Recent Advancements in Adaptive Kinetic Monte Carlo

Samuel T. Chill and Graeme Henkelman

Henkelman Group
Department of Chemistry
The University of Texas at Austin

June 7, 2013

Overview

Background: Intoduction to Adaptive Kinetic Monte Carlo (AKMC)

Two Improvements to AKMC:
Part I: Improved saddle searches using molecular dynamics

- Compare effiency to min-mode following searches
- Estimate the uncertainty in the rate table

Part II: Treating superbasins using absorbing Markov chains

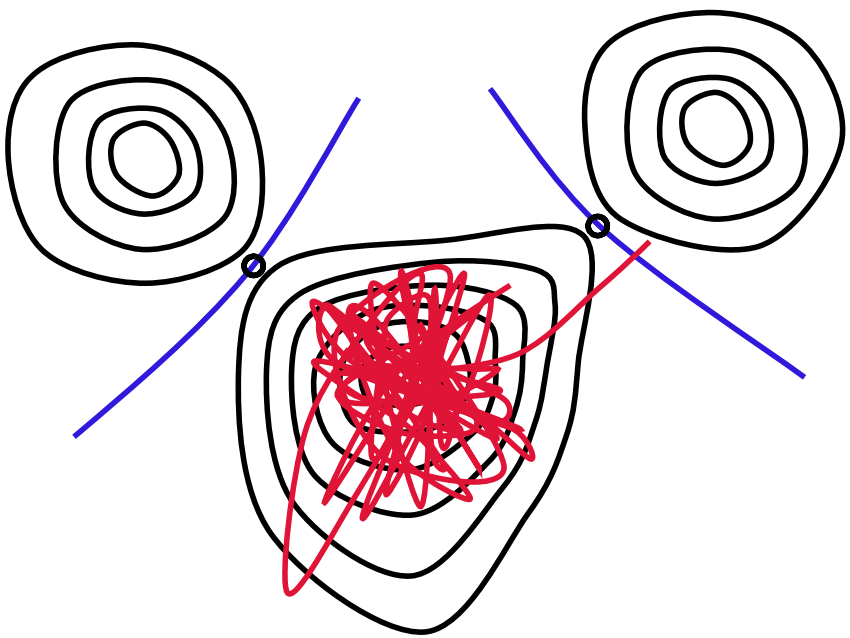
- When isn't KMC fast enough
- Monte Carlo with absorbing Markov Chains (MCAMC)

Part I Problem Description

Rare Event System

A chemical system where the atoms spend large amount of time in each energy basin before transitioning to the next.

- Cannot use molecular dynamics



Reproduced from Art Voter's "Introduction to The Kinetic Monte Carlo Method"

How to efficiency model the state-to-state dynamics?

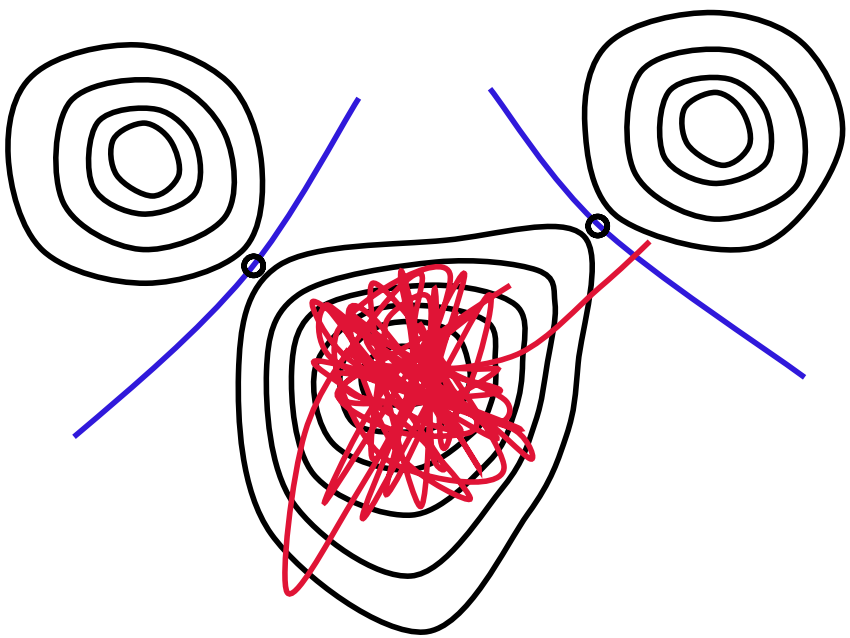
- Parallelize over time
- Accelerated Dynamics
  - Alter potential energy surface
  - Increase temperature
- Statistical mechanics

Part I Problem Description

Rare Event System

A chemical system where the atoms spend large amount of time in each energy basin before transitioning to the next.

- Cannot use molecular dynamics



Reproduced from Art Voter's "Introduction to The Kinetic Monte Carlo Method"

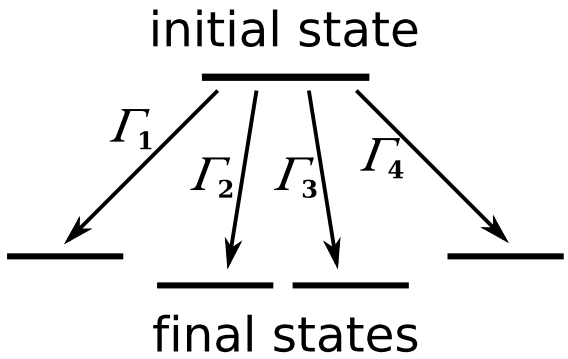
How to efficiency model the state-to-state dynamics?

- Parallelize over time
- Accelerated Dynamics
  - Alter potential energy surface
  - Increase temperature
- Statistical mechanics

Kinetic Monte Carlo

Models state-to-state dynamics as a Markov chain

- The states must be Markovian
- Next event is chosen in proportion to its rate
- Escape time drawn from exponential distribution



$$P\left[\Delta t\right] = \exp\left(-\Delta t \sum_i \Gamma_i\right)$$

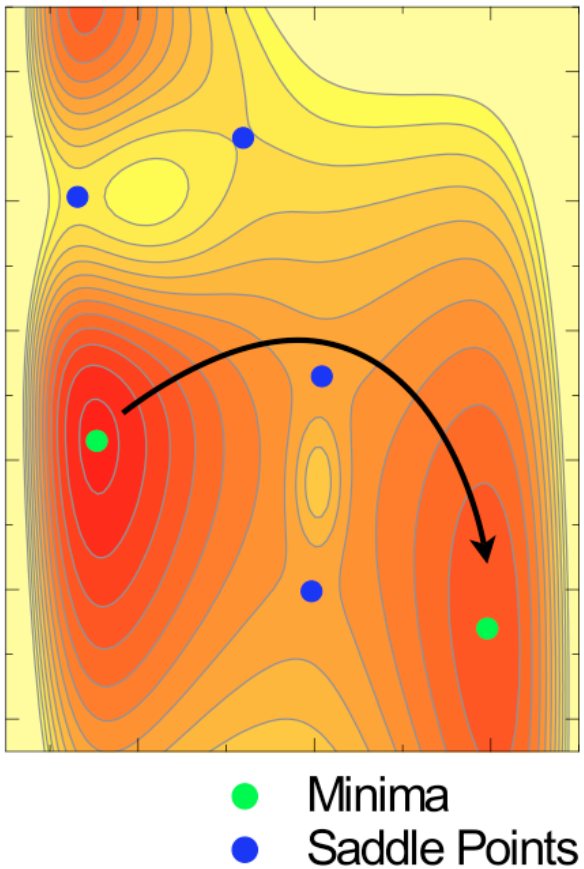
- Very fast (two random numbers and some book keeping)

How to build the Markov model? How to get the states and the rates?

Adaptive Kinetic Monte Carlo

Algorithm:

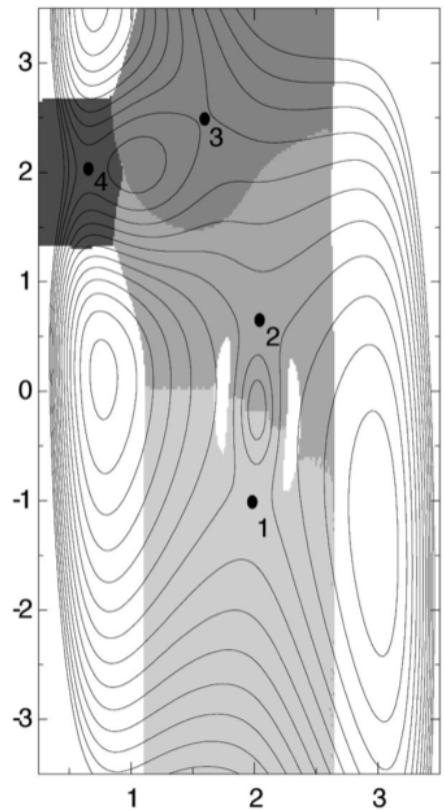
1. Initialize a saddle search by displacing randomly from the current minimum
2. Use a min-mode saddle search with the minimum curvature mode estimated with dimer or Lanczos algorithms
3. Minimize along negative curvature at saddle to find new product state
4. Ensure connectivity
5. Calculate rate using HTST
6. Estimate confidence that all important saddles have been found
7. Take KMC steps until a new state is reached



Adaptive Kinetic Monte Carlo

Algorithm:

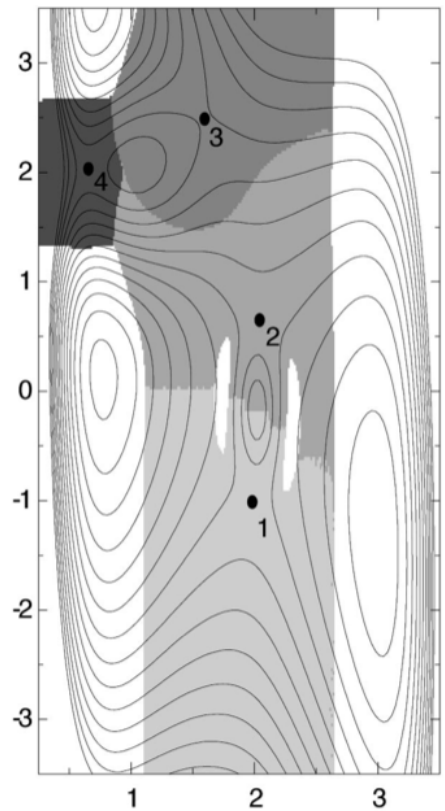
1. Initialize a saddle search by displacing randomly from the current minimum
2. Use a min-mode saddle search with the minimum curvature mode estimated with dimer or Lanczos algorithms
3. Minimize along negative curvature at saddle to find new product state
4. Ensure connectivity
5. Calculate rate using HTST
6. Estimate confidence that all important saddles have been found
7. Take KMC steps until a new state is reached



Adaptive Kinetic Monte Carlo

Algorithm:

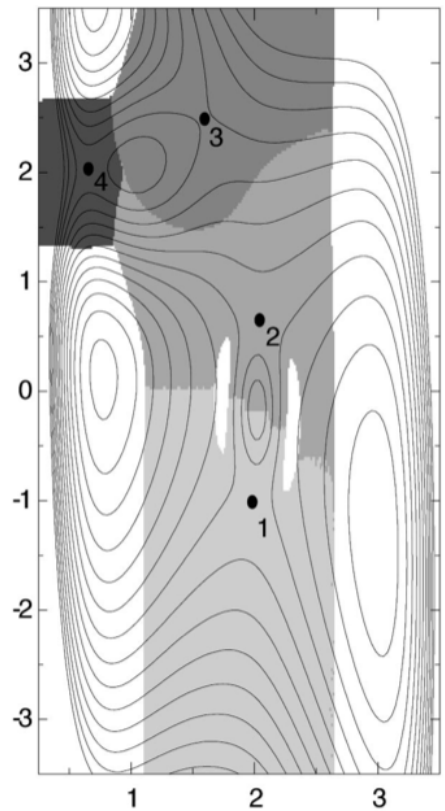
1. Initialize a saddle search by displacing randomly from the current minimum
2. Use a min-mode saddle search with the minimum curvature mode estimated with dimer or Lanczos algorithms
3. Minimize along negative curvature at saddle to find new product state
4. Ensure connectivity
5. Calculate rate using HTST
6. Estimate confidence that all important saddles have been found
7. Take KMC steps until a new state is reached



Adaptive Kinetic Monte Carlo

Algorithm:

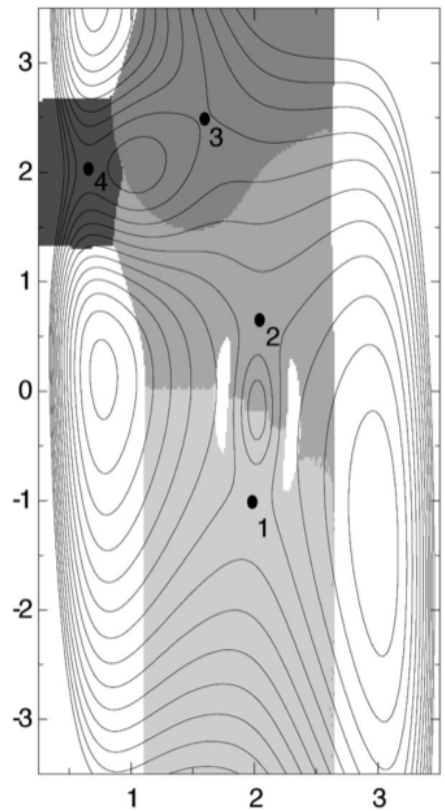
1. Initialize a saddle search by displacing randomly from the current minimum
2. Use a min-mode saddle search with the minimum curvature mode estimated with dimer or Lanczos algorithms
3. Minimize along negative curvature at saddle to find new product state
4. Ensure connectivity
5. Calculate rate using HTST
6. Estimate confidence that all important saddles have been found
7. Take KMC steps until a new state is reached



Adaptive Kinetic Monte Carlo

Algorithm:

1. Initialize a saddle search by displacing randomly from the current minimum
2. Use a min-mode saddle search with the minimum curvature mode estimated with dimer or Lanczos algorithms
3. Minimize along negative curvature at saddle to find new product state
4. Ensure connectivity
5. Calculate rate using HTST
6. Estimate confidence that all important saddles have been found
7. Take KMC steps until a new state is reached

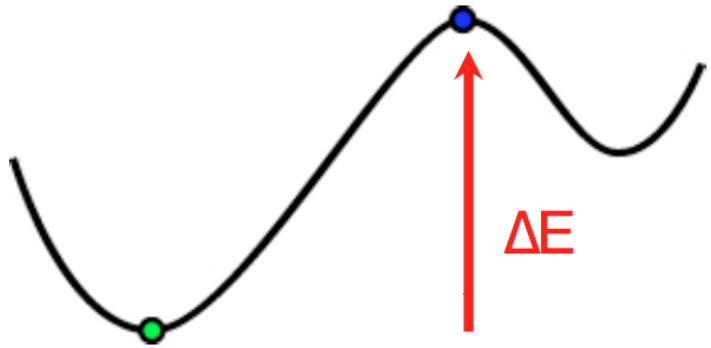


Calculating Reaction Rates using HTST

Use harmonic transition state theory (HTST) to calculate rates for solid state systems:

$$A = \frac{\prod_i^{3N} \nu_i^{\text{min}}}{\prod_i^{3N-1} \nu_i^{\text{saddle}}}$$

$$r = \underbrace{A}_{\text{Attempt Freq.}} \cdot \underbrace{\exp\left[-\Delta E / k_{\text{B}} T\right]}_{\text{Boltzmann Probability}}$$



Assumptions:

- Potential energy surface is harmonic
- The reactant is in equilibrium with activated complex (saddle)
- Trajectories at the saddle will always go to the product state (no recrossing)

Works very well for solid state systems at low (room) temperature.

Min-mode searches Pros and Cons

Pros

- Computationally cheap per search (3-4x minimization)
- Easy to bias searches to subsets of atoms

Cons

- Prior knowledge required to perform displacements
- Can find saddles of high energy
- Important events can be difficult to find
- Difficult to estimate the error in the total rate
- Easy to find saddles that are not connected to initial state

Using High Temperature Molecular Dynamics to Find Saddles

Algorithm:

1. Run high temperature MD (possibility at different temperatures)
2. Periodically minimize the system to see if it has exited
3. Connect the initial minimum to the product minimum via a chain-of-states method such as Nudged Elastic Band
4. Find first maxima along the minimized band
5. Start a min-mode following saddle search to converge to saddle point

Pros and Cons of MD Searches compared to Min-mode

Pros

- Finds events with probability proportional to their rate
- Reduced chance of finding disconnected events
- Can estimate a confidence in the total rate found
- No prior knowledge needed about reaction coordinate

Cons

- Each MD search is more expensive than min-mode following search
- NEB is not guaranteed to find all pathways between the two states

Estimating the Error in the Rate

$$C(T_{\mathrm{KMC}}, T_{\mathrm{MD}}, t) = \frac{\left\langle \widetilde{\Gamma}(t) \right\rangle}{\widetilde{\Gamma}} = \frac{\sum_{i=1}^{N} \gamma_i(T_{\mathrm{KMC}})(1 - \exp(-t\gamma_i(T_{\mathrm{MD}})))}{\sum_{i=1}^{N} \gamma_i(T_{\mathrm{KMC}})}$$

N is the total number of events found thus far.

t is the total accumulated MD time.

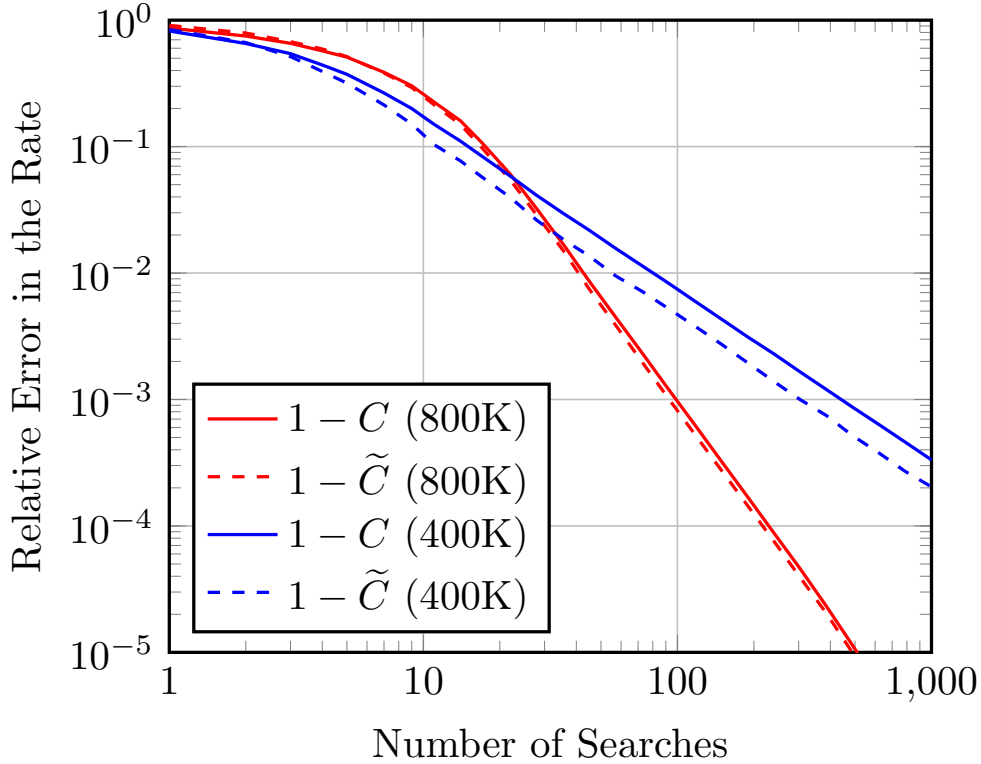
$\gamma_i(T)$ is a function that maps the temperature to the rate (can use HTST)

$1 - \exp(-t\gamma_i(T_{\mathrm{MD}}))$ represents the probability that an event with rate $\gamma_i(T_{\mathrm{MD}})$ has occurred after t seconds of MD time (CDF of exp. dist.)

$\widetilde{\Gamma}$ is the maximum likelihood estimate of the rate.

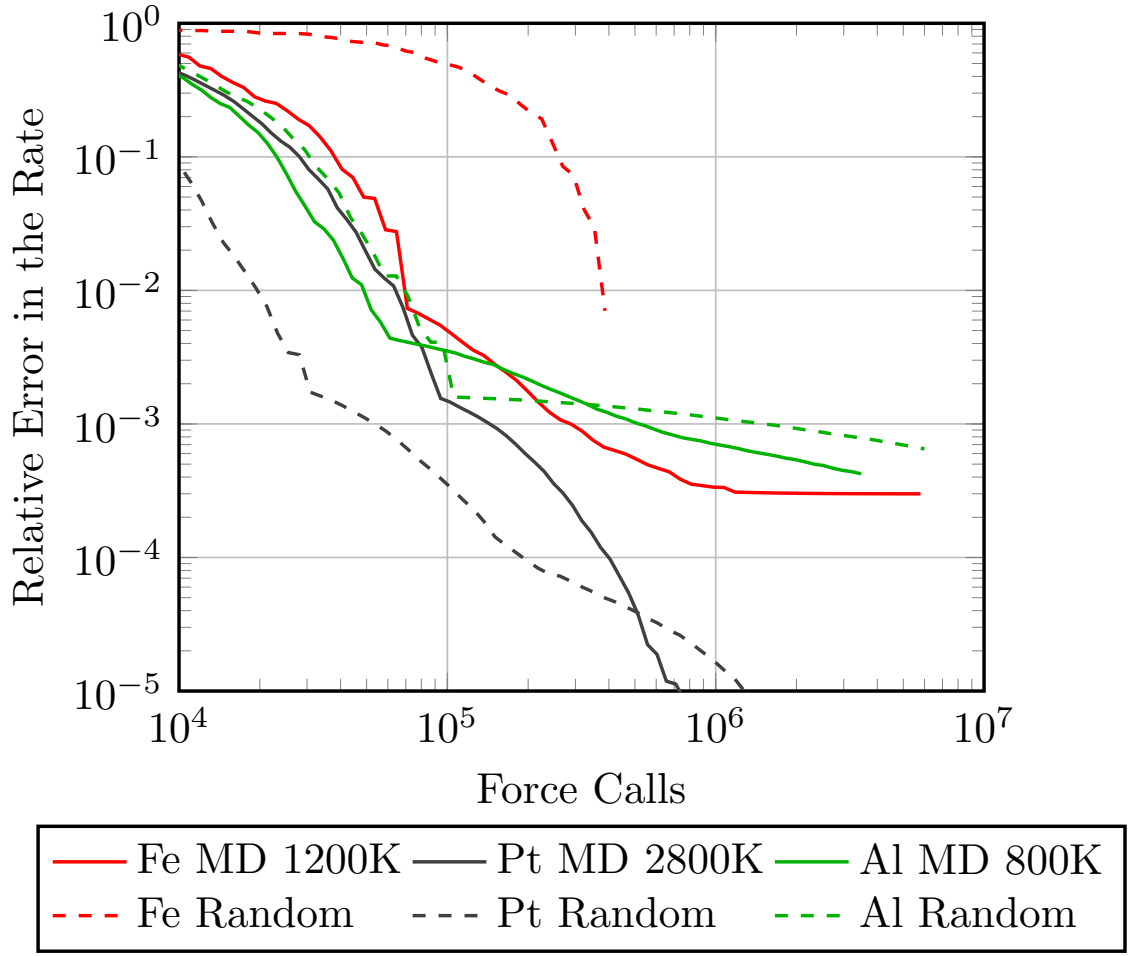
$\left\langle \widetilde{\Gamma}(t) \right\rangle$ is the expected rate after t seconds, which depends on the inter-arrival times and thus the total rate.

Evaluating The Quality of the Error Estimator



Test using a harmonic system with rates calculated via HTST.
20 barriers linearly spaced between 0.1 and 0.4 eV.
Higher temperature sampling converges the total rate faster and reduces the bias of the estimator.

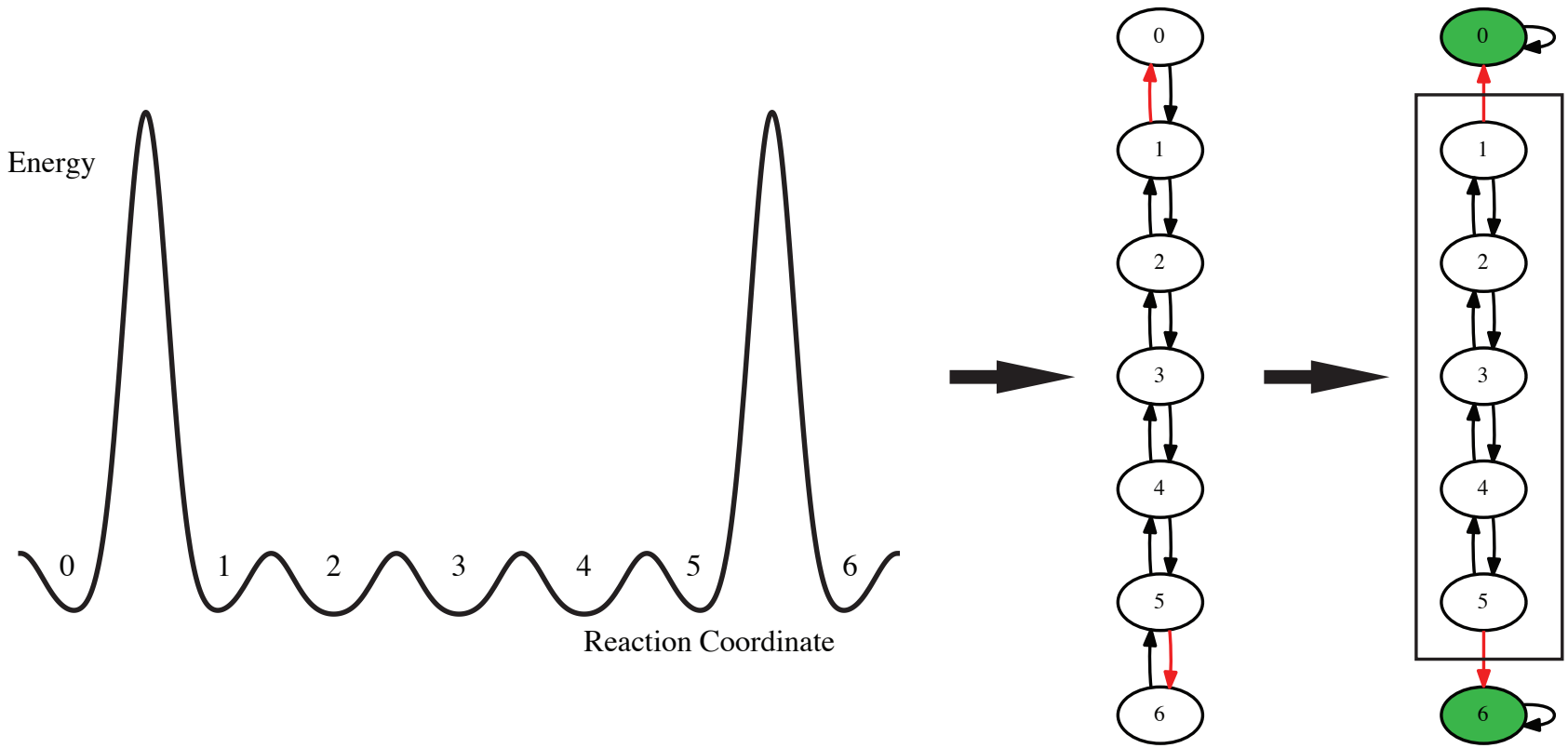
Saddle Search Method Comparison



Part II A second class of rare event problems

Superbasin

Superbasins are features of the potential energy surface, where there exist reactions that occur on widely differing timescales.



Monte Carlo with Absorbing Markov Chains

The *absorbing states*, r, are the exits from the superbasin and the *transient states*, s, are within the superbasin.

$$\mathbf{M}_{(r+s)\times(r+s)} = \begin{pmatrix} \mathbf{T}_{s\times s} & \mathbf{R}_{s\times r} \\ \mathbf{0}_{r\times s} & \mathbf{I}_{r\times r} \end{pmatrix}$$

Fundamental Matrix: mean number of times to visit state j starting at state i

$$\mathbf{N} = \sum_{k=0}^{\infty} \mathbf{T}^k = (\mathbf{I} - \mathbf{T})^{-1}$$

Matrix of absorption probabilities:

$$\mathbf{B} = \mathbf{NR}$$

$\mathbf{B}_{ij}$ is the probability of ending in state j when starting in state i

Mean time until absorption:

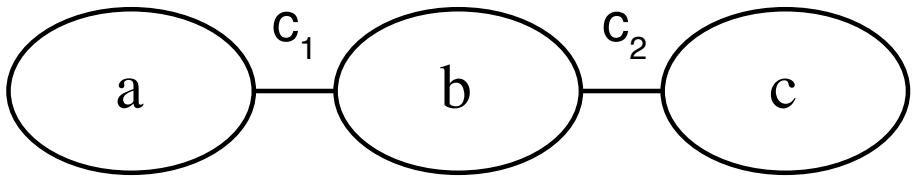
$$\mathbf{t} = \mathbf{N}\tau$$

τ is a vector of mean escape times from the transient states

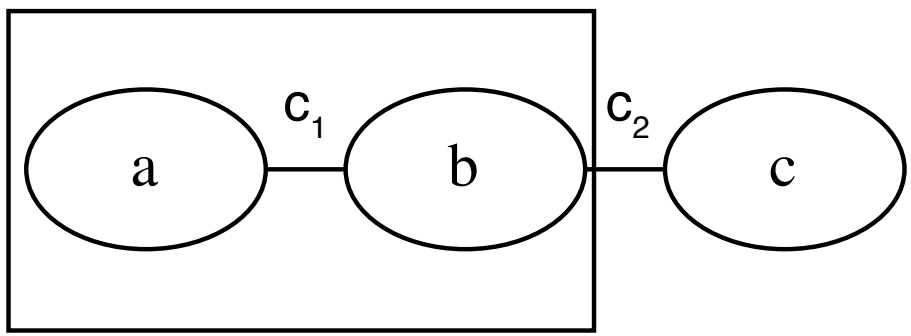
Grouping States During a KMC Simulation

Transition Counting

Identifying superbasins based on the number of times a transition has occurred.

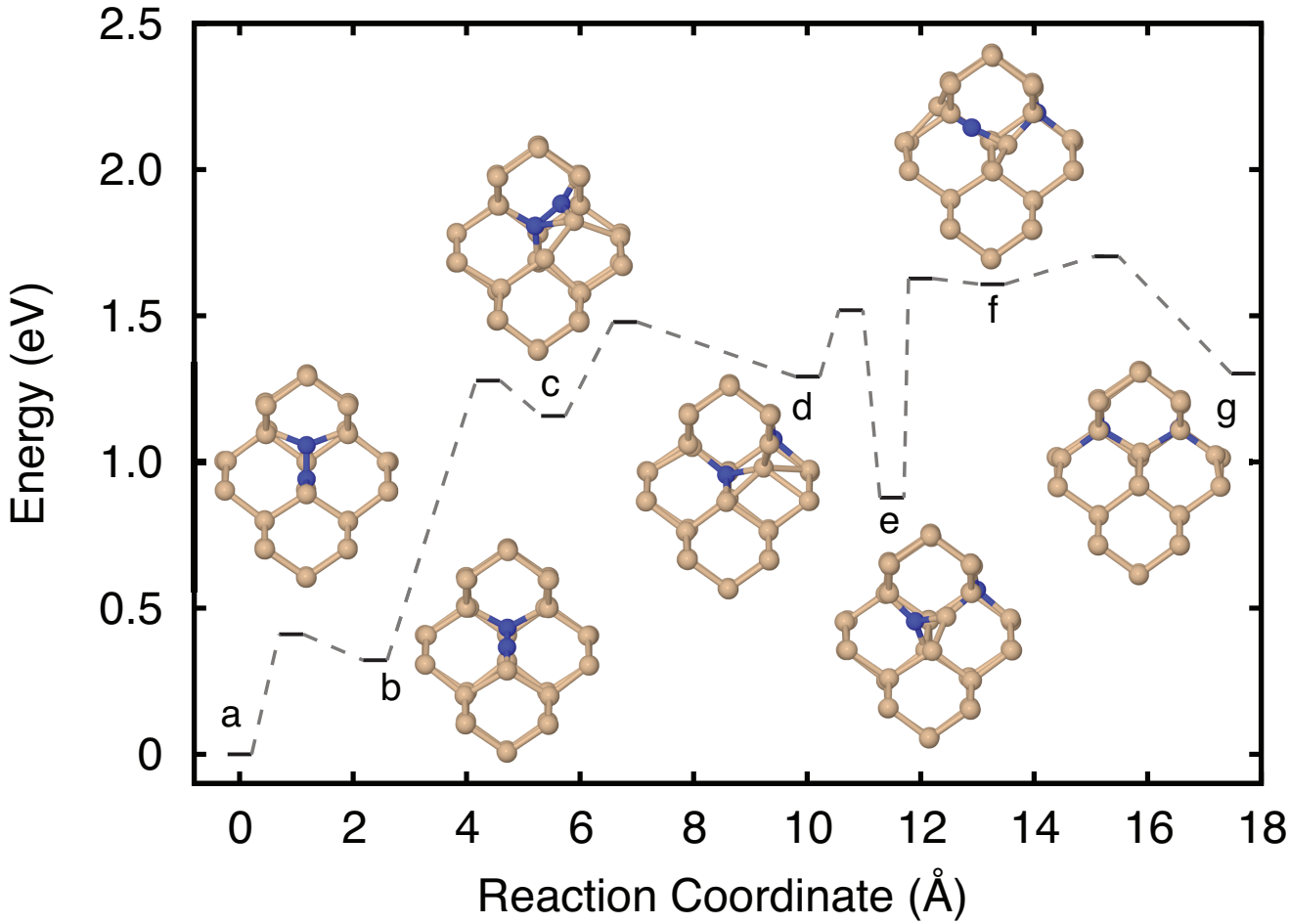


Increment the counter c_i each time a processes is followed. When $c_i \geq c_{\max}$ group the two states together.



Modeling SiB_2 cluster break-up with AKMC and DFT

States a and b form a superbasin that takes 5 billion steps to escape from on average at 300 K (about 20 days of computer time).



Performance of MCAMC

When modeling events on disparate timescales, extended precision types are sometimes needed.

N	`float` (7)	`double` (16)	`dd` (32)	`qd` (64)	`arb.` (154)
10	0.0004	0.0002	0.0004	0.0016	0.0083
100	0.0009	0.0011	0.0181	0.1768	1.0971
200	0.0025	0.0038	0.0932	0.9876	5.8999
500	0.0301	0.0439	1.1355	11.9715	64.7225
800	0.0602	0.1065	3.6915	39.6592	247.9850

Wall clock time in seconds for solving an absorbing Markov chain with N transient states. Computational effort is dominated by LU decomposition.

Drunkard's Walk

Open-source code for solving AMC problems in high precision.
`https://github.com/SamChill/drunkardswalk`

Eon: Open-source software for long timescale dynamics

Eon is software package that implements serveral long timescale dynamics methods. Details on obtaining the code at the end of this talk.

Methods

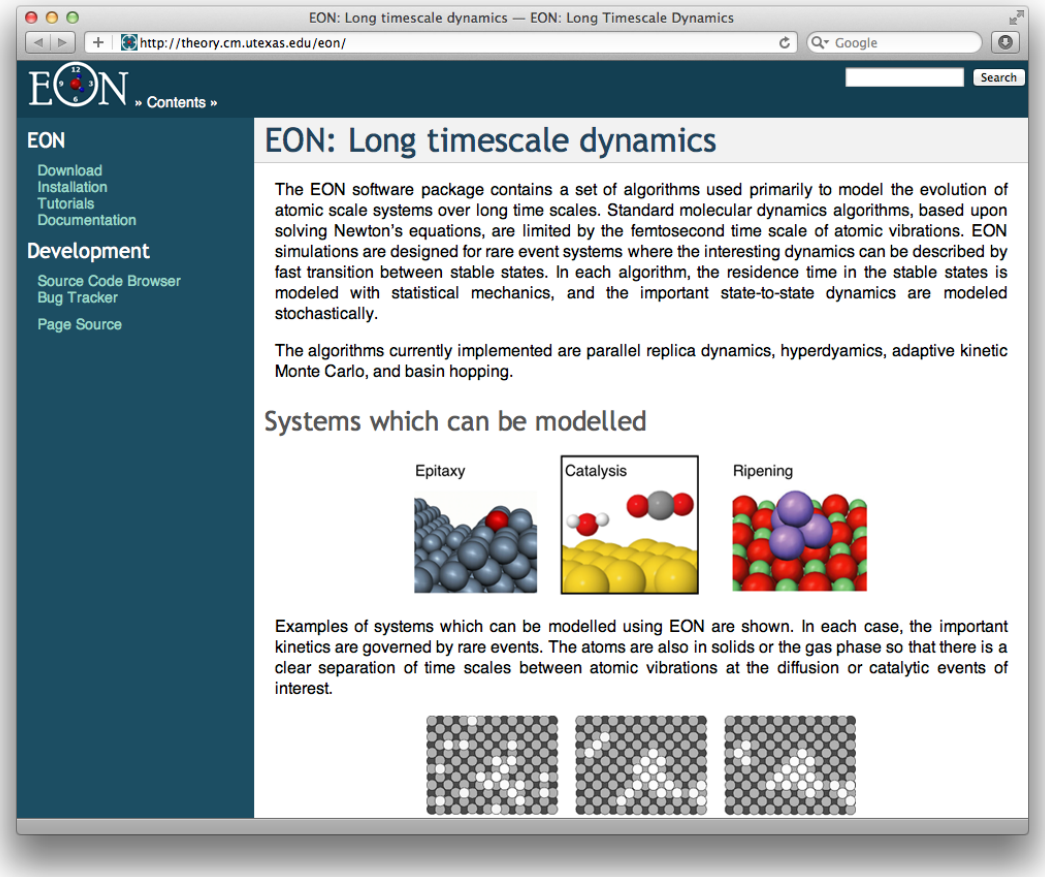
- Adaptive Kinetic Monte Carlo
- Parallel Replica
- Hyper Dynamics (Bond Boost)
- Basin Hopping
- κ-dynamics (in progress)
- Temperature Accelerated Dynamics (in progress)

Parallelization Options

- Local
- Job Queuing System (PBS, SGE, etc)
- BOINC (Distributed Computing)
- MPI

Potentials: EAM, LAMMPS, GPAW, VASP, and others.

Eon Homepage: http://theory.cm.utexas.edu/eon



S. T. Chill, M. Welborn, R. Terrell, L. Zhang, J. C. Berthet, A. Pedersen, H. Jonsson, and G. Henkelman *Model. Simul. Mater. Sci. Eng*, **Submitted 2013**.

Parallel Replica Dynamics

Pros

- Can be as accurate as vanilla MD
- Simple to implement
- Scales well with number of CPUs
- Can be used with other accelerated MD methods (hyperdynamics)

Cons

- No acceleration per CPU
- How to efficiently draw replicas from QSD
- Does not address hierarchy of rare event problems

Hyperdynamics

Pros

- Can be as accurate as vanilla MD (tricky, see below)
- Exponential speedup to see escapes (depends on bias potential)

Cons

- Does not address hierarchy of rare event problems

How to construct good bias potential?

- Must go to zero at transition surface
- Exponential average must converge
- Boost should still be large

Temperature Accelerated Dynamics

Pros

- Exponential speedup
- Requires less prior knowledge than hyperdynamics

Cons

- hTST must hold at the dynamics temperature (typically large)
- Does not address hierarchy of rare event problems

Adaptive Kinetic Monte Carlo

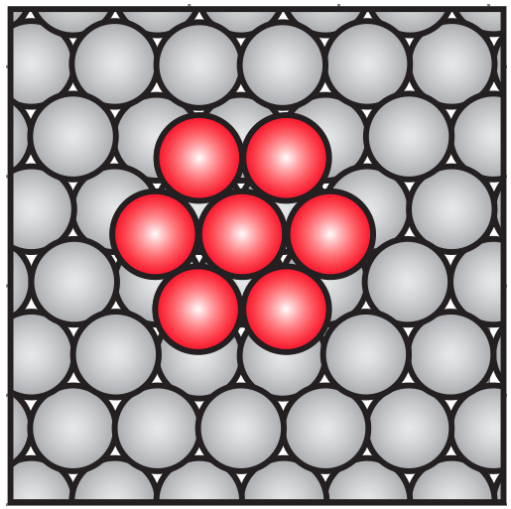
Pros

- Large speedups compared to molecular dynamics
- Calculate temperature dependence with minimal effort
- Re-use information from previous states (recycling, KDB)
- Quickly calculate time distributions from Markov chain

Cons

- hTST must hold at the simulation temperature
- How to ensure states are Markovian?
- Difficulty in calculating confidence/error
- How to find the important events without prior knowledge
- Does not address hierarchy of rare event problems

Method Comparison



Method	Escape Rate (1/s)	Force Calls
Parallel Replica	$(7.7 \pm 3.8) \times 10^5$	1×10^9
Parallel Replica / HD	$(3.4 \pm 1.7) \times 10^5$	1×10^7
AKMC	7.0×10^5	2×10^4

Comparison of the rate of escape from the Pt heptamer island state at 400K using different long timescale methods.