Comparison of double-ended transition state search methods

Elena F. Koslover and David J. Wales
University Chemical Laboratories, Lensfield Road, Cambridge CB2 1EW, United Kingdom
(Received 16 April 2007; accepted 9 July 2007; published online 2 October 2007)

While a variety of double-ended transition state search methods have been developed, their relative performance in characterizing complex multistep pathways between structurally disparate molecular conformations remains unclear. Three such methods (doubly-nudged elastic band, a string method, and a growing string method) are compared for a series of benchmarks ranging from permutational isomerizations of the seven-atom Lennard-Jones cluster (LJ7) to highly cooperative LJ38 and LJ75 rearrangements, and the folding pathways of two peptides. A database of short paths between LJ13 local minima is used to explore the effects of parameters and suggest reasonable default values. Each double-ended method was employed within the framework of a missing connection network flow algorithm to construct more complicated multistep pathways. We find that in our implementation none of the three methods definitively outperforms the others, and that their relative effectiveness is strongly system and parameter dependent. © 2007 American Institute of Physics. [DOI: 10.1063/1.2767621]

I. INTRODUCTION

The calculation of optimal pathways between stable molecular conformations is the key to predicting the reaction mechanisms and rates that determine the kinetics of a chemical system. The search for minimum energy pathways (MEP’s) can be reduced to the problem of locating transition states on the potential energy surface (PES). Each such transition state, defined geometrically as a stationary point with exactly one negative Hessian eigenvalue, yields a unique reaction vector, which can be followed downhill in the forward and backward directions via steepest-descent paths to connect two local minima. In order to find transition states on the energy landscape, one must maximize the energy for one degree of freedom while minimizing for all the others—in effect, balancing on a narrow ridge in configuration space.

A number of double-ended search algorithms have previously been developed to find MEP’s connecting two pre-determined local minima. As many of these algorithms yield initial guesses for the intervening transition states rather than converged stationary points, it is often useful to employ single-ended search methods, such as hybrid eigenvector following, to optimize the transition state candidates, and then recalculate the corresponding steepest-descent paths. However, there is no guarantee that these steepest-descent paths will yield a connected path between the original start and end minima. Instead, subsequent double-ended transition state searches may be run for the resulting pairs of unconnected minima, allowing for the generation of complicated multistep pathways, which could not have been resolved in a single double-ended search. The Dijkstra missing connection algorithm described in previous work provides an efficient way to automate this procedure; it uses a distance-based criterion to select pairs of local minima for subsequent connection attempts.

Some of the most commonly used double-ended methods employ a “chain-of-states” approach to discretize the pathway into a series of configuration points, or “images” of the system. The challenge with such methods lies in keeping the images well distributed along the path to provide an accurate representation of the entire path throughout the optimization process. The methods considered in this study—doubly-nudged elastic band (DNEB) and the evolving and growing string methods (ES, GS)—differ primarily in their approach to this image spacing problem.

DNEB is based on the elastic band method originally described by Elber and Karplus and further developed by Henkelman and Jónsson to mitigate corner cutting and sliding down problems. The “doubly-nudged” modification by Trygubenko and Wales was found to improve stability for minimization using a quasi-Newton approach with a loose convergence criterion, a procedure designed to produce candidates for subsequent refinement by hybrid eigenvector-following. The images are kept equidistant by the addition of an artificial “spring force” pulling each of them toward its neighbors, with the interfering components of the spring force and the true potential energy gradient projected out appropriately. In addition to efficiently finding transition states involved in small Lennard-Jones cluster rearrangements, the DNEB method has been successfully employed in generating complex multistep pathways via discrete path sampling, in systems such as buckminsterfullerene, the GB1 hairpin, and the tryptophan zipper.

The zero-temperature string method, proposed by E and Ren, models the path as a single “string” with an inherent parametrization. The image points are distributed at equal arclength intervals along the string and allowed to evolve according to the normal component of the force. Rather than introducing an additional potential to keep the images equidistant as they evolve, the entire string is periodically reparametrized and the image points are simply redistributed along the interpolated path. Such an approach avoids corner-
cutting and sliding-down issues, as there is no extra force preventing the formation of high curvature regions, and the relative image spacing is never allowed to vary beyond a given cutoff value. However, this method is not without its disadvantages, as the perpendicular force on each image tends to jump each time it is shifted along the string, so that the convergence of the entire path is set back with each reparametrization step.

Both the string method and DNEB require an initial interpolation to start off the algorithm. The commonly used approach is simply to use a linear interpolation between the start and end structures. However, such an initial guess can often yield images that are chemically unrealistic, with severe clashes caused by overlapping atoms and even undefined energy or gradient values for some potentials. This problem is addressed in the “growing string method” (GS) developed by Peters et al.\textsuperscript{12} This approach avoids the need for a starting interpolation by initially keeping the center of the string empty of images, and evolving only short fractions of the string adjacent to the start and end points. As the calculation progresses, more and more images are added toward the center from both sides, until the entire string is populated and the final optimization is reduced to the original string method.

While string methods have been compared to nudged elastic bands for a few very simple model systems, such as the Müller-Brown PES and alanine dipeptide,\textsuperscript{12} there have as yet been no benchmarking studies to evaluate their performance in modeling more complex transitions. The present work aims to compare the effectiveness of DNEB and the evolving string and growing string methods for a variety of nontrivial molecular rearrangements, both on their own and within the framework of a more sophisticated multi-step connection algorithm.\textsuperscript{8,9}

II. METHODS

A. Multistep connection framework

Given a starting point and an end point in configuration space, our previously described iterative connection algorithm\textsuperscript{8,9} was used to generate a minimum energy pathway connecting the two. We define an MEP here as a curve satisfying $\nabla V \perp = 0$ at all points, where $\nabla V \perp$ refers to the component of the potential energy gradient perpendicular to the path. A pair of minima is said to be adjacent\textsuperscript{16} if a transition state has been found such that the approximate steepest-descent paths defined by the unique negative Hessian eigenvector connect the two minima. A path connecting distant minima can then be built up from a sequence of adjacent minima and the intervening transition states.

Before beginning the calculations, the distance between the start and end points was minimized with respect to overall translation and rotation.\textsuperscript{17} A double-ended search method (DNEB, ES, or GS) was then employed to generate a set of intermediate configurations between the two minima. Image density and iteration density parameters were used to determine the number of images (per end point separation distance) and the maximal number of iterations (per image) that were used in each run. Those image points that were higher in energy than both their neighbors were selected as candidates for single-ended transition state searches using hybrid eigenvector-following.\textsuperscript{18} Approximate steepest-descent paths from the resulting transition states were then calculated to determine the adjacent local minima. If this procedure did not result in a connected path leading from the starting point to the end point, it was repeated for unconnected pairs of local minima selected by the Dijkstra connection algorithm.\textsuperscript{9}

The three double-ended methods used to generate the transition state candidates at each such iteration are discussed below.

For the connected paths found with the above procedure, the corresponding mean first passage times (MFPT’s) were calculated as described in previous work,\textsuperscript{19,20} taking into account recrossings of intervening minima. Harmonic densities of states and conventional transition state theory were employed to calculate rate constants for individual transitions between adjacent minima.

B. Doubly-nudged elastic bands

The DNEB method is based on the elastic band approach originally described by Elber and Karplus,\textsuperscript{13} where the path is modeled as a series of images connected by elastic bands, so that a spring potential is added to the true potential energy function, helping to keep the images equispaced. As in the NEB method of Jónsson et al.,\textsuperscript{10} the parallel component of the true gradient ($\mathbf{g}$) and the perpendicular component of the spring gradient ($\mathbf{g}^\perp$) are projected out to avoid sliding down and corner cutting, respectively. However, to improve stability under L-BFGS optimization,\textsuperscript{21} the component of $\mathbf{g}$ that does not interfere with $\mathbf{g}^\perp$ is then added back on to the total gradient, resulting in a “doubly-nudged elastic band.”\textsuperscript{15} The force used in DNEB is thus given by

$$f_{\text{DNEB}} = -\mathbf{g}^\perp - \mathbf{g}^\parallel - \mathbf{g},$$

$$\mathbf{g} = \mathbf{g}^\perp - (\mathbf{g}^\perp \cdot \mathbf{g}^\parallel) \mathbf{g}^\parallel.$$  \hspace{1cm} (1)

The band evolves according to this force until either the maximal number of iterations is reached or the root mean square of $\mathbf{g}^\perp$ drops below a convergence tolerance. As we use the DNEB procedure to generate transition state candidates for further refinement by hybrid eigenvector-following, the convergence does not need to be tight. In fact, tight convergence of all the images does not generally seem to be possible for the current formulation of DNEB. Instead, it might be necessary to revert to the NEB formulation of the gradient\textsuperscript{10,11} and an alternative minimization scheme if a converged set of images is required. However, within the present framework we are not aiming to represent the MEP by converging all the DNEB images, since we calculate approximate steepest-descent paths from all the distinct transition states.

C. Evolving strings

The evolving string method employed in the present study is based on the zero-temperature strings proposed by E and Ren.\textsuperscript{4} The primary alteration made here is that the number of discrete image points, rather than the density of im-
ages along the string, is kept constant throughout the calculation. This modification allows for a more direct comparison with DNEB and with the growing string method, described below, where allowing the number of images to vary would result in arbitrarily long string ends at the start of the process.

The string was evolved according to the perpendicular component of the force acting on each image, given by

\[
f^\perp = -g^\perp = -g + (g \cdot \hat{t}) \hat{f},
\]

where \(g\) is the gradient and \(\hat{t}\) is the unit tangent vector to the path. The evolution steps were carried out according to the L-BFGS quasi-Newton algorithm\(^\text{22}\) as described below. After each step, the string was reinterpolated between the image points using either a linear interpolation or cubic splines, and new tangent vectors were recalculated. The smooth cubic spline interpolation provides tangent vectors at each point, while the default tangent approximation used for the linearly interpolated case was the upwind tangent scheme of Henkelman and Jónsson.\(^\text{11}\)

The distances (or arc lengths in the cubic spline case) between neighboring images were checked at each step, and a reparametrization step was deemed necessary if the following condition held:

\[
\min_i \frac{|x_i - x_{i+1}|}{\max_i |x_i - x_{i+1}|} < r
\]

where \(x_i\) is the position of the \(i\)th image. Thus, the string was reparametrized and the images were redistributed when the ratio of the shortest to the longest interimage distance dropped below the tolerance value \(r\). The reparametrization step consisted of sliding the images along the interpolated string to ensure that they were equidistant, and then reinterpolating the string. During the reparametrization, the images generally moved over a much greater distance than the maximal allowed step length for the optimization steps. Consequently, the L-BFGS minimizer was reset after each redistribution of the images, as the approximate inverse Hessian information was assumed to be no longer valid. The string continued to evolve in this manner until either the maximum number of iterations was reached or the following condition was satisfied for a preset convergence tolerance, \(c\):

\[
\frac{\sum_{i=1}^{N} |f^\perp_i|^2}{f_{\text{RMS}}^\perp} < c,
\]

where \(f^\perp_i\) refers to the perpendicular force on image \(i\), as defined in Eq. (3).

D. Growing strings

The growing string method, first proposed by Peters et al.,\(^\text{12}\) works in much the same way as the evolving string described above. In our implementation, the only difference between the two methods lies in the number and the location of images used to discretize the string. At the beginning of the algorithm, there were no images in the central portion of the string and only a few near the start and end points. The two images closest to the center on either side are referred to as the left frontier (lf) and right frontier (rf), respectively. As the string evolved, we kept track of the root mean square (RMS) perpendicular force (\(f^\perp\)) on the frontier images. When \(f^\perp_{\text{lf}}\) dipped below the “growth tolerance” parameter, an extra image was added to the left side of the string according to (for the linear case)

\[
x_{\text{new}} = x_{\text{lf}} + \frac{(x_{\text{hf}} - x_{\text{lf}})}{N - n + 1},
\]

where \(N\) is the desired final number of images and \(n\) is the current number of images. In the cubic spline interpolation case, the new image was placed in an analogous manner along the spline component leading from the left to the right frontier. Similarly, when \(f^\perp_{\text{rf}}\) dropped below the growth tolerance, a new image was added to the right side of the string, with the left and right indices reversed in Eq. (6) above. As in the evolving string method, the growing string is reparametrized when the interimage distances become overly uneven. However, in this case the distances were scaled according to the desired lengths, so that a reparametrization step was taken if

\[
\min_i \frac{|x_i - x_{i+1}|/d_i}{\max_i |x_i - x_{i+1}|/d_i} < r,
\]

where \(d_i\) is the desired interimage distance given by

\[
d_i = \begin{cases} 
L/(N + 1) & \text{for } i \neq \text{rf} \\
L/[1 - n/(N + 1)] & \text{for } i = \text{rf}
\end{cases}
\]

and \(L\) is the total length of the string. Once the left and right frontiers have joined, the subsequent calculations are identical to the evolving string method.

E. Cubic spline interpolation

All cubic spline interpolations of strings were performed parametrically, for each coordinate separately.\(^\text{23}\) The parametrization was based on the cumulative interimage distance and the not-a-knot end point condition was applied.\(^\text{24}\) All arc lengths were calculated via adaptive Gauss-Legendre quadrature\(^\text{25}\) to an approximate tolerance of \(10^{-5}\) length units. Image redistribution to the appropriate arc lengths was carried out using a one-dimensional Newton’s method with a convergence tolerance of \(10^{-4}\) length units. In the few cases where Newton’s method failed to converge, bisection steps were introduced to reset the minimizer.

F. L-BFGS optimization algorithm

The limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) quasi-Newton optimization algorithm\(^\text{21,22,26}\) was employed for each double-ended method to evolve the images according to the specified force vectors. This algorithm is particularly efficient for large systems because it never stores the approximation to the inverse Hessian (\(H^{-1}\))
explicitly but rather calculates $H^{-1}f$ recursively, using information about the changing force vector from $m$ previous steps. In our implementation of the algorithm described by Nocedal, the line search along the quasi-Newton step direction is avoided and the step size for each image is limited by a maximum step parameter. In the calculations described here we used $m=4$, with a maximum step size per image of 0.2 length units.

The L-BFGS method was also used to optimize the local minima that served as end points for the double-ended search methods, to a convergence of at least $10^{-6}$ RMS force units.

III. RESULTS

A. Comparison of methods for LJ$_{13}$ rearrangements

Lennard-Jones (LJ) clusters bound by the pairwise potential

$$V = 4\epsilon \left[ \left( \frac{\sigma}{r} \right)^{12} - \left( \frac{\sigma}{r} \right)^{6} \right],$$

where $r$ is the interatomic separation, provide convenient models for benchmarking transition state search algorithms. A database of global minima for clusters of varying sizes has been compiled, as have transition state and pathway data-bases for some of the more commonly studied clusters. In order to assess the performance of string methods relative to DNEB, we first focus on the thirteen-atom LJ cluster, for which over 6000 distinct transition states structures were downloaded from an online database. For each transition state we employed L-BFGS minimization to find the two adjacent minima. Over 4000 of the transition states considered in the present work connected minima separated by a minimized Euclidean distance of at least 1. These distances were also minimized with respect to permutations among twelve of the thirteen atoms—that is, we consider an LJ$_{13}$ cluster with one labeled atom. We then selected at random 1000 pairs of these adjacent local minima, each connected via a single transition state.

Barrier heights between the starting point and the single intervening transition state for each of these minima pairs ranged between 0.003 and 8.322. At a temperature of 0.15/k the mean first passage times for these rearrangements range from 1.2 to 1.5 in reduced LJ units. The paths corresponding to these fastest and slowest transformations are shown in Fig. 1. Both pathways involve the global minimum Mackay icosahedron, with highly asymmetric potential energy profiles. The fastest path is easiest to describe in reverse. The central atom of the icosahedron migrates into the surface, producing a high energy minimum, which can be described as an octahedron sharing a triangular face and a threefold symmetry axis with

FIG. 1. Energy plotted as a function of integrated path length for the slowest and fastest paths between two adjacent LJ$_{13}$ minima used in the benchmarking studies. The structures of the start and end points, as well as the intervening transition states, are illustrated. The mean first passage time (in LJ units), at a temperature of 0.15$\epsilon/k$, is given for each pathway.

FIG. 2. (Color online) The effectiveness of various double-ended search methods in connecting randomly selected pairs of adjacent LJ$_{13}$ minima is shown in terms of (a) the fraction of pairs successfully connected and (b) the average number of gradient calls for the successful runs. The RMS convergence condition for stationary points was $10^{-6}$ force units. All input parameters were set to the default values listed in Table I.
The number of gradient calls required. The three tangent definitions shown here are

\[
\begin{align*}
\tau_i &= x_{i+1} - x_{i-1}, \\
\tau_i &= \frac{x_i - x_{i+1} - x_{i-1}}{|x_i - x_{i+1} - x_{i-1}|},
\end{align*}
\]

less effective in finding connecting transition state candidates than the linearly interpolated version. We see that the evolving string method and the growing string method have a similar success rate for all image densities studied. In terms of the fraction of successfully connected minima pairs, both of these linearly-interpolated string methods outperform DNEB at higher image densities. However, this is not the case at lower image densities. The number of gradient calls required for the cubic-spline interpolated growing string increases far more rapidly with image density than the other methods discussed here. Furthermore, the total string or elastic band length is significantly shorter for the linearly-interpolated methods than for the cubic-spline growing string, especially at higher image densities (Fig. 3). Thus, at least in our implementation, the latter method can only generate coarsely discretized approximations to the minimum energy path and does not seem suited to cases where closely spaced images along the path are required.

The use of a linear interpolation for the approximate string raises the question of how to define a tangent at each of the image points. The effect of different tangent approximations on both the NEB and DNEB methods has been previously characterized, with the conclusion that an upwind tangent scheme best helps to minimize the development of kinks in the path. The tangent vectors play a somewhat different role in the growing string method, as they do not affect the distribution of images along the string but only the calculation of the perpendicular force. We therefore examined the performance of three possible tangent schemes within the growing string method, as applied to adjacent LJ13 minima. Two of these schemes are based on finite difference approximations, which define the tangent as

\[
\tau_i = x_{i+1} - x_{i-1},
\]

or

\[
\tau_i = \frac{x_i - x_{i+1} - x_{i-1}}{|x_i - x_{i+1} - x_{i-1}|},
\]

where \(x_i\) refers to the position of the \(i\)th image and the resulting tangent vectors are then normalized. The third approach considered was the upwind difference scheme described by Henkelman et al., where the neighboring image...
of highest energy is used to generate the tangent, with a switching function employed at local extrema along the path. Figure 4 shows the performance of each tangent scheme at different image densities. We find that the upwind tangent scheme is indeed more efficient than the other two possibilities, especially in terms of the number of gradient calls required for convergence at higher image densities.

B. The effect of input parameters on string method performance

In addition to the image density, four other input parameters need to be specified for the growing string method, with three of them also applicable to evolving strings. The set of 1000 randomly selected pairs of adjacent LJ13 minima discussed in the previous section was used as a benchmarking system to determine the effects of these parameters and to provide suggestions for reasonable default settings.

The iteration density and the convergence tolerance parameters together determine when the evolution of a string or elastic band is terminated. In the limit of a continuous distribution of images between the two end points, the perpendicular force image on each image goes to zero. Calculations can therefore be terminated when \( f_{\text{RMS}} \) as defined in Eq. (5), becomes so small that the images are essentially no longer moving. On the other hand, if \( f_{\text{RMS}} \) does not converge to zero, as is usually the case with DNEB, the evolution of the string is stopped when some maximal number of steps is reached. The effects of each of these termination criteria are shown in Fig. 5. We find that above an iteration density of approximately 5.0 there is no significant increase in the fraction of minima pairs successfully connected, indicating this value to be an appropriate default for the parameter.

Here we note again that we are not attempting to represent the MEP by converging all the images, but are instead seeking transition state candidates for further refinement. This is the approach that we have found to be most effective for complex multistep paths in large systems. Benchmarking the efficiency with which a single MEP could be located would require some measure of how well a discrete set of images can represent a path. It would probably also require a modification of the DNEB procedure, where tight convergence of all the images is generally not possible. Accurate convergence within our implementation was also problematic for the strings approaches in some of the systems we have considered. In the present work we are therefore assuming that we wish to identify the minimum image density and iteration density required to produce viable transition state candidates for refinement by hybrid eigenvector-following.

In the string method, the distribution of the images along the estimated path is kept approximately uniform by periodically reparametrizing the string and redistributing the images. Thus, one parameter that potentially affects performance is the tolerance for how much the interimage distances are allowed to deviate from the desired value before a reparametrization step is taken [Eq. (4)]. If this parameter is set too low, the images will tend to slide down toward the end points of the string, leading to a poor representation of the path. On the other hand, every time the string is reparametrized, we lose the information from previous steps used to build up an estimate of the inverse Hessian in the L-BFGS scheme, and the steps immediately following reparametrization tolerance parameter on the fraction of successfully connected LJ13 minima pairs and average number of gradient calls required for the linearly interpolated GS approach. An image density of 10.0 was used, with all other parameters set to the default values. For each run, the string was reparametrized whenever the minimal ratio of the actual inter-image distance to the desired distance fell below the reparametrization tolerance. Thus, higher values on the abscissa correspond to more frequent reparametrizations.
parametrization are less productive. Thus, too high a tolerance results in too frequent a redistribution of images, and much slower convergence. Furthermore, when using a linear interpolation between the images each reparametrization step shortens the string upon reinterpolation, leading to a potential corner-cutting problem similar to that encountered with nudged elastic bands if the reparametrization is carried out too frequently. Figure 6 indicates that the fraction of successfully connected minima pairs peaks at a reparametrization tolerance of approximately 0.5, and that overly frequent reparametrizations result in a sharp increase in the number of gradient calls required, without increasing the effectiveness of the method. These results indicate that a good default approach is to reparametrize the string when the shortest interimage distance falls below half the length of the longest distance. The same default reparametrization tolerance parameter was found to be reasonable for the evolving string as well as the growing string method (results not shown).

A parameter unique to the growing string method is the growth tolerance, which determines when the number of images on the string is increased. Too low a growth tolerance will result in excessive amounts of time spent evolving very short left and right string ends before the string is allowed to grow inward to regions more likely to yield transition state candidates. A high growth tolerance would cause the growing string method to become synonymous with an evolving string, leading to the usual problems associated with an initial linear interpolation. Figure 7 indicates that, for the LJ13 minima pairs used in this benchmark, there is little correlation between the growth tolerance and the fraction of minima successfully connected, if the tolerance value is set above approximately 0.2. This result is hardly surprising considering that the growing string and evolving string methods performed very similarly in this study. However, we can see from the same figure that using too low a growth tolerance should be avoided as it leads to a sharp increase in the number of gradient calculations required. Furthermore, at lower growth tolerances, the fraction of successfully connected minima actually decreases, indicating that over-evolving the string while there are only a few images at either end is counterproductive. A reasonable default for the growth tolerance therefore seems to be around 0.25.

Assuming that the observed trends can be extrapolated to other systems, we can use the results generated for the LJ13 minima pairs to suggest appropriate default values of the various parameters involved in the growing string method. Table I summarizes these suggestions, and the parameters listed therein were used as the defaults for all examples described here.

C. LJ7 permutational isomerizations

While the LJ13 results described above show no significant difference in the effectiveness of the growing string versus the evolving string method, it is interesting to compare them for a system where the linear interpolation yields a particularly poor guess for the initial string. Such a comparison is made here for permutational isomerizations of LJ7, where clashes between atoms along the initial linear interpolation lead to singular starting forces on the images. The global minimum was obtained from the Cambridge Cluster Database and double-ended transition state searches were run to connect permutational isomers of this minimum involving exchanges of two adjacent atoms [Fig. 8(a)]. Only one connection attempt was made in each case, as for LJ13 above.

All three double-ended search methods resulted in the same final paths, which are illustrated in Fig. 9. The mean first passage times for the rearrangements in question are 1731 and 4224 LJ units. The 1–2 permutation proceeds via an initial diamond-square-diamond (DSD) rearrangement to the capped octahedron, followed by an edge migration and a final DSD transformation back to the pentagonal prism. The second pathway involves a particularly high energy transition state corresponding to edge migration, where a number of nearest-neighbor contacts are initially broken, leading to a large barrier. This step is followed by two successive DSD processes, which are much faster.

The results of the double-ended transition state search calculations are plotted in Fig. 10. For both rearrangements, the RMS perpendicular force for the DNEB and the ES methods are initially very large, but quickly decrease and

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image density</td>
<td>System dependent</td>
</tr>
<tr>
<td>Iteration density</td>
<td>5.0</td>
</tr>
<tr>
<td>Reparametrization tolerance</td>
<td>0.5</td>
</tr>
<tr>
<td>Growth tolerance</td>
<td>0.25</td>
</tr>
<tr>
<td>Convergence tolerance</td>
<td>0.001</td>
</tr>
</tbody>
</table>

FIG. 7. Effect of the growth tolerance parameter on the fraction of successfully connected LJ13 minima pairs and the average number of gradient calls required for the linearly interpolated GS approach. An image density of 10.0 was used, with all other parameters set to default values. For each run, the number of images was increased when the perpendicular force on one of the frontier images dropped below the growth tolerance. At higher abscissa values the GS method approaches evolving strings.

TABLE I. Suggested default values for parameters used in the string methods, based on results from LJ13 cluster benchmarks. Note that there is no way to predict the appropriate minimal image density necessary to resolve the path, as this would require knowledge of the number of intervening transition states. It is advisable to start with quite a low image density and raise it progressively if no connecting path is found (Ref. 8).
become comparable with the growing string values. In terms of the number of gradient calls, comparable RMS forces occur before the growing string ends join and thus before it is possible to halt the evolution of the string and optimize the transition state candidates along it.

Figure 10 shows that the RMS perpendicular force for DNEB does not converge to zero for either of the two rearrangements, indicating that this parameter is not a good criterion for the accuracy of the estimated MEP for this system. This result reflects the observation that the DNEB formulation appears to accelerate the initial convergence of the images from a poor starting guess, but may actually hinder tight convergence. Instead, we have considered the proximity of the closest image to the highest energy converged transition state obtained by hybrid eigenvector-following. The results are shown in Fig. 10(a), which indicates that the ES method performs noticeably better than GS in one of the rearrangements and that DNEB outperforms both of these methods for both cases.

Overall, our results for LJ\textsubscript{7} isomerizations indicate that while an initial linear interpolation may lead to atom clashes and extremely high forces on the string or elastic band, these problems can be resolved in fewer gradient calls than are required to grow the string step by step from the end points in the GS method.

D. Comparing double-ended search methods for multistep connections

Both the string methods and the DNEB approach have been implemented in our public domain OPTIM program, and they can therefore be used within the Dijkstra connection approach for distant end points, which requires multiple double-ended searches. We have therefore compared how the three methods perform for the highly cooperative rearrangements in LJ\textsubscript{38} and LJ\textsubscript{75} clusters between the global and the second-lowest minima. These rearrangements correspond to overall changes in morphology where the end points are separated by high potential energy barriers. In all cases, the calculations were run with the default parameters listed in Table I, which, while not necessarily optimized for these systems, are intended to provide an appropriate initial guess. For each rearrangement, two different permutational isomers of the end points were considered, and the runs are labeled according to the minimized Euclidean distance between the start and end points.

The results shown in Fig. 11 and Table II indicate no clear superiority of any one of the three methods discussed here. All of the multistep pathways found for the LJ\textsubscript{38} and LJ\textsubscript{75} rearrangements included both cooperative steps, with entire sections of the cluster moving in a single minimum to
FIG. 10. (Color online) Comparison of methods for LJ7 permutational rearrangements. The RMS perpendicular force on the images (bottom) and the proximity of the closest image point to the highest energy final transition state (TS) (top) are plotted as a function of the number of gradient calls. The vertical lines (i) indicate the minimum number of gradient calls such that if the evolution of the DNEB band was stopped, and the TS candidates optimized as in Sec. I, a connected path would be found. The vertical lines (ii) indicate the point at which the growing string ends joined. Twenty image points were used for the 1–2 permutation and thirty image points for the 5–6 permutation. Linear interpolation was used for the string methods. The resulting pathways are illustrated in Fig. 9.

FIG. 11. (Color online) Energy plotted as a function of integrated path length (s) for the final path linking the two lowest minima of LJ38 (top) and LJ75 (bottom). The end point structures are shown in Fig. 8. Two permutational isomers of the minima were used in each case, and the different runs are labeled according to the distance separating the minima. Paths were calculated using the GS, ES, and DNEB double-ended search methods within the framework of the multistep Dijkstra connection algorithm (Ref. 9). A starting image density of 1.0 was used for the calculations, with a 50% increase in up to three repeated connection attempts. The first double-ended search was performed with the maximal image density (2.5) in each case.
minimum connection, and steps characterized by the migration of only one or two atoms on the surface. The maximal barrier heights for the LJ38 rearrangement varied from 3.1e to 6.3e, depending on the double-ended search method employed, while the maximum barriers for the LJ75 pathways ranged from 6.8e to 11.8e. We note that these barriers are significantly higher than those on the kinetically relevant paths.14,15 Locating the latter paths would require additional cycles to grow the stationary point database via further double-ended searches, as in the discrete path sampling procedure.14,15 For the initial paths located, the mean first passage times at a temperature of 0.15e/k (listed in Table II) varied by many orders of magnitude, confirming that the different double-ended methods yielded very different pathways. Each of the three methods was found to be the most efficient for at least one of the rearrangements, both in terms of the number of gradient calculations required and in terms of the mean first passage time over the resulting pathway. The path found using GS for the LJ38 end points corresponding to the shorter first passage time over the resulting pathway. The path found using GS for the LJ38 end points corresponding to the shorter first passage time over the resulting pathway.

In addition to the Lennard-Jones cluster rearrangements described above, the multistep connection approach using different double-ended search methods was also applied to the folding pathways of two biomolecules: the C-terminal fragment of the B1 domain of protein G (GB1) (Ref. 39) and CCβ, a peptide used to model amyloid formation.40 The sixteen-residue GB1 peptide forms a β hairpin both in the complete protein G (Ref. 39) and as an isolated fragment in solution.41 We modeled the formation of the hairpin conformation from an unfolded state. The 17-residue CCβ was designed to transform from a trimeric coiled-coil to an amyloid fibril structure at elevated temperatures.40 Here we considered the transition of a single monomer peptide from an α helix to a β strand conformation. For both peptides the CHARMM19 force field with the EEF1 implicit solvation model was employed to define the potential energy surface.42

The results are shown in Table III and Figs. 12 and 13. For the GB1 folding pathway, the growing string method yielded by far the best pathway, both in terms of the mean first passage time and the number of gradient calls required for the calculation. Intermediate structures along this path are characterized by the stacking of two aromatic residues, helping to stabilize the hairpin through hydrophobic interactions prior to the formation of the β-strand hydrogen bonds. For the CCβ conformational transition the growing string method also yielded the fastest pathway, but the DNEB method required fewer gradient calls. The intermediate structures in this rearrangement are characterized by extensive interactions between neighboring arginine and glutamate side chains, mediating the transition between the hydrogen-bonding patterns for the α helix and the β sheet. The evolving string method yielded the geometrically longest, though not necessarily the slowest, pathway for both of these two biomolecules. More extensive comparisons will be required before any general conclusions can be drawn for such systems.

We note that the pathways found in these calculations have mean first passage times ranging between 0.23 s and 10^20 s at room temperature, emphasizing that the initial

### Table III

<table>
<thead>
<tr>
<th>System</th>
<th>Method</th>
<th>GB1</th>
<th>CCβ</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$N_r$</td>
<td>$N_g$</td>
<td>$N_f$</td>
</tr>
<tr>
<td></td>
<td>MFPT</td>
<td>MFPT</td>
<td></td>
</tr>
<tr>
<td>GS</td>
<td>109</td>
<td>17</td>
<td>1427280</td>
</tr>
<tr>
<td>ES</td>
<td>307</td>
<td>39</td>
<td>4295555</td>
</tr>
<tr>
<td>DNEB</td>
<td>222</td>
<td>89</td>
<td>5621191</td>
</tr>
</tbody>
</table>
paths located for such multi-step processes are typically kinetically irrelevant. Such paths provide only the initial starting point for further optimization by methods such as discrete path sampling.\textsuperscript{14} Subsequent refinement of the stationary point database via additional double-ended searches must reduce barriers on both the local (adjacent minima) and nonlocal (well-separated minima) scale, before more realistic reaction rates will be attained. We will report in detail on both these systems in future work using the discrete path sampling approach. For comparison, rates have previously been obtained for GB1 in reasonable agreement with experiment using both discrete path sampling (DPS) (Ref. 43) and transition path sampling.\textsuperscript{44} In a reference calculation using DPS combined with DNEB for the transition state searches we found that the rate approached the experimental value once a database containing around 50 000 transition states had been obtained. We would expect comparable behavior for the CCβ peptide, as it contains a similar number of residues to GB1.

IV. CONCLUSIONS

Results from the various benchmarks described above indicate that there is no definitive advantage in using growing strings, evolving strings, or DNEB. While each method can outperform the others for a particular system and tuning of parameters, we find that in the case of large systems and complicated reaction paths there is no clear way to predict a priori which method will prove most efficient. We have not attempted to systematically optimize the parameters for each individual path. Hence it is likely that faster performance or shorter paths could be found in each case for every method. However, such optimization is only possible once the path has already been extensively characterized. Instead, we have focused upon finding parameter sets that are likely to give reasonable results for different paths and different systems.

The performance of transition state searching algorithms is notoriously implementation dependent. All the computer codes developed in the present work can be accessed in our public domain OPTIM program.\textsuperscript{34} Our results should therefore be reproducible by other groups, since the starting points and pathways are also available from an online pathway database.\textsuperscript{29} Any future improvements to our implementation will be documented and made available in OPTIM.

ACKNOWLEDGMENTS

One of the authors (E.F.K.) would like to acknowledge the Winston Churchill Foundation for financial support. We thank Dr. Brigit Strodel for providing the CCβ end points used in this study.

\begin{itemize}
  \item[8.] A. Trygubenko and D. J. Wales, J. Chem. Phys. 120, 2082 (2004).
  \item[13.] S. A. Trygubenko and D. J. Wales, Mol. Phys. 100, 3285 (2002).
  \item[14.] D. J. Wales, Mol. Phys. 102, 891 (2004).
\end{itemize}
33 C. de Boor, Cubic spline interpolant various endpoint conditions, http://www.netlib.org/pppack/cubspl.f
42 D. J. Wales, Science 293, 2067 (2001).
44 D. J. Wales, OPTIM: A program for optimizing geometries and calculating reaction pathways, http://www-wales.ch.cam.ac.uk/OPTIM/