PERM and all that
a comparison of growth algorithms

Thomas Prellberg
School of Mathematical Sciences
Queen Mary, University of London

Monte Carlo Algorithms in Statistical Physics
Melbourne, July 26-28
Introduction

A Zoology of Growth Algorithms
- Which Algorithm is Best?
- ISAW - the canonical lattice model

2 The ‘Old’ Algorithms
- Rosenbluth²
- PERM
- Multicanonical PERM
- FlatPERM

3 The ‘New’ Algorithms
- New Ideas
- GARM
- GAS

4 Conclusion
- Outlook
- Thanks

Thomas Prellberg

PERM and all that
Outline

1 Introduction
   - A Zoology of Growth Algorithms
   - Which Algorithm is Best?
   - ISAW - the canonical lattice model

2 The ‘Old’ Algorithms
   - Rosenbluth
   - PERM
   - Multicanonical PERM
   - FlatPERM

3 The ‘New’ Algorithms
   - New Ideas
   - GARM
   - GAS

4 Conclusion
   - Outlook
   - Thanks
These days there exists a zoo of growth algorithms

- 1997: PERM
- 2003: nPERMss/nPERMIs
- 2003: Multicanonical PERM
- 2004: flatPERM
- 2008: GARM/flatGARM
- 2009: GAS
- 201?: flatGAS
These days there exists a zoo of growth algorithms

- 1997: PERM
- 2003: nPERMss/nPERMIs
- 2003: Multicanonical PERM
- 2004: flatPERM
- 2008: GARM/flatGARM
- 2009: GAS
- 201?: flatGAS

All of this is based on

- 1955: Rosenbluth & Rosenbluth
Which Algorithm is Best?

I don't really know.
or, perhaps slightly better,
It depends...
Which Algorithm is Best?

I don’t really know.
Which Algorithm is Best?

I don’t really know.

or, perhaps slightly better,

It depends . . .
Why? It’s just easiest to use your own algorithm
Why? It’s just easiest to use your own algorithm

- The flatPERM algorithm (and some pedagogical applications):
Why? It’s just easiest to use your own algorithm

- The flatPERM algorithm (and some pedagogical applications):

- Bulk vs surface:
Why? It’s just easiest to use your own algorithm

- **The flatPERM algorithm (and some pedagogical applications):**

- **Bulk vs surface:**

- **Hydrogen-bond type interactions:**
Why? It’s just easiest to use your own algorithm

The flatPERM algorithm (and some pedagogical applications):

Bulk vs surface:

Hydrogen-bond type interactions:

Alternative lattice models:
As of July 25th,

- PERM (1997): 245 citations
- nPERM (2003): 65 citations
- Multicanonical PERM (2003): 45 citations
- flatPERM (2004): 34 citations
- GARM/flatGARM (2008): 3 citations
- GAS/flatGAS (2009): 1 citation
As of July 25th,
- PERM (1997): 245 citations
- nPERM (2003): 65 citations
- Multicanonical PERM (2003): 45 citations
- flatPERM (2004): 34 citations
- GARM/flatGARM (2008): 3 citations
- GAS/flatGAS (2009): 1 citation

This should be compared with e.g.
- Umbrella Sampling (1977): 994 citations
ISAW - the canonical lattice model

Interacting Self-Avoiding Walk (ISAW)

- Physical space $\rightarrow$ simple cubic lattice $\mathbb{Z}^3$
- Polymer $\rightarrow$ self-avoiding $N$-step random walk (SAW) $\varphi$
- Quality of solvent $\rightarrow$ short-range interaction $\epsilon$, Energy $E_N(\varphi) = m(\varphi)\epsilon$

Quality of solvent $\rightarrow$ short-range interaction $\epsilon$, Energy $E_N(\varphi) = m(\varphi)\epsilon$
ISAW - the canonical lattice model

Interacting Self-Avoiding Walk (ISAW)

- Physical space → simple cubic lattice $\mathbb{Z}^3$
- Polymer → self-avoiding $N$-step random walk (SAW) $\varphi$
- Quality of solvent → short-range interaction $\epsilon$, Energy $E_N(\varphi) = m(\varphi)\epsilon$

Partition function:

$$Z_N(\beta) = \sum_m C_{N,m} e^{-\beta m\epsilon}$$

$C_{N,m}$ is the number of SAWs with $N$ steps and $m$ interactions
ISAW - the canonical lattice model

Interacting Self-Avoiding Walk (ISAW)

- Physical space $\rightarrow$ simple cubic lattice $\mathbb{Z}^3$
- Polymer $\rightarrow$ self-avoiding $N$-step random walk (SAW) $\varphi$
- Quality of solvent $\rightarrow$ short-range interaction $\epsilon$, Energy $E_N(\varphi) = m(\varphi)\epsilon$

Partition function:

$$Z_N(\beta) = \sum_m C_{N,m} e^{-\beta m\epsilon}$$

$C_{N,m}$ is the number of SAWs with $N$ steps and $m$ interactions

Thermodynamic Limit for a dilute solution:

$$V = \infty \quad \text{and} \quad N \rightarrow \infty$$
Extensions of the Model

- In addition to
  - polymer and solvent modelling (bulk interaction)
- add
  - protein-like structure (HP interactions)
  - adsorption (surface interaction)
  - micromechanical deformations e.g. force on chain end (optical tweezers)
- Complete description through high-dimensional density of states: (a) bulk and (b) surface interactions, (c) positions of chain end
Outline

1. Introduction
   - A Zoology of Growth Algorithms
   - Which Algorithm is Best?
   - ISAW - the canonical lattice model

2. The ‘Old’ Algorithms
   - Rosenbluth$^2$
   - PERM
   - Multicanonical PERM
   - FlatPERM

3. The ‘New’ Algorithms
   - New Ideas
   - GARM
   - GAS

4. Conclusion
   - Outlook
   - Thanks
Rosenbluth versus Simple Sampling

Simple Sampling (for SAW)
- Choose starting vertex at the origin
- Draw one of the neighbouring sites uniformly at random
- If occupied, reject entire walk and start again
- If unoccupied, accept and repeat (up to some maximal walk length)
Rosenbluth versus Simple Sampling

Simple Sampling (for SAW)
- Choose starting vertex at the origin
- Draw one of the neighbouring sites uniformly at random
- If occupied, reject entire walk and start again
- If unoccupied, accept and repeat (up to some maximal walk length)

Rosenbluth Sampling (for SAW)
- Choose starting vertex at the origin
- Draw one of the unoccupied neighbouring sites uniformly at random
- If there is none, reject entire walk and start again
- If unoccupied, accept and repeat (up to some maximal walk length)

Rosenbluth versus Simple Sampling

Simple Sampling (for SAW)
- Choose starting vertex at the origin
- Draw one of the neighbouring sites uniformly at random
- If occupied, reject entire walk and start again
- If unoccupied, accept and repeat (up to some maximal walk length)

Rosenbluth Sampling (for SAW)
- Choose starting vertex at the origin
- Draw one of the unoccupied neighbouring sites uniformly at random
- If there is none, reject entire walk and start again
- If unoccupied, accept and repeat (up to some maximal walk length)


(Augment with Importance Sampling for ISAW)
Rosenbluth versus Simple Sampling

Simple Sampling

- Large attrition, so very inefficient
- Uniform, independent samples
Rosenbluth versus Simple Sampling

Simple Sampling
- Large attrition, so very inefficient
- Uniform, independent samples

Rosenbluth Sampling
- Less attrition (but still exponential)
Rosenbluth versus Simple Sampling

Simple Sampling
- Large attrition, so very inefficient
- Uniform, independent samples

Rosenbluth Sampling
- Less attrition (but still exponential)
- Non-uniform sampling
Rosenbluth versus Simple Sampling

Simple Sampling
- Large attrition, so very inefficient
- Uniform, independent samples

Rosenbluth Sampling
- Less attrition (but still exponential)
- Non-uniform sampling

Walks with large weights dominate ensemble
Rosenbluth versus Simple Sampling

Simple Sampling
- Large attrition, so very inefficient
- Uniform, independent samples

Rosenbluth Sampling
- Less attrition (but still exponential)
- Non-uniform sampling

At step $k$, $a_k$ possibilities with probability $p_k = 1/a_k$

An $N$-step walk $\phi$ has weight

$$W(\phi) \propto \prod_{k<N} a_k(\phi)$$

Walks with large weights dominate ensemble
PERM: “Go with the Winners”

PERM = Pruned and Enriched Rosenbluth Method

Modify Rosenbluth Sampling by controlling the weights

\[ W_\beta(\varphi) = W(\varphi) e^{-\beta E(\varphi)} \]

PERM: “Go with the Winners”

PERM = Pruned and Enriched Rosenbluth Method

- Modify Rosenbluth Sampling by controlling the weights
  \[ W_\beta(\varphi) = W(\varphi) e^{-\beta E(\varphi)} \]
- Combat large weights by **Enrichment**:
  Weight \( W_\beta(\varphi) \) too large \( \Rightarrow \) make copies of the walk

P Grassberger, Phys Rev E 56 (1997) 3682

**PERM and all that**
PERM: “Go with the Winners”

PERM = Pruned and Enriched Rosenbluth Method

P Grassberger, Phys Rev E 56 (1997) 3682

- Modify Rosenbluth Sampling by controlling the weights
  \[ W_\beta(\varphi) = W(\varphi) e^{-\beta E(\varphi)} \]
  
  1. Combat large weights by **Enrichment**: 
     Weight \( W_\beta(\varphi) \) too large ⇒ make copies of the walk
  2. Combat small weights by **Pruning**: 
     Weight \( W_\beta(\varphi) \) too small ⇒ remove walks occasionally
PERM: "Go with the Winners"

PERM = Pruned and Enriched Rosenbluth Method

- Modify Rosenbluth Sampling by controlling the weights
  \[ W_\beta(\varphi) = W(\varphi)e^{-\beta E(\varphi)} \]

  1. Combat large weights by **Enrichment**:
     Weight \( W_\beta(\varphi) \) too large \( \Rightarrow \) make copies of the walk

  2. Combat small weights by **Pruning**:
     Weight \( W_\beta(\varphi) \) too small \( \Rightarrow \) remove walks occasionally

- Parameters: upper thresholds \( T_N \) and \( t_N \), pruning probability \( q \)

P Grassberger, Phys Rev E 56 (1997) 3682

Thomas Prellberg

PERM and all that
PERM: “Go with the Winners”

PERM = Pruned and Enriched Rosenbluth Method

- Modify Rosenbluth Sampling by controlling the weights
  \[ W_\beta(\varphi) = W(\varphi)e^{-\beta E(\varphi)} \]

  1. Combat large weights by Enrichment:
     Weight \( W_\beta(\varphi) \) too large \( \Rightarrow \) make copies of the walk
  2. Combat small weights by Pruning:
     Weight \( W_\beta(\varphi) \) too small \( \Rightarrow \) remove walks occasionally

- Parameters: upper thresholds \( T_N \) and \( t_N \), pruning probability \( q \)
- Adapt \( T_N \) and \( t_N \) during simulation, keep \( T_N/t_N \) roughly constant

P Grassberger, Phys Rev E 56 (1997) 3682

Thomas Prellberg

PERM and all that
PERM: “Go with the Winners”

PERM = Pruned and Enriched Rosenbluth Method

P Grassberger, Phys Rev E 56 (1997) 3682

Modify Rosenbluth Sampling by controlling the weights

\[ W_\beta(\varphi) = W(\varphi) e^{-\beta E(\varphi)} \]

1. Combat large weights by Enrichment:
   Weight \( W_\beta(\varphi) \) too large \( \Rightarrow \) make copies of the walk

2. Combat small weights by Pruning:
   Weight \( W_\beta(\varphi) \) too small \( \Rightarrow \) remove walks occasionally

Parameters: upper thresholds \( T_N \) and \( t_N \), pruning probability \( q \)

Adapt \( T_N \) and \( t_N \) during simulation, keep \( T_N / t_N \) roughly constant

nPERM = New PERM


Significant improvement: when enriching, force distinct copies
PERM: “Go with the Winners”

PERM = Pruned and Enriched Rosenbluth Method

 Modify Rosenbluth Sampling by controlling the weights

\[ W_\beta(\varphi) = W(\varphi) e^{-\beta E(\varphi)} \]

1. **Combat large weights by Enrichment:**
   Weight \( W_\beta(\varphi) \) too large ⇒ make copies of the walk

2. **Combat small weights by Pruning:**
   Weight \( W_\beta(\varphi) \) too small ⇒ remove walks occasionally

- Parameters: upper thresholds \( T_N \) and \( t_N \), pruning probability \( q \)
- Adapt \( T_N \) and \( t_N \) during simulation, keep \( T_N / t_N \) roughly constant

nPERM = New PERM


- Significant improvement: when enriching, force *distinct* copies
  (Augment with Importance Sampling: nPERMis)
Multicanonical PERM

- Sample the density of states with respect to an umbrella density

Sample the density of states with respect to an umbrella density


For uniform sampling of the density of states $C_{N,m}$, we need to use weights

$$W_{\text{flat}}(\varphi) = \frac{W(\varphi)}{C_{N,m}}$$
Multicanonical PERM

- Sample the density of states with respect to an umbrella density
  
  \[ W_{\text{flat}}(\varphi) = \frac{W(\varphi)}{C_{N,m}} \]


- For uniform sampling of the density of states $C_{N,m}$, we need to use weights

  \[ W_{\text{flat}}(\varphi) = \frac{W(\varphi)}{C_{N,m}} \]

- As $C_{N,m}$ is unknown, compute iteratively an approximation $C_{N,m}^{\text{approx}}$ and perform a final run with

  \[ W_{\text{flat}}^{\text{approx}}(\varphi) = \frac{W(\varphi)}{C_{N,m}^{\text{approx}}} \]

  (Multicanonical Method)

Multicanonical PERM

- Sample the density of states with respect to an umbrella density
- For uniform sampling of the density of states $C_{N,m}$, we need to use weights
  $$W_{\text{flat}}(\varphi) = W(\varphi)/C_{N,m}$$
- As $C_{N,m}$ is unknown, compute iteratively an approximation $C_{N,m}^{\text{approx}}$ and perform a final run with
  $$W_{\text{flat}}^{\text{approx}}(\varphi) = W(\varphi)/C_{N,m}^{\text{approx}}$$
  (Multicanonical Method)
- The resulting algorithm is called multicanonical PERM
  M Bachmann and W Janke, PRL 91 (2003) 208105
Revisit PERM

- Exact enumeration: choose all a continuations with weight 1
Revisit PERM

- Exact enumeration: choose all a continuations with weight 1
- Rosenbluth sampling: chose one continuation with weight a
Revisit PERM

- Exact enumeration: choose all \( a \) continuations with weight 1
- Rosenbluth sampling: chose one continuation with weight \( a \)

View Rosenbluth Sampling as \textit{approximate enumeration}
Revisit PERM

- Exact enumeration: choose all continuations with weight 1
- Rosenbluth sampling: chose one continuation with weight a

View Rosenbluth Sampling as *approximate enumeration*

- If an $N$ step walk $\varphi$ gets assigned a weight $W(\varphi) = \prod_{k<N} a_k(\varphi)$
  then $S$ walks with weights $W(\varphi_i)$ give an estimate

$$C_N^{est} = \langle W \rangle_N = \frac{1}{S} \sum_i W(\varphi_i)$$
Revisit PERM

- Exact enumeration: choose all a continuations with weight 1
- Rosenbluth sampling: chose one continuation with weight a

View Rosenbluth Sampling as approximate enumeration

- If an $N$ step walk $\varphi$ gets assigned a weight $W(\varphi) = \prod_{k<N} a_k(\varphi)$ then $S$ walks with weights $W(\varphi_i)$ give an estimate

$$C_N^{\text{est}} = \langle W \rangle_N = \frac{1}{S} \sum_i W(\varphi_i)$$

- Add pruning/enrichment with respect to the ratio

$$r = W(\varphi)/\langle W \rangle_N$$
Revisit PERM

- Exact enumeration: choose all \(a\) continuations with weight 1
- Rosenbluth sampling: chose one continuation with weight \(a\)

View Rosenbluth Sampling as \textit{approximate enumeration}

- If an \(N\) step walk \(\varphi\) gets assigned a weight \(W(\varphi) = \prod_{k<N} a_k(\varphi)\) then \(S\) walks with weights \(W(\varphi_i)\) give an estimate

\[
C_{N}^{est} = \langle W \rangle_N = \frac{1}{S} \sum_{i} W(\varphi_i)
\]

- Add pruning/enrichment with respect to the ratio

\[
r = \frac{W(\varphi)}{\langle W \rangle_N}
\]

1. If \(r > 1\), make \(c = \min([r], a_N)\) distinct copies and update

\[
W(\varphi) \leftarrow W(\varphi)/c
\]

2. If \(r < 1\), prune with probability \(1 - r\) and update

\[
W(\varphi) \leftarrow W(\varphi)/r
\]
From PERM to flatPERM

An important observation:
- Number of samples generated for each $N$ is roughly constant
- We have a flat histogram algorithm in system size
From PERM to flatPERM

An important observation:
- Number of samples generated for each $N$ is roughly constant
- We have a flat histogram algorithm in system size

flatPERM = flat histogram PERM

T Prellberg and J Krawczyk, PRL 92 (2004) 120602
From PERM to flatPERM

An important observation:

- Number of samples generated for each $N$ is roughly constant
- We have a flat histogram algorithm in system size

flatPERM = flat histogram PERM

PERM: estimate number of walks $C_N$

- $C_N^{\text{est}} = \langle W \rangle_N$
- $r = W(\varphi)/C_N^{\text{est}}$

T Prellberg and J Krawczyk, PRL 92 (2004) 120602
From PERM to flatPERM

An important observation:

- Number of samples generated for each $N$ is roughly constant
- We have a flat histogram algorithm in system size

flatPERM = flat histogram PERM

- PERM: estimate number of walks $C_N$
  - $C_N^{est} = \langle W \rangle_N$
  - $r = W(\varphi)/C_N^{est}$

- PERM at finite temperature: estimate partition function $Z_N(\beta)$
  - $Z_N^{est}(\beta) = \langle W \exp(-\beta E) \rangle_N$
  - $r = W(\varphi) \exp(-\beta E(\varphi))/Z_N^{est}(\beta)$

T Prellberg and J Krawczyk, PRL 92 (2004) 120602
From PERM to flatPERM

An important observation:
- Number of samples generated for each $N$ is roughly constant
- We have a flat histogram algorithm in system size

flatPERM = flat histogram PERM

PERM: estimate number of walks $C_N$
- $C_N^{est} = \langle W \rangle_N$
- $r = W(\varphi)/C_N^{est}$

PERM at finite temperature: estimate partition function $Z_N(\beta)$
- $Z_N^{est}(\beta) = \langle W \exp(-\beta E) \rangle_N$
- $r = W(\varphi) \exp(-\beta E(\varphi))/Z_N^{est}(\beta)$

flatPERM: estimate density of states $C_{N,\bar{m}}$
- $C_{N,\bar{m}}^{est} = \langle W \rangle_{N,\bar{m}}$
- $r = W(\varphi)/C_{N,\bar{m}}^{est}$

T Prellberg and J Krawczyk, PRL 92 (2004) 120602
From PERM to flatPERM

An important observation:
- Number of samples generated for each $N$ is roughly constant
- We have a flat histogram algorithm in system size

flatPERM = flat histogram PERM

PERM: estimate number of walks $C_N$
- $C_N^{\text{est}} = \langle W \rangle_N$
- $r = W(\varphi) / C_N^{\text{est}}$

PERM at finite temperature: estimate partition function $Z_N(\beta)$
- $Z_N^{\text{est}}(\beta) = \langle W \exp(-\beta E) \rangle_N$
- $r = W(\varphi) \exp(-\beta E(\varphi)) / Z_N^{\text{est}}(\beta)$

flatPERM: estimate density of states $C_{N,\vec{m}}$
- $C_{N,\vec{m}}^{\text{est}} = \langle W \rangle_{N,\vec{m}}$
- $r = W(\varphi) / C_{N,\vec{m}}^{\text{est}}$

Parameter-free implementation

T Prellberg and J Krawczyk, PRL 92 (2004) 120602
Example: 2dim ISAW simulation up to $N = 1024$

- flatPERM starts with poor estimates of the average weights $\langle W \rangle$
- To stabilise algorithm (avoid initial overflow/underflow):
  delay growth of large configurations by increasing lengths gradually
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 1,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 10,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 20,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 30,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 40,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 50,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 60,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 70,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 80,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 90,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 100,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 110,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 120,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 130,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 140,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 150,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 160,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 170,000,000
Example: 2dim ISAW simulation up to $N = 1024$
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 190,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 200,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 210,000,000
Example: 2dim ISAW simulation up to \( N = 1024 \)

Total sample size: 220,000,000
Example: 2dim ISAW simulation up to \( N = 1024 \)

Total sample size: 230,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 240,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 250,000,000

$\log_{10}(C_{nm})$

$S_{nm}$
Example: 2dim ISAW simulation up to \( N = 1024 \)

Total sample size: 260,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 270,000,000

- Rosenbluth
- PERM
- Multicanonical PERM
- FlatPERM

Thomas Prellberg

PERM and all that
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 280,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: 290,000,000
Example: 2dim ISAW simulation up to $N = 1024$

Total sample size: $300,000,000$

$\log_{10}(C_{nm})$

$S_{nm}$
2dim ISAW density of states

- 2d ISAW up to \( n = 1024 \)
- One simulation suffices
- 400 orders of magnitude

T Prellberg and J Krawczyk, PRL 92 (2004) 120602
Outline

1. Introduction
   - A Zoology of Growth Algorithms
   - Which Algorithm is Best?
   - ISAW - the canonical lattice model

2. The ‘Old’ Algorithms
   - Rosenbluth
   - PERM
   - Multicanonical PERM
   - FlatPERM

3. The ‘New’ Algorithms
   - New Ideas
   - GARM
   - GAS

4. Conclusion
   - Outlook
   - Thanks
Revisit Rosenbluth Sampling

- Each configuration grown uniquely by appending edges to endpoint

![Diagram of configuration growth](image-url)
Revisit Rosenbluth Sampling

- Each configuration grown uniquely by appending edges to endpoint

- Generating tree
  - Each node of tree is a configuration
Revisit Rosenbluth Sampling

- Each configuration grown uniquely by appending edges to endpoint

Generating tree
- Each node of tree is a configuration
- Sample by growing unique “sample path” down the tree
Revisit Rosenbluth Sampling

- Each configuration grown uniquely by appending edges to endpoint

- Generating tree
  - Each node of tree is a configuration
  - Sample by growing unique “sample path” down the tree
  - The weight of sample path is $W(\varphi) = \prod_{k<N} a_k(\varphi)$
From Generating Trees to Generating Graphs

- Unique way to construct walks
From Generating Trees to Generating Graphs

- Unique way to construct walks
- No obvious unique way to construct polygons
From Generating Trees to Generating Graphs

- Unique way to construct walks
- No obvious unique way to construct polygons
- Can we generalize from generating trees?
From Generating Trees to Generating Graphs

- Unique way to construct walks
- No obvious unique way to construct polygons
- Can we generalize from generating trees?

Generating graph
- Each node of graph is a configuration
From Generating Trees to Generating Graphs

- Generating graph
  - Sample by growing non-unique path down the graph
  - Each node of graph is a configuration

- Weight of the sample path is \( \prod_{k < N} a_k(\phi) \)

- Unique way to construct walks
  - No obvious unique way to construct polygons
  - Can we generalize from generating trees?

New Ideas
- GARM
- GAS

From Generating Trees to Generating Graphs

- New way to construct walks
- Can we generalize from generating trees?
From Generating Trees to Generating Graphs

- Unique way to construct walks
- No obvious unique way to construct polygons
- Can we generalize from generating trees?

Generating graph
- Each node of graph is a configuration
- Sample by growing non-unique path down the graph
- The weight of the sample path is $W(\varphi) \neq \prod_{k<N} a_k(\varphi)$
Atmospheres

- Positive and negative atmospheres of the configuration
  - Let $a^+$ be the number of ways a configuration can grow
  - Let $a^-$ be the number of ways a configuration can shrink
Atmospheres

- Positive and negative atmospheres of the configuration
  - Let $a^+$ be the number of ways a configuration can grow
  - Let $a^-$ be the number of ways a configuration can shrink
- Generating tree: bijection between sample paths and configurations
Positive and negative atmospheres of the configuration

- Let $a^+$ be the number of ways a configuration can grow
- Let $a^-$ be the number of ways a configuration can shrink

Generating tree: bijection between sample paths and configurations

Rosenbluth Sampling (with $a^- = 1$)

The weight $W(\varphi)$ and probability $Pr(\varphi)$ of a sample path $\varphi$ are

$$W(\varphi) = \prod_{k<N} a_k^+ (\varphi) \quad Pr(\varphi) = 1/W(\varphi)$$
Atmospheres

- Positive and negative atmospheres of the configuration
  - Let $a^+$ be the number of ways a configuration can grow
  - Let $a^-$ be the number of ways a configuration can shrink
- Generating tree: bijection between sample paths and configurations
- Rosenbluth Sampling (with $a^- = 1$)
  - The weight $W(\varphi)$ and probability $\Pr(\varphi)$ of a sample path $\varphi$ are
    \[ W(\varphi) = \prod_{k < N} a^+_k(\varphi) \quad \Pr(\varphi) = 1/W(\varphi) \]
- This implies
  \[ \sum_{\varphi} W(\varphi) \Pr(\varphi) = \sum_{\varphi} 1 = C_N \]
From Rosenbluth Sampling to GARM

- Generating tree: bijection between sample paths and configurations

\[ W(\varphi) = \prod_{k<N} a_k^+(\varphi) \quad \sum_\varphi W(\varphi) \Pr(\varphi) = C_N \]
From Rosenbluth Sampling to GARM

- **Generating tree:** bijection between sample paths and configurations

\[ W(\varphi) = \prod_{k<N} a_k^+(\varphi) \quad \sum_{\varphi} W(\varphi) \Pr(\varphi) = C_N \]

- **Generating graph:** many sample paths give the same configuration

\[ W(\varphi) = \prod_{k<N} a_k^+(\varphi) \quad \sum_{\varphi} W(\varphi) \Pr(\varphi) \gg C_N \]
From Rosenbluth Sampling to GARM

- Generating tree: bijection between sample paths and configurations
  \[ W(\varphi) = \prod_{k<N} a_k^+(\varphi) \quad \sum_{\varphi} W(\varphi) \Pr(\varphi) = C_N \]

- Generating graph: many sample paths give the same configuration
  \[ W(\varphi) = \prod_{k<N} a_k^+(\varphi) \quad \sum_{\varphi} W(\varphi) \Pr(\varphi) \gg C_N \]

- The correct weight
  \[ W(\varphi) = \prod_{k<N} \frac{a_k^+(\varphi)}{a_k^-(\varphi)} \quad \sum_{\varphi} W(\varphi) \Pr(\varphi) = C_N \]

EJJ van Rensburg and A Rechnitzer, J Phys A 41 (2008) 442002
From Rosenbluth Sampling to GARM

- Generating tree: bijection between sample paths and configurations

\[ W(\varphi) = \prod_{k < N} a_k^+(\varphi) \quad \sum_{\varphi} W(\varphi) \Pr(\varphi) = C_N \]

- Generating graph: many sample paths give the same configuration

\[ W(\varphi) = \prod_{k < N} a_k^+(\varphi) \quad \sum_{\varphi} W(\varphi) \Pr(\varphi) \gg C_N \]

- The correct weight

\[ W(\varphi) = \prod_{k < N} \frac{a_k^+(\varphi)}{a_k^-(\varphi)} \quad \sum_{\varphi} W(\varphi) \Pr(\varphi) = C_N \]

- GARM = Generalized Atmospheric Rosenbluth Method

EJJ van Rensburg and A Rechnitzer, J Phys A 41 (2008) 442002
GARM is a genuine generalization of Rosenbluth sampling
Features of GARM

GARM is a genuine generalization of Rosenbluth sampling

- Can easily substitute GARM for Rosenbluth sampling
  - Thermal GARM
  - Pruned Enriched GARM
  - Multicanonical GARM (not done yet!)
  - Flat Histogram GARM

Drawback: atmospheres may be expensive to calculate

Important Extension

Can include conventional canonical Monte Carlo moves

Need to know \( a_0 \), the atmosphere of neutral moves

Good ideas are welcome!
Features of GARM

GARM is a genuine generalization of Rosenbluth sampling

- Can easily substitute GARM for Rosenbluth sampling
  - Thermal GARM
  - Pruned Enriched GARM
  - Multicanonical GARM (not done yet!)
  - Flat Histogram GARM

- Applicable to polygons, branched polymers, lattice animals, ...

- Drawback: atmospheres may be expensive to calculate
Features of GARM

GARM is a genuine generalization of Rosenbluth sampling

- Can easily substitute GARM for Rosenbluth sampling
  - Thermal GARM
  - Pruned Enriched GARM
  - Multicanonical GARM (not done yet!)
  - Flat Histogram GARM

- Applicable to polygons, branched polymers, lattice animals, . . .
- Drawback: atmospheres may be expensive to calculate

Important Extension

- Can include conventional canonical Monte Carlo moves
- Need to know $a^0$, the atmosphere of neutral moves

Good ideas are welcome!
Grow and Shrink

- GARM works for 2d polygons, but not 3d polygons
Grow and Shrink

- GARM works for 2d polygons, but not 3d polygons
- There are 3 minimal 3d unknots on $\mathbb{Z}^3$ and 3328 minimal trefoils cannot reach one minimal configuration from another by growing
Grow and Shrink

- GARM works for 2d polygons, but not 3d polygons
- There are 3 minimal 3d unknots on $\mathbb{Z}^3$ and 3328 minimal trefoils cannot reach one minimal configuration from another by growing
- Use moves that grow and shrink
Grow and Shrink

- GARM works for 2d polygons, but not 3d polygons
- There are 3 minimal 3d unknots on $\mathbb{Z}^3$ and 3328 minimal trefoils cannot reach one minimal configuration from another by growing
- Use moves that grow and shrink

- Moves from the BFACF algorithm
  - C Aragão de Carvalho, S Caracciolo and J Fröhlich, Nucl Phys B 215 (1983) 209

- Ergodic on each knot-type
  - EJJ van Rensburg, J Phys A 25 (1992) 1031
Grow and Shrink

- GARM works for 2d polygons, but not 3d polygons
- There are 3 minimal 3d unknots on $\mathbb{Z}^3$ and 3328 minimal trefoils cannot reach one minimal configuration from another by growing
- Use moves that grow and shrink

![Diagram of growing and shrinking](image)

- Generating graph still exists, but now sample paths are not directed
- Need to “redirect” the graph
Take an arbitrary generating graph
Derivative graph

- Copy the initial vertex
Derivative graph

- What vertices does it see? — add them to the next row
Derivative graph

What vertices do these see? — both up and down
Keep adding new rows in this way
Keep adding new rows in this way
This gives the “derivative graph”
GAS = Generalized Atmospheric Sampling = Grow And Shrink

- Do GARM sampling on the derivative graph
From GARM to GAS

GAS = Generalized Atmospheric Sampling = Grow And Shrink

EJJ van Rensburg and A Rechnitzer, J Phys A 42 (2009) 335001

- Do GARM sampling on the derivative graph
- Weight is a simple function of $a^\pm(\varphi), a^0(\varphi)$

$$\frac{\langle W(\varphi) \rangle_N}{\langle W(\varphi) \rangle_M} = \frac{C_N}{C_M}$$
GAS = Generalized Atmospheric Sampling = Grow And Shrink

Do GARM sampling on the derivative graph
Weight is a simple function of $a^\pm(\varphi), a^0(\varphi)$

$$\frac{\langle W(\varphi) \rangle_N}{\langle W(\varphi) \rangle_M} = \frac{C_N}{C_M}$$

Generalizes to Thermal GAS and Pruned Enriched GAS
From GARM to GAS

GAS = Generalized Atmospheric Sampling = Grow And Shrink

- Do GARM sampling on the derivative graph
- Weight is a simple function of $a^\pm(\varphi), a^0(\varphi)$

$$\frac{\langle W(\varphi) \rangle_N}{\langle W(\varphi) \rangle_M} = \frac{C_N}{C_M}$$

- Generalizes to Thermal GAS and Pruned Enriched GAS
- Multicanonical and Flat Histogram GAS seems harder

Under development

EJJ van Rensburg and A Rechnitzer, J Phys A 42 (2009) 335001

A Rechnitzer, private communication
**GAS Application: Minimal Polygons**

- Known exactly for trefoil $C_{24}(3_1) = 3328$

  Y. Diao, JKTR 2 (1993) 413
GAS Application: Minimal Polygons

- Known exactly for trefoil $C_{24}(3_1) = 3328$  
  \[ Y \text{ Diao, JKTR 2 (1993) 413} \]
- Need to estimate numerically for other knot types
  - Draw a knot $K$ on the cubic lattice
  - Run GAS with BFACF moves and extract the minimal polygons
  \[ EJJ \text{ van Rensburg and A Rechnitzer, JKTR, in print} \]
GAS Application: Minimal Polygons

- Known exactly for trefoil $C_{24}(3_1) = 3328$  
  Y Diao, JKTR 2 (1993) 413
- Need to estimate numerically for other knot types
  - Draw a knot $K$ on the cubic lattice
  - Run GAS with BFACF moves and extract the minimal polygons
  EJJ van Rensburg and A Rechnitzer, JKTR, in print
- Resulting numbers

$$C_{24}(3_1) = 3328$$
$$C_{30}(4_1) = 2648$$
$$C_{34}(5_1) = 6672$$
$$C_{36}(5_2) = 114912$$

see also R Scharein et al, J Phys A 42 (2009) 475006
GAS Application: Minimal Polygons

- Known exactly for trefoil $C_{24}(3_1) = 3328$  
  Y Diao, JKTR 2 (1993) 413
- Need to estimate numerically for other knot types
  - Draw a knot $K$ on the cubic lattice
  - Run GAS with BFACF moves and extract the minimal polygons
  EJJ van Rensburg and A Rechnitzer, JKTR, in print
- Resulting numbers
  
<table>
<thead>
<tr>
<th>Knot Type</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>$C_{24}(3_1)$</td>
<td>3328</td>
</tr>
<tr>
<td>$C_{30}(4_1)$</td>
<td>2648</td>
</tr>
<tr>
<td>$C_{34}(5_1)$</td>
<td>6672</td>
</tr>
<tr>
<td>$C_{36}(5_2)$</td>
<td>114912</td>
</tr>
</tbody>
</table>

  see also R Scharein et al, J Phys A 42 (2009) 475006

- This can now be used to estimate e.g. the number of figure eight knots

\[
\frac{C_N(4_1)}{C_{30}(4_1)} = \frac{\langle W(\varphi) \rangle_N}{\langle W(\varphi) \rangle_{30}}
\]
Outline

1 Introduction
   - A Zoology of Growth Algorithms
   - Which Algorithm is Best?
   - ISAW - the canonical lattice model

2 The 'Old' Algorithms
   - Rosenbluth
   - PERM
   - Multicanonical PERM
   - FlatPERM

3 The 'New' Algorithms
   - New Ideas
   - GARM
   - GAS

4 Conclusion
   - Outlook
   - Thanks
Comparing the Algorithms?

- Testing flatPERM using 1-dim random walk

  JD Jiang and YN Huang, Comp Phys Commun 180 (2009) 177
Comparing the Algorithms?

- Testing flatPERM using 1-dim random walk

  JD Jiang and YN Huang, Comp Phys Commun 180 (2009) 177

- Difference between multicanonical PERM and flatPERM?
Comparing the Algorithms?

- Testing flatPERM using 1-dim random walk
  
  JD Jiang and YN Huang, Comp Phys Commun 180 (2009) 177

- Difference between multicanonical PERM and flatPERM?
  
  About the same...

  M Bachmann, private communication
Comparing the Algorithms?

- Testing flatPERM using 1-dim random walk
  
  JD Jiang and YN Huang, Comp Phys Commun 180 (2009) 177

- Difference between multicanonical PERM and flatPERM?
  About the same...

- More importantly, are GARM/GAS better?
Comparing the Algorithms?

- Testing flatPERM using 1-dim random walk
  JD Jiang and YN Huang, Comp Phys Commun 180 (2009) 177

- Difference between multicanonical PERM and flatPERM?
  About the same...  
  M Bachmann, private communication

- More importantly, are GARM/GAS better?
  YES, as GARM/GAS works where Rosenbluth does not
  NO, as large atmospheres might be very expensive to compute
  (but maybe a trade-off with better sampling?)
Comparing the Algorithms?

- Testing flatPERM using 1-dim random walk
  
  JD Jiang and YN Huang, Comp Phys Commun 180 (2009) 177

- Difference between multicanonical PERM and flatPERM?
  
  About the same…
  
  M Bachmann, private communication

- More importantly, are GARM/GAS better?
  
  YES, as GARM/GAS works where Rosenbluth does not
  
  NO, as large atmospheres might be very expensive to compute
  (but maybe a trade-off with better sampling?)

- Algorithms are ‘dirty’ (highly correlated data)
Comparing the Algorithms?

- Testing flatPERM using 1-dim random walk
  JD Jiang and YN Huang, Comp Phys Commun 180 (2009) 177

- Difference between multicanonical PERM and flatPERM?
  About the same...
  M Bachmann, private communication

- More importantly, are GARM/GAS better?
  YES, as GARM/GAS works where Rosenbluth does not
  NO, as large atmospheres might be very expensive to compute
    (but maybe a trade-off with better sampling?)

- Algorithms are ‘dirty’ (highly correlated data)
  Can one prove anything useful?
Comparing the Algorithms?

- Testing flatPERM using 1-dim random walk
  
  JD Jiang and YN Huang, Comp Phys Commun 180 (2009) 177

- Difference between multicanonical PERM and flatPERM?
  About the same...

- More importantly, are GARM/GAS better?
  YES, as GARM/GAS works where Rosenbluth does not
  NO, as large atmospheres might be very expensive to compute
  (but maybe a trade-off with better sampling?)

- Algorithms are ‘dirty’ (highly correlated data)
  Can one prove anything useful?

Many more applications for GARM/GAS?
Monte Carlo Collaborators

- Jason Doukas (Kyoto)
- Jarek Krawczyk (Dortmund)
- Aleks Owzcarek (Melbourne)
- Andrew Rechnitzer (Vancouver)
- Buks van Rensburg (Toronto)

Monte Carlo methods for the self-avoiding walk, J Phys A 42 (2009) 323001

$$$  
- Deutsche Forschungsgemeinschaft (DFG)  
- MASCOS  
- Royal Society

Special thanks to Andrew for the GARM/GAS figures