MULTISCALE MODEL. SIMUL.
Vol. 5, No. 2, pp. 393–419

COMPUTING BEST TRANSITION PATHWAYS IN HIGH-DIMENSIONAL DYNAMICAL SYSTEMS: APPLICATION TO THE $\alpha_L \leftrightharpoons \beta \leftrightharpoons \alpha_R$ TRANSITIONS IN OCTAALANINE*

FRANK NOÉ†, MARCUS OSWALD‡, GERHARD REINELT‡, STEFAN FISCHER§, AND JEREMY C. SMITH¶

Abstract. The direct computation of rare transitions in high-dimensional dynamical systems such as biomolecules via numerical integration or Monte Carlo is limited by the sampling problem. Alternatively, the dynamics of these systems can be modeled by transition networks (TNs) which are weighted graphs whose edges represent transitions between stable states of the system. The computation of the globally best transition paths connecting two selected stable states is straightforward with available graph-theoretical methods. However, these methods require that the energy barriers of all TN edges be determined, which is often computationally infeasible for large systems. Here, we introduce energy-bounded TNs, in which the transition barriers are specified in terms of lower and upper bounds. We present algorithms permitting the determination of the globally best paths on these TNs while requiring the computation of only a small subset of the true transition barriers. Several variants of the algorithm are given which achieve improved performance, including a parallel version. The effectiveness of the approach is demonstrated by various benchmarks on random TNs and by computing the refolding pathways of a polypeptide: the best transition pathways between the α_L helix, α_R helix, and β-hairpin conformations of the octaalanine (Ala_8) molecule in aqueous solution.

Key words. protein, peptide, transition network, pathway, polyalanine

AMS subject classification. 70

DOI. 10.1137/050641922

1. Introduction.

Complex dynamical systems with many degrees of freedom are ubiquitous. Examples include climate systems, stock markets, and condensed-phase molecular systems, among which biomolecules such as polypeptides, nucleic acids, or proteins are of particular interest. The immense number of possible states and state transitions poses a challenge to the simulation of these systems [1, 2, 3]. However, the qualitative and quantitative analysis of transitions between stable states is at the heart of understanding their dynamics [4, 5, 6, 7]. Here we present methods that are designed to enhance or enable the analysis of transitions between distant states in complex molecules. However, many of the principles described here are also applicable to nonmolecular systems.

Molecular dynamical systems are often modeled using a potential energy function $U(\mathbf{x}) : \mathbb{R}^D \to \mathbb{R}$, where D is the number of degrees of freedom of the system. Dynamical trajectories typically reside most of the time within the energy basins of

*Received by the editors October 4, 2005; accepted for publication (in revised form) February 13, 2006; published electronically June 30, 2006.
http://www.siam.org/journals/mms/5-2/64192.html

†Computational Molecular Biophysics Group and Computational Biochemistry Group, IWR, Universität Heidelberg, Im Neuenheimer Feld 368, 69120 Heidelberg, Germany (frank.noe@iwr.uni-heidelberg.de).

‡Discrete and Combinatorial Optimization Group, IWR, Universität Heidelberg, Im Neuenheimer Feld 368, 69120 Heidelberg, Germany (marcus.oswald@informatik.uni-heidelberg.de, gerhard.reinelt@informatik.uni-heidelberg.de).

§Corresponding author. Computational Biochemistry Group, IWR, Universität Heidelberg, Im Neuenheimer Feld 368, 69120 Heidelberg, Germany (stefan.fischer@iwr.uni-heidelberg.de).

¶Computational Molecular Biophysics Group, IWR, Universität Heidelberg, Im Neuenheimer Feld 368, 69120 Heidelberg, Germany (jeremy.smith@iwr.uni-heidelberg.de).

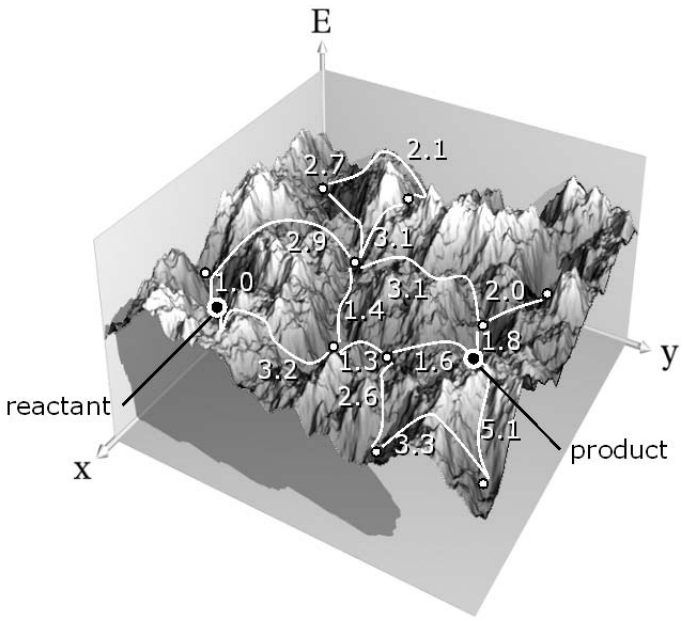

FIG. 1. *TN on a schematic two-dimensional energy surface. The network vertices (white bullets) correspond to low-energy intermediates between the reactant and product end-states of the transition (black bullets). The network edges (white lines) correspond to subtransitions between the vertices and are associated with the rate-limiting energy barriers along the subtransitions (white numbers).*

$U(\mathbf{x})$ and occasionally jump to neighboring basins [8]. The dynamics of the system can be simulated by numerically integrating the equations of motion involved. For stability, the integration time step must not exceed a value that depends on the fastest motions in the system and is often many orders of magnitude below the time scale during which the transitions of interest occur [9]. Larger steps are possible in Monte Carlo simulations but lead to a considerable reduction of the acceptance ratio [10]. These difficulties often lead to an insufficient number (if any) of occurrences of the transitions being investigated. Despite considerable progress in enhancing sampling methods [9, 11], this *sampling problem* is still the main obstacle to using direct simulation methods for the characterization of rare transitions.

An alternative approach to exploring $U(\mathbf{x})$ directly is to "map" its interesting features into a *transition network* (TN). A TN consists of vertices representing stable states (the energy basins of $U(\mathbf{x})$) and edges representing transition states connecting pairs of stable states (saddle regions of $U(\mathbf{x})$). Moreover, each vertex and each edge is assigned an energy. In the simplest case, these energies are potential energies of minima and saddle points, but ideally they are free energies of the corresponding states and transition states. Figure 1 shows a schematic representation of a TN on a potential energy surface. TNs have been constructed for various molecular systems [6, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26]. Free energy TNs have been shown to reproduce thermodynamic properties of the system [23, 24]. The kinetics between groups of states may be recovered using a master-equation dynamics [6, 13, 15, 18, 19, 20, 21, 22, 23, 24, 26, 27], kinetic Monte Carlo [26], or discrete path sampling [24, 26, 28].

Here we are interested in identifying the most populated transition pathways between two possibly distant system states (the transition end-states). In a related

study [29], a method was proposed to construct a contiguous transition pathway connecting possibly distant end-states. In this method, a graph-theoretic shortest-path algorithm is used to iteratively connect already-found minima by new minima on the potential energy surface, so as to yield a contiguous series of minima between the end-states. A transition pathway, which is short in terms of an Euclidean measure, is not necessarily highly populated but may serve as a starting point for a discrete path sampling procedure [28] which then identifies a physically meaningful ensemble of transition pathways. The advantage of this approach is that no a priori definition of the TN is required but rather is generated on the fly. The approach has a limitation in cases where multiple reaction channels (i.e., separate bundles of pathways) exist, as there is a high chance that not all of them will be sampled from a single starting pathway.

This danger of getting caught in locally optimal reaction channels can be avoided by approaching the task as a global optimization problem to find the k best paths on the network connecting two selected vertices. If the TN topology and energies are known, and a theory for transforming the energies into a *cost* for the transition along each edge is given, this problem can be solved with available graph-theoretical methods [30, 31]. In practice, the most difficult part is to construct an appropriate TN in the first place. Stable states can be identified by local optimization starting from conformational ensembles that are generated by high-temperature molecular dynamics [6, 16, 19, 21, 32], by parallel tempering methods [33, 34], or by direct manipulation of selected degrees of freedom. These stable states yield the TN vertices. Edges may be defined between all pairs of vertices within a certain cutoff distance d, yielding the network topology. In a reasonably dense network, the number of edges may be very large (e.g., $> 10^5$). As the determination of an edge energy requires a relatively CPU-intensive optimization of the transition state [35, 36] or calculation of the free energy barrier [37], it is usually infeasible to determine all edge energies of the TN. *How can one compute the k best paths while having only incomplete knowledge of the edge energies?* This question is in the focus of the present study.

The key to the solution lies in the definition of lower and upper bounds which bracket the true unknown edge energies. This concept is similar to a special case of fuzzy graphs, where it is possible to have a distribution of cost values associated with each edge instead of a concrete cost [38]. Methods are available to compute best paths in such fuzzy graphs [38, 39, 40], depending on the probability distribution of the individual edge costs. While this approach might also be helpful in the present context, the situation addressed in this paper has an important difference: In our case, the cost of a TN edge can be exactly identified (exact within the computational model), but this is very expensive. The question above can therefore be concretized as follows: *Given a TN with unknown edge energies and a set of lower and upper bounds on these energies, which edge energies need to be determined in order to quickly identify the best path (the k best paths)?* Here we present algorithms which address this problem and test their performance.

The rest of the article is organized as follows. In section 2, we formally introduce the concept of TNs with bounded edge energies and bounded edge costs. We also describe graph-theoretical methods for determining best paths in networks for which all edge energies are given. Furthermore, we describe methods for generating random TNs with certain desired properties, which are subsequently used for most of the benchmarks in this study. Section 3 presents an algorithm which computes the globally best path (and k best paths) on a TN with bounded energies by iteratively computing the energy of selected edges which are likely to lead to a quick determina-

tion of the best path. Section 4 presents methods to improve the performance of the algorithm. A parallel version is also given. Benchmark results on random TNs illustrate the performance of the algorithms presented. In section 5, we demonstrate the effectiveness of the approach by computing the refolding pathways of a biomolecule: the best transition pathways between the helical (α_L and α_R) and β-hairpin conformations of the octaalanine (Ala_8) polypeptide in an implicit solvent model.

2. TNs and k best paths.

2.1. Energy-bounded TNs. An energy-bounded TN is a weighted undirected graph $\mathcal{G} = (\mathcal{V}, \mathbf{E}^S, \mathcal{E}, \mathbf{E}^{TS,\min}, \mathbf{E}^{TS,\max})$. $\mathcal{V}$ is a list of vertices, representing system states whose energies are given by $\mathbf{E}^S$. $\mathcal{E}$ is an edge-list, defining which pairs of vertices are connected by a direct transition. $\mathbf{E}^{TS,\min}$ and $\mathbf{E}^{TS,\max}$ are the lower and upper bounds, respectively, of the transition states' energies.

The list of *vertices*, $\mathcal{V} = (1, \ldots, |\mathcal{V}|)$, represents states of the dynamical system. The number of states in the TN is equal to the size of this list, $|\mathcal{V}|$. Each state is associated with a region R of the system's configuration space. The precise definition of R depends on the application. A simple example for R is an attraction basin, i.e., the set of configurations that are mapped to the same local minimum by a direct minimization [1, 4].

The *vertex energy* vector, $\mathbf{E}^S = (E^S_1, \ldots, E^S_{|\mathcal{V}|})$, assigns an energy to each vertex in the order given by the list $\mathcal{V}$, ideally the free energy of region R. Depending on the application, different approximations to the free energy may be employed. In the simplest case, the entropy of region R is neglected and the energy is taken as the potential energy of an energy minimum in R.

The list of *edges*, $\mathcal{E}$, represents the transition states of the system. It is a list of pairs, each pair defining a connection between two vertices. For example, (u, v) corresponds to a connection between vertices u and v. As the TNs here are undirected, (u, v) and (v, u) refer to the same edge. Each edge appears only once in $\mathcal{E}$; i.e., the total number of edges in the TN is given by the size of the list, $|\mathcal{E}|$.

Each transition (u, v) can be associated with the free energy of the transition state, E^{TS}_{uv}. As the E^{TS}_{uv} are initially unknown, we instead use *lower* and *upper edge energy bounds*, $E^{TS,\min}_{uv}$ and $E^{TS,\max}_{uv}$, with $E^{TS,\min}_{uv} \le E^{TS}_{uv} \le E^{TS,\max}_{uv}$. The vectors $\mathbf{E}^{TS,\min}$ and $\mathbf{E}^{TS,\max}$ are of size $|\mathcal{E}|$ each and assign lower and upper edge energy bounds to each edge in the order given by the list $\mathcal{E}$.

The a priori values for the bounds of each edge (u, v) are given by

$$E^{TS,\min}_{uv} := \max\{E^S_u, E^S_v\}, \tag{1}$$

$$E^{TS,\max}_{uv} := \max\{E^S_u, E^S_v\} + M, \tag{2}$$

where M is a number that is larger than the anticipated maximum energy barrier.[1] In some cases, it is convenient to refer to relative barriers instead of absolute energies, so we define the *edge barrier* B_{uv}:[2]

$$B^{TS}_{uv} := E^{TS}_{uv} - \max\{E^S_u, E^S_v\}, \tag{3}$$

[1] M is used instead of ∞ because this avoids some numerical problems.

[2] B^{TS}_{uv} is to be distinguished from the usual definition of an energy barrier, which is the difference between transition state and reactant energies. Such a definition would produce two different barriers for each edge ($E^{TS}_{uv} - E^S_u$ and $E^{TS}_{uv} - E^S_v$).

such that (1) and (2) translate into equations for the *a priori barrier bounds*:

$$B_{uv}^{TS,\min} := 0, \tag{4}$$

$$B_{uv}^{TS,\max} := M. \tag{5}$$

An edge is said to be *undetermined* if its edge energy is unknown (i.e., if $E_{uv}^{TS,\min} < E_{uv}^{TS,\max}$ and $B_{uv}^{TS,\min} < B_{uv}^{TS,\max}$). It is said to be *determined* if its edge energy is known (i.e., $E_{uv}^{TS} = E_{uv}^{TS,\min} = E_{uv}^{TS,\max}$ and $B_{uv}^{TS} = B_{uv}^{TS,\min} = B_{uv}^{TS,\max}$).

2.2. Best paths in TNs. Here we derive a theory that allows us to transform the TN energies into edge costs and give methods that allow us to compute pathways of minimal cost through the network. For the present section, assume that all edge energies, E_{uv}^{TS}, are known and given by the vector $\mathbf{E}^{TS}$. Consider a particular transition between two vertices, $u \to v$, with energies E_u^S and E_{uv}^{TS}. We can express the transition rate from state u into state v, k_{uv}, as a product of the probability, p_u, to be in state u and the rate constant for the transition $u \to v$, k'_{uv}:

$$k_{uv} = p_u k'_{uv}. \tag{6}$$

Using free energies for the vertices, the probability p_u to be in vertex u, at equilibrium, is given by the ratio of partition functions for u and the full configuration space [41]:

$$p_u = \frac{\exp(-E_u^S/k_BT)}{\sum_{v=1}^{|\mathcal{V}|}\exp(-E_v^S/k_BT)}. \tag{7}$$

We assume that the local transition $u \to v$ can be modeled by a rate theory which takes an Arrhenius form [42] and, moreover, that all subtransitions have a similar dynamic prefactor, ν (including, e.g., k_BT/h and expressions for the friction or viscosity). Thus, the rate constant can be expressed as

$$k'_{uv} = \nu \exp\frac{-(E_{uv}^{TS} - E_u^S)}{k_BT}, \tag{8}$$

where k_B and h are Boltzmann and Planck constants, respectively, and T is the temperature. Substituting (7) and (8) into (6), we see that the equilibrium flux for the transition $u \to v$ is proportional to the Boltzmann weight of E_{uv}^{TS}:

$$k_{uv} = \frac{\nu}{\sum_{v=1}^{|\mathcal{V}|}\exp(-E_v^S/k_BT)} \exp\left(-\frac{E_{uv}^{TS}}{k_BT}\right). \tag{9}$$

The expected mean time, τ_{uv}, between two subsequent transition events from u to v is the inverse of the rate: $\tau_{uv} = k_{uv}^{-1}$. We define the *edge costs*, c_{uv}, as the normalized τ_{uv} which are obtained by setting the constant rate factor to unity. They are therefore equal to the inverse Boltzmann weight of the edge energies:

$$c_{uv} = \exp\left(\frac{E_{uv}^{TS}}{k_BT}\right). \tag{10}$$

The best path connecting vertices v_1 and v_m, $P = (v_1, \ldots, v_m)$, is one which minimizes the cumulative edge cost:

$$C(P) = \sum_{k=1}^{m-1} c_{v_k v_{k+1}}. \tag{11}$$

This definition of a best path is similar to notion of the path with the "maximum flux" or "minimum resistance," as given in [43, 44]. The main difference is that the minimum resistance path is defined as a continuous line integral connecting the end-states on the potential energy surface, while (11) is a discrete sum over a contiguous series of transition states, connecting the end-states on a free energy surface.

To determine the best path in practice, the edge energy vector $\mathbf{E}^{TS}$ is transformed into a cost vector $\mathbf{c}$ using (10). $\mathbf{c}$ has size $|\mathcal{E}|$ and assigns a cost c_{uv} to each edge (u, v) in $\mathcal{E}$. Then the Dijkstra algorithm [30] is used to identify a best path between the two end-states through a weighted network defined by $(\mathcal{V}, \mathcal{E}, \mathbf{c})$. Such a path minimizes the path cost given in (11).

Because of the exponential weighting of energies in C, the best path tends to be one that minimizes the highest barrier along the path; i.e., it optimizes the rate-limiting step. A single best path dominates the transition only if the barriers of alternative pathways are considerably higher. However, the best path furnishes a preliminary understanding of the transition [45] or may be used as a guess for a reaction coordinate to obtain a free energy profile along the transition [46]. To obtain a better representation of the ensemble of accessible pathways, it is useful to compute the k best pathways $(P_1, P_2, \ldots, P_k)$ with path costs $(C_1 \leq C_2 \leq \cdots \leq C_k)$. "$k$ best path" problems are well established in graph theory [31] and vary in the criterion by which paths are treated as "different." For molecular systems, it is useful to distinguish paths which include different rate-limiting steps. Therefore, here two paths are treated as different only if they do not coincide in the highest-energy edge. The k best paths are determined in k steps, using the following simple protocol: The ith best path is given by computing the best path while "blocking" all edges (u, v) associated with the highest energy barrier along each of the $(i - 1)$ previously found best paths by setting their $B_{uv}^{TS} = M$.

When the best path is recomputed after changing only a single edge energy, it is not necessary to repeat the whole Dijkstra algorithm from scratch. Rather, the algorithms of Frigioni and Marchetti-Spaccamela [47] are used to dynamically update the shortest path tree and thereby recompute the next best path with minimal effort.

2.3. Cost bounds and best paths. To compute best paths (and k best paths) in energy-bounded TNs, we must take into account that there is no unique energy vector $\mathbf{E}^{TS}$ given but lower bounds $\mathbf{E}^{TS,\min}$ and upper bounds $\mathbf{E}^{TS,\max}$. Therefore, we transform $\mathbf{E}^{TS,\min}$ into a vector of lower cost bounds $\mathbf{c}^{\min}$ and $\mathbf{E}^{TS,\max}$ into a vector of upper cost bounds $\mathbf{c}^{\max}$, as specified in section 2.2. As c_{uv} increases monotonously with E_{uv}^{TS} (see (10)), $c_{uv}^{\min}$ and $c_{uv}^{\max}$ are bounds for the unknown true edge cost: $c_{uv}^{\min} \leq c_{uv} \leq c_{uv}^{\max}$.

For a given TN $\mathcal{G}$, we can compute two different best paths: an "optimistic" best path, $P^{\min}$, using the minimum edge costs, $\mathbf{c}^{\min}$, in (11) and a "pessimistic" best path, $P^{\max}$, using the maximum edge costs $\mathbf{c}^{\max}$. Obviously, if all edges are determined $(\mathbf{c}^{\min} = \mathbf{c}^{\max})$, then $P^{\min}$ and $P^{\max}$ are identical, and the true best path is identified.

The idea of Algorithm 3 in section 3.1 is to determine only a small subset of edges and still identify the best path (which involves that $P^{\min} = P^{\max}$). Whenever an edge (u, v) is determined, and thereby $E_{uv}^{TS,\min}$ and $E_{uv}^{TS,\max}$ are modified, the corresponding edge cost bounds, $c_{uv}^{\min}$ and $c_{uv}^{\max}$, must be updated.

2.4. Random TNs. For testing the algorithms presented in this paper, it is useful to have a model that generates random TNs. Depending on the underlying physical system, TNs can have various different topologies and edge energies. However, TNs for molecules are not well represented by purely random graphs [48] with

random energies, the reason being that they have a number of special properties which are discussed in the present section. Below we give the details of how to generate the random TNs used in this study.

Embedding. The TN vertices are embedded in a D-dimensional space. Typically, the degrees of freedom of the molecule are strongly correlated. Therefore, the system configurations mostly reside in an essential subspace with an effective dimensionality much lower than D. For example, it has been shown that, for proteins with many thousands degrees of freedom, less than 1% of the degrees of freedom is sufficient to cover most of the variance in the atomic motions [49]. Therefore, the random TN vertices are embedded in a ($C \ll D$)-dimensional space.

For the vertices of all random TNs in this study we choose a $C = 5$-dimensional hypercubic space. Our application system (section 5) is a peptide whose conformation is characterized by a number of internal torsion angles. To achieve a similar coordinate system for random TNs, each dimension of the embedding space has values in $]-180, 180]$ degrees and periodic boundary conditions. Initially, all $|\mathcal{V}|$ vertices are embedded as random points in this space.

Connectivity. Given the positions of the vertices, the connectivity of the network is specified by defining an edge between all vertex pairs within a distance, $d_{\max}$. Here a root mean square distance of 40° is used; i.e., for two vertices (u, v) to be connected by an edge it must hold that

$$d_{\max} = \sqrt{\frac{\sum_{i=1}^{C} d(\theta_{u,i}, \theta_{v,i})^2}{C}} < 40^\circ, \tag{12}$$

where $\mathbf{x}_u = (\theta_{u,1}, \ldots, \theta_{u,C})$ and $\mathbf{v}_u = (\theta_{v,1}, \ldots, \theta_{v,C})$ are the embedding coordinates of vertices u and v, respectively, and $d(\theta_{u,i}, \theta_{v,i})$ is the minimal difference between the two angles $\theta_{u,i}$ and $\theta_{v,i}$. A common measure for the connectivity of the network is its degree distribution $p(k)$: The degree of a vertex, k, is its number of neighbors; the degree distribution is the distribution of all vertex degrees in the network. For a random TN to be representative, we require it to have a degree distribution that is typical for the class of physical systems of interest.

To obtain a random network with a predefined degree distribution $p_{\text{ref}}(k)$, we use the following Monte Carlo algorithm which modifies the initial vertex embedding until the desired distribution $p_{\text{ref}}(k)$ is obtained.

Algorithm 1 (random TN embedding).

(1) Given an initial vertex embedding $(\mathbf{x}_1, \ldots, \mathbf{x}_{|\mathcal{V}|})$, compute the neighborhoods for each vertex and from this the degree distribution $p(k)$. Compute the distribution error $\epsilon_{p(k)} = \sum_{k=0}^{\infty} (p(k) - p_{\text{ref}}(k))^2$.
(2) Set $i := 0$. While $i < i_{\max}$ and $\epsilon_{p(k)} > \epsilon_{\text{tol}}$, repeat:
  (2.1) Randomly choose a vertex v with embedding $\mathbf{x}_v$ and randomly choose a new point $\mathbf{x}'_v$ in the embedding space.
  (2.2) Compute the degree distribution $p(k)'$ for the case that v is moved to $\mathbf{x}'_v$ and the distribution error $\epsilon_{p(k)'}$.
  (2.3) If $\epsilon_{p(k)'} < \epsilon_{p(k)}$, accept move:
  $\mathbf{x}_v := \mathbf{x}'_v$, $p(k) := p(k)'$, $\epsilon_{p(k)} := \epsilon_{p(k)'}$.

The algorithm terminates when the maximum allowed error in the distribution, ϵ_{tol}, or the maximum number of iterations, $i_{\max}$, is reached (here $\epsilon_{\text{tol}} = 0.01$ and $i_{\max} = 10^4$). The degree distribution of the octaalanine TN analyzed in section 5 is a Poisson distribution that can be fitted by a Gaussian with mean $\mu = 12.4$ and a

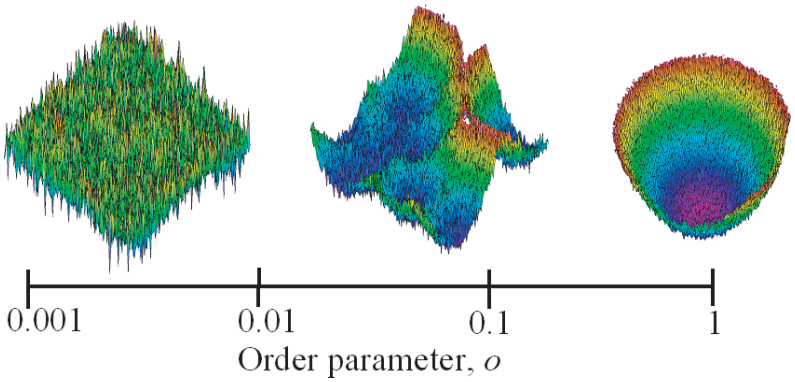

FIG. 2. *Illustration of the effect of the order parameter, o, on the energy surface underlying random networks. Here an energy surface defined over two coordinates, $E(x, y)$, is shown. A minimum value of $o = 1/|\mathcal{V}|$ (here 0.001) corresponds to a random noise surface, while $o = 1$ represents a harmonic surface with some local roughness.*

standard deviation $\sigma = 6.1$. This is used as reference degree distribution, $p_{\text{ref}}(k)$, for all random TNs in this study.

TN energies. In TNs, the energies correspond to the form of the energy surface of the underlying physical system. Two extreme cases are when (a) the energy surface has the global form of one large basin (with some roughness) and (b) the energy surface has no underlying form, where the energies are just uncorrelated random numbers. Between these two extremes are the cases in which the energy surface has some local form: (c) the energy surface has a number of basins (with some roughness), which are mutually connected. Figure 2 illustrates these cases.

To simulate different energy surfaces, we use following method to assign edge energies, which can be controlled by an order parameter o to switch between cases (a), (b), and (c). The algorithm selects $n_b = 1/o$ vertices as centers for harmonic energy basins. Next, for each vertex v, the distance d to the closest basin center (the number of edges in the shortest path) is computed. Vertex v is treated as a part of that closest basin; i.e., its energy is set to $E_v^S = d^2$. To account for the local roughness of the energy surface, a random value is added to each vertex energy. Each edge barrier B_{uv}^{TS} is simply given by a positive random number.

ALGORITHM 2 (random TN).

(1) Generate TN topology according to $p_{\text{ref}}(k)$ (Algorithm 1).
(2) Randomly select a set of $n_b = 1/o$ different vertices, which act as harmonic basin centers.
(3) For each vertex, $v \in \mathcal{V}$:
  (3.1) Set $d :=$ number of edges on shortest path to nearest basin center.
  (3.2) Set $E_v^S := d^2 + \text{Gaussian}(0, \sigma)$.
(4) For each edge, $(u, v) \in \mathcal{E}$:
  (4.1) Set $B_{uv}^{TS} := |\text{Gaussian}(0, \sigma)|$.

Here $\text{Gaussian}(\mu, \sigma)$ generates a random value drawn from a Gaussian distribution with mean μ and standard deviation σ. The order parameter o, which is the inverse of the number of basins, n_b, quantifies the amount of order on the potential energy surface. For $o = 1$ ($n_b = 1$), we have a single harmonic basin (case (a)). For $o = |\mathcal{V}|^{-1}$ ($n_b = |\mathcal{V}|$), the order is minimal, as all energies are determined by random values. This is the *random noise network* (case (b)). Order values between these extremes

are associated with energy surfaces having some local form (case (c)). Unless stated otherwise, $\sigma = 1$ kcal/mol and $o = |\mathcal{V}|^{-1} = 0.001$ are used in this study. Section 3.2 treats the effects of using different values of o.

In this study, best path(s) are computed on random TNs. This requires the definition of a pair of end-states for each computation. As transition end-states usually are in the energy minima of the end-state basins, the pair of end-states was not selected in a completely random way. Rather, two random vertices were chosen and then both were minimized on the network; i.e., each end-state was repeatedly moved to the lowest-energy neighboring vertex until no neighboring vertices had a lower energy. If, after this minimization, both vertices coincided, the process was repeated until two distinct end-states were found.

3. Efficient computation of best paths. We propose an iterative algorithm to compute the best path in an energy-bounded TN that requires the determination of only a small number of edge energies.

3.1. Algorithm. Given a TN, $\mathcal{G} = (\mathcal{V}, \mathbf{E}^S, \mathcal{E}, \mathbf{E}^{TS,\min}, \mathbf{E}^{TS,\max})$, whose edge energies are bounded by $\mathbf{E}^{TS,\min}$ and $\mathbf{E}^{TS,\max}$, the following algorithm iteratively determines the best path through the network. For this the "optimistic" best path $P^{\min}$ is identified using the minimum edge costs $c^{\min}$ (see section 2.3). A "critical edge," (u, v), is identified, defined as the highest-energy undetermined edge along $P^{\min}$. Then the CPU-intensive step is performed by determining the real energy E_{uv}^{TS}. We set $E_{uv}^{TS,\min} = E_{uv}^{TS,\max} := E_{uv}^{TS}$ and update $c_{uv}^{\min}$ and $c_{uv}^{\max}$. This may lead to a different $P^{\min}$ in the next iteration. These steps are repeated until all edge energies along the optimistic best path are determined, giving the truly best path.

To obtain a preliminary estimate of the best path's energy barrier, the same computations can be performed using the maximum edge costs $\mathbf{c}^{\max}$, yielding the "pessimistic" best path $P^{\max}$. The rate-limiting barrier of the true best path is bounded by those of $P^{\min}$ and $P^{\max}$. During successive iterations, these bounds converge to the true value (see Figure 3).

The following pseudocode gives a formal representation of the algorithm, while Figure 4 shows a graphical illustration.

ALGORITHM 3 (best path).

(1) Compute two best paths: $P^{\min}$ using $\mathbf{c}^{\min}$ and $P^{\max}$ using $\mathbf{c}^{\max}$ between the transition end-states.
(2) If all edge energies along $P^{\min}$ are determined (i.e., $E_{uv}^{TS,\min} = E_{uv}^{TS,\max}\ \forall$ (u, v) along $P^{\min}$), RETURN($P^{\min}$).
(3) Choose from $P^{\min}$ the edge (u, v) with $E_{uv}^{TS,\min} = \max\{E_{wy}^{TS,\min} | ((w, y)$ edge along $P^{\min}) \wedge (E_{wy}^{TS,\min} < E_{wy}^{TS,\max})\}$ (critical edge with undetermined energy). If (u, v) exists: Determine (u, v).
(4) GOTO 1.

Termination: Determined edges are never selected as critical edges in step (3); therefore each edge can be determined only once. At last, when all edges are determined, the algorithm returns in step (2). The algorithm thus terminates after at most $|E|$ cycles.

Correctness: The returned path is the globally best path after termination.

Assume the algorithm returns the path P_{ret}, but the real best path is $P \neq P_{ret}$ with $C(P) < C(P_{ret})$.

Case 1: All edge energies of P are determined. Then P is computed as the best path in step (1), and it must be $P \equiv P^{\min} \equiv P^{\max}$ in step (2), and therefore $P \equiv P_{ret}$ (a contradiction).

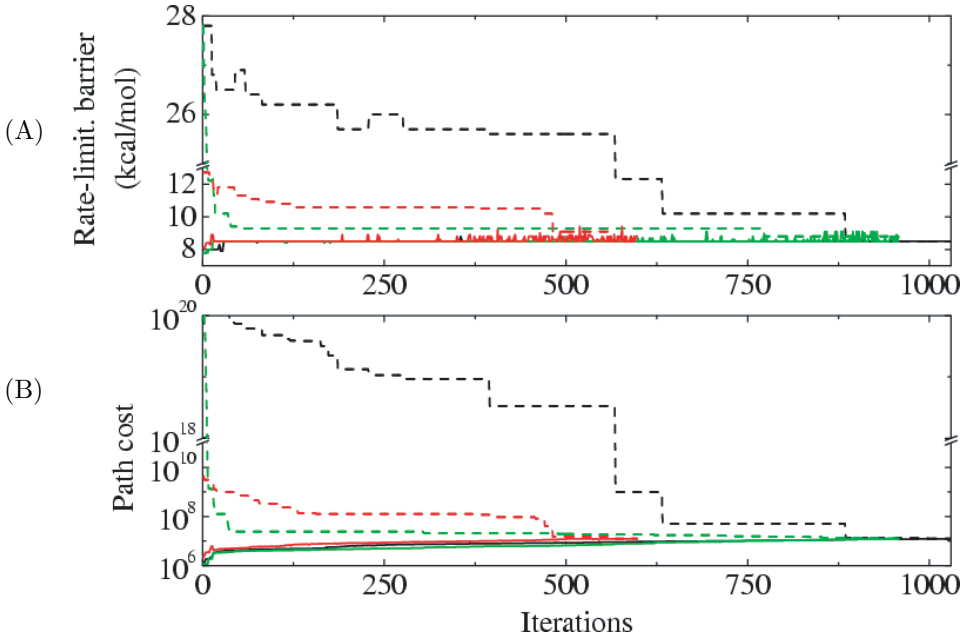

FIG. 3. *Convergence behavior of the best-path algorithm on a random TN, using different settings. The algorithm determines an "optimistic" and a "pessimistic" guess of the best path in each iteration, allowing the derivation of a lower and an upper bound for* (A) *the highest energy barrier of the true best path and* (B) *its path cost. These optimistic (solid lines) and pessimistic (dashed lines) pairs of values are equal when the true best path is determined. If a priori bounds* $[0, M]$ *are used on the edge energy barriers (black lines), the upper bound for the path energy and the path cost are very far from their true values for many iterations but then quickly converge when any full pathway between the transition end-states has been determined on* $\mathcal{G}^{\max}$ *(i.e., when the current pessimistic hypothesis of the best path does not contain any uncomputed barriers with height* M*). If the values of edge energies are refined in two steps (green lines, discussed in section* 4.3*) rather than one, the initial convergence is much faster, because the first step of refinement is first performed on all edges along the current best-path hypothesis before the CPU-intensive full determinations of the edge energies begin. The use of statistical estimates for the edge energy bounds (red lines, discussed in section* 4.1*) also allows the lower and upper estimates for the best-path energy and cost to converge faster and may significantly speed up the computation.*

Case 2: Not all edge energies of P are determined. Let $\{E^{TS}_{1,2}, E^{TS}_{2,3}, \ldots, E^{TS}_{m-1,m}\}$ be the (unknown) true edge energies of the edges along P. Since the $\mathbf{E}^{TS,\min}$ are lower bounds to the edge energies ($E^{TS,\min}_{uv} \leq E^{TS}_{uv}\ \forall\ (u,v)$), the path cost of $P^{\min}$ in step (1) is always less than or equal to the true cost of P, in particular in the last iteration, in which P_{ret} is returned. With the above assumption $C(P) < C(P_{ret})$ it follows that $C(P^{\min}) < C(P_{ret})$, which means that $P^{\min} \neq P_{ret}$. However, in step (2) $P^{\min}$ is returned, so $P^{\min} \equiv P_{ret}$ (a contradiction). □

3.2. Performance. The CPU time needed for the best-path algorithm is dominated by the determination of the edge energies. Therefore, to evaluate the performance of the algorithm, we calculate the number, n_{ec}, of determined edges necessary to determine the best path.

Approximate upper bound for n_{ec}. We first derive an approximate upper bound for n_{ec}. Consider a pathway, P. The cost of this path is likely to be dominated by the highest edge energy along P as a result of the exponential weighting in (10). We can therefore formulate following proposition, which claims that the costs of different

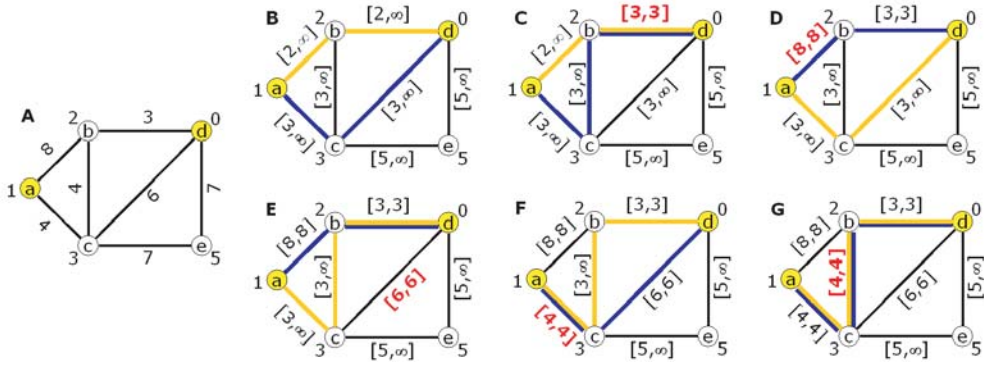

FIG. 4. *Illustration of Algorithm* 3 *which determines the best path through an energy-bounded TN.* A. *The TN with its (initially unknown) true edge energies.* B. *Each edge is given lower and upper bounds for the edge energies (here a priori bounds are shown). Best paths are computed between the end-states* (a) *and* (d) *using the optimistic (yellow) and pessimistic bounds (blue). The critical edge* (b)-(d) *is the highest-energy undetermined edge along the optimistic path. Its true energy is determined.* C–G. *The process is iteratively repeated until all edge energies along the optimistic best path are determined. Both the optimistic and pessimistic best path coincide with the true best path.*

pathways can be approximately ordered by their highest edge energies.

PROPOSITION. *For any two pathways,* $P_1 = (u_1, \ldots, u_m)$ *and* $P_2 = (v_1, \ldots, v_n)$*, along series of edges with edge energies* $(E^{TS}_{u_1u_2}, \ldots, E^{TS}_{u_{m-1}u_m})$ *and* $(E^{TS}_{v_1v_2}, \ldots, E^{TS}_{v_{n-1}v_n})$*, respectively, it holds that*

$$\max\{E^{TS}_{u_k u_{k+1}} | k \in \{1, \ldots, m-1\}\} < \max\{E^{TS}_{v_k v_{k+1}} | k \in \{1, \ldots, n-1\}\} \Rightarrow C(P_1) \lessapprox C(P_2).$$

Assuming this proposition is exact rather than approximate, then edges (u, v) with $E^{TS,\min}_{uv} > E_{\text{peak}}$ are never refined, where E_{peak} is the highest edge energy of the best path.[3] Thus, n_{ec} is approximately bounded from above by the number of low-energy edges, n_{low}:

$$n_{ec} \lessapprox n_{\text{low}} = |\{(u, v) \in \mathcal{E} | E^{TS,\min}_{uv} \leq E_{\text{peak}}\}|. \tag{13}$$

This upper bound is only approximate because the above proposition is itself only approximate. However, not a single case with $n_{ec} > n_{\text{low}}$ was observed in the present simulations.

In most cases, it holds that $n_{ec} < n_{\text{low}}$, as it is not necessary that all edges with $E^{TS,\min}_{uv} \leq E_{\text{peak}}$ be computed. Some edges may lie in regions of the network which are separated from the transition end-states by edges with $E^{TS,\min}_{uv} > E_{\text{peak}}$ and are therefore never considered by the algorithm.

[3] *Proof.* In each iteration of Algorithm 3, $P^{\min}$ is computed using the costs obtained from the lower energy bounds, $E^{TS,\min}_{uv}$. As an edge determination may only increase but never decrease $E^{TS,\min}_{uv}$, the next iteration's $P^{\min}$ is guaranteed to have an equal or higher cost than the current one. Therefore, the costs of $P^{\min}$ are nondecreasing until termination. If the above proposition holds exactly, i.e., if the different pathways would be exactly ordered in the same way by cost or by maximum edge energy, then the maximum edge energies of $P^{\min}$ are also nondecreasing until termination. When the algorithm terminates, the maximum edge energy of $P^{\min} \equiv P^{\max}$ equals, by definition, E_{peak}. This is then the highest energy edge that was refined. ∎

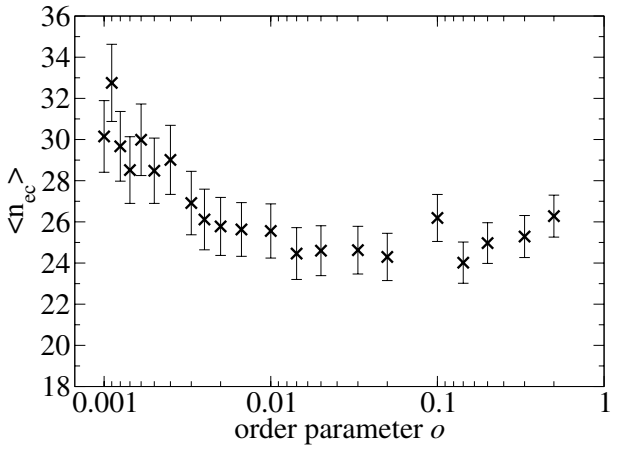

FIG. 5. *Dependence of the average number of computed edges, $\langle n_{ec} \rangle$, on the amount of order on the energy surface. $o = 0.5$ represents two large harmonic wells with some superimposed noise, while $o = 0.001$ represents the random noise network (see also Figure 2).*

n_{ec} varies strongly depending on the topology of the network, on its edge energies, and on the location of the two end-state vertices. The performance of the best-path algorithm was evaluated on an ensemble of random networks and measured in terms of the average number of edges computed to determine the best path(s), $\langle n_{ec} \rangle$.

Evaluating the average number of computed edges, $\langle n_{ec} \rangle$, on random TNs. Here we describe how to evaluate $\langle n_{ec} \rangle$ on random networks. This forms the basis for all benchmarks given in sections 3 and 4.

For each calculated value of $\langle n_{ec} \rangle$, 50 random networks with $|\mathcal{V}| = 1000$ vertices were generated as described in section 2.4. For each of these networks, 50 pairs of end-states were randomly chosen as described in section 2.4, yielding 2500 random setups. The a priori barrier bounds $B_{uv}^{TS,\min} = 0$ and $B_{uv}^{TS,\max} = 5$ kcal/mol were used. As a standard deviation of $\sigma = 1$ kcal/mol was used in Algorithm 2 to generate the edge barriers, above bounds were valid for the vast majority of edges. The best paths between the selected end-states were determined using Algorithm 3, giving 2500 values for n_{ec}. From this ensemble we obtain the mean value $\langle n_{ec} \rangle$ and the standard deviation $\sigma(n_{ec})$. The error bars shown in Figures 5, 7, 8, and 10 are given by the values $\langle n_{ec} \rangle \pm 2\sigma(n_{ec})$.

We first tested how the form of the underlying energy surface influences the performance of the best-path algorithm by computing $\langle n_{ec} \rangle$ for random networks with different order parameter values o. These results are shown in Figure 5. For random noise networks, $\langle n_{ec} \rangle$ has a maximum value, while for networks with some local structure, $\langle n_{ec} \rangle$ decreases significantly, approaching a constant value ($o > 0.01$). This result is expected, as random noise networks contain many more pathways of similar energies than TNs with more order. The random noise network is therefore a worst-case scenario. It is used as a model for all computations of $\langle n_{ec} \rangle$ in sections 3 and 4, unless stated otherwise.

Figure 6 shows a correlation of n_{ec} with n_{low} for all 2500 best-path computations on random noise networks. Most values of n_{ec} are about 2 orders of magnitude below the approximate upper bound $n_{ec} = n_{\text{low}}$. The average number of computed edges, $\langle n_{ec} \rangle$, increases linearly with n_{low} for large values of n_{low}.

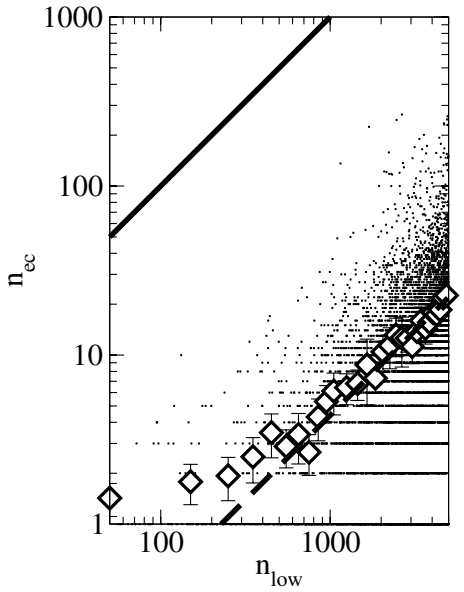

FIG. 6. *Correlation between the number of computed edges, n_{ec}, for best paths on random networks with the number of low-energy edges, n_{low} (defined in* (13)*). The black dots show all value pairs generated by simulations on random noise networks. All data points are well below the approximate upper bound of $n_{ec} = n_{\mathrm{low}}$ (solid line). The average number of computed edges, $\langle n_{ec} \rangle$ (diamonds), computed for several windows of n_{low} approaches $\langle n_{ec} \rangle = 0.0044\, n_{\mathrm{low}}$ (dashed line).*

3.3. *k* best paths. The best-path Algorithm 3 can be used directly to compute multiple, k best paths if the protocol described in section 2.2 is followed. Typically, there is an overlap between the sets of edges which need to be determined to compute each of the k best paths individually. Therefore, the number of edges required to compute k best paths, $\langle n_{ec,k} \rangle$, is expected to be less than k times $\langle n_{ec,1} \rangle$, where $n_{ec,1}$ is the number of edges required to compute the best path. This is indeed visible in Figure 7, which shows $\langle n_{ec,k} \rangle$ that has been computed for values of $k \leq 16$. $\langle n_{ec,k} \rangle$ increases linearly with k for the numbers of $\langle n_{ec,k} \rangle$ observed here,[4] approaching the function $\langle n_{ec,k} \rangle = 0.78\, k \langle n_{ec,1} \rangle$.

4. Improving performance.

4.1. Statistical edge energy estimates. The computing time can be drastically reduced, at the expense of possibly failing to identify the true best path, if the a priori bounds for the edge barriers $B_{uv}^{TS,\min} = 0$ and $B_{uv}^{TS,\max} = M$ are replaced by statistical estimates. Such estimates can be obtained if a reasonable number of edge energies is already known, such that a probability distribution for energy barriers can be proposed. If a lower-energy estimate $B_{uv}^{TS,\min} > 0$ is used, the problem exists that it is not necessarily a true lower bound and may therefore overestimate some barriers. That is, edges which are not included in the resulting best path and have been rejected based on their lower estimate $B_{uv}^{TS,\min}$ might in fact have a true barrier $B_{uv}^{TS} < B_{uv}^{TS,\min}$. The maximum overestimation possible, $\mathrm{err}_{\max}$, is given by

[4] $\langle n_{ec,k} \rangle$ is, of course, bounded from above by $|\mathcal{E}|$, and can therefore not continue to rise linearly, but the present simulation values for $n_{ec,k}$ do not come close to $|\mathcal{E}|$.

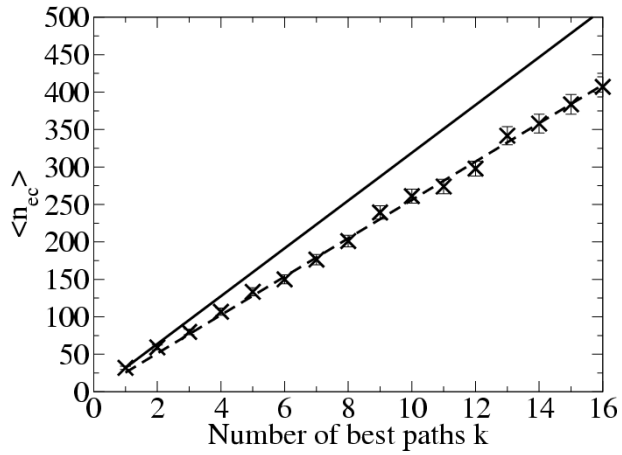

FIG. 7. *Dependence of the average number of computed edges for k best paths on random networks, $\langle n_{ec,k}\rangle$, on the number k of best paths that were computed. $\langle n_{ec,k}\rangle$ increases slower than the theoretical maximum $\langle n_{ec,k}\rangle = k\langle n_{ec,1}\rangle$ (solid line), approaching the function $\langle n_{ec,k}\rangle = 0.78\, k\langle n_{ec,1}\rangle$.*

the maximum lower barrier bound:

$$\mathrm{err}_{\max} = \max\{B_{uv}^{TS,\min} | (u,v) \in \mathcal{E}\}. \tag{14}$$

Thus, $\mathrm{err}_{\max}$ also gives the maximum possible error on the rate-limiting barrier of the path.

To give benchmarks for the efficiency of statistical estimates, we define a confidence ratio, r_{conf}: For each random network, the lower estimate $B^{TS,\min}$ was set such that $r_{conf}|\mathcal{E}|$ edge energies were greater than $B^{TS,\min}$, and likewise the upper estimate $B^{TS,\max}$ was set such that $r_{conf}|\mathcal{E}|$ edge energies were smaller than $B^{TS,\max}$ (i.e., r_{conf} is the fraction of correctly estimated bounds). Figure 8 shows the dependence of $\langle n_{ec}\rangle$ for computing the (estimated) best path in random noise networks on r_{conf}. Comparing with Figure 5, which shows $\langle n_{ec}\rangle$ for best-path computations with a priori bounds, reveals that even a maximum confidence ratio, $r_{conf} = 1$, gave $\langle n_{ec}\rangle \approx 15$, which is only about half the runtime compared to the value of $\langle n_{ec}\rangle \approx 30$ with a priori bounds. In general, $r_{conf} = 1$ does not guarantee that $B^{TS,\min}$ are true lower bounds for the barrier. This is because usually only a small subset of all barrier energies is used to set up the barrier statistics, such that $r_{conf} = 1$ means only that $B^{TS,\min}$ and $B^{TS,\max}$ are true bounds for all *observed* barriers. However, the above results show that a statistical estimate involving only a small potential error can save a considerable amount of CPU time. Smaller values of r_{conf} can further reduce $\langle n_{ec}\rangle$ by a factor of three.

4.2. Partial computation of best paths. So far, we assumed that the best path is computed in its full detail. However, one is often not interested in the details of how the best path travels in the low-energy regions, since the highest-energy edges along the whole path are rate-determining. Computation time can thus be saved if only the high-energy edges of the best path, i.e., those with energies within ΔE_{sure} of the highest energy along the path, E_{peak}, are requested to be correct (see Figure 9). To achieve this, Algorithm 3 is extended as follows: The computation proceeds as above until the energy of the highest energy barrier along the best path, E_{peak}, is identified

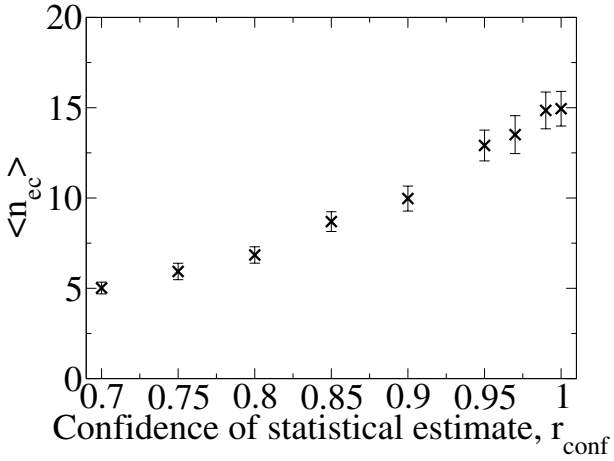

FIG. 8. *Dependence of the average number of computed edges, $\langle n_{ec}\rangle$, for best paths on random noise networks, on the confidence of statistically estimated edge energy bounds. Even a confidence value of $r_{\mathrm{conf}} = 1$ (energy bounds are true bounds for all observed edge energies) allows a factor of two to be saved compared to the use of a priori edge energy bounds (compare with Figure 5).*

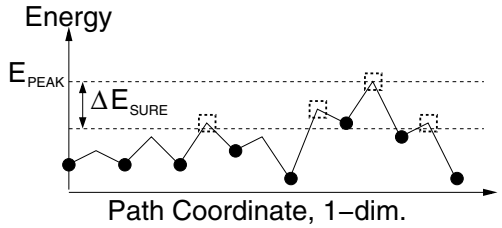

FIG. 9. *Schematic representation of the concept of partial determination of best paths. A profile of vertex and edge energies along a pathway through the network is shown. Given ΔE_{sure}, the edges in the range $[E_{\mathrm{peak}} - \Delta E_{\mathrm{sure}}, E_{\mathrm{peak}}]$ are guaranteed to belong to the true best path (indicated by squares).*

(this is the case when $P^{\min}$ and $P^{\max}$ agree in this energy). From this moment on, whenever a barrier with energy $E_{uv}^{TS} < E_{\mathrm{peak}} - \Delta E_{\mathrm{sure}}$ is computed, we update the edge energies by setting $E_{uv}^{TS,\min} = E_{uv}^{TS,\max} := \max\{E_u^S, E_v^S\}$, i.e., as if the transition was barrierless. This prevents the next Dijkstra computation from finding a different best path that would circumvent the low-energy barriers of the current best path.

To evaluate the effect of using $\Delta E_{\mathrm{sure}} < \infty$ on random TNs, we must take into account that the values of E_{peak} for different random TN setups can be very different. We therefore define the ratio

$$r_{\mathrm{sure}} := \frac{\Delta E_{\mathrm{sure}}}{E_{\mathrm{peak}} - E_{\min}}, \tag{15}$$

where $E_{\min}$ is the lowest vertex energy in the path. r_{sure} is therefore the "fraction" of the best path that is determined (on the energy scale). Figure 10A shows that using $r_{\mathrm{sure}} < 1$ can considerably reduce $\langle n_{ec}\rangle$. The amount of reduction, however, depends on the amount of order on the energy surface. For random noise networks $\langle n_{ec}\rangle$ is reduced only by a small fraction when only the highest barrier along the path is identified ($r_{\mathrm{sure}} = 0$) instead of the full path ($r_{\mathrm{sure}} = 1$). The reduction amounts to more than 50% for networks with $o = 0.2$. The reason for this difference is that

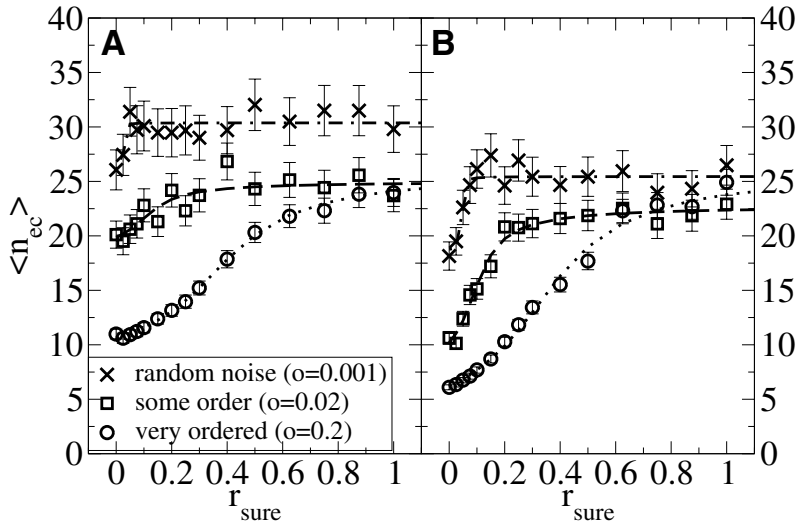

FIG. 10. *Dependence of the average number of computed edges, $\langle n_{ec}\rangle$, for best paths on random networks, on the fraction of the best path that has been determined, r_{sure} (defined in* (15)*).* A. *Edge energies refined in a single step.* B. *Edge energies refined in two steps (discussed in section* 4.3*). After the first step the upper bound was* 0.5 *above the true edge energy. A partial computation of the best path ($r_{\text{sure}} < 1$) performs better with well-ordered energy surfaces (bullets, squares) than with random noise networks (crosses). A two-step refinement enhances the savings at low values of r_{sure}.*

for energy surfaces with a significant amount of underlying order the edge energy bounds already give a good estimate of where the highest edge of the best path may be located. In contrast, for random noise networks, nearly all edges are candidates for these highest edges, so that the reduction of $\langle n_{ec}\rangle$ is small compared with a full computation of the best path.

4.3. Stepwise determination of edge energies. In the previous sections we always assumed that edges are determined in a single step: In step (3) of Algorithm 3, a program was called to compute the edge energy, E_{uv}^{TS}, and the initial bounds were immediately replaced: $E_{uv}^{TS,\min} := E_{uv}^{TS}$ and $E_{uv}^{TS,\max} := E_{uv}^{TS}$. Now we will consider that an edge is only *refined* in step (3) of Algorithm 3; i.e., we call a program which computes some information on the transition $u \to v$ that provides tighter energy bounds ($E_{uv}^{TS,\min}$ is increased and $E_{uv}^{TS,\max}$ is decreased). To guarantee the termination of the algorithm, we require that only a fixed number of refinements of the same edge are necessary to determine the edge, i.e., to deliver bounds that are equal to E_{uv}^{TS}.

To simulate this approach, we used a two-step refinement algorithm: The first refinement changes the bounds $[E_{uv}^{TS,\min}, E_{uv}^{TS,\max}]$ into $[E_{uv}^{TS,\min}, E_{uv}^{TS} + 0.5\,\text{kcal/mol}]$, and the second refinement delivers the determined edge energy $[E_{uv}^{TS}, E_{uv}^{TS}]$. The motivation for this choice is that the program we use here for the determination of energy barriers in the application (section 5) approaches the transition state energy from above; i.e., a preliminary computation can be used to improve the upper bound. To take advantage of the improved upper bounds, we must modify the criterion by which the critical edge is identified in Algorithm 3. Here we chose the critical edge to be an undetermined edge in $P^{\min}$ which has the highest upper energy bound $E_{uv}^{TS,\max}$, instead of the highest lower energy bound.

Such a multistep determination approach alone does not guarantee finding the

best path faster (i.e., to reduce n_{ec}). However, it considerably improves the lower and upper bounds for the energy barrier and the best-path cost during the runtime of the algorithm, as shown in Figure 3 (green line). While this is actually an improvement of perceived performance (the user gets a good estimation of the result at an earlier time), a multistep determination approach can also improve real performance (i.e., a reduction of n_{ec}) substantially when used in combination with the method of partial computation introduced in section 4.2. Figure 10B shows the reduction of $\langle n_{ec}\rangle$ when only the rate-limiting step of the path is computed ($r_{\text{sure}} = 0$) instead of the full path ($r_{\text{sure}} = 1$), the edge energies being refined in two steps rather than in a single step. For random noise networks, the reduction of $\langle n_{ec}\rangle$ is already significant (about 30%), while for ordered networks with $o = 0.2$, the savings amount to more than 70%. The reason for this is that for many edges which are below $E_{\text{peak}} - \Delta E_{\text{sure}}$ a partial refinement is already sufficient to obtain an upper-barrier bound of $E_{uv}^{TS,\max} < E_{\text{peak}} - \Delta E_{\text{sure}}$, after which the edge energy is no longer improved. Since $\langle n_{ec}\rangle$ counts the number of full edge refinements, partial refinements do contribute to it.

4.4. Parallelization. Here we propose a modified version of Algorithm 3 that can be executed in parallel on p processors. The communication is realized through a common database of edge energies that is accessed by each process. To avoid that two processes compute the same edge energy simultaneously, the processes need some synchronization. For this, it is necessary that an edge can be flagged in the database as being currently computed. If one of the processes determines a currently flagged edge (u, v) to be a critical edge, this process temporarily assigns a hypothetical edge energy $E_{uv}^{TS,\text{est}}$ to it, which is used only within that process. $E_{uv}^{TS,\text{est}}$ can either be predefined by the user, be given by a statistical estimate (see section 4.1), or be simply $E_{uv}^{TS,\text{est}} = \frac{1}{2}(E_{uv}^{TS,\min} + E_{uv}^{TS,\max})$. The edge (u, v) is added to a set of estimated edges and the algorithm continues to the next iteration, determining another critical edge. In each iteration, the list of estimated edges is checked. If the computing flag for any of these edges has meanwhile been removed, that edge is removed from the list of estimated edges and its energy bounds are reset to the current database values. As the use of estimated edges may produce wrong best paths, the algorithm requires that the list of estimated edges be empty before it can terminate successfully.

ALGORITHM 4 (parallel best path).

(1) Let $F := \varnothing$ be a set of estimated edges.
(2) For each member $(u, v) \in F$ not flagged as being currently computed:
set $E_{uv}^{TS,\min}$ and $E_{uv}^{TS,\max}$ according to their database values, remove (u, v) from F.
(3) Compute two best paths: $P^{\min}$ using $\mathbf{c}^{\min}$ and $P^{\max}$ using $\mathbf{c}^{\max}$ between the transition end-states.
(4) If all edge energies along $P^{\min}$ are determined (i.e., $E_{uv}^{TS,\min} = E_{uv}^{TS,\max}\ \forall$ (u, v) along $P^{\min}$) and $F = \varnothing$: RETURN($P^{\min}$).
(5) Choose from $P^{\min}$ the edge (u, v) with $E_{uv}^{TS,\min} = \max\{E_{wy}^{TS,\min} | ((w, y)$ edge along $P^{\min}) \wedge (E_{wy}^{TS,\min} < E_{wy}^{TS,\max})\}$ (critical edge with undetermined energy). If no such edge exists, GOTO 2.
(6) If (u, v) is flagged as being currently computed:
assign a hypothetical edge energy to (u, v): $E_{uv}^{TS,\min} := E_{uv}^{TS,\max} := E_{uv}^{TS,\text{est}}$.
Add this edge to the set of estimated edges: $F := F \cup \{(u, v)\}$.
Else flag (u, v) as being currently computed, determine it, and remove the flag thereafter.
GOTO 2.

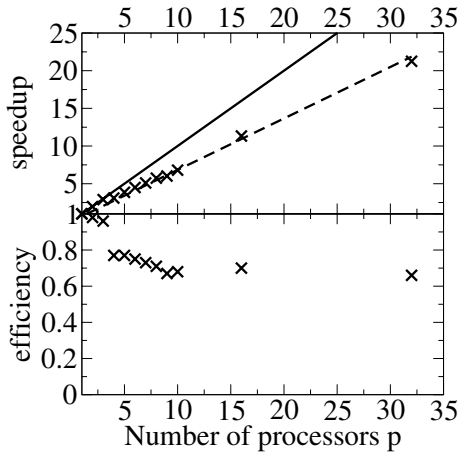

FIG. 11. *Dependence of the parallelization efficiency and speed-up for best-path computations on random networks, on the number of parallel processors, p, on which the job was distributed. The speed-up for parallel best paths approaches* $0.7p$ *(crosses). The solid black line shows the theoretical maximum (speed-up* $= p$*). The efficiency is nearly* 1 *for up to three processors and drops to a value of* 0.7 *for* $p > 10$.

As the estimated edges may temporarily have wrong energies, this error may lead the individual processes to explore regions of the network which are not relevant to determining the best path. Because of these unnecessary computations, there is some loss of efficiency with an increasing amount of parallelization. To quantify this effect, we have simulated the computation of the best path on random networks in parallel on a number, p, of virtual processors. In this simulation, all edge computations were assumed to have equal runtimes. The total number of edges computed in one such simulation is $\langle n_{ec,p} \rangle$, giving rise to an average runtime per processor of $\langle n_{ec,p} \rangle / p$. The normalized runtime compared to the single-processor process is $\langle n_{ec,p} \rangle / (\langle n_{ec,1} \rangle p)$. Thus, the speed-up, S_p, is defined as

$$S_p := \frac{\langle n_{ec,1} \rangle p}{\langle n_{ec,p} \rangle}$$

and the efficiency ϵ_p as

$$\epsilon_p := \frac{\langle n_{ec,1} \rangle}{\langle n_{ec,p} \rangle}.$$

The results are shown in Figure 11. For a large number of processors, the speed-up increases linearly with the number of processors and the efficiency is therefore constant. Computing best paths on up to three processors is practically lossless, the efficiency being near 1. For a larger number of processors, the speed-up is $S_p \approx 0.7p$ (efficiency $\epsilon_p \approx 0.7$).

5. Application to octaalanine. To examine the effectiveness of the algorithms, they are now applied to a real world problem: the computation of the best transition pathways for conformational changes in the peptide octaalanine (Ala_8). Our goal here is not to explain the function and dynamics of polyalanines, which are well studied by both experiment and simulation (see, e.g., [21, 18, 3]). Rather we use octaalanine as a test system to show that multiple optimal pathways can be computed at a very

modest computational cost for a system which allows for an immense number of possible pathways.

5.1. Ala$_8$-network setup. To generate the energy function $U(\mathbf{x})$, the CHARMM force field was used with a united-carbon-atom model (parameter set 19) in the CHARMM program [50]. The electrostatic interactions were computed with ACE 2 [51], which mimics the electrostatic screening effects of aqueous solution. The octaalanine molecule was set up with charged termini (NH_3^+ and COO^-).

To obtain the system states, $5 \cdot 10^5$ conformations were generated in which all ϕ/ψ-backbone dihedral angles were drawn from a uniform random distribution. Each conformation was energy-minimized on $U(\mathbf{x})$, first using steepest descent and then using adopted-basis Newton–Raphson minimizers to a root mean square of the gradient (GRMS) of 10^{-3} kcal mol^{-1} Å^{-1}. To remove redundancy from the resulting conformations, all conformations $\mathbf{x}_u$ were sorted in the order of increasing energies, and each conformation was accepted only if the distance $\mathrm{dist}_{\phi\psi}(\mathbf{x}_u, \mathbf{x}_v)$, calculated as the root mean square deviation in ϕ/ψ coordinates (as defined in (12)), to the nearest already-accepted conformation $\mathbf{x}_v$ was more than 20°. A TN vertex was assigned to each of these diverse energy minima, obtaining $|\mathcal{V}| = 166\,233$ vertices. Their energies, $\mathbf{E}^S$, were given by the potential energies at the minima, minus a constant E_0: $\mathbf{E}^S = (U(\mathbf{x}_1) - E_0, \ldots, U(\mathbf{x}_{|\mathcal{V}|}) - E_0)$. E_0 was chosen to be the absolute potential energy of the lowest-energy minimum, in order to make all energy values nonnegative. Entropical contributions of the energy basins were not taken into account.

State transitions, and thereby the TN edges, were defined between all pairs of vertices $\mathbf{x}_u, \mathbf{x}_v$ with $\mathrm{dist}_{\phi\psi}(\mathbf{x}_u, \mathbf{x}_v) \leq 40°$, yielding $|\mathcal{E}| = 772\,420$ TN edges. This left only 0.3% of the vertices unconnected, which were dismissed from the TN. The a priori edge energy bounds were chosen to be $B_{uv}^{TS,\min} = 0$ kcal/mol and $B_{uv}^{TS,\min} = 100$ kcal/mol.

Energy barriers were determined by computing a minimum energy path between any selected pair of minima $\mathbf{x}_u, \mathbf{x}_v$ using the conjugate peak refinement (CPR) algorithm [35]. CPR optimizes a given initial guess for the transition pathway (here the linear interpolation between $\mathbf{x}_u$ and $\mathbf{x}_v$) by first searching for its maximum and then conducting a multidimensional minimization in the conjugate subspace. This series of maximizations along the path coordinate and minimizations in the remaining subspace results in first-order saddle points that are the local energy maxima along the path. Here the method was terminated if the highest saddle point was refined to a GRMS of 10^{-2} kcal mol^{-1} Å^{-1} for seven successive line minimizations. The potential energy of the highest saddle point, minus E_0, was used as the edge energy, E_{uv}^{TS}. As the CPR calculation identifies the highest saddle point between $\mathbf{x}_u$ and $\mathbf{x}_v$, the "resolution" of the TN (i.e., the minimum distance between pairs of vertices, which was 40° ϕ/ψ RMS in our case) is not critical. A finer resolution, however, might produce lower-energy pathways.

5.2. Best paths: Structural analysis. To determine the end-states, vertices with α-helical and β-hairpin structures were identified based on the shortest hydrogen-bond distances. Left- or right-handed helices in octaalanine form up to four hydrogen bonds between the backbone atoms of residues i and $i+4$, with $i \in \{1, 2, 3, 4\}$. The β-hairpin forms a turn in the middle of the peptide at residues 4–5, and the peptide ends are connected by hydrogen bonds between residue pairs 1–8, 2–7, 3–6. According to these criteria, three vertices corresponding to a right-handed α-helix (α_R), a left-handed α-helix (α_L), and a best β-hairpin (β) were selected as transition end-states.

Figure 12 shows visualizations of these structures. It was found that the best

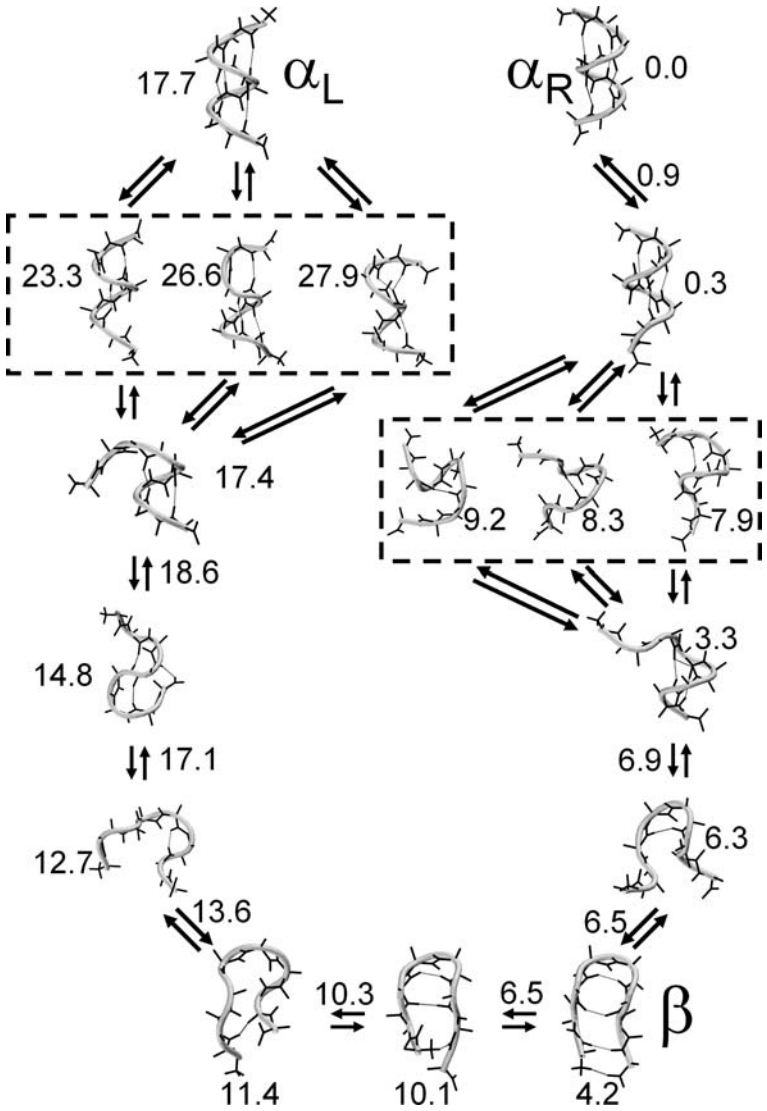

FIG. 12. *Ala_8 structures on the best paths for the $\alpha_L \rightleftharpoons \beta \rightleftharpoons \alpha_R$ reactions. The peptide backbone is drawn as a gray tube, the bonds are represented in black sticks, and hydrogen bonds are drawn as dashed lines. For both subreactions, three alternative rate-limiting transition structures are drawn (dashed boxes). All other structures correspond to energy minima. The numbers near the structures correspond to the vertex energies (potential energy of the minima, relative to α_R structure), and the numbers near the arrows give the edge energies (potential energy of the saddle points, relative to α_R structure). A description of the structural events is given in the text.*

paths between the left- and right-handed helices lead through the β-structure as an intermediate, so that we have following multistep transition:

$$\alpha_L \rightleftharpoons \beta \rightleftharpoons \alpha_R.$$

The energies shown in Figure 12 are given relative to the α_R structure, which has the lowest energy of the structures shown here. The relative energies of α_L and β are 17.7 and 4.2 kcal/mol, respectively. The rate-limiting transition state of

the $\alpha_L \rightleftharpoons \beta$ transition lies very close to the energetically unfavorable α_L structure and has an energy of 23.3 kcal/mol. The two next-best paths were also computed. They differ only in the rate-limiting transition-state, having energies of 26.6 and 27.9 kcal/mol, respectively. The unfolding of the α_L helix starts at one end of the peptide, breaking two successive hydrogen bonds. Then the open half of the peptide forms and inverts its turn, forming an "S"-shaped structure stabilized by two hydrogen bonds. Subsequently, the S twists and a "U"-shaped precursor of the β-hairpin is formed. The hydrogen bonds form from the hairpin to the termini, such that the two β-strands close like a zipper. The first steps of the reaction are likely to follow a well-defined pathway, as the transition energies are high and the alternative best-path energies are considerably higher. However, this is not the case for the subsequent, low-energy steps of the reaction, where the pathway is likely to branch into various alternative pathways towards the β-hairpin structure.

The rate-limiting transition state of the $\beta \rightleftharpoons \alpha_R$ reaction lies approximately equidistant from the β and α_R structures, when about one helix turn has formed. It has a barrier energy of 7.9 kcal/mol. In the best path, one helix turn is formed at the C-terminus. This turn subsequently shifts towards the center of the peptide. The termini then also assume helical turns until the α_R structure is formed. The next-best pathways have rate-limiting steps which are energetically close to the 7.9 kcal/mol barrier, indicating that a large number of different pathways is accessible at physiological temperature.

5.3. Contribution of the individual paths to the overall rate. In order to decide the number, k, of best paths required for a comprehensive description of the transition, it is desirable to calculate their contribution to the overall transition rate. This can be done by formulating the master equation for the system on the network. The time evolution of the population at vertex u is then expressed as

$$\frac{dp_u}{dt} = \sum_{\text{neighbors } v} (k_{vu} - k_{uv}) = \sum_{\text{neighbors } v} (p_v k'_{vu} - p_u k'_{uv}). \tag{16}$$

The rate constants, k'_{uv} and k'_{vu}, may be computed using transition state theory (see (8)), using an expression for the dynamic prefactor ν that is appropriate for the individual system. If only a qualitative convergence criterion is required, as is the case here, the dynamic prefactor can be ignored and rate constants are given in units of ν:

$$k'_{uv,0} = \frac{k'_{uv}}{\nu} = \exp\left(\frac{-(E^{TS}_{uv} - E^{S}_{u})}{k_B T}\right). \tag{17}$$

To test the (relative) contribution of k best paths between vertices v_R and v_P, $(\mathcal{P}_1, \ldots, \mathcal{P}_k)$, we compute the master-equation dynamics in the best-path network which includes all vertices and edges of $(\mathcal{P}_1, \ldots, \mathcal{P}_k)$. This requires first defining an initial probability distribution $(p_1, \ldots, p_{|\mathcal{V}|})$. To do this, each path $\mathcal{P}_i = (v_R \ldots, v_{C1}, v_{C2}, \ldots, v_P)_i$ is split into two path segments $(v_R, \ldots, v_{C1})_i$ and $(v_{C2}, \ldots, v_P)_i$, such that the edge $E^{TS}_{v_{C1}v_{C2}}$ is the highest-energy edge along $\mathcal{P}_i$. The probabilities of all vertices in the best-path network which are on the reactant side of the highest barrier, i.e., $U = \bigcup_{i=1}^{k} (v_R, \ldots, v_{C1})_i$, are initialized with a Boltzmann distribution, i.e.,

$$p_{u \in U} = \frac{\exp(-E^S_u / k_B T)}{\sum_{v=1}^{|U|} \exp(-E^S_v / k_B T)}.$$

We now conduct k master-equation dynamics simulations, starting from the same initial probability distribution each time. In the ith simulation, only the rate-limiting edges (v_{C1}, v_{C2}) of the i best paths are present, while the rate-limiting edges of the remaining $k - i$ best paths are missing from the network. For each simulation, it is calculated how the summed probability of the product side of the highest barrier, $V = \bigcup_{i=1}^{k} (v_{C2}, \ldots, v_P)_i$, evolves:

$$p_V(t) = \sum_{u=1}^{|V|} p_{v \in V}(t).$$

The overall rate constant of the transition $U \to V$ is thus

$$k'_{UV}(t) = \frac{1}{p_V(t)} \frac{dp_V(t)}{dt}.$$

Many transitions exhibit simple kinetics, in the sense that the probability evolution can be fitted by a single exponential, in which case the rate constant is time independent. By calculating the rate constant $k'_{UV}(t)_i$ for each simulation i, the relative contributions of the individual pathways to the overall reaction $U \to V$ can be estimated.

For the present application to Ala_8, a rigorous calculation of the rates is not possible, as the energies used in the TN are potential energies and do not include entropical contributions. Nevertheless, the best-path evaluation was performed in order to give an approximate idea of the number of best paths required. For this, the above evaluation was conducted, integrating the master equations numerically using a fixed time step of 10^{-15} s. This analysis showed that the transition $\alpha_L \to \beta$ is well described by the single best pathway alone, as the contribution of the second-best pathway to the rate constant is already three orders of magnitude smaller than that of the best pathway. In contrast, the rate for the $\beta \to \alpha_R$ does not converge with less than 50 pathways (see Figure 13). Despite the fact that the best pathway contributes around 40% of the total rate for $k = 50$, this confirms that the $\beta \to \alpha_R$ transition is highly disordered and involves significant contributions of configurational entropy. The corresponding pathway shown in Figure 12 must therefore be understood as an example of a large set of possible pathways.

5.4. Best paths: Algorithmic performance. Out of a total number of $|\mathcal{E}| = 772\,420$ TN edges, a number of $n_{low} = 74515$ and $n_{low} = 2568$ edges were below the transition state energy E_{peak} of the $\alpha_L \rightleftharpoons \beta$ and $\beta \rightleftharpoons \alpha_R$ reactions, respectively. Using Algorithm 3, only $n_{ec} = 870$ and $n_{ec} = 865$ edges, respectively, were required to be computed to determine the best path for each of both reactions.

Next, we evaluated the effect of using an statistical estimate for the energy barriers (section 4.1). To set up efficient statistics, it is helpful to find an available measure which correlates well with the barrier energy. For this, each vertex configuration $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_N)$ was characterized by the interatomic distance matrix $\mathbf{D}(\mathbf{x})$, defined as

$$\mathbf{D}(\mathbf{x}) := \begin{bmatrix} 0 & |\mathbf{x}_1 - \mathbf{x}_2| & \cdots & |\mathbf{x}_1 - \mathbf{x}_N| \\ \vdots & \ddots & & \vdots \\ \vdots & & \ddots & |\mathbf{x}_{N-1} - \mathbf{x}_N| \\ 0 & \cdots & \cdots & 0 \end{bmatrix}. \tag{18}$$

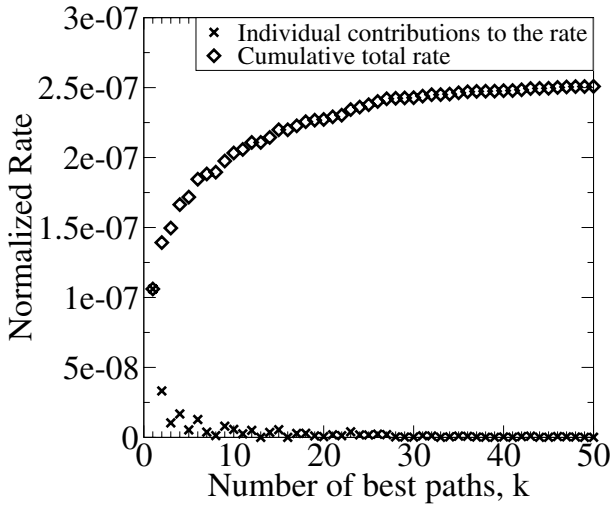

FIG. 13. *The rate constant for the $\beta \to \alpha_R$ transition evaluated on the network defined by $k = 50$ best paths. The contributions to the rate constant from the individual pathways are shown as crosses, and the cumulative rate constant is shown by the diamonds. The overall rate converges around $k = 50$, showing that a large number of different pathways is thermally accessible.*

Only the C_α atom coordinates were used to compute $\mathbf{D}(\mathbf{x})$. $\mathbf{D}(\mathbf{x})$ is one way to characterize individual configurations of atoms, based on interatomic distances. Each edge (u, v) connects two configurations $\mathbf{x}_u, \mathbf{x}_v$. A distance between the vertex pairs u, v was defined according to the distances between the coordinate distance matrices $\mathbf{D}(\mathbf{x}_u)$, $\mathbf{D}(\mathbf{x}_v)$:

$$\text{dist}_{\mathbf{D}}(\mathbf{x}_u, \mathbf{x}_v) := \sqrt{\frac{\sum_{i=1}^{N-1} \sum_{j=i+1}^{N} (\mathbf{D}(\mathbf{x}_u)_{ij} - \mathbf{D}(\mathbf{x}_v)_{ij})^2}{N(N-1)/2}}. \tag{19}$$

As the molecular energy function used here is also based on interatomic distances, the distance-matrix-distance $\text{dist}_{\mathbf{D}}(\mathbf{x}_u, \mathbf{x}_v)$ is correlated with the energy change for a small perturbation of the molecular structure from $\mathbf{x}_u$ to $\mathbf{x}_v$. To set up the energy barrier statistics, 1000 already-computed edges were considered. For all 1000 edges (u, v), the distances $\text{dist}_{\mathbf{D}}(\mathbf{x}_u, \mathbf{x}_v)$ were correlated with the energy barriers, B_{uv}^{TS}. Distance-dependent lower (upper) energy barrier estimates, $B_{uv}^{TS,\min}(\text{dist}_{\mathbf{D}}(\mathbf{x}_u, \mathbf{x}_v))$ ($B_{uv}^{TS,\max}(\text{dist}_{\mathbf{D}}(\mathbf{x}_u, \mathbf{x}_v))$), were generated by choosing values which were less (greater) than a fraction of $r_{conf} = 0.9$ of the data points. The data points and the values of $B_{uv}^{TS,\min}$ and $B_{uv}^{TS,\max}$ are shown in Figure 14. For the $\beta \rightleftharpoons \alpha_R$ reaction, using these estimates reduced n_{ec} by nearly a factor of 2 to $n_{ec} = 458$, whereas the use of estimates for the best-path computation of the $\alpha_L \rightleftharpoons \beta$ reaction provided no significant speed-up (see Table 1). For both reactions, the use of statistical estimates instead of hard bounds led to best paths with the same rate-limiting barriers; i.e., there was no significant error in the result arising from the use of estimates.

We also conducted partial determinations of the best path(s), using finite values of ΔE_{sure} (see Figure 9 and section 4.2). When applied to the TN with a priori edge energy bounds ($B_{uv}^{TS,\min} = 0$, $B_{uv}^{TS,\max} = M$) this did not considerably reduce n_{ec}, as the definite highest edge in the best path was determined only close to the end of the computation. However, when used with the TN with statistically estimated energy

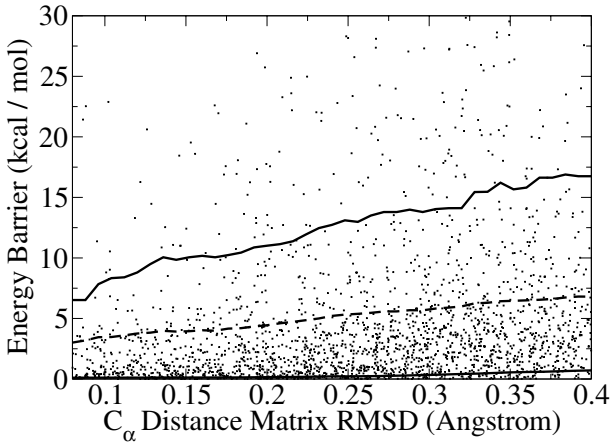

FIG. 14. *Distance-dependent statistical estimate of the edge energy bounds. The distance is calculated as the RMS distance between the C_α-distance matrices (see* (18) *and* (19)*). The energy barrier is calculated as the excess energy of the transition state energy over the maximum of the energies of the two vertices connected by that edge (B_{uv}^{TS}). Each dot corresponds to one out of* 1000 *already-determined edges and represents a distance-barrier pair. The solid lines mark the lower and upper barrier estimates, which are below and above* 90% *of the data points, respectively, for each window of distance values. The dashed line marks the average barrier.*

TABLE 1

Number of computed edges, n_{ec}, for the Ala_8 pathways $\alpha_L \rightleftharpoons \beta$, $\beta \rightleftharpoons \alpha_R$, and $\alpha_L \rightleftharpoons \alpha_R$. (a) *$n_{\text{low}}$, the total number of edges with energies below the highest best-path energy (approximate upper bound for n_{ec}).* (b) *n_{ec} required with Algorithm* 3*, with a priori barrier bounds $B_{uv}^{\min} = 0$ and $B_{uv}^{\max} = 100$ kcal/mol.* (c) *n_{ec} with statistical estimates for the barrier bounds, using a confidence ratio of* 0.9. (d) *Same as* (c) *but with only the highest edge of the best path guaranteed to be optimal ($\Delta E_{\text{sure}} = 0$).* (e) *Same as* (d) *but with edge energies determined in a two-step refinement procedure. The first step produced barrier bounds of $B_{uv}^{TS,\min} = 0$ and $B_{uv}^{TS,\max} = B_{uv}^{TS} + 0.5$ kcal/mol.*

	$\alpha_L \rightleftharpoons \beta$	$\beta \rightleftharpoons \alpha_R$	$\alpha_L \rightleftharpoons \alpha_R$
(a) n_{low}	74515	2568	74515
(b) n_{ec}, normal	870	865	1016
(c) n_{ec}, 90% est.	860	458	970
(d) n_{ec}, 90% est., $\Delta E_{\text{sure}} = 0$	184	458	199
(e) n_{ec}, 2 refinements, $\Delta E_{\text{sure}} = 0$	63	450	71

bounds, the effect was significant for the $\alpha_L \rightleftharpoons \beta$ transition. As shown in Table 1, the rate-limiting transition state ($\Delta E_{\text{sure}} = 0$) was determined with only $n_{ec} = 184$ edges. n_{ec} for the $\beta \rightleftharpoons \alpha_R$ reaction was not reduced.

As was performed for random networks in section 4.3, we also computed the best paths with $\Delta E_{\text{sure}} = 0$ and a two-step refinement scheme. The first step of refinement delivered energies of $B_{uv}^{TS,\min} = 0$ and $B_{uv}^{TS,\max} = E_{uv} + 0.5$ kcal/mol. Using this setup, the $\alpha_L \rightleftharpoons \beta$ reaction was determined with only $n_{ec} = 63$ edges, while, again, n_{ec} for the $\beta \rightleftharpoons \alpha_R$ reaction did not decrease significantly.

6. Summary and conclusion. In the present paper algorithms are introduced that enable efficient computation of the lowest-energy transition pathways for high-dimensional dynamical systems. These systems are modeled using energy-bounded TNs, which require information on the stable system states together with their relative energies and positions. The network connectivity, and thereby the existence of transitions between system states, is defined via a distance cutoff. The exact knowledge of

transition energy barriers, which is very expensive to obtain, is a priori not required, and the barriers are instead bracketed by pairs of lower and upper bounds. Given a program which refines selected energy barriers, and thereby is able to replace the lower and upper bounds by definite values, the algorithm presented here iteratively selects and refines a small subset of energy barriers, such as to obtain the best path. An extension to compute the k best paths is given.

The average number of edges, $\langle n_{ec} \rangle$, that needs to be computed to obtain the best path is shown to increase linearly with the number of edges n_{low} with lower energy bounds below the highest energy barrier of the best path. Moreover, it is shown that one can expect $n_{ec} < n_{\text{low}}$. These theoretical results allow one to predict the computational effort of a best-path computation when the user can guess the energy barrier of the best path.

The precise energy barrier can be quickly determined by performing a partial best-path computation. The performance of this computation is generally improved if edge energy barriers are refined in two or more steps. This is an inexpensive way of isolating the rate-limiting transition state in the network which may already give significant insight in the analyzed process.

As soon as a few hundred barriers have been determined, statistics can be set up which allow one to improve the bounds on the TN energies by statistical estimates. This considerably speeds up further best-path computations. If the individual computations are CPU intensive, it is recommended to use the parallel version of the best-path algorithm, which was shown to exhibit a high efficiency (≥ 0.7) on up to 32 processors.

The methods were applied to the computation of best transition pathways between three conformations of the octaalanine (Ala_8) molecule, showing that the algorithms given here are capable of computing the best paths of complex transitions in high-dimensional dynamical systems in moderate CPU time. For the Ala_8 network, which had $> 10^5$ edges, the best path(s) can be determined by computing only the barriers of several hundred edges.

The method introduced here is widely applicable in the study of the dynamics of complex systems. One particularly promising area, however, is in the determination of best transition pathways for large biomolecules, such as proteins, for which the computation of a single transition barrier may take several hours of CPU time. These transitions, between states of different function, play crucial roles in the workings of the living cell.

REFERENCES

[1] F. H. Stillinger and T. A. Weber, *Hidden structure in liquids*, Phys. Rev. A (3), 25 (1982), pp. 978–989.

[2] R. Elber and M. Karplus, *Multiple conformational states of proteins: A molecular dynamics study of myoglobin*, Science, 235 (1987), pp. 318–321.

[3] M. A. Miller, J. P. K. Doye, and D. J. Wales, *Energy landscapes of model polyalanines*, J. Chem. Phys., 117 (2002), pp. 1363–1376.

[4] F. H. Stillinger and T. A. Weber, *Packing structures and transitions in liquids and solids*, Science, 228 (1984), pp. 983–989.

[5] F. H. Stillinger, *A topographic view of supercooled liquids and glass formation*, Science, 267 (1995), pp. 1935–1939.

[6] R. S. Berry and R. Breitengraser-Kunz, *Topography and dynamics of multidimensional interatomic potential surfaces*, Phys. Rev. Lett., 74 (1995), pp. 3951–3954.

[7] C. L. Brooks, III, J. N. Onuchic, and D. J. Wales, *Taking a walk on a landscape*, Science, 293 (2001), pp. 612–613.

[8] C. Schütte and W. Huisinga, *Biomolecular conformations can be identified as metastable sets of molecular dynamics*, in Handbook of Numerical Analysis, Volume Computational Chemistry, P. G. Ciarlet and J.-L. Lions, eds., North–Holland, Amsterdam, 2003, pp. 699–744.

[9] T. Schlick, E. Barth, and M. Mandziuk, *Biomolecular dynamics at long timesteps: Bridging the timescale gap between simulation and experimentation*, Annu. Rev. Biophys. Biomol. Struct., 26 (1997), pp. 181–222.

[10] D. Frenkel, *Introduction to Monte Carlo methods*, in Computational Soft Matter: From Synthetic Polymers to Proteins, NIC Series 23, N. Attig, K. Binder, H. Grubmüller, and K. Kremer, eds., John von Neumann Institute for Computing, Julich, Germany, 2004, pp. 29–59.

[11] T. Schlick, R. D. Skeel, A. T. Brunger, L. V. Kalé, J. A. Board, Jr., J. Hermans, and K. Schulten, *Algorithmic challenges in computational molecular biophysics*, J. Comput. Phys., 151 (1999), pp. 9–48.

[12] R. Czerminski and R. Elber, *Reaction path study of conformational transitions and helix formation in a tetrapeptide*, Proc. Natl. Acad. Sci. USA, 86 (1989), pp. 6963–6967.

[13] R. Czerminski and R. Elber, *Reaction path study of conformational transitions in flexible systems: Application to peptides*, J. Chem. Phys., 92 (1990), pp. 5580–5601.

[14] K. D. Ball, R. S. Berry, R. E. Kunz, F.-Y. Li, A. Proykova, and D. J. Wales, *From topographies to dynamics on multidimensional potential energy surfaces of atomic clusters*, Science, 271 (1996), pp. 963–967.

[15] O. M. Becker and M. Karplus, *The topology of multidimensional potential energy surfaces: Theory and application to peptide structure and kinetics*, J. Chem. Phys., 106 (1996), pp. 1495–1517.

[16] Y. Levy and O. M. Becker, *Effect of conformational constraints on the topography of complex potential energy surfaces*, Phys. Rev. Lett., 81 (1998), pp. 1126–1132.

[17] M. A. Miller and D. J. Wales, *Energy landscape of a model protein*, J. Chem. Phys., 111 (1999), pp. 6610–6616.

[18] P. N. Mortenson and D. J. Wales, *Energy landscapes, global optimization and dynamics of the polyalanine $Ac(ala)_8NHMe$*, J. Chem. Phys., 114 (2001), pp. 6443–6453.

[19] Y. Levy and O. M. Becker, *Energy landscapes of conformationally constrained peptides*, J. Chem. Phys., 114 (2001), pp. 993–1009.

[20] Y. Levy, J. Jortner, and O. M. Becker, *Dynamics of hierarchical folding on energy landscapes of hexapeptides*, J. Chem. Phys., 115 (2001), pp. 10533–10547.

[21] Y. Levy, J. Jortner, and O. M. Becker, *Solvent effects on the energy landscapes and folding kinetics of polyalanines*, Proc. Natl. Acad. Sci. USA, 98 (2001), pp. 2188–2193.

[22] P. N. Mortenson, D. A. Evans, and D. J. Wales, *Energy landscapes of model polyalanines*, J. Chem. Phys., 117 (2002), pp. 1363–1376.

[23] D. A. Evans and D. J. Wales, *Free energy landscapes of model peptides and proteins*, J. Chem. Phys., 118 (2003), pp. 3891–3897.

[24] D. A. Evans and D. J. Wales, *The free energy landscape and dynamics of met-enkephalin*, J. Chem. Phys., 119 (2003), pp. 9947–9955.

[25] D. J. Wales and J. P. K. Doye, *Stationary points and dynamics in high-dimensional systems*, J. Chem. Phys., 119 (2003), pp. 12409–12416.

[26] D. A. Evans and D. J. Wales, *Folding of the GB1 hairpin peptide from discrete path sampling*, J. Chem. Phys., 121 (2004), pp. 1080–1090.

[27] F. Despa, D. J. Wales, and R. S. Berry, *Archetypal energy landscapes: Dynamical diagnosis*, J. Chem. Phys., 122 (2005), 024103.

[28] D. J. Wales, *Discrete path sampling*, Mol. Phys., 100 (2002), pp. 3285–3305.

[29] J. M. Carr, S. A. Trygubenko, and D. J. Wales, *Finding pathways between distant local minima*, J. Chem. Phys., 122 (2005), 234903.

[30] E. Dijkstra, *A note on two problems in connexion with graphs*, Numer. Math., 1 (1959), pp. 269–271.

[31] D. Eppstein, *Finding the k shortest paths*, SIAM J. Comput., 28 (1998), pp. 652–673.

[32] Y. Levy, J. Jortner, and O. M. Becker, *Dynamics of hierarchical folding on energy landscapes of hexapeptides*, J. Chem. Phys., 115 (2001), pp. 10533–10547.

[33] K. Hukushima and K. Nemoto, *Exchange Monte Carlo method and application to spin glass simulations*. J. Phys. Soc. Japan, 65 (1996), pp. 1604–1608.

[34] Y. Sugita and Y. Okamoto, *Replica-exchange molecular dynamics method for protein folding*, Chem. Phys. Lett., 314 (1999), pp. 141–151.

[35] S. Fischer and M. Karplus, *Conjugate peak refinement: An algorithm for finding reaction paths and accurate transition states in systems with many degrees of freedom*, Chem. Phys. Lett., 194 (1992), pp. 252–261.

[36] H. Jónsson, G. Mills, and K. W. Jacobsen, *Nudged elastic band method for finding minimum energy paths of transitions*, in Classical and Quantum Dynamics in Condensed Phase Simulations, World Scientific, Singapore, 1998, pp. 385–404.

[37] G. M. Torrie and J. P. Valleau, *Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling*, J. Comput. Phys., 23 (1977), pp. 187–199.

[38] M. Blue, B. Bush, and J. Puckett, *Unified approach to fuzzy graph problems*, Fuzzy Sets and Systems, 125 (2002), pp. 355–368.

[39] J.-S. Yao and F.-T. Lin, *Fuzzy shortest-path network problems with uncertain edge weights*, J. Inf. Sci. Eng., 19 (2003), pp. 329–351.

[40] E. E. Kerre, C. Cornelis, and P. De Kesel, *Shortest paths in fuzzy weighted graphs*, Int. J. Intell. Syst., 19 (2004), pp. 1051–1068.

[41] D. A. McQuarrie and J. D. Simon, *Molecular Thermodynamics*, University Science Books, Sausalito, CA, 1999.

[42] P. Hänggi, P. Talkner, and M. Borkovec, *Reaction rate theory: Fifty years after Kramers*, Rev. Modern Phys., 62 (1990), pp. 251–342.

[43] M. Berkowitz, J. D. Morgan, J. A. McCammon, and S. H. Northrup, *Diffusion-controlled reactions: A variational formula for the optimum reaction coordinate*, J. Chem. Phys., 79 (1983), pp. 5563–5565.

[44] S. Huo and J. E. Straub, *The MaxFlux algorithm for calculating variationally optimized reaction paths for conformational transitions in many body systems at finite temperature*, J. Chem. Phys., 107 (1997), pp. 5000–5006.

[45] F. Noé, F. Ille, J. C. Smith, and S. Fischer, *Automated computation of low-energy pathways for complex rearrangements in proteins: Application to the conformational switch of Ras p21*, Proteins, 59 (2005), pp. 534–544.

[46] R. Elber and M. Karplus, *A method for determining reaction paths in large molecules: Application to myoglobin*, Chem. Phys. Lett., 139 (1987), p. 375.

[47] D. Frigioni and A. Marchetti-Spaccamela, *Fully dynamic algorithms for maintaining shortest paths trees*, J. Algorithms, (2000), pp. 251–281.

[48] M. Karoński and A. Ruciński, *The origins of the theory of random graphs*, in The Mathematics of Paul Erdős, Algorithms Combin. 13, R. L. Graham and J. Nešetřil, eds., Springer, Berlin, 1997, pp. 311–336.

[49] A. Amadei, A. B. Linssen, and H. J. C. Berendsen, *Essential dynamics of proteins*, Proteins, 17 (1993), pp. 412–425.

[50] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus, *CHARMM: A program for macromolecular energy, minimization, and dynamics calculations*, J. Comput. Chem., 4 (1983), pp. 187–217.

[51] M. Schaefer and M. Karplus, *A comprehensive analytical treatment of continuum electrostatics*, J. Chem. Phys., 100 (1996), pp. 1578–1599.