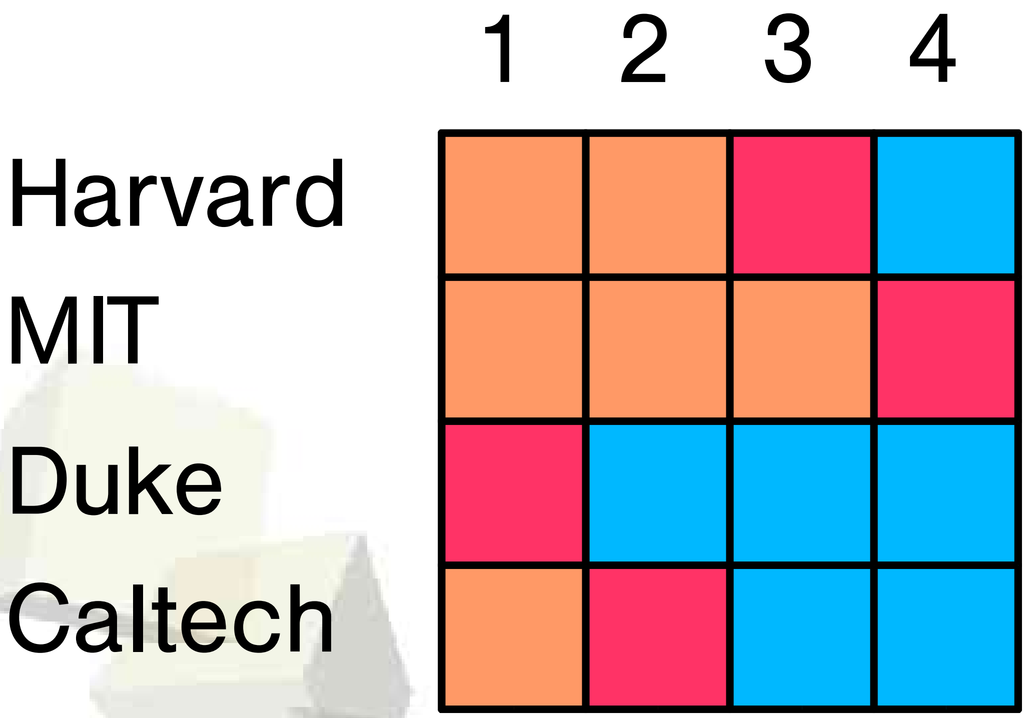
Time Dependent Update Functions for Perfect Sampling

Mark Huber
Dept. of Mathematics and Inst. of Statistics and Decision Sciences
Duke University
mhuber@math.duke.edu
www.math.duke.edu/~mhuber

Results for the Putnam Mathematical Competition in 2000



Facts:

- Harvard came in 3 (or better)
- Caltech came in 2 (or better)
- Duke came in first

Simple idea can form the basis for algorithms for generating weighted permutations

Each permutation x given weight $\mu(x)$

Goal generate random variates from

$$\pi(x) = \frac{\mu(x)}{\sum_y \mu(y)}$$

In this talk

- A weighted permutation problem
- Perfect sampling with CFTP
- Time dependent update functions
- Bounding chains
- BC for weighted permutations
  - does not work with original CFTP
- Solving the permutation problem
- A continuous example

The Goal

Generate uniformly from the set of random linear extensions of a poset

$$\mu(x)=\begin{cases}1 & \text{if } x \text{ is a linear extension}\\ 0 & \text{otherwise}\end{cases}$$

Linear Extensions

Let $N = \{1, 2, \ldots, n\}$

A partial order $\leqslant$ on N is

1) Reflexive $a \leqslant a$
2) Antisymmetric $a \leqslant b$ and $b \leqslant a$ implies $a = b$
3) Transitive $a \leqslant b$ and $b \leqslant c$ implies $a \leqslant c$

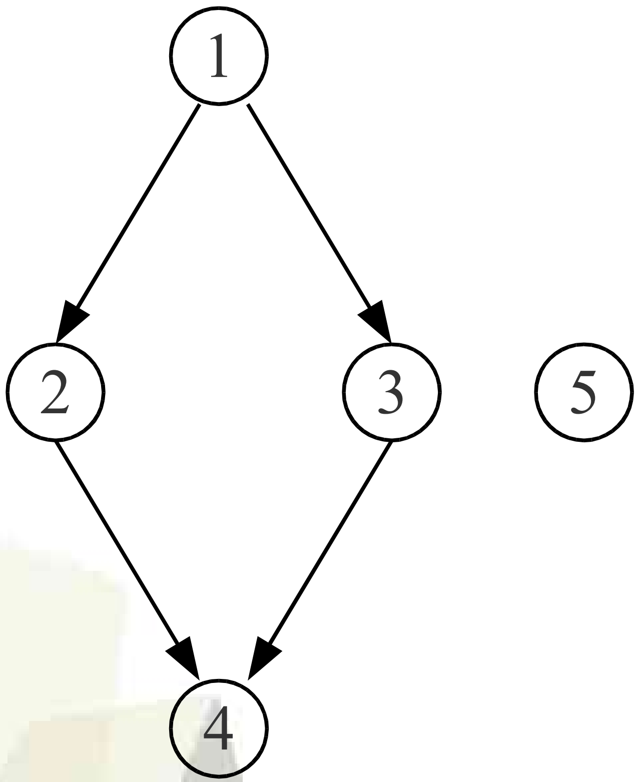
A linear extension is a permutation x where

$$a \leqslant b \text{ implies } x(a) \leqslant x(b)$$

Suppose throughout $1,2,\ldots,n$ is a linear ext.

Example

$n = 5$



Some Linear Extensions

1 2 3 4 5

1 3 2 4 5

1 2 3 5 4

1 3 2 5 4

1 2 5 3 4

1 3 5 2 4

⋮

Say item 3 is in position 2

Questions

Question: How many linear extensions?

Bad news: #P complete

[Brightwell, Winkler 1991]

Good news: selfreducible so generating samples leads to efficient (approx) counting

[Jerrum, Valiant, Vazirani 1986]

Question: Average # of inversions?

Variant of nonparametric Kendall's tau test

[Efron, Petrosian 1999]

Markov chain approach

Describe Markov chain via update function

$$f : \Omega \times [0,1] \to \Omega$$

$$U_1, U_2, .. \sim \mathrm{Unif}[0,1] \quad \text{(iid)}$$

$$X_{t+1} = f(X_t, U_{t+1})$$

Computer simulation of Markov chains
(use pseudorandom numbers)

Also known as
transition function
stochastic recursive scheme

Markov chain approach

One step in linear extensions chain

$y = f(x, U)$

1] **Choose** i **uniformly from** $\{1, 2, \ldots n-1\}$

2] **Choose** B **uniformly from** $\{0, 1\}$

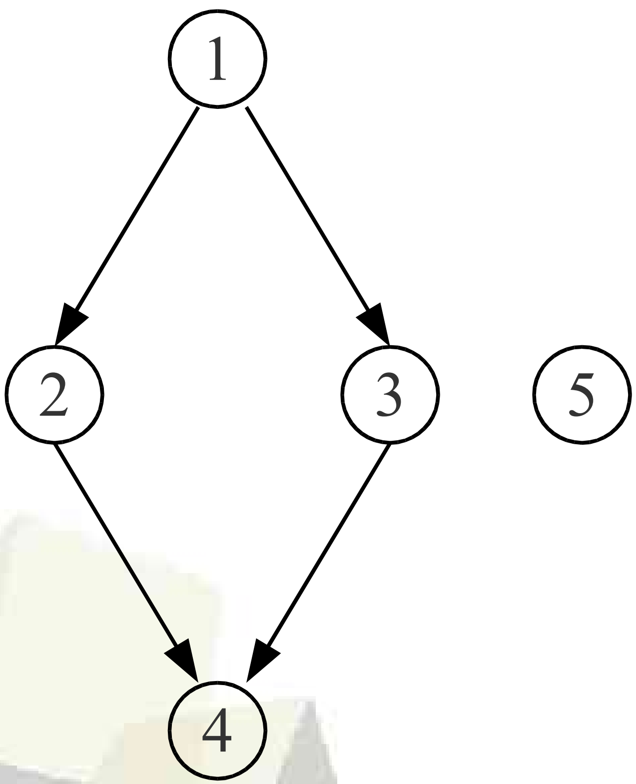
3] **Let** $y \leftarrow x$

4] **If** $x(i) \leqslant x(i+1)$ or $B = 0$ **do nothing**

Else $y(i+1) \leftarrow x(i),\ y(i) \leftarrow x(i+1)$

(Pick random position, flip fair coin,
if coin heads and does not violate partial order
then swap items at position and one to the right)

Example steps



State	Randomness
1 2 5 3 4	$i=2, B=1$
1 5 2 3 4	$i=3, B=1$
1 5 2 3 4	$i=1, B=0$
1 5 2 3 4	

What's known

[Bubley, Dyer 1999] Different Markov chain mixes $O(n^3 \ln(n/\epsilon))$ steps

[Wilson 2004] Original chain mixes $O(n^3 \ln(n/\epsilon))$

[Felsner, Wernich 1997] Perfect sampling when poset two dimensional

This work: perfect sampling arbitrary poset

$$O(n^3 \ln(n))$$

Perfect Sampling Algorithm

A perfect sampling algorithm has 3 properties:

1) Generates exact random variates from the target distribution

2) Running time is random with exponential tails

$$P(T > 2kE[T]) < (1/2)^k$$

3) No knowledge of the normalizing constant (good since we do not know $\mu(\Omega)$)

Comments

1) Usually samples are only approximately from target distribution

2) Running time very unlikely to be much larger than expected value (otherwise Dyer showed any approximate sampler is a perfect sampler)

3) When $\mu(\Omega)$ known it is a direct sampler

The Good News

- Generates exactly from desired distribution
- Can be used for continuous or discrete
- True algorithms
  (Markov chain methods are not algorithms unless the mixing time is known)
- Useful even if running time unknown

The Bad News

- Not a magic solution to slow Markov chains
- Requires more effort than Metroplis-Hastings
- Methods more complex

Coupling From the Past

CFTP [Propp, Wilson '96]

Ingredients: update function for chain,

$$\ldots, U_{-2}, U_{-1}, U_0 \sim \text{Unif}[0,1] \text{ (iid)}$$

1) pretend have unknown stationary rand. var.
2) run chain forward fixed number of steps
3) if state becomes known, output
4) else call CFTP recursively
5) run chain forward fixed number of steps

An example

Example

Start T = -5

If had $X_{-5} \sim \pi$

then $X_{-4} = f(X_{-5}, U_{-4})$

$$X_{-3} = f(X_{-4}, U_{-3})$$

$$X_{-2} = f(X_{-3}, U_{-2})$$

$$X_{-1} = f(X_{-2}, U_{-1})$$

$$X_0 = f(X_{-5}, U_0)$$

output $X_0 \sim \pi$

First level of CFTP

Example

Start T = -5

Suppose I do not know $X_{-5} \sim \pi$, set $Z_{-5} = \Omega$

let $Z_{-4} = f(Z_{-5}, U_{-4})$ and $X_{-4} \in Z_{-4}$

$$Z_{-3} = f(Z_{-4}, U_{-3}) \text{ and } X_{-3} \in Z_{-3}$$

$$Z_{-2} = f(Z_{-3}, U_{-2}) \text{ and } X_{-2} \in Z_{-2}$$

$$Z_{-1} = f(Z_{-2}, U_{-1}) \text{ and } X_{-1} \in Z_{-1}$$

$$Z_0 = f(Z_{-1}, U_0) \text{ and } X_0 \in Z_0$$

if $Z_0 = \{x\}, X_0 = x$

else ....

Second level of CFTP

Go back farther in time…

Start T = -100

Do not know $X_{-100} \sim \pi$, set $Z_{-100} = \Omega$

find

Z_{-5} using $U_{-99}, \ldots, U_{-5}$

if $Z_{-5} = \{y\}, X_{-5} = y$

$X_0 = f(f(f(f(f(X_{-5}, U_{-4}), U_{-3}), U_{-2}), U_{-1}), U_0)$

else ….

Third level of CFTP

Go back farther in time…

Start T = -1000

Do not know $X_{-1000} \sim \pi$, set $Z_{-1000} = \Omega$

then find

Z_{-100} using $U_{-999}, \ldots, U_{-100}$

if $Z_{-100} = \{y\}, X_{-100} = y$

find X_0 using $U_{-99}, \ldots, U_0$

else [keep going back in time until success]

Techniques

How to keep track of Z_t

Monotonicity [Propp, Wilson 1996]

Multigamma coupling [Murdoch, Green 1998]

Bounding chains [Häggström, Nelander 1999], [H. 1999, 2004]

Multishift coupling [Wilson 2000]

Time dependent update functions

Original CFTP

Use same update function every time

$$X_{t+1} = f(X_t, U_{t+1})$$

(is a Markovian coupling)

Time dependent CFTP

Let update function change each step

$$X_{t+1} = f(X_t, U_{t+1}, U_t, U_{t-1}, \ldots)$$

(non-Markovian coupling)

Sufficient conditions on f

Our requirement

Suppose $U \sim \mathrm{Unif}[0,1]$, then

$$P(X_{t+1} \in A \mid X_t = x) = P(f(x, U, u_{1,} u_{2,} \ldots) \in A)$$

for all values of

$$u_{1,} u_{2,} \ldots$$

Proof that Time dependent CFTP works essentially same as original proof

Bounding chains

- Is itself a Markov chain
- Bound each dimension separately
- Keeps set of possible values at each coordinate
- For permuations, keep interval for each item

$$1 \quad 5 \quad 2 \quad 3 \quad 4$$

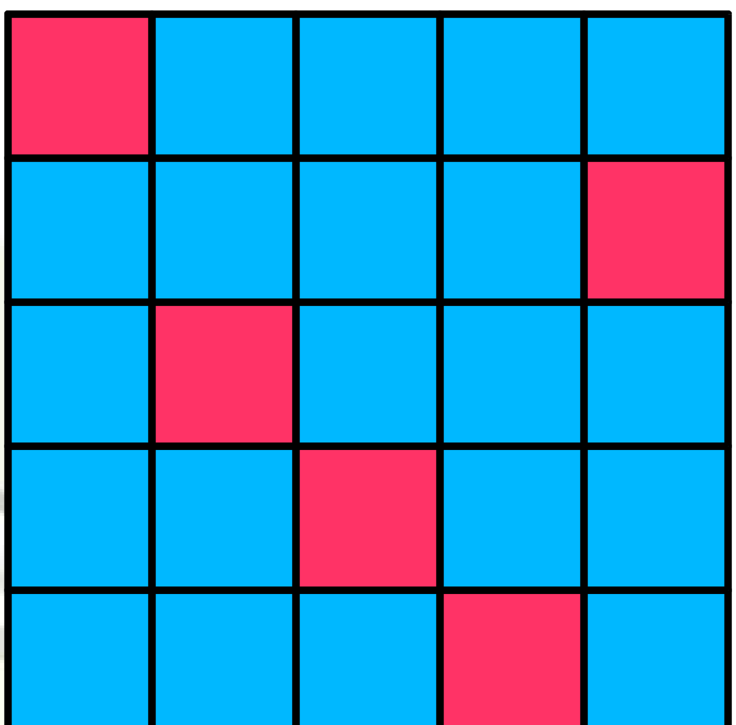
bounded by

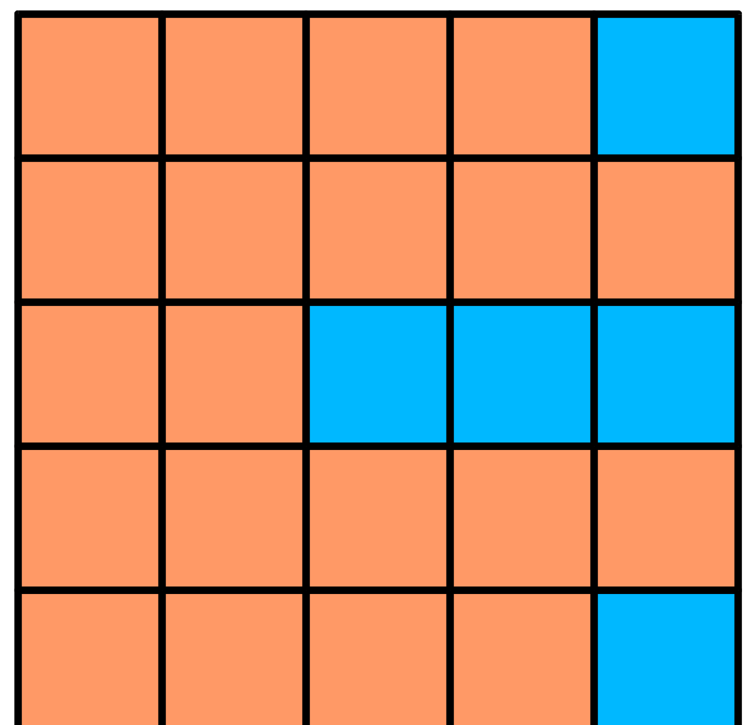
$$\{1,2,3,4\} \ \{1,2,3,4,5\} \ \{1,2\} \ \{1,2,3,4,5\} \ \{1,2,3,4\}$$

Pictorially

$$1 \quad 5 \quad 2 \quad 3 \quad 4$$

$$\{1,2,3,4\}, \{1,2,3,4,5\},$$
$$\{1,2\}, \{1,2,3,4,5\},$$
$$\{1,2,3,4\}$$



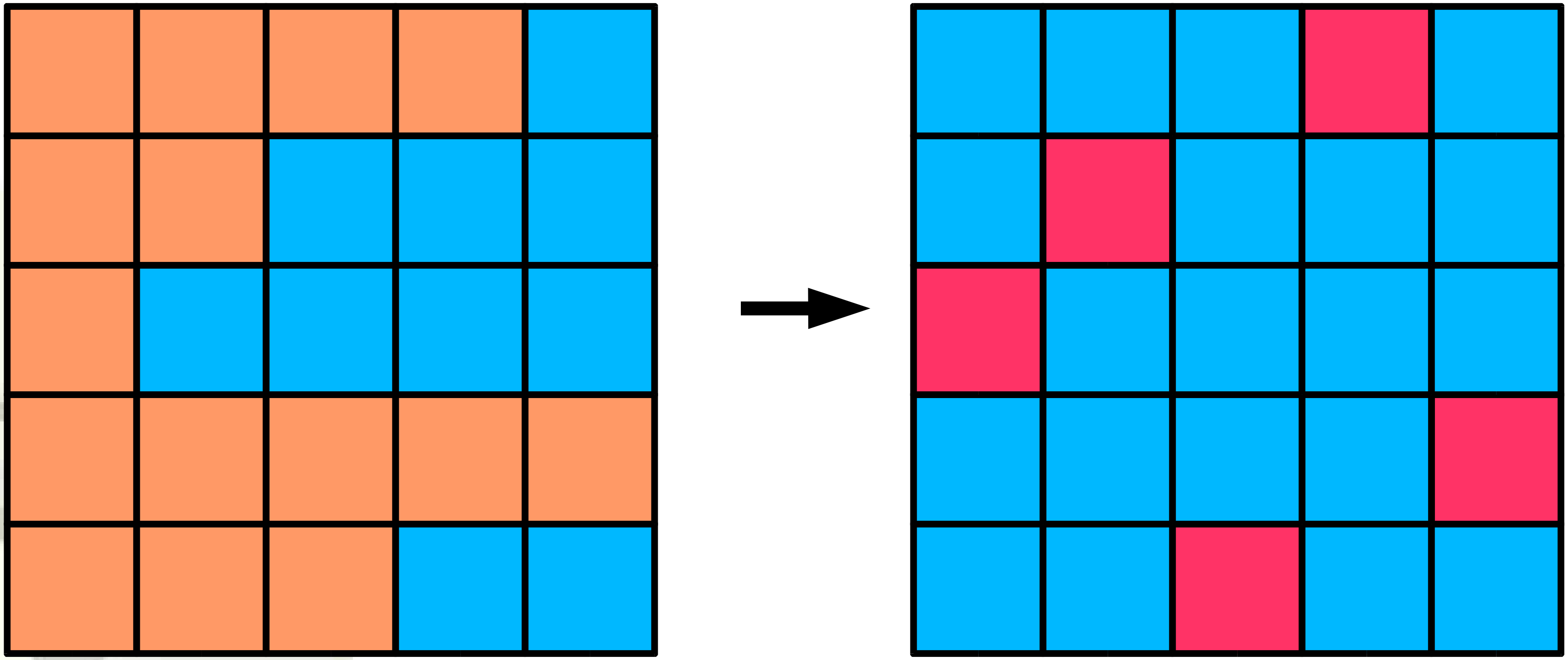


Trick from beginning

When bounding state

$$\{1,\ldots,a_1\},\ldots,\{1,\ldots,a_n\}$$

has $a_{1,}\ldots,a_n$ all different $Z_t = \{x\}$



Running the bounding chain

Suppose bounding state

$$Y_t = \{1,\ldots,a_1\},\ldots,\{1,\ldots,a_n\}$$

want to run the chain forward so that if

$$X_t(i) \in \{1,\ldots,a_i\} \text{ for all } i$$

then

$$X_{t+1}(i) \in \{1,\ldots,a_i\} \text{ for all } i$$

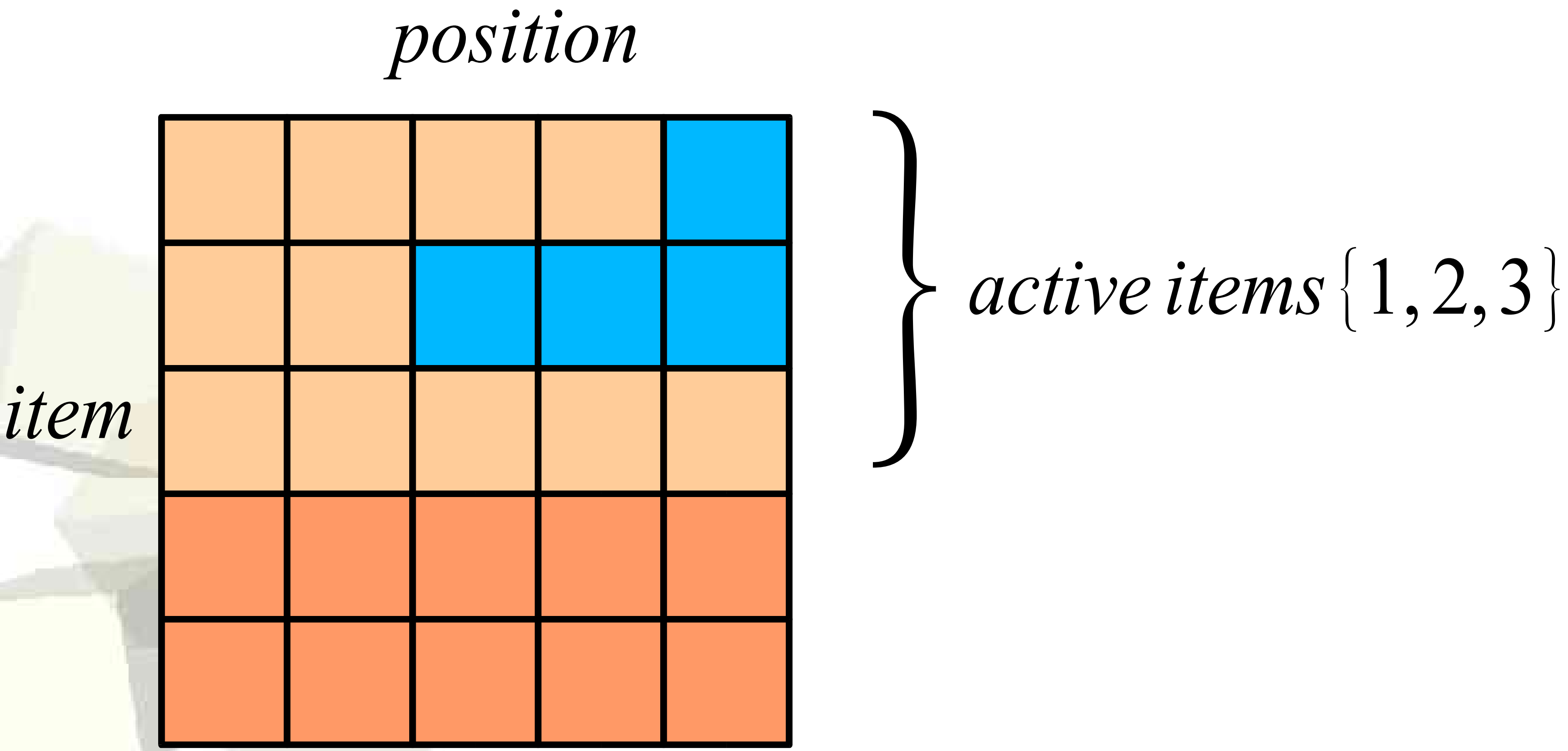
Active items

Call item i active if

$$a_i \notin \{a_{1,} \ldots, a_{i-1}, a_{i+1}, \ldots, a_n\}$$

or

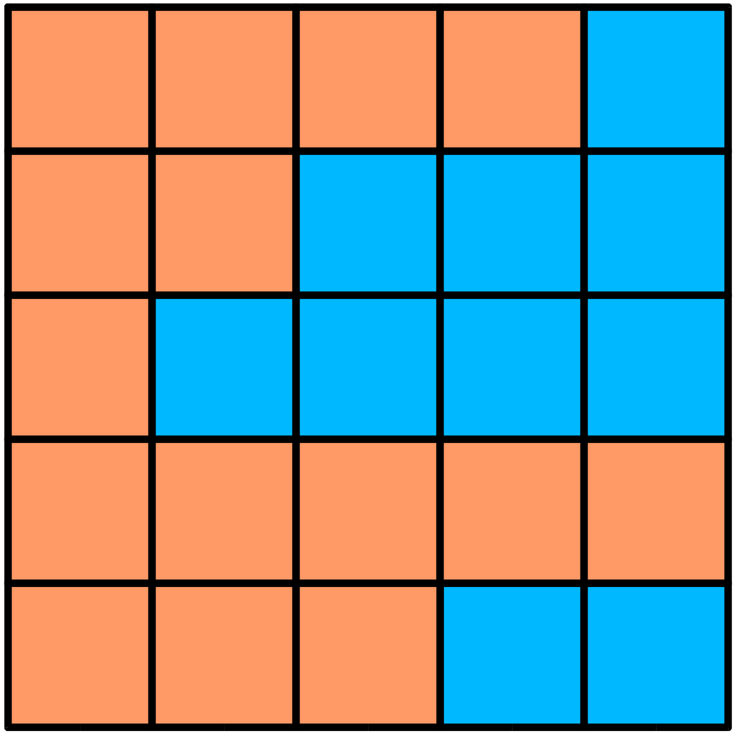
$$a_i = n \text{ and } a_i \notin \{a_{1,} \ldots a_{i-1}\}$$



Why look at active

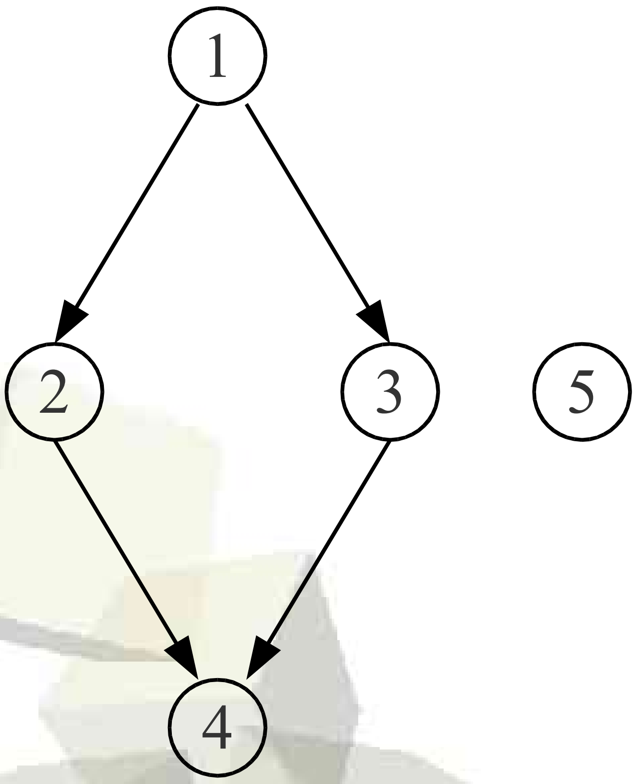
When all items active

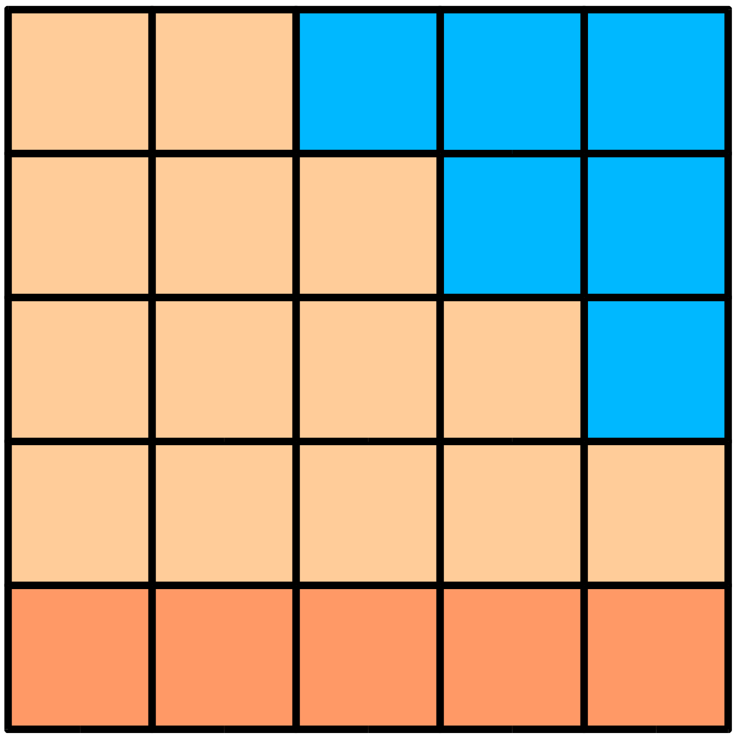
$$Z_t = \{x\}$$



Example starting bounding chain

Since item 1 preceeds 3 items, $X_t(1) \leq 2$



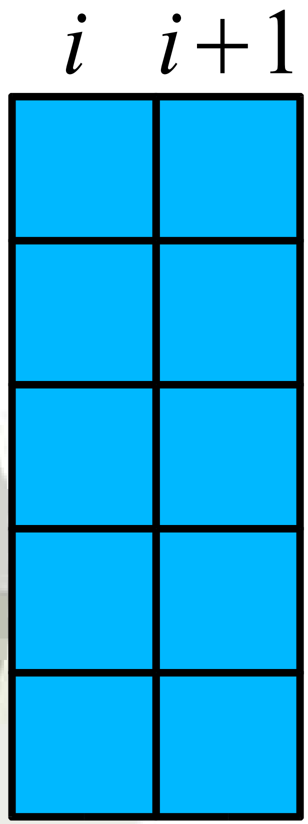


Case by case

Choose $i \sim \text{Unif}\{1,2,\ldots n-1\}$

Choose $B \sim \text{Unif}\{0,1\}$

Case I: no active items at position i or $i+1$



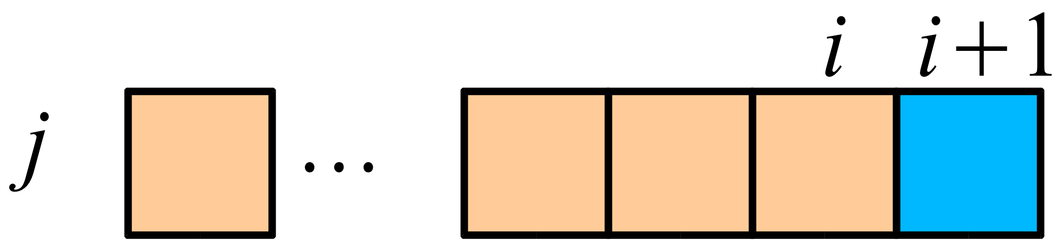
Action

bounding state unchanged

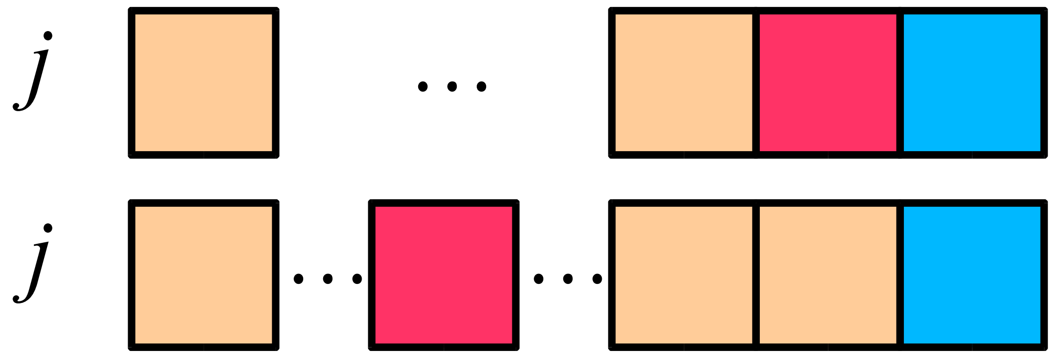
$$Y_{t+1} \leftarrow Y_t$$

Case II

Case II: one active item j at position i



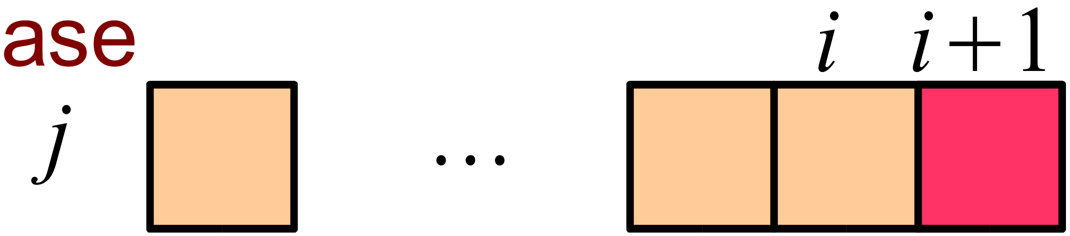
Could be



Worst case $B = 1$

Case II continued

Worst case



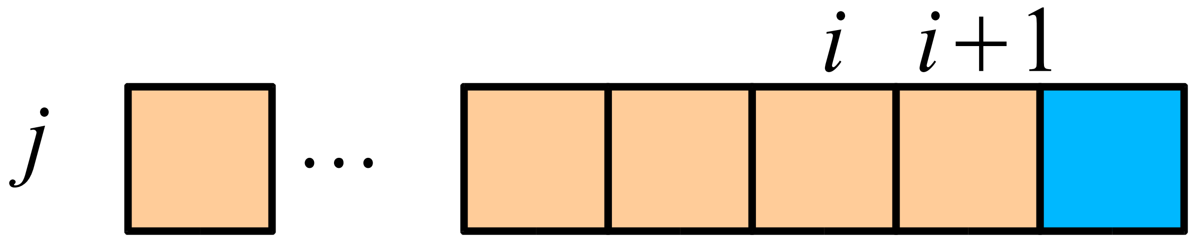
Action

If $B=1$

$$Y_{t+1}(j) \leftarrow Y_t(j)+1$$

Case III

Case III: one active item j at position $i+1$



Note: any other active item j' with

$$j' \leqslant j$$

has

$$Y_t(j') \leq Y_t(j) - 2$$

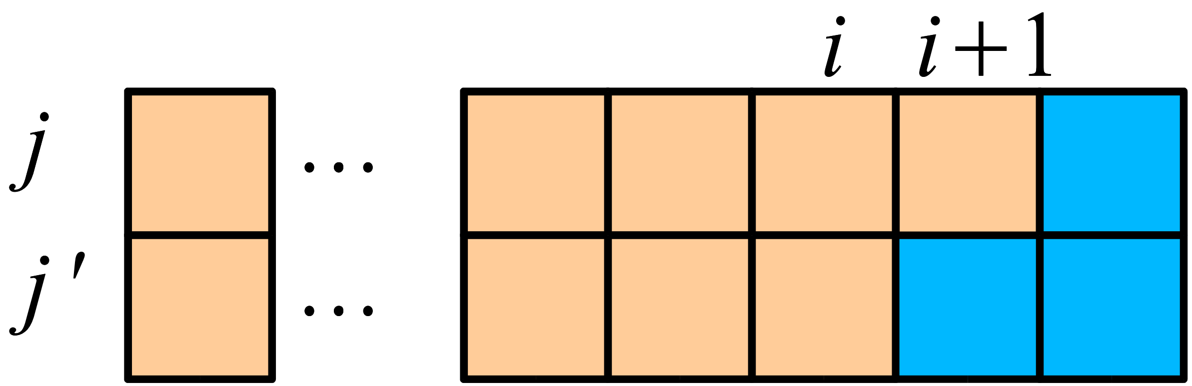
Action

If $B=1$

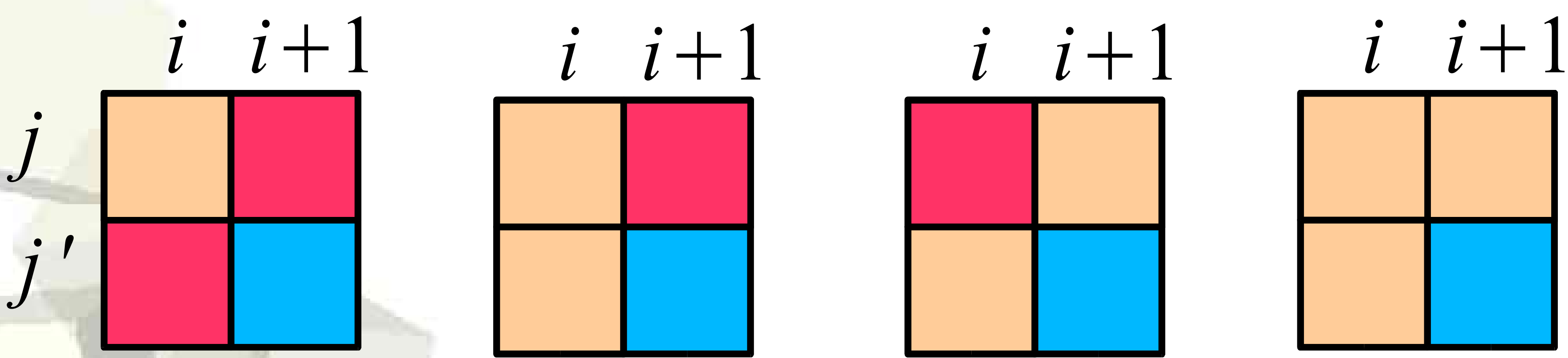
$$Y_{t+1}(j) \leftarrow Y_t(j) - 1$$

Case IV

Case IV: active items j, j' at positions $i, i+1$



4 subcases when $j' \leqslant j$



Case IV continued

See what happens in each subcase

Action

If $B=0$ or $j' \leqslant j$ do nothing

else $Y_{t+1}(j) \leftarrow Y_t(j) - 1$

$Y_{t+1}(j') \leftarrow Y_t(j') + 1$

This update function is time dependent

The update depends on the state of the bounding chain

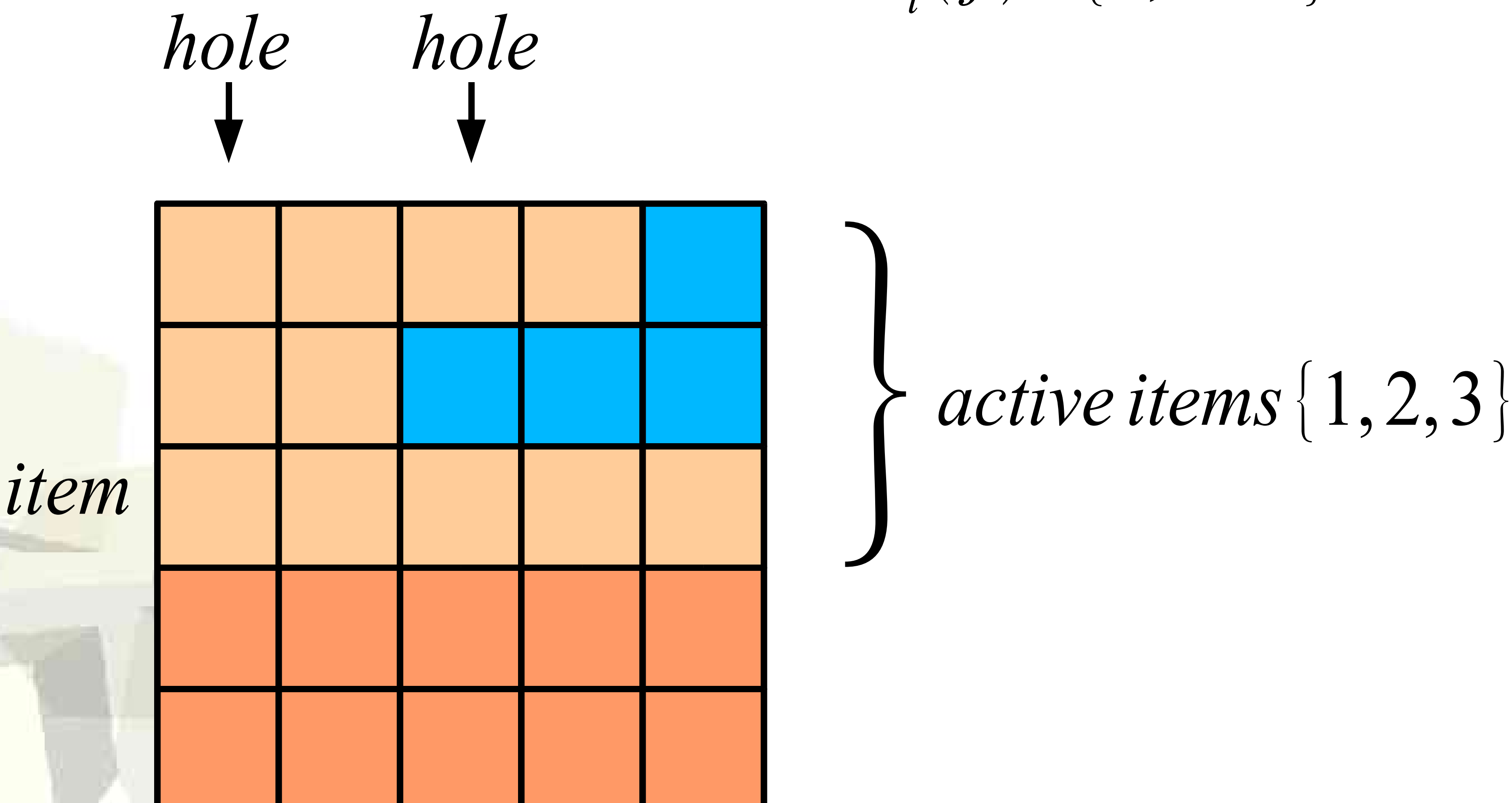
The bounding chain state depends on

$$U_t, U_{t-1}, U_{t-2}, \dots$$

Analysis

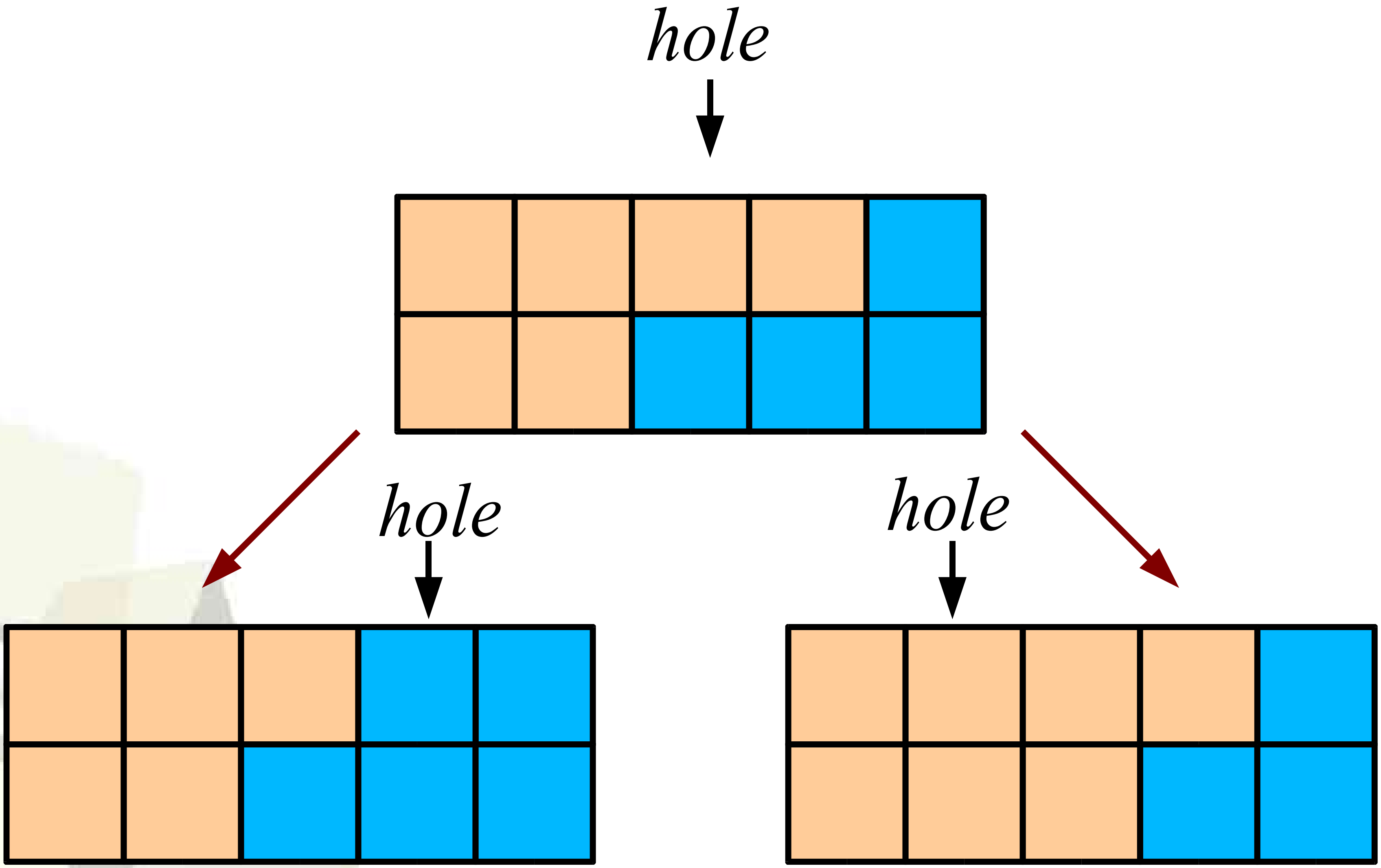
Key fact: The number of active states can only stay the same or increase

Call position i a hole if $Y_t(j) \neq \{1,\ldots,i\}$ for all j



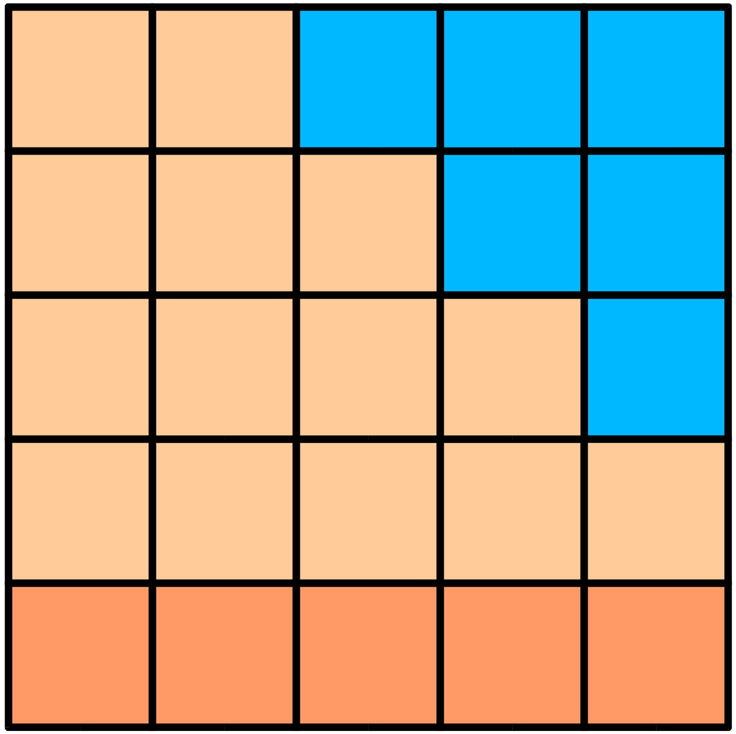
Analysis

Roughly speaking: clear positions take a simple random walk on $\{1,\ldots,n\}$



Analysis

When a hole reaches n it creates a new active item



Time for algorithm is time for all holes to reach state n

Analysis continued

A hole needs n^2 moves in expectation to reach n

A hole is moved one in every $n/2$ steps

There are at most n holes at the start

Expected Running time

$$O\left(n^3 \ln(n)\right)$$

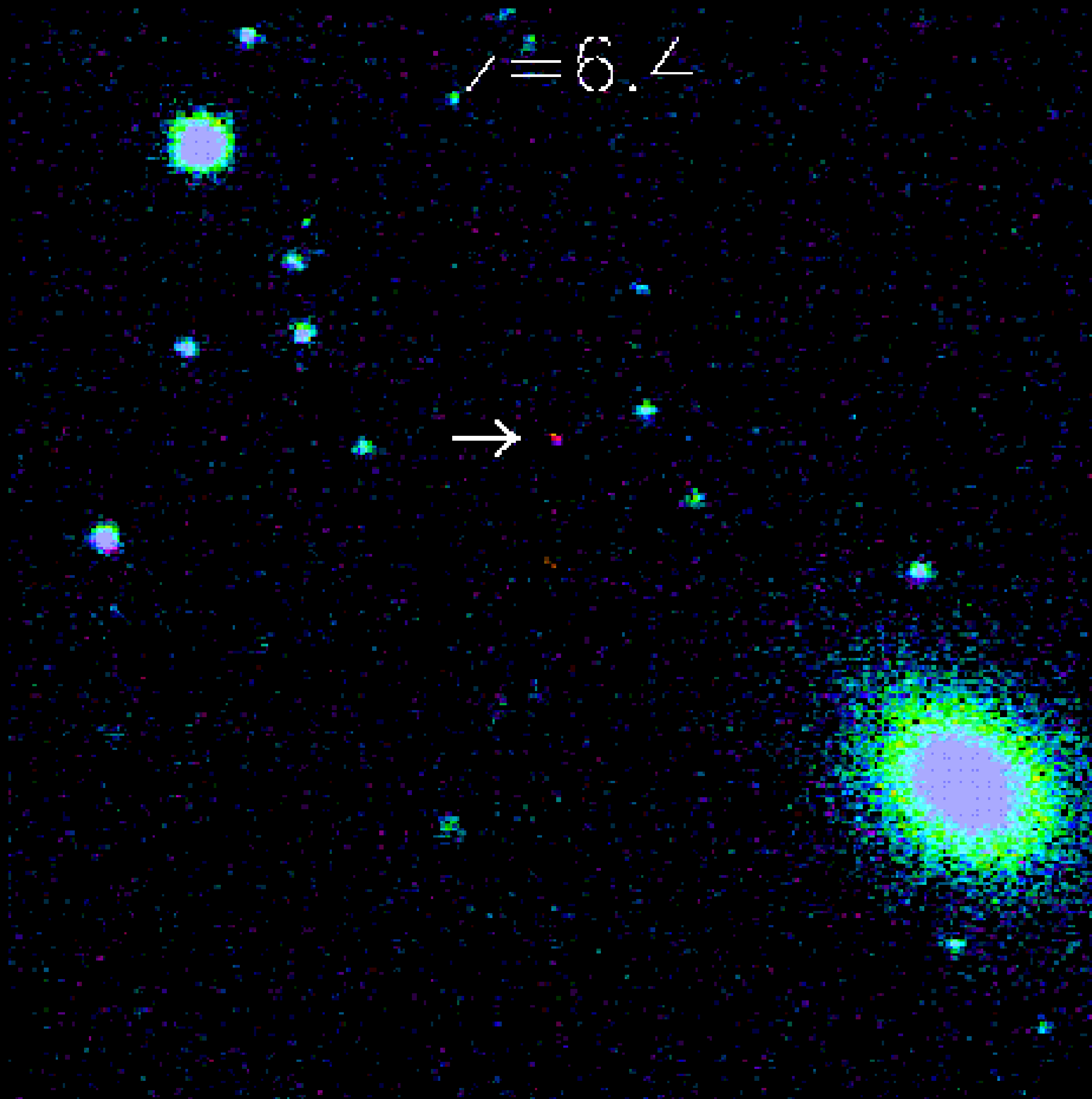
A potential proof

Create a potential function

$$\phi(Y_t) = \sum_{i \text{ is a hole}} \left[n^2 - i^2\right]$$

Can show

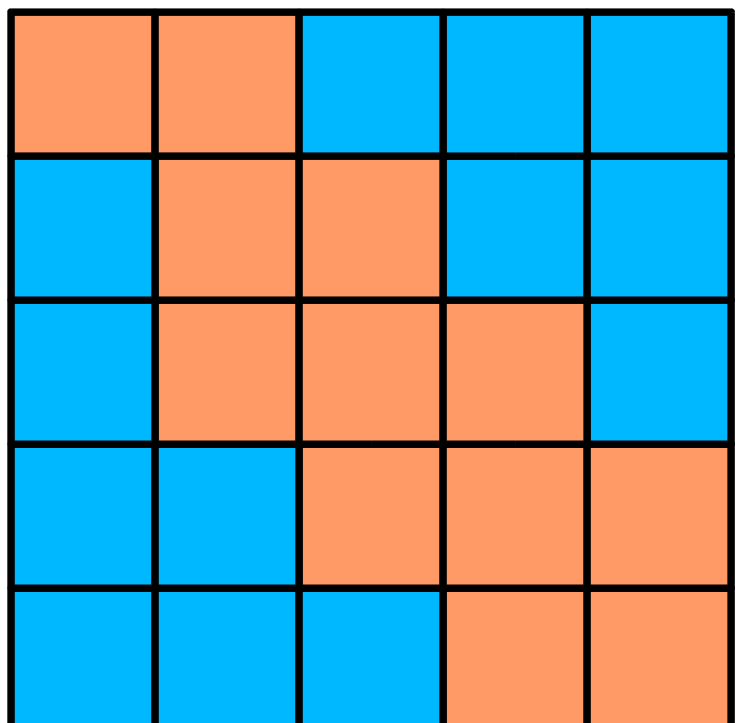
$$E\left[\phi(Y_{t+1}) \middle| Y_t\right] \le \phi(Y_t)\left[1 - \frac{2}{n^3}\right]$$



CREDIT: Sloan Digital Sky Survey at Apache Point Observatory

A related problem

[Efron, Petrosian 1999] Quasar luminosity data is doubly truncated, making testing correlation difficult. They suggest nonparametric test: to find *p*-value, need samples from interval permutations



Analysis difficult

Unfortunately, holes cannot move freely in this example, and so currently no a priori estimates on the running time of this algorithm are known

On the other hand: who cares? Run the algorithm, if it's fast use it.

What are update functions?

Update functions are examples of couplings

A coupling runs two processes simultaneously

- Marginally, each process looks like the original chain
- Their moves can be dependent

A coupled process $\{A_t, B_t\}$ is faithful if

$$A_t = B_t \text{ implies } A_{t+1} = B_{t+1} \text{ for all } t$$

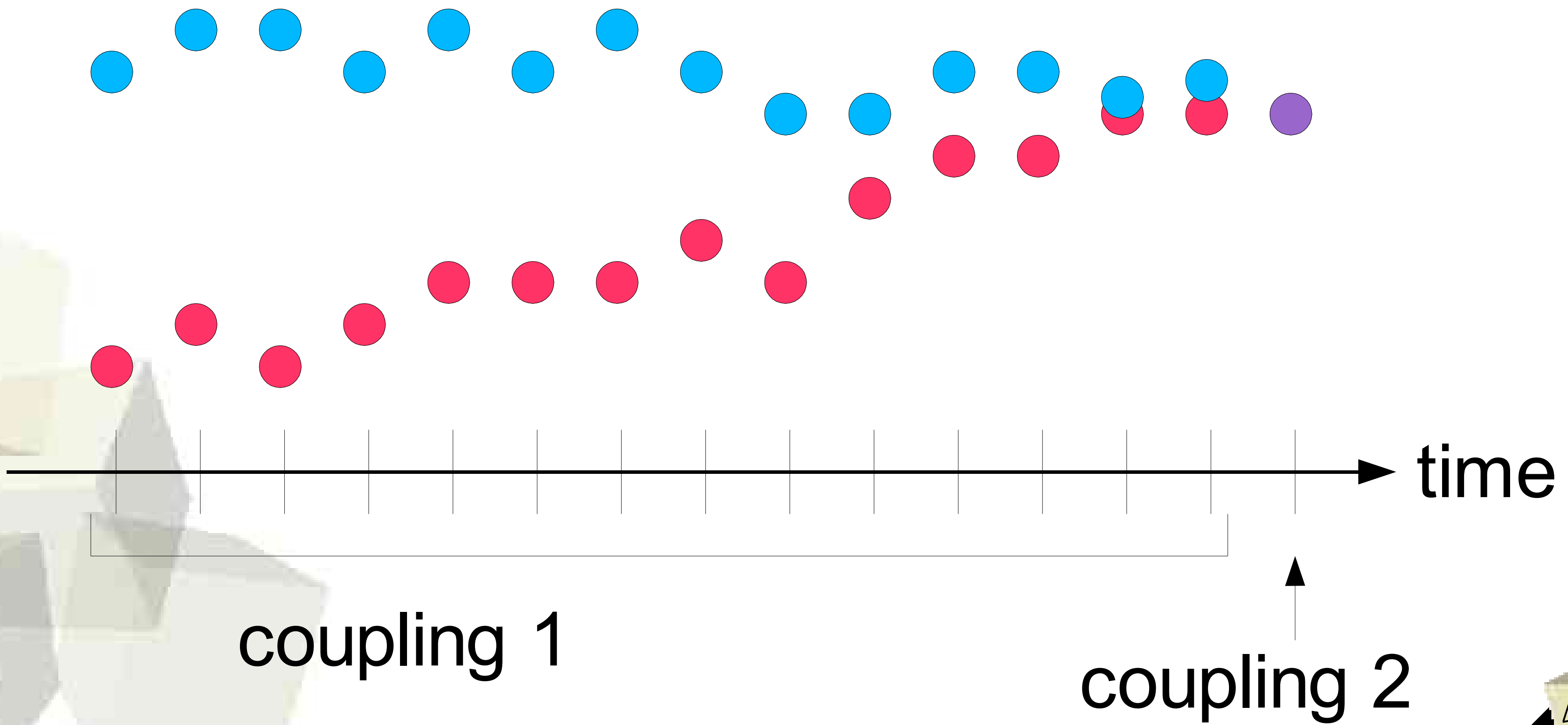
The coupling lemma says that the time until coupling is a bound on the mixing time of the Markov chain

Continuous problems

Time dependent couplings have already been used for continuous problems

Used for proving mixing times

Not for perfect sampling



Continuous Coupling

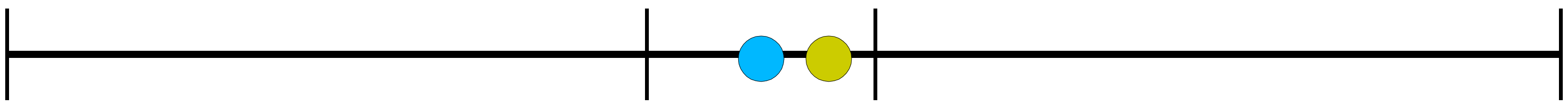
Type 1 coupling brings the states "close"
Type 2 coupling finishes the job

[Wilson 1999] Multishift coupling for type 2 works for unimodal distribution on moves

With time dependent CFTP, need much weaker type 2 coupling

Example

Continuous random walk on $[0, n]$



$$X_{t+1} \sim \text{Unif}\left[\max\{X_t - 1, 0\}, \min\{X_t + 1, n\}\right]$$

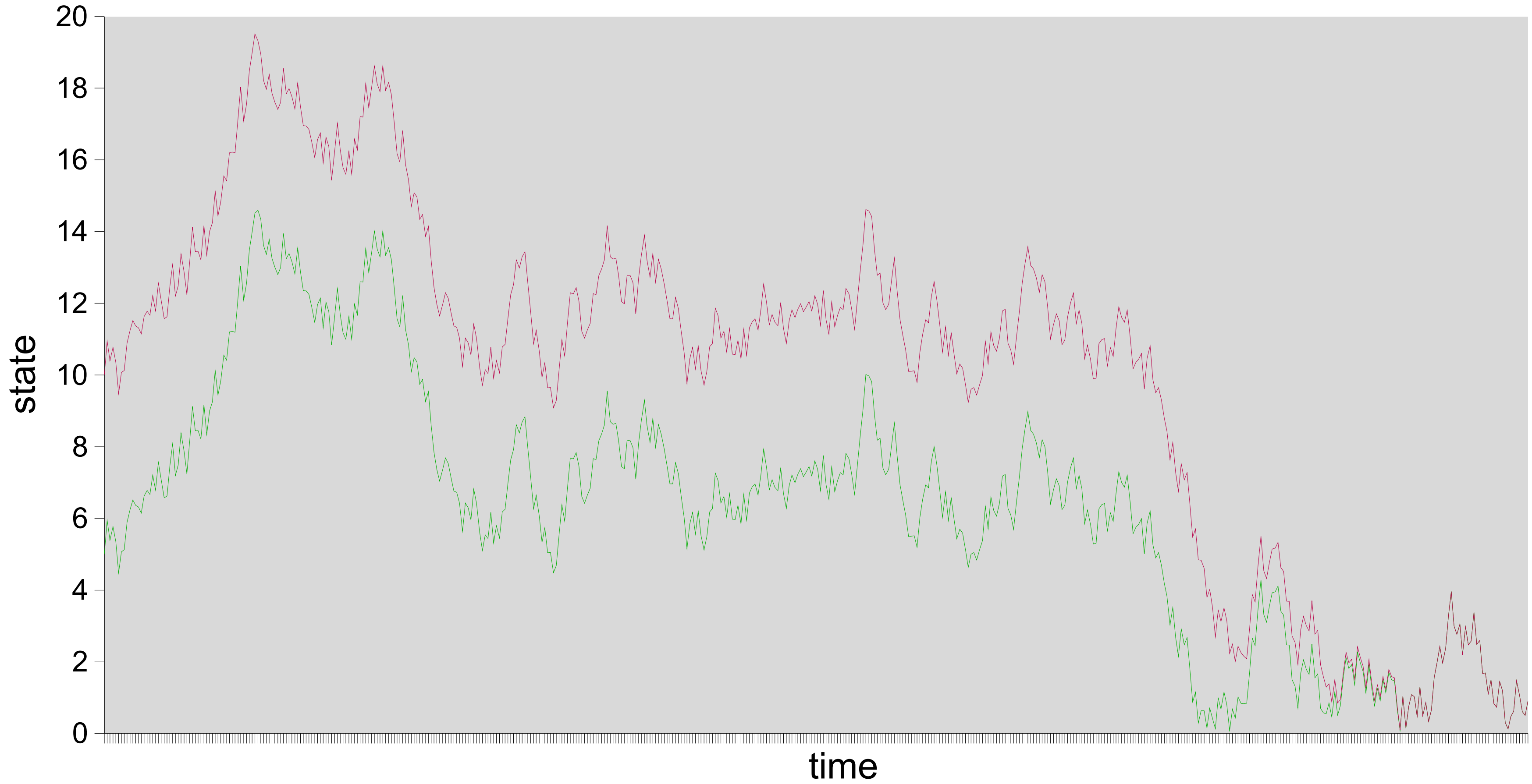
Type 1 update function

$$b = \min\{X_t + 1, n\}$$

$$a = \max\{X_t - 1, 0\}$$

$$f(x, u) = u(b - a) + a$$

Type 1 in action



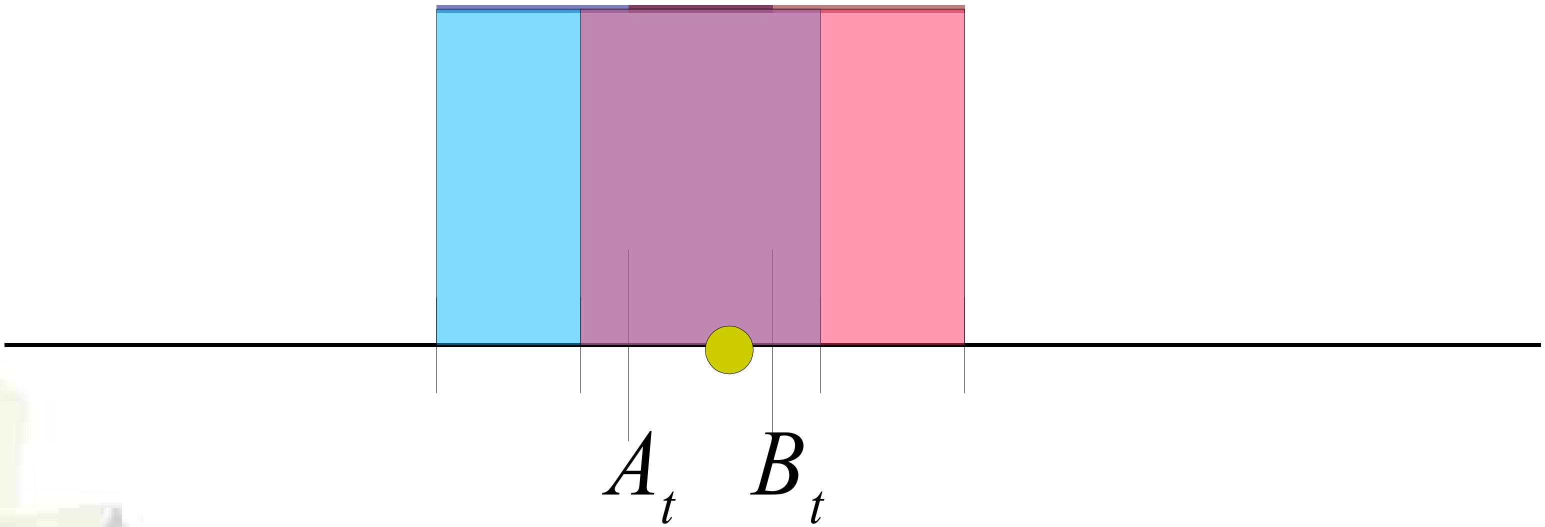
Starting difference: 5 on [0,20]

Number of steps: 500

Final difference: 0.000000078

Type 2 coupler

Idea: Distribution of A_{t+1}, B_{t+1} overlaps once A_t, B_t close



A draw that lands in shared density region works for both processes

Multishift and time dependent

To perfectly sample

- Use type 1 update function in time [-T,1]
- Use type 2 update function at time 1 to finish

For most problems, this idea makes continuous no more difficult than discrete

Highlights

Adding time dependent update functions increases the power of CFTP for perfect sampling

For linear extensions, provides fastest known method of generating samples

$$O\left(n^3 \ln\left(n\right)\right)$$

Also works for interval permuations

Considerably simplifies algorithms for continuous problems

References

G. Brightwell and P. Winkler, Counting linear extensions. Order 8(3): 1—17, 1991

R. Bubley and M. Dyer, Faster random generation of linear extensions, Disc. Math. 201 (1999) 81—88

B. Efron and V. Petrosian. Nonparametric methods for doubly truncated data. J. of Am. Stat. Assoc. Sep (1999) 824—834

S. Felsner and L. Wernisch. Markov chains for linear extension, the two-dimensional case. In Proc. of 8th Annual ACM-SIAM Symposium on Discrete Algorithms, (1997) 239—247

M. Huber. Perfect sampling using bounding chains. *Annals of Applied Probability*, 2004, to appear

M. Huber. Fast perfect sampling from linear extensions (submitted)

References

M. Jerrum, L. Valiant, and V. Vazirani, Random generation of combinatorial structures from a uniform distribution. Theoret. Comput. Sci. 43 (1986) 169—188

D. Wilson, Mixing times of lozenge tilings and card-shuffling Markov chains, Annals of Applied Probability, 2004

My website:
http://www.math.duke.edu/~mhuber

David Wilson's perfect sampling page
http://dbwilson.com/exact/