Noname manuscript No.
(will be inserted by the editor)

Efficient Simulation of Markov Chains using Segmentation

Tim Brereton · Ole Stenzel
Björn Baumeier · Denis Andrienko
Volker Schmidt · Dirk Kroese

Received: date / Accepted: date

Abstract A methodology is proposed that is suitable for efficient simulation of continuous-time Markov chains that are nearly-completely decomposable. For such Markov chains the effort to adequately explore the state space via Crude Monte Carlo (CMC) simulation can be extremely large. The purpose of this paper is to provide a fast alternative to the standard CMC algorithm, which we call Aggregate Monte Carlo (AMC). The idea of the AMC algorithm is to reduce the jumping back and forth of the Markov chain in small subregions of the state space. We accomplish this by aggregating such problem regions into single states. We discuss two methods to identify collections of states where the Markov chain may become 'trapped': the stochastic watershed segmentation from image analysis, and a graph-theoretic decomposition method. As a motivating application, we consider the problem of estimating the charge carrier mobility of disordered organic semiconductors, which contain low-energy regions in which the charge carrier can quickly become stuck. It is shown that the AMC estimator for the charge carrier mobility reduces computational costs by several orders of magnitude compared to the CMC estimator.

Keywords Markov chain · nearly-completely decomposable · Monte Carlo · segmentation · watershed · graph-theoretic decomposition · electron transport · mobility · organic semiconductor

1 Introduction

Dynamic processes in complex physical systems often take place on multiple time scales, rendering a direct full evolution of the system, e.g., by solving a set of explicit deterministic equations, impossible. However, in infrequent-event systems of this type, i.e., systems with well-defined states and only occasional transitions between them, the dynamics can be treated in terms of the state-to-state

T. Brereton · D. P. Kroese
School of Mathematics and Physics, The University of Queensland,
Brisbane 4072, AUSTRALIA
E-mail: tim.brereton@uqconnect.edu.au
E-mail: kroese@maths.uq.edu.au

B. Baumeier · D. Andrienko
Max Planck Institute for Polymer Research,
55128 Mainz, GERMANY
E-mail: baumeier@mpip-mainz.mpg.de
E-mail: denis.andrienko@mpip-mainz.mpg.de

O. Stenzel · V. Schmidt
Institute of Stochastics, Ulm University,
89069 Ulm, GERMANY
E-mail: ole.stenzel@uni-ulm.de
E-mail: volker.schmidt@uni-ulm.de

transitions, described by a matrix of transition rates. An example of such a process is charge transport in disordered organic semiconductors. The dynamics of charges can be modeled by means of a continuous-time Markov chain (CTMC) on a graph, where the graph represents the transporting medium and the state of the process represents the positions of the charge carriers. The transition (or *hopping*) rates of the CTMC between adjacent vertices of the graph can be determined using charge transfer theories from the electronic and quantum mechanical properties of the semiconductor (see, for example, Rühle et al (2011)). One of the main characteristics to be measured is the *charge carrier mobility* $\mu = v/|E|$, which (in the limit of a single mobile charge) depends on the *drift velocity*, v, of the charge carrier as it passes through the random medium in an external electric field E. The drift velocity v is the average velocity of the charge carrier in the direction of the electric field. This quantity can be estimated via Monte Carlo simulation of the CTMC — called *kinetic Monte Carlo* or *dynamic Monte Carlo* in the physics literature (see, e.g., Pasveer et al (2005); Jansen (2012)) Since the charge carrier mobility influences the performance of a material in technological applications, e.g., the efficiency of organic solar cells, simulation models for charge transport are a key ingredient of intensive efforts in in-silico design of high-efficiency organic semiconductors, see Baumeier et al (2012).

The simulation approach introduced by Schönherr et al (1981) and Bässler (1993), which we call *crude Monte Carlo* (CMC), has become a well-established method (see, e.g., Tessler et al (2009); van der Holst et al (2011)). A major problem, however, is that for a large variety of materials the energy landscape associated with the random medium contains regions in which the charge carrier quickly becomes stuck. A consequence of this is that the estimation of drift velocity via CMC is not only very time-consuming (often to the extent of being practically infeasible) but also leads to unreliable estimators that can have a bias of several orders of magnitude. In addition, processes in organic electronics take place on multiple time and spatial scales. Therefore, it is essential to have computationally fast methods that analyze large system sizes over a long physical time without losing too much information about finer scale behavior.

The purpose of this paper is to provide a fast alternative to the standard CMC algorithm. In Brereton et al (2012), we introduced a new approach to the estimation of charge carrier mobility in a simple 1-dimensional setting. In the present paper, we focus on a 3-dimensional system based on microscopic simulations as presented in Rühle et al (2011). We introduce a novel approach to the problem of deep energy traps, that improves the efficiency of the estimator by coarsening the state space. We use two different algorithms — a stochastic watershed algorithm and a graph-based segmentation algorithm, respectively — to identify problem regions. We then construct a coarsened state-space model, under which the problem regions can be traversed in a single step of the simulation. In the literature, this type of problem has been discussed in the context of, e.g., nearly-completely decomposable Markov processes (Simon and Ando (1961); Courtois (1977); Conway and Georganas (1982)) and multiple time-scale Markov processes (Tse et al (1995); Evans (1996)). It is shown that the AMC algorithm, presented in this paper, reduces computational costs for estimating charge carrier mobility by several orders of magnitude. A similar strategy of coarsening the state space is used in Somoza and Ortuño (2005) to study the relaxation of Coulomb glasses at low temperature. However, the latter approach only considers pairs of problem states, and does not segment the state space prior to the simulation.

The paper is organized as follows. In Section 2 we briefly explain how electron transport in disordered organic semiconductors can be modelled by a CTMC on a geometric graph of charge hopping sites. Then, in Section 3 the CMC estimator of the average charge carrier mobility is described. In Section 4, the idea of the aggregate Monte Carlo (AMC) algorithm is introduced, which is based on coarsening the state space of the CTMC. Two different methods of aggregation are discussed: the stochastic watershed segmentation from image analysis, and a graph-theoretic decomposition method, see Section 5. Numerical results are presented in Section 6 which show the enormous advantage of the AMC estimators in comparison to the standard CMC estimator of the average charge carrier mobility. Conclusions are given in Section 7.

2 Model

In this paper, we consider the problem of estimating the charge mobility of a weighted spatial graph of 4096 tris(8-hydroxyquinolinato)aluminium (Alq3) molecules; see Figure 1. Alq3 is a green light emitter employed in early realizations of organic light emitting diodes. By a weighted spatial graph, we mean a graph $G = (V, C, Q)$ where $V = \{V_1, \ldots, V_n\}$ describes the set of vertices in a bounded observation window $W \subset \mathbb{R}^3$ with positive volume $|W|$, $C = \{(V_{i_1}, V_{j_1}), \ldots, (V_{i_m}, V_{j_m})\}$ the edges, and $Q = \{q_{i_1 j_1}, \ldots, q_{i_m j_m}\}$ the edge weights. For more information on graphs, we refer to Diestel (2005). We assume that the graph is completely connected and that cyclic boundary conditions are applied. More precisely, let $W = [0, b_x] \times [0, b_y] \times [0, b_z]$ for some $b_x, b_y, b_z > 0$, then the coordinate-wise distance vector $\mathbf{d}_{ij}$ between two vertices $V_i = (x_i, y_i, z_i)$ and $V_j = (x_j, y_j, z_j)$ is given by

$$\mathbf{d}_{ij}^{\intercal} = (d_x, d_y, d_z)\,, \tag{1}$$

where

$$d_x = \begin{cases} x_j - x_i & \text{if } |x_j - x_i| < \frac{b_x}{2} \\ b_x + (x_j - x_i) & \text{if } |x_j - x_i| > \frac{b_x}{2} \text{ and } x_i > x_j \\ -b_x + (x_j - x_i) & \text{if } |x_j - x_i| > \frac{b_x}{2} \text{ and } x_i < x_j \end{cases}$$

and d_y, d_z follow analogously.

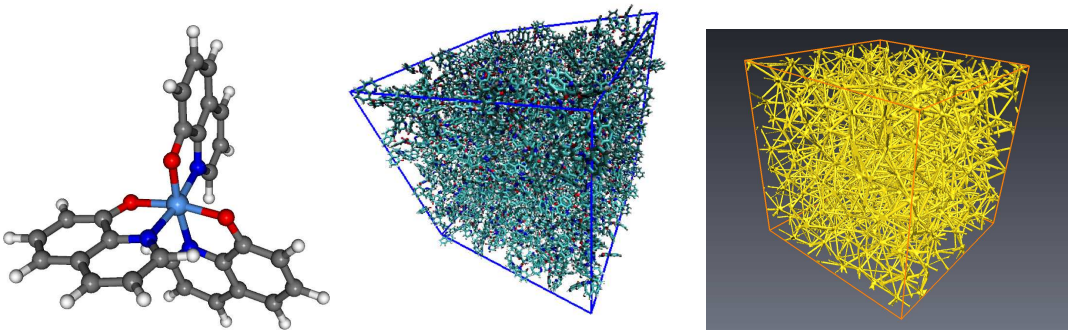

Fig. 1 Left: Alq3 molecule. Center: morphology of 4096 Alq3 molecules. Right: corresponding graph (cut-out).

The graph on the right in Figure 1 is obtained by a *microscopic approach*, see Rühle et al (2011), where a large-scale morphology of Alq3 molecules is simulated by atomistic molecular dynamics including cyclic boundary conditions (Figure 1 (center)). The barycenters of the molecules form the vertices of the graph and two vertices are connected via an edge if the distance between two molecules is lower than a threshold, here 0.8 *nm*. In addition, charge transfer rates are determined using first principles calculations. These charge transfer rates determine the dynamics of the charge carriers. More precisely, charge dynamics (i.e., the random movement of the charge carrier along the edges of the graph) can be described by a CTMC: if a charge carrier is at vertex V_i, it hops to neighboring (i.e., connected) vertex V_j with rate

$$q_{ij} = \frac{2\pi}{\hbar} \frac{J_{ij}^2}{\sqrt{4\pi\lambda_{ij} k_{\mathrm{B}} T}} \exp\left[-\frac{(\Delta E_{ij} - \lambda_{ij})^2}{4\lambda_{ij} k_{\mathrm{B}} T}\right], \quad i \neq j\,, \tag{2}$$

where $\hbar$ is the reduced Planck constant, λ_{ij} the (constant) reorganization energy, k_B the Boltzmann constant, T the temperature, J_{ij} the transfer integral and ΔE_{ij} the difference in energy between the two vertices. Throughout the manuscript, the values for the energies are given in eV. Furthermore, we define

$$q_i = \sum_{j \neq i} q_{ij}\ .$$

Thus, the random sojourn time τ_i in state i is $\tau_i \sim \mathsf{Exp}(q_i)$. Given the spatial positions of the vertices V_i, V_j and the electric field E, the energy difference is given by $\Delta E_{ij} = qE^\intercal \mathbf{d}_{ij} + (\eta_i - \eta_j)$, where q is the charge of an electron and $\mathbf{d}_{ij}$ is the coordinate-wise distance vector between the vertices V_i and V_j. Note that the electric field E can be presented by its magnitude $|E|$ and its direction vector $\mathbf{e}^\intercal = (e_1, e_2, e_3)$, i.e., $E = |E|\mathbf{e}$. The main driving force for the dynamics of charge carriers hopping through the network of molecules, are their site energies η_i, i.e., the energy that is associated with the corresponding vertex V_i. In amorphous Alq3, these energies follow a Gaussian distribution, and they are spatially, positively correlated. In organic electronic systems, one characteristic that influences efficiency is the speed at which charges traverse the network of molecules for a given electric field. More precisely, the characteristic commonly considered is the *charge carrier mobility* defined as $\mu = v/|E|$, where v is the drift velocity of the charge carrier. Thus, an accurate estimate of v is highly important.

2.1 General context

We consider a CTMC $\{M_t, t \geq 0\}$, whose state space is the set of vertices of the spatial graph, which will be represented by the set of integers $A = \{1, \ldots, n\}$, where the transition rates are given by the edge weights and the edges indicate transitions with non-zero transition rate. Since the state space is finite and irreducible by the complete connectivity of the graph, $\{M_t, t \geq 0\}$ is ergodic.

We are interested in the limit behavior of such a system. More precisely, let $\boldsymbol{\pi} = (\pi_1, \ldots, \pi_n)$ be the limiting distribution, i.e., $\pi_i = \lim_{t\to\infty} P(M_t = i)$. Let M_∞ be a random variable with distribution $\boldsymbol{\pi}$ and $f : A \to \mathbb{R}^d$ an arbitrary function, then we are interested in the expectation $\mathbb{E}f(M_\infty)$. Due to ergodicity, we know that $\mathbb{E}f(M_\infty) = \lim_{T\to\infty} \frac{1}{T}\int_0^t f(M_t)\mathrm{d}t$ with probability 1. Thus, $\mathbb{E}f(M_\infty)$ can be estimated by simulating the CTMC $\{M_t, t \geq 0\}$ for a limited, but large time horizon T and then take $\frac{1}{T}\int_0^T f(M_t)\mathrm{d}t = \frac{1}{T}\sum_{i=1}^{N_T} f(\widetilde{M}_i)\tau_{\widetilde{M}_i}$ as an estimate, where $\left\{\widetilde{M}_1, \widetilde{M}_2, \ldots\right\}$ is the embedded Markov chain whose transition matrix will be denoted by $J = (p_{ij})$, N_T is the random number of transitions up to time T and τ_i the random sojourn time in state i.

2.2 Example

Given that the CTMC $\{M_t, t \geq 0\}$ has the properties of a nearly-completely decomposable Markov chain, the time T or the number of steps that have to be simulated is extremely large, as the state space is otherwise not adequately explored. As an example, consider a CTMC, where $n = 6$ and the embedded Markov chain $\left\{\widetilde{M}_1, \widetilde{M}_2, \ldots\right\}$ has the following transition matrix J of transition probabilities:

$$J = \begin{pmatrix} 0 & 0.5 & 0 & 0 & 0 & 0.5 \\ 10^{-7} & 0 & 1-10^{-7} & 0 & 0 & 0 \\ 0 & 1-10^{-7} & 0 & 10^{-7} & 0 & 0 \\ 0 & 0 & 0.5 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 10^{-7} & 0 & 1-10^{-7} \\ 10^{-7} & 0 & 0 & 0 & 1-10^{-7} & 0 \end{pmatrix}.$$

The above described system has two pairs of states that build loops with high probability: states $2, 3$ and $5, 6$. If the simulation of the CTMC $\{M_t, t \geq 0\}$ is terminated too early, the result of the estimate $\frac{1}{T}\sum_{i=1}^{N_T} f(\widetilde{M}_i)\tau_{\widetilde{M}_i}$ will be highly dependendent on where the simulation started.

The idea is therefore to coarsen the state space, in this case, to merge states $2, 3$ and $5, 6$ into two *super-states*, $\{2, 3\}$ and $\{5, 6\}$. The CTMC is then simulated on the state space of super-states.

2.3 Drift velocity

In the present paper, we apply this idea of *coarse graining* to the CTMC describing charge transport in disordered organic semiconductors, as explained at the beginning of Section 2. In particular, we show how the idea of coarse graining can be used for the improvement of Monte Carlo estimators of the drift velocity v of the CTMC $\{M_t, t \geq 0\}$, defined by

$$v = \sum_{i \in A} \pi_i \left(\sum_{j \in A} q_{ij} \mathbf{e}^\intercal \mathbf{d}_{ij} \right) = \mathbb{E} f(M_\infty), \tag{3}$$

with $f(i) = \sum_{j \in A} q_{ij} \mathbf{e}^\intercal \mathbf{d}_{ij}$, where q_{ij} a is the transition rate from state i to state j given in (2).

3 Crude Monte Carlo

3.1 Description of the CMC algorithm

The CMC estimator of the average drift velocity v is $\widehat{v} = d_T/T$, where d_T is the distance traveled by the charge carrier by time T. This quantity is computed in a straightforward manner by simulating the path of the charge carrier through the random medium, tracking the distance travelled in the direction of the electric field E.

Algorithm 3.1 (Crude Monte Carlo estimation of drift velocity)

For a fixed $T > 0$,

1. Set $d = 0$, $\ell = 0$ and $t = 0$, where d is the distance of the charge carrier travelled in the direction of the electric field E, ℓ denotes the number of transitions and t denotes the total physical time of the movement of the charge carrier.
2. Draw $\widetilde{M}_0$ uniformly from $\{1, \ldots, n\}$.
3. Given $\widetilde{M}_\ell = i$, choose the next state $\widetilde{M}_{\ell+1}$ according the probabilities given by the i-th row vector of J.
4. Draw $\tau_\ell \sim \mathsf{Exp}(q_i)$.
5. Set $t = t + \tau_\ell$.
6. Set $\ell = \ell + 1$.
7. If $t > T$, break. Otherwise, set $d = d + \left(\mathbf{e}^\intercal \mathbf{d}_{\widetilde{M}_\ell, \widetilde{M}_{\ell+1}} \right)$, where $\mathbf{d}_{ij}$ is defined in (1), and repeat from Step 3.
8. Return $\hat{v} = d/t$.

In this paper we are concerned with minimizing the computational effort required to arrive at an accurate estimate of the drift velocity v defined in (3). From this perspective, we consider computational time, rather than physical time, as measured by the variable t.

We identify the computational time required by the algorithm with the number of iterations N the algorithm requires in order to return an accurate estimate (this is measured by ℓ in Algorithm 3.1). Thus, in analyzing the algorithms in this paper, as opposed to using them in practice, we will terminate the algorithm when $\ell > N$, rather than when $t > T$.

3.2 Performance of CMC

Figure 2 shows the average value of the estimator (over many independent copies) as a function of the computational effort. From this, it is immediately apparent that the CMC algorithm performs poorly. Roughly 10^7 steps of the algorithm are required to return an accurate estimate of charge mobility, μ, (the normalized drift velocity).

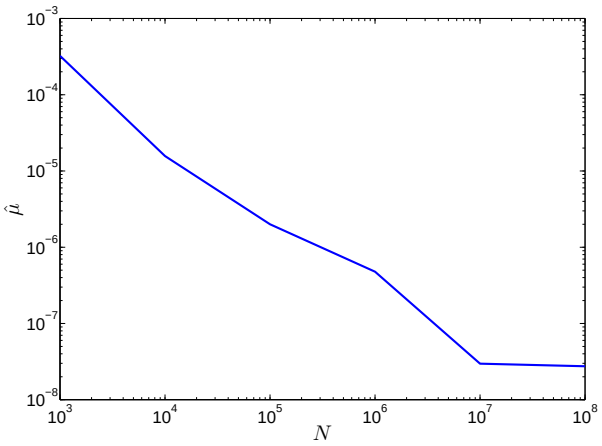

Fig. 2 Averaged CMC estimates $\hat{\mu}$ of μ vs. the number of steps, N, of the algorithm

The reason that the algorithm performs so badly is that the charge is randomly injected on a site and its initial energy is (on average) higher than the average transport energy (average energy of a charge being in equilibrium according to the diffusion process). Hence at the initial stage a non-equilibrium hopping of a charge takes place during which the mobility of such a 'hot' carrier (i.e., with too high energy) is initially higher than the equilibrium mobility. Since the energy surface contains many regions of relatively low energy, the charge carrier becomes trapped in these regions and makes a very large number of small jumps in a small amount of time. This, on the one hand, slows down getting into equilibrium (overestimating the mobility) but can also lead to pathological situations when the charge hops between, e.g. two very deep energetic traps, which gives smaller than expected average mobilities.

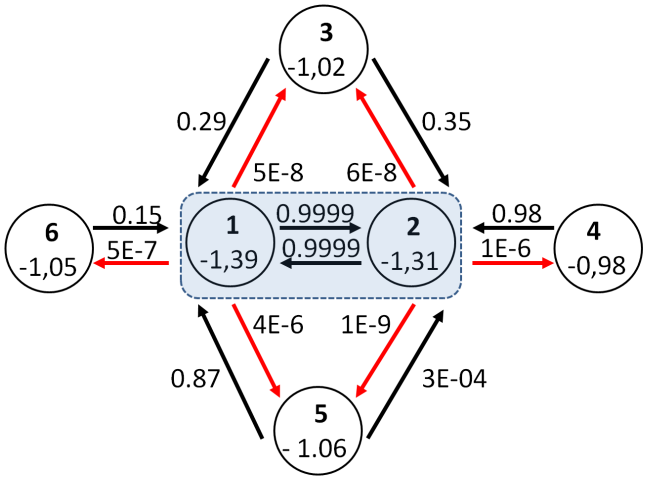

Fig. 3 Cut-out of the graph with six states. The state numbers are indicated by bold integers. The corresponding energies are given below the state numbers (e.g., state 1 has energy -1.39). Hopping probabilities p_{ij} are shown by arrows. The electric field E is set to zero. For clarity only the main transitions to and from states 1 and 2 are displayed. Note that since not all possible transitions from and to states 1 and 2 are displayed, the transition probabilities do not sum up to 1.

More precisely, we consider a pair of sites which have a very low energy; see Figure 3. Here state 1 has the third lowest energy of the 4096 sites and state 2 the seventh lowest energy. These two states

are in a valley according to the energy landscape, i.e., the surrounding sites have a significantly larger energy, which results in extremely low probabilities of escaping the pair $\{1, 2\}$. If the charge carrier is e.g., in state 4, it will hop with probability 0.98 into state 2, where it is 'trapped' in $\{1, 2\}$ for a long time. If the CMC algorithm terminates before the charge carrier has left this trap, this will clearly bias the estimate of the drift velocity v and, therefore, the charge carrier mobility $\mu = v/|E|$. Let us analyze more clearly why the energy landscape is the main driving force for the dynamics of the charges. Consider the exponential term in (2) of the hopping rate q_{ij}, i.e.,

$$\exp\left[-\frac{(\Delta E_{ij} - \lambda_{ij})^2}{4\lambda_{ij} k_{\mathrm{B}} T}\right] = \exp\left[-43\,(\Delta E_{ij} - 0.23)^2\right], \tag{4}$$

given $\lambda_{ij} = 0.23$ and $k_B T = 0.025$. Consider the transition from state 4 to state 2. The energy difference (assuming no electric field) is $\Delta E_{42} = +0.33$ which yields a value of the exponential term considered in (4) of 0.66. The transition from state 2 to state 4 gives an exponential term of 1.9×10^{-6}. Thus, the hopping rates are highly dependent on the energy landscape. Let us now assume that the energy of site 4 is decreased from -0.98 to -1.39, which is the same energy as state 1. Then the transition probability from site 2 to site 4 increases to 0.23, whereas the transition probability from site 4 to site 2 decreases to 0.84.

4 Aggregate Monte Carlo

The idea of the aggregate Monte Carlo (AMC) algorithm is to coarsen the state space in such a way that we avoid spending too much simulation effort jumping back and forth in the low-energy regions. We accomplish this by aggregating problem regions into single states. We replace the state space of our original process with a smaller state space, labeled $\tilde{1}, \ldots, \tilde{n}$, where each state in the new state space consists of at least one — but often more — states of the original state space $A = \{1, \ldots, n\}$. These new 'super-states' are chosen such that each problem region in the original state space is contained within a single state of the coarsened state space $\widetilde{A} = \{\tilde{1}, \ldots, \tilde{n}\}$.

Although the continuous-time process describing the position of the charge carrier on the new state space is no longer a CTMC, it is still possible to describe the jumps of the charge carrier by a discrete-time Markov chain (DTMC). This is done by tracking the point at which the charge carrier enters each super-state. The associated process corresponds to a DTMC with a state space consisting of all of the states on the boundaries of the super-states (i.e., states with non-zero transition probabilities out of super-states).

Because the new continuous-time process is not Markovian, the times taken to cross super-states are no longer exponentially distributed random variables but rather have phase-type distributions. Generating phase-type random variables is computationally very expensive, so we replace the times of the sojourns in the super-states with their expected values. Thus, rather than simulating a CTMC on the original state space, we simulate a DTMC on a coarsened state space, with the time taken by the chain growing by a deterministic value at each step. This process can also be viewed as a semi-Markov process, with deterministic, state-dependent sojourn times.

In brief, the aggregation process consists of the following steps:

1. Find a segmentation of the state space $A = \{1, \ldots, n\}$ into states $\tilde{1}, \ldots, \tilde{n}$ such that problem regions are contained in super-states.
2. Consider a DTMC on the state space which consists of states on the boundaries of super-states. Calculate the transition probabilities for this process.
3. Calculate the expected sojourn times in each super-state, conditional on the points of entrance into, and exit from, the super-state.
4. Simulate the DTMC. Track the time taken by adding the appropriate expected sojourn time at each step. Track the distance by adding the distance travelled in the direction E by the charge carrier between the two states.

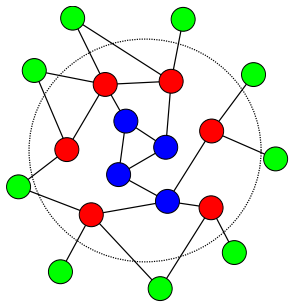

Fig. 4 The states within the dotted circle constitute a super-state. The lines indicate non-zero one-step transition probabilities between states. The red states are outer states, the blue states are inner states and the green states are adjacent states.

4.1 Calculating transition probabilities and expected sojourn times

Given a segmentation of the state space, we need to determine the transition probabilities of the charge carrier on the coarsened state space and the expected times taken to carry out these jumps.

For each super-state, $i \in \{\tilde{1}, \ldots, \tilde{n}\}$, there are $A_i > 0$ states outside the super-state with non-zero one-step transition probabilities into the super-state. We will call these the 'adjoining' states, labelled $a_1^{(i)}, \ldots, a_{A_i}^{(i)}$. Within the super-state there are $O_i > 0$ states with non-zero one-step transition probabilities into one or more of the adjoining states. We will call these the 'outer' states, labelled $o_1^{(i)}, \ldots, o_{O_i}^{(i)}$, see Figure 4. When the charge carrier enters a super-state, it can only do so into one of the outer states. When it leaves the super-state it can only do so into one of the adjoining states. These adjoining states are outer states of other super-states. Thus, from a practical point of view, we consider a jump chain on a state space which consists only of outer states of the super-states. This state space is $o_1^{(\tilde{1})}, \cdots, o_{O_{\tilde{1}}}^{(\tilde{1})} \ldots, o_1^{(i)}, \cdots, o_{O_i}^{(i)}, \ldots, o_1^{(\tilde{n})}, \cdots, o_{O_{\tilde{n}}}^{(\tilde{n})}$. For convenience, we will relabel these states as $\bar{1}, \ldots, \bar{n}$. We denote the transition matrix of the jump chain by $\bar{J}$.

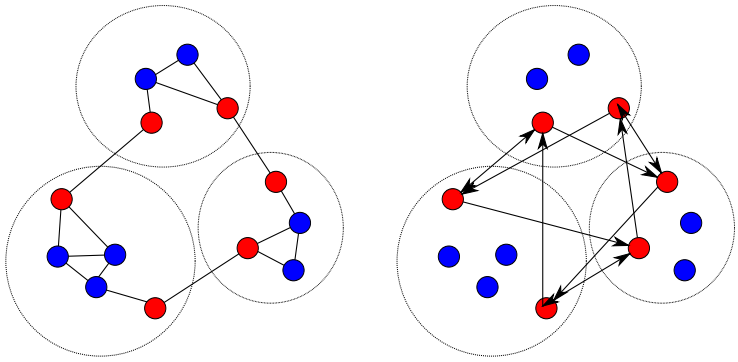

Fig. 5 Left: The initial jump chain, with the segmentation indicated by the dotted circles. Right: The new jump chain, after the state space has been coarsened. In the new chain, $\bar{J}_{kj} > 0$ does not imply $\bar{J}_{jk} > 0$. The arrows indicate the directions of non-zero transition probabilities.

We track the charge carrier's progress over these outer states. When a charge carrier jumps from an outer state, it passes through the super-state and emerges at one of the adjoining states, see Figure 5. Thus, we need to calculate the probability that a particle in state j passes through its super-state and exits to a given adjoining state, k. To calculate this probability, we consider a CTMC with state space that consists of all of the states within the super-state i (of which state j is an outer state), with the adjoining states, $a_1^{(i)}, \ldots, a_{A_i}^{(i)}$ acting as absorbing barriers. The generator

Q_i and transition matrix J_i of this CTMC can be written in the following forms

$$Q_i = \begin{pmatrix} 0 & 0 \\ S_{0_i} & S_i \end{pmatrix} \quad \text{and} \quad J_i = \begin{pmatrix} I & 0 \\ R_i & T_i \end{pmatrix},$$

respectively. The matrix $P_i = \left(p_{jk}^{(i)}\right)$ of absorption probabilities from transient state j into absorbing state k is given by $P_i = (I - T_i)^{-1} R_i$. To find the expected times until absorption, we note that the matrix of the densities of the absorption times, $(f_{jk}(t))$, from transient state j into absorbing state k is given by $\mathrm{e}^{tS_i} S_{0_i}$, where e^{tS_i} is a matrix exponential. If we set $f_{jk}^{\tau}(t) = f_{jk}(t)/p_{jk}^{(i)}$, we obtain the densities of the conditional absorption times. The matrix of expected absorption times, $(\mathbb{E}t_{jk})$, is given by $\left(S_i^2\right)^{-1} S_{0_i}$. Thus, the expected conditional absorption times are given by $\tau_{jk} = \mathbb{E}t_{jk}/p_{jk}^{(i)}$.

4.2 Description of the AMC algorithm

The AMC estimator $\hat{v}$ of the average drift velocity v defined in (3) is computed in the following way.

Algorithm 4.1 (Aggregate Monte Carlo estimation of drift velocity)

For a fixed $T > 0$,

1. Set $d = 0$, $\ell = 0$ and $t = 0$.
2. Draw $\widetilde{M}_0$ uniformly from $\{\bar{1}, \ldots, \bar{n}\}$.
3. Given $\widetilde{M}_\ell = i$, choose the next state $\widetilde{M}_{\ell+1}$ according the probabilities given by the i-th row vector of $\bar{J}$.
4. Set $t = t + \tau_{\widetilde{M}_\ell, \widetilde{M}_{\ell+1}}$.
5. Set $\ell = \ell + 1$.
6. If $t > T$, break. Otherwise, set $d = d + (\mathbf{e}^{\intercal} \mathbf{d}_{\widetilde{M}_\ell, \widetilde{M}_{\ell+1}})$, where $\mathbf{d}_{ij}$ is defined in (1), and repeat from step 3.
7. Return $\hat{v} = d/t$.

5 Aggregation algorithms

The idea of aggregation is to reduce the computational time spent jumping back and forth in low-energy regions. This is accomplished by aggregating collections of problem states into single states. There are a number of different ways to identify collections of states in which the charge carrier may become 'trapped'. One way to identify these problem states is by identifying the corresponding low-energy regions of the energy surface. The stochastic watershed algorithm is one method of identifying such low-energy regions. An alternative approach is to look at the (embedded) jump process associated with the CTMC and try to identify collections of states which have high transition probabilities between one another compared to those they have with the remainder of the state space. The second approach, based on methods developed in the theory of almost decomposable Markov chains, is such a strategy.

In addition to requiring that the aggregation scheme chooses super-states such that every problem region is contained in a super-state, we also require that the segmentation is not too coarse. By this, we mean that the coarsened state space still contains a large number of states and that the number of states in each super-state is not too large. There are several reasons why a fine segmentation is desirable.

1. We often wish to calculate certain path dependent properties at the same time that we calculate charge carrier mobility. A fine segmentation retains a lot of information about the paths taken by the charge carrier.

2. There are less numerical issues in finding the absorption probabilities and expected sojourn times for small super-states.
3. The simulation is less computationally expensive if each super-state has only a small number of adjoining states.

Note that the fineness of the segmentation is largely determined by the parameters in the aggregation algorithms.

5.1 The stochastic watershed approach

The main driving force for the dynamics of the charge carrier is the difference in site energies, see (2). In particular, the charge carrier tends to become trapped in regions with low energy. This is also due to the spatial correlation of the site energies, which can be seen in Figure 6 (right), see also Rühle et al (2011). The charge carrier gets trapped in local energy minima since charge hopping rates to sites with higher energy are very small. Thus, the key idea for this segmentation is to identify these low-energy regions and collapse all sites in these regions into a super-state, see Section 4. In order to identify low energy regions, the sites are transformed on a 3D lattice and their energy values are scaled to gray values ranging from 0 to 255, where 0 (black) indicates the lowest energy observed in the system and 255 (white) the highest energy. The gray values for the remaining voxels (the hopping sites of the system only cover a fraction of the voxel grid) are gained using a Gaussian kernel with cyclic boundary conditions, see Figure 6. The details are explained in Section 5.1.1 below. Having constructed a 3D energy landscape on a voxel grid, the problem of identifying low-energy regions corresponds to the problem of image segmentation in image analysis.

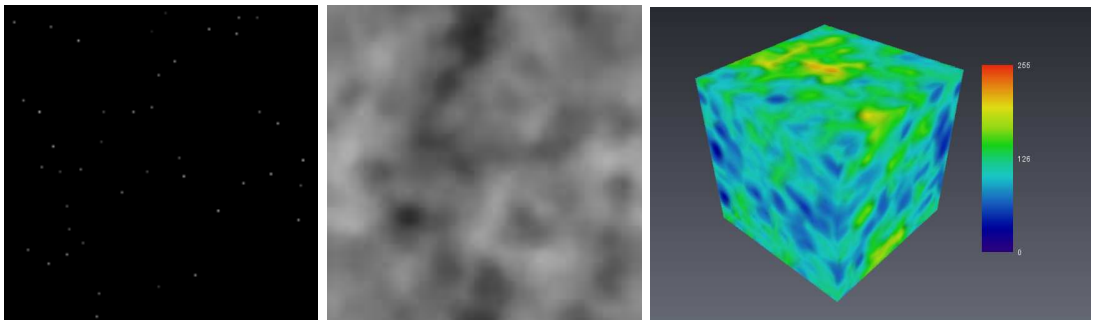

Fig. 6 Left: 2D slice of hopping sites transformed on 3D voxel grid, where gray values indicate energy (0 indicates the lowest energy of the system and 255 the highest possible energy). Center: Corresponding 2D slice of 3D image gained by Gaussian interpolation. Right: 3D visualization of energy landscape.

One of the most applied algorithms for image segmentation is the watershed algorithm. Its key idea was introduced by Beucher and Lantuéjoul (1979). The version of this algorithm commonly used nowadays was developed by Beucher and Meyer (1993). Its basic idea is to consider the image as a topographic relief. This topographic relief is then flooded with water starting at the local minima. If water from different sources (local minima) merge at a certain point, a watershed marker is set. The set of watershed markers segment the image into disjoint regions, so-called basins; see Figure 7 for a schematic example in 1D. The use of watershed segmentation to identify the super-states has been successfully applied to a similar problem in 1D, see Brereton et al (2012).

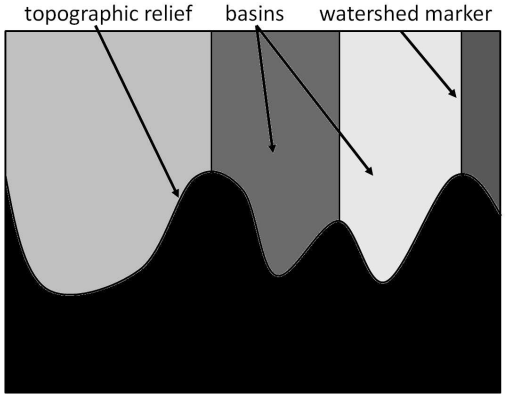

Fig. 7 Watershed transformation.

The idea of the watershed approach corresponds nicely to the goal of identifying low-energy regions: Starting from local minima (i.e., low-energy voxels), the watershed segmentation separates regions of low energy.

5.1.1 Transformation of site energies to gray scale image

We transform all hopping sites on a voxel grid having 137^3 voxels, which is a reasonably high resolution, see Figure 6 (left).

The energy of the hopping sites is suitably rescaled linearly to gray values ranging from 0 to 255. Transforming our prototype system of 4096 molecules on a 137^3 voxel grid corresponds to a volume fraction of 1.6%. The gray values of the remaining voxels (98.4%) are determined by Gaussian interpolation: Let $\tilde{V} = \left\{\tilde{V}_1, \ldots, \tilde{V}_n\right\}$ be the set of vertices of the graph scaled on the voxel grid with energies $\eta_1, \ldots, \eta_n$ (given in Euclidean values, i.e., not yet scaled to gray values). Then the energy $\eta_{(x,y,z)}$ of a voxel $(x, y, z) \notin \tilde{V}$ is given by the weighted sum

$$\eta_{(x,y,z)} = G^{-1} \sum_{i=1}^{n} \eta_i \, \varphi(|(x, y, z) - \tilde{V}_i|; 0, \sigma^2) \,,$$

where $\varphi(\cdot; 0, \sigma^2)$ is a Gaussian density with mean 0 and variance σ^2, and G is a normalizing constant, i.e. $G = \sum_{i=1}^{n} \varphi(|(x, y, z) - \tilde{V}_i|; 0, \sigma^2)$. The variance σ^2 is estimated by likelihood cross validation, see e.g. Silverman (1986). The energies of all voxels are then scaled linearly to gray scale values ranging from 0 to 255, see Figure 6 (center and right). Note that for all operations cyclic boundary conditions are applied.

5.1.2 Stochastic watershed segmentation

A limitation of the standard watershed transformation is that it often yields oversegmentation, i.e., the segmentation into basins (here: super-states of the jump process to be constructed) is too fine. We therefore use the stochastic watershed transformation proposed by Angulo and Jeulin (2007), see also Faessel and Jeulin (2010). The idea is to use the points of a homogeneous Poisson process with some intensity λ, discretized on the voxel grid, as starting points of flooding instead of the local minima. The outcome is a random set of watershed markers separating the (random) basins. The procedure is repeated m times (here: $m = 250$) and the relative frequencies of markers on the discrete lattice are computed, see Figure 8 (left).

In 1D or 2D, it would be sufficient to apply a binarization in order to obtain a segmentation given the result of the stochastic watershed. In 3D, however, watershed lines obtained by binarization, may not yield closed basins. Therefore, we use the following approach: First, a lowpass filter is applied

to the result of the stochastic watershed, i.e., each watershed line whose intensity is lower than a certain threshold ξ, is set to 0. On this modified result of the stochastic watershed, a standard watershed segmentation is applied. Figure 8 (center, right) shows that the obtained segmentation result is reasonable. In particular, the watershed segmentation described in this section does indeed separate regions of low energy, see Figure 8 (right).

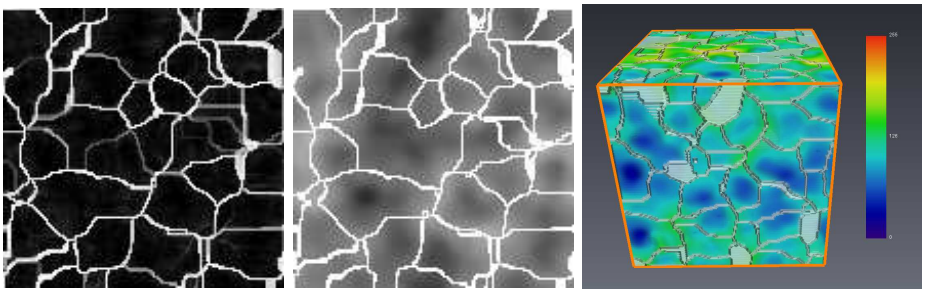

Fig. 8 Left: 2D slice of watershed intensities. Center: A 2D slice of final watershed segmentation superimposed on the corresponding 2D slice of energy landscape. Right: 3D visualization of stochastic watershed segmentation with corresponding energy landscape.

The choice of the intensity, λ, and the threshold, ξ, influence the result of the segmentation. Generally, λ is chosen reasonably high (here $\lambda = 10^{-3}$). Given a realization of the watershed intensities, the value for ξ is chosen manually to adjust the number and size of the basins, where a higher threshold ξ leads to a smaller number of basins that are larger.

The quality of the segmentation based on the stochastic watershed can be analyzed in the following way. To evaluate the goodness-of-fit, we consider all edges (V_i, V_j) such that the transition probabilities p_{ij} and p_{ji} from V_i to V_j and vice versa are larger than a threshold ν, i.e., $\min\{p_{ij}, p_{ji}\} > \nu$. As we want to use the basins (i.e., super-states) in order to reduce computational time by avoiding loops, ideally all of these edges should be in the same basin, i.e. the starting- and endpoint V_i, V_j should be contained in the same super-state. We therefore compute the relative frequencies of these edges contained in the same basin, see Table 1.

Table 1 Percentage of edges contained in a common super-state for different values of threshold ν

$\nu = 0.9$	$\nu = 0.95$	$\nu = 0.99$
69.4	76.1	92.3

5.1.3 Optimization of segmentation

To improve the efficiency gains from the segmentation, we perform a postprocessing where we merge basins (super-states) according to the following criterion: If two connected hopping sites V_i and V_j have a high probability of building a loop, i.e., $\min\{p_{ij}, p_{ji}\} > \nu$, and the sites are in different, neighboring basins, we merge these two basins to a new super-state. The numbers and sizes of clusters for the stochastic watershed segmentation and its postprocessing (with $\nu = 0.9$) are given in Table 2, where $\bar{t} = \mathbb{E}\left(\tau_{\widetilde{M_1},\widetilde{M_2}} + \ldots + \tau_{\widetilde{M_N},\widetilde{M_{N+1}}}\right)$ denotes the average physical time achieved in $N = 10^5$ steps. As can be seen, in both cases, the average cluster sizes are reasonable small. This

means not too much information is lost by the segmentation. Further numerical results are given in Section 6 below.

Table 2 Analysis of super-states for stochastic watershed segmentation (WS) and postprocessed segmentation (PS)

	# of clusters	avg. size	# of clusters with size > 1	avg. size if > 1	max. size	$\bar{t}$
WS	1003	4.08	145	28.25	153	3.2×10^{-3}
PS	970	4.22	129	31.75	205	3.4×10^{-3}

5.2 A graph-theoretic decomposition

The DTMC $\{\widetilde{M}_1, \widetilde{M}_2, \ldots\}$ with transition matrix J introduced in Section 2.1 is said to be nearly-completely decomposable if it can be written in the form $J = J^* + \varepsilon C$ for sufficiently small $\varepsilon > 0$, where J^* is a block diagonal matrix. A good summary of the theory of nearly-completely decomposable Markov chains is given in Conway and Georganas (1982). The blocks in J^* correspond to subsets of states which have much higher levels of coupling to each other than they do to other states. Because the probabilities of moving within the subsets defined by the diagonal blocks in the J^* matrix are much higher than the probabilities of moving between subsets, these subsets are natural candidates for super-states in the coarsened state space model. Thus, another method of identifying super-states is to permute the J matrix so that it is in the form $J = J^* + \varepsilon C$.

5.2.1 Description of the decomposition algorithm

We use an algorithm given in Choi and Szyld (1996) in order to implement this method. It exploits the fact that the transition matrices of DTMCs can be represented as graphs. The set of vertices, V, represents the states, and the set of weighted edges, E, represents the transition probabilities. The algorithm works by taking a vertex of minimum degree in the collection of unclassified states. It uses this vertex as the basis of a super-state P. The algorithm considers every vertex adjacent to P. If the graph formed by the adjacent vertex, p, and P satisfies certain criteria, then p is added to P. This process continues until no more vertices can be added to P. At that stage, the vertices in P are classified as a super-state, and the algorithm begins again, considering the remaining unclassified vertices.

The criteria that the graph of $\{p\} \cup P$ — by which we mean the subgraph of the original graph defined by the vertex set $\{p\} \cup P$ — must satisfy are:

1. Either a *completeness* criterion or a *fullness* criterion.
   - The completeness criterion requires that $\frac{\phi_{\{p\}\cup P}}{\phi_P} > \alpha$, where ϕ_G is the ratio of the number of edges in the graph G to the number of edges G would have if it were complete.
   - The fullness criterion requires that p is adjacent to at least a proportion β of vertices in P.
2. A threshold criterion. This requires that at least one transition probability from p into a state in P is bigger than γ and that at least one transition probability from P to p is bigger than γ.

Thus, the algorithm given in Choi and Szyld (1996) can be summarized as follows.

Algorithm 5.1 (Graph-theoretic decomposition of the jump chain)

1. Given a collection of vertices C.
2. Set $P = Q = \emptyset$.
3. Choose from C a vertex c of minimum degree, mark it and add it to P.
4. Move to Q all vertices adjacent to c.
5. Choose a vertex p in Q.
6. If the fullness or connectivity criterion is satisfied and the threshold criterion is satisfied, then move p to P and add to Q all vertices in C adjacent to p. Otherwise, move p to C.
7. If $Q \neq \emptyset$, repeat from Step 5.
8. Set aside the vertices in P as a super-state.
9. If $C \neq \emptyset$, repeat from Step 2.

5.2.2 Choice of parameters

There are three parameters, α, β and γ, that control the segmentation. The parameters α and β both ensure that a state is only included in a super-state if it is connected to a sufficient proportion of the other states in the super-state. In practice, the charge carrier is often trapped in a region consisting of a number of interconnected states. However, it is possible that the charge carrier may be trapped in a high probability cycle (i.e. it may pass through the same sequence of states for a very large number of iterations) in which each state does not have many non-zero transition probabilities to other states in the cycle. For this reason, we do not set α or β too high. We set α and β equal to the same value, as choosing different values for both parameters does not have a significant impact on the segmentation. While α and β control the degree of connectivity required in super-states, γ ensures that the connections are sufficiently strong. That is, a state is only considered to be sufficiently well connected to other states in the super-state if it has at least one transition probability into the super-state that is greater than γ. Likewise, the super-state needs to be well connected to the state to be added, in the sense of having at least one transition probability greater than γ. This value is not set too high, in order to ensure that states ringing problem regions are included in segments.

Table 3 shows the effect of the different parameters on the fineness of the segmentation. To assess the efficiency of the simulation, we use the average physical time, $\bar{t}$, achieved in $N = 10^5$ steps as a proxy for efficiency. A higher physical time means that the charge carrier has explored more of the state space.

As can be seen from Table 3, there is a trade-off between the fineness of the segmentation and the efficiency of the resulting mobility estimator. In everything that follows we use the values $\alpha = \beta = 0.2$ and $\gamma = 0.02$, as these give both a fine segmentation and a reasonably efficient estimator.

6 Numerical results

The aim of the AMC approach is to maximize the amount of the state space that is explored in a given number of jumps by the charge carrier. This allows the stochastic process followed by the charge carrier to converge more quickly to stationarity, and gives correspondingly more accurate estimates of quantities such as drift velocity and charge carrier mobility, respectively. We measure the efficiency of the estimators in two different ways: the average estimate of $\hat{\mu}$ (estimated by repeated samples) and the average physical time $\bar{t}$ achieved in a given number of simulation steps.

Table 3 Analysis of super-states for the graph-theoretic segmentation

	# of clusters	avg. size	# of clusters with size > 1	avg. size if > 1	max. size	$\bar{t}$
$\gamma = 0.01$						
$\alpha = \beta$						
0.1	1121	3.66	469	7.35	65	4.8×10^{-3}
0.2	1217	3.37	532	6.41	31	3.6×10^{-3}
0.3	1334	3.07	611	5.52	21	3.3×10^{-3}
$\gamma = 0.02$						
$\alpha = \beta$						
0.1	1564	2.62	622	5.07	55	4.8×10^{-3}
0.2	1630	2.51	658	4.75	30	4.4×10^{-3}
0.3	1725	2.38	707	4.36	20	4.2×10^{-3}
$\gamma = 0.05$						
$\alpha = \beta$						
0.1	2283	1.79	821	3.21	24	5.8×10^{-4}
0.2	2284	1.79	822	3.20	24	5.8×10^{-4}
0.3	2294	1.79	828	3.18	17	5.9×10^{-4}

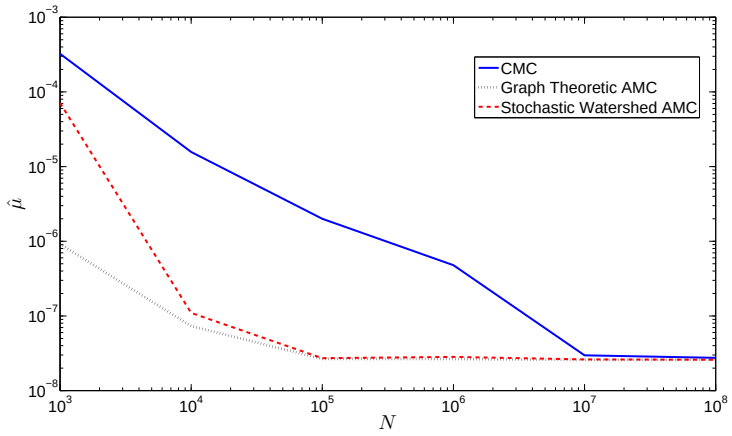

Fig. 9 Estimates of drift velocity for the CMC estimator and the AMC estimators based on the stochastic-watershed and graph-theoretic segmentations.

To estimate these quantities we calculate a large number of replicates of each estimator. The obtained average values of $\hat{\mu}$ are presented in Figure 9, see also Table 4, which show the enormous advantage of the AMC estimators in comparison to the standard CMC estimator. For a small number of steps, N, the WS-AMC estimator gives slightly larger values of $\hat{\mu}$ than the GT-AMC estimator. However, beginning from $N = 10^4$, both AMC estimators produce similar average values of $\hat{\mu}$ and reach the stability level of $\hat{\mu} = 2.6 \times 10^{-8}$ at $N = 10^5$. In contrast, for $N < 10^7$, the standard CMC estimator gives much larger values of $\hat{\mu}$ than the AMC estimators, reaching the stability level of $\hat{\mu} = 2.6 \times 10^{-8}$ only at $N = 10^7$.

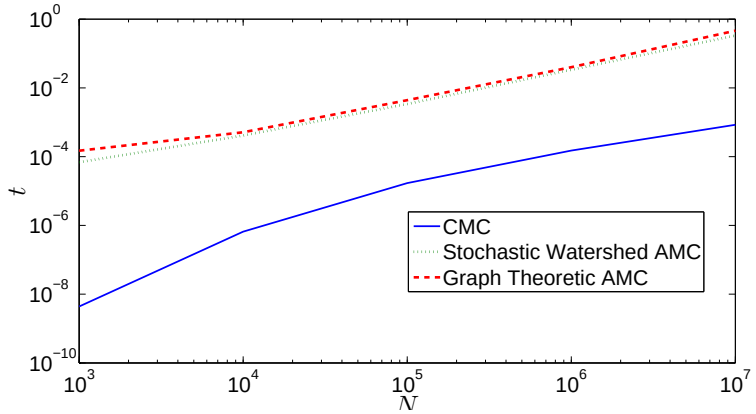

Fig. 10 Average physical times $\bar{t}$ of the simulation for the CMC estimator and the AMC estimators based on the stochastic-watershed and graph-theoretic segmentations.

A similar situation can be observed regarding the average physical time $\bar{t}$. For all considered step numbers (up to $N = 10^7$), the values of $\bar{t}$ produced by the AMC estimators are much larger than those of the standard CMC estimator, see Figure 10 and Table 5.

Table 4 Means of estimates of $\hat{\mu}$ (in nm/s) for the CMC estimator, stochastic-watershed (WS) AMC estimator and graph-theoretic (GT) AMC estimator, starting in state 1.

N	CMC	WS-AMC	GT-AMC
10^3	3.24×10^{-4}	7.11×10^{-5}	9.18×10^{-7}
10^4	1.57×10^{-5}	1.10×10^{-7}	7.32×10^{-8}
10^5	2.00×10^{-6}	2.83×10^{-8}	2.66×10^{-8}
10^6	4.78×10^{-7}	2.62×10^{-8}	2.65×10^{-8}
10^7	2.97×10^{-8}	2.59×10^{-8}	2.58×10^{-8}
10^8	2.75×10^{-8}	2.58×10^{-8}	2.59×10^{-8}

Table 5 Average physical times, $\bar{t}$, for the CMC estimator, WS-AMC estimator and GT-AMC estimator, starting in state 1.

N	CMC	WS-AMC	GT-AMC
10^3	4.39×10^{-10}	6.87×10^{-5}	1.49×10^{-4}
10^4	6.58×10^{-7}	4.12×10^{-4}	5.11×10^{-4}
10^5	1.70×10^{-5}	3.40×10^{-3}	4.40×10^{-3}
10^6	1.50×10^{-4}	3.37×10^{-2}	4.00×10^{-2}
10^7	8.44×10^{-4}	3.33×10^{-1}	4.60×10^{-1}

7 Conclusions

A methodology has been proposed that is suitable for efficient simulation of continuous-time Markov chains on weighted geometric graphs that are connected but nearly-completely decomposable. In this case, the number of steps that have to be simulated using the Crude Monte Carlo (CMC) algorithm is extremely large as the Markov chain can become stuck in small subregions of the state space and, therefore, the state space is not adequately explored. As an efficient alternative, Aggregate Monte Carlo (AMC) algorithms have been proposed, the idea of which is to coarsen the state space in such a way that spending too much simulation effort jumping back and forth in small subregions of the state space is avoided by aggregating such problem regions into single states. Two different methods to identify collections of states in which the Markov chain may become 'trapped' have been considered: the stochastic watershed segmentation from image analysis, and a graph-theoretic decomposition method. As an example application, the estimation the charge carrier mobility in disordered organic semiconductors has been investigated. Such semiconductors contain low-energy regions in which charge carriers can quickly become stuck. It turned out that the AMC estimators converge more quickly to stationarity, and give correspondingly more accurate estimates of quantities such as charge carrier mobility than the standard CMC estimator.

Acknowledgements

This work was supported by the DAAD (German Academic Exchange Service) / Go8 Australia-Germany Joint Research Cooperation Scheme. Furthermore, this work was partially supported by the DFG (German Research Foundation) under the priority programme 'Elementary Processes of Organic Photovoltaics' (SPP 1355). Dirk Kroese acknowledges the support of the Australian Research Council under grant number DP0985177. We are grateful to Kostas Daoulas for a critical reading of the manuscript.

References

1. Angulo J, Jeulin D (2007) Stochastic watershed segmentation. In: Proceedings of the 8th International Symposium on Mathematical Morphology, pp 265–276
2. Bässler H (1993) Charge transport in disordered organic photoconductors. Physica Status Solidi (b) 175(15):15–56
3. Baumeier B, May F, Lennartz C, Andrienko D (2012) Challenges for in silico design of organic semiconductors. Journal of Materials Chemistry 22:10,971–10,976
4. Beucher S, Lantuéjoul C (1979) Use of watersheds in contour detection. In: Proceedings of the International workshop on image processing, real-time edge and motion detection/estimation, pp 2.1–2.12
5. Beucher S, Meyer F (1993) The morphological approach to segmentation: the watershed transformation. In: Mathematical Morphology in Image Processing, vol 34, pp 433–481
6. Brereton T, Stenzel O, Baumeier B, Schmidt V, Kroese DP (2012) Efficient simulation of charge transport in deep-trap media. In: Laroque C, Himmelspach J, Pasupathy R, Rose O, Uhrmacher AM (eds) Proceedings of the 2012 Winter Simulation Conference
7. Choi H, Szyld DB (1996) Application of threshold partitioning of sparse matrices to Markov chains. In: Proceedings of the IEEE International Computer Performance and Dependability Symposium, IPDS'96, IEEE Computer Society Press, pp 158–165
8. Conway A, Georganas ND (1982) Queueing Networks - Exact Computational Algorithms. The MIT Press, Cambridge, Massachusetts
9. Courtois P (1977) Decomposability. ACM Monograph Series, Academic Press, New York
10. Diestel R (2005) Graph Theory. Springer, Berlin
11. Evans S (1996) Analysing system behaviour on different time scales. In: Kelly F, Zachary S, Ziedins I (eds) Stochastic Networks - Theory and Applications, Clarendon Press, Oxford, pp 231–246

12. Faessel M, Jeulin D (2010) Segmentation of 3d microtomographic images of granular materials with the stochastic watershed. Journal of Microscopy 239(1):17–31
13. van der Holst J, van Oost F, Coehoorn R, Bobbert P (2011) Monte Carlo study of charge transport in organic sandwich-type single-carrier devices: effects of Coulomb interactions. Physical Review B 83:085,206–1 – 085,206–13
14. Jansen A (2012) An Introduction to Kinetic Monte Carlo Simulations of Surface Reactions. Springer, Berlin
15. Pasveer W, Cottaar J, Tanase C, Coehoorn R, Bobbert P, Blom P, de Leeuw D, Michels M (2005) Unified description of charge-carrier mobilities in disordered semiconducting polymers. Physical Review Letters 94:206,601–1–206,601–4
16. Rühle V, Lukyanov A, May F, Schrader M, Vehoff T, Kirkpatrick J, Baumeier B, Andrienko D (2011) Microscopic simulations of charge transport in disordered organic semiconductors. Journal of Chemical Theory and Computation 7(10):3335–3345
17. Schönherr G, Bässler H, Silver M (1981) Dispersive hopping transport via sites having a Gaussian distribution of energies. Philosophical Magazine B 44(1):47–61
18. Silverman B (1986) Density Estimation for Statistics and Data Analysis. Chapman and Hall, London
19. Simon H, Ando A (1961) Aggregation of variables in dynamic systems. Econometrica 29(2):111–138
20. Somoza A, Ortuño M (2005) Monte Carlo method for relaxation in electron glasses. Physical Review B 72:224,202–1 – 224,202–6
21. Tessler N, Preezant Y, Rappaport N, Roichman Y (2009) Charge transport in disordered organic materials and its relevance to thin-film devices: a tutorial review. Advanced Materials 21:2741–4761
22. Tse D, Gallager R, Tsitsiklis J (1995) Statistical multiplexing of multiple time-scale Markov streams. IEEE J Sel Areas Commun 13(6):1028–1038