LLNL-TR-517795

LAWRENCE LIVERMORE NATIONAL LABORATORY

A new class of accelerated kinetic Monte Carlo algorithms

V. V. Bulatov, T. Oppelstrup, M. Athenes

December 2, 2011

This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

A new class of accelerated kinetic Monte Carlo algorithms

Vasily V. Bulatov, Tomas Oppelstrup and Manuel Athenes

Introduction and background

Kinetic (*aka* dynamic) Monte Carlo (KMC) is a powerful method for numerical simulations of time dependent evolution applied in a wide range of contexts including biology, chemistry, physics, nuclear sciences, financial engineering, etc. Generally, in a KMC the time evolution takes place one event at a time, where the sequence of events and the time intervals between them are selected (or sampled) using random numbers. While details of the method implementation vary depending on the model and context, there exist certain common issues that limit KMC applicability in almost all applications. Among such is the notorious "flicker problem" where the same states of the systems are repeatedly visited but otherwise no essential evolution is observed. In its simplest form the flicker problem arises when two states are connected to each other by transitions whose rates far exceed the rates of all other transitions out of the same two states. In such cases, the model will endlessly hop between the two states otherwise producing no meaningful evolution. In most situation of practical interest, the trapping cluster includes more than two states making the flicker somewhat more difficult to detect and to deal with.

Several methods have been proposed to overcome or mitigate the flicker problem, exactly [1-3] or approximately [4,5]. Of the exact methods, the one proposed by Novotny [1] is perhaps most relevant to our research. Novotny formulates the problem of escaping from a trapping cluster as a Markov system with absorbing states. Given an initial state inside the cluster, it is in principle possible to solve the Master Equation for the time dependent probabilities to find the walker in a given state (transient or absorbing) of the cluster at any time in the future. Novotny then proceeds to demonstrate implementation of his general method to trapping clusters containing the initial state plus one or two transient states and all of their absorbing states. Similar methods have been subsequently proposed in [refs] but applied in a different context. The most serious deficiency of the earlier methods is that size of the trapping cluster size is fixed and often too small to bring substantial simulation speedup. Furthermore, the overhead associated with solving for the probability distribution on the trapping cluster sometimes makes such simulations less efficient than the standard KMC.

Here we report on a general and exact accelerated kinetic Monte Carlo algorithm generally applicable to arbitrary Markov models[1]. Two different implementations are attempted both based on incremental expansion of trapping sub-set of Markov

[1] Practical applicability of the new method is limited to Markov models with low count (sparsity) of connections (transitions) from individual Markov states.

states: (1) numerical solution of the Master Equation with absorbing states and (2) incremental graph reduction followed by randomization. Of the two implementations, the 2nd one performs better allowing, for the first time, to overcome trapping basins spanning several million Markov states. The new method is used for simulations of anomalous diffusion on a 2D substrate and of the kinetics of diffusive 1st order phase transformations in binary alloys. Depending on temperature and (alloy) super-saturation conditions, speedups of 3 to 7 orders of magnitude are demonstrated, with no compromise of simulation accuracy.

Problem setting

At present, two distinctly different computational approaches to time-dependent evolution of Markov models exist: (1) to directly solve of the set of ODE representing the Master Equation (ME) for time-dependent occupation probabilities and (2) to use KMC method that samples a single stochastic trajectory from the unknown solution of the underlying ME. The first method is realistically applicable only to models with finite (and not too large) number of Markov states whereas the KMC method is applicable even when the number of accessible Markov states is infinite. Here we envision an adaptive method that combines the standard KMC sampling with solving the ME, depending on trapping (or no trapping) situation. We intend to allow the walker to explore the space by random walks when there is no serious trapping. On the other hand, when trapping is detected and deemed sufficiently serious, we intend to build a Master Equation on the trapping cluster of Markov states and use its solution to help the walker to escape out of the trap. Viewed from this angle, our method is a hybrid between the standard KMC [6] and the finite-state projection method developed in [7].

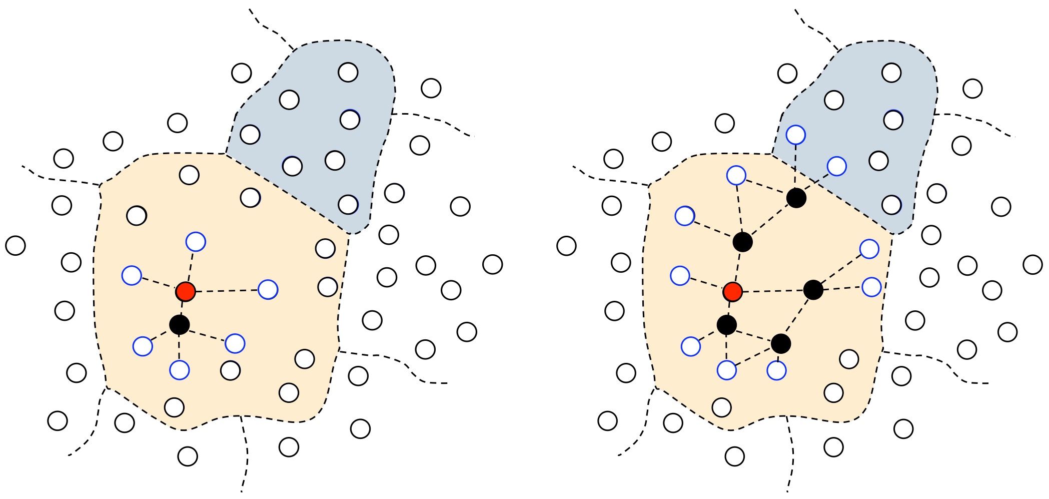

Figure 1: Two stages of Markov web expansion. On the left: one extra transient state (black) is already added to the initial state (red). The frontier states are a subset of unknown states (open) that are connected to the transient states by direct transitions (dashed lines). On the right: the web has expanded and now reaches from the initial trapping basin (beige) into the neighboring trapping basin (gray).

To enable adaptivity our hybrid algorithm should be able to tell when building and solving of the ME is justified and when it is best to leave the walker to simply walk. Our approach is inspired by the ideas advanced by D. Frenkel, G. Boulogouris and D, Theodorou [8,9] who proposed to enhance Monte Carlo sampling efficiency by using a Markovian web, i.e. expanding cluster of "known" states that a walker visits over a finite number of hops. All the states that are connected to the known states but have not been visited can be referred to as "unknown" or "frontier" states.

The idea is illustrated in Figure 1 where several stages of web expansion are schematically shown. One possible algorithm to expand the web is to promote one (or several) of the frontier states into a known state and inspect and establish all of its (their) connections to the already existing and new frontier states. We prefer to delay sampling an actual escape out of a so-constructed web up to the size when further web expansion is no longer justified by the expected speedup. As we will show in the subsequent sections, it is possible to maintain, while expanding the web, a running average for the expected speedup that can be used to decide which next frontier state to add and when to stop growing the web and switch to the random walks. Alternative to this is an algorithm in which random walks are observed for a while and a running measure of the walk compactness is used to detect trapping and make a decision to start building the web. In fact, the two strategies can be combined for a higher efficiency.

Algorithm 1: Numerical solution of ODEs defined on the trapping basin

The first algorithm solves the ME on the sub-set of Markov states in which all of the known states are treated as transient states whereas all the frontier states are treated as absorbing states in Novonty's sense [1]. The underlying ME is

$$\dot{\mathbf{p}}^{T} = \mathbf{p}^{T}\mathbf{Q}$$

where **p** is the vector of time-dependent occupation probabilities and **Q** is the usual transition rate matrix for the continuous time Markov chain[2].

Given a cluster of transient and absorbing states, the solution of this set of ODE allows one to sample an escape out of the cluster into one of the absorbing states as follows. First, the partial sum of occupation probabilities taken at time *t* over all the transient states defines the total (survival) probability *S(t)* that the walker has not escaped the transient sub-set by time *t*. Given this function, one can obtain a random sample of the escape time t_e from the condition $t_e = S^{-1}(\eta)$ where η is a

[2] The usual initial condition is that the system starts in some particular state α at time t=0, that is $p_\alpha(0)= 1$ and $p_\beta(0)=0$ for all $\beta\neq\alpha$. However, other initial conditions can be useful depending on one's specific interest.

random number uniformly distributed on [0,1). Then, given the escape time, one can use another random number to select from among the absorbing states with weights equal to the ratios of the occupation probabilities of the absorbing states $p_\beta(t_e)$ at time t_e, to $1\text{-}\eta$:

$$w_\beta = \frac{p_\beta}{1-\eta}.$$

One obvious problem with this approach is that, to be able to sample the random escape time one needs to have the solution for the escape probability S at all times, from zero to infinity. This is costly and does not bode well for our strategy to build the web incrementally and adaptively.

The problem is partially solved by observing that, rather than sampling an escape time from the full solution for $S(t)$ at all stages of web expansion, it is possible to pre-sample a specific value of S_e at which the escape is to take place in advance, i.e. before actually building the web. This way, in order to sample an escape from the current web one needs to know the solution for the occupation probabilities **p** only for time $t_e = S^{-1}(S_e)$. Because the sequence of web expansion is arbitrary and affects only the numerical efficiency, the web expansion steps can be tailored to maximally extend the escape times for a given pre-sampled escape value S_e (S_e close to 1.0 defines an "early" escape whereas close to 0.0 defines a "late" escape). Furthermore, by focusing attention on a specific escape value, this trick should allow to incrementally solve for t_e each time the web is expanded.

We have experimented with iterative stiff ODE solvers based on the Krylov subspace projections, to solve incrementally for t_e on the expanding web for the case of diffusion on a 2D random landscape. Initial numerical experiments showed considerable promise but we decided, for now, to focus on the alternative acceleration method developed in the next section.

Algorithm 2: Graph reduction and randomization

The idea of using graph reduction was recently proposed by Tregubenko and Wales [10]. They showed how, given a graph of connected Markov states (a Markovian web), it is possible to compute transition probabilities and mean transition times between any two states (or group of states) on the graph using a sequence of incremental graph reductions: at each step one intermediate state is eliminated (deleted) but the transition probabilities between any two remaining states are updated to account for the paths through the just deleted state. Our second algorithm uses this graph reduction while incrementally building the Markov web: once a new transient state is added to the web, it is immediately eliminated and the transition probabilities are updated for the previously existing states and computed for any new states. This way, the only states that remain in the reduced graph are the (renormalized) initial state and all the absorbing states. Eventually all or most of the remaining states become mutually connected.

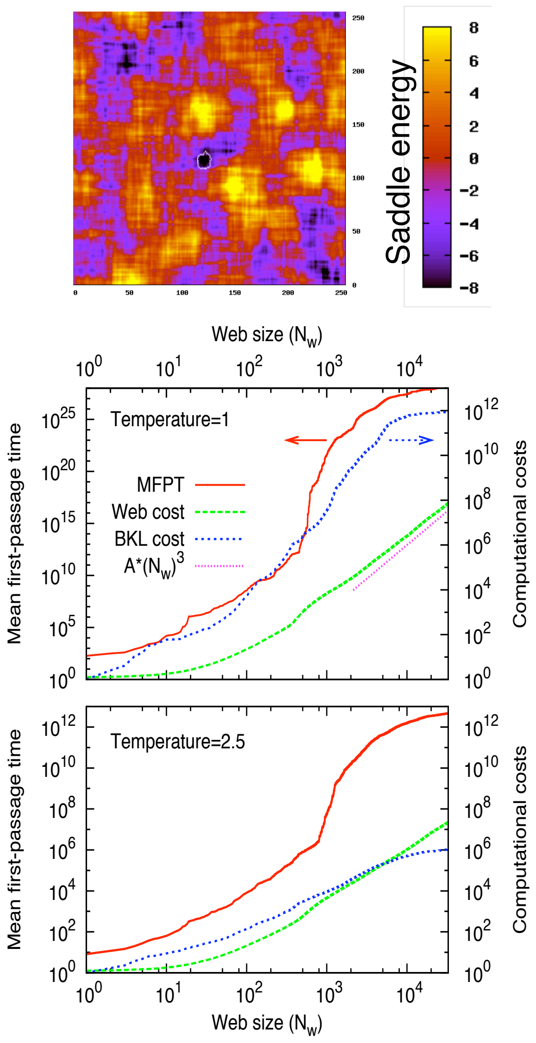

Figure 2: At the top: the 256x256 square lattice of states are connected by transitions with rates defined by the saddle-point energies distributed as shown (the saddle energies are in arbitrary units). At the bottom: the evolution of the mean first-passage time (red) and of the computational costs for the web algorithm (dashed green) and the standard random walks by BKL algorithm (dashed blue) are shown as functions of the expanding web size, up to the web filling the whole square substrate, at two different temperatures, T = 1.0 and T=2.5 (in the units of the saddle energy). At the higher temperature, the web-based simulation becomes less efficient than the random walks at the web size of several thousand states where the green and the blue line intersect. The speedup achieved by the web-based algorithm is the distance between the blue and the green lines and reaches some six orders of magnitude at the web size of about 10^4 states at temperature T=1.0.

Given that transition probabilities to all absorbing states are available at all stages of web expansion, one can decide which of the absorbing states to promote to a transient rank and delete next. This can be done so that to maximize the increment of the mean escape time $\langle t_e \rangle$ that can be easily computed for each candidate absorbing state. However, our most important observation is that, once the escape state is randomly sampled, one can use randomization to sample an escape time t_e from the sub-set of paths conditioned on the selected final escape. The randomization proceeds by first randomly sampling how many times a given transient deleted state was visited on the way to the final escape. Then, given the number of returns for each transient state, another random number is used to sample the total time the walker resided in the given state. The resulting escape time is simply the sum of so-sampled residence times over all the deleted states. We found that such path-reconstruction by randomization can be made efficient if one stores the fractional probabilities computed during the forward process of web reduction. The numerical cost of incremental web-expansion and reduction is defined by the number of deleted states and the number of absorbing states remaining in the web. The path-reconstruction by randomization is still costlier but no worse than $O(N^3)$ in the number of states in the web.

It turns out that the web-reduction algorithm proposed by Wales and Tregubenko is precisely equivalent to Gaussian elimination. This can be effectively utilized by re-using the LU factorization to compute not only the mean escape time, but also a number of the moments of the escape time distribution *S(t).* This information should be sufficient to approximate the unknown full solution for *S(t)* from which to randomly sample the escape time t_e.

This algorithm was used to simulate two test problems. The first one was random walks on a random 2D substrate. The size of the square lattice was 256x256. Although assigned randomly, the transition rates were such that the spectrum of the transition rate matrix **Q** was wide and "bad" in the sense that it contained no gaps. This was done specifically to make the system stiff and difficult to solve using approximate methods based on adiabatic elimination [refs]. The tests shown in Figure 2 demonstrate that the resulting speedup increases with the increasing width of transition rate distribution and depends also on the size of the Markov web relative to the size of the trapping basin. The graphs also show that, depending on the properties of the trapping basin being explored, one can decide when to switch from web expansion to simple random walk and back so that the method is never less efficient than the standard KMC.

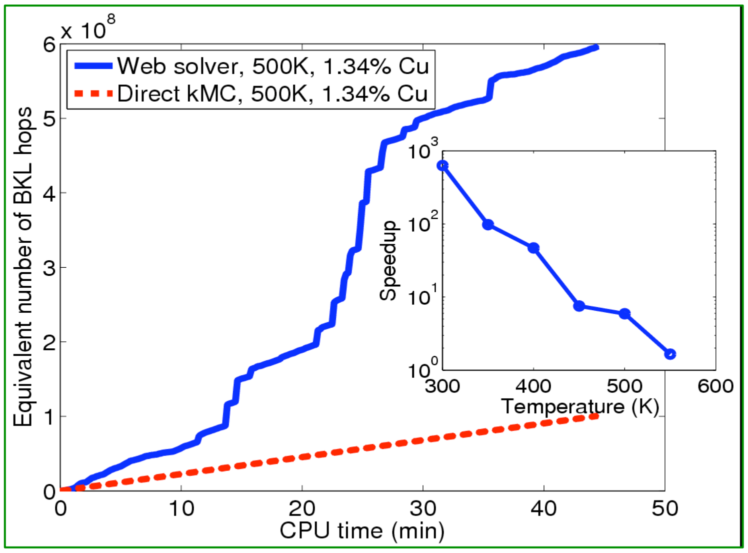

Figure 3: The speedup achieved using the web-based algorithm relative to the standard KMC simulation. The blue curve shows the number of moves it would have taken using the standard algorithm to evolve the ABV model over the same length of time. The red line is for the standard algorithm. The inset shows the speedup as a function of temperature.

The second test case was simulations of the kinetics of 1st order phase transition in a binary alloy under-cooled from above to just under the miscibility line. In our simulations we used the ABV lattice model for Fe-Cu alloy as defined by Soisson [11]. The inefficiency of KMC sampling in this case results from attraction of the vacancy to and its trapping in the clusters of metal B. Depending on B-V binding parameters and temperature, trapping can lead to a change in the mode of growth of B particles, from a classical Oswald ripening to cluster coalescence [12] resulting in a gross modification of the phase separation kinetics. At the same time, the very same trapping renders KMC simulations inefficient, requiring months of continuous number crunching and still only reaching early stages of coarsening [11]. As shown in Figure 3, application of the web-expansion in combination with the web-reduction algorithm described above speeds up KMC simulations considerably, especially in cases where trapping is strong.

Summary

We proposed a new class of Monte Carlo algorithms in which random walks in the space of Markov states are performed whenever the walks are expansive (non-returning) but, whenever trapping is detected (the walk becomes compact), a solution of the Master Equation on the trapping basin is used to help overcome the trapping. The new method is in principle exact and contains internal criteria for deciding whether any trapping is taking place and whether the trapping is serious enough to deserve building the Master Equation for accelerated escape out of the trapping basin. Thus the new method will never underperform the standard KMC algorithms since it switches back to the random walks when no acceleration is needed. At the same time, being exact, the method does not depend on the spectral properties of the resulting transition sub-matrix defined on the trapping basin.

Literature

1. M.A. Novotny, *Phys. Rev. Lett.* **74**, 1-4 (1995).

2. M. Athenes, P. Bellon and G. Martin, *Philos. Mag.* A **76**, 565-585 (1997).

3. D. R. Mason, R. E. Rudd and A. P. Sutton, *Comp. Phys. Comm*. **160**, 140-157 (2004).

4. C. Dew. Van Siclen, *J. Phys.: Cond. Matt.* **19**, 072201 (2007).

5. B. Puchala, M. L. Falk and K. Garikipati, *J. Chem. Phys.* **132**, 134104 (2010).

6. A. B. Bortz, M. H. Kalos and J. L. Lebowitz, *J. Comp. Phys*. **17**, 10-18 (1975).

7. B. Munsky and M. Khammash, *J. Chem. Phys.* **124**, 044104 (2006).

8. G. C. Boulougouris and D. Frenkel, *J. Chem. Theory Comput*. **1**, 389 (2005).

9. G. C. Boulougouris and D. N. Theodorou, *J. Chem. Phys.* **127**, 084903 (2007).

10. S. A. Trygubenko and D. J. Wales, *J. Chem. Phys*. **124**, 234110 (2006).

11. F. Soisson and C.-C. Fu, *Phys. Rev.* B **76**, 214102 (2007).

12. R. Wienkamer, P. Fratzl, H. S. Gupta, O. Penrose and J. L. Lebowitz, *Phase Transitions* **77**, 433-456 (2004).