A Motion Planning Approach to Folding:
From Paper Craft to Protein Folding*

Guang Song — gsong@cs.tamu.edu
Nancy M. Amato — amato@cs.tamu.edu

Technical Report TR00-017
Department of Computer Science
Texas A&M University
July 21, 2000

Abstract

In this paper, we present a framework for studying folding problems from a motion planning perspective. The version of the motion planning problem we consider is that of determining a sequence of motions to transform some configuration of a foldable object (the start) into another configuration (the goal). Modeling foldable objects as tree-like multi-link objects allows us to apply recent techniques developed in the robotics motion planning community for articulated objects with many degrees of freedom (many links) to folding problems. An important feature of this approach is that it not only allows us to study foldability questions, such as, can one object be folded (or unfolded) into another object, but also provides us with another tool for investigating the dynamic folding process itself.

The framework proposed here has application to traditional motion planning areas such as automation and animation, and presents a novel approach for studying protein folding pathways. Preliminary experimental results with traditional paper crafts (e.g., box folding) and small proteins (approximately 60 residues) are quite encouraging.
Keywords: motion planning, probabilistic roadmap methods, paper folding, protein folding, folding pathways.

*This research supported in part by NSF CAREER Award CCR-9624315 (with REU Supplement), NSF Grants IIS-9619850 (with REU Supplement), EIA-9805823, and EIA-9810937, by the Texas Higher Education Coordinating Board under grant ARP-036327-017.

1 Introduction

Folding is a very common process in our lives, ranging from the macroscopic level – paper folding or gift wrapping – to the microscopic level – protein folding. In most instances, while one desires a particular final state to be reached (e.g., the package is wrapped, or the protein's structure is obtained), the knowledge of the dynamic folding process used to reach a particular state is of interest as well. For this reason, we believe motion planning has great potential to help us understand folding. In particular, while motion planning does have the ability to answer questions about the reachability of certain goal states from other states, its primary objective is to in fact determine the motions required to reach the goal.

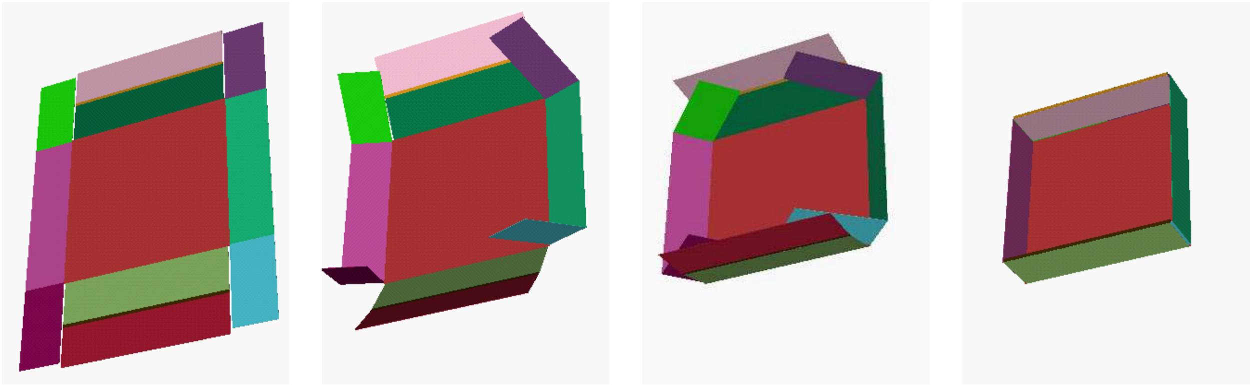
The problem of folding (and unfolding) is an interesting research topic and has been studied in several application domains. Lu and Akella [32] consider a carton folding problem and its applications in packaging and assembly. In computational geometry, there are various paper folding problems, such as, given gluing instructions for a polygon, construct the unique convex polyhedron to which it folds [35]. In computational biology, one of the most important outstanding problems is protein folding, i.e., folding a one-dimensional amino acid chain into a three-dimensional protein structure. The reason we consider them together here is because we will use the same approach to study these seemingly different but actually related problems.

There are large and ongoing research efforts whose goal is to determine the native folds of proteins (see, e.g., [20, 29]). In this paper, we assume we already know the native fold, and our focus is on the folding process, i.e., how the protein folds to that state from some initial state. Many researchers have remarked that knowledge of the folding pathways might provide insights into and a deeper understanding of the nature of protein folding [18, 36]. Although there have been some recent experimental advances [13], computational techniques for simulating this process are important because it is difficult to capture the folding process experimentally. While we recognize that protein folding is vastly more complicated than paper folding, we believe that our successes with paper folding provide some evidence of the feasibility and appropriateness of our motion planning approach for determining folding sequences.

Our approach is based on the successful *probabilistic roadmap (*PRM*)* motion planning method [22]. We have selected the PRM paradigm due to its proven success in exploring high-dimensional configuration spaces (the configuration space, or C-space, of a movable object is the space consisting of all possible positions and orientations of the object). A major strength of PRMs is that they are quite simple to apply, even for problems with high-dimensional configuration spaces, requiring only the ability to randomly generate points in C-space, and then test them for feasibility. To apply the PRM framework to folding processes, we must define the configuration spaces of the objects we are interested in folding. In particular, we model both the paper polygon and the amino acid sequence as multi-link tree-like articulated 'robots', where fold positions (polygon edges or atomic bonds) correspond to joints and areas that cannot fold (polygon faces or atoms) correspond to links. Using the same basic formulation for the various folding problems enables us to utilize the same methodology to study them all. The protein folding problem has an additional complication in that the way in which the protein folds depends on factors other than the purely geometrical constraints which govern the polygonal problems. Nevertheless, we show that these additional factors can be dealt with in a reasonable fashion within the PRM framework.

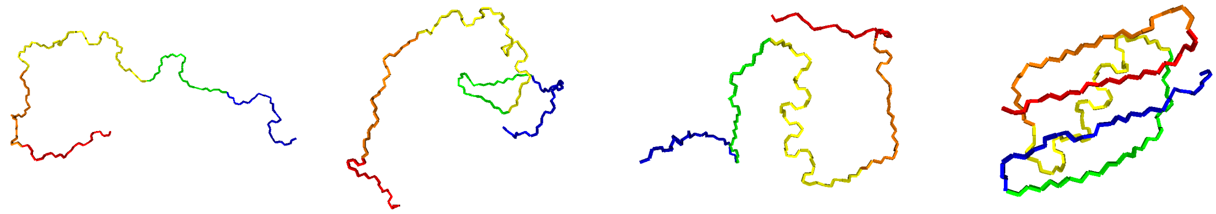
Our preliminary experimental results with traditional paper crafts and small proteins of approximately 60 residues, or 120 degrees of freedom, are quite promising. See Figures 1 and 2 for some path snapshots.

Before describing our approach, we first review related work in computational geometry and



(a) (b) (c) (d)

Figure 1: Snapshots of a carton folding.



(a) (b) (c) (d)

Figure 2: Snapshots of protein GB1 folding.

computational biology.

1.1 Paper folding

Many problems related to the folding and unfolding of polyhedral objects have recently attracted the attention of the computational geometry community [35]. One class of problems concerns itself with the constructibility of certain polygonal or polyhedral structures. Several interesting algorithmic questions relating to origami have attracted the attention of computational geometers, who have obtained some remarkable results. For example, [11] answers a long-standing open problem in origami design by showing that every polygon region (with holes) is the silhouette of some flat origami. They also show that every polyhedron can be 'wrapped' by folding a strip of paper around it, which addresses a question arising in three-dimensional origami, e.g., [1]. There are a number of other interesting questions related to three-dimensional polyhedral objects. For instance, can every convex polytope's surface be unfolded to a non-overlapping simple polygon by cutting along its edges [37]? This problem has application in manufacturing parts from sheet metal [15] (they are in fact more interested in non-convex polyhedra where results are only known for some particular classes of polyhedra [5]). The inverse problem of folding a polygon into a particular polyhedron has also been studied, and results have been obtained for some special cases (e.g., [33]).

Although the problems discussed above can be modeled as articulated objects, in most cases they cannot be modeled as trees. In particular, the incident faces surrounding a given face will form a cycle in the linkage structure. In terms of motion planning, these cycles, often called closed chains, impose additional constraints on the problem (see, e.g., [17, 26]). In this paper we are interested in problems with tree-like linkage structures. Although one might suspect this requirement significantly reduces the complexity, there are in fact some very difficult problems with this property. For example, it is still an open problem to determine whether a simple polygonal chain in the plane can be straightened in such a manner so that all intermediate configurations are simple (edges intersect only at vertices) [27]. However, it has recently been shown that not every tree-like linkage in the plane can be 'straightened' (called 'locking'), that is, there are some pairs of configurations of the linkage which cannot be connected if the links are not allowed to cross [4]. In three dimensions, it has recently been shown that there exist open (and closed) chains that can lock [4, 7], which is a relevant result for the protein folding problem. Finally, in dimensions higher than three, it has recently been established that neither open nor closed chains can lock [8].

The randomized motion planning approach we advocate here is somewhat different in nature to the previous approaches used to study these problems in the computational geometry community. In particular, as the methods we employ are not complete (i.e., they are not guaranteed to find a solution if one exists), they cannot be used to definitively answer a particular question. However, they can provide theorists with a valuable tool for understanding and isolating the difficulty (the 'bottleneck') of a particular folding problem, which might lead to important insights needed to obtain further theoretical results.

1.2 Protein folding and folding pathways

Proteins are the building materials for all life forms: they work as either structural elements or catalysts (enzymes) for synthesizing other proteins. A protein's function is solely determined by its three-dimensional structure, its tertiary structure. This three-dimensional conformation, in turn, is determined by the protein's amino acid sequence, the so-called primary structure of protein. The protein folding problem is to predict a protein's three-dimensional conformation based solely on its amino acid sequence. The spontaneous folding processes are critical in the functioning of all life forms, which makes understanding the mechanism of protein folding one of the most important problems in biology.

The fact that a protein's three-dimensional structure is determined by its amino acid sequence was first demonstrated in Anfinsen's pioneering work [3]. Since then, many different approaches for predicting protein structure have been explored (see [39] for a review). In folding simulations, several computational approaches have been applied to this exponential-time problem, including energy minimization [30, 41], molecular dynamics simulation [28], Monte Carlo methods [9, 23], and genetic algorithms [6, 40]. Among these, molecular dynamics is most closely related to our approach. Much work had been carried out in this area [10, 12, 16, 28], which tries to simulate the true dynamics of the folding process using the classical Newton's equations of motion. The forces applied are usually approximations computed using the first derivative of an empirical potential function. The advantage of using molecular dynamics is that it helps us understand how proteins fold in nature. It also provides a way to study the underlying folding mechanism, to investigate folding pathways, and can provide intermediate folding states. However, the simulations required for this approach are computationally intensive and time-dependent. The simulation result also depends heavily on the start conformation and can easily result in local minima.

Most of the proposed techniques have tremendous computational requirements because they attempt to simulate complex kinetics and thermodynamics. In this paper, we present an alternative approach that finds approximations to the folding pathways while avoiding detailed simulations. Our motion planning approach is based on the successful *probabilistic roadmap* (PRM) method [22]. The PRM methodology has been used to study the related problem of ligand binding [21, 38], which is of interest in drug design. The results were quite promising. The advantages of the

PRM approach are that it efficiently covers a large portion of the planning space, in this case, the conformation space, and that it also provides an effective way to incorporate and study various initial conformations.

2 Preliminaries: Models, C-Space, and Energy Calculations

As mentioned above, PRMs work by sampling points from the moving object's *configuration space*, and retaining those that satisfy certain *feasibility requirements* as roadmap nodes. Then, attempts are made to connect pairs of nearby nodes using (simple) *local planning* methods; successful connections will be saved as roadmap edges.

2.1 C-spaces of folding objects

Both the paper polygon and the amino acid sequence are modeled as multi-link tree-like articulated 'robots', where fold positions (polygon edges or atomic bonds) correspond to joints and areas that cannot fold (polygon faces or atoms) correspond to links. The fold positions of the paper polygon are modeled as revolute joints. For the amino acid sequence of the protein, we consider all atomic bond lengths and bond angles to be constants, and consider only torsional angles (phi and psi angles), which we also model as two revolute joints (2 dof). Thus, in both cases, our models will consist of $n+1$ links and n revolute joints.

The joint angle of a revolute joint takes on values in $[0, 2\pi)$, with the angle 2π equated to 0, which is naturally associated with a unit circle in the plane, denoted by S^1. Assuming some position and orientation for one of the links (the base), the positions of each of the remaining links can be specified by the *joint angle* between the link and some adjacent link. Thus, since in the folding problems considered here we are not concerned with the absolute position and orientation of the object in the environment (i.e., we can use any nominal position for the base link), a *configuration* of a tree-like articulated object can be specified by a vector of n joint angles. That is, the configuration space of interest for our multi-link objects can be expressed as:

$$\mathcal{C} = \{q \mid q \in S^n\}. \tag{1}$$

Note that $\mathcal{C}$ simply denotes the set of all possible configurations, but says nothing about their feasibility. The validity of a point in $\mathcal{C}$ will be determined by collision detection for the polygon problems and by potential energy computations for the proteins.

2.2 Potential energy computations

Finally, another complication is that the way in which the protein folds depends on factors other than the purely geometrical constraints (e.g., no self-collision) which govern the polygonal problems. In particular, there are constraints on the feasible configurations (often called *conformations*) that are related to the potential energy of the conformation. Thus, during the node generation stage, we will reject all nodes whose potential energy is above some predetermined maximum value, $E_{\max}$. The potential energy of the conformations is also needed to simulate the protein folding process, to discover the folding pathways, and to determine if a path is energetically feasible or not. For potential energy calculations, we start with:

$$\begin{aligned} U_{tot} &= \sum_{restraints} K_d\{[(d_i - d_0)^2 + d_c^2]^{1/2} - d_c\} \\ &+ \sum_{atom\ pairs} (A/r_{ij}^{12} - B/r_{ij}^6), \end{aligned}$$

which is similar to the potential used in [28]. The first term represents constraints which favor the known secondary structure through main-chain hydrogen bonds and disulphide bonds. The van der Waals interaction among atoms is considered in the second term. All parameters can be found in [28].

However, even for relatively small proteins (around 60 residues), there will be nearly one thousand atoms. Non-hydrogen atoms also number in the hundreds. Therefore, performing all pairwise van der Waals potential calculations (the second summation) can be computationally intensive. To reduce this cost, we use a step function approximation of the van der Waals potential component. This is computed by considering only the contribution from the side chains and modeling each side chain with a fixed-size rigid sphere (a further approximation). The side chain was chosen for this purpose because it mainly reflects the geometric configuration of a residue. By doing this, the computational cost is reduced by two orders of magnitude. Our results indicate that enough accuracy seems to be retained to still capture the main features of the interaction.

3 Motion Planning using Probabilistic Roadmap Methods

Given a description of the environment and a movable object (the 'robot'), the motion planning problem is to find a feasible (e.g., collision-free) path that takes the movable object from a given start to a given goal configuration. Automatic motion planning has applications in many areas such as robotics, virtual reality systems, and computer-aided design. Although many different motion planning methods have been proposed, most are not used in practice since they are computationally infeasible except for some restricted cases, e.g., when the movable object has very few degrees of freedom (dof) [25]. Indeed, there is strong evidence that any complete planner (one that is guaranteed to find a solution or determine that none exists) requires time exponential in the number of dof of the movable object. For this reason, attention has focussed on randomized or probabilistic motion planning methods. In particular we note the *probabilistic roadmap methods*, or (PRMs), that have recently proven successful on many previously unsolved problems involving high-dimensional C-spaces (see, e.g., [2, 22]).

As mentioned in Section 1, our approach to the folding problem is based on the PRM approach to motion planning [22]. Briefly, PRMs work by sampling points 'randomly' from C-space, and retaining those that satisfy certain feasibility requirements (e.g., they must correspond to collision-free configurations of the movable object). Then, these points are connected to form a graph, or roadmap, using some simple planning method to connect 'nearby' points. During query processing, paths connecting the start and goal configurations are extracted from the roadmap using standard graph search techniques. (See Figure 3.)

A major strength of PRMs is that they are quite simple to apply, even for problems with high-dimensional configuration spaces, requiring only the ability to randomly generate points in C-space, and then test them for feasibility (the local connection can often be effected using multiple applications of the feasibility test).

The folding problems, especially protein folding, have a few notable differences from usual PRM applications. First, as our problems are not posed in an environment containing external obstacles, the only collision constraint we impose is that our configurations be self-collision free. Also, for the protein folding problem, our preference for low energy conformations leads to an additional constraint on the feasible conformations. Second, in PRM applications, it is usually considered sufficient to find *any* feasible path connecting the start and goal configurations. This is because the problems studied are so difficult that simply determining if a path exists is very challenging. For our folding problems, however, we are interested not only in whether there exists a path, but

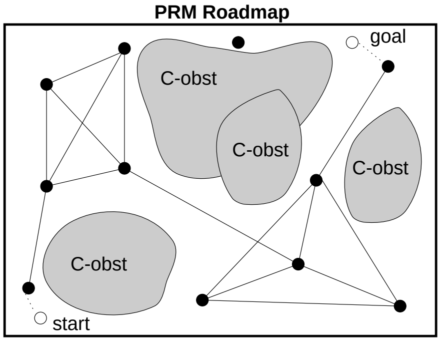


Figure 3: Querying a PRM roadmap (C-space).

we are also interested in the *quality* of the path. For example, for the paper folding problems, one is interested in a path which makes a minimal number of folds, and for the protein folding we are interested in low energy paths.

The particular PRM used for the paper folding problems is the obstacle-based PRM called OBPRM[2]. Briefly, in OBPRM, the roadmap nodes are generated on constraint surfaces (C-obstacles surfaces). Although there are no external obstacles present in our environments, OBPRM still proves effective since many important configurations are close to self-collision configurations (C-obstacles represent self-collision) The results presented for the protein folding currently employ the basic PRM approach [22] which uses uniform sampling in C-space.

3.1 Node generation

As described in Section 2.1, since all joints are revolute, a configuration $q \in \mathcal{C}$ can be generated by assigning each joint angle a value in its allowable range. Once all the joint angles are set, the object's three-dimensional structure is fully determined.

For the paper folding, the configuration of each link is then calculated and self-collision among the links is checked. The node is discarded if any collision occurs.

For the protein molecular model, after the joint angles are known, the coordinates of each atom in the system are calculated, and these are then used to determine the potential energy of the conformation, as defined in Section 2.2. The node is accepted and added to the roadmap based on its potential energy E with the following probability [38]:

$$P(E) = \begin{cases} 1 & \text{if } E < E_{\min} \\ \frac{E_{\max}-E}{E_{\max}-E_{\min}} & \text{if } E_{\min} \leq E \leq E_{\max} \\ 0 & \text{if } E > E_{\max} \end{cases}$$

This filtering helps us to generate more nodes in low energy regions, which is desirable since we are interested in finding the pathways that are most energetically favorable (low energy). If one thinks of the potential field in C-space as a high-dimensional terrain, a folding path snakes along the valleys. In our case, we set $E_{\min} = 50000$ KJouls/mol and $E_{\max} = 89000$ KJouls/mol, which favors configurations with well separated side chain spheres. For example, a configuration with overlapping side chain spheres has higher potential and is thus rejected during node generation.

3.2 Constructing the roadmap

The second phase of the algorithm is roadmap connection. For each node, we first find its k nearest neighbors in the roadmap (using Euclidean distance in C-space), for some small constant k, and then try to connect it to them using some simple local planner. For both the paper folding and protein folding models, each connection attempt performs feasibility checks for N intermediate configurations between the two corresponding nodes as determined by the chosen local planner (the number of such configurations is, e.g., the resolution used for collision detection, which may be set by the user). If there are still multiple connected components in the roadmap after this stage (which is generally the case, and in fact is sometimes unavoidable, see, e.g., [4, 7]), other techniques will be applied to try to connect different connected components (see [2] for details).

When two nodes are connected, the corresponding edge is added to the roadmap. We associate a weight factor with each edge. For the paper folding, the weight factor is simply N, the number of intermediate configurations on the edge. For the protein folding, the weight is calculated in a different way. For two consecutive intermediate conformations i and $i+1$ on the edge, we first calculate their potential energies, i.e., E_i and E_{i+1}, and then the probability of moving from conformation i to $i+1$ is determined by:

$$P_i = \begin{cases} e^{\frac{-\Delta E}{kT}} & \text{if } \Delta E > 0 \\ 1 & \text{if } \Delta E \leq 0 \end{cases}$$

which keeps the detailed balance between two adjacent states. And the total weight of the edge is [38]:

$$Weight = \sum_{i=0}^{N-1} -log(P_i),$$

By assigning the weights in this manner, we can find the shortest or most energetically feasible path when performing subsequent queries.

3.3 'Querying' the roadmap

The resulting roadmap can be used to find a feasible path between given start and goal configurations. This is done a bit differently than what is normally done at this stage. Usually, attempts are made to connect the start and the goal configurations to the same connected component of the roadmap. If this succeeds, a path is returned, otherwise failure is reported.

For our folding problems, it is convenient to actually connect the start and the goal into the roadmap, just as was done for the other roadmap nodes in the connection phase. Dijkstra's algorithm is then used to find the smallest weight path between the start and goal configurations. For the protein folding, if the potential of some intermediate node is too large (as compared to some predetermined maximum), a failure is reported, otherwise the path is returned. The advantage of adding the start and the goal to the roadmap is two-fold. First, the roadmap is augmented after each query is performed (this has been noted as a possible optimization for regular PRM applications as well). Second, this facilitates our search for the lowest weight path.

4 Validation of Folding Pathways

For the protein folding pathways found by our PRM framework to be useful, we must find some way to validate them with known results. Even though the folding pathways provided by PRMs

Paper Folding Roadmap Construction Statistics					
Model	**dof**	**Gen**	**Con**	**#CC**	**#Nodes**
Box	12(5)	38.7	201	1	1035
Periscope	11	13	177	1	883

Table 1: Roadmap construction statistics for the Box and Periscope models. The Box has 12 links, but its dof becomes 5 after symmetry is exploited. 'Gen' and 'Con' represent node generation and connection times in seconds, resp. #Nodes and #CC are the number of nodes and connected components, resp., in the resulting roadmap.

cannot be explicitly associated with actual timesteps, they do provide us with a temporal ordering. Therefore, we could study the following features:

- The intermediate or transition states on the pathway, and the order in which they are obtained.
- The formation order of secondary structures.

Folding intermediates have been an active research area over the last few years, even though there is still debate about whether a protein must go through intermediate states to reach the native conformation, see, e.g., [34]. (This is thought to be true for some, but not all, proteins.) Therefore, one possibility is to compare our folding pathways with experimental results known about folding intermediates. If it is useful to identify intermediate states, and the PRM technique is shown to be successful in determining them, then this approach could prove to be a valuable tool for studying protein structure formation.

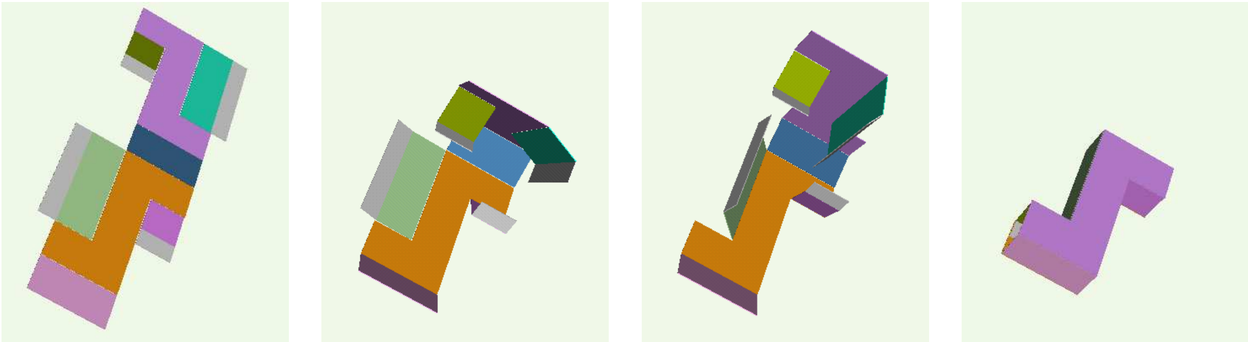
The formation order of secondary structures is also a very important feature since it is related to a fundamental question in protein folding: do secondary structures always form before the tertiary structure, or is tertiary structure formed in a one-stage transition? In this paper, we focus on validating our folding pathways by comparing the order in which the secondary structures form in our paths to results for some small proteins that have been determined by pulse labeling and native state out-exchange experiments [31]. Alternatively, one could compare our paths to those found by other simulation methods. We plan to investigate this approach in future work.

5 Results and Discussion

We now describe results on paper folding and protein folding problems obtained using our PRM-based approach. In this paper we can only show path snapshots; movies can be found at `http://www.cs.tamu.edu/faculty/amato/dsmft`.

5.1 Implementation details

The PRM code we used was our OBPRM software library [2] (for the protein folding simulation, nodes were generated by standard PRM uniform sampling [22]). We used the RAPID [14] package for three-dimensional collision detection. The experiments were performed on an SGI Octane R10000 machine.



(a) (b) (c) (d)

Figure 4: Snapshots of the periscope folding.

Protein Folding Roadmap Construction Statistics								
Model	**dof**	**Gen**	**Con**	**#N sam**	**#N ret**	**#N BigCC**	**#edges**	**#N path**
Protein GB1	112	130	1500	5000	594	559	898	1
		500	5300	20000	2508	2381	3890	2
		2600	42300	10000	12392	11865	20433	3
Protein A	120	400	1300	5000	555	508	767	5
		1600	5800	20000	2308	2140	3352	4
		9500	50100	100000	11715	11057	17719	6

Table 2: Roadmap construction statistics for Protein GB1 and Protein A. 'Gen' and 'Con' represent the node generation and connection times in seconds, resp. '#N sam' is the number of sampled nodes and '#N ret' is the number of nodes retained after rejecting nodes with high potentials. '#N BigCC' is the number of the nodes in the biggest connected component of the roadmap, '#edges' is the total number of edges, and '#N path' is the number of roadmap nodes in the final folding path.

5.2 Models studied

We study two paper folding models: a *box* and a *periscope*. The periscope has 11 degrees of freedom (11 joints) and the box has 12. However, for the box, the number of dof can be reduced to five using symmetry arguments. Both foldings are non-trivial, and in fact, correspond to what is known as 'narrow passage' problems [19], which are thought to be the last major challenge for path planning of rigid bodies in static environments.

We present results for two small proteins. Protein GB1 has 56 residues (112 dof) and consists of one alpha helix and two beta-sheets. Its structure has been determined by both NMR and crystallography. Protein A has 60 residues (120 dof) and consists of three alpha helices. The pdb files used for the proteins were 1GB1.pdb and 1BDD.pdb, respectively, from the Protein Data Bank at `http://www.rcsb.org/pdb/`.

5.3 Paper folding results

Some statistics regarding the roadmaps constructed for the paper folding problems are shown in Table 1. As can be seen, in both cases the problems were solved rather quickly with relatively small roadmaps. These results are really quite remarkable as the problems are actually considered to be quite challenging motion planning problems. Nevertheless, we see that just a few minutes were needed to construct roadmaps containing solution paths. We believe our success with these

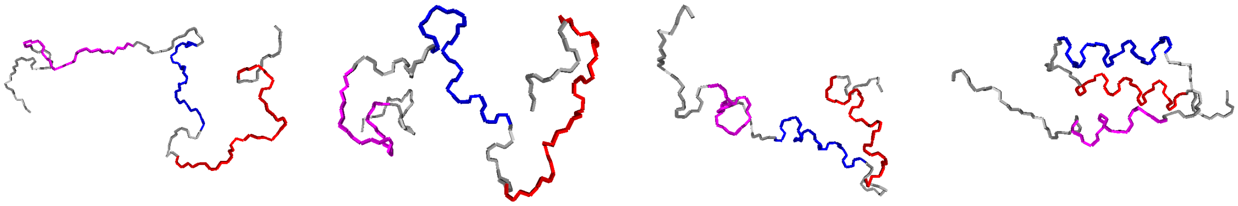


Figure 5: Snapshots of protein A folding.

problems can be attributed to the tendency of the OBPRM roadmaps to contain nodes near the constraint surfaces (i.e., near self-collision configurations) which include configurations necessary for successful paths. For example, configurations in which the flaps of the box fold over other flaps. Snapshots of the folding paths found are shown in Figures 1 and 4 for the box and the periscope, respectively.

5.4 Protein folding results

The results for the protein folding examples are also very interesting. Some statistics regarding the roadmaps constructed for the protein folding problems are shown in Table 2. We provided the goal conformations beforehand, and then searched in the roadmap for the minimum weight path connecting the extended amino acid chain to the final three-dimensional structure. Snapshots of folding paths found by our planner for protein GB1 and protein A are shown in Figure 2 and Figure 5, respectively.

5.4.1 Validation of folding pathways

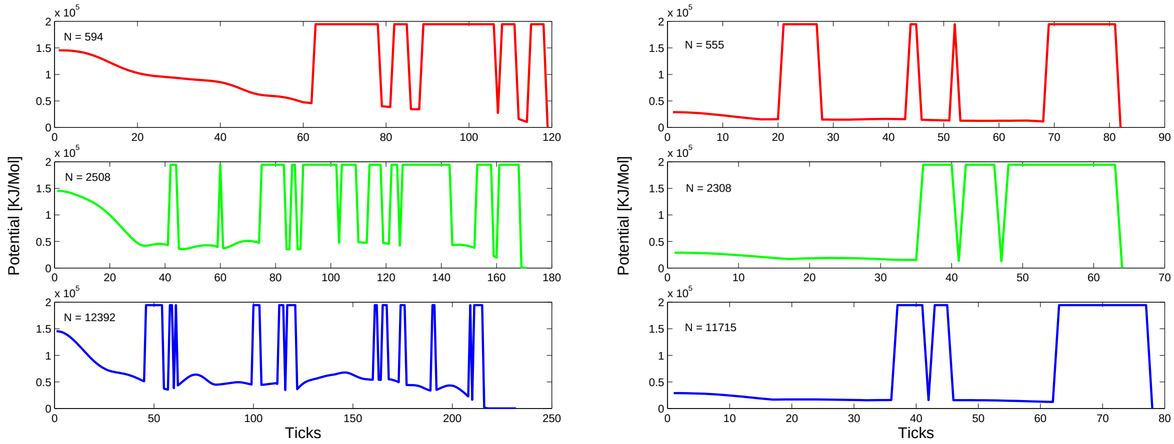
Protein GB1 has 56 residues (112 dofs), and consists of a central alpha helix and two beta-sheets, each composed of two beta strands. Pulse labeling experimental results [24, 31] indicate that the alpha helix and beta strand 4 form first and are protected during hydrogen-deuterium exchanges. This was consistent with the path found by our method. For example, from the snapshots shown in Figure 2, one can clearly see that alpha helix in the middle of the polypeptide forms first.

Protein A has 60 residues (120 dofs), and consists of three alpha helices. The pulse labeling results [31] show that the three alpha helices form at about the same time. As seen in the path snapshots in Figure 5, our paths seem to be consistent with these results.

In general, these results are very encouraging – in both cases, the formation order of the secondary structures seems to agree with the results of the pulse labeling experiments. Thus, while further investigation and tuning of the PRM technique for proteins is still needed, our preliminary findings show that this motion planning approach is a potentially valuable tool. For example, it could be used to study the secondary structure formation order for proteins where this has not yet been determined experimentally.

5.4.2 Analyzing folding pathways

By analyzing the paths found, we may be able to gain some insight into the natural folding process. Towards this end, we analyzed the profiles of the potential energies of the intermediate conformations on the folding paths. This is shown for proteins GB1 and A in Figure 6(a) and 6(b), respectively. We expect that as the number of nodes sampled increases (the sampling is denser),



(a) (b)

Figure 6: Potential along the folding path shown for each intermediate configuration on the path ('tick') for different sized roadmaps. (a) Protein GB1, roadmaps with $N = 594, 2508, 12392$ nodes (top to bottom), (b) Protein A, roadmaps with $N = 555, 2308, 11715$ nodes (top to bottom).

our roadmaps will contain better and better approximations of the natural folding path. Our results support this belief, and moreover, enable us to estimate of how many nodes should be sampled. In particular, we can see in the plots that as the number of nodes, N, is increased, the paths seem to improve in quality, and have fewer and smaller peaks in their profiles.

Another interesting point is the similarity among the paths for all roadmap sizes. In particular, they all illustrate that there is a peak (or peaks) near the goal conformation. Some researchers believe such energy barriers around a folding state are crucial for a stable fold. Also, the profiles clearly show that the peak(s) right before the final fold are contributed by the van der Waals interaction, which is consistent with the tight packing of atoms in the native fold. The similarity among these paths also implies that they may share some common conformations, or subpaths, and this knowledge could be used to bias our sampling around these regions, hopefully further improving the quality of the paths.

6 Conclusion and Future Work

In this paper, we present a framework for studying folding problems from a motion planning perspective. Our approach, which is based on the PRM motion planning method, was seen to produce interesting results for representative problems in paper folding and protein folding. One of the most important benefits of this approach to folding problems is that it enables one to study the dynamic folding process itself. Unfortunately, it is difficult to appreciate this from the few path snapshots we are able to display in a paper. (Movies can be viewed at `http://www.cs.tamu.edu/faculty/amato/dsmft`.) Nevertheless, we believe that our results establish that this is a promising approach which deserves further investigation. In current work, we are further refining our potential energy approximation, investigating more sophisticated sampling techniques to concentrate more nodes near the peaks observed in the path profiles, and are exploring additional validation mechanisms (e.g., comparison to other simulation approaches). We are also performing more extensive analysis of our paths and are studying more proteins.

Acknowledgements

We would like to thank Jean-Claude Latombe for pointing out to us the connection between box folding and protein folding. We would also like to thank Marty Scholtz for suggesting validation using the pulse labeling results, and Michael Levitt and Vijay Pande for useful suggestions.

References

[1] Jin Akiyama. Why Taro can do geometry. In *Proc. 9th Canad. Conf. Comput. Geom.*, page 112, 1997.

[2] N. M. Amato, O. B. Bayazit, L. K. Dale, C. V. Jones, and D. Vallejo. OBPRM: An obstacle-based PRM for 3D workspaces. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 155–168, 1998.

[3] C.B. Anfinsen. Principles that govern the folding of protein chains. *Science*, 181:223–230, 1973.

[4] T. Biedl, E. Demaine, M. Demaine, S. Lazard, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, I. Streinu, G. Toussaint, and S. Whitesides. Locked and unlocked polygonal chains in 3D. In *Proc. 10th ACM-SIAM Sympos. Discrete Algorithms*, pages 866–867, January 1999.

[5] T. Biedl, E. Demaine, M. Demaine, A. Lubiw, J. O'Rourke, M. Overmars, S. Robbins, and S. Whitesides. Unfolding some classes of orthogonal polyhedra. In *Proc. 10th Canad. Conf. Comput. Geom.*, pages 70–71, 1998.

[6] J.U. Bowie and D. Eisenberg. An evolutionary approach to folding small α-helical proteins that uses sequence information and an empirical guiding fitness function. *Proc. Natl. Acad. Sci. USA*, 91:4436–4440, 1994.

[7] J. Cantarella and H. Johnston. Nontrivial embeddings of polygonal intervals and unknots in 3-space. *J. Knot Theory Ramifications*, 7:1027–1039, 1998.

[8] R. Cocan and J. O'Rourke. Polygonal chains cannot lock in 4D. In *Proc. 11th Canad. Conf. Comput. Geom.*, pages 5–8, 1999.

[9] D.G. Covell. Folding protein α-carbon chains into compact forms by monte carlo methods. *Proteins: Struct. Funct. Genet.*, 14:409–420, 1992.

[10] V. Daggett and M. Levitt. Realistic simulation of naive-protein dynamics in solution and beyond. *Annu Rev Biophys Biomol Struct*, 22:353–380, 1993.

[11] E. D. Demaine, M. L. Demaine, and J. S. B. Mitchell. Folding flat silhouettes and wrapping polyhedral packages: New results in computational origami. In *Proc. 15th Annu. ACM Sympos. Comput. Geom.*, pages 105–114, June 1999.

[12] Y. Duan and P.A. Kollman. Pathways to a protein folding intermediate observed in a 1-microsecond simulation in aqueous solution. *Science*, 282:740–744, 1998.

[13] W.A. Eaton, V. Mu noz, P.A.Thompson, C. Chan, and J. Hofrichter. Submillisecond kinetics of protein folding. *Curr. Op. Str. Bio.*, 7:10–14, 1997.

[14] S. Gottschalk, M.C. Lin, and D. Manocha. Obb-tree: A hierarchical structure for rapid interference detection. Technical Report TR96-013, University of N. Carolina, Chapel Hill, CA, 1996.

[15] S. K. Gupta, D. A. Bourne, K. H. Kim, and S. S. Krishnan. Automated process planning for sheet metal bending operations. *J. Manufacturing Systems*, 17(5):338–360, 1998.

[16] J.M. Haile. *Molecular Dynamics Simulation: elementary methods.* Wiley, New York, 1992.

[17] L. Han and N. M. Amato. A kinematics-based probabilistic roadmap method for closed chain systems. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 2000.

[18] B. Honig. Protein folding: From the levinthal paradox to structure prediction. *J. Mol. Bio.*, 293:283–293, 1999.

[19] D. Hsu, L. Kavraki, J-C. Latombe, R. Motwani, and S. Sorkin. On finding narrow passages with probabilistic roadmap planners. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, 1998.

[20] G. N. Reeke Jr. Protein folding: Computational approaches to an exponential-time problem. *Ann. Rev. Comput. Sci.*, 3:59–84, 1988.

[21] L. Kavraki. Geometry and the discovery of new ligands. In *Proc. Int. Workshop on Algorithmic Foundations of Robotics (WAFR)*, pages 435–448, 1996.

[22] L. Kavraki, P. Svestka, J. C. Latombe, and M. Overmars. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. *IEEE Trans. Robot. Automat.*, 12(4):566–580, August 1996.

[23] A. Kolinski and J. Skolnick. Monte carlo simulations of protein folding. *Proteins Struct. Funct. Genet.*, 18:338–352, 1994.

[24] J. Kuszewski, G.M. Clore, and A.M. Gronenborn. Fasting folding of a prototypic polypeptide: The immunoglobulin binding domain of streptococcal protein g. *Protein Science*, 3:1945–1952, 1994.

[25] J. C. Latombe. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, MA, 1991.

[26] S.M. LaValle, J.H. Yakey, and L.E. Kavraki. A probabilistic roadmap approach for systems with closed kinematic chains. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1999.

[27] W. J. Lenhart and S. H. Whitesides. Reconfiguring closed polygonal chains in Euclidean d-space. *Discrete Comput. Geom.*, 13:123–140, 1995.

[28] M. Levitt. Protein folding by restrained energy minimization and molecular dynamics. *J. Mol. Biol.*, 170:723–764, 1983.

[29] M. Levitt, M. Gerstein, E. Huang, S. Subbiah, and J. Tsai. Protein folding: the endgame. *Annu. Rev. Biochem.*, 66:549–579, 1997.

[30] M. Levitt and A. Warshel. Computer simulation of protein folding. *Nature*, 253:694–698, 1975.

[31] R. Li and C. Woodward. The hydrogen exchange core and protein folding. *Protein Science*, 8:1571–1591, 1999.

[32] L. Lu and S. Akella. Folding cartons with fixtures: A motion planning approach. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 1570–1576, 1999.

[33] A. Lubiw and J. O'Rourke. When can a polygon fold to a polytope? Technical Report 048, Dept. Comput. Sci., Smith College, June 1996. Presented at AMS Conf., 5 Oct. 1996.

[34] C.R. Matthews. Pathways of protein folding. *Annu. Rev. Biochem.*, 62:653–683, 1993.

[35] J. O'Rourke. Folding and unfolding in computational geometry. In *Proc. Japan Conf. Discrete Comput. Geom. '98*, pages 142–147, December 1998. Revised version submitted to LLNCS.

[36] E.I. Shakhnovich. Theoretical studies of protein-folding thermodynamics and kinetics. *Curr. Op. Str. Bio.*, 7:29–40, 1997.

[37] G. C. Shephard. Convex polytopes with convex nets. *Math. Proc. Camb. Phil. Soc.*, 78:389–403, 1975.

[38] A.P. Singh, J.C. Latombe, and D.L. Brutlag. A motion planning approach to flexiable ligand binding. In *7th Int. Conf. on Intelligent Systems for Molecular Biology (ISMB)*, pages 252–261, 1999.

[39] M. J. Sternberg. *Protein Structure Prediction*. OIRL Press at Oxford University Press, 1996.

[40] S. Sun. Reduced representation model of protein structure prediction: statistical potential and genetic algorithms. *Protein Sci.*, 2:762–785, 1993.

[41] S. Sun, P.D. Thomas, and K.A. Dill. A simple protein folding algorithm using a binary code and secondary structure constraints. *Protein Eng.*, 8:769–778, 1995.