L(2, *)$ formed by A_1 and A_6
- x is incomparable to z



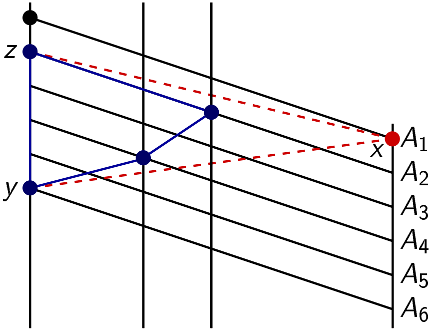
idea of the proof

- consider $L(2,*)$ formed by A_1 and A_6
- x is incomparable to z
- x is incomparable to $y\Uparrow \cap z\Downarrow$



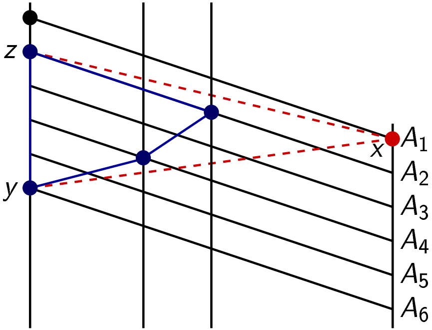
idea of the proof

- consider $L(2, *)$ formed by A_1 and A_6
- x is incomparable to z
- x is incomparable to $y{\Uparrow} \cap z{\Downarrow}$
- $y{\Uparrow} \cap z{\Downarrow}$ induces a subwall of width $w - 1$



idea of the proof

- consider $L(2,*)$ formed by A_1 and A_6
- x is incomparable to z
- x is incomparable to $y\Uparrow \cap z\Downarrow$
- $y\Uparrow \cap z\Downarrow$ induces a subwall of width $w-1$
- by induction, if the wall $y\Uparrow \cap z\Downarrow$ is large enough then it contains $L(3,4)$ ladder



open problems

We have:

open problems

We have:

- a subexponential bound on $f_{L(2,m)}$, M. Smith, Kierstead 2011

open problems

We have:

- a subexponential bound on $f_{L(2,m)}$, M. Smith, Kierstead 2011
- a bound on $f_{L(s,m)}$, Bosek, Krawczyk, Matecki 2010

open problems

We have:

- a subexponential bound on $f_{L(2,m)}$, M. Smith, Kierstead 2011
- a bound on $f_{L(s,m)}$, Bosek, Krawczyk, Matecki 2010
- there is no constant c such that $f_Q(w) \leqslant w^c$ for every poset Q of width 2, Bosek, Matecki 2012

Open problems:

open problems

We have:

- a subexponential bound on $f_{L(2,m)}$, M. Smith, Kierstead 2011
- a bound on $f_{L(s,m)}$, Bosek, Krawczyk, Matecki 2010
- there is no constant c such that $f_Q(w) \leqslant w^c$ for every poset Q of width 2, Bosek, Matecki 2012

Open problems:

- find posets Q for which $f_Q(w) \leqslant poly(w)$

open problems

We have:

- a subexponential bound on $f_{L(2,m)}$, M. Smith, Kierstead 2011
- a bound on $f_{L(s,m)}$, Bosek, Krawczyk, Matecki 2010
- there is no constant c such that $f_Q(w) \leqslant w^c$ for every poset Q of width 2, Bosek, Matecki 2012

Open problems:

- find posets Q for which $f_Q(w) \leqslant poly(w)$
- prove a subexponential bound on $f_{L(3,m)}$ and then for $f_{L(s,m)}$ (and hence for every Q of width 2)

Thank You!