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

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

Technical Report 00-001
Department of Computer Science
Texas A&M University
January 17, 2000

Abstract

In this paper, we present a framework for studying folding problems from a motion planning perspective. In particular, all folding objects are modeled as tree-like multi-link articulated 'robots', where fold positions correspond to joints and areas that cannot fold correspond to links. This formulation 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 benefit 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 enables us to study the dynamic folding process itself.

The framework proposed here has application in traditional motion planning areas such as automation, teaching through demonstration, animation, and most importantly, presents a different approach to the most profound problem in computational biology: protein struction prediction. Indeed, our preliminary experimental results with traditional paper crafts (e.g., box folding) and a relatively small protein are quite promising.

*Submitted to the ACM Symposium on Computational Geometry (SoCG 2000). 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), it is actually the knowledge of the dynamic folding process used to reach that state which must be understood. 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 a given initial state, 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. 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 [33]. While 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 rather related problems.

Our approach to the folding problem is based on the successful *probabilistic roadmap (PRM)* motion planning method [25]. 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). Briefly, PRMs work by randomly sampling points 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 1.) 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).

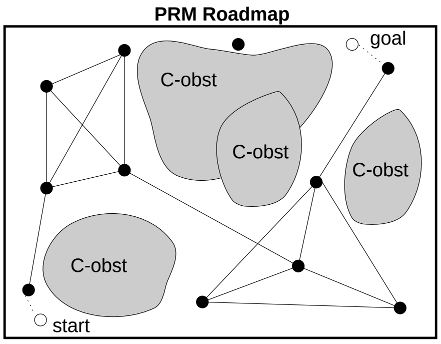
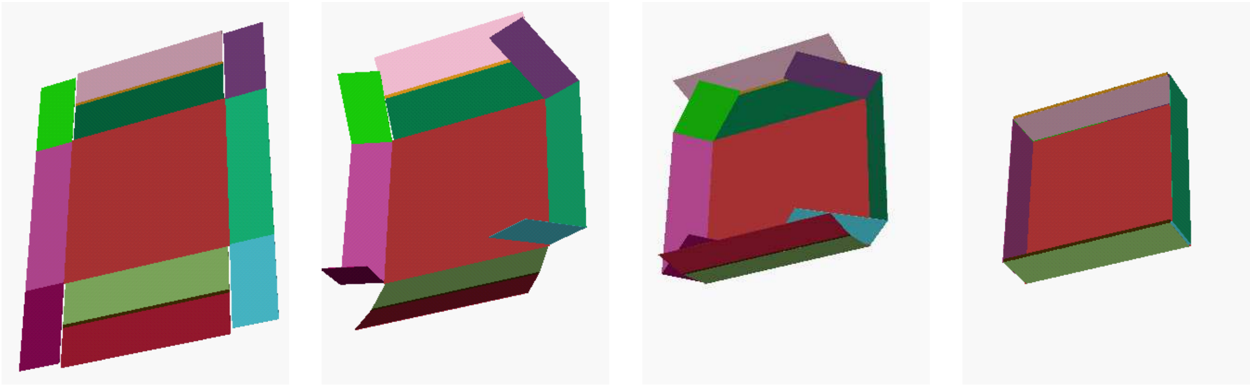


Figure 1: Querying a PRM roadmap (C-space).

To apply the PRM framework to folding problems, 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)



(I) (II) (III) (IV)

Figure 2: Snapshots of a carton folding path found by our planner.

correspond to links. Using the same basic formulation for the various folding problems will enable us to utilize the same methodology to study them all. One complication we deal with in our models is that the kinematics of our tree-like structures are more complex and arbitrary than the serial linkages generally dealt with in the robotics literature (where one can often look-up or obtain nice closed-form solutions). To address this, we decouple the specification of the link's reference frame from its joint specifications, which results in a more general formulation that can be applied to arbitrary tree-like linkages. Finally, another complication is 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 (e.g., box folding, see Figure 2) and a small protein are quite promising.

Before describing our approach, we first review related work in computational geometry and 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 (see, e.g., [8, 14, 15, 16, 33]).

One class of problems concerns itself with the constructability 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 (see, e.g., [6, 16, 26, 27, 28]). For example, [16] 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., [2]. There are a number of other interesting questions related to three-dimensional polyhedral objects [33]. For instance, can every convex polytope's surface be unfolded to a non-overlapping simple polygon by cutting along its edges [35]? This problem has application in manufacturing parts from sheet metal [19], where they are in fact more interested in the case of non-convex polyhedra, for which results are known for some particular classes of polyhedra [8]. 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., [32]).

Although the problems discussed above can be modeled as articulated objects, in most cases

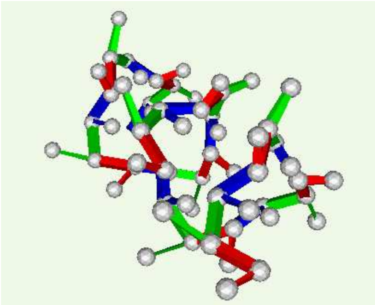


Figure 3: A protein.

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. In this paper we are interested in problems with tree-like linkage structures, i.e., objects whose linkage structures do not contain cycles. 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) [30]. 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 [7]. This result is relevant to the paper folding problems considered here. In three dimensions, it has recently been shown that there exist open (and closed) chains that can lock [7, 10], 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 [11].

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

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, which is also called 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. These 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 amimo acid sequence was first demonstrated in Anfinsen's pioneering work [5]. Since then, many different approaches have been attempted to tackle this problem (see [37] for a review). A general, comprehensive answer is still unknown due to the intrinsic complexity of the problem.

Several computational approaches have been applied to this exponential-time problem, including energy minimization, molecular dynamics simulation, statistical methods, and combinational methods (see [23] and references therein). Among these, molecular dynamics is most closely related to our approach. Much work had been carried out in this area [13, 17, 20, 31], 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 understand folding pathways, and can naturally 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.

In this work, we use a motion planning approach based on the highly successful *probabilistic roadmap* (PRM) method [25] to study the folding process. The PRM methodology has been used to study the related problem of ligand binding [24, 36], 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. As explained below, these same strengths are applicable to the more complex protein folding problem as well.

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 bonds angles to be constants, and consider only torsional angles, which we also model as revolute joints. 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 position of all the remaining links can be specified by the *joint angle* between each 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. In our folding problems, we are not concerned with other obstacles in the environment. However, we do require that no collision exists among the links (this is usually called self-collision in the robotics literature). We denote by $\mathcal{C}_{free}$ the set of configurations $q \in \mathcal{C}$ which do not cause any self-collision among the links.

2.2 Kinematics of folding objects

To specify the connection between each pair of links, Denavit-Hartenberg (DH) notation is adopted [12]. The kinematics of each link (e.g., its position and orientation) can then be computed in a systematic way. In Craig [12], formulae are given for calculating the kinematics for serial linkages (such as most industry robots). However, these cannot be directly applied to our tree-like models since each link can have multiple forward links. The problem is that each link's body frame (local reference frame) is determined by the locations of its joints (one forward, one backward) and the system becomes overconstrained when more joints are present.

We solve this problem in the following way:

- For each link i, we set a body frame F_i, which is independent of any joint connections. Usually, we choose the center of mass as the origin of the body frame.
- For a joint that links body i and j, we use DH notation to define the transformation generated by this joint connection. To express this, we assign a 'DH-frame' to body i and to j, denoted by DH_i and DH_j, respectively, and then use DH parameters to specify the connection between DH_i and DH_j. (In general, the DH-frames are different from the body frames.)
- To get the transformation from F_i to F_j, we first do the transformation from F_i to DH_i, then to DH_j, and finally to F_j.

The advantage of this approach is that a link's body frame and its joint specifications are decoupled, which makes the approach itself more general and applicable to any tree-like linked structure.

2.3 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. For example, 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.

In our potential energy calculations, we use the empirical potential when the molecule is considered to be in vacuum:

$$\begin{aligned} U_{tot} = & \sum_{bonds} \frac{1}{2} K_b (R - R_0)^2 + \\ & \sum_{angles} \frac{1}{2} K_a (\theta - \theta_0)^2 + \\ & \sum_{torsions} K_\phi [1 + cos(n\phi - \delta)] + \end{aligned}$$

$$\sum_{pairs} (A/r_{ij}^{12} - B/r_{ij}^{6}) +$$

$$\sum_{electrostatic} \frac{kq_iq_j}{r_{ij}}$$

The first two terms vanish since we have set the bond lengths and angles as constants. The electrostatic potential is considered only in terms of hydrogen bonds, which is represented in the special van der Waals's parameters between hydrogen atoms that are attached to peptide or amino nitrogen and oxygen atoms in a carbonyl or carboxyl group. The bond twisting constant K_ϕ is set as $2.0k$ Jouls/mol.

The van der Waals parameters, which are taken from [31], are listed in Table 1.

Van der Waals Parameters				
Bond Type	A	B	ε	r^0
H..H	290	1.07	0.0010	2.8525
O..O	145,834	328	0.18479	3.1005
N..N	3952850	2556	0.41315	3.8171
C..C	3075695	953	0.07382	4.3150
A..A	1200965	425	0.03763	4.2202
H..O	2913	241	5.0	1.7

Table 1: Van der Waals Parameters. -ε is the minimum potential energy at separaton r^0, which is the equilibrium radius. They are presented here for comparative purposes. The atom types are defined as follows: O is oxygen, N is nitrogen, H is hydrogen, C denotess extended carbon atoms, and A denotes carbon atoms in a carbonyl or carboxyl group. For the interactons of other atom pairs, we use the geometric means of the A and B values of the atoms involved, for example, $A_{O..N} = (A_{O..O} * A_{N..N})^{1/2}$[31].

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) [29]. 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 [34]. 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., [1, 3, 9, 22, 25]).

As mentioned in Section 1, our approach to the folding problem is based on the PRM approach to motion planning. We selected the PRM paradigm due to its proven success in exploring high-dimensional configuration spaces, which our folding problems have. Briefly, PRMs work by randomly sampling points 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 1.)

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 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 in this work is the obstacle-based PRM called OBPRM, [3, 4]. 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 for the folding problems since there still exist C-obstacles representing self-collision configurations.

3.1 Node generation

As described in Section 2.1, since all joints are revolute, a random configuration $q \in \mathcal{C}$ can be generated by assigning each joint angle a random value in its allowable range. Once all the joints 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 collisions are found.

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 2.3. The node is accepted and added to roadmap based on its potential energy E with the following probability [36]:

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

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 is a path somehow snaking along the valleys. In our case, we chose $E_{\max} = 500kJouls/mol$ and $E_{\min} = 100kJouls/mol$.

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., [7, 10]), other techniques will be applied to try to connect different connected components (see [3] 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:

$$P_i = \frac{1}{1+e^{\frac{E_{i+1}-E_i}{kT}}} = \frac{1}{1+e^{\frac{\Delta E}{kT}}},$$

and the total weight of the edge is [36]:

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

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. In the process, the roadmap is kept intact.

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. If the path's weight 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 Results and Discussion

In this section we describe the use of the PRM motion planning approach to the study of three folding problems, two drawn from the realm of paper folding and one from protein folding.

4.1 Implementation details

The PRM code we used was OBPRM [3], which required only minimal additions to support the decoupling of the reference frames of the links (body frames) from the joint specifications (DH

Roadmap Construction Statistics (for successful folding)					
Model	**dof**	**Gen**	**Con**	**#CC**	**#Nodes**
Box	12(5)	38.7	201	1	1035
Periscope	11	13	177	1	883
Protein	20	8.58	53.8	1	47
		35.5	206	1	182
		116	733	1	624
		429	2770	1	2347

Table 2: Roadmap construction statistics for the Box, Periscope, and Protein models. The Box has 12 links, but its *dof* becomes 5 after symmetry is exploited. The label 'Gen' stands for node generation time and 'Con' for node connection time. #Nodes and #CC are the number of nodes and connected components, respectively, in the resulting roadmap. All times are shown in seconds.

parameters) as described in Section 2.2. We used the RAPID [18] package for three-dimensional collision detection.

The experiments were performed on an SGI Octane R10000 machine.

4.2 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 require some intelligence: the flaps have to be folded in the proper order in order to reach the final folded state. Thus, these problems are not trivial, and in fact, correspond to what is known as the 'narrow passage' problem [21], which is thought to be the last major challenge for path planning of rigid bodies in static environments.

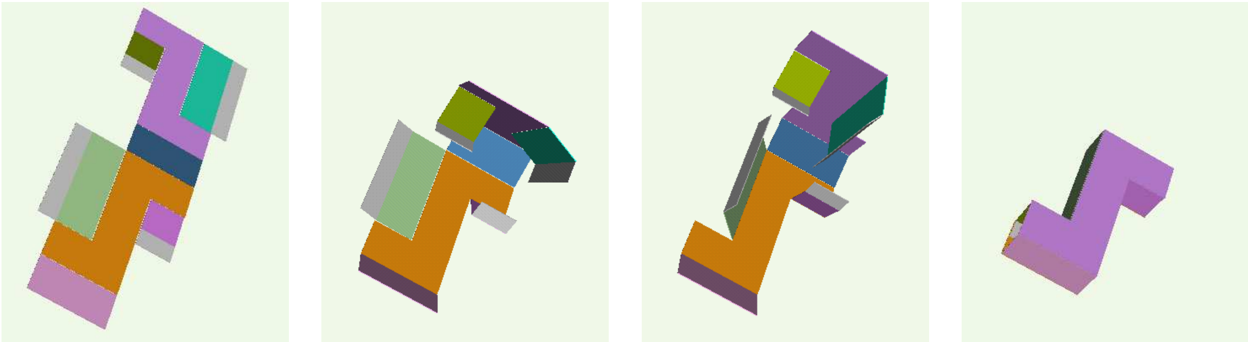
The protein is a small amino acid chain that consists of ten Alanine amino acids, corresponding to 20 degrees of freedom. It folds from its extended (straight) conformation to a second structure known as an *alpha helix*, which is one of the common 'sub-structures' in proteins. The extended conformation's potential energy is around $85k$ Jouls/Mol and the right-helix has potential energy about $13k$ Jouls/Mol.

4.3 Results

Table 2 shows some statistics regarding the roadmaps constructed for the three problems. As can be seen, in all cases the problems were 'solved' rather quickly with relatively small roadmaps.

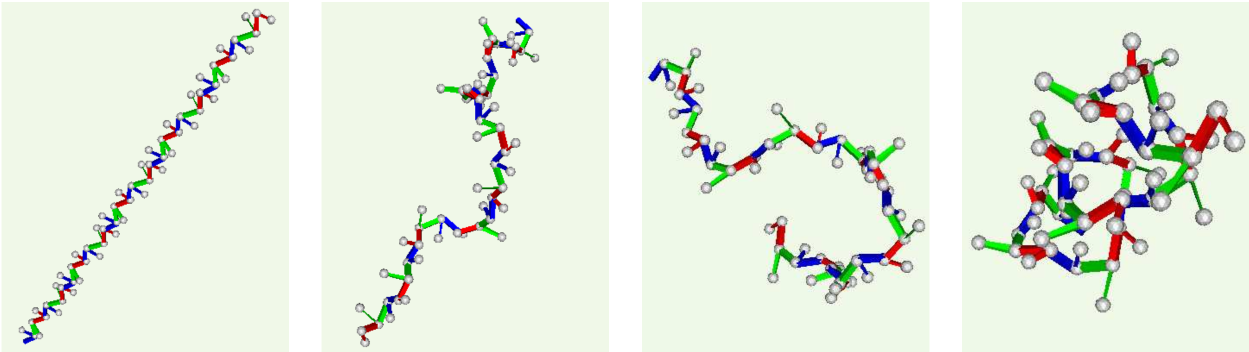
The results for the paper folding examples are really quite remarkable. These problems are actually considered to be quite challenging motion planning problems. Nevertheless, we see that just a few minutes was needed to construct roadmaps containing solution paths. We believe our success with these problems 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. In addition, it supports the modeling framework we adopted for tree-like linkages, which decouples the joint specifications from the body frames. Snapshots of the folding paths found are shown in Figures 2 and 4 for the box and the periscope, respectively.

The results for the protein folding example are also very interesting. In this study, we provided



(I) (II) (III) (IV)

Figure 4: Snapshots of the folding of the periscope model.



(I) (II) (III) (IV)

Figure 5: Snapshots of the protein folding.

the goal conformation beforehand, and then searched in the roadmap for the minimum energy path connecting the extended aminio acid chain to the final three-dimensional structure, an alpha helix. By analyzing the paths found, we will hopefully be able to gain some insight into the natural folding process, which is also biased towards low energy conformations. Our work can be easily extended to study larger and more complicated protein structures, and we plan to do that soon. Snapshots of a folding path found are shown in Figure 5.

In order to study the folding process, we analyzed the profile of the potential energies of the intermediate conformations on the folding path. This is shown in Figure 6 for roadmaps of four different sizes (with $47, 182, 624$ and 2347 nodes, respectively). It should be anticipated that as the number of roadmap nodes approaches ∞, the folding path should approach the 'ideal' path, i.e., the path with minimum folding difficulty (lowest energy), which is probably the way proteins fold in nature. Based on this, we can say that when N, the number of roadmap nodes, is big enough, we will have a good approximation to the natural path. The results shown in Figure 6 tend to support this belief. In particular, we see that as the number of nodes, N, is increased, the paths seem to improve in quality, and most notably 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) around the goal conformation. Their similarity also implies that they may share some common conformations, or subpaths, and this knowledge could be used to to bias our sampling around these paths, hopefully further improving the quality of the roadmaps.

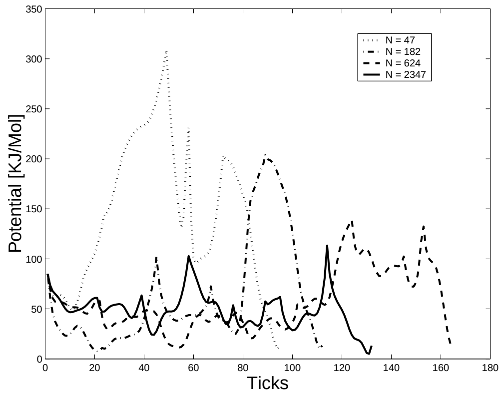


Figure 6: Potential along the folding path shown for each intermediate configuration on the path ('tick').

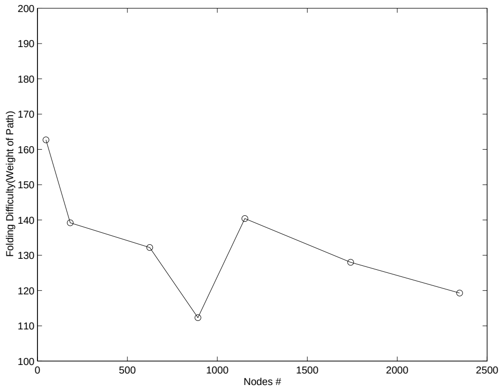


Figure 7: Folding Difficulty (weight of folding path) with respect to the number of nodes in the roadmap.

The folding difficulty of a path, which is proportional to the logarithm of the inverse of the probability, is shown in Figure 7. As anticipated from our discussion above, one can see that the folding difficulty initially decreases quite rapidly as more roadmap nodes are added, but begins to level off fairly quickly (ignoring what seem to be a few outliers).

5 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 very 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. Nevertheless, it is clear that this is a promising approach which deserves further investigation.

In future work, we intend to apply our method to larger, more complex, proteins.

Acknowledgements

We would like to thank Jean-Claude Latombe for pointing out to us the connection between box folding and protein folding.

References

[1] J. M. Ahuactzin and K. Gupta. A motion planning based approach for inverse kinematics of redundant robots: The kinematic roadmap. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 3609–3614, 1997.

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

[3] 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.

[4] N. M. Amato and Y. Wu. A randomized roadmap method for path and manipulation planning. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 113–120, 1996.

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

[6] Marshall Bern and Barry Hayes. The complexity of flat origami. In *Proc. 7th ACM-SIAM Sympos. Discrete Algorithms*, pages 175–183, 1996.

[7] 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.

[8] 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.

[9] V. Boor, M. H. Overmars, and A. F. van der Stappen. The gaussian sampling strategy for probabilistic roadmap planners. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1999. To appear.

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

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

[12] John J. Craig. *Introduction to Robotics: Mechanics and Control, 2nd Edition.* Addison-Wesley Publishing Company, Reading, MA, 1989.

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

[14] E. D. Demaine, M. L. Demaine, and A. Lubiw. Folding and cutting paper. In *Proc. Japan Conf. Discrete Comput. Geom.*, Lecture Notes Comput. Sci., Tokyo, Japan, December 1998. Springer-Verlag.

[15] E. D. Demaine, M. L. Demaine, and A. Lubiw. Folding and one straight cut suffice. In *Proc. 10th Annu. ACM-SIAM Sympos. Discrete Alg.*, pages 891–892, January 1999.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

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

[21] 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.

[22] D. Hsu, J-C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. In *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, pages 2719–2726, 1997.

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

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

[25] 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.

[26] F. Kawahata. *Fantasy Origami.* Gallery Origami House, Tokyo, Japan, 1995.

[27] T. Kawasaki. On the relation between mountain-creases and valley-creases of a flat origami. In H. Huzita, editor, *Origami Science and Technology*, pages 229–237. 1989.

[28] Robert J. Lang. A computational algorithm for origami design. In *Proc. 12th Annu. ACM Sympos. Comput. Geom.*, pages 98–105, 1996.

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

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

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

[32] 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.

[33] 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.

[34] J. Reif. Complexity of the piano mover's problem and generalizations. In *Proc. IEEE Symp. Foundations of Computer Science (FOCS)*, pages 421–427, 1979.

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

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

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