Spatial Birth-Death-Swap Chains

Why swapping at birth is a good thing

Mark Huber

Department of Mathematics
Claremont-McKenna College

14 Jan, 2010

Competition is everywhere

Towns compete for space

Trees compete for sunlight

Effects of competition

For spatial data...

- points look like they are repelling one another
- more regularly spaced than if locations independent

Models

- Begin with standard Poisson point process
- Penalize configurations where points are close together

Today

- Need to be able to sample from models to analyze behavior
- A new Markov chain method for dealing with these models

Effects of competition

Markov chains

- make small random changes to configuration
- birth=add a point
- death=remove a point

Today

- swap=one point added + one point removed
- First used for discrete time and space (Luby, Vigoda 1997)
- Here extended to continuous time and space
- Result: faster chain (theoretically and experimentally)

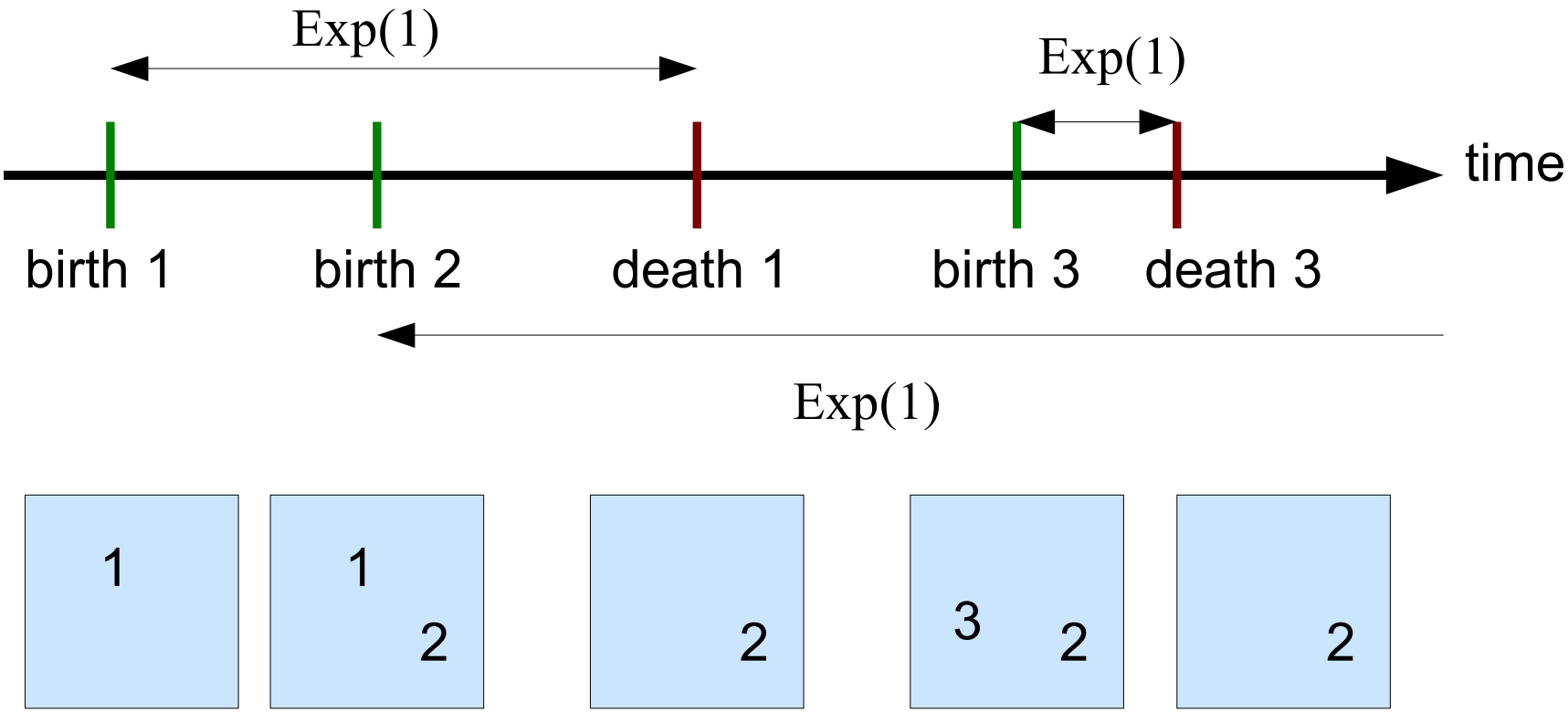
Continuous time Birth Death chains

Jump processes:

- Points born at times separated by time Exp($\lambda \cdot \mathrm{area}(S)$)
- When point born, decide “lifetime" that is Exp(1)
- After lifetime, the point dies and is removed from process

Stationary distribution Poisson point process

Illustration of birth death chain



Purpose of birth death chains

Metropolis-Hastings

- Birth Death process plays role of proposal chain
- Preston's [3] approach: always accept deaths
- Only sometimes accept briths

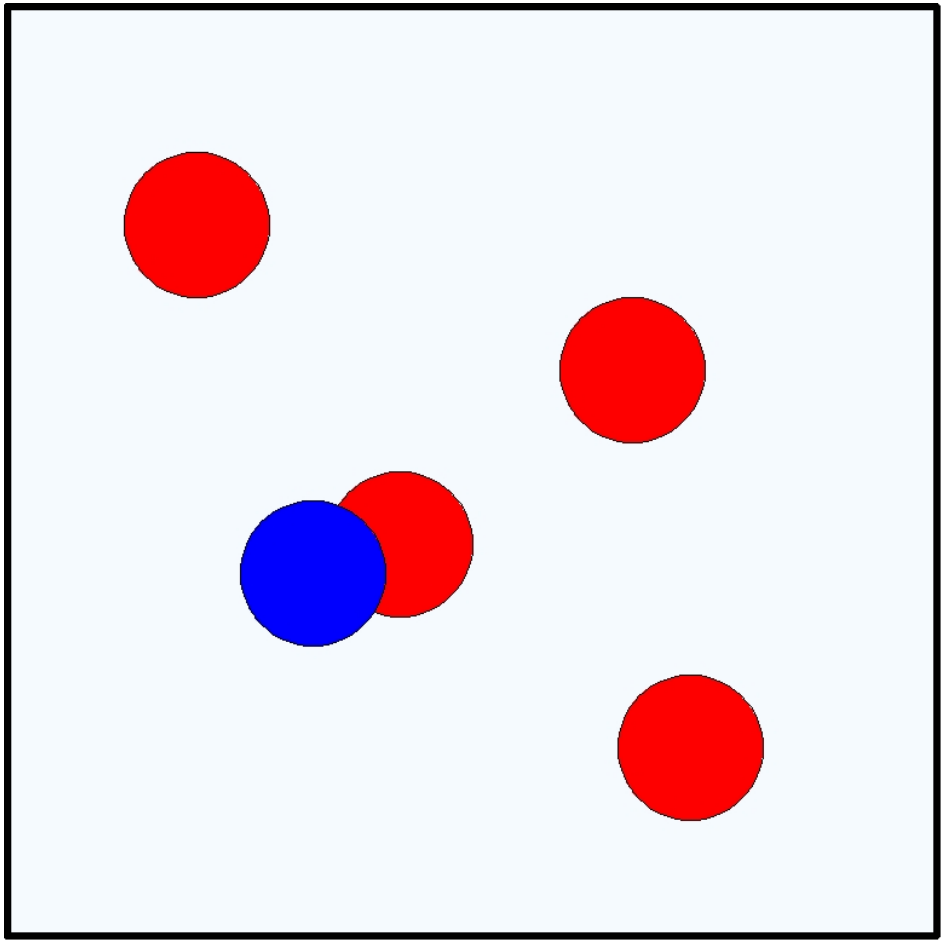
By only accepting some births...

- Ensures jump process equivalent of reversibility
- Works for locally stable densities

Hard core model: example of rejected birth

Hard core means discs are not allowed to overlap

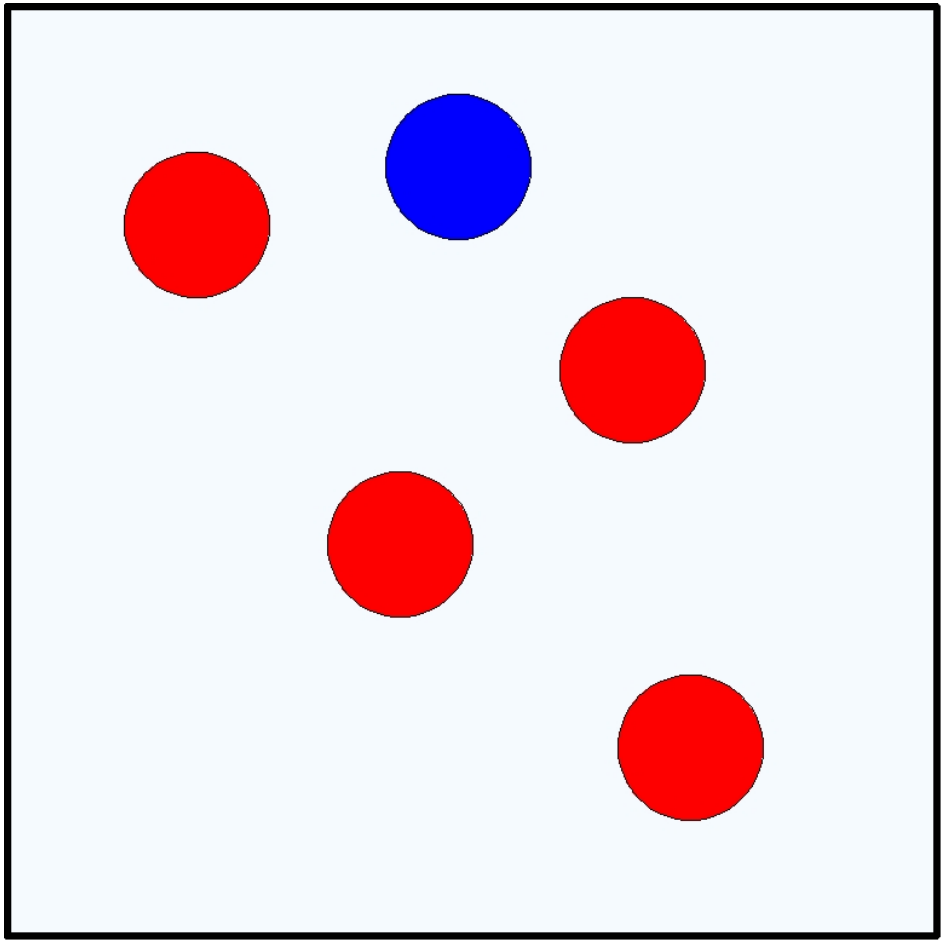
New point (in blue) is rejected as too close to existing points



Example of accepted birth

New point (in blue) is accepted an added to configuration

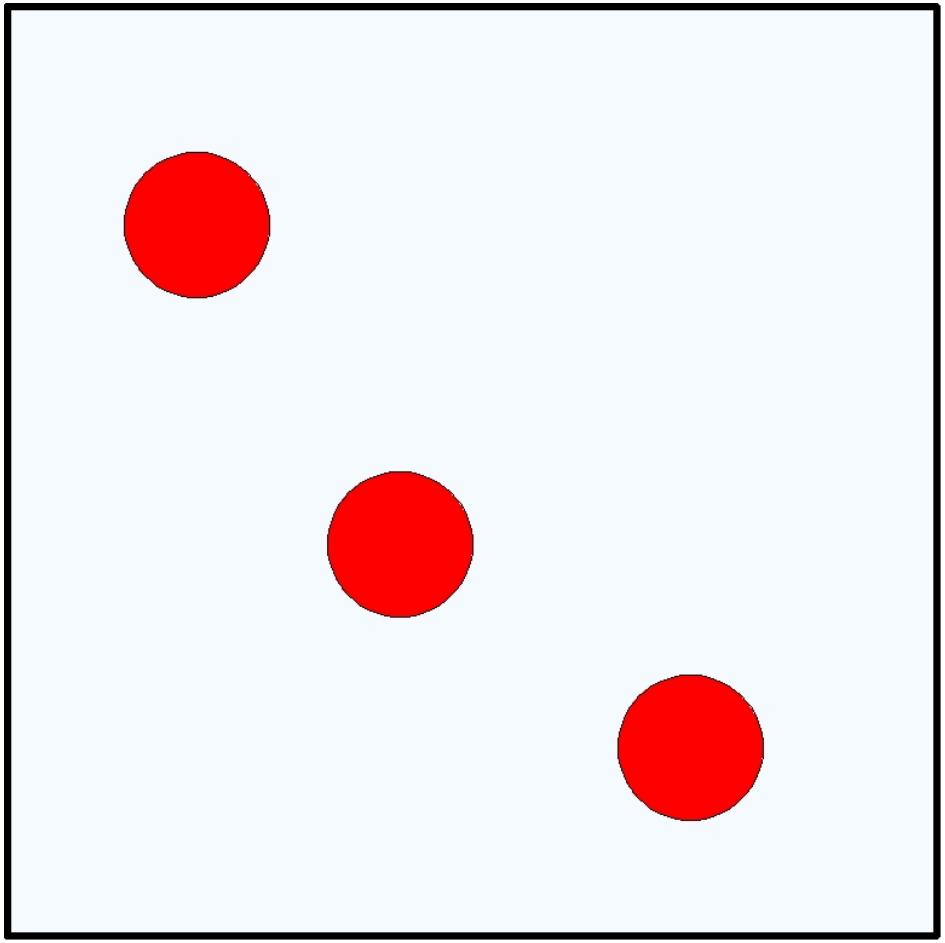
Too close to existing points



Example of death

Deaths are always accepted

(Removing point never violates hard core constraint)



To speed up chain, add a move

Old moves

- Birth: addition of point
- Death: removal of point

New move

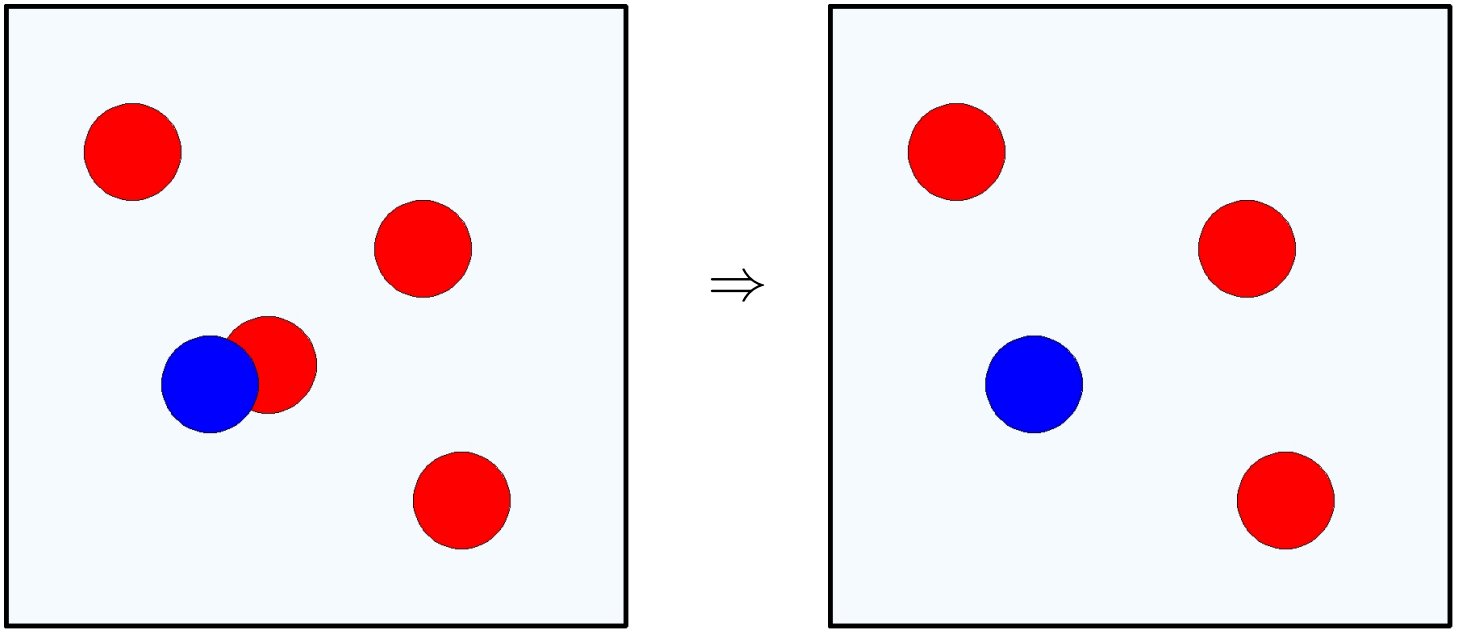
- Swap: addition and removal happen simulataneously

History

- Used in discrete context by Broder (1986) for perfect matchings
- Used for discrete hard core processes by Luby & Vigoda (1997)

Example of swap for hard core gas model

When blocked by exactly one point, “swap” with blocking point:



Some details

Things to consider:

- Does swapping give correct distribution?
- Does it improve performance in a theoretical way?
- Does the swap move generalize?

Preprint: Huber [2]

- Can set up probability of swapping to give correct distribution
- Rates for swap either direction must be equal
- Does work faster than original chain
- Speeds up perfect simulation algorithm

Some details

Things to consider:

- Does swapping give correct distribution?
- Does it improve performance in a theoretical way?
- Does the swap move generalize?

Preprint: Huber [2]

- Can set up probability of swapping to give correct distribution
- Rates for swap either direction must be equal
- Does work faster than original chain
- Speeds up perfect simulation algorithm

Perfect Sampling

"Practice makes perfect, but nobody's perfect, so why practice?"

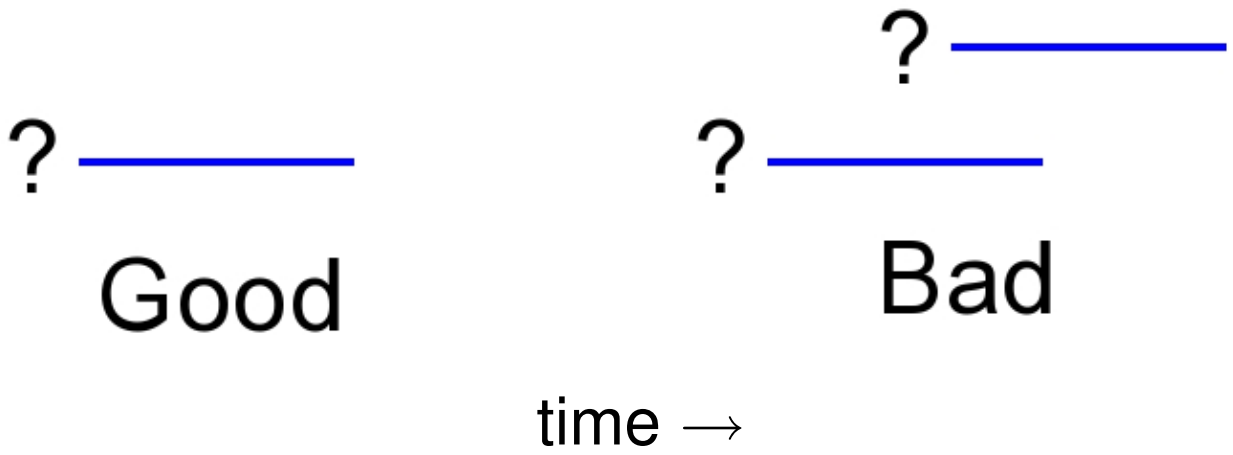
Problem with Markov chains

- How long should they be run?
- Perfect sampling algorithms share good properties of Markov chains...
- ...but terminate in finite time (with probability 1)

Dominated Coupling From the Past

Kendall and Møller [1]: Keep track of unknown point over lifetime:

- Say we don't know if a point should be in the set or not
- If it dies, great!
- If point born within range before it dies, bad



When is procedure fast?

The unknown points (?) are like an infection

- Die out when average # of children < 1
- Let a be area of ball of radius R
- Average # children before death is λa

When are you guaranteed good performance?

Theorem (Huber [2])

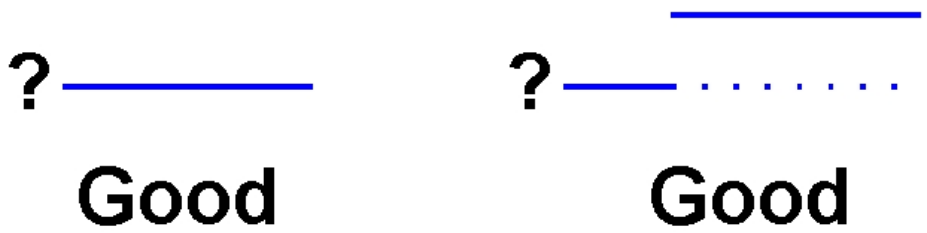
Running time of dCFTP (no swap) is $\Theta(\mu(S)\ln\mu(S))$ *for* $\lambda \leq f(R)$

Theorem (Huber [2])

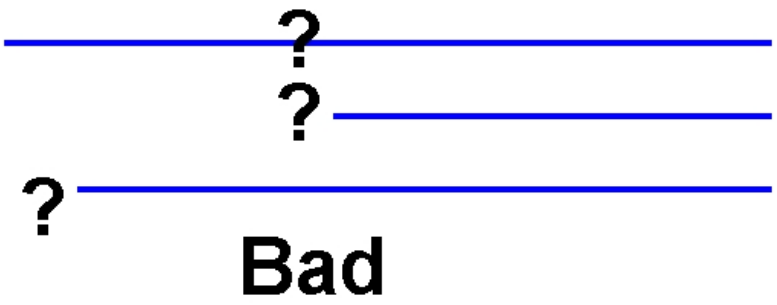
Running time of dCFTP (sometimes swap) is $\Theta(\mu(S)\ln\mu(S))$ *for* $\lambda \leq 2f(R)$

Swap moves can help or hurt...

Situation 1: When only a single ? in range, swapping helps:



Situation 2: When more than one neighbor in range, swapping hurts:



Sometimes swapping

When given opportunity to swap:

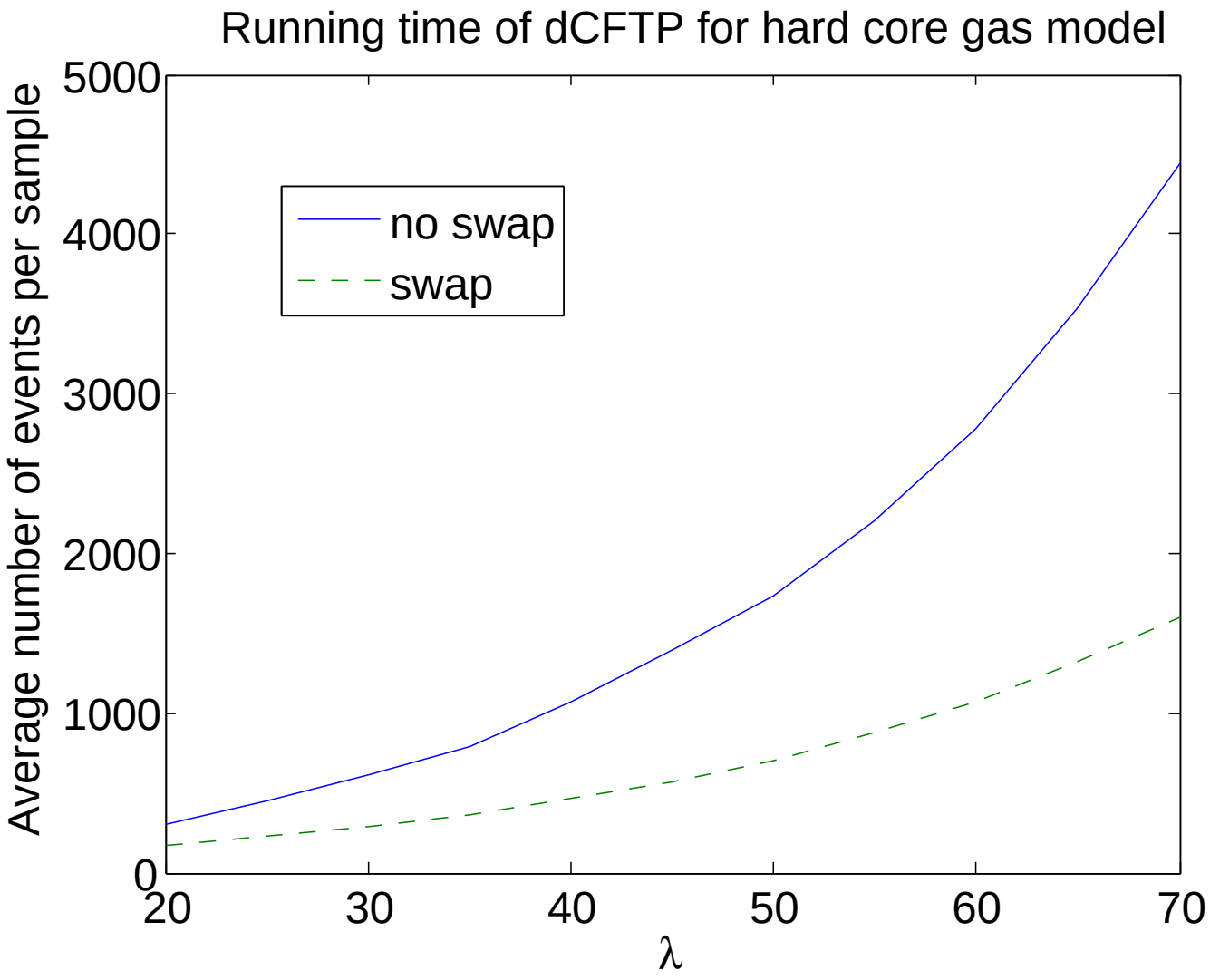
- Execute swap with probability p_{swap}
- Otherwise no swap

Set $p_{\text{swap}} = 1/4$:

- Situation 1: +1 ?'s with prob 3/4, -1 ?'s with proba $1/4$
- Situation 2: 0 ?'s with prob 3/4, +2 ?'s with prob $1/4$
- Either way: rate of ?'s is
  $(+1)(3/4) + (-1)(1/4) = 2(1/4) = 1/2$

Effectively, ?'s born at half the rate they were with no swap move

Running time results



Conclusions about swap move

What is known:

- Swap move easy to add to point processes
- Also can be used in dCFTP to get perfect sampling algorithm
- Results in about a 4-fold speedup for hard-core gas model

Future work:

- Running time comparison for Strauss process
- Improvement near phase transition
- Experiment better than theory–can theory be improved?

References

W.S. Kendall and J. Møller.
Perfect simulation using dominating processes on ordered spaces, with application to locally stable point processes.
Adv. Appl. Prob., 32:844–865, 2000.

M. L. Huber.
Spatial Birth-Death-Swap Chains
preprint, 2007

C.J. Preston.
Spatial birth-and-death processes.
Bull. Inst. Int. Stat., 46(2):371–391, 1977.