Stochastic Conformational Roadmaps for Computing Ensemble Properties of Molecular Motion

Mehmet Serkan Apaydın[1], Douglas L. Brutlag[1], Carlos Guestrin[1], David Hsu[2⋆], and Jean-Claude Latombe[1]

[1] Stanford University, Stanford, CA, USA
[2] National University of Singapore, Singapore

Abstract. A key intuition behind probabilistic roadmap planners for motion planning is that many collision-free paths potentially exist between two given robot configurations. Hence the connectivity of a robot's free space can be captured effectively by a network of randomly sampled configurations. In this paper, a similar intuition is exploited to preprocess molecular motion pathways and efficiently compute their ensemble properties, i.e., properties characterizing the average behavior of many pathways. We construct a directed graph, called *stochastic conformational roadmap*, whose nodes are randomly sampled molecule conformations. A roadmap compactly encodes many molecular motion pathways. Ensemble properties are computed by viewing the roadmap as a Markov chain. A salient feature of this new approach is that it examines all the paths in the roadmap simultaneously, rather than one at a time as classic methods such as Monte Carlo (MC) simulation would do. It also avoids the local-minima problem encountered by the classic methods. Tests of the approach on two important biological problems show that it produces more accurate results and achieves several orders of magnitude reduction in computation time, compared with MC simulation.

1 Introduction

Probabilistic roadmap (PRM) [1,8,9,13,16,17,25] planners have been successfully used in recent years to compute collision-free paths for robots with many degrees of freedom. A classic PRM planner [16] samples at random a robot's configuration space to construct a network that approximates the connectivity of the free space, and then searches the roadmap to process path planning queries. A key intuition behind PRM planners is that many collision-free paths potentially exist between two given robot configurations. Hence the connectivity of a robot's free space can be captured effectively by a network of randomly sampled configurations connected by collision-free curves. In this paper, a similar intuition is exploited to develop an efficient approach for analyzing the motion pathways of molecules during vital biological processes, such as protein folding and ligand-protein binding.

Molecules can be modeled approximately as articulated structures in 3-D space. They move under the influence of an energy field that includes van der Waals, electrostatic, and other potentials. For instance, to carry out biological functions, protein molecules remarkably assemble themselves, or *fold*, into unique 3-D structures called *native folds*. Protein folding plays a central role in biological processes essential to life. Failing to fold into the correct structures has serious consequences, including well-known diseases such as the Creutzfeldt-Jacobs (mad cow)

⋆ Work completed at the University of North Carolina at Chapel Hill.

or Alzheimer's disease. Despite its importance, the protein folding process remains a mystery. While it is traditionally studied through tedious and costly laboratory experiments, computer simulation plays an increasingly important role.

Classic techniques for simulating molecular motion, including Monte Carlo [15] and molecular dynamics [12] methods, have two major drawbacks:

1. They compute individual pathways, one at a time; however, many interesting properties of molecular motion, in particular, the *ensemble properties*, are best characterized statistically over many pathways. For instance, the "new view" of protein folding asserts that proteins fold in a multi-dimensional energy funnel by following a myriad of pathways, all leading to the same native structure.
2. A typical molecular energy function contains many local minima, and the classic simulation techniques waste considerable computation time trying to escape from these local minima. This is similar to the behavior of potential field planners (see, e.g., [6]) in robot motion planning.

The high computational cost of these existing techniques prevents them from being used to analyze many pathways.

The *Stochastic Roadmap Simulation* (SRS) framework described in this paper overcomes both drawbacks [3]. In SRS, we build a network, called *stochastic conformational roadmap*, or just *roadmap* for short. A roadmap compactly encodes many pathways and captures the stochastic nature of molecular motion. More precisely, a roadmap is a directed graph, whose nodes are randomly sampled molecule conformations. The *conformation* of a molecule specifies its 3-D structure; the concept is similar to that of the *configuration* of a robot. An edge between two nodes v_i and v_j in the roadmap carries a weight P_{ij}, which estimates the probability for the molecule to transition from v_i to v_j. Every path in the roadmap corresponds to a potential motion pathway of the molecule. A roadmap thus combines a huge number of pathways, weighted by the probabilities that the molecule may follow these pathways. The construction of the roadmap also circumvents the local-minima problem encountered by the classic simulation techniques.

The probabilities attached to the edges of a roadmap directly express the stochastic nature of molecular motion. We view the motion of molecules on the roadmap as a random walk similar to a Monte Carlo (MC) simulation run. At each step of the random walk, a molecule either stays at the current node or moves to a neighboring node according to the assigned transition probabilities. However, to compute efficiently the ensemble properties of molecular motion, we avoid explicit simulation. Instead, we treat the roadmap as a Markov chain and apply the first-step analysis [31] from the Markov chain theory to process all pathways in the roadmap simultaneously, rather than one at a time as classic methods such as MC simulation would do. Conceptually, this is equivalent to performing infinitely many simulation runs simultaneously and extracting statistics from them, but it results in tremendous gain in computational efficiency.

SRS is by necessity more coarse-grained in sampling than MC simulation. While a MC simulation run focuses on one individual pathway, SRS must spread the samples over the entire conformation space. On the other hand, SRS examines many

motion pathways at once and obtains information not easily accessible by classic methods. Tests of SRS on a number of protein folding and ligand-protein binding examples indicate empirically that SRS computes ensemble properties satisfactorily even with rather coarse roadmaps. In addition, it can be shown formally that, with appropriately defined edge probabilities, SRS and MC simulation converge to the same sampling distribution—the Boltzmann distribution—in the limit.

SRS is inspired by the PRM methods for robot motion planning. The stochastic conformational roadmap is a generalization of the probabilistic roadmap in PRM planners. In motion planning, the configuration space of a robot is the domain of a binary function specifying whether a configuration is collision-free. The conformation space of a molecule or a collection of molecules is the domain of a continuous energy function governing the motion of the molecules. As a result, the edges of a stochastic conformational roadmap are weighted with transition probabilities, while the edges of a probabilistic roadmap are unweighted.

Singh et al. introduced the PRM methods to the study of molecular motion in their work on ligand-protein binding [27]. Their approach has since been applied to protein folding as well [2,5,29]. Earlier work treats the roadmap as a deterministic graph, with heuristic edge weights based on the energy difference between molecule conformations. The heuristic edge weights measure the energetic difficulty of transitioning along the edges. Graph search techniques are used to extract "low-energy" paths from the roadmap. Our stochastic conformational roadmap is fundamentally different: it defines a Markov chain that captures the stochastic nature of molecular motion and enables us to analyze globally all the pathways contained in a roadmap by applying tools from Markov Chain theory. It also allows us to establish a formal relationship between SRS and the well-established MC method.

This paper combines and extends results presented in [3,4]. It provides additional experimental results and give new ideas on how efficient sampling strategies can be eventually designed to extend SRS to very high dimensional conformation spaces. In the following, we first cover some preliminaries (Sect. 2). We then describe how to construct a roadmap (Sect. 3) and query it to compute ensemble properties (Sect. 4). We used SRS to address two biological problems: computing the probability of folding in protein folding and estimating the escape time in ligand-protein binding. These experimental results are reported in Sects. 5 and 6.

2 Preliminaries

2.1 Conformation space

The conformation of a molecule determines its 3-D structure. Conformations can be specified in various ways. For a protein molecule, we can specify the positions of the constituent atoms or the backbone torsional angles ϕ and ψ. SRS applies to many different representations, provided that the conformation of a molecule is specified by a finite number of parameters that uniquely determine the 3-D position of every atom in the molecule. Formally, a conformation q of d parameters is specified by a vector $(q_1, q_2, \ldots, q_d)$. The set of all conformations form the *conformation space* $\mathcal{C}$.

By determining the molecule's 3-D structure, the conformational parameters also determine the interactions between the atoms of the molecule and between the molecule and the medium, e.g., van der Waals and electrostatic interactions. These interactions give rise to the attractive and repulsive forces that govern molecular motion. SRS assumes that these interactions are described by an energy function $E(q)$ that depends only on the conformation q of the molecule; it does not require E to have any particular properties or functional forms.

2.2 Monte Carlo simulation

MC simulation—more precisely, the Metropolis algorithm [20]—is one of the most common methods for simulating molecular motion. It samples the conformation space $\mathcal{C}$ of a system of molecules in order to study how they relax to or fluctuate around the equilibrium state. A key property of MC simulation is that, in the limit, the conformation space is sampled according to the Boltzmann distribution [18].

MC simulation starts at some initial conformation and performs a random walk in $\mathcal{C}$. Let q be the conformation at the current step of this random walk. To obtain the next conformation, a conformation q' is chosen from a small neighborhood of q, with a uniform or Gaussian distribution centered at q. The move to q' is accepted with a probability A that depends on the energy difference $\Delta E = E(q') - E(q)$. Define the *Boltzmann factors* $\varepsilon = \exp(-E(q)/k_{\text{B}}T)$ and $\varepsilon' = \exp(-E(q')/k_{\text{B}}T)$, where k_{B} is the Boltzmann constant, and T is the temperature of the system. The Metropolis criterion prescribes the acceptance probability as

$$A = \begin{cases} \exp(-\Delta E/k_{\text{B}}T) & \text{if } \varepsilon'/\varepsilon < 1 \\ 1 & \text{otherwise.} \end{cases} \tag{1}$$

Since $\varepsilon'/\varepsilon = \exp(-\Delta E/k_{\text{B}}T)$, the condition $\varepsilon'/\varepsilon < 1$ holds if and only if $\Delta E > 0$. So, if a move decreases the energy, it is always accepted; otherwise, it is accepted with probability $\exp(-\Delta E/k_{\text{B}}T)$. If the move from q to q' is accepted, the simulation transitions to q'; otherwise, it stays at q. The procedure then repeats to generate more sampled conformations, until termination conditions are met (e.g., the maximal number of steps is reached, or the quantity being computed stabilizes).

This procedure guarantees that if the number of simulation steps is sufficiently large, the sampled conformations are distributed according to the Boltzmann distribution with the density function

$$\beta(q) = \frac{1}{Z_\beta} \exp(-E(q)/k_{\text{B}}T),$$

where $Z_\beta = \int_{\mathcal{C}} \exp(-E(q)/k_{\text{B}}T)\, dq$ is a normalization constant. So any subset $S \subset \mathcal{C}$ is sampled with probability $\beta(S) = \int_S \beta(q)\, dq$.

MC simulation is an important tool for studying molecular motion. However, it is computationally intensive. Each simulation run yields a single pathway, and the simulation must be run many times over extended durations in order to produce accurate statistical results. Moreover, the energy function E typically contains many local minima. A MC simulation run spends most of its time overcoming energy barriers to escape from these local minima. Many similar conformations are sampled near the same local minimum, but they contain little new information.

2.3 Stationary distribution of a Markov chain

A Markov chain is a stochastic process that takes values from a finite or countable set of states $v_1, v_2, \ldots$ [31]. The probability of going from state v_i to v_j is P_{ij}, which depends only on v_i and v_j. Under suitable conditions, a Markov chain has a limit distribution $\pi = (\pi_1, \pi_2, \ldots)$ that can be obtained as follows. Starting at an initial state, perform a random walk over the set of all possible states. At each step of the random walk, make a move to the next state with the transition probability P_{ij}. If the random walk continues infinitely, then under the condition that the Markov chain is *ergodic*, each node v_i is visited with a fixed probability π_i in the limit, regardless of the starting state [31]. So π describes the limit behavior of *all* possible random walks. The probability π_i gives the fraction of the time that v_i is visited in the limit.

The limit distribution π satisfies the following equations [31]:

$$\pi_i = \sum_j \pi_j P_{ji} \quad \text{for all } i. \tag{2}$$

With the additional constraints $\pi_i \geq 0$ for all i and $\sum_i \pi_i = 1$, the solution to (2) is guaranteed to be a well-defined probability distribution. Equation (2) says that, in the limit, the distribution π no longer changes from one step of the random walk to the next. For this reason, π is called the *stationary distribution*.

If a conformation space is discretized into a finite set of states, MC simulation over this discretized space can be described by a Markov chain with appropriately defined transition probabilities. The stationary distribution of the Markov chain then gives the limit behavior of the MC simulation.

3 Stochastic conformational roadmaps

In Stochastic Roadmap Simulation, we preprocess molecular pathways by precomputing a roadmap, which provides a discrete representation of molecular motion. A roadmap compactly encodes a large number of MC simulation paths simultaneously and enables us to perform key computation efficiently.

3.1 Roadmap construction

A roadmap G is a directed graph. Each node of G is a randomly sampled conformation in $\mathcal{C}$. Each (directed) edge from node v_i to node v_j carries a weight P_{ij}, which represents the transition probability of the molecule to move from conformation v_i to v_j. The probability P_{ij} is 0 if there is no edge from v_i to v_j. Otherwise, it depends on the energy difference $\Delta E_{ij} = E(v_j) - E(v_i)$.

To construct a roadmap, our algorithm first samples conformations independently at random from $\mathcal{C}$. Our current implementation samples uniformly by picking a value for each conformational parameter $q_1, q_2, \ldots$ uniformly at random from its allowable range (see Section 7 for a discussion of more efficient sampling strategies). Next, for each node v_i, the algorithm finds its nearest neighbors according to a suitable metric such as the root mean squared distance [18]. It then creates

an edge between v_i and every neighboring node v_j and attaches to it the transition probability

$$P_{ij} = \begin{cases} \frac{1}{d_j} \exp(-\Delta E_{ij}/k_{\text{B}}T) & \text{if } \frac{\varepsilon_j/d_j}{\varepsilon_i/d_i} < 1 \\ \frac{1}{d_i} & \text{otherwise,} \end{cases} \tag{3}$$

where ε_i and ε_j are the Boltzmann factors at v_i and v_j, and d_i and d_j are the numbers of neighbors of v_i and v_j. If there is no edge between v_i and v_j, then they are considered too far apart for their energy difference to be a good basis for estimating the transition probability, and we set $P_{ij} = 0$. The molecule can still move from v_i to v_j, but the move must traverse at least one other node of the roadmap. Finally, a self-transition probability $P_{ii} = 1 - \sum_{j \neq i} P_{ij}$ is attached to each node v_i, ensuring that the transition probabilities from any node sum up to 1.

In contrast to the heuristic edge weights used in [5,27,29], the transition probabilities used in SRS allow us to establish a formal relationship between SRS and MC simulation [3]. We now describe this important relationship.

3.2 Relationship with Monte Carlo simulation

In MC simulation, we perform a random walk in the conformation space $\mathcal{C}$. We can perform a similar random walk on the roadmap G: at node v_i of G, we choose a node v_j uniformly at random from the set of neighbors of v_i and propose a move to v_j. The move is accepted with probability

$$A_{ij} = \begin{cases} \frac{d_i}{d_j} \exp(-\Delta E_{ij}/k_{\text{B}}T) & \text{if } \frac{\varepsilon_j/d_j}{\varepsilon_i/d_i} < 1 \\ 1 & \text{otherwise.} \end{cases} \tag{4}$$

Expressions (1) and (4) are similar, except for the extra factor d_i/d_j, which is needed because the neighborhood sizes of all conformations are the same in MC simulation, but the number of neighbors varies from one node to another for the random walk on the roadmap. Since node v_i has d_i neighbors and each one is chosen with probability $1/d_i$, the transition probability from v_i to v_j is $(1/d_i)A_{ij}$, which, after simplification, is equal to P_{ij} given in (3). Hence, with our choice of transition probabilities, every path in the roadmap corresponds to a MC simulation run.

We have also stated in Sect. 2.2 that MC simulation samples conformations with a distribution that converges to the Boltzmann distribution β. So, in the limit, the probability of sampling any subset $S \subset \mathcal{C}$ is

$$\beta(S) = \frac{1}{Z_\beta} \int_S \exp(-E(q)/k_{\text{B}}T)\, dq.$$

Now we ask the question: what is the limit behavior of SRS? In other words, if we perform an arbitrary long random walk on the roadmap as described above, what is the probability of sampling a subset $S \subset \mathcal{C}$? Since a roadmap defines a Markov chain with transition probabilities P_{ij}, the limit behavior of SRS is governed by the the stationary distribution of the Markov chain:

Lemma 1. *A stochastic conformational roadmap defines a Markov chain with stationary distribution*

$$\pi_i = \frac{1}{Z_\pi} \exp(-E(v_i)/k_{\text{B}}T) \quad \textit{for all } i, \tag{5}$$

where $Z_\pi = \sum_i \exp(-E(v_i)/k_{\text{B}}T)$ *is a normalization constant.*

See Appendix A for the proof. To estimate the probability of sampling a set S, we simply sum the stationary distribution π over all the nodes v_i that lie in the set S:

$$\pi(S) = \sum_{v_i \in S} \pi_i = \frac{1}{Z_\pi} \sum_{v_i \in S} \exp(-E(v_i)/k_{\mathrm{B}}T).$$

If SRS represents the stochastic motion of a molecule with the same limit behavior as MC simulation, we expect the limit distributions of these two methods to converge. In other words, given a suitably dense roadmap, $\pi(S)$ should approximate $\beta(S)$ to any arbitrary precision, a result proven in our earlier paper [3].

Although SRS is closely related to MC simulation, it is far more efficient. A roadmap constructed by SRS combines many MC simulation paths, which can be processed together using tools from Markov chain theory, as we will see next.

4 Roadmap query

A roadmap G encodes considerable information on molecular motion. Given two nodes v_i and v_j in G, we can compute the most likely pathway from v_i to v_j by searching for a minimum-weight path from v_i to v_j in a graph similar to G, but with $-\ln P_{ij}$ as edge weights. This leads to results similar to those in earlier work [5,27,29]. However, since a roadmap explicitly represents the stochastic nature of molecular motion, it allows us to take advantage of powerful tools from the Markov chain theory. We now focus on one such tool, the *first-step analysis*.

To illustrate our description with a concrete example, consider a roadmap G built in the conformation space of a protein whose native fold is known. Let $\mathcal{F}$ be a set of nodes of G in the folded state. In other words, they are structurally close to the native fold. We are interested in finding, for every node v_i in G, the expected number of transitions t_i needed to go from v_i to the folded state, i.e., any node in $\mathcal{F}$. A naive way to compute t_i would be to perform many MC simulation runs, starting from v_i, and average the number of transitions taken by each run. Due to the potentially high variance among independent runs, it takes a large number of simulation runs for each node v_i in order to get accurate results.

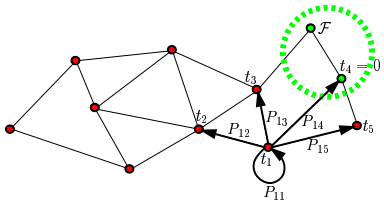

Fig. 1. First-step analysis.

In contrast, the first-step analysis proceeds by conditioning on the first transition. Suppose that we start at some node $v_i \notin \mathcal{F}$. After one step of transition, t_i is increased by one, and we either enter the folded state or reach another node $v_j \notin \mathcal{F}$. In the former case, we simply stop. In the latter case, the expected number of steps from then on is t_j. So we have the following system of linear equations:

$$t_i = 1 + \sum_{v_j \in \mathcal{F}} P_{ij} \cdot 0 + \sum_{v_j \notin \mathcal{F}} P_{ij} \cdot t_j \quad \text{for every } v_i \notin \mathcal{F}. \tag{6}$$

In the second term of (6), P_{ij} is multiplied by zero, because we stop as soon as we enter the folded state. See Fig. 1 for an illustration. The linear system (6) contains

one equation and one unknown for each node $v_i \notin \mathcal{F}$. By solving this system, we obtain t_i for all the nodes simultaneously, without any explicit simulation.

To solve the linear system, let us rewrite (6) in the matrix form:

$$\mathbf{t} = \mathbf{Q} \cdot \mathbf{t} + \mathbf{b}, \tag{7}$$

where $\mathbf{Q}$ is a matrix with the transition probabilities P_{ij} as the entries, $\mathbf{t}$ is the vector of unknowns t_i, and $\mathbf{b}$ is a vector collecting the remaining constant terms in (6). Since a roadmap usually contains many nodes, the size of the coefficient matrix $\mathbf{I} - \mathbf{Q}$ is large ($\mathbf{I}$ is the identity matrix). Direct methods for solving (7), based on, e.g., Gaussian elimination, are impractical. However, it can be shown that a unique solution to (7) exists. So iterative methods can be used instead. In fact, the naive iteration $\mathbf{t}^{(k+1)} = \mathbf{Q} \cdot \mathbf{t}^{(k)} + \mathbf{b}$ is guaranteed to converge to the unique solution. This simple iterative method amounts to performing many simulation runs simultaneously using matrix multiplication. More efficient iterative methods, such as the conjugate gradient method [23], can also be used. Finally every roadmap node is, by construction, connected to only a small number of neighboring nodes, resulting in a sparse matrix $\mathbf{Q}$. The sparsity can be exploited to accelerate the iterative solver.

5 Computing the probability of folding

In this and next section, we use SRS to compute two ensemble properties: the *probability of folding* in protein folding and the *escape time* in ligand-protein binding.

Protein folding is one of the most marvelous processes in nature. Under suitable conditions, proteins go through a series of geometric transformations and arrive at the native folds where they perform intricate biological functions. There are large on-going efforts to understand the folding process (e.g., [14,22]): What geometric transformations does a protein go through during folding? Which conformations are "closer" to the native fold along the folding pathways?

To address this type of questions, the probability of folding ($\mathrm{P_{fold}}$)—also known as the transmission coefficient—has been introduced to measure how far away a protein conformation is from the native fold kinetically [11]. For a folding process dominated by two stable states, a folded state $\mathcal{F}$ and an unfolded state $\mathcal{U}$, the $\mathrm{P_{fold}}$ value τ for a conformation q is the probability of reaching $\mathcal{F}$ before $\mathcal{U}$, starting from q. If $\tau > 0.5$, then the protein is more likely to fold first than to unfold first, and therefore q is kinetically closer to the folded state [11]. Trivially, if q is in $\mathcal{F}$, then $\tau = 1$, and if q is in $\mathcal{U}$, then $\tau = 0$. The $\mathrm{P_{fold}}$ value at q is not associated with any particular folding pathway, but depends on all possible pathways from q. It is thus an ensemble property that describes the average behavior of the folding process.

5.1 Algorithmic details

Using SRS, we can compute $\mathrm{P_{fold}}$ as follows. Let $v_i, i = 1, 2, \ldots$ be the nodes in a roadmap, and τ_i be the $\mathrm{P_{fold}}$ value for v_i. After constructing the roadmap, first-step analysis yields the following equation for every node v_i not in $\mathcal{F}$ or $\mathcal{U}$:

$$\tau_i = \sum_{v_j \in \mathcal{F}} P_{ij} \cdot 1 + \sum_{v_j \in \mathcal{U}} P_{ij} \cdot 0 + \sum_{v_j \notin (\mathcal{F} \cup \mathcal{U})} P_{ij} \cdot \tau_j, \tag{8}$$

which is obtained by conditioning on the first transition. After one step of transition, we have three cases. In the first case, we reach a node in $\mathcal{F}$. Then, we have reached $\mathcal{F}$ before $\mathcal{U}$ with probability 1. In the second case, we reach a node in $\mathcal{U}$. Then, we have reached $\mathcal{U}$ before $\mathcal{F}$, and the probability of reaching $\mathcal{F}$ before $\mathcal{U}$ is 0. In the third case, we reach a node v_j in neither $\mathcal{F}$ nor $\mathcal{U}$. The probability τ_i then depends on the value of τ_j. Linear system (8) has the same matrix form as the example in Sect. 4. A unique solution exists and can be obtained by an iterative solver.

We can improve the accuracy and potentially the speed of the iterative solver by setting all the self-transition probabilities in the roadmap to 0 and renormalizing the other probabilities. Set

$$\begin{aligned} P'_{ii} &= 0 && \text{for all } i, \\ P'_{ij} &= P_{ij} / \textstyle\sum_{k \neq i} P_{ik} && \text{for all } i \neq j \end{aligned} \tag{9}$$

and solve the linear system

$$\tau_i = \sum_{v_j \in \mathcal{F}} P'_{ij} \cdot 1 + \sum_{v_j \in \mathcal{U}} P'_{ij} \cdot 0 + \sum_{v_j \notin (\mathcal{F} \cup \mathcal{U})} P'_{ij} \cdot \tau_j. \tag{10}$$

If we think in terms of performing a random walk on the roadmap as described in Sect. 3.2, then setting the self-transition probabilities to 0 means never staying at the current node. It is easy to verify that linear systems (8) and (10) have the same solution by substituting (9) into (10). However, if we write (10) in the matrix form, the coefficient matrix $\mathbf{I} - \mathbf{Q}$ contains 1 in all its diagonal entries, which are greater than or equal to the corresponding entries in the matrix for (8). So (10) tends to be better conditioned, resulting in more stable, faster solution by iterative methods.

5.2 Experimental results

We now present results on three examples. The first one uses a relatively simple synthetic energy function in a 2-D conformation space, and the other two use real protein data. We compare the results from SRS to those from MC simulation, and demonstrate that SRS reduces the running time by several orders of magnitude and is more accurate. The main reason for using synthetic data in the first example is that MC simulation takes extremely long computation time on real proteins. The simpler synthetic energy function allows us to perform more extensive comparisons.

Synthetic data The synthetic energy function in a 2-D conformation space is constructed from a linear combination of radially symmetric Gaussians, with a paraboloid centered at the origin. The centers, the decay rates, and the heights of the Gaussians are picked at random. The function has two global minima, one of which represents the native fold. When constructing a roadmap in this hypothetical conformation space, we use the Euclidean distance in the 2-D space for finding neighboring nodes.

We used SRS to compute P_{fold} for 101 sampled conformations with a roadmap of 10102 nodes, and

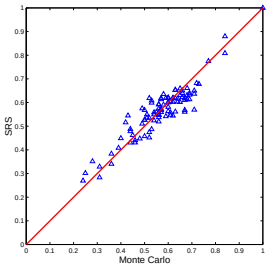

Fig. 2. The scatter plot of P_{fold} values from MC simulation and from SRS on a synthetic energy function.

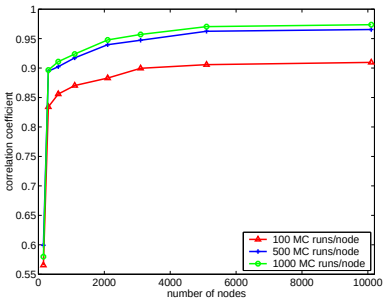

Fig. 3. The correlation coefficient κ as a function of the number of nodes in the roadmap. The three curves correspond to MC simulation with $r = 100, 500, 1000$ independent runs per node. As r increases, the correlation between the results from the two methods improves.

then used MC simulation to compute the results for the same conformations. In the MC simulation, we performed 500 independent runs for each conformation. Each run stops as soon as it enters a small neighborhood of a conformation in the folded or the unfolded state. The results computed with the two methods are plotted along the two axes in Fig. 2. All the points in the plot lie close to the diagonal line, indicating that the results from the two methods are in good correspondence.

We conducted further tests by varying the number of nodes sampled by SRS and the number of independent MC simulation runs per node. In each test, we summarize the correspondence between the results from the two methods by their (normalized) correlation coefficient, which is defined as

$$\kappa(x, y) = \frac{\langle xy \rangle - \langle x \rangle \langle y \rangle}{\sqrt{(\langle x^2 \rangle - \langle x \rangle^2)(\langle y^2 \rangle - \langle y \rangle^2)}}$$

for two vectors x and y, where $\langle \cdot \rangle$ denotes the operation of taking the average. Note that κ is always between -1 and 1, with 1 indicating perfect correlation, -1 indicating perfect inverse correlation, and 0 indicating no correlation. Fig. 3 shows the results of these additional tests. The horizontal axis of the graph is the number of nodes in the roadmap, and the vertical axis is the correlation coefficient κ. The graph contains three curves, corresponding to different numbers of independent MC simulation runs per node. These curves show a generally similar trend. Initially κ improves quickly as the number of nodes in the roadmap increases. The curves then flatten out after a certain point. It is not immediately clear whether they will reach 1, which indicates perfect correlation. Since κ measures only the correspondence between the two methods and we do not know the ground truth, these general trends do not tell us whether the discrepancy is due to the inaccuracy in SRS or the variance inherent in MC simulation. However, we can get a hint by comparing the three curves. For a roadmap of a given size, κ generally improves as we increase the number of independent MC simulation runs per node. This seems to indicate that SRS is more accurate: when the number of independent MC simulation runs per node increases, the variance of MC simulation decreases, and simultaneously, the results get closer to those obtained from SRS.

We also compared the running time of the two methods. The running time of SRS consists of the time to construct the roadmap and the time to solve a linear system of equations. On real proteins, the first part is dominated by the time to evaluate the energy of sampled nodes and the time to find neighboring nodes. The

Table 1. Running times of 100 MC simulation runs per conformation on the synthetic energy function. The first row gives the number of conformations for which P_{fold} is computed. The second row gives the corresponding running times.

No. Conf.	10	20	30	40	50	60	70	80	90	100
Time (sec.)	866	1588	2356	3191	4026	4913	5621	6404	7203	8077

second part depends on the size of the linear system, hence on the number of nodes in the roadmap. The running time of MC simulation is dominated by the time to compute the energy of sampled conformations.

In our current implementation, the roadmap construction part of SRS is coded in C++, and the linear system solver, in Matlab. MC simulation is implemented entirely in C++. The timing results reported here were obtained on a 1GHz Pentium-III PC with 1GB of memory. In a typical run on the synthetic energy function, SRS took about 8 seconds to construct a roadmap of 10,000 nodes, and 3 seconds to solve the linear system and obtain P_{fold} values for all the nodes. The running times of MC simulation is tabulated in Table 1. As expected, the running time of MC simulation is linear in the number of conformations processed. Although we did not perform MC simulation on all 10,000 conformations, we can easily infer that that the running time would be around 800,000 seconds.

1ROP and 1HDD We also tested SRS on two real proteins (Fig. 4), repressor of primer and engrailed homeodomain, which are identified as 1ROP and 1HDD, respectively, in the Protein Data Bank [7]. 1ROP is a dimer that consists of two identical parts called monomers. As in [30], we study one monomer in isolation. The monomer contains 56 residues forming two α helices connected by a loop. 1HDD contains 57 residues forming three α helices packed against each other.

Our implementation represents a protein as a sequence of vectors, each representing a secondary structure element [5,26]. In this vector-based representation, 1ROP has 6 degrees of freedom (dofs), and 1HDD has 12 dofs. Our energy function uses an H-P model [30] consisting of two terms, measuring the hydrophobic interaction and the excluded volume, respectively. In both SRS and MC simulation, we discard conformations that cause steric hindrance, i.e., self-collision of atoms in the protein. We define the folded state to contain all conformations within a small root mean squared distance (RMSD) of the native fold (3 Å for 1ROP and 5 Å for 1HDD), and the unfolded state to contain all the conformations within 10 Å of the fully-extended conformation. The roadmap construction software uses the RMSD as a metric to find neighboring nodes, as it better measures the similarity between two protein conformations than the Euclidean distance.

For each protein, we computed the P_{fold} values at about 45 randomly selected conformations using both SRS and MC simulation. With SRS, we computed the estimates with roadmaps having increasing numbers of nodes. In MC simulation, we performed up to 300 independent runs for each of the selected conformations. The results, shown in Fig. 5, suggest conclusions similar to those obtained from the synthetic energy function. First, SRS estimates generally improve very fast as

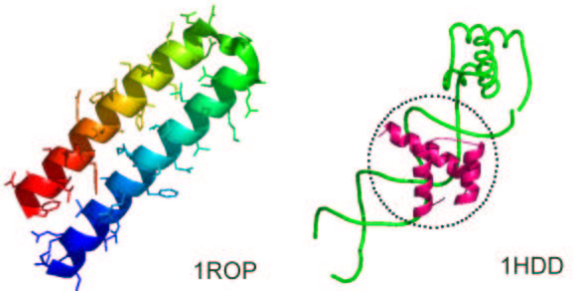

Fig. 4. Two proteins used in the experiments: 1ROP and 1HDD (circled) in complex with DNA.

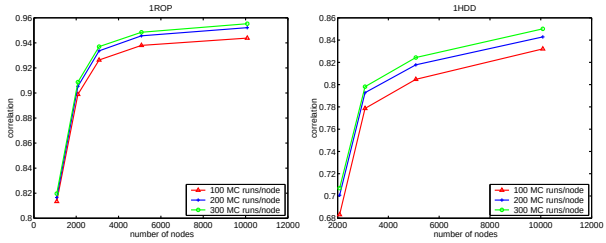

Fig. 5. The correlation of P_{fold} values for 1ROP and 1HDD, computed by SRS and MC simulation.

the roadmap size increases. Second, the correlation tends to increase as we perform more MC runs per node. We also compared the P_{fold} values obtained from the two methods using their average absolute differences, instead of their correlation coefficients. The conclusion is similar.

The total time to generate a roadmap of 5000 nodes and compute the P_{fold} values for all these nodes was about 1.5 hours. In comparison, it took an average of five to six hours on the same machine to execute 300 MC simulation runs required to estimate P_{fold} at just *one* conformation. To compute P_{fold} at the 45 selected conformations, MC simulation took about 250 hours, while SRS took only *1.5 hours* for all 5000 conformations. So SRS reduced the running time by at least four orders of magnitude in these examples.

6 Estimating the escape time from a ligand-protein binding site

Ligand-protein binding is another important biological process, in which a small molecule, called a *ligand*, is bound to a specific site, usually a cavity on the surface of a receptor protein in order to inhibit or enhance activities of the protein. Studying ligand-protein binding helps in discovering new pharmaceutical drugs. For example, drug molecules have been designed to bind to the active site of the enzymatic protein HIV-1 protease. They block access to the active site of the amino acid chains forming part of the HIV virus and thus disable the activation of the mature virus.

A receptor protein may have several potential binding sites. Therefore, it is important to be able to predict which is the active site, the site that enables specific biological functions, e.g., inhibition or catalysis. Let us consider the conformation space $\mathcal{C}$ of a ligand-protein complex with a suitably defined energy function. A bound conformation $q \in \mathcal{C}$ generally corresponds to a local energy minimum and

has a "funnel of attraction" around q to stabilize the ligand. At the active site, the ligand is usually bound with very high affinity, and so it takes much longer to escape from the corresponding funnel, compared with other potential binding sites. We hypothesize that the longer escape time results from higher energy barriers around the active site and it may serve as a basis for prediction. We thus examine the ligand-protein binding process using a dynamic model. In contrast, most existing methods for analyzing ligand-protein binding use static models and consider only the energy of the final bound conformation of the ligand-protein complex (e.g., see [21,33]).

Following [10], we define the funnel of a bound conformation q as the set of conformations within 10 Å of q in RMSD. We then use SRS to compute the expected number of transitions for the ligand to reach a conformation outside of the funnel and use it as an approximation to the escape time.

6.1 Algorithmic details

Given a suitable energy function for the ligand-protein complex, we first construct a roadmap to capture the motion of the ligand and then apply first-step analysis to obtain a system of equations similar to (6). Let $\mathcal{A}$ be the set of roadmap nodes in the funnel of the bound conformation q_{b}. Let t_i be the expected number of transitions to reach a conformation outside of $\mathcal{A}$, starting from a node $v_i \in \mathcal{A}$. We have

$$t_i = 1 + \sum_{v_j \notin \mathcal{A}} P_{ij} \cdot 0 + \sum_{v_j \in \mathcal{A}} P_{ij} \cdot t_j \quad \text{for every } v_i \in \mathcal{A}. \tag{11}$$

The solution to this system of equations gives an approximation of the escape time for every node in the funnel, including the bound conformation q_{b}.

6.2 Experimental results

We applied our method to seven different ligand-protein complexes whose active sites are known. They are listed in Table 2. For each complex, we assume that the protein is rigid, and that the ligand is flexible with varying number of torsional dofs listed in column 3 of the table. Our energy function models the ligand-protein interactions by incorporating terms for van der Waals and electrostatic interactions [28].

To find potential binding sites, we picked 10,000 conformations at random and performed descent from them until we reached local energy minima. For each complex, four low-energy conformations that are less than 5 Å to the protein surface and are separated by greater than 10 Å in RMSD were selected along with the known active site as the potential bound conformations.

We computed 20 roadmaps of 10,000 nodes each for every potential binding site. The nodes were uniformly sampled in a region within 15 Å in RMSD of the bound conformation. We then solved for the escape times using (11). The results were averaged and listed in columns 5–8 of Table 2. Every row of the table shows the estimates of escape times for the various binding sites of a ligand-protein complex. In four of the seven cases, the escape time for the active site is larger than those for the other binding sites by at least two orders of magnitude, clearly distinguishing the active site. In two other cases (1LDM and 1CJW), the escape time for

Table 2. Escape times from potential binding sites for seven ligand-protein complexes.

Protein	Ligand	dofs	Binding Sites				
			Active	1	2	3	4
1LDM	oxamate	7	5.8e+06	1.6e+07	1.1e+06	3.7e+06	4.5e+05
1AO5	IPM	10	4.1e+10	1.2e+07	7.9e+06	1.2e+05	2.9e+04
3TPI	PTI	13	1.0e+10	1.1e+06	1.8e+05	1.0e+05	6.6e+05
4TS1	hydroxylamine	9	2.4e+10	5.4e+06	4.2e+07	7.2e+05	2.2e+06
1CJW	COT	21	6.3e+06	8.2e+06	5.6e+05	1.5e+05	1.9e+05
1AID	THK	14	1.4e+06	2.8e+07	5.0e+05	1.2e+05	2.1e+06
1STP	streptavidin	11	7.0e+08	6.4e+06	2.2e+06	8.5e+05	2.0e+06

the active site is close to the largest. In one case (1AID), the escape time fails to give a clear indication on the active site. The failure may have several causes. The size of the roadmaps may be too small to estimate the escape times accurately. The energy function that we use may not be detailed enough to capture all significant interactions between the ligand and the protein. Finally, it is possible that the active site often, but not always, has the highest escape time in nature.

For each binding site, our software took about 7 minutes in total to construct the roadmap and solve the linear system to obtain an estimate of the escape time.

Using a similar approach, we also studied the effect of mutating individual amino acids in the catalytic site of enzymatic proteins. These additional experimental results and others are available in [4,32].

7 Conclusion and future work

Stochastic Roadmap Simulation is a new framework for computing ensemble properties of molecular motion. It is closely related to MC simulation. A path in a stochastic conformational roadmap can be interpreted as a MC simulation run. Furthermore SRS converges to the same sampling distribution as MC simulation. A salient feature of SRS is that it compactly encodes many motion pathways and processes them together by solving linear equations, rather than considering them one at a time as the classic Monte Carlo and molecular dynamic methods would do. It also avoids the local-minima problem that plagues the existing methods. As a result, SRS gains tremendous computational efficiency, as demonstrated by our experiments.

We applied SRS to two biological problems. In the first problem, we computed the probability of folding, which measures the "kinetic distance" between a protein conformation and the native fold. Our experiments show that SRS reduced the running times by several orders of magnitude and obtained arguably more accurate results, compared with MC simulation. In the second problem, we computed estimates of the expected time for a ligand to escape from a binding site and used it to compare the active site of a protein to other potential binding sites.

SRS can be extended in several ways. An important algorithmic question is to develop sampling strategies that allow us to study larger molecules with more complex energy models. Currently we sample the conformation space $\mathcal{C}$ or a subset of it uniformly at random. As the dimension of $\mathcal{C}$ increases, it becomes more difficult to

obtain biologically interesting conformations with uniform sampling, and the quality of results obtained from uniformly sampled roadmaps is likely to degrade. This is similar to the development of PRM planners for motion planning. To solve complex motion planning problems in high-dimensional configuration spaces, efficient sampling strategies are needed (see [24] for a review).

To address the problem, one approach is to construct a sampling distribution that favors low-energy conformations over high-energy ones, because molecules are more likely to stay in low-energy states. We achieve this by first sampling conformations uniformly at random and then retaining them with probability that decreases as the the energy of the conformations increases. We may also resample near known low-energy conformations to boost the density of roadmap nodes in these potentially interesting regions. Equally important is to identify energy barriers between neighboring nodes in a roadmap, when computing transition probabilities. To this end, we may sample the straight-line path between two neighboring nodes and compute the energy of intermediate conformations along the path. Finally, biologically interesting conformations are known to be often located in regions where the energy function undergoes significant variations, e.g., protein conformations in the transition state. To increase the sampling density in these regions, techniques such as the Gaussian sampling [9] can be used to sample a pair of conformations and retain a sample with higher probability when the pair exhibits very different energies.

Assume that a good sampling distribution σ can be constructed. We must then adjust the transition probabilities to account for the non-uniformity of the roadmap nodes so that SRS still converges to the Boltzmann distribution in the limit. One possibility is to define the new transition probability

$$P_{ij} = \begin{cases} \frac{\sigma_i}{d_j \sigma_j} \exp(-\Delta E_{ij}/k_{\text{B}}T) & \text{if } \frac{\varepsilon_j/d_j\sigma_j}{\varepsilon_i/d_i\sigma_i} < 1 \\ \frac{1}{d_i} & \text{otherwise,} \end{cases}$$

where σ_i and σ_j are the probabilities of sampling nodes v_i and v_j. Work is underway to investigate the validity of this transition probability assignment.

Another interesting extension is to add nodes incrementally to the roadmap and refine the solution of SRS, thus avoiding the need to choose the number of roadmap nodes in advance.

We are also interested in applying SRS to other important problems related to molecular motion. For instance, our transition probabilities are closely related to the master equation [19], which has been used to study the rate of protein folding. Earlier work on this problem exhaustively enumerates all conformations and is thus limited to very small proteins on a lattice (in the plane). Our experience indicates that SRS will likely remove the need for such costly enumeration.

Acknowledgements This work has been funded in part by NSF ITR grants ACI-0082554 and CCR-0086013 and a grant from Stanford Bio-X program. Apaydin was supported by the D.L. Cheriton Stanford Graduate Fellowship. Brutlag was supported by NHGRI grant HGF02235. Guestrin was supported by a Siebel Scholarship and by the Sloan Foundation. We thank D. Koller, V. Pande, A. Singh, and J. Snoeyink for helpful discussions. We also thank C. Varma for preparing the energy files for some of the ligand-protein complexes.

References

1. N.M. Amato, O.B. Bayazit, L.K. Dale, C. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In P.K. Agarwal et al., editors, *Robotics: The Algorithmic Perspective: 1998 Workshop on the Algorithmic Foundations of Robotics*, pages 155–168. A. K. Peters, Wellesley, MA, 1998.
2. N.M. Amato, K.A. Dill, and G. Song. Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 2–11, 2002.
3. M.S. Apaydin, D.L. Brutlag, C. Guestrin, D. Hsu, and J.C. Latombe. Stochastic roadmap simulation: An efficient representation and algorithm for analyzing molecular motion. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 12–21, 2002.
4. M.S. Apaydin, C.E. Guestrin, Chris Varma, D.L. Brutlag, and J.C. Latombe. Stochastic roadmap simulation for the study of ligand-protein interactions. *Bioinformatics*, 18(supplement 2):18S–26S, 2002.
5. M.S. Apaydin, A.P. Singh, D.L. Brutlag, and J.C. Latombe. Capturing molecular energy landscapes with probabilistic conformational roadmaps. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 932–939, 2001.
6. J. Barraquand and J.C. Latombe. Robot motion planning: A distributed representation approach. *Int. J. Robotics Research*, 10(6):628–649, 1991.
7. F.C. Bernstein et al. The protein data bank: A computer-based archival file for macromolecular structure. *J. Mol. Biol.*, 112(3):535–542, 1977.
8. R. Bohlin and L.E. Kavraki. Path planning using lazy PRM. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 521–528, 2000.
9. V. Boor, M.H. Overmars, and F. van der Stappen. The Gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. on Robotics & Automation*, pages 1018–1023, 1999.
10. C.J. Camacho and S. Vajda. Protein docking along smooth association pathways. *Proc. Nat. Acad. Sci. USA*, 98(19):10636–10641, 2001.
11. R. Du, V. Pande, A.Y. Grosberg, T. Tanaka, and E. Shakhnovich. On the transition coordinate for protein folding. *J. Chem. Phys.*, 108(1):334–350, 1998.
12. J.M. Haile. *Molecular Dynamics Simulation: Elementary Methods*. John Wiley & Sons, New York, 1992.
13. D. Hsu, J.C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *Int. J. Computational Geometry & Applications*, 9(4 & 5):495–512, 1999.
14. IBM Blue Gene Team. Blue gene: A vision for protein science using a petaflop supercomputer. *IBM Systems Journal*, 40(2):310–327, 2001.
15. M.H. Kalos and P.A. Whitlock. *Monte Carlo Methods*, volume 1. John Wiley & Son, New York, 1986.
16. L.E. Kavraki, P. Švestka, J.C. Latombe, and M.H. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration space. *IEEE Trans. on Robotics & Automation*, 12(4):566–580, 1996.
17. S.M. LaValle and J.J. Kuffner. Randomized kinodynamic planning. *Int. J. Robotics Research*, 20(5):278–400, 2001.
18. A.R. Leach. *Molecular Modelling: Principles and Applications*. Longman, Essex, England, 1996.
19. M.Cieplak, M.Henkel, J. Karbowski, and J.R.Banavar. Master equation approach to protein folding and kinetic traps. *Phys. Rev. Let.*, 80:3654, 1998.
20. N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of state calculations by fast computing machines. *J. Chem. Phys.*, 21:1087–1092, 1953.

21. G.M. Morris, D.S. Goodsell, R.S. Halliday, R. Huey, W.E. Hart, R.K. Belew, and A.J. Olson. Automated docking using a Lamarckian genetic algorithm and an empirical binding free energy function. *J. Comput. Chem.*, 19(14):1639–1662, 1998.
22. V.S. Pande et al. Atomistic protein folding simulations on the hundreds of microsecond timescale using worldwide distributed computing. *Biopolymers*, to appear.
23. Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS, New York, 1996.
24. G. Sanchez and J.C. Latombe. On delaying collision checking in PRM planning—application to multi-robot coordination. *Int. J. Robotics Research*, 21(1):5–26, 2002.
25. T. Siméon, J.P. Laumond, and F. Lamiraux. Move3D: A generic platform for motion planning. In *Proc. IEEE Int. Symp. on Assembly & Task Planning*, 2001.
26. A.P. Singh and D.L. Brutlag. Hierarchical protein structure superposition using both secondary structure and atomic representations. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology*, pages 284–293, 1997.
27. A.P. Singh, J.C. Latombe, and D.L. Brutlag. A motion planning approach to flexible ligand binding. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology*, pages 252–261, 1999.
28. K. Smith and B. Honig. Evaluation of the conformational free energies of loops in proteins. *Proteins: Structure, Function, and Genetics*, 18:119–132, 1994.
29. G. Song and N.M. Amato. Using motion planning to study protein folding pathways. In *Proc. ACM Int. Conf. on Computational Biology (RECOMB)*, pages 287–296, 2001.
30. S. Sun, P.D. Thomas, and K.A. Dill. A simple protein folding algorithm using a binary code and secondary structure constraints. *Protein Engineering*, 8:769–778, 1995.
31. H. Taylor and S. Karlin. *An Introduction to Stochastic Modeling*. Academic Press, New York, 1994.
32. C. Varma. Computing protein-ligand interaction kinetics using Markov methods. Master's thesis, Dept. of Computer Science, Stanford University, Stanford, CA, 2002.
33. J. Wang, P.A. Kollman, and I.D. Kuntz. Flexible ligand docking: A multiple strategy approach. *Proteins: Structure, Function, and Genetics*, 36(1):1–19, 1999.

A Proof of Lemma 1

Proof. We would like to show that the distribution π given in (5) is the stationary distribution for the Markov chain induced by the roadmap G. First, note that it suffices to show that π satisfies the detailed balance [31]:

$$\pi_i P_{ij} = \pi_j P_{ji}, \tag{12}$$

because if (12) holds, then $\sum_j \pi_j P_{ji} = \sum_j \pi_i P_{ij} = \pi_i \sum_i P_{ij} = \pi_i$, as required by the condition for a stationary distribution, given in (2). Now consider two nodes v_i and v_j from the roadmap. Without loss of generality, assume $\frac{\varepsilon_j / d_j}{\varepsilon_i / d_i} < 1$. We have

$$P_{ij} = \frac{1}{d_j} \exp(-\Delta E_{ij} / k_{\mathrm{B}} T) \quad \text{and} \quad P_{ji} = \frac{1}{d_j}.$$

Substituting these expressions into (12), we can easily verify that (12) is satisfied, after simplification. □