

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par :

Institut National Polytechnique de Toulouse (INP Toulouse)

Discipline ou spécialité :

Intelligence Artificielle

Présentée et soutenue par :

M. LAURENT DENARIE

le mercredi 12 avril 2017

Titre :

Robotics-inspired methods to enhance protein design

Ecole doctorale :

Mathématiques, Informatique, Télécommunications de Toulouse (MITT)

Unité de recherche :

Laboratoire d'Analyse et d'Architecture des Systèmes (L.A.A.S.)

Directeurs de Thèse :

M. THIERRY SIMEON

M. JUAN CORTES

Rapporteurs :

M. CHARLES ROBERT, UNIVERSITE PARIS 6

Mme MARILENA VENDITELLI, UNIV. DEGLI STUDI DE ROME SAPIENZA

Membres du jury :

M. RACHID ALAMI, LAAS TOULOUSE, Président

M. JUAN CORTES, LAAS TOULOUSE, Membre

M. STEPHANE REDON, INRIA GRENOBLE - RHONE ALPES, Membre

M. THIERRY SIMEON, LAAS TOULOUSE, Membre

Acknowledgments

Cette thèse est le fruit de plus de trois années de travail au cours desquelles j'ai pu bénéficier de nombreux soutiens. Cette section est dédiée à toutes celles et ceux qui ont contribué, de près ou de loin, à la réussite de ce projet.

Je remercie chaleureusement mon directeur et mon co-directeur de thèse, Thierry SIMÉON et Juan CORTÉS. Thierry a été présent dans les moments importants et a toujours su donner des conseils avisés. Juan a été un soutien sans faille durant ces trois années de travail. Toujours présent lors de mes nombreuses sollicitations, il a su guider mes pas sur le chemin difficile du doctorat, faisant sans cesse naître de nouvelles idées.

Je remercie Charles ROBERT et Marilena VENDITTELLI, tout deux rapporteurs de mon manuscrit de thèse, pour m'avoir fait l'honneur d'accepter d'évaluer mon travail et de me faire bénéficier de leur expertise.

Je remercie également Stéphane REDON pour avoir accepté de siéger en tant qu'examinateur au sein mon jury.

Je remercie Rachid ALAMI pour m'avoir accueilli au sein de son équipe au LAAS-CNRS et pour avoir accepté de siéger en tant qu'examinateur au sein mon jury.

Je souhaite également remercier mes collègues qui ont contribué d'une manière ou d'une autre à ce travail et plus particulièrement :

- Marc VAISSET pour ses conseils toujours avisés, sa disponibilité, et son efficacité à résoudre les bugs. Son professionnalisme et à sa bonne humeur font de lui un collègue hors pair.
- Kevin MOLLOY pour ses nombreux conseils et pour sa convivialité. Mon travail n'aurait pas été aussi abouti sans son aide précieuse. Nos chemins se recroiseront.
- Didier DEVAURS pour sa disponibilité.
- Ellon PAIVA MENDES pour son aide précieuse en LaTeX.
- Alejandro ESTAÑA, Amélie BAROZET, et Maud JUSOT pour m'avoir supporté pendant tout ce temps, pour avoir mis le nez dans mon code, et pour avoir contribué à la bonne humeur de l'équipe.

Je remercie Stéphane CAMBON pour m'avoir mis en contact avec Rachid ALAMI et pour m'avoir permis de commencer cette aventure.

Ces remerciements seraient incomplets sans y inclure toutes les personnes qui ont indirectement permis de mener ce projet à bien.

Je remercie tout d'abord ma compagne, Anne BRUNO, qui a été le socle sur lequel j'ai pu me reposer tout au long de mon doctorat. Elle a su me supporter dans les moments difficiles et a été une source de motivation. Elle a également été une source d'inspiration, et le sera encore.

Je remercie mes parents qui ont su faire de moi ce que je suis aujourd'hui.

Je remercie Pierrick, Elena, Artur, et Aïva pour les moments de détente aussi bien sur les fauteuils de la cafétériat que dans les baudriers.

Je remercie Laurent, Laura, Jessica, et David pour les bouffées d'air pur données au cours de ce marathon.

Je remercie Nicolas, Harmish, Renaud, Alexandre, Marco, Michelangelo, Grégoire, Ellon (à nouveau), Benjamin, Wuwei, Jules, César, Amandine, Sandra, Sarah, Arthur, Lancelot, Christophe, Clara, Etienne, Raphaël, et Mamoun pour la bonne humeur apportée au sein du laboratoire, et souvent en dehors également, que ce soit autour d'un café, d'une bière ou d'une table de pingpong.

Contents

Glossary vii

Introduction 1

1 Scientific Context 5
1.1 Proteins sequence and structure . . . 5
1.1.1 Amino acids, peptides, and proteins . . . 5
1.1.2 Sequence-function relationship . . . 8
1.2 Protein modeling . . . 8
1.2.1 Cartesian coordinates . . . 9
1.2.2 Internal coordinates . . . 9
1.2.3 Dimensionality reduction . . . 10
1.2.4 Coarse grained models . . . 11
1.3 Energy landscape . . . 12
1.3.1 Physical theory . . . 12
1.3.2 Energy functions . . . 13
1.4 Computational methods for the exploration of protein conformations 14
1.4.1 Molecular dynamics . . . 15
1.4.2 Monte Carlo methods . . . 16
1.4.3 Robotics-inspired algorithms for the exploration of the conformational space . . . 17
1.5 Computational Protein Design . . . 21
1.5.1 The CPD problem . . . 22
1.5.2 The search space . . . 23
1.5.3 Current methods for CPD . . . 24
1.5.4 CPD challenges . . . 25

2 Protein modeling and local conformational sampling 27
2.1 Mechanistic model . . . 28
2.2 Tripeptide decomposition . . . 29
2.3 Devising move classes . . . 30
2.3.1 Perturbing particles . . . 31
2.3.2 Solving inverse kinematics for a tripeptide . . . 33
2.4 Results . . . 34
2.4.1 Implemented move classes and parameter settings . . . 34
2.4.2 Test systems . . . 36
2.4.3 Computational performance . . . 36
2.4.4 Distribution of sampled states . . . 37
2.4.5 Exploration efficiency analysis . . . 40
2.5 Conclusion . . . 44

3 Exploration of the conformational energy landscape 47
3.1 The exploration-exploitation dilemma . . . 48
3.2 Algorithms . . . 49
3.2.1 T-RRT . . . 49
3.2.2 Transition-based EST . . . 49
3.3 Empirical comparative analysis . . . 54
3.3.1 Molecular systems . . . 54
3.3.2 Experiment setup . . . 54
3.3.3 Results . . . 56
3.4 Conclusion . . . 59

4 Toward protein motion design 63
4.1 Problem formulation and approach . . . 64
4.1.1 Problem definition . . . 64
4.1.2 Approach . . . 65
4.2 Algorithm . . . 66
4.2.1 Simultaneous Design And Path-planning algorithm . . . 66
4.2.2 Controling tree expansion . . . 67
4.2.3 Theoretical analysis . . . 68
4.3 Empirical analysis and results . . . 69
4.3.1 Test system description . . . 70
4.3.2 Benchmark results . . . 72
4.4 Application SDAP to protein motion design . . . 76
4.4.1 Problem definition . . . 76
4.4.2 Additional simplifications . . . 77
4.4.3 Preliminary experiments . . . 79
4.5 Conclusions and future work . . . 81

Conclusion 85

A French summary 89
A.1 Introduction . . . 89
A.2 Contexte scientifique . . . 90
A.2.1 Séquence des protéines et structure . . . 90
A.2.2 Modélisation des protéines . . . 91
A.2.3 Paysage énergétique . . . 92
A.2.4 Méthodes d'exploration de l'espace des conformations des protéines . . . 93
A.2.5 Computational Protein Design . . . 95
A.3 Modélisation des protéines et échantillonnage local des conformations 97
A.3.1 Modèle mécanique . . . 98
A.3.2 Décomposition en tripeptide . . . 98
A.3.3 Création de classes de mouvement . . . 99
A.3.4 Résultats . . . 99

A.4 Exploration du paysage énergétique des protéines 100
A.4.1 Le dilemme exploration-exploitation 101
A.4.2 Algorithmes 101
A.4.3 Analyse comparative empirique 102
A.4.4 Résultats 103
A.5 Vers la conception de mouvements de protéine 103
A.5.1 Définition du problème et approche 105
A.5.2 Algorithme 105
A.5.3 Analyse empirique et résultats 105
A.5.4 Application de SDAP à la conception d'un mouvement de protéine 106
A.6 Conclusions 107

Bibliography **111**

Glossary

6R Six-revolute. Used to indicate that a serial manipulator has six revolution joints. This corresponds to the shortest full instantaneous mobility of the end frame relatively to the base frame of the manipulator.

C_α α-carbon of an amino acid to which is attached the side-chain.

*ec*DHFR *Escherichia coli* DHFR

CFN Cost Function Network

CPD Computational Protein Design

DEE Dead-End Elimination

DHF 7,8-dihydrofolate

EST Expansive-Spaces Trees

GA Genetic Algorithms

MC Monte Carlo

MD Molecular Dynamics

mDH modified Denavit-Hartenberg

NMR Nuclear Magnetic Resonance

PCA Principal Component Analysis

PDB Protein Data Bank - Worldwide database of protein structures accessible online (`http://www.rcsb.org/`)

PRM Probabilistic Roadmap

RMSD Root Mean Square Deviation

RRT Rapidly-exploring Random Tree

SDAP Simultaneous Design And Path-planning

SDAP Simultaneously Design And Path-planning algorithm

T-RRT Transition-based RRT

THF 5,6,7,8-tetrahydrofolate

Voronoi regions A Voronoi region is associated to every vertex in a search space. It denotes the region of space that is closer to that vertex than to other vertices.

WCSP Weighted Constraint Satisfaction Problem

Introduction

Proteins probably are the molecules the most representative of life. They are present is all living cells, and they are involved in most of the biological processes. They fulfill a wide range of functions, such as catalysis, regulation, signaling, transport, storage, and structural functions. In addition to their primary importance in biology, proteins are also key items in other domains. They are pharmaceutical targets of drugs, their catalytic properties are exploited in biotechnology, and they are used as components of nano-devices in the rising field of bionanotechnology. Although the properties of natural proteins can be directly exploited in all these domains, the ability to create new proteins with improved properties or new functions is of major interest.

Proteins are complex molecules. These chains of amino acids are flexible structures whose shape, determined by the amino acid sequence, is strongly related to the function. Thus, designing a protein with the desired function consists in finding an amino acid sequence yielding the suitable structure. Considering the current knowledge on proteins and the experimental tools which are available today, achieving such a task is plausible. Yet, the number of possible sequences to test is so large, and the experimental cost of synthesizing and testing a single sequence is so high that it is necessary to resort to computational methods. Those methods, called Computational Protein Design (CPD) methods, cannot replace experimental testing. Nevertheless, they allow to guide the design process toward a narrow number of candidate sequences on which experimental ressources will be focused.

CPD methods have been developed for more than a decade and they have already permitted the creation of a few new proteins. Current CPD methods rely on a common approach that consist in finding, from a goal 3D scaffold, amino acid sequences that will fold into that scaffold. This problem is translated to an optimization problem. The main challenge lies in the nature and high dimensionality of the space to explore. This hybrid space has a discrete component, that corresponds to the set of all the possible amino acid sequences, and a continuous component, that corresponds to the possible configurations of the protein. Therefore, solving the optimization problem requires the use of algorithms allowing to efficiently explore such large spaces. In that regard, algorithms coming from the robotic community have demonstrated very promising abilities.

This thesis presents contributions toward the goal of solving such type of optimization problems in hybrid spaces. These contributions are both at a sampling level, to improve the efficiency of algorithms designed for exploring the conformational space of proteins (*e.g.* the space of protein's spatial arrangement), and also at the algorithmic level. The first chapter of this thesis provides some background on protein modeling and design. It first introduces the basics of protein systems modeling. Then, it gives an overview of state of the art algorithms used to explore the conformational landscape of proteins. Finally, the protein design problem, is

introduced together with the current approaches to solve it and the limitations they suffer.

The second chapter presents a framework to enhance the sampling of proteins conformational space using stochastic algorithms such as Monte Carlo methods. By using a mechanistic representation of proteins, involving segmentation into small fragments of three amino acid residues, this framework simplifies the conception of new local backbone perturbation methods. This framework is demonstrated by the construction of several Monte Carlo move classes, all operating on a common protein representation. These sampling techniques are then compared on two different protein systems.

The third chapter presents a comparison of four conformational space exploration algorithms. Two existing ones, the T-RRT algorithm and a simple MC simulation, and two new ones, adapted from the robot motion planning algorithm EST. An empirical comparative analysis shows how T-RRT is superior in its ability to quickly discover transition path between basins in the energy landscape of a protein.

Finally, the fourth chapter deals with an optimization problem that combines design and motion planning. The goal is to find the design (among a large set of possibilities) that optimizes the motion of the system between two given configurations. For this, the optimal path for all possible designs has to be searched. An algorithm to solve this problem is proposed and demonstrated on a simple academic system. Then, the application of the approach for designing a protein (or protein fragment) to perform a desired motion is investigated and discussed.

Contributions of the thesis

The work presented in this thesis is part of the development of a robotics-inspired algorithm library for structural biology. The previous developments performed by LAAS-CNRS in this context have shown the promising potential of these techniques for the study of protein flexibility [Cortés 2005]. A practical example of application of these methods to the simulation of protein-ligand unbinding is available at `http://moma.laas.fr/` [Devaurs 2013a].

In the *ProtiCAD* project founded by the *Agence nationale de la recherche* (ANR), the goal was to adapt robotics-inspired algorithms to be part of a CPD procedure taking protein flexibility into account. In this context, my scientific contributions are on multiple levels:

- *Enhancement of local conformation sampling:* We extended the sampling methods previously published in [Cortés 2012] by implementing a new move class (the *Hinge* move class) and by performing a comparative analysis of the different local move class involving the tripeptide decomposition model. This work led to the submission of a journal paper at JCTC [Denarie 2017].
- *Study of fast conformational landscape exploration methods:* We implemented two variants of the EST motion planning algorithm and combined them with

the transition test already used in the T-RRT costspace exploration technique [Jaillet 2010].

- *Adaptation of robotics-inspired algorithm for CPD:* We formalized a new optimization problem inspired from CPD and proposed an algorithm, called SDAP, to solve it. This work has been published at the Workshop on the Algorithmic Foundations of Robotics 2016 [Denarie 2016].

Chapter 1

Scientific Context

Contents

1.1 Proteins sequence and structure 5
1.1.1 Amino acids, peptides, and proteins 5
1.1.2 Sequence-function relationship 8
1.2 Protein modeling 8
1.2.1 Cartesian coordinates 9
1.2.2 Internal coordinates 9
1.2.3 Dimensionality reduction 10
1.2.4 Coarse grained models 11
1.3 Energy landscape 12
1.3.1 Physical theory 12
1.3.2 Energy functions 13
1.4 Computational methods for the exploration of protein conformations 14
1.4.1 Molecular dynamics 15
1.4.2 Monte Carlo methods 16
1.4.3 Robotics-inspired algorithms for the exploration of the conformational space 17
1.5 Computational Protein Design 21
1.5.1 The CPD problem 22
1.5.2 The search space 23
1.5.3 Current methods for CPD 24
1.5.4 CPD challenges 25

Recent advances in computational structural biology are in a large part due to the improvement of simulation algorithms combined with the evolution of protein modeling. This chapter presents the basics of protein constitution and working in order to understand the high complexity of those systems. Some focus is made on the different modeling methods that were developed over the years before introducing the notion of energy landscape. Then, an overview of the methods used to characterize or to explore the energy landscape is performed to finally present the computational protein design problem and explain the current approaches and the future challenges.

1.1 Proteins sequence and structure

1.1.1 Amino acids, peptides, and proteins

Amino acids are the building blocks of proteins. They contain an amine group ($-NH_2$), a carboxylic acid group ($-COOH$), and a connecting carbon atom, called α-carbon (C_α) , to which is attached a group of atoms called the side-chain as shown in Figure 1.1. The side-chain, denoted by R, determines the physico-chemical properties of each amino acid type. In nature, there are twenty different types of side-chains corresponding to twenty different types of amino acids. They are listed in table 1.1 with their one-letter and three-letter codes.

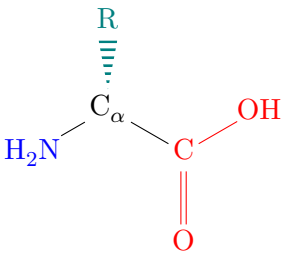


Figure 1.1: Representation of an amino acid with its α-carbon (black), its amine group (blue), its carboxylic acid group (red), and its side chain (green).

Alanine	*Arginine*	*Asparagine*	*Aspartic acid*	*Cysteine*
Ala	Arg	Asn	Asp	Cys
A	R	N	D	C
Glutamic acid	*Glutamine*	*Glycine*	*Histidine*	*Isoleucine*
Glu	Gln	Gly	His	Ile
E	Q	G	H	I
Leucine	*Lysine*	*Methionine*	*Phenylalanine*	*Proline*
Leu	Lys	Met	Phe	Pro
L	K	M	F	P
Serine	*Threonine*	*Tryptophan*	*Tyrosine*	*Valine*
Ser	Thr	Trp	Tyr	Val
S	T	W	Y	V

Table 1.1: Amino acids list with their 3-letter and 1-letter codes.

The amine group of an amino acid can react with the carboxylic acid group of another amino acid to form a peptide bond (see Figure 1.2). This process, called condensation, joins two amino acids together forming a dipeptide. When involved in a peptide or a polypeptide, amino acids are referred as residues. The amine group of the first amino acid and the carboxylic acid group from the second amino acid are preserved, so the condensation reaction can be repeated again and again to build a longer chain of amino acid residues. Such a chain is called a peptide,

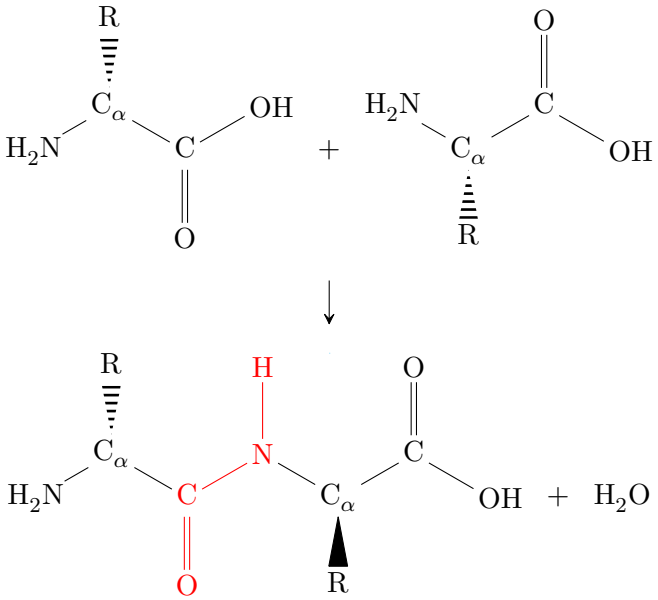


Figure 1.2: Formation of a peptide bond (red).

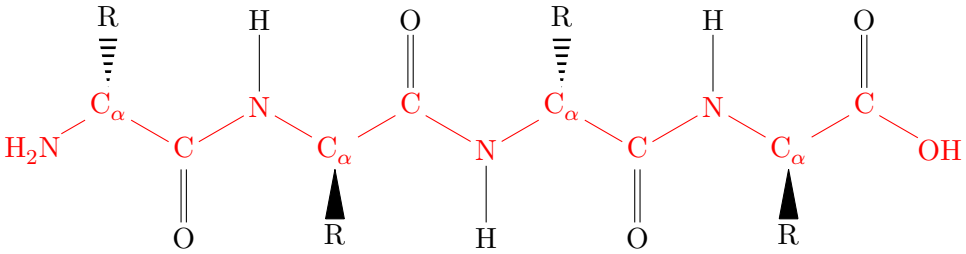
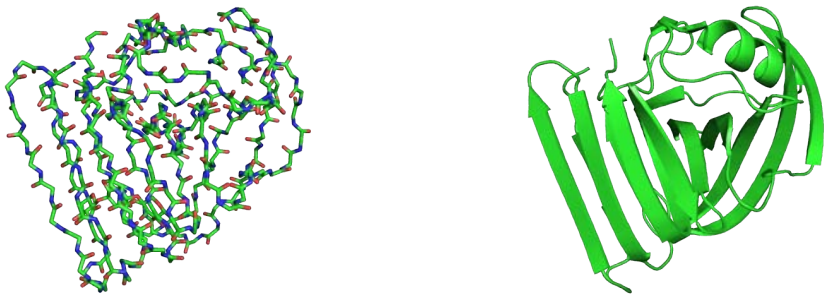


Figure 1.3: Backbone of a 4 residue peptide (in red). The N-terminus is on the left side of the figure while the C-terminus is on the right side of the figure.

or a polypeptide. A continuous thread of covalent bonds can be followed from the first to the last amino acid successively joining an amino nitrogen to an α-carbon, an α-carbon to a carboxylic carbon, and a carboxylic carbon to the next amino nitrogen. This chain of atom is called the backbone. The first residue's amine group is called the N-terminus, and the last residue's carboxylic acid group is called the C-terminus. A representation of the backbone can be seen in Figure 1.3. The sequence of a peptide/polypeptide is described by the list of the amino acids in the chain from the N-terminus to the C-terminus. It is usually represented using the one-letter code of the amino acids.

Peptides/polypeptides are flexible molecules and may take different spatial arrangements. Such an arrangement is called a conformation. Because of that, atoms from a residue may have interactions not only with adjacent residues' atoms and external atoms (from the solvent, a ligand, or another peptide) but also with atoms of other residues far away in the polypeptide's sequence. These interactions can form recognizable local structures, which are relatively stable and which are used



(a) (b)

Figure 1.4: Representation of a conformation of a protein (xylanase) using (a) stick representation (b) cartoon representation. In both figures, only the backbone is represented. The cartoon representation uses some symbol to recognize usual secondary structures like α-helices or β-sheet.

to simplify the representation of a molecule conformation. Figure 1.4 shows an example of such a representation.

Protein is the term employed to describe a polypeptide or a conglomerate of polypeptides bound together that have a biological function.

1.1.2 Sequence-function relationship

Proteins cover a very wide range of biological functions. They can be enzymes that catalyze some chemical reactions. They can be part of the process of signal transmission (this is the case of insulin for instance) or work as receptors of such a signal. They can bind other molecules called ligands, or they can dock on other macromolecules. They can even play structural roles. These different properties all rely on the protein having the correct spatial arrangement.

This functional spatial arrangement is called the native state. As explained later in paragraph 1.3.1, this state is not one rigid conformation but consists of an ensemble of conformations fluctuating around a stable energy minimum. The process of a protein passing from a random conformation to its native state is called *folding*. It is determined by the interactions of atoms in the different amino acids of the protein, relative to their interaction with the solvent. Therefore it is dependent on the sequence of the protein. In the same conditions, two proteins with the same sequence will, in general, always fold to the same biologically-active structure. Many small proteins, when denatured, spontaneously self-assemble into their native biologically-active structure [Anfinsen 1972] and many diseases are believed to be caused by mutations in a protein causing a misfold resulting in a dysfunctional spatial arrangement [Neudecker 2012, Soto C 2008].

1.2 Protein modeling

An appropriate mathematical representation of proteins is necessary in order to perform molecular simulations. It must be suitable to represent the spatial arrangement of the protein and to compute the physical properties while being computationally efficient. Many different representations have been created over the years. This section presents the most used ones.

1.2.1 Cartesian coordinates

The most straightforward representation to geometrically represent a protein is the cartesian coordinates representation. For a protein containing N atoms, a conformation C is represented by a vector $(A_{1x}\ A_{1y}\ A_{1z}\ ...\ A_{Nx}\ A_{Nx}\ A_{Nz})$ where $A_{ix}\ A_{iy}\ A_{iz}$ are the cartesian coordinates of atom A_i. These coordinates are sufficient for an atomistic description of the protein. Information about chemical bonds can be computed using the distance between atoms and the knowledge about their types. The cartesian coordinates model is generally used for energy calculation as energy functions need to compute distances between pairs of atoms (see paragraph 1.3.2). This model is used by the Protein Data Bank (PDB) [Berman 2000], a database of protein models built by scientists around the world from X-Ray and nuclear magnetic resonance (NMR) measurements.

However using this model to explore the conformational space can be inefficient. Each atom of the protein adds 3 degrees of freedom (DOF). For a protein containing N atoms, it results in a $3N$-dimensional conformational space. Even a small protein contains several hundred atoms. In addition, the cartesian coordinates of each atom do not generally change independently of the others due to bond geometry constraints. Searching through such constrained, high-dimensional space is very computationally expansive with classical search algorithms.

Another drawback of this model is that it is dependent on the chosen reference frame. For instance, recognizing two representations of the same protein with the same conformation is not straightforward if the reference frames are different. In order to compare two conformations, it is first necessary to align the two structures (using the method proposed in [Kabsch 1976] for example). This operation is computationally expensive.

1.2.2 Internal coordinates

The internal coordinates representation addresses the redundancy existing in the cartesian coordinates model. Knowing the atomic bonds of the protein, the conformation of the protein can be fully described using only bond lengths, bond angles, dihedral angles, and the position and orientation of a single frame attached to an atom (See Figure 1.5 for illustration):

- a bond length is the distance between two bonded atoms;
- a bond angle is the angle between two consecutive bonds;

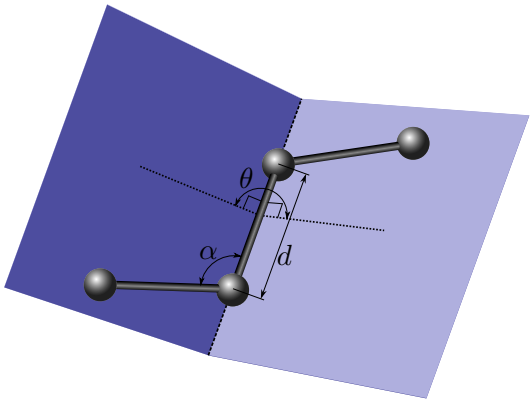


Figure 1.5: Illustration of the internal coordinate representation parameters. d is a bond length. α is a bond angle. θ is a dihedral angle.

- dihedral angle, also called bond torsion angle, is the angle formed by a group of four consecutively bonded atoms around the central bond.

Using forward kinematics, those parameters allow to recover the cartesian coordinates model when needed [Spong 2005].

This representation allows to reduce the number of DOFs by doing quite reasonable assumptions. A statistical analysis of protein structures reveals that bond lengths and bond angles are constrained to characteristic values at equilibrium. As a consequence, those parameters can be removed from the list of DOFs and considered constant [Scott 1966, Engh 1991]. This is called the rigid geometry assumption. Thus, the protein conformation is fully described by the vector of its dihedral angles. This representation is widely used by algorithms to sample the conformational space. It is worth noticing that the internal coordinates are not dependant on a particular reference frame, so the conformations of a protein can easily be compared using this model.

1.2.3 Dimensionality reduction

Internal coordinates together with the rigid geometry assumption drastically reduce the number of degrees of freedom required to model proteins. Yet, for typical problems involving proteins, it is common to reach more than 1000 dihedral angles. Exploring such a high-dimensional conformational space is challenging. Different strategies have been used to further reduce the dimensionality of the search space.

One possible approach is to use stronger assumptions and consider some subset of DOFs as constant. For example, in molecular docking problems, it was very common to consider that the protein is rigid and that only the ligand is flexible [Leach 2001] even though it has been shown that this assumption leads to unrealistic solutions [Cavasotto 2005]. Some more realistic assumptions based on prior knowledge of the protein [Jones 1997, Apostolakis 1998, Pak 2000] target the dihedral angles that contribute the most to the motions of the molecule and consider

the rest of the protein as rigid. Some studies have tried to automatically identify which parts of the protein can be considered rigid using methods based on rigidity theory [Thomas 2013].

A second approach to reduce the dimensionality of the search space is to map its DOFs into a lower-dimension space using statistical knowledge of the studied system [Fodor 2002, Van Der Maaten 2009]. Principal component analysis (PCA), for instance, can be used to analyze molecular dynamics simulations data (see paragraph 1.4.1) to capture important collective motion features [Mu 2005, Altis 2007]. Another method allowing to capture collective motion features is the isometric feature mapping (IsoMap) method [Tenenbaum 2000, Das 2006]. More recently, the locally scaled diffusion map method (LSDMap) was created to take into account local variation of molecular configuration space [Rohrdanz 2011]. Those methods require obtaining prior data on the system which can be a difficult process in itself.

A third approach to reduce dimensionality is to use normal mode analysis [Cui 2005]. It has been shown that large-amplitude motions in proteins are related to low-frequency normal modes [Hinsen 1998, Tama 2001]. This method does not require prior knowledge or data about the studied protein as the NMA can be performed from a single conformation. It has been applied to find transition pathways between conformations [Kirillova 2007].

1.2.4 Coarse grained models

A more radical approach to reduce the number of dimensions is to consider a coarse grained representation of the molecular system. All-atom representations of the protein allow for accurate energy calculation but at a high computational cost. Coarse grained representations sacrifice structural details in order to reduce the dimensionality of the search space and improve the speed of energy calculations. Such representations change both the coordinates of the conformational space and the corresponding potential energy model. Coarse graining may for example take into account only a few representative atoms of each residue. It allows to drastically reduce the number of DOFs in the problem without reducing the flexibility of the system. Of course, this comes at a cost. The loss of the full atom representation introduces inaccuracies that can yield unrealistic results.

Early coarse grained representations were lattice-based. C_α were the only represented atoms and were only allowed to lie on a lattice [Taketomi 1975, Yue 1995, Hinds 1994, Kolinski 1994, Unger 1993]. By making the conformational space discrete, those models pushed the limits of computational capabilities of protein modeling. They are still used to deal with very large protein systems out of reach of current computational capabilities of more accurate models [Dotu 2011].

Off-lattice coarse grained models are also very commonly used [Tozzini 2005]. They differ in how many atoms are represented in the backbone and in the side-chains. One-bead models use a single coarse grained atom to represent an entire amino acid residue. For instance, The Gō model is a very simple representation where one bead represents each amino acid at the position of its C_α [Taketomi 1975].

In this model, the interactions between beads are the attractive and repulsive forces based on the protein native structure. This native centric view of the protein makes this kind of model only useful in specific contexts. The Gō representation has been widely used in the protein folding community and many variations of this representation are still used [Clementi 2008]. Another example of one bead model is the BLN model. Each residue is modeled by one bead that can have one of three labels depending on its chemical properties (B for hydrophobic, L for hydrophilic, N for neutral). These labels mainly determine the interactions between beads [Oakley 2011]. The BLN model is described more in depth in Section 4.4.2. Two-bead models allow to roughly represent the side-chains and their interactions. A second bead is placed at the β-carbon or at the centroïd of the side-chain [Bahar 1997, Mukherjee 2004, Khalili 2004, Zacharias 2003]. Models with more beads also exist, each bead increasing the accuracy of the model while adding DOFs. For instance, the OPEP coarse grained model uses up to 6 beads to represent amino acids [Sterpone 2014]. The MARTINI coarse grained model also uses a variable number of bead for each type of amino acid [Monticelli 2008].

1.3 Energy landscape

1.3.1 Physical theory

Explicitly modeling protein dynamics is a complex process considering the size of those systems. In the classical approximation, Newton's laws theoretically allow to predict protein dynamics making it possible to fully understand their folding process or their interactions with other molecules. Given the position of every atoms in the system and their initial velocity, the energy of the system can be computed as the sum of the kinetic energy and potential energy. Newton's equations then give the dynamics of the system. The kinetic energy is only dependent on the momenta of the atoms in the system: it is a quadratic function of the atoms' velocities and masses. The potential energy, on the other end, is much more complex. It depends on the positions of the atoms, ie. on the configuration of the system, which in the absence of explicit solvent molecules is just the conformation of the protein. If we look at the potential energy as an altitude, the potential energy function can be seen as an hypersurface drawn over the conformational space. This hypersurface is called the potential energy landscape [Wales 2004, Schön 2009].

The potential energy term is so complex that obtaining a good approximation is a whole field of research, as will be explained in paragraph 1.3.2. Furthermore, analytically solving the Newton's equations of motion is not feasible for systems as complex as proteins. It is possible to numerically solve these equation with a very small time step (see paragraph 1.4.1). Yet, the study of the energy landscape of a protein gives important information about the states of that protein. Minima of the energy landscape correspond to locally stable, or metastable conformations of the system. Of course, it should be mentioned that atomic fluctuations actually forbid the system to stop in a single stable conformation. Instead, the system wiggles

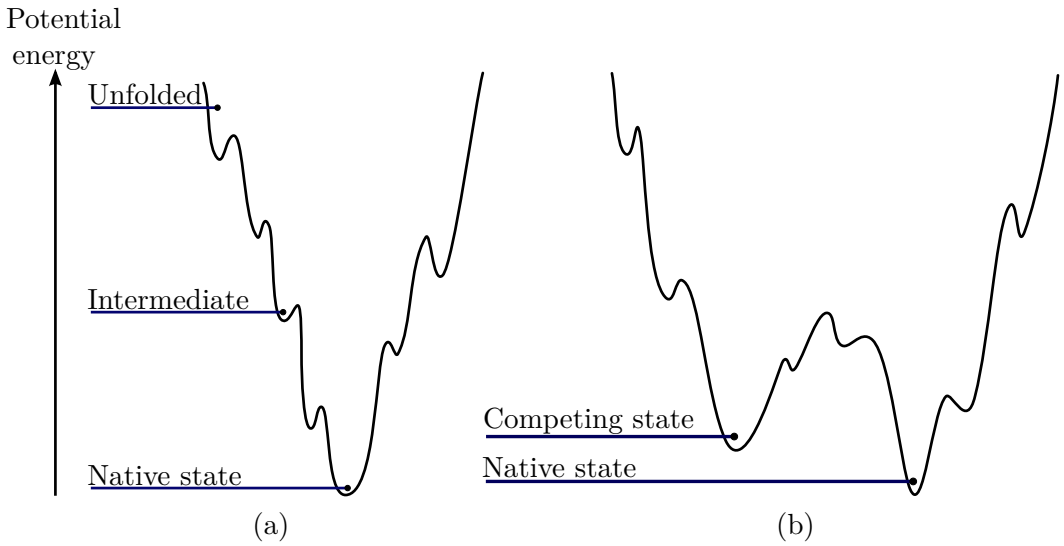


Figure 1.6: (a) Simplified view of a funnel-like energy landscape. High energy values correspond to unstructured state of the system while the lowest energy region corresponds to the native state energy basin. (b) Simplified view of a landscape with two competing low energy basins joined by a low energy saddle.

around an energy minimum and eventually jumps to another basin, slowly reaching a lower energy region in which the system will be trapped for a longer time. Low energy basins surrounded by high energy regions corresponds to the stable states of the system. Many proteins have a unique stable state corresponding to their biologically active state, ie. their native state. In this case, the underlying energy landscape is funnel-like with one deep energy basin corresponding to that native state. An example of a funnel-like landscape is shown in Figure 1.6 (a). Because of structural frustrations, the landscape is typically rugged with a high number of local minima [Onuchic 1997, Onuchic 2004]. Even though most proteins have a funnel-like energy landscape, some protein can have multiple competing low energy basins with eventually low energy transition paths between them [Okazaki 2006]. An example of a landscape with multiple accessible basins is shown in Figure 1.6 (b). Some other proteins even have very flat energy landscape with a lot of competing states. These proteins are called intrinsically disordered proteins.

1.3.2 Energy functions

Obtaining a good approximation of the potential energy surface is not trivial. Great effort are made to produce more and more accurate energy functions.

Energy functions are usually built as the sum of several terms:

$$E = E_{bond} + E_{angle} + E_{torsion} + E_{electro} + E_{vdW}$$

where E_{bond}, E_{angle}, and $E_{torsion}$ are terms concerning the local interactions of atoms constrained by atomic bonds. They respectively enforce the constraints on

the bond lengths, the bond angles, and the dihedral angles. $E_{electro}$ and E_{vdW} are terms concerning the long range interactions of atoms which are not neighbor in the molecular topology (not connected by one or two consecutive atomic bonds). $E_{electro}$ corresponds to the electrostatic interactions between atoms, and E_{vdW} corresponds to their van der Waals interactions [Bondi 1964]. Examples of well known physical force fields include the AMBER [Kollman 1997] and CHARMM [Brooks 2009] force-fields.

Protein systems are not lying in vacuum. They are surrounded by a solvent (usually water). Even though the contribution of the solvent to the potential energy can be taken into account using similar energy terms as those mentioned before, it highly increases the complexity of the system as every molecule of water must be modeled at the cost of additional DOFs and high computational time. A simpler way to deal with the solvent is to use an implicit model [Roux 1999]. The solvent molecules are omitted in the system representation and a specific energy term $E_{solvent}$ is added to the potential energy. $E_{solvent}$ may thus be a complicated function of the conformation of the protein. The most used implicit solvent models are the Coulomb/Accessible Surface Area (CASA), the Poisson-Boltzmann equation (PB), and the Generalized Born model (GB).

In addition to the physics-based energy functions, explained above, knowledge-based functions have been proposed as an alternative approach to evaluate molecular conformations. Knowledge-based energy functions, also called statistical energy functions, rely on the growing data that is available nowadays in the PDB to parametrize each term. The Rosetta energy function [Simons 1999, Das 2007] is an example of force field that mixes both physical terms and knowledge-based terms.

Even with relatively simple energy terms, the evaluation of the energy potential for a protein system is quite computationally expansive. This high cost is detrimental to protein sampling methods that extensively rely on energy evaluation. Many strategies can be adopted to improve the speed of energy calculations. An example of such a strategy is to perform a pairwise decomposition of the energy function [Dahiyat 1997, Gaillard 2014] for which internal energies of residues and interaction energies between pairs of residues are precomputed. The energy function is constructed in such a way that it is a sum of those internal energies and residue-residue interaction energy.

Coarse grained energy function were created together with the coarse grained models mentioned in paragraph 1.2.4 with the aim to reduce computational cost with respect to all-atom representations. They have the advantage to be much faster to compute and to yield much smoother energy landscape. Many coarse grained energy functions incorporate some knowledge about the native states of the protein, rewarding native contacts while penalizing non-native ones [Clementi 2008]. This kind of native structure bias is very commonly used in the field of proteins folding. On the other hand, some coarse grained energy functions do not incorporate prior knowledge of the protein structure to compute energy. This kind of function is especially used in *de-novo* structure prediction where they yielded several successes [Mukherjee 2004, Colubri 2004]. Nevertheless, although they are good alternatives

to all-atom energy functions computationally speaking, they introduce inaccuracies and often fail to discriminate good structures from bad ones (*e.g.* [Bowman 2009]).

1.4 Computational methods for the exploration of protein conformations

Different algorithms exist to explore the conformational space and the associated energy landscape of a protein systems. This section gives an overview of the most used ones, and also presents some more recent algorithms originating from the field of robotics.

1.4.1 Molecular dynamics

Molecular Dynamics (MD) methods simulate the classical dynamics of a molecular system in order to approximate *in silico* what could be the temporal evolution of the configuration of the system. They rely on a model, usually an all-atom one in cartesian coordinates, associated with a potential energy function. Starting from an initial configuration C_0 with randomly assigned initial velocities, it builds a trajectory of configurations [Frenkel 2001]. This trajectory can then be analyzed, for instance to observe a folding process or to understand what are the characteristic states of the system. The initial configuration of the system is built depending on the goal of the simulation. For example, if the goal of the simulation is to understand protein folding, an extended configuration of the protein will be built and used as the initial state. If the goal of the simulation is to observe the transition of the protein from one state to another, a configuration of the protein in one of those states (coming from X-ray diffraction, or from NMR for instance) will be chosen to start the simulation. The initial velocities are randomly chosen depending on the simulated temperature. At each step of the simulation, the configuration $C_{t+\delta t}$ of the system at time $t + \delta t$ is computed from the configuration C_t at time t by numerically solving Newton's equations of motion: accelerations of every atom can be computed from the gradient of the potential energy allowing to compute the new configuration and velocities of the system after a time δt.

MD simulations can be used in many different cases. An observable can be defined and measured during the simulation giving an estimate of its real value, or the sequence of configurations can be stored to be analyzed. A big advantage of MD simulation over other types of algorithms applied to sample a protein conformational space is that MD gives access to an actual trajectory of the system with configurations and energy values, but also with the underlying velocities. These data allow to compute many different properties such as the free energies which offer a statistical view of the energy landscape.

Although MD is very often used to study protein systems, it is computationally very demanding. The choice of the time step δt is crucial for the accuracy of the simulation and a typical choice for protein systems is a value around 1-2 femtosec-

onds. This very small time step is a big limitation. In practice, only simulations covering nanoseconds or a few microseconds can be performed in feasible time (typically several days/weeks). This duration has to be compared with the time scale of protein's reactions. For example, protein folding can last from a few milliseconds for the fastest proteins to several seconds for slower and larger ones. For other processes, like the transition of a protein from one state to another competing state, it can even be harder: the transition process might be quite fast, but the simulation may spend a substantial amount of time trapped in the energy basin of the initial state before the transition can be observed.

For proteins with complex energy landscapes, basic MD has very limited sampling capabilities compared to other algorithms. It spends most of its simulation time in low energy regions of the conformational space while interesting regions denoting state transitions have higher energies. Different methods have been developed to overcome this problem. The replica exchange MD (REMD) [Sugita 1999] reproduces the ideas of the parallel tempering method [Geyer 1991] (See 1.4.2) to apply them to MD: several isothermal MD simulations are run with different temperatures and the simulations are regularly swapped between temperatures with some acceptance probability. REMD allows crossing high energy barriers and has been widely used for protein simulations [Periole 2007, Zhang 2005, Nguyen 2005, Beck 2007]. Another approach that allows MD to cross high energy barriers is steered MD (SMD) [Suan Li 2012, Park 2004]. This methods simulates a pulling force that will hopefully cause the conformational changes necessary to cross the energy barrier and observe the desired trajectory. Metadynamics is another powerful method that improves MD sampling capabilities [Laio 2002]. It is based on two ideas: a dimensionality reduction is performed by the use of carefully chosen collective coordinates, and the energy force field is biased by the addition of a Gaussian term that progressively fills already explored regions of the landscape.

1.4.2 Monte Carlo methods

The Monte Carlo (MC) method [Metropolis 1953] is a stochastic algorithm. It explores the conformational space with a random walk favoring low energy regions. Starting from an initial configuration, a sequence $C_1, \ldots, C_n$ of configurations is built. At each step, or move, the last configuration C_t is randomly perturbed. The obtained configuration $C_{candidate}$ is accepted or rejected with a probability P such as

$$P = \begin{cases} e^{\frac{-\Delta E}{k_b T}} & \text{if } \Delta E > 0 \\ 1 & \text{otherwise} \end{cases} \tag{1.1}$$

where ΔE is the potential energy variation from C_t to $C_{candidate}$, k_b is the Boltzmann constant, and T is a temperature parameter. This test is called the Metropolis Criterion. The temperature parameter T allows to control the greediness of the exploration. At low temperature, the simulation will quickly converge to a nearby

energy minimum but won't be able to cross high energy barriers, while at high temperature, the simulation will be able to occasionally cross high energy barriers, exploring a larger region of the energy landscape at the cost of a longer convergence time.

Unlike MD where the changes between two consecutive conformations are computed from Newton's laws of motion, the perturbations performed at each MC step are random. They do not even have to be realistic moves as long as they are coupled with the Metropolis Criterion mentioned earlier. Nevertheless, the choice of a move scheme strongly affects the efficiency and the quality of the sampling. The moves should provide a good coverage of the conformational space while being computationally efficient and with a good acceptance rate. This subject will be discussed more in depth in chapter 2. For chain molecules such as proteins, a few standard move classes can be mentioned. The pivot move perturbs a single dihedral angle randomly chosen in the polypeptide chain. This move is the most popular and simple one. The concerted rotation is another type of move which has the particularity to be local and only affect a small number of atoms in the system: a dihedral angle from the main-chain of the polypeptide is randomly perturbed and the six following dihedral angles are computed in order to ensure that the succeeding atoms of the chain do not move (this is in general possible and will be explained more in detail in chapter 2).

Although MC has higher sampling capabilities than MD, it nevertheless loses kinetic information. The resulting trajectory cannot be considered as an actual trajectory but only as a sampling of the conformational space and only statistical analysis of the results can be performed. However, when performed carefully, ie. when MC moves satisfy detailed balance[1], the distribution of the conformations is guaranteed to follow the Boltzmann distribution and statistical properties of the system, like free energies, can still be computed accurately.

Even though MC simulation is generally faster than MD to explore the conformational space, processes like protein folding or like transitions between states are still very hard to observe using this method. Similarly to what is done in MD, the replica exchange MC (or parallel tempering) method simultaneously runs multiple MC simulations with different temperatures [Swendsen 1986, Earl 2005]. In this context, each simulation is called a walker. Regularly, a swap of temperature is tried between the walkers. Other variants can be cited: umbrella sampling [Torrie 1977] and energy landscape flattening [Zhang 2002] try to bias the transition test to favor transitions between energy basins while basin hopping [Wales 1997] and simulated annealing [Kirkpatrick 1983] aim at finding the global minimum of the energy landscape.

[1] Detailed balance requires that each transition $x \to y$ is reversible, *i.e.* for every pair of states x, y, the probability of being in state x and transitioning to state y must be equal to the probability of being in state y and transitioning to state x, $P(x)P(y|x) = P(y)P(x|y)$.

1.4.3 Robotics-inspired algorithms for the exploration of the conformational space

The exploration of the conformational space of a protein has similarities with another widely studied problem from the robotics community: the motion planning problem. This problem, whose goal is to compute the motion to take a robot from one configuration to another, has been the subject of active research for more than forty years [Latombe 1991, Choset 2005] and has yield significant advances in domains such as industrial manufacturing and computer animation. The algorithms solving motion planning problems are called planners.

A parallel can be drawn between the notions of configuration space in robotics and conformational space in structural biology. The configuration space [Lozano-Perez 1983] is the space of all the possible configurations that a robot can take. Its dimension depends on the chosen representation for the robot and it is usually constrained by obstacles. A protein can be considered as a robot with multiple articulated bodies. Therefore, the configuration space of that robot corresponds to the conformational space of the protein system and the obstacles are the regions of the conformational space where the protein is in collision with itself or with other molecules. The similarities can even go further when we notice that the internal coordinates representation is actually very similar to the representation of an articulated kinematic chain in robotics.

Those similarities have been used to apply robotics-inspired algorithms to solve computational structural biology problems since the 1990s [Parsons 1994] and many adaptations of motion planning algorithms have been created in recent years [Moll 2008, Al-Bluwi 2012]. These algorithms are mainly variants of the Probabilistic Roadmap (PRM), the Rapidly-exploring Random Tree (RRT), and the Expansive-Spaces Trees (EST). The basic principles of these three algorithms are presented below. As the goal of the motion planning problem is to find the trajectory between two configurations, the main way they are applied in structural bioinformatics is to find transition trajectory between different conformations of proteins. Though, as will be explained in Chapter 3, these algorithms can be adapted to explore the energy landscape.

1.4.3.1 PRM

The Probabilistic Roadmap algorithm [Kavraki 1996] is a stochastic algorithm introduced in the 1990's. If many variations of the PRM algorithm have been created over the years to improve its efficiency ([Amato 1998, Wilmarth 1999, Siméon 2000, Sánchez 2003, Geraerts 2004]), the basic principle remains the same. Adaptations of the PRM to work on molecular systems were developed very early [Singh 1999, Amato 2003] and the framework has been improved to work with large proteins [Thomas 2005, Thomas 2007, Molloy 2014]. The algorithm works in two separate phases: the roadmap construction, and the query phase.

Roadmap construction phase First, a roadmap is built by performing random samples from the configuration space. Sampled configurations are checked for collisions and collision-free configurations are added to the roadmap as nodes. This process is repeated until n nodes have been created. Then, for each node, the k closest neighbors are identified and a local planner is called to try to connect each neighbor to the node. When the connection is successful, an edge is added to the roadmap.

This process builds a graph, called the roadmap (see Figure 1.7), which tends to cover the whole configuration space and gives connectivity information between configurations. If the construction principle of the roadmap is very simple, it is nevertheless the most important step of the PRM algorithm and every operation must be performed carefully:

1. Collision checking is quite straightforward for a robot system, but when working with a molecular system, a simple collision test is not sufficient to ensure a realistic conformation. The potential energy needs to be taken into account to ensure more realistic conformations, for example, by choosing a rejection threshold.

2. A uniform sampling of the configuration space is a good approach for low dimensional problems, but in many cases, and especially in the case of protein systems where the conformational space is very high-dimensional, a biased sampling is required. For instance, in [Molloy 2014], samples are built using a database of backbone fragments from native protein structures to increase the chances of sampling low energy conformations. Furthermore, the fragments are chosen using a heuristic that aims to maximize conformational space coverage.

3. The local planner that determines if two nodes can be connected is also critical. A simple strategy could be to do a linear interpolation on each DOF between the two nodes and to check for collisions/energy with a regular check step. More sophisticated approaches can be adopted involving local sampling and probabilistic transition test like the Metropolis Criterion (1.1).

Query phase The goal of the query phase is to use the roadmap built in the first stage of the algorithm to solve the motion planning problem. The initial and the goal configurations are connected to their nearest neighbors in the roadmap and a graph search algorithm such as Dijkstra's shortest path [Dijkstra 1959] or A* [Hart 1968] is used to find the shortest path connecting the two configurations. The particularity of the PRM algorithm is that the same roadmap can be used for multiple queries potentially saving a lot of computing time.

However having a good configuration space coverage can be a difficult task. If the initial or the goal configuration cannot be connected to the roadmap, or if they fall in disconnected components of the roadmap, the query fails. Though,

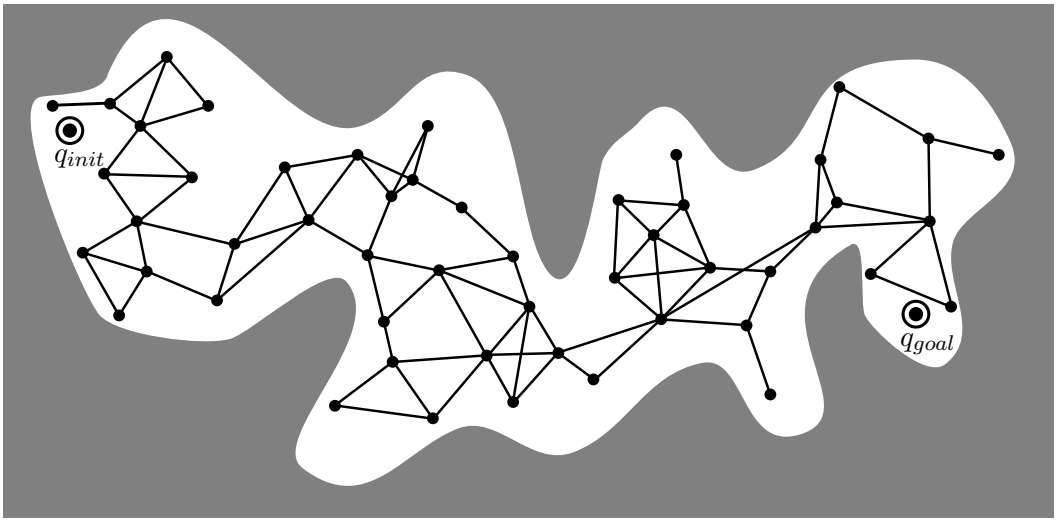


Figure 1.7: Illustration of a PRM roadmap. The white areas represent collision-free regions of the configuration space.

it does not mean that there is no realistic path between the two configurations. For that reason, the PRM algorithm is not a complete algorithm. PRM is said to be probabilistically complete, ie. when the number of sampled nodes approaches infinity, the probability that PRM finds a solution if one exists approaches 1.

1.4.3.2 RRT

The RRT algorithm is a tree-based motion planner [Lavalle 1998, LaValle 2001]. Starting from an initial configuration, it iteratively grows a tree of configurations until the goal configuration can be connected to the tree. Once this condition is met, a simple tree search gives the solution path. The exploration strategy of RRT uses an implicit Voronoï bias to quickly expand toward unexplored regions of the space [Lindemann 2004]: a random configuration q_{rand} is sampled from the configuration space and the next candidate node q_{new} is created by moving an incremental distance δ from the node q_{near} in the direction of q_{rand} (see Figure 1.8), *e.g.* using a linear interpolation. This new configuration is accepted if it is collision free and if it can be connected to q_{near} using the local planner. The details of the algorithm and the adaptations for molecular simulations are explained in Chapter 3. The RRT algorithm has been shown to be probabilistically complete [LaValle 2001]. Many variants of this algorithm have been developed to improve its efficiency and/or to treat specific problems. We will only mention RRT-connect [Kuffner Jr 2000], real-time RRT [Bruce 2002], resolution complete RRT [Cheng 2002], obstacle based RRT [Rodriguez 2006], or RRT* [Karaman 2010]. Two variants of the RRT algorithm have been developed with a particular focus on problems coming from structural biology: ML-RRT [Cortés 2008, Cortés 2010b], and T-RRT [Jaillet 2008, Devaurs 2013b].

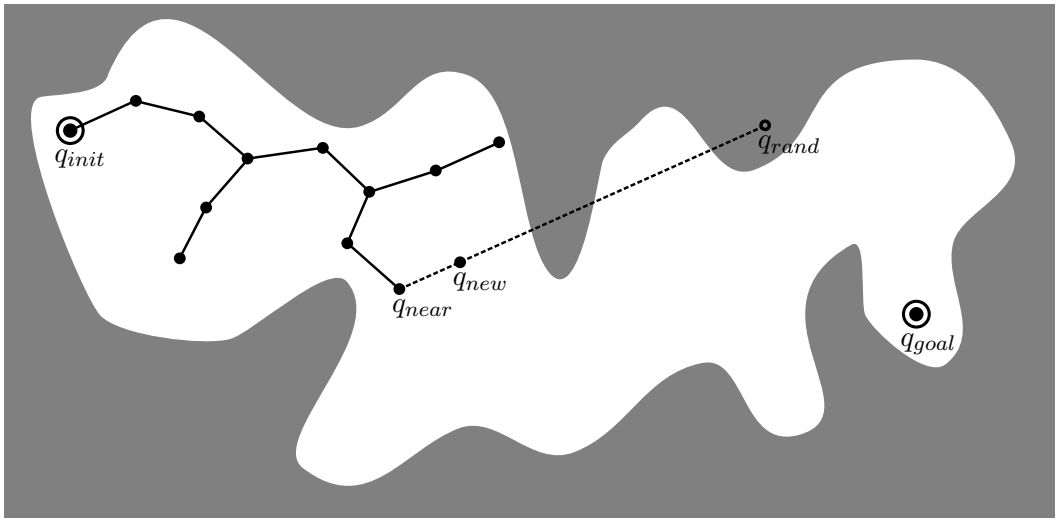


Figure 1.8: Illustration of an iteration of the RRT algorithm. A node q_{rand} is sampled and the closest node in the tree q_{near} is pulled toward q_{rand} to create q_{new}. If q_{new} is collision free, it is added to the tree. This process is repeated until q_{goal} can be connected to the tree.

1.4.3.3 EST

Similarly to the RRT algorithm, the EST approach grows a tree of configurations in the search space [Hsu 1997, Hsu 2000, Hsu 2002]. At each iteration, a configuration q is chosen in the tree with some probability $P(q)$. Then, a random configuration q_{new} is sampled from a uniform distribution in the neighborhood of q (see Figure 1.9). If this configuration is collision-free and can be connected to q using the local planner, it is added into the tree. The EST algorithm has also been shown to be probabilistically complete [Hsu 2000]. EST is a very general algorithm as the choice of the probability function P that will determine which node will be extended can be adapted depending on the goal. Furthermore, if the basic version of EST samples q_{rand} from a uniform distribution in the neighborhood of q, this step can easily be biased to yield a more efficient sampling of the search space. This is particularly interesting to explore the high-dimensional conformation space of protein systems where the sampling can be restricted to realistic moves. Chapter 2 gives some examples of such kind of moves and an example of implementation of EST will be developed in Chapter 3. A successful variant of the EST algorithm is KPIECE [Sucan 2012]. In this version of the algorithm, the probability function P is built in such a way that nodes that are the most likely to improve the space coverage are chosen more often.

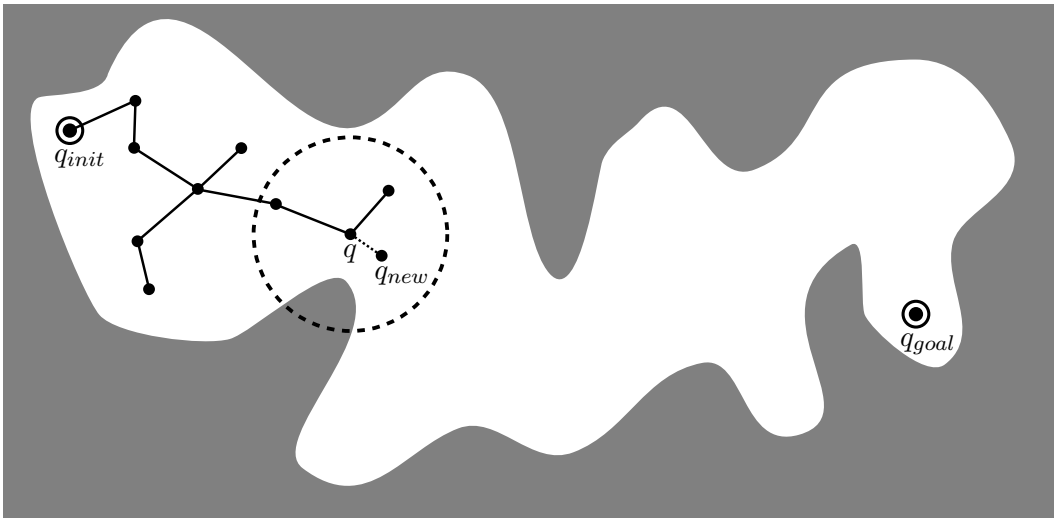


Figure 1.9: Illustration of an iteration of the EST algorithm. A node q_{new} is sampled in the neighborhood of a chosen node q. This process is repeated until q_{goal} can be connected to the tree.

1.5 Computational Protein Design

Protein design is the process of finding the amino acid sequence to build a protein with the desired function. Considering the huge number of possible designs (there are 20^N possible sequences to try for a N amino acid long protein), it is not possible in practice to synthesize and test all of them. Computational protein design (CPD) has been developed as a tool to identify the most promising candidates and is probably the most practical option to speed up that process. Over the last decades, significant progress has been made, from early redesign of protein cores [Hurley 1992, Harbury 1995, Desjarlais 1995, Betz 1996, Dahiyat 1996] to complete *de novo* protein design [Dahiyat 1997, Kuhlman 2003]. A good review of the progress made up to a decade ago is presented in [Lippow 2007]. The range of problems addressed by CPD has been extended to the improvement of protein drug properties [Luo 2002], the redesign of protein-protein interfaces [Clark 2006, Fleishman 2011] and the creation of a metal-protein interface [Yosef 2009, Der 2012]. Other important results are presented in [Röthlisberger 2008, Jiang 2008, Siegel 2010].

1.5.1 The CPD problem

As we explained in section 1.1.2, the interaction of a protein with its environment is mainly determined by the spatial arrangement of its composite atoms. For this reason, the CPD problem is usually expressed as follows: find sequences of amino acids that will fold into the desired spatial arrangement. Solving this problem is very complex and is usually decomposed in four different stages:

1. The first stage is to define the protein scaffold, which will in turn specify

the goal arrangement of the protein. This stage requires expert knowledges of proteins and of the desired interactions. Therefore, it is hard to fully automate this stage. It is usual to rely on known protein scaffolds and only concentrate on designing the geometry of the active site. This phase also defines the constraints on the protein, like required amino acids near the active site, or a set of mutable amino acid positions.

2. The second stage is the search for the sequence that will fold into the defined scaffold. Since solving the folding problem while exploring the sequence space is extremely complex, CPD solves a simplified instance The addressed problem consists in searching for the sequence that will best stabilize the goal structure. This search is expressed as an optimization problem. An objective function encoding the stability of the goal structure is defined, and the solution sequence is the one that minimizes this function. Of course, it does not guarantee that the protein will actually fold into the desired structure. Therefore, in general, the search usually looks for multiple probable solutions with values close to the global minimum.

3. The third stage is the analysis of the results. It consists in performing molecular simulations (such as MD, MC, etc.) in order to check how the candidate sequences perform in more realistic conditions. Depending on the result of this stage, the initial problem might need to be reworked and another iteration of stages one and two might be needed.

4. Finally, the most promising candidate sequences can be synthesized for experimental validation. The feedback from this stage can be used to refine the solution sequence, or to step back to previous stages of the process.

In the next sections, the focus will be on the second stage of the CPD process, which is the most related one with this thesis.

1.5.2 The search space

The space explored during the second phase of the CPD process is the product of two heterogeneous components:

- A discrete component corresponding to all the possible amino acid sequences. For instance, if the CPD problems has to find the amino acid for every position in the sequence in a 100-amino-acid-long protein, it results in 20^{100} possible sequences. The reader should notice that exploring such a space entirely is out of reach of current computational capabilities.

- A continuous component corresponding to the conformational space for each of the possible sequences. The dimension of this component corresponds to the number of DOFs in the chosen representation of the protein.

An important part of the work in the definition of a CPD problem is to reduce the search space in order to make the problem tractable. This work includes the selection of mutable residues, the constraints on the amino acid types depending on the position, and the conformational variability of the side-chains: which side-chain are actually allowed to move. Other approximations are usually applied to further reduce the complexity of the problem.

First, a statistical analysis of the amino acid side-chain conformations in protein databases reveals that side-chains only populate a reduced number of clusters around low-energy conformations. This result was exploited to build databases of side-chain conformations called rotamers. The use of these rotamer libraries transforms a part of the continuous component of the search space into a discrete component where each amino acid has a limited number of possible conformations. It should be mentioned that several works relax this approximation by considering continuous rotamer libraries where fluctuations of side-chains around their equilibrium positions are taken into account [Gainza 2012].

A second common simplification is the fixed backbone approximation [Ponder 1987]. As the CPD problem aims at optimizing the sequence for a particular backbone conformation, it makes sense to only consider the side-chain variability and to try to fit the amino acids on the backbone conformation corresponding to the designed scaffold. Combined with the previous approximation, the CPD problem is reduced to a search for the optimal rotamers to fit a given backbone. Although these simplifications have enabled significant advances in the CPD community, the treated problem is nevertheless unrealistic. Backbone fluctuations generally have stronger effects on energy than side-chains have. Furthermore, the fact that a sequence minimizes the objective function does not guarantee that the backbone conformation is actually stable for that sequence. It might just lie on the slope of a basin corresponding to a different stable state in the energy landscape.

Several methods have been developed to take the flexibility of the backbone into account during the CPD process [Georgiev 2007, Fung 2007, Hu 2007, Murphy 2009]. For example, the multi-copy backbone approach simulates the flexibility of the backbone by incorporating into the objective function the stability of an ensemble of backbone conformations which are considered to be representative of the designed state [Fung 2008]. Efficient sampling techniques, like the ones mentioned in paragraph 1.4.3, can also be used to explore the conformational space. For instance in the approach presented in [Kuhlman 2003], sequence optimization phases are alternated with backbone optimization phases.

1.5.3 Current methods for CPD

Even with the approximations mentioned previously (discrete rotamer library and fixed backbone approximation) and using a pairwise decomposable energy function, finding the set of rotamers that minimizes the objective function has been shown to be a NP-hard problem [Pierce 2002]. Both deterministic and stochastic algorithms exist to solve this problem [Wernisch 2000].

1.5.3.1 Deterministic algorithms

The Dead-End Elimination algorithm (DEE) prunes the search space by iteratively removing rotamers that can be proven not to be part of the optimal solution [Desmet 1992]. The algorithm iterates until no more dead-end rotamer can be found. Although the DEE algorithm does not always reduce the space to one single rotamer sequence, it nevertheless highly reduces the search space allowing a complete algorithm like A* [Hart 1968] to be used to find the lowest energy sequence [Leach 1998]. The DEE algorithm has been extended to improve its efficiency and the range of problem it can tackle. More advanced criteria have been added to the original pruning criterion, and variants of the algorithm allow to include some conformational variability [Goldstein 1994, Pierce 2000, Georgiev 2006, Georgiev 2007, Georgiev 2008].

The Cost Function Network (CFN) is an extension of a mathematical model called Constraint Network where constraints are replaced by cost functions. The CPD problem can also be modeled in a CFN and formulated as a weighted constraint satisfaction problem (WCSP) where the goal is to find the set of variables (the rotamers) that minimizes the sum of all cost functions (representing the objective function). The algorithm implemented in *toulbar2* demonstrated an important speed-up compared the the DEE/A* algorithms [Allouche 2012, Traoré 2013, Allouche 2014].

1.5.3.2 Stochastic algorithms

If deterministic algorithms have the advantage to guarantee that the best sequence will be found, the complexity of CPD problems make them unusable to design big protein systems. Stochastic CPD algorithms usually cannot guarantee that the best solution will be found, but they are able to find some candidate solutions in a relatively short time [Voigt 2000]. Considering that, with all the approximations made to solve a CPD problem, finding the sequence that minimizes the objective function does not guarantee that this sequence will actually fold into the goal structure, it makes sense to consider other sequences with a good score as equally valid candidate for the design.

The most common stochastic algorithms in CPD use the MC method with its numerous variants. They are similar to the MC method explained in Section 1.4.2 except that they work at two different levels. One level corresponds to the sequence space, where each MC step is performed by changing the sequence and where the Metropolis Criterion (1.1) is applied with the objective function instead of the potential energy function. The other level corresponds to the conformational space, aiming to converge to a minimum energy conformation for the current sequence [Polydorides 2011].

Another type of algorithm used in CPD is based on Genetic Algorithm (GA). GA works by maintaining a population of candidate solutions that will be slowly improved at each iteration by performing operations derived from the biological process of natural selection, including mutations, selections, and crossovers. A

good description of GA can be found in the literature [Weise 2009]. It has been applied with success in CPD [Jones 1994, Desjarlais 1995].

Finally, FASTER is an example of algorithm that combines stochastic and deterministic approaches. This combination allowed to find solutions nearly identical to the optimum in a very low run time [Desmet 2002, Allen 2006].

1.5.4 CPD challenges

During the last decade, CPD has achieved significant breakthroughs. The increasing computational capacities combined with the development of more efficient algorithms and energy functions with more realistic models allowed to design proteins involved in complex interactions [Suárez 2009]. One of the main challenges that remains open is the multiple objective design. Proteins binding to other proteins, or to ligands are often subject to conformational changes between their bound and unbound states. In this context, solving the CPD problem implies finding a sequence for which both the unbound and the bound states of the protein are stable and for which the transition between those two states is achievable. Of course, the number of sequences that will satisfy these conditions is much smaller than the number of sequences that will stabilize only one conformation. To solve this problem, current approaches use a multi-objective function that will balance the score relative to the unbound state stability with the score relative to the binding capacity of the protein [Suárez 2008, Suárez 2010]. Nevertheless, this approach cannot guarantee that the two bound and unbound states are actually reachable (a high energy barrier might prevent the transition between the two states to happen), nor does it guarantee that the two states are actually stable.

In fact, those limitations are linked to the current formulation of the CPD problem both for single and multiple objective designs. The objective function being minimized aims at finding a sequence that will minimize the potential energy for the designed states without any guarantee that those states correspond to actual minima in the conformational space corresponding to that sequence and without any knowledge about the possible transitions between those states. A better objective function would need to take such kind of information into account. Except in some rare cases where some transition states are known [Neudecker 2012, De Simone 2015] and can be accounted for, the solution to those problems implies the use of a design procedure that includes some degree of characterization of the conformational landscape. In fact, current procedures only aim at designing static states of proteins, whereas dynamic processes are implicated in many function such as binding of a ligand or a protein, release of a product, or allosteric transitions. A design process that would take into account protein dynamics would open new possibilities.

Chapter 2

Protein modeling and local conformational sampling

Contents

2.1 Mechanistic model 28
2.2 Tripeptide decomposition 29
2.3 Devising move classes 30
2.3.1 Perturbing particles 31
2.3.2 Solving inverse kinematics for a tripeptide 33
2.4 Results 34
2.4.1 Implemented move classes and parameter settings 34
2.4.2 Test systems 36
2.4.3 Computational performance 36
2.4.4 Distribution of sampled states 37
2.4.5 Exploration efficiency analysis 40
2.5 Conclusion 44

This chapter presents an approach to enhance conformational exploration methods. It is based on a mechanistic view of proteins. The idea is to cut the protein into small fragments of three amino acid residues, which we refer to as *tripeptides*. Each fragment can be represented as a kinematic chain, similar to a robotic manipulator. Such a representation enables the conception of effective methods to locally deform the protein model, which preserve bond geometry, using closed-form inverse kinematic solvers. Although this chapter focuses on a specific application of this approach for devising MC move classes, the tripeptide-based representation can be exploited within other types of methods.

One of the main difficulties involving the application of the MC method to proteins consists in devising suitable trial move classes for complex chain-like molecules. As was mentioned in Chapter 1, an effective move class should yield a good acceptance rate, while enabling the exploration of large regions of the conformational space. Several types of trial move classes have been proposed over the years to enhance the efficiency of MC methods applied to proteins. The approach presented below enables devising different types of move classes that can be easily implemented using a unique molecular representation and a single inverse kinematics solver.

This chapter presents the general aspects of the mechanistic protein representation using the tripeptide decomposition. Then, it explains how to implement several move classes based on this representation. The performances of these move classes is then analyzed through several tests using different types of proteins. This chapter is an extension of a preliminary work [Cortés 2012]. A new move class (the *Hinge* move) is implemented and the analysis of the different move classes goes more in depth with the addition of more quantitative metrics like the time dependent RMSD function and autocorrelation.

2.1 Mechanistic model

In the following sections and chapters, the internal coordinate representation with the fixed bond lengths and angles of the rigid geometry assumption will be used to model proteins (see Section 1.2). In these conditions, the dihedral angles are the only degrees of freedom of the molecule. Figure 2.1 illustrates this model. An additional assumption that will be made is to consider that double bonds, such as peptide bonds in proteins, are rigid connections (ie. the dihedral angles ω_i associated with the peptide bonds are constant). In summary, the variable parameters that define the conformation of a protein backbone are the pairs of dihedral angles, ϕ_i and ψ_i, of all its amino-acid residues. The conformations of the side-chains are determined by a variable number of dihedral angles $\chi_i^{(k)}$. As will be mentioned later in Section 2.3, both the rigid geometry and the rigid peptide bonds assumptions can be relaxed to allow small variations, which can be important from a structural point of view [Ulmschneider 2004].

Using the internal coordinate representation described above, proteins can be modeled as articulated mechanisms. The bodies of the mechanism correspond to rigidly bonded atom groups, and the joints are the bond torsions. The kinematic chains corresponding to the protein backbone and side-chains can then be modeled using standard conventions usually applied in robotics. In the following, the modified Denavit-Hartenberg (mDH) convention is used [Craig 2004]. Following this convention, a Cartesian coordinate frame F_i is attached to each rigid atom group. The relative location of frame F_i relative to F_{i-1} can then be defined by a homogeneous transformation matrix of the form:

$$ {}^{i-1}T_i = \begin{pmatrix} \cos\theta_i & -\sin\theta_i & 0 & a_{i-1} \\ \sin\theta_i\cos\alpha_{i-1} & \cos\theta_i\cos\alpha_{i-1} & -\sin\alpha_{i-1} & -d_i\sin\alpha_{i-1} \\ \sin\theta_i\sin\alpha_{i-1} & \cos\theta_i\sin\alpha_{i-1} & \cos\alpha_{i-1} & d_i\cos\alpha_{i-1} \\ 0 & 0 & 0 & 1 \end{pmatrix} $$

The elements of ${}^{i-1}T_i$ depend on the bond geometry. Assuming constant bond lengths and bond angles, the mDH parameters d_i and α_{i-1} are constant and can be computed from the initial conformation of the system. The bond torsion angle θ_i is the only variable parameter. Using this convention, the cartesian coordinates model of the molecule can be computed from a simple succession of transformation

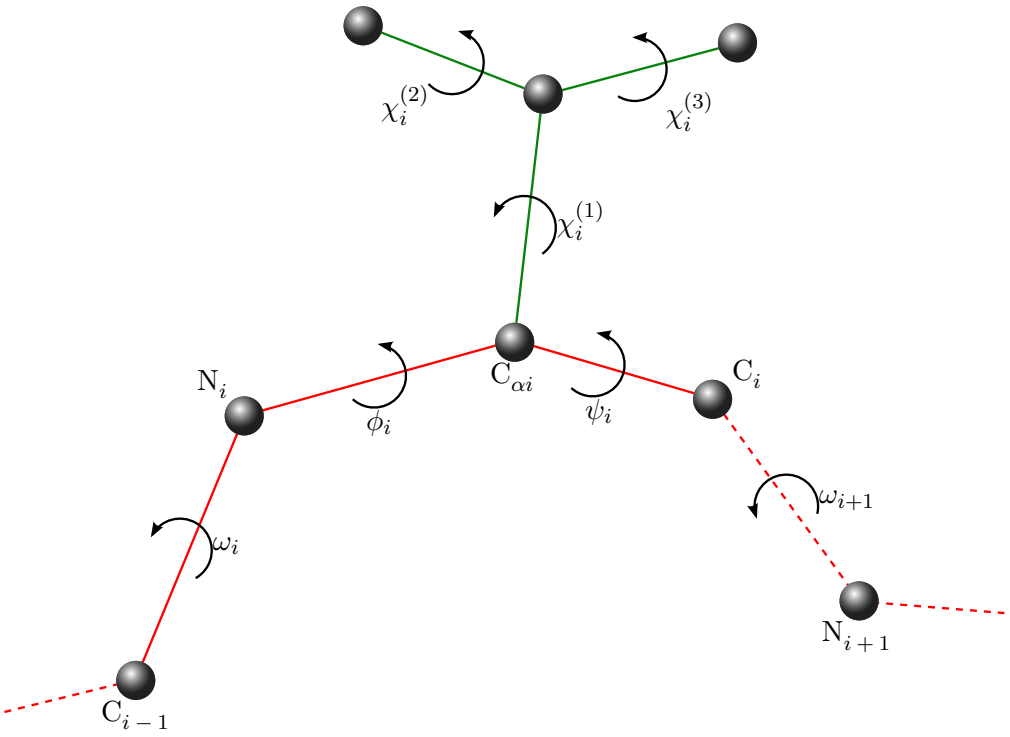


Figure 2.1: Illustration of the dihedral angles of residue i in a peptide. The backbone is represented in red and the side-chain in green. The backbone of residue i contains 3 bond torsion angles ω_i, ϕ_i, and ψ_i. The number of bond torsion angles defined by the side-chain i depends on the type of the amino acid i. They are denoted $\chi_1, \ldots, \chi_k$ (for instance, $k = 3$ in this figure).

matrix multiplications. This is called forward kinematics in the robotics context.

2.2 Tripeptide decomposition

The main idea explained in this section is the segmentation of the protein chain into fragments of three amino acid residues, which we refer to as *tripeptides*. The reason for choosing such a subdivision is that each tripeptide backbone involves six degrees of freedom (three pairs of ϕ, ψ angles), corresponding to the shortest fragment with full instantaneous mobility of the end-frame relatively to the base-frame (i.e. the relative position and orientation of the two frames can change in any direction). Figure 2.2 illustrates this idea. Figure 2.2 (a) shows a protein model with a cartoon representation of the backbone embedded in the model of the protein surface. Figure 2.2 (b) represents the protein backbone trace with the frames corresponding to the ends of the tripeptides. Figures 2.2 (c) and 2.2 (d) represent the chemical and the mechanistic models of the backbone of a single tripeptide, respectively. As depicted in the figure, the tripeptide backbone can be seen as a robotic manipulator with six revolute joints. The base of the manipulator

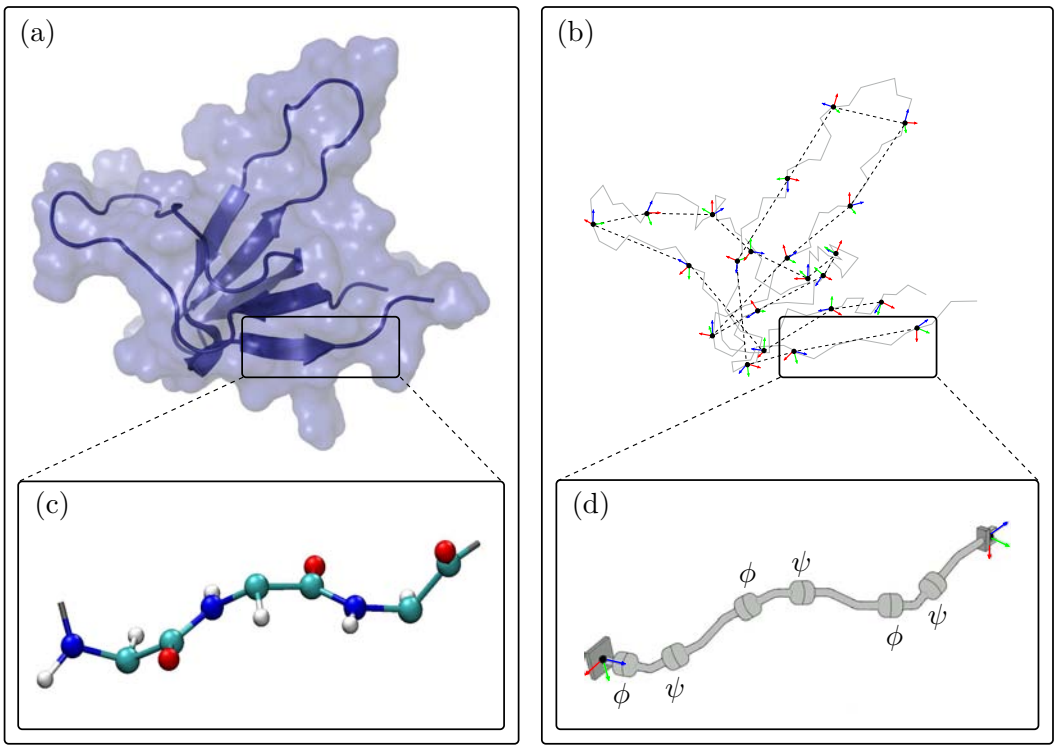


Figure 2.2: Illustration of the protein subdivision approach. Fragments of three amino-acid residues are treated as kinematic chains, similar to robotic manipulators. (Adapted with permission from ASME. Copyright 2012 ASME.)

corresponds to the first body of the tripeptide backbone (i.e. the first rigid atom group in the backbone of the first amino-acid residue).

Since tripeptides are linked through rigid peptide bonds, the location of the end effector of tripeptide i can be determined from the base-frame of tripeptide $i+1$ by a constant transformation. Given the location of the base-frame and the end-frame, the conformation of a tripeptide backbone can be computed using *inverse kinematics* (IK). The IK solver applied in this work is described in paragraph 2.3.2. Consequently, the conformation of the whole protein backbone can be determined from the pose of a single reference frame for each tripeptide, the one attached to the first body of each tripeptide backbone. In the following, we will refer to these reference frames as (oriented) *particles*. The last affirmation is true for all the protein backbone except two short fragments at the N-terminal and C-terminal ends of the chain. Since the choice of the first residue for the decomposition into tripeptides is arbitrary (and may change during the conformational exploration process) the polypeptide chain model involves two terminal fragments, containing up to three residues, which require a particular treatment. The conformation of these terminal fragment are directly defined by their internal bond torsions.

2.3 Devising move classes

The following section presents a unified approach for devising different move classes that can be employed in the context of a Monte Carlo framework. These classes all utilize the tripeptide-based representation described above. The principle consists in perturbing the pose (position and orientation) of a set of particles, and then to adapt the conformation of several tripeptides using IK in order to keep the integrity of the molecular chain while maintaining the local geometry of the bonds (i.e. constant bond lengths and bond angles). Several strategies can be considered for perturbing the pose of particles. The number of particles selected for perturbation and the correlation/uncorrelation between the motion direction of several particles will lead to different move classes, more or less local, and more or less collective. Although we only talk about unbiased move in this chapter, the presented approach is also suitable for devising biased moves. In such a case, the selection of the particles to be perturbed and the motion directions would be determined depending on the specific context. For instance, moves could be devised to deform proteins while simulating the interaction with other molecules, or for applications such as all-atom model fitting into lower-resolution electron density maps.

The move classes presented below, as well as other ones such as pivot moves applied to a single bond torsion, can be combined within a higher-level sampling protocol that selects a move class at each iteration. In addition, it is possible to relax the constraints on bond lengths, bond angles, and peptide bond torsions imposed by the tripeptide-based model. This can be done by performing separate MC moves on these parameters, or by slightly perturbing the geometry of some or all the bonds in a tripeptide before applying the IK solver.

It should be noted that side-chain conformations are not modified by move classes designed by the proposed approach. In order to take into account side-chain conformations, a specific treatment should be applied (usually by performing separate moves for side-chains). Side-chain conformations can be sampled by simple perturbations of the bond torsion angles χ_i or following more sophisticated approaches [Wu 1999b, Nilmeier 2008]. Moreover, different strategies can also be adopted to combine backbone and side-chain trial moves in a suitable manner [Nilmeier 2009].

2.3.1 Perturbing particles

Figure 2.3 illustrates three move classes that can be easily implemented from the proposed tripeptide-based model. They involve perturbations of one or several particles as explained in the following paragraphs.

2.3.1.1 One-particle moves

Local moves can be implemented by perturbing the pose of a single particle, as depicted in Figure 2.3 (a). Such a perturbation implies that the two tripeptides linked through this particle (i.e. with end-frame or base-frame defined from it) are subject to a backbone conformation change in order to close the chain. In other

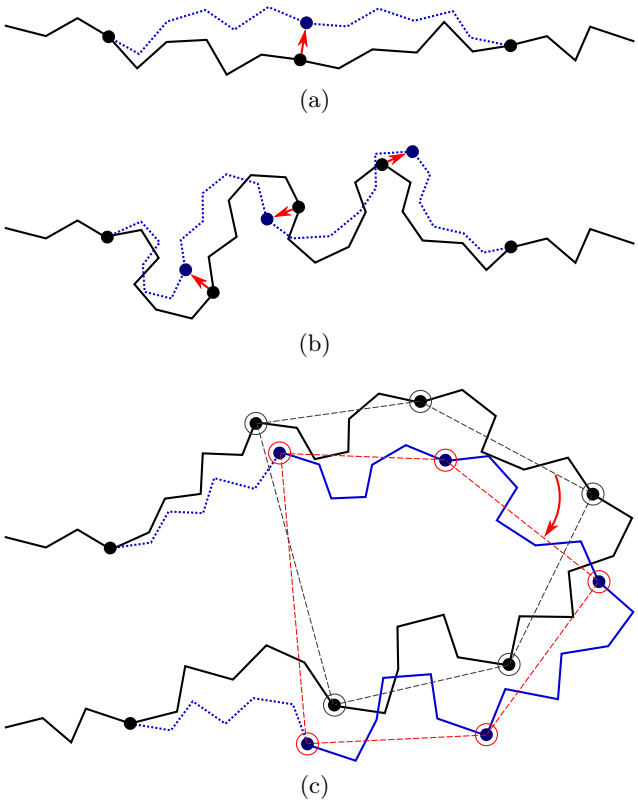


Figure 2.3: Illustration of three move classes devised from a tripeptide-based representation of proteins: (a) one-particle move, (b) flexible fragment move, (c) rigid body block move. The perturbations are represented with red arrows. The protein backbone trace before and after the move are drawn in black and blue, respectively. Tripeptides with conformational change are represented with a dashed line.

words, 12 consecutive bond torsions are modified, while the rest of the protein conformation remains unchanged. This type of moves will have a similar effect to other local, fixed-end move classes [Dodd 1993, Leontidis 1994, Wu 1999a] proposed from the seminal work of Gō and Scheraga [Gō 1970].

2.3.1.2 Flexible fragment moves

The previous move class can be extended to larger fragments by applying perturbations to a set of n consecutive particles. The move class is illustrated in Figure 2.3 (b) for the case of three particles. If the particles are perturbed independently (i.e. in different random directions), the backbone of $n+1$ tripeptides is affected by the move. This move class will have a similar effect to moves based on the cyclic coordinate descent method (CCD) [Canutescu 2003]. Such moves consist in breaking the chain at an arbitrary point, performing a random perturbation of several bond torsions at one of the sides, and applying CCD to close the chain again (but with a different, perturbed conformation). An interesting advantage of

the proposed move class with respect to CCD-based moves is that the deformation of sub-fragments can be easily modulated (e.g. larger perturbations for the middle particles in the fragment and smaller perturbations near the ends).

2.3.1.3 Rigid body block moves

A simple variant of the previous move class may produce a very different effect, as illustrated in Figure 2.3 (c). In this case, n consecutive particles are also perturbed, but the perturbations are correlated in such a way that the particles do not move with respect to each other. Indeed, the perturbation is applied to a virtual rigid body formed by the set of n particles. In principle, a random translation and rotation around an arbitrary axis could be tried. Nevertheless, it may be more interesting to apply moves that simulate hinge motions. Note that only two "hinge" tripeptides, the ones preceding and following the selected particle sequence, are subject to a change of conformation. Hinge-like moves, called closed rigid-body rotation under bond-angle restraints (CRRUBAR) moves [Betancourt 2005], have been shown to be particularly efficient for sampling conformations of proteins. Although the proposed method involves more complex algebraic operations than CRRUBAR, it presents the advantage that bond angles do not need to be distorted.

2.3.2 Solving inverse kinematics for a tripeptide

Once the location of the particles is set, obtaining the conformation of each tripeptide requires solving an IK problem for the kinematic chain corresponding to its backbone. As explained above, the model of a tripeptide backbone is similar to a six-revolute (6R) serial manipulator with general geometry. The method applied in this work for solving the IK problem for a general 6R serial kinematic chain has been adapted from the solver proposed by Renaud [Renaud 2000, Renaud 2006]. This solver is based on algebraic elimination theory, and develops an ad-hoc resultant formulation inspired by the work of Lee and Liang [Lee 1988b, Lee 1988a]. Starting from a system of equations representing the IK problem (the formulation involves the product of homogeneous transformation matrices), the elimination procedure leads to an 8-by-8 quadratic polynomial matrix in one variable. The problem can then be treated as a generalized eigenvalue problem, as was previously proposed by Manocha and Canny [Manocha 1994], for which efficient and robust solutions are available [Golub 2012]. Our implementation applies the Schur factorization from LAPACK [Anderson 1999]. Further details on the applied IK solver are provided in the technical report of Renaud [Renaud 2006].

This solver has been successfully applied in previous works on protein and polymer modeling [Cortés 2004, Cortés 2010a]. The advantage of this semi-analytical method with respect to numerical (optimization-based) methods, such as CCD, is that it provides the exact solution in a single iteration, not suffering from slow convergence issues. The solver is very computationally efficient, requiring about 0.2 milliseconds on a single processor. Note however that our approach is not depen-

dent on this solver, so that other IK methods [Manocha 1994, Coutsias 2004] could be applied.

In general, the IK problem for a 6R serial kinematic chain has a finite number of solutions (up to 16 in the most general case). All the solutions correspond to geometrically valid conformations of the tripeptide backbone with fixed ends defined by the pose of the particles. Depending on the type of application, several strategies can be adopted to select one of the solutions. The simplest strategy within a MC method consists in selecting one at random. However, if the moves are performed to find energy minima, all the conformations can be evaluated in order to keep the best one (in terms of the Boltzmann factor, for instance). If detailed balance needs to be satisfied for a correct sampling of equilibrium fluctuations in the canonical ensemble, some works recommend to take one of the solutions with a probability that depends on the Boltzmann factor and another term, called the Jacobian, which attempts to correct for the non-uniformity in the distribution of the torsion angles introduced by the closed-chain moves [Dodd 1993, Wu 1999b]. Otherwise, when the goal is to simulate continuous motions, the closest conformation to the one prior to the perturbation is selected in order to minimize the jumps in conformational space (if none of the solutions remains within a distance threshold that depends on the perturbation step-size, the local move is rejected).

2.4 Results

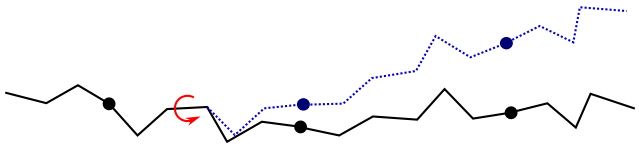
In order to test the approach in realistic conditions, some of the proposed move classes have been implemented and a comparative analysis of MC simulations using those move classes has been performed. For two of the implemented move classes, the analysis is pushed further by studying their autocorrelation function.

2.4.1 Implemented move classes and parameter settings

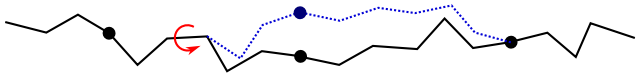
Four move classes were implemented. Three of them, producing fixed-end moves, are based on the proposed approach. The other one is simply based on the variation of a single bond torsion at each iteration. More precisely:

- The simplest class of trial moves, largely applied to sample the conformation of chain-like molecules, consists in perturbing a randomly selected bond torsion and then propagating the motion toward the end of the chain. Such moves, usually called pivot moves, are named here *OneTorsion* moves. They are illustrated in Figure 2.4 (a).

- The second move class is named *ConRot*, since it is inspired from the *concerted rotations* proposed by Dodd *et al.* [Dodd 1993]. It has been implemented using the tripeptide-based model as follows: an amino-acid residue is randomly selected and one of its bond torsions (ϕ or ψ) is randomly perturbed; the backbone conformation of the next three residues (the next tripeptide) is modified



(a) *OneTorsion*



(b) *ConRot*

Figure 2.4: Illustration of two frequently used move classes within MC methods applied to chain-like molecules: (a) pivot move, (b) concerted rotation.

by inverse kinematics in order to maintain fixed ends. The move class is illustrated in Figure 2.4 (b).

- The third move class, called *OneParticle* moves, corresponds to the simplest move class involving particle perturbations, as described in previous section, and illustrated in Figure 2.3 (a).
- The last move class, called *Hinge* moves, corresponds to the rigid-body block moves described in previous section, and illustrated in Figure 2.3 (c). The number of consecutive particles affected by the move is randomly sampled at each iteration between 3 and 10 (i.e. moves involve between 9 and 30 residues).

These four move classes have been applied within a basic MC method, using the Metropolis criterion (1.1) to accept or to reject trial moves. At each iteration, the algorithm randomly chooses between performing either a backbone move or a side-chain move. A side-chain move consists of randomly selecting a side-chain and perturbing all of its dihedral angles χ_i. We have tested the four backbone move classes individually as well as a simple combination of all of them. The *Mixed* move class selects one of the four backbone move classes with equal probability at each iteration.

At each iteration of the MC method, the conformational parameters involved in the applied move class (bond torsions or oriented particle poses) are perturbed by adding a random value to their current value, sampled in the interval $[-\delta, \delta]$. Thus, the parameter δ defines the maximum perturbation step-size. For a meaningful comparative analysis, the values used for the perturbation of bond torsions (δ_b), particle translations (δ_{pt}) and particle rotations (δ_{pr}) in each move class have to be chosen in such a way that they will produce average atom displacements of similar length. In general, this also implies that MC acceptance rates will be similar for all move classes. The values used in this work for the three systems introduced below are presented in Table 2.1. They were first chosen to produce similar average displacements compared to the *ConRot* move class. Then, they were adapted to obtain acceptance rates around 50%.

	OneTorsion	*ConRot*	*OneParticle*		*Hinge*
	δ_b	δ_b	δ_{pt}	δ_{pr}	δ_{pr}
SH3 domain	0.01 rad.	0.025 rad.	0.05 Å	0.003 rad.	0.01 rad.
Sic1 protein	0.02 rad.	0.025 rad.	0.05 Å	0.003 rad.	0.02 rad.
14-alanine	—	0.025 rad.	0.02 Å	0.006 rad.	—

Table 2.1: Perturbation step-sizes

Energy evaluation were performed using an in-house implementation of the AMBER parm96 force-field [Kollman 1997] with an implicit representation of the solvent using the Generalized Born (GB) approximation. A geometric filter is applied before energy evaluation with the aim of improving computational efficiency[1]. After applying each trial move, the model is checked for atom overlaps: a trial move is rejected if the distance between two non-bonded atoms is less than 70% of the van der Waals equilibrium distance [Bondi 1964]. If a trial move passes the geometric filter, the Metropolis criterion is applied. In addition, for *ConRot*, *OneParticle* and *Hinge* move classes, a trial move is rejected if the IK solver fails to find a solution. All the tests have been performed at a temperature of 300 K.

2.4.2 Test systems

Three different system were chosen to evaluate the performance of the move classes. The first one is the *SH3 domain* of *obscurin*, represented in Figure 2.2(a). This is a small globular protein composed of 68 amino acid residues. It presents a relatively rigid beta-barrel-like core and two flexible loops. Its crystal structure is available in the Protein Data Bank under the PDB ID: 1V1C. The second test system is an intrinsically disordered protein called the *Sic1 protein*, which contains 77 residues. A model of this protein is shown in Figure 2.5. The model was generated using the Flexible-Meccano method [Bernado 2005] for sampling a statistically probable backbone conformation, and SCWRL4 [Krivov 2009] for placing the side chains. The third test system is a much smaller peptide that consist of 14 consecutive alanine. This system is only used to study the autocorrelation of MC simulations using the *ConRot* and the *OneParticle* move classes (see paragraph 2.4.5.2). All the systems were locally energy-minimized before running the tests. The results presented in this section are not aimed to provide new insights into these biological systems, but to serve as a proof of concept and to show the interest of the proposed approach.

[1] Our implementation of the energy force-field is not optimal in terms of computing time, being the solvent contribution the most time-consuming part.

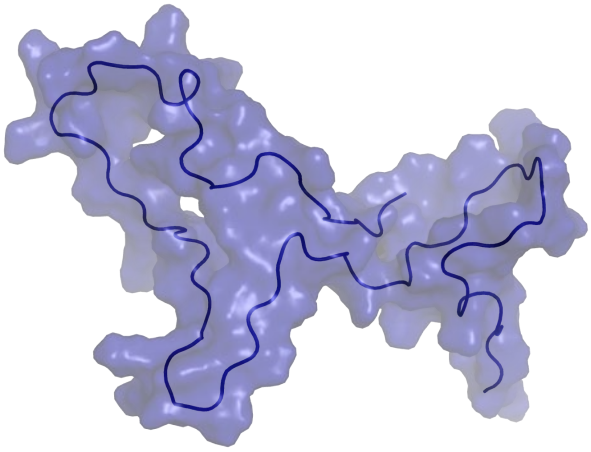


Figure 2.5: Representation of the Sic1 protein.

2.4.3 Computational performance

In order to compare the different move classes, long MC simulations were run on the SH3 domain and on the Sic1 protein. Starting from the minimized conformations, the MC method was iterated until 2×10^6 samples were accepted for each molecule, using the four move classes individually or combined together. Each tests was repeated 3 times. All tests using the same settings provided similar results.

Table 2.2 contains results on the computational performance of the methods, averaged over the 3 runs. It shows the MC acceptance rate of backbone moves, the total number of iterations (considering backbone and side-chain moves) and the overall CPU time[2] for each system and move class. These results show that, for a similar MC acceptance rate, generating a given number of samples (2×10^6 in this case) requires more computational resources using the *OneTorsion* move class compared to the other move classes, which apply fixed-end motions. The reason is that, although the *OneTorsion* move class does not require solving inverse kinematics, a significant amount of computing time is needed for propagating atom motions along the chain by forward kinematics and to recompute interaction energies between atom pairs. In other words, the computing time needed by the *ConRot*, *OneParticle* and *Hinge* move classes to solve inverse kinematics is largely compensated by the local nature of these move. Since the positions of only a small number of atoms need to be updated after each move, it reduces the cost of forward kinematic and energy recalculation.

Nevertheless, computing time is probably not the most important performance

[2]Tests were run on a single Intel(R) Xeon(R) E5-1650 processor at 3.2 GHz.

	Move Class	Acc. Rate	# Iterations	T_{CPU}
SH3 domain	*OneTorsion*	0.68	3.28×10^6	63 h.
	ConRot	0.56	3.55×10^6	51 h.
	OneParticle	0.42	4.06×10^6	56 h.
	Hinge	0.59	3.46×10^6	57 h.
	Mixed	0.56	3.58×10^6	57 h.
Sic1 protein	*OneTorsion*	0.56	3.54×10^6	89 h.
	ConRot	0.65	3.25×10^6	63 h.
	OneParticle	0.52	3.66×10^6	69 h.
	Hinge	0.53	3.60×10^6	75 h.
	Mixed	0.57	3.49×10^6	74 h.

Table 2.2: Computational performace

indicator of the different move classes. Indeed, the numbers in Table 2.2 have to be analyzed together with data related to the quality of the sampling strategy. This is the subject of the following sections.

2.4.4 Distribution of sampled states

In order to compare the different move classes more in-depth, we need to understand how well they explore the conformational space. Figure 2.6 shows plots aimed at comparing the performance of the different move classes in terms of conformational space coverage for the SH3 domain and the Sic1 protein. They represent the projection of the sampled states on two dimensions: the distance with respect to the initial structure and the potential energy. The distance is measured as the root mean square deviation (RMSD) of the bond torsion angles. For clarity reasons, only one for every one hundred sampled conformations has been plotted (i.e. 20,000 samples from each run). Each plot includes the sum of the results of the three runs. Table 2.3 provides quantitative values, for each move class, of the RMSD distance of the furthest reached configuration (second column), and of the energy variation of the lowest energy configuration (third column).

Several conclusions can be extracted from the analysis of these plots:

- The *Mixed* move class shows the best performance. The combination of move-classes provides samples that have lower energies, and conformational coverage (distances from a reference conformation) is comparable or better than for the individual move classes.

- Move classes involving a larger number of atoms, i.e. *OneTorsion* and *Hinge*, provide better results than more local move classes on the disordered Sic1 protein, which presents higher conformational variability than the globular SH3 domain. For the Sic1 protein, these move classes reach lower-energy regions compared to *ConRot* and *OneParticle*.

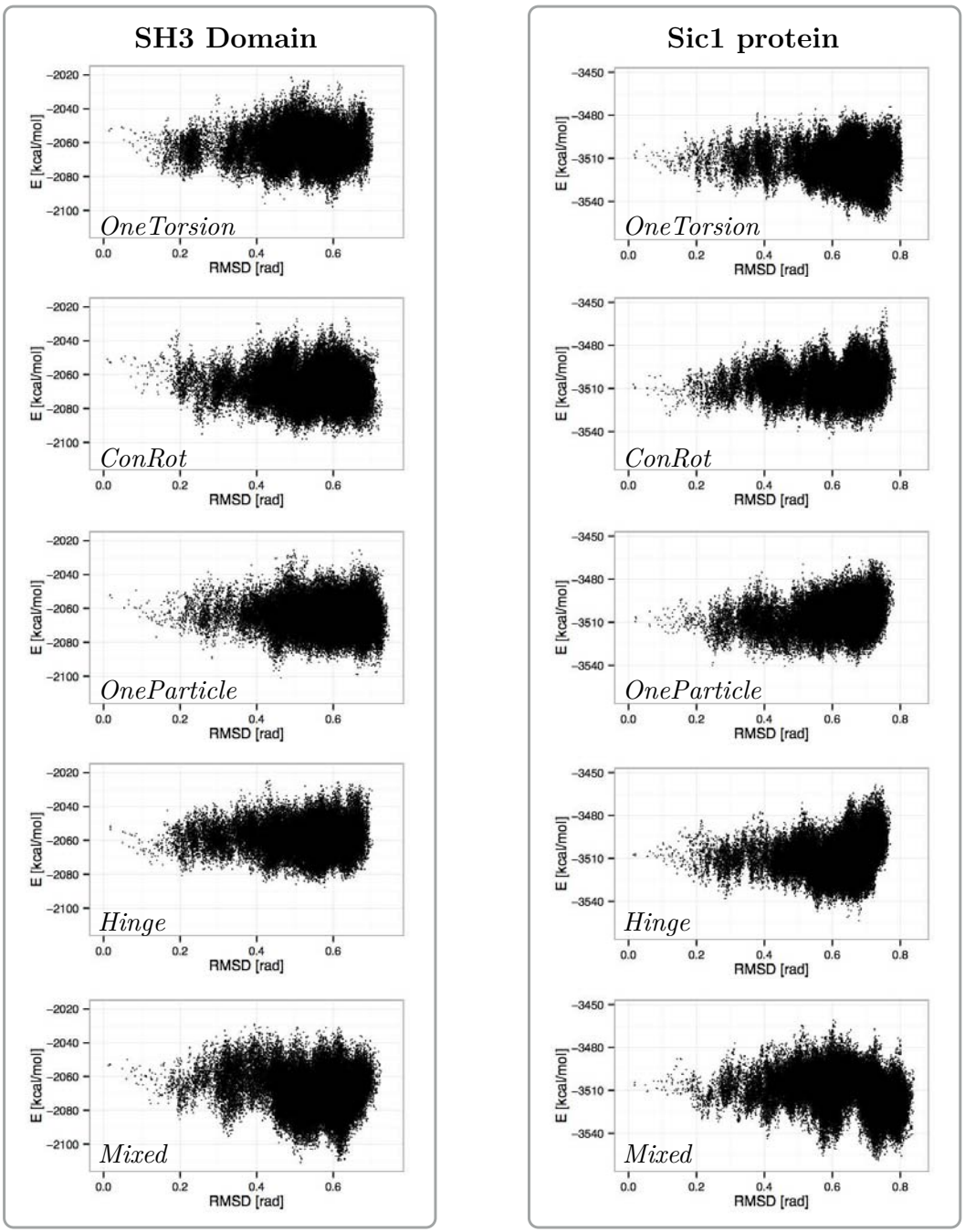


Figure 2.6: Projection of sampled states on distance vs. energy plots for the SH3 domain and the Sic1 protein.

	Move Class	Max. RMSD [rad]	Min. ΔE [kcal/mol]
SH3 domain	*OneTorsion*	0.71	-50
	ConRot	0.73	-49
	OneParticle	0.75	-51
	Hinge	0.70	-39
	Mixed	0.72	-62
Sic1 protein	*OneTorsion*	0.81	-55
	ConRot	0.79	-45
	OneParticle	0.78	-44
	Hinge	0.77	-54
	Mixed	0.84	-61

Table 2.3: Computational performace

- As it could be expected, the performances of *ConRot* and *OneParticle* moves are very similar for both test systems.
- The *Hinge* move class shows slightly poorer exploration capabilities compared to the other move classes. The maximum distances of the samples to the reference conformation are smaller when using this move class alone. The difference is more significant for the SH3 domain.

2.4.5 Exploration efficiency analysis

The distance vs. energy plots permit to compare the different move classes in terms of coverage, but they do not allow us to quantitatively evaluate their exploration efficiency. Furthermore, these plots illustrate the performance at the end of a long MC run, but do not show the short term exploration capabilities of the different move classes, which are of interest in the context of CPD, where such long simulations are not possible in practice. This section presents additional tests allowing such an analysis: the time dependent RMSD function and the autocorrelation time.

2.4.5.1 Time dependent RMSD function

The time dependent RMSD function, rmsd(τ), is aimed to show the rapidity of the exploration process. It represents the average distance between conformations separated by τ MC steps:

$$\text{rmsd}(\tau) = \langle \text{RMSD}(C_k, C_{k+\tau}) \rangle_k \tag{2.1}$$

where C_k is the conformation of the system after the k$^{\text{th}}$ trial in the MC simulation, and $\langle X(k) \rangle_k$ denotes the average of the observable X over all k.

Figure 2.7 plots the time dependent RMSD function of each move class and for the two molecules. It provides interesting additional information with respect

to the analysis in Section 2.4.4 and allows to highlight more significant differences between the move classes:

- First, it confirms the previous observation that the *Mixed* move class performs better than the other individual move classes.
- Yet, if the *OneParticle* and *ConRot* move classes still perform very similarly, it appears that the *OneParticle* move class explores a bit faster than *ConRot*. In the case of the SH3 domain, *OneParticle* can even compete with the *Mixed* move class on small time scales.
- The *Hinge* move class performs only half as well as the *Mixed* move class for both proteins while the *OneTorsion* move class performs almost four times worse. This last result, together with results in the previous section, shows that, although *OneTorsion* moves may provide good performance in terms of coverage (particularly for the disordered Sic1 protein), convergence can be slow.

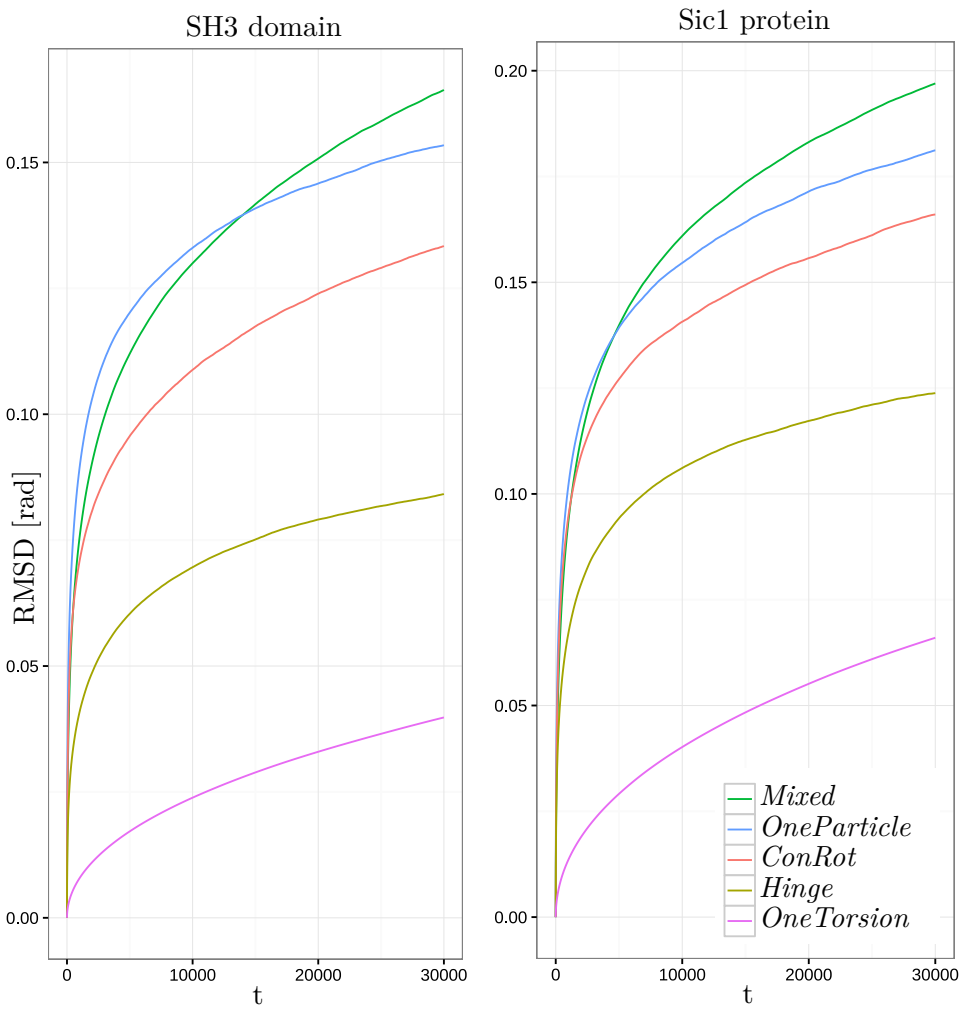
2.4.5.2 Autocorrelation time

Autocorrelation is a statistical tool that computes the correlation of a time series with a lagged version of itself. In the context of a MC simulation, it is used to characterize how new accepted conformations gradually become independent from the past conformations. Formally, the autocorrelation of an observable O with a lag τ is defined by:

$$\mathrm{acf}(\tau) = \frac{\mathrm{E}[(O_k - \langle O \rangle)(O_{k+\tau} - \langle O \rangle)]}{\mathrm{Var}(O)} \tag{2.2}$$

where $\mathrm{E}[X]$ denotes the expected value of X, $\mathrm{Var}(O)$ denotes the variance of O, and O_k and $O_{k+\tau}$ represent two series of observations of O during the MC simulation with a lag of τ steps (the k^{th} observation of $O_{k+\tau}$ is equal to the $(k+\tau)^{\text{th}}$ observation of O_k). This formula only holds true when the simulation is stationary. In the context of MC simulations, this requirement implies that the simulation is long enough for the autocorrelation value to converge. Since this would require very long computing time for large systems such as the SH3 domain or the Sic1 protein, a smaller system was used for this test. Inspired by related work [Ulmschneider 2003, Bottaro 2012], a 14-alanine was used with a structural constraint: starting from a low energy stable state, corresponding to an α-helical conformation, the two end residues were blocked in position and orientation so the system could only fluctuate around that state. This constraint, associated with the small size of the system, restricts the possible move classes to *ConRot*, *OneParticle*, and a *Mixed* move class involving both of them.

Starting from the α-helical conformations, represented in figure 2.8, the MC method was iterated until 10^8 samples were accepted for the two move classes individually or combined together (no side-chain move were performed). Figure 2.9



(a) (b)

Figure 2.7: Plots of the function rmsd(t) of each move class during the MC simulations for (a) SH3 domain and (b) the Sic1 protein.

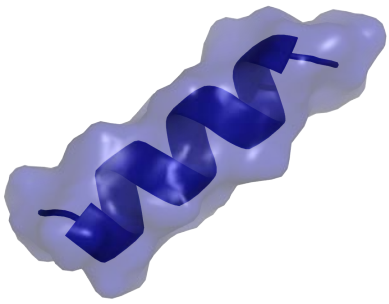


Figure 2.8: Representation of the 14-alanine peptide.

shows the average autocorrelation function over the 20 central dihedral angles for each move class. We can observe that the *Mixed* move class yields the fastest independence time, whereas the autocorrelation of *ConRot* seems to stagnate when approaching zero.

For a more precise evaluation, the characteristic autocorrelation time, also called the integral autocorrelation time, was computed. It is defined by:

$$\tau_{int} = \frac{1}{2} + \sum_{\tau=1}^{\infty} \mathrm{acf}(\tau) \tag{2.3}$$

In practice, to reduce computational cost and noisy values, the summation is usually truncated. We have summed for all $\tau \leq 5\tau_{int}$[3]:

$$\tau_{int} = \frac{1}{2} + \sum_{\tau=1}^{5\tau_{int}} \mathrm{acf}(\tau) \tag{2.4}$$

The autocorrelation times of the 20 central dihedral angles are plotted in Figure 2.10 and the average for each move class is summarized in table 2.4. These results allow us to quantify the improvement of autocorrelation time between the different move classes. The *OneParticle* move class has an autocorrelation time about 1.8 times lower on average than the *ConRot* move class, while the *Mixed* move class consistently outperforms the *OneParticle* move class by a factor of 2. These results confirm the fact that the *Mixed* move class is much more efficient than the other individual move classes. It also shows how important the choice of a move class can be for the efficiency of a simulation and how the mixture of several types of move class highly improve the quality of the MC simulation.

[3]In practice, the summation is iteratively grown until τ_{int} reaches a value such as the maximum summation index is higher than $5\tau_{int}$

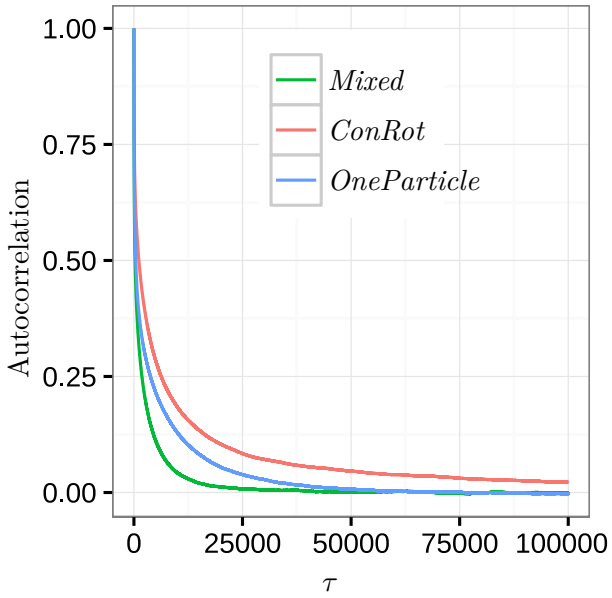


Figure 2.9: Plots of the average autocorrelation function over the 20 central dihedral angles of 14-alanine for the *ConRot*, *OneTorsion*, and *Mixed* move classes.

Move Class	Average	Min	Median	Max
ConRot	6016	512	5315	14070
OneParticle	3426	343	1215	11281
Mixed	1518	157	985	3706

Table 2.4: Autocorrelation times τ_{int}(in MC steps) of the *ConRot*, *OneTorsion*, and *Mixed* move classes

2.5 Conclusion

In this chapter, a unified approach to devise efficient MC move classes for chain-like molecules was presented. Based on a subdivision of the protein into tripeptides and on the application of 6R inverse kinematics, many of the move class that have been proposed in the last decades to enhance protein backbone sampling can be easily implemented. First results for proteins from different structural classes (globular and disordered) have been presented as a proof of concept. The overall conclusion of these results is that mixing move classes provides better exploration capabilities than using a single move class, as also suggested by related work on MC methods. Combining different move classes is a straightforward task following the proposed approach.

Only results of simple, unbiased sampling were presented. However, the approach could also be implemented within more advanced methods for conformational sampling [Okamoto 2004] or global optimization [Carr 2005]. Furthermore,

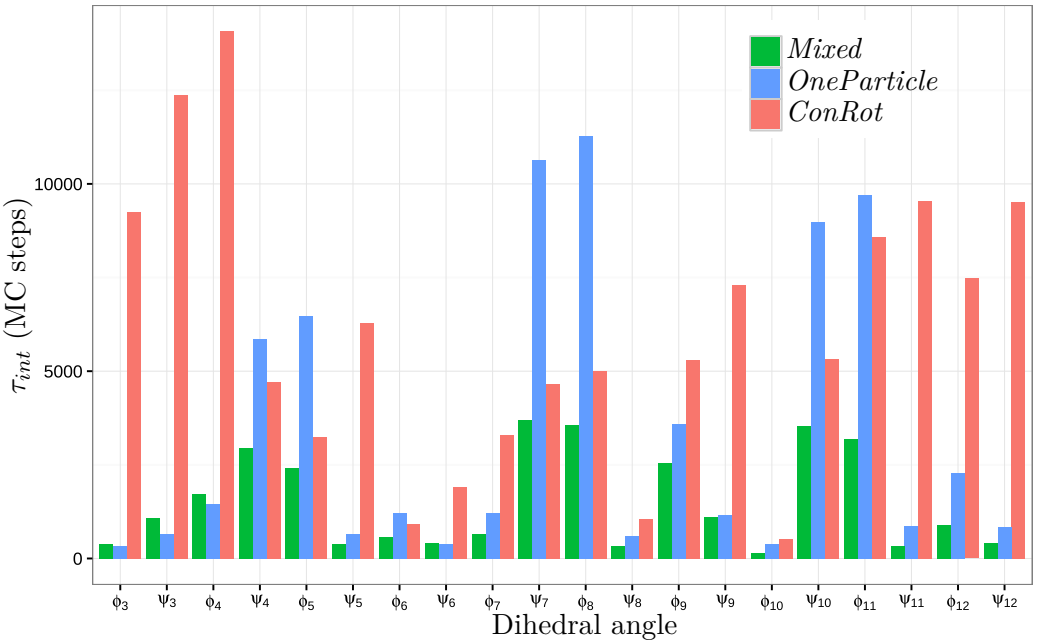


Figure 2.10: Autocorrelation times of the 20 central dihedral angles of 14-alanine for the *ConRot*, *OneParticle*, and *Mixed* move classes.

the approach would allow to easily implement biased moves that deform regions of the protein with respect to interactions with other molecules. This could have application in conformational space exploration methods involving protein-ligand interactions, or protein-protein docking. Finally, this approach is particularly interesting in the context of CPD. It has been shown that introducing local backbone perturbation may significantly improve the results provided by protein design methods [Mandell 2009].

Chapter 3

Exploration of the conformational energy landscape

Contents

3.1 The exploration-exploitation dilemma 48
3.2 Algorithms 49
3.2.1 T-RRT 49
3.2.2 Transition-based EST 49
3.3 Empirical comparative analysis 54
3.3.1 Molecular systems 54
3.3.2 Experiment setup 54
3.3.3 Results 56
3.4 Conclusion 59

Efficiently exploring the conformational energy landscape of a molecular system is not solely a matter of move classes used to locally deform the structure. The global strategy used to sample the conformational space is of great importance. As we explained in Section 1.4, if Molecular Dynamics (MD) and Monte Carlo (MC) simulations have some very useful properties allowing a statistical analysis of a protein conformational landscape, they are not the most efficient algorithms to quickly discover possible transition path between different conformations. Algorithms adapted from the robotics community have had some great success in that regard. In this chapter, different algorithms will be compared using small but highly flexible peptide systems:

- a simple MC method,
- the Transition-based Rapidly-exploring Random Tree (T-RRT) algorithm,
- an adaptation of the Expansive-Spaces Trees (EST) algorithm using a similar approach as the KPIECE algorithm [Sucan 2012] for node selection,
- and a second adaptation of the EST algorithm using a scoring function based on the new node acceptance rate to balance the exploration.

Both EST adaptations use a transition test similar to the one used by T-RRT to accept or reject a new configuration. These transition-based EST algorithms are new approaches developed and analyzed in the framework of this thesis. As will be shown in this chapter, they need to be improved before they could be considered as alternatives to T-RRT.

3.1 The exploration-exploitation dilemma

In high dimensional space, such as the conformational space of a protein, obtaining a full coverage of the space is not possible in practice. In order to gain knowledge about the conformational energy landscape, one must focus the exploration on specific regions of interest. As mentioned in Section 1.4, many different strategies have been developed over the years. Some methods, like the one presented in [Molloy 2014], use databases of backbone conformations to focus the computational resources on parts of the space known to be structurally stable. However, considering the difficulty to observe and measure the intermediate conformations along transitions from one state to another, some parts of the landscape, particularly those corresponding to transition regions, cannot be explored with this kind of algorithms. Some other approaches, like the one presented in [Brunette 2008], build an approximate and partial model of the energy landscape aiming to focus the search on regions of interest. Nevertheless, this method cannot ensure that transition paths between conformations will be identified as regions of interest. Indeed, conformations inside transition paths may have significantly higher energy compared to conformations in low energy basins and thus may be rejected in the search.

Another approach to efficiently explore the conformational space without relying on prior knowledge is to deal with the exploration-exploitation dilemma. The principle is to balance the computational resources allocated to the exploration of unknown areas of the conformational space with the computational resources allocated to the exploitation of the regions of the space that were identified as *of intererst* by the previous explorations. This principle is exploited by the FAST method [Zimmerman 2015] for *de novo* structure prediction. The algorithms presented in this chapter are also based on balancing exploration and exploitation.

3.2 Algorithms

3.2.1 T-RRT

The transition-based RRT (T-RRT) algorithm [Jaillet 2008] is an extension of the RRT algorithm (see Section 1.4.3.2) adapted for cost-space path planning. It can be applied in computational structural biology by using the potential energy function as the cost-function on the configuration space [Jaillet 2011]. The T-RRT algorithm, as used in computational structural biology, is described in Algorithm 3.1. T-RRT combines the exploration power of RRT with a stochastic transition test enabling

Algorithm 3.1: T-RRT Algorithm

input : the configuration space $\mathcal{C}$; the start state q_{init};
the initial temperature T_{init}
output: the tree $\mathcal{T}$

```
1 T ← InitTree(q_init)
2 T ← T_init
3 while not StoppingCriterion (T, MaxTime) do
4     q_rand ← Sample(C)
5     q_near ← NearestNeighbor(T, q_rand)
6     q_new ← Extend(q_rand, q_near)
7     if TransitionTest(T, q_near, q_new, T) then
8         AddNode(T, q_near, q_new)
```

it to favor the exploration of low-energy regions of the configuration space. This transition test, presented in Algorithm 3.2 is based on the Metropolis Criterion (1.1) typically used in MC methods (see Section 1.4.2). It is used to accept or reject a new candidate configuration q_{new} based on the energy variation involved in the transition from q_{near} to q_{new}. First a geometric filter is applied on the configuration q_{new}. The model is checked for atom overlaps, and if a collision is found, *i.e.* if the distance between two non-bonded atoms is less than 70% of the van der Waals equilibrium distance [Bondi 1964], the configuration q_{new} is rejected. Otherwise, the Metropolis Criterion is applied. As described in Section 1.4.2, two cases are possible:

1. A transition corresponding to a downhill move in the conformational energy landscape ($E_{\text{new}} \leq E_{\text{near}}$) is always accepted.

2. A transition corresponding to a uphill move in the conformational energy landscape is accepted with the probability $\exp(-(E_{\text{new}} - E_{\text{near}})/(K \cdot T))$ that decreases exponentionally with the energy variation.

The level of selectivity of the transition test is controlled by the adaptive temperature parameter T. Low temperatures limit the expansion to gentle slopes of the energy landscape, while high temperatures enable to climb steep slopes. The temperature is dynamically tuned during the exploration process. After each accepted uphill transition, T is decreased to avoid exploring high-energy regions. After each rejected uphill transition, T is increased to facilitate exploration and avoid being trapped in a local minimum. This adaptative tuning of the temperature allows T-RRT to automatically balance its bias toward low-energy regions with the Voronoï bias of RRT toward unexplored regions. The parameter T_{rate} controls the rate of the temperature increases. A high value leads to a greedy exploration, while a low value is more conservative.

Algorithm 3.2: TransitionTest($\mathcal{T}$, q_{near}, q_{new}, T)

input : the input tree $\mathcal{T}$; parent node q_{near}; new node q_{new};
the temperature T; the energy function E;
the temperature adjustment rate T_{rate}; the Boltzmann constant K

output : A boolean indicating if the transition test passed or not

```
1  if CollisionTest(q_new,d) == False then
2      E_near = E(q_near)
3      E_new = E(q_new)
4      ΔE = E_new - E_near
5      if ΔE < 0 then
6          return (true)
7      else
8          if exp(−ΔE / (K · T)) > UniformRand() then
9              T ← T / 2^((ΔE) / energyRange(T))
10             return (true)
11         else
12             T ← T · 2^T_rate
13             return (false)
14 else
15     return (false)
```

3.2.2 Transition-based EST

The EST algorithm, explained in section 1.4.3.3, is much more flexible than the RRT algorithm. In fact, the choice of the node to perturb at each iteration of the algorithm corresponds to a heuristic that has to be defined. While the RRT-based algorithms, such as T-RRT, spend a huge amount of time trying to explore all the unexplored areas of the space, the EST heuristic allows to bias the exploration in favour of the regions of interest. For instance, promising directions could be could be favored and explored intensively to the detriment of other regions of the space. In the case of the exploration of a high-dimensional energy landscape, our goal is to find a heuristic that will push the exploration towards transition paths between basins while reducing the computation time spent trying to cross high-energy barriers. In the following, we will propose two different heuristics to achieve this goal. The first one is inspired from the multi-armed bandit robot problem [Auer 2002]. The second one reproduces the heuristic used in the KPIECE algorithm [Sucan 2012]. The transition-based EST, described in Algorithm 3.3, is an EST where the transition test described in Algorithm 3.2 is used with the same temperature parameter as the one of T-RRT and the same conditions to update the temperature. The *NodeSelection* function corresponds to the heuristic that determines which node is perturbed at each iteration of the algorithm. The *Perturb* function is another parameter defining how the selected conformation is perturbed

Algorithm 3.3: Transition-based EST Algorithm

input : the configuration space $\mathcal{C}$; the start state q_{init};
the initial temperature T_{init}; a perturbation function Perturb
output: the tree $\mathcal{T}$

$\mathcal{T} \leftarrow$ `InitTree`(q_{init})
$T \leftarrow T_{\text{init}}$
while *not StoppingCriterion* $(\mathcal{T}$, *MaxTime)* **do**
  $q \leftarrow$ `NodeSelection`$(\mathcal{T}, \mathcal{C})$
  $q_{\text{new}} \leftarrow$ `Perturb`(q)
  if *TransitionTest*$(\mathcal{T}, q, q_{\text{new}}, T)$ **then**
    `AddNode`$(\mathcal{T}, q, q_{\text{new}})$

(for instance, the different move classes defined in Chapter 2 can be applied here).

3.2.2.1 Node selection heuristic: success score

The first implemented heuristic is an approach based on the multi-armed bandit problem [Auer 2002]. This idea has already been applied, in a different manner, to the problem of secondary structure prediction (predicting the local regular (secondary) structure of a protein from its sequence) [Zimmerman 2015]. The idea is to balance the exploration by keeping track, for each node, of the expected progress in the coverage of the conformational space. To do that, each node q in the EST tree $\mathcal{T}$ is given a reward function indicating how well the exploration was improved by extending this node. The reward function is balanced with another term quantifying the uncertainty on the actual progress that will be achieved by selecting each node. These two terms are combined into a score function of the form:

$$score(q) = \begin{cases} reward(q) + \beta \cdot \sqrt{\log\left(\frac{I}{N_{trial}(q)}\right)} & \text{if } N_{trial}(q) > 0 \\ \langle reward(x) \rangle_{x \in \mathcal{T}} + \beta \cdot \sqrt{\log(I)} & \text{otherwise} \end{cases} \tag{3.1}$$

where I is the number of the current iteration, $N_{trial}(q)$ is the number of times q was selected for extension, β is a parameter that controls the balance between exploration of uncertain nodes and exploitation of nodes with high reward, and $reward(q)$ is the current expected value of the reward for node q. When β is high, the score favors the nodes that have been chosen the least. When β is low, the score favors the nodes that have the best expected reward.

For the purpose of this work, a very simple reward function was implemented:

$$reward(q) = \frac{N_{success}(q)}{N_{trial}(q)} \tag{3.2}$$

where $N_{success}(q)$ is the number of times a new node was created from the selection of node q by the algorithm. As it will been shown later, this reward function is too

simple to yield good result in the exploration, though it has the advantage to be very fast to compute.

3.2.2.2 Node selection heuristic: KPIECE like

KPIECE [Sucan 2012] is a motion planning algorithm specifically designed for systems with complex dynamics. It uses a projection combined with a grid-based discretization of the space to estimate its coverage, detect its boundaries, and focus the search in the less explored areas. The node selection heuristic described in this section is inspired from this method.

The idea behind KPIECE is to use a projection to have a simplified view of the coverage of the exploration in the search space (the conformational space in our case). This projection is a parameter of the algorithm and must be chosen carefully to capture as much as possible the topology of the space. Using this projection, each node in the EST tree corresponds to a point in a k-dimensional euclidean space. In this space, a grid is built and each node in the EST tree is associated with the cell that contains the projected point. From this grid, we can estimate how much each cell has been explored comparatively to the others by simply counting the number of nodes they contain. Furthermore, we can easily determine if the cell lies in the interior of the explored region or at its border: if all the neighboring cells of a cell contain nodes, then this cell is considered to be in the interior of the explored region, while, if a cell has neighboring cells without any associated nodes, then this cell is at the border of the search space. By chosing more often cells lying at the exterior of the search space, and by chosing more often cells with a low coverage, the algorithm will favor a fast exploration of the search space, and thus, the discovery of potential transition paths.

More formally, a projection $Proj$ is defined from the conformational space to $\mathbb{R}^k$ where k is the dimension of the projection. At each node q of the tree corresponds a point $(p_1, \ldots, p_k)$ in the projection space.

A $Coord$ function then converts this point into coordinates in $\mathbb{Z}^k$, where $\mathbb{Z}$ denotes the set of integers:

$$Coord((p_1, \ldots, p_k)) = \left(\left\lfloor \frac{p_1 - o_1}{d_1} \right\rfloor, \ldots, \left\lfloor \frac{p_k - o_k}{d_k} \right\rfloor \right) \tag{3.3}$$

where $\lfloor \cdot \rfloor$ round the value to the nearest smaller integer, $(o_1, \ldots, o_k)$ is an arbitrary point of $\mathbb{R}^k$ chosen as the origin, and $d_1, \ldots, d_k$ are positive real numbers defining the size of the grid cells in each dimension. The result of the $Coord$ function is interpreted as the coordinate of a cell in a k-dimensional grid of unit size hypercubes. A node q is said to be inside a cell z of coordinates $(z_1, \ldots, z_k)$ if $Coord(Proj(q)) = (z_1, \ldots, z_k)$.

It is now possible to define the coverage of a cell. As the problem we are addressing is different from the motion planning problem addressed by KPIECE, the notion of coverage differs from the one defined in the original paper. Here, the coverage of a cell is defined as the number of contained nodes plus one.

For every cell $z = (z_1, \ldots, z_k)$, we define the neighbors of z by:

$$Neighbors(z) = \{(z_1, \ldots, z_{i-1}, y, z_{i+1}, \ldots, z_k), \\ \text{for } i \in [\![1, k]\!] \text{ and } y = z_i + 1 \text{ or } y = z_i - 1\} \quad (3.4)$$

A cell is considered exterior if at least one of its neighboring cells does not contain any node.

From that, an importance score can be given to each cell containing at least one node. It is defined by:

$$Importance(z) = \frac{\log(I(z)) \cdot score(z)}{S(s) \cdot (1 + |Neighbors(z)|) \cdot Coverage(z)} \quad (3.5)$$

where $I(z)$ is the iteration at which a node was first added into cell z, $S(z)$ is the number of times cell z was selected for expansion by the algorithm, $|Neighbors(z)|$ is the number of neighboring cells which are not empty, and $coverage(z)$ is a value that tries to reflect how much expansion of the coverage of the space was performed by selecting cell z previously. The value of $score(z)$ is initialized to 1 and is updated each time a cell is selected for expansion following these rules:

$$score = \begin{cases} score \cdot \min(\alpha + \beta, 1) & \text{if the expansion was successful} \\ score \cdot \min(\alpha, 1) & \text{otherwise} \end{cases} \quad (3.6)$$

where α and β are parameters of the algorithm. α and β are positive number and α must be less than 1. The α parameter determines how fast the cell scores decrease at each iteration, favoring newly created cells, while the β parameter adapts the decrease rate for cells yielding good acceptance rate.

Using this importance score, the choice of the node to perturb in the EST algorithm proceeds in three steps. First, a random choice is made between expanding from an interior cell or an exterior cell. The decision is biased toward exterior cells with a probability P_{ext} (typically 75%). Then, the cell with the highest importance is selected within the set of interior/exterior cells. Finally, a random node is chosen inside that cell: if the cell contains m nodes, the nodes are ordered by their order of creation (the most recent one being the first), and a node is selected using a half-normal distribution (a normal distribution folded about the y-axis at 0) with standard deviation $m/3$.

3.3 Empirical comparative analysis

In this section, the T-RRT algorithm, and the two versions of the EST algorithm described in section 3.2 are applied to the exploration of the conformational energy landscape of two peptides. Their ability to discover a transition path between two stable states is analyzed and compared to a simple MC method. In the following, the version of the EST algorithm with the success score heuristic described in

Section 3.2.2.1 will be designated as EST_{ss}, and the version of the EST algorithm with the KPIECE-like heuristic described in Section 3.2.2.2 will be designated as EST_{kpiece}.

3.3.1 Molecular systems

Two small systems have been used to compare the algorithm. Note that the results presented in this section are not aimed to provide new insights into these biological systems, but to serve as a benchmark to compare the proposed algorithms. The first one is *met-enkephalin*: a pentapeptide of sequence YGGFM (PDB ID: 1PLX). Its highly variable structure makes it a good candidate to compare exploration algorithms. Furthermore, it has been extensively studied, and its conformational energy landscape has been globally characterized [Devaurs 2013b]. The second system is *chignolin*: an artificially designed mini-protein consisting of 10 amino acid residues. Its sequence is GYDPETGTWG, and a crystal structure can be found in the Protein Data Bank under the PDB ID: 1UAO. This protein has been studied in recent years as a system model to understand protein folding mechanisms. *Chignolin*'s native state corresponds to a β-hairpin structure with two hydrogen bonds:

- one between the nitrogen atom N of the third residue and the oxygen atom O of the eighth residue (this hydrogen bond will be denoted as Asp3N-Thr8O);
- one between the oxygen atom O of the third residue and the nitrogen atom N or the seventh residue (this hydrogen bond will be denoted as Asp3O-Gly7N).

It has been established that *chignolin* has a misfolded state where the first hydrogen bond Asp3N-Thr8O is replaced by another hydrogen bond between the nitrogen atom N of the third residue and the oxygen atom O of the seventh residue (this hydrogen bond will be denoted as ASP3N-Gly7O). The computing time required to find a transition between the folded and misfolded states of *chignolin* provides a good metric to compare our conformational landscape exploration algorithms.

3.3.2 Experiment setup

In order to compare the efficiency of each algorithm to explore and discover transition paths between states, the same amount of computing time is allocated to each of them (20 minutes for the *met-enkephalin* and 8 hours for the *chignolin*). All the experiments are performed using the mechanistic model described in section 2.1. The algorithms only explore the backbone DOFs. This considerably reduces the dimension of the conformational space to be explored as there are only two DOFs per residue, corresponding to the ϕ and ψ dihedral angles.

The side-chains are placed following a common procedure independent from the exploration algorithm. Before each energy evaluation, a short MC minimization is run on the side-chains DOFs using the Metropolis Criterion with a very low temperature (0.1 K) and the side-chain configurations are extracted from the lowest energy configuration. At each iteration of this MC minimization, a random number

of side-chain dihedral angles are perturbed by adding a random value, sampled in the interval $[-0.1, 0.1]$ radian, to their original value.

3.3.2.1 Parameter settings

Energy and temperature: Energy evaluation is performed using the AMBER parm96 force-field [Kollman 1997] with an implicit representation of the solvent using the Generalized Born (GB) approximation (same as in Section 2.4.1). In addition to the Metropolis Criterion, an energy threshold E_{max} was set above which no conformations were accepted. This parameter was set to 0 kcal/mol[1] for *met-enkephalin* and to -300 kcal/mol[2] for *chignolin*. The value of Boltzmann constant used for this experiment is $k_b = 0.00198721$ kcal/(mol.K)[3]. Except for the basic MC method, all the algorithms follow the same strategy for the temperature update, described in Algorithm 3.2. The temperature rate parameter T_{rate} is set to 0.1 in all the cases. In order to limit the temperature to reasonable values, a threshold T_{max} was set over which the temperature would not increase anymore. When the temperature update yields a temperature superior to T_{max}, the temperature is set to T_{max}. The temperature of the MC method was set to the average temperature of the T-RRT run (300 K).

T-RRT: An important parameter in the T-RRT algorithm is the distance function used for the nearest neighbor search. We use a distance function that weights each dihedral angle according to its position in the system. If q_1 and q_2 are two conformations of respective coordinates $(\phi_1^{(q_1)}, \psi_1^{(q_1)}, \ldots, \phi_n^{(q_1)}, \psi_n^{(q_1)})$ and $(\phi_1^{(q_2)}, \psi_1^{(q_2)}, \ldots, \phi_n^{(q_2)}, \psi_n^{(q_2)})$, the distance between q_1 and q_2 is defined by:

$$distance(q_1, q_2) = \sqrt{\sum_{i=1}^{n} w_{\phi i} \cdot \mathrm{ad}(\phi_i^{(q_1)}, \phi_i^{(q_2)}) + w_{\psi i} \cdot \mathrm{ad}(\psi_i^{(q_1)}, \psi_i^{(q_2)})} \tag{3.7}$$

where $\mathrm{ad}(\theta_1, \theta_2)$ is the angular distance between θ_1 and θ_2, and w_θ is the weight associated to this variable θ. The weights are computed from the initial (folded) conformation of the studied system. It is equal to the maximum distance between an atom of the molecule and the axis of each dihedral angle. Hence, the distance is measured in ångström.

The extend operation in T-RRT performs a linear interpolation of all the DOFs (the backbone dihedral angles) between q_{near} and q_{rand} and generates a new state q_{new} at distance δ from q_{near}. For both test systems, δ was set to 0.08 Å.

[1] These energy values do not correspond to the total energy: our in-house implementation of the AMBER force-field excludes the terms which are constant under the rigid geometry assumption (rigid bond length and bond angle).

[2] See footnote 1.

[3] In fact, this value corresponds to $k_b \cdot N_A$ where N_A is the Avogadro constant. This is necessary as the used energy functions provides energies in kcal/mol.

Monte Carlo method: At each iteration of the MC method, the last accepted conformation is perturbed using the following move class: for each backbone dihedral angle, a random value sampled in the interval $[-\delta_b, \delta_b]$ is added to its original value. For both test systems, δ_b was set to 0.02 rad. It should be noted that the move classes described in Chapter 2, which were devised for proteins, are not usable for the molecular systems used in this experiment since they are too small.

EST$_{ss}$: The EST$_{ss}$ algorithm uses the same perturbation method as the MC method with the same value for δ_b. The β parameter described in Section 3.2.2.1 is set to 0.5 for both systems.

EST$_{kpiece}$: The EST$_{kpiece}$ algorithm uses the same perturbation method as the MC method with the same value for δ_b. For both systems, the projection used by the algorithm is an orthogonal projection on two characteristic dihedral angles:

- for *met-enkephalin*, the two backbone dihedral angles ϕ_3 and ψ_3 of the central glycine (third residue);
- for *chignolin*, the two backbone dihedral angles ϕ_7 and ψ_7 of the seventh residue (Glycine), whose flexibility has been identified as key in the folding process.

The d_1 and d_2 parameters setting the dimension of the grid cells are both set to 0.02 rad. The α, β, and P_{ext} parameters described in Section 3.2.2.2 are set to $\alpha = 0.7$, $\beta = 0.2$, and $P_{bias} = 0.7$.

3.3.3 Results

3.3.3.1 Met-enkephalin

Figure 3.1 shows the coverage of the conformational landscape achieved by each algorithm after a 20 minutes run. Each heat-map represents the projection of the *met-enkephalin* energy landscape on the dihedral angles of the third residue ϕ_3 and ψ_3 (left column) and on the dihedral angles of the fourth residue ϕ_4 and ψ_4 (right column). These 2-D maps were generated using an exhaustive search procedure by varying both dihedral angles with a 10° step size and finding the lowest energy conformation corresponding to each (ϕ, ψ) pair using a MC-based minimization procedure [Devaurs 2015]. Displayed energy values correspond to relative energies with respect to the lowest-energy conformation. The black points represent the conformations that were explored by the algorithms.

EST$_{ss}$ and EST$_{kpiece}$ both show poor exploration capabilities. The MC method performs a little better but the short running time did not permit the method to explore further than the initial energy basin. The space covered by T-RRT is much wider. The projection on the (ϕ_4, ψ_4) angles even shows that the simulation crossed a relatively high energy saddle region between two low-energy basins. If the good performances of the T-RRT algorithm were expected, it seems surprising that

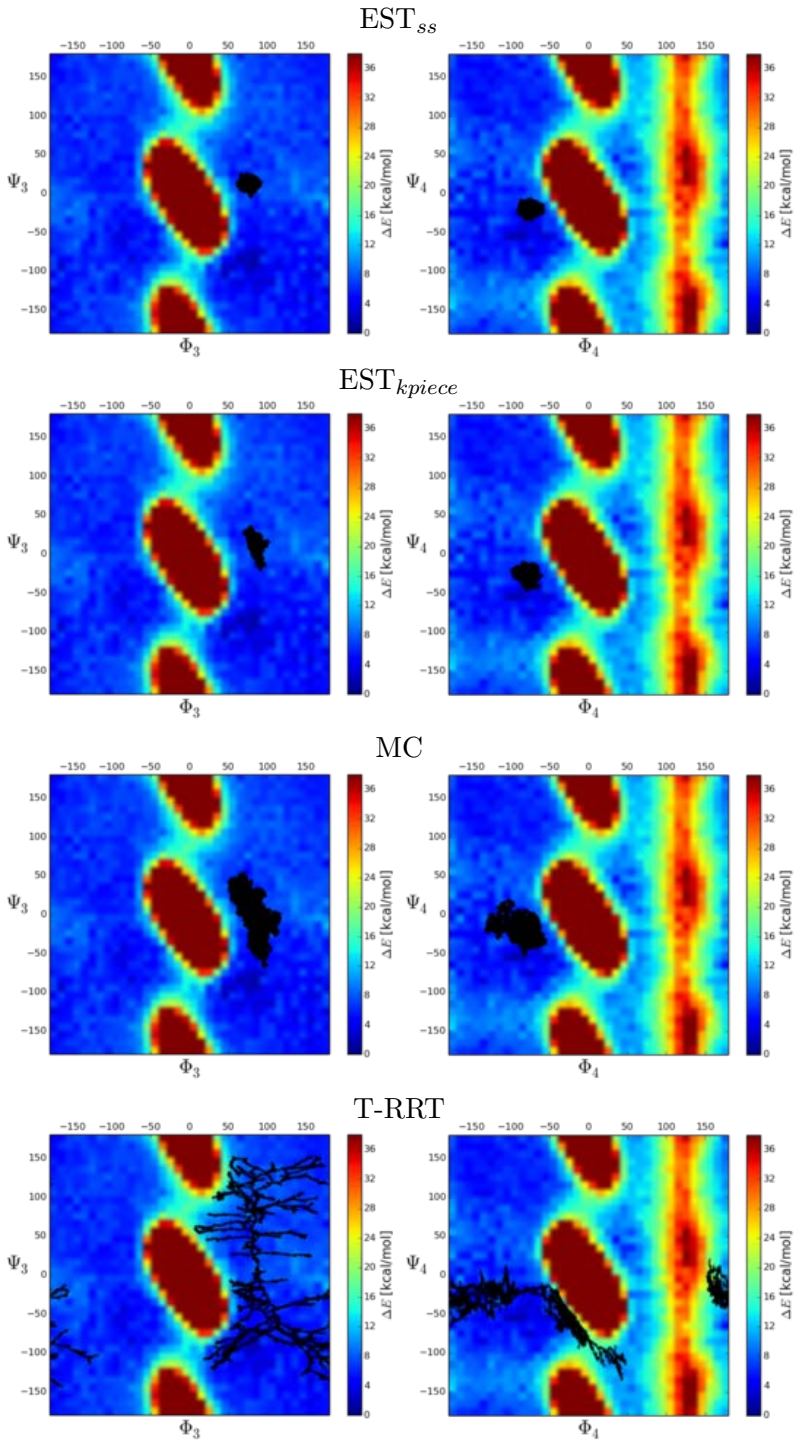


Figure 3.1: *Met-enkephalin* conformational space coverage of the EST$_{ss}$, EST$_{kpiece}$, MC, and T-RRT algorithms projected on the (ϕ, ψ) angles of the third (resp. fourth) residue on the left (resp. right). Black points represents conformations that were explored by the algorithms.

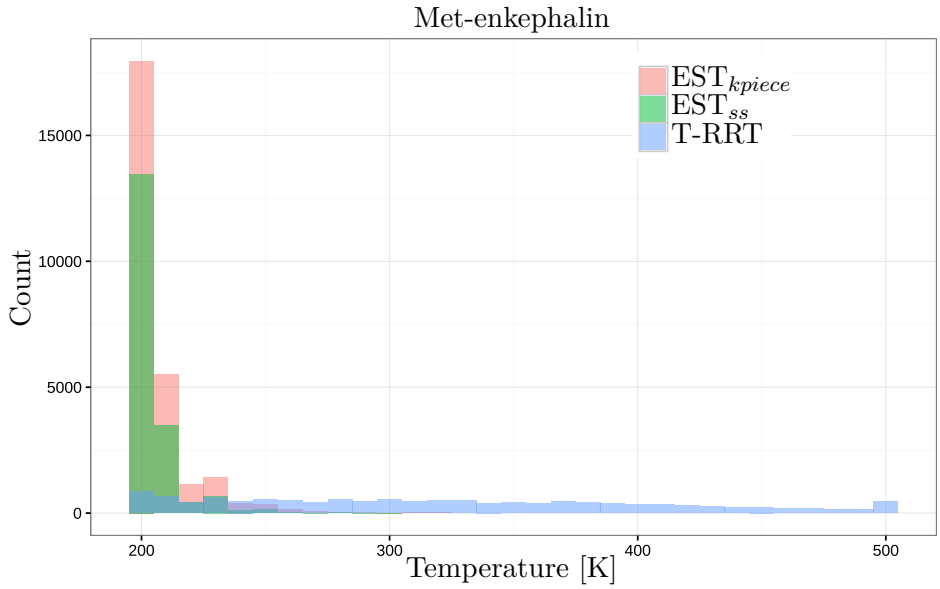


Figure 3.2: Temperature distribution during the EST$_{ss}$, EST$_{kpiece}$, and T-RRT runs on *met-enkephalin*. The temperature of the MC run is constant at 300 K.

the EST$_{ss}$ and the EST$_{kpiece}$ algorithm both perform worse than the MC method considering that they share the same transition test as T-RRT, with the variable temperature parameter. The distribution of the temperature during each run is plotted in Figure 3.2. While the temperature of T-RRT is evenly spread between 200 K and 500 K with a mean of 323 K, the temperature of EST$_{ss}$ and EST$_{kpiece}$ are right-skewed with an average of 205 K and 207 K respectively. These low temperatures bring to light the reason why these two algorithms could not get out of the initial energy basin.

This behavior can be explained by considering the temperature update strategy in combination with the conformation perturbation strategy. In the T-RRT algorithm, the Voronoï bias implies that most of the new configurations q_{new} are pushed away from the initial energy basin. This means that many new configurations q_{new} will have a higher energy than the nearest node conformation q_{near} yielding a high probability of increasing the temperature. In the cases of the EST algorithms, for which the node selection is biased, but for which the conformation perturbation is not, the probability of sampling a conformation in the direction of the positive energy slope are lower. This means that in many cases, the new configurations will have an energy lower or only slightly higher than the selected configuration, yielding a high probability of decreasing the temperature.

3.3.3.2 Chignolin

Figure 3.3 shows the conformational energy landscape coverage achieved by each algorithm after an 8 hour run. For each simulation, two plots were generated corre-

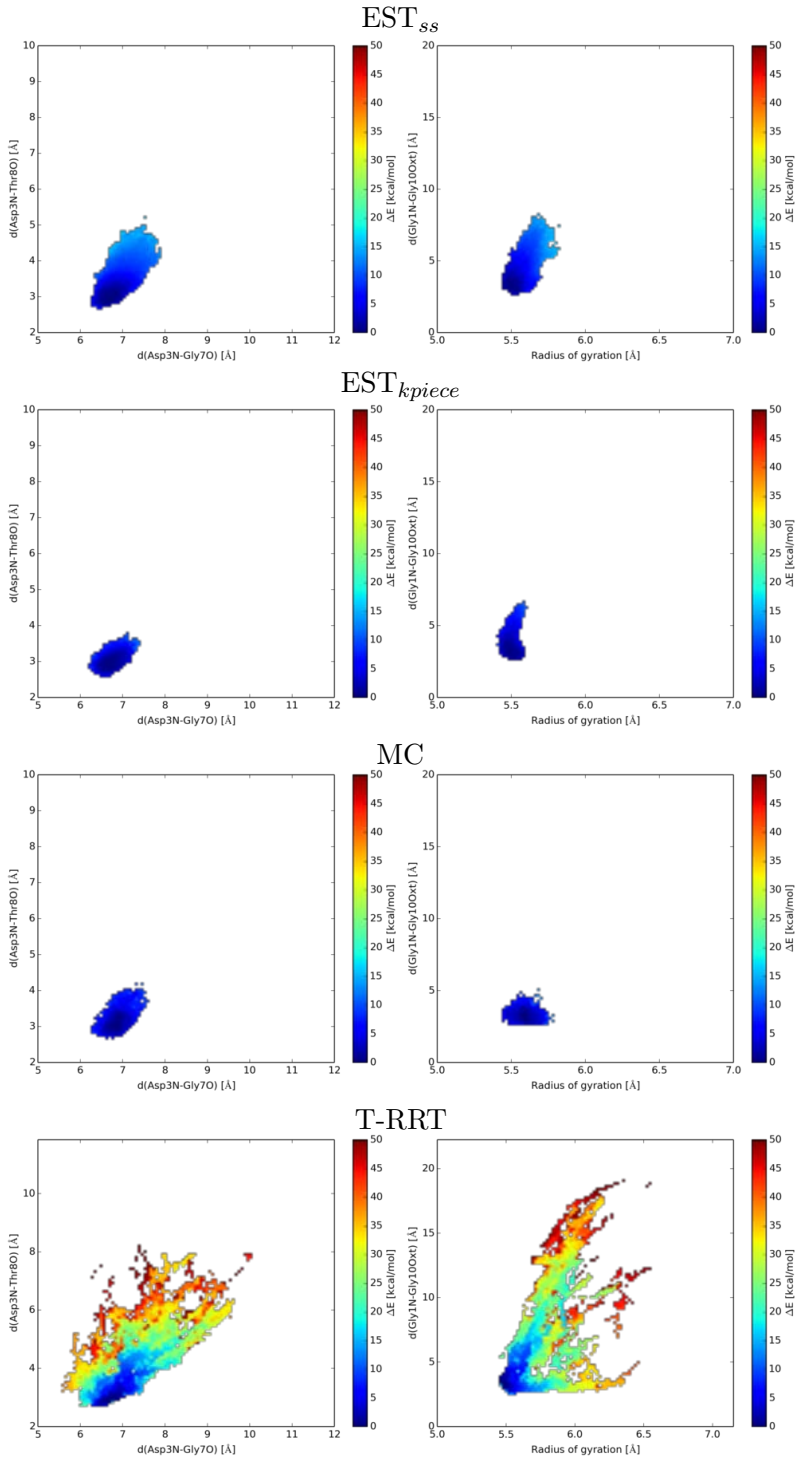


Figure 3.3: *Chignolin* conformational space coverage of the EST$_{ss}$, EST$_{kpiece}$, MC, and T-RRT algorithms.

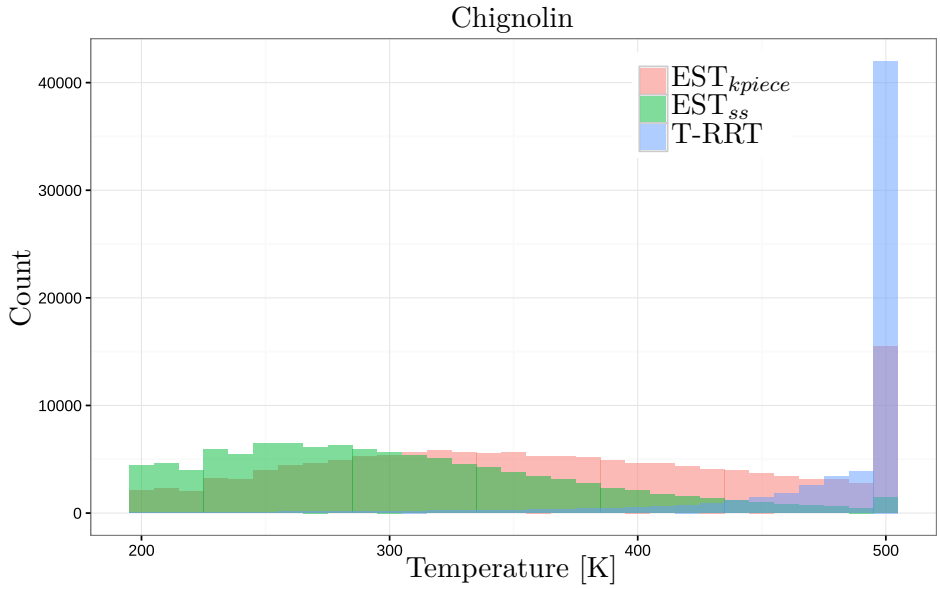


Figure 3.4: Temperature distribution during the EST$_{ss}$, EST$_{kpiece}$, and T-RRT runs on *chignolin*. The temperature of the MC run is constant at 300 K.

sponding to two different projections. The first projection, on the left, corresponds to the distances between the donor and acceptor atoms in the Asp3N-Gly7O hydrogen bond (abscissa) and in the Asp3N-Thr8O hydrogen bond (ordinate). The second projection, on the right, corresponds to the radius of gyration (abscissa) versus the distance between the N atom of the N-terminus and the Oxt atom of the C-terminus. The space covered by each plot was subdivided in a 100×100 grid. The color of each cell of the grid corresponds to the minimum energy conformation projected on it. White areas indicate parts of the conformational space that were not explored by the algorithm. Energy values are relative to the global minimum obtained during the exploration.

As for *met-enkephalin*, the EST$_{kpiece}$ performs poorly, and it appears that the MC method does not perform better. The EST$_{ss}$ algorithm, this time, seems to explore a wider area. Yet, similarly to what was observed for *met-enkephalin*, T-RRT explores a much wider area than the other algorithms. Nevertheless, none of the algorithm was able to find the misfolded state reported in the literature [Satoh 2006, Harada 2011] during the 8 hour runs. Once again, the distribution of the temperature during the different simulations, plotted in Figure 3.4, gives an insight of the reasons why T-RRT performs much better than the other algorithms. The temperature during the T-RRT simulation reached the threshold value of 500 K and had a mean of 480 K. If this allowed the exploration of a much larger area of the conformational space, it is reasonable to think that some of the reached conformations are not realistic. The temperature are more evenly distributed for the two EST algorithms. EST$_{kpiece}$ has an overall higher temperature with an average of 364 K against the average of 305 K for the EST$_{ss}$ simulation.

3.4 Conclusion

In this Chapter, we have presented three algorithms designed to efficiently explore the conformational energy landscape of flexible molecules: the T-RRT algorithm, which had already proven its efficiency, and two EST-based algorithms incorporating the same transition test as T-RRT (EST_{ss} and EST_{kpiece}). These three algorithms have been compared on two small systems with a simple MC method using an unbiased sampling. The results of this comparison indicate that the greediness of T-RRT toward the exploration of new region of the space, yielded by the Voronoï bias, makes it much more efficient at exploring large areas of the conformational space in a short time. Nevertheless, for higher dimension problems where the probability of finding a transition path is low, this greediness seems to push T-RRT toward high energy regions of the space.

If the two EST-based algorithms performed poorly when combined with the temperature update mechanism, it should be mentioned that they are very simple approaches that could be improved. The temperature update mechanism, which is very adapted for the T-RRT, could be modified to take into account the specificity of each algorithm. For the EST_{kpiece} algorithm, the temperature could be dependent on the cell score. For the EST_{ss} algorithm, the temperature could also take into account the success score of each node. These adaptations, together with further improvements of the T-RRT algorithm seem to be promising directions for future work.

Chapter 4

Toward protein motion design

Contents

4.1 Problem formulation and approach 64
4.1.1 Problem definition 64
4.1.2 Approach 65
4.2 Algorithm 66
4.2.1 Simultaneous Design And Path-planning algorithm 66
4.2.2 Controling tree expansion 67
4.2.3 Theoretical analysis 68
4.3 Empirical analysis and results 69
4.3.1 Test system description 70
4.3.2 Benchmark results 72
4.4 Application SDAP to protein motion design 76
4.4.1 Problem definition 76
4.4.2 Additional simplifications 77
4.4.3 Preliminary experiments 79
4.5 Conclusions and future work 81

System design and path planning problems are usually treated independently. In robotics, criteria such as workspace volume, workload, accuracy, robustness, stiffness, and other performance indexes are treated as part of the system design [Gosselin 1991, Merlet 2005]. Path planning algorithms are typically applied to systems with completely fixed geometric and kinematic features. In this chapter, we propose an extension of the path planning problem, in which some features of the mobile system are not fixed *a priori*. The goal is to find the best design (i.e. values for the variable features) to optimize the motion between given configurations.

A brute-force approach to solve this problem would consist of individually solving motion planning problems for all possible designs, and then selecting the design providing the best result for the (path-dependent) objective function. However, because of the combinatorial explosion, only simple problems involving a small number of variable design features can be treated using this naive approach. In this chapter, we propose a more sophisticated approach that simultaneously considers system design and path planning. A related problem is the optimization of geometric and kinematic parameters of a robot to achieve a given end-effector trajectory, usually referred to as kinematic synthesis [McCarthy 2001]. Nonetheless, the problem we

address in this work (see Section 4.1.1 for details) is significantly different, since we assume that all kinematic parameters and part of the geometry of the mobile system are provided as input. The design concerns a discrete set of features that can be associated to the bodies of the mobile system, such as shape or electrostatic charge, aiming to find the best possible path between two given configurations provided a path cost function. Very few works have considered such a hybrid design and path planning problem. One of the rare examples is a recently proposed method for UAV (Unmanned Aerial Vehicle) path planning [Rudnick-Cohen 2015] where the optimal path planning algorithm considers several possible flying speeds and wing reference areas to minimize path risk and time. Since the considered configuration space is two-dimensional, the proposed solution is based on an extension of Dijkstra's algorithm working on a discrete representation of the search-space. This type of approach cannot be applied in practice to higher-dimensional problems, such as the ones we address.

Sampling-based algorithms have been developed since the late 90s for path planning in high-dimensional spaces [Kavraki 1996, LaValle 2006], which are out of reach for deterministic, complete algorithms. Our work builds on this family of algorithms, which we extend to treat a combinatorial component in the search-space, associated to the systems design, while searching for the solution path. Our approach presents some similarities with methods that extend sampling-based path planning algorithms to solve more complex problems such as manipulation planning [Siméon 2004] or multi-modal motion planning [Hauser 2010], which also involve search-spaces with hybrid structure. As in these other works, the proposed algorithm simultaneously explores multiple sub-spaces aiming to find solutions more efficiently. Nevertheless, the hybrid design problem addressed here is completely different.

This chapter presents a more sophisticated approach, the Simultaneous Design And Path-planning algorithm (SDAP), which is based on the T-RRT algorithm [Jaillet 2010]. As explained in Section 4.1.2, the choice of T-RRT as a baseline is guided by the type of cost function we apply for the evaluation of path quality. Nevertheless, other sampling-based algorithms, as, for instance, the EST-based algorithms presented in Chapter 3, can be extended following a similar approach.

The good performance of the SDAP algorithm method is demonstrated on relatively simple, academic examples (Section 4.3). These simple examples allow us to apply the naive exhaustive method, whose results can be used as a reference to evaluate the performance and the quality of the solutions produced by the SDAP algorithm. Results show that SDAP is able to find the best path-design pairs, requiring much less computing time than the naive method. This advantage increases with the complexity of the problem.

Although the problem tackled in this chapter is formulated in a general manner, as an extension of the standard path planning problem in robotics, our goal in a close future is to address problems related to the design of proteins (or protein fragments) to perform specific motion. Section 4.4 explains how these two problems are equivalent and shows how SDAP can be used to solve a CPD problem. In

addition to computational protein design, applications of the proposed approach in robotics can be envisioned, as briefly mentioned in Section 4.5.

4.1 Problem formulation and approach

This section defines the design problem that will be addressed in this chapter, along with some notation, and presents an overview of the proposed approach.

4.1.1 Problem definition

Let us consider an articulated linkage $\mathcal{A}$ consisting of n rigid bodies, $A_1, \ldots, A_n$. The kinematic parameters of $\mathcal{A}$ are static and are supplied as input. The geometry of the rigid bodies A_i can admit some variability, as well as other physical properties (mass, electrostatic charge, ...). More precisely, a discrete set of m design features, $f_1, \ldots, f_m$, is defined and each body $A_i \in \mathcal{A}$ is assigned a design feature $f_j \in \mathcal{F}$. We denote d as a vector of length n that represents the design features assigned to all the rigid bodies in $\mathcal{A}$, i.e. d defines a particular design. $\mathcal{D}$ denotes the set of possible combinations of assignments of features for $\mathcal{A}$, i.e. $\mathcal{D}$ defines all possible designs. $\mathcal{D}$ is referred to as the design space, which is a discrete space containing m^n elements. A given configuration of $\mathcal{A}$ is denoted by q. Let $\mathcal{C}$ denote the configuration space. Note that for each $q \in \mathcal{C}$, only a subset of the possible designs $\mathcal{D}$ can be assigned, since some designs are not compatible with some configurations due to self-constraints or environment constraints.

The workspace of $\mathcal{A}$ is constrained by a set of obstacles $O_i \in \mathcal{O}$. $\mathcal{C}^d_{free}$ denotes all valid, collision-free configurations of $\mathcal{A}$ for a given vector d of design features. A path P connecting two configurations q_{init} and q_{goal} of $\mathcal{A}$ with design d is defined as a continuous function $P : [0,1] \to \mathcal{C}$, such that $P(0) = q_{\text{init}}$ and $P(1) = q_{\text{goal}}$. The path is said to be collision-free if $\forall t \in [0,1], P(t) \in \mathcal{C}^d_{free}$. $\mathcal{C}_{free}$ is the union of all individual $\mathcal{C}^d_{free}$:

$$\mathcal{C}_{free} = \bigcup_{d \in \mathcal{D}} \mathcal{C}^d_{free}$$

$\mathcal{P}_{free}$ denotes the set of all feasible, collision-free paths connecting q_{init} to q_{goal}, considering all possible designs ($\forall d \in \mathcal{D}$).

A cost function $c : \mathcal{C}_{free} \times \mathcal{D} \to \mathbb{R}_+$ associates to each pair (q, d) a positive cost value, $\forall q \in \mathcal{C}_{free}$ and $\forall d \in \mathcal{D}$. Another cost function $c_P : \mathcal{P}_{free} \times \mathcal{D} \to \mathbb{R}_+$ is also defined to evaluate the quality of paths. In this work, the path cost function c_P is itself a function of the configuration cost function c, i.e. c_P is a functional. More precisely, we consider the *mechanical work* criterion as defined in [Jaillet 2010, Devaurs 2016] to evaluate paths, which aims to minimize the variation of the configuration cost c along the path. This criterion is a suitable choice to evaluate path quality in many situations [Jaillet 2010], and is particularly relevant in the context of molecular modeling. Nevertheless, other cost functions can be considered, such as the integral of c along the path. A discrete approximation, with

constant step size $\delta = 1/l$, of the mechanical work (MW) cost of a path P for a system design d can be defined as:

$$c_P(P,d) = \sum_{k=1}^{l} \max\left\{0\,,\; c\left(P\left(\frac{k}{l}\right),d\right) - c\left(P\left(\frac{k-1}{l}\right),d\right)\right\} \tag{4.1}$$

The goal of our method is to find the best pair (P^*, d^*) such that:

$$c_P(P^*,d^*) = \min\{c_P(P,d) \mid P \in \mathcal{P}_{free}, d \in \mathcal{D}\} \tag{4.2}$$

4.1.2 Approach

A naive approach to solve the problem would be to compute the optimal cost path for each design $d \in \mathcal{D}$, and then choose the optimal design d^* that minimizes c_P. Such a brute-force approach can be applied in practice to simple problems involving a small number n of variable bodies and/or a few m design features (recall that the design space is size m^n). The method proposed below aims to solve the problem much more efficiently by combining both the discrete (design) and continuous (path) optimization in a single stage.

We assume that, for most problems of interest, the configuration space $\mathcal{C}$ is high-dimensional, so that exact/complete algorithms cannot be applied in practice to solve the path-planning part of the problem. For this, we build sampling-based algorithms [Lavalle 2000, Kavraki 1996]. We also assume that the cardinality of the design space $\mathcal{D}$ is moderately high, such that a relatively simple combinatorial approach can be applied to treat the design part of the problem.

The idea is to explore $\mathcal{C}_{free}$ to find paths between q_{init} and q_{goal} simultaneously considering all possible designs $d \in \mathcal{D}$. To reduce the number of configuration-design pairs (q,d) to be evaluated during the exploration, it is important to apply an effective filtering strategy. The choice of the particular sampling-based path planning algorithm and filtering strategy mainly depend on the type of objective function c_P being considered. The approach described below has been developed to find good-quality solutions with respect to the MW path evaluation criterion (4.1). In this work, we extend the T-RRT algorithm (explained in Section 3.2.1), which finds paths that tend to minimize cost variation by filtering during the exploration tree nodes that would produce a steep cost increase. Following a similar approach, alternative algorithms and the associated filtering strategies could be developed to optimize other path cost functions. For instance, variants of RRT* [Karaman 2011] or FMT* [Janson 2015] could be considered for optimizing other types of monotonically increasing cost functions.

4.2 Algorithm

This section presents the SDAP algorithm, building upon a single-tree version of T-RRT. However, the approach is directly applicable to multi-tree variants which

Algorithm 4.1: SDAP Algorithm

input : the configuration space $\mathcal{C}$; the design space $\mathcal{D}$; the cost function c;
the start state q_{init}; the goal state q_{goal};
number of iterations $MaxIter$
output: the tree $\mathcal{T}$

```
1  T ← InitTree(q_init, D)
2  while not StoppingCriterion (T, q_goal, MaxIter) do
3      q_rand ←Sample(C)
4      Neighbors ← NearestNeighbors(T, q_rand, D)
5      TransitionTest.Init()
6      for s_near ∈ Neighbors do
7          q_new ← Extend(q_rand, s_near)
8          D ← TransitionTest(T, s_near, q_new, c)
9          if not Empty(D) then
10             AddNode(T, s_near, q_new, D)
```

are more efficient at solving path planning problems [Devaurs 2014]. First, the basic algorithm is introduced, followed by additional explanation on the tree extension strategy and a brief theoretical analysis.

4.2.1 Simultaneous Design And Path-planning algorithm

The Simultaneous Design And Path-planning algorithm (SDAP) pseudo-code is shown in Algorithm 4.1. A search tree, $\mathcal{T}$, is created with q_{init} as the root node. The tree is grown in configuration space through a series of expansion operations. Each node s in $\mathcal{T}$ encodes a configuration q and a set of designs $D \subseteq \mathcal{D}$ for which the configuration is valid. Each node's set of designs D is a subset of its parent's designs, i.e. $Designs(s) \subseteq Designs(Parent(s))$.

During each iteration, a random configuration (q_{rand}) is generated (line 3). In T-RRT, a new node (q_{new}) is created by expanding the nearest node in $\mathcal{T}$ (q_{near}) in the direction of q_{rand} for a distance δ. q_{new} is then conditionally added to $\mathcal{T}$ based on the transition test explained in Algorithm 3.2. Transitions to lower cost nodes are always accepted and moves to higher costs nodes are probabilistically accepted. The probability to transition to a higher cost node is controlled by a temperature variable T.

SDAP modifies the expansion and transition test functions of the standard T-RRT algorithm in order to address the design and path planning problems simultaneously. At each iteration, SDAP attempts to expand at least one node per design in $\mathcal{D}$. The process is shown in Figure 4.1, where each design $d \in \mathcal{D}$ is encoded as a color on each node of the tree. During each iteration, SDAP expands a set of nodes that covers all designs $\mathcal{D}$. In other words, the `NearestNeighbors` function (line 4) returns a set, *Neighbors*, containing the closest node to q_{rand} for each design. In Figure 4.1, q_{rand} is shown in black and the 3 nodes in the set *Neighbors* are circled

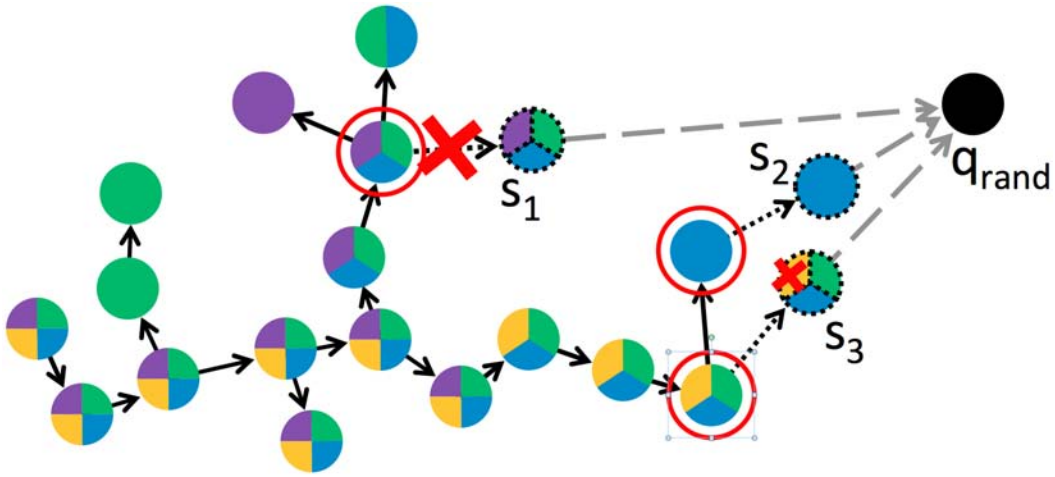


Figure 4.1: An expansion operation for SDAP. Designs are encoded as colors within each node. The nodes being expanded are circled in red. The expansion towards node s_1 fails for all 3 designs, the expansion to s_2 succeeds, and the expansion to s_3 succeeds for 2 of the 3 designs.

in red. Each node in *Neighbors* is extended towards q_{rand} (lines 6 - 10) creating new candidate nodes which are labeled s_1, s_2, and s_3 in Figure 4.1. All 3 designs in s_1 fail the transition test, so the new node is not added to $\mathcal{T}$. For s_2 the blue design passes the transition test and the node is added to $\mathcal{T}$. Finally for s_3, 1 of the 3 designs (yellow) fails the transition test, resulting in a node with 2 designs being added to $\mathcal{T}$.

4.2.2 Controling tree expansion

The transition test of T-RRT is governed by the temperature parameter (T). As explained in Section 3.2.1, it automatically adjusts T during the exploration. This has been shown highly effective in balancing tree exploration and tree refinement [Jaillet 2010]. At each iteration, T-RRT adjusts T by monitoring the acceptance rate of new nodes. SDAP extends this idea by maintaining a separate temperature variable $T(d)$ for each design $d \in \mathcal{D}$. A given design d can appear in multiple nodes in *Neighbors*. For each design d, the node in *Neighbors* closest to q_{rand} is identified. The temperature $T(d)$ is adjusted based on the success or failure of the extension operation from this node for design d.

The pseudocode for the transition test function is shown in Algorithm 4.2. The *Neighbors* set is processed in ascending order of the distance of each node from q_{rand} (line 6 of Algorithm 4.1). For the node being expanded (s_{near}), each design d has its cost evaluated (line 4). Transitions to lower cost nodes are always accepted (line 8). Transitions to higher cost nodes are subjected to probabilistic acceptance (line 10). The set V (lines 12 and 18) tracks designs which have had their temperature adjusted during this iteration. The function returns the set D of designs that pass the transition test.

Algorithm 4.2: TransitionTest($\mathcal{T}, s_{\text{near}}, q_{\text{new}}, c$)

input : the input tree $\mathcal{T}$; vector of temperatures T;
parent node s_{near}; new node q_{new}; the cost function c;
temperature adjustment rate T_{rate}; Boltzmann constant K
internal: set of designs V with adjusted temperatures in this iteration
output : vector of designs D that pass the transition test

```
1  S ← ∅
2  for d ∈ Designs(s_near) do
3      if CollisionTest(q_new,d) == False then
4          c_near = c(Config(s_near),d); c_new = c(q_new,d)
5          success ← false
6          Δc = c_new - c_near
7          if Δc < 0 then
8              success ← true
9          else
10             if exp(−Δc / (K · T(d))) > UniformRand() then
11                 success ← true
12             if d ∉ V then
13                 if success then
14                     T(d) ← T(d) / 2^((Δc) / energyRange(T,d))
15                 else
16                     T(d) ← T(d) · 2^T_rate
17         if success then
18             D ← D ∪ d
19         V ← V ∪ d
20 return (D)
```

4.2.3 Theoretical analysis

In this section we provide some theoretical analysis of SDAP algorithm's completeness and path optimality. A theoretical analysis of the complexity of SDAP is difficult because of its stochastic nature. Instead, Section 4.3 provides some empirical results that clearly show SDAP's efficiency compared to an exhaustive search of paths for all possible designs.

4.2.3.1 Probabilistic Completeness

SDAP's probabilistic completeness directly derives from that of RRT [Lavalle 2000], which is inherited by T-RRT under the condition to guarantee a strictly positive probability of passing the transition test as explained in [Jaillet 2010]. Since SDAP maintains this property by incorporating temperatures in the transition test for each given design $d \in \mathcal{D}$, and since the number of design in $\mathcal{D}$ is finite and constant, it also

ensures the positive transition probability and that each $\mathcal{C}^d_{free}$ will be completely sampled, thus maintaining the probabilistic completeness of the algorithm.

4.2.3.2 Path Optimality

The current SDAP implementation is based on T-RRT, which has been empirically shown to compute paths that tend to minimize cost with respect to the MW criterion [Jaillet 2010], but without theoretical guarantee of optimality. Using anytime variants of T-RRT (AT-RRT or T-RRT*) [Devaurs 2016] would provide asymptotic convergence guarantee. Implementing these within SDAP remains as future work.

4.3 Empirical analysis and results

As a proof of concept, SDAP is applied to a set of academic problems. SDAP is implemented as an adaptation of the Multi-T-RRT algorithm [Devaurs 2014], with two trees growing from the initial and goal configurations. The search stops when the algorithm is able to join the two trees. For each problem, SDAP is compared against a naive approach consisting of multiple independent runs of Multi-T-RRT on each designs $d \in \mathcal{D}$.

4.3.1 Test system description

The test system is a 2D articulated mechanism with a fixed geometry surrounded with fixed obstacles. The bodies $A_1, \ldots, A_n$ are circles with radius R. The first body A_1 is a fixed base. The other bodies $A_2, \ldots, A_n$ are articulated by a rotational joint centered on the previous rigid body that can move in the interval $[0, 2\pi)$. A configuration q is described by a vector of $n-1$ angles corresponding to the value of each rotational joint. The features $f_1, \ldots, f_n$ assigned to each body are electrostatic charges in $\mathcal{F} = \{-1, 0, 1\}$ (i.e. $m = 3$). The design vector d contains n charges $f_1, \ldots, f_n$ associated to each rigid body $A_1, \ldots, A_n$ of the mechanism. In the following, d can be written as a string, with each charge (-1, 0, 1) corresponding to N, U, and P respectively. For example, the design NPUN corresponds to the vector $d = (-1, 1, 0, -1)$. Obstacles $O_1, \ldots, O_k$ are circles of radius R and have electrostatic charges with predefined values $g_i \in \mathcal{F}$.

The cost function is inspired by a simple expression of the potential energy of a molecular system. It contains two terms, one corresponding to the Lennard-Jones potential and the other to the electrostatic potential. It is defined as:

$$c(q, d) = LJ(q, d) + ES(q, d) \tag{4.3}$$

with:

$$LJ(q,d) = \varepsilon_H \sum_{i=1}^{|\mathcal{A}|-2} \left[\sum_{j=i+2}^{|\mathcal{A}|} \left(\frac{2 \cdot R}{\|A_i A_j\|} \right)^{12} - \left(\frac{2 \cdot R}{\|A_i A_j\|} \right)^{6} \right] + \varepsilon_H \sum_{i=1}^{|\mathcal{A}|} \left[\sum_{j=1}^{|\mathcal{O}|} \left(\frac{2 \cdot R}{\|A_i O_j\|} \right)^{12} - \left(\frac{2 \cdot R}{\|A_i O_j\|} \right)^{6} \right] \quad (4.4)$$

$$ES(q,d) = \sigma_H \sum_{i=1}^{|\mathcal{A}|-2} \left[\sum_{j=i+2}^{|\mathcal{A}|} \left(\frac{f_i \cdot f_j}{\|A_i A_j\|} \right) \right] + \sum_{i=1}^{|\mathcal{A}|} \left[\sum_{j=1}^{|\mathcal{O}|} \left(\frac{f_i \cdot g_j}{\|A_i O_j\|} \right) \right] \quad (4.5)$$

where ε_H and σ_H sets the energy scale, and where $\|X_i X_j\|$ represents the Euclidean distance between the centers of the bodies/obstacles X_i and X_j.

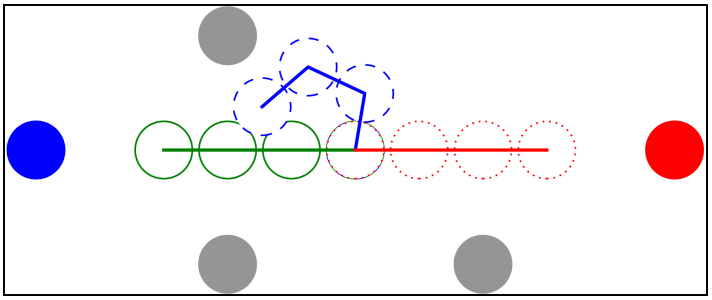
SDAP is empirically tested using a 4 body and a 10 body scenarios described below. The objective is to find the path-design pair (P^*, d^*) that minimizes c_P.

4.3.1.1 Small 4 Body System

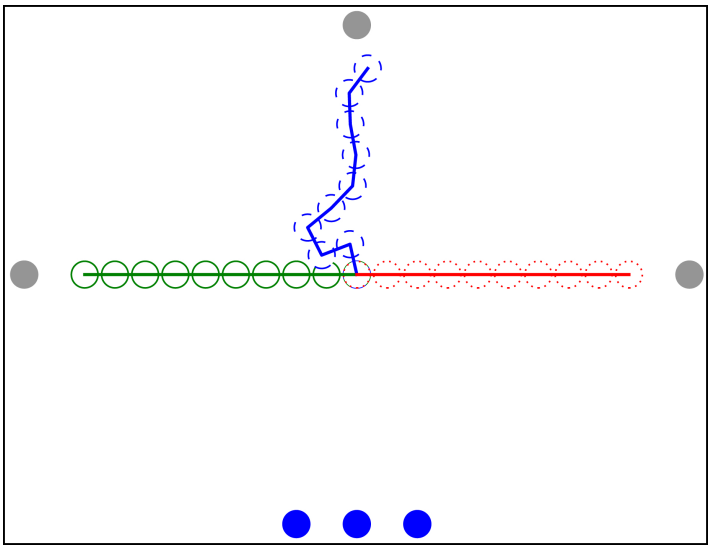
The first system consists of four bodies and five obstacles as shown in Figure 4.2. q_{init} and q_{goal} correspond to fully stretched configurations, to the left and to the right, represented with solid and dashed outlines respectively. The design space consists of $3^4 = 81$ possible combinations and the configuration space is 3 dimensional. This scenario favors designs with a negatively charged end-effector. The uncharged obstacles at the top and bottom of the workspace create a narrow passage that all solutions must pass through. Figure 4.3 shows a projection of the configuration-space costmap for two designs along with a solution path. One design has a negatively charged end-effector (UUUN) and one has a positively charged end-effector (UUUP). Angles 1 and 2 are projected onto the x and y axis respectively, with angle 3 being set to minimize the cost function. For the UUUN design, the costmap is highly favorable to the desired motion, starting at a high cost and proceeding downhill to a low cost area. The costmap associated with the UUUP design shows a non-favorable motion between the two states.

4.3.1.2 Larger 10 Body System

A larger system with 10 bodies and six obstacles is shown in Figure 4.2. The design space $\mathcal{D}$ contains $3^{10} = 59049$ possibilities, which cannot be exhaustively explored within a reasonable computing time, and is also challenging for SDAP because of memory issues (see discussions in Section 4.5). For that reason, two simplified versions of this scenario are constructed. The first one fixes the design for the first seven bodies $A_1, \ldots, A_7$ as UUUUUUU. The remaining bodies (A_8, A_9, and A_{10}) can be designed, resulting in a design space of $3^3 = 27$ designs. The second version expands the design space to the last 4 bodies $(A_7, \ldots, A_{10})$, resulting in a design space of $3^4 = 81$ designs. It both cases, the configuration space is 9



(a)



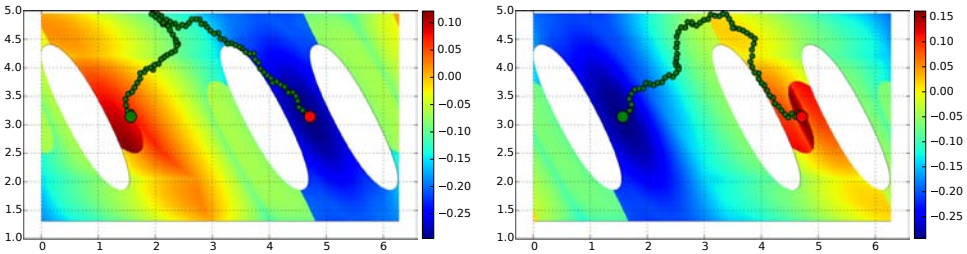
(b)

Figure 4.2: A 4-body (a) and 10-body (b) scenario. Obstacles with positive charges are shown in solid red, negative in solid blue, neutral in gray. The initial state is shown in green with a solid line, a transition state shown in blue, and the goal state is shown in red with a dashed line.

dimensional. Both versions of the 10-body system are constrained with the same obstacles. They were chosen so that designs with strongly positively or negatively charged end effectors will be trapped at local minima resulting from attractive or repulsive forces generated by the bottom obstacles.

4.3.2 Benchmark results

We compare SDAP to a naive approach (solving individual problems for each design $d \in \mathcal{D}$) using the same Multi-T-RRT implementation. In other words, we compare one run of the SDAP algorithm against $|\mathcal{D}|$ runs of a single-design path search. Multiple runs are performed (100 for the 4-body scenario, 50 for the 10-body scenario with 3 designed bodies, and 20 for the 10-body scenario with 4 designed bodies) to reduce the statistical variance inherent with stochastic methods. The single-



(a) (b)

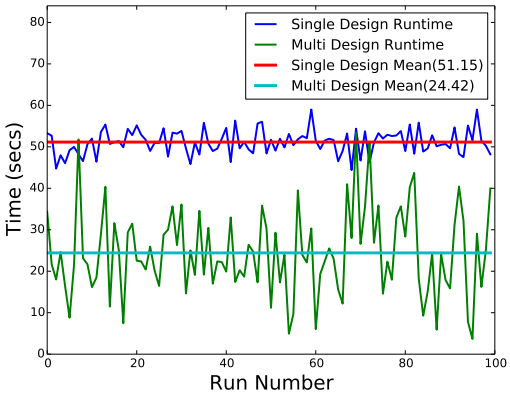
Figure 4.3: Configuration-space costmap of the 4-body system projected onto the first two DOFs of the system expressed in radians. The initial configuration is indicated by the green dot on the left, and the goal by the red dot on the right. (a) Costmap for the UUUP design. (b) Costmap for the UUUN design. Each cell's cost is computed by finding the values of the 3rd DOFs that minimize the cost.

design explorations can spend time trying to escape local minima associated with the costmaps of unfavorable designs, causing very long execution times and high-cost paths. As we are not interested in finding a solution path for every possible design but only for the designs with low-cost paths, a timeout is enforced for the single-design explorations of 300 seconds for the 4-body scenario and 1,200 seconds for the 10-body scenarios. The SDAP algorithm was considered unsuccessful if it did not find a solution within 2,400 seconds.

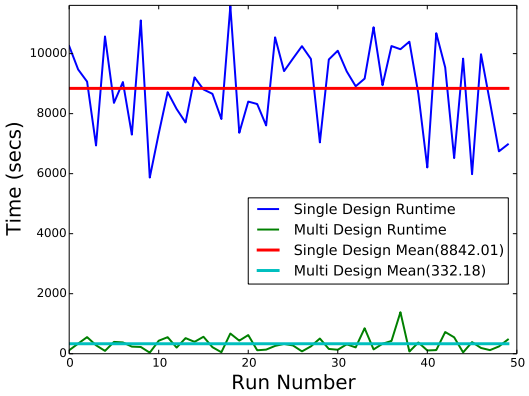
All the runs were performed in a single threaded process on a Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz processor with 32GB of memory.

4.3.2.1 Small Scenario Results

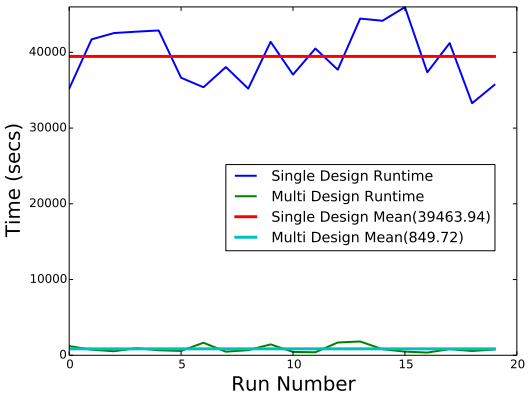
The runtimes for the small scenarios are shown in Figure 4.4 (a). For the single-design approach, the summed runtimes for the 81 runs to cover $\mathcal{D}$ are plotted versus the SDAP runtime. The figure shows that SDAP is twice as fast as the single-design approach. Although the variance in execution time for SDAP seems much larger than for the naive approach, recall that each complete run of the latter involves 81 runs of the Multi-T-RRT algorithm, which attenuates the overall variance. However, the computing time variance for a specific design can be much larger. Figure 4.5 (a) compares the solutions found by the two methods. SDAP successfully identifies the designs corresponding to the lowest-cost paths. Recall that the current implementation of SDAP terminates when one valid path is found. Asymptotic convergence to the global optimum could be guaranteed by implementing an anytime variant of the algorithm such as AT-RRT [Devaurs 2016] (this remains as future work). The high density of nodes created by the SDAP algorithm (19,784 nodes on average compared to 698 for each single-design search) could be well exploited to improve the path cost by incrementally adding cycles.



(a)

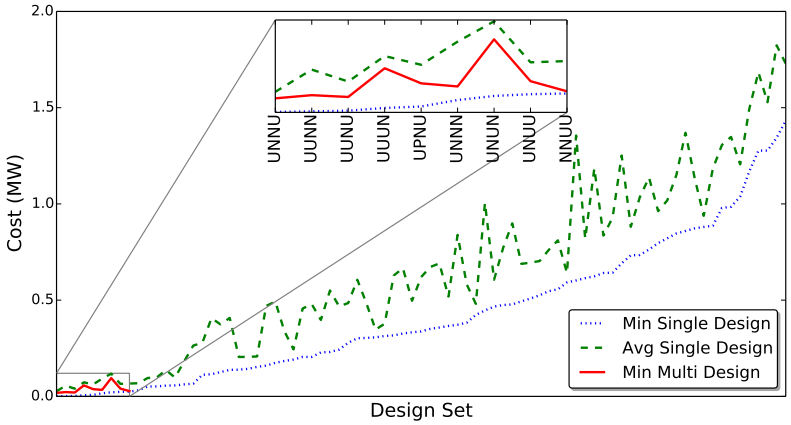


(b)

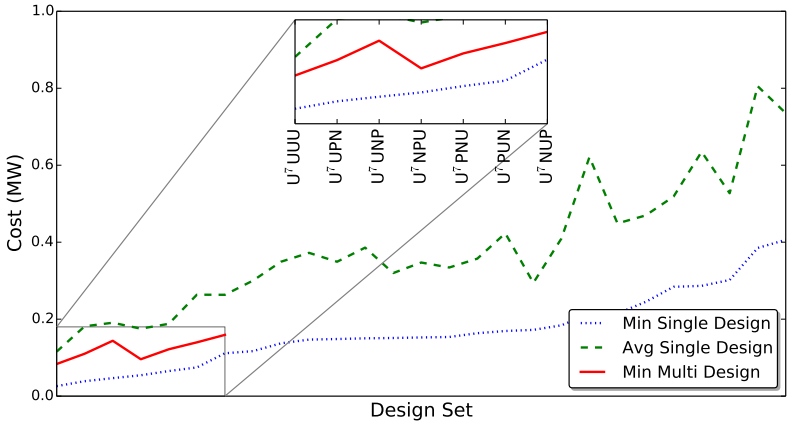


(c)

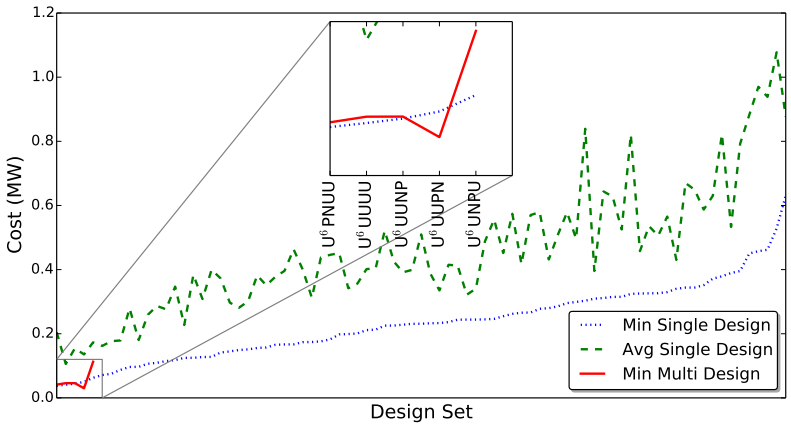
Figure 4.4: Run times comparisons for the 4-body scenario (a), 10-body scenario with 3 designed bodies (b) and 4 designed bodies (c). SDAP (green line) is compared against an exhaustive search using single-design T-RRT explorations (blue line).



(a)



(b)



(c)

Figure 4.5: The best cost paths for single-design T-RRT runs (dotted blue and dashed green lines) and SDAP (red line) for the small scenario (a) and larger 3-designed-body (b) and 4-designed-body (c). Designs set members are ordered by the minimum-cost found by the single-design runs. SDAP solutions shown only for discovered paths (does not exhaustively search). SDAP discovers the same low-cost designs as the exhaustive single-design searches.

4.3.2.2 Larger scenario results

The runtimes for both versions of the larger scenario are shown in Figure 4.4 (b) and (c). In both cases, the difference in computing time between the two approaches increases significantly compared to the 4-body scenario. In the 3-designed-body version, SDAP is 26 times faster than the single-design search on average. In the 4-designed-body version, SDAP is 46 times faster. Note that, while the cardinality of design space $\mathcal{D}$ is multiplied by 3 between the two versions of the large scenario, the execution time of SDAP is only multiplied by 2.5 on average. The variance of the execution times is now lower for SDAP compared to the naive approach. The reason is that the performance of Multi-T-RRT highly depends on the roughness of the configuration-space costmap. In a smooth costmap, Multi-T-RRT will be quite fast with a low variance, whereas the time required to find a solution in a rugged costmap will be higher and will have a larger variance. SDAP's computing time is only dependent on the difficulty to find the best designs, which typically have a smoother costmap, whereas the single-design search has to find a solution for every design, including those with a very rugged costmap. The 4-body scenarios is relatively simple, and thus even for very bad designs, a solution was found in close-to-constant time. But for the 10-body scenario, the problem is more complex, and the 27 (resp. 81) runs are not enough to attenuate a very high variance.

The single-design search reached the timeout 4 times over the 27 runs on average for the 3-designed-body version of the 10-body scenario, and 19 times over the 81 runs on average for the 4-designed-body version of the problem. The SDAP algorithm always found a solution before the timeout.

Figure 4.5 (b) and (c) compares the solutions found by the two searches. Once again, SDAP successfully identifies designs that yield the best path cost.

4.4 Application SDAP to protein motion design

If SDAP shows good results on simple academic problems, applying it to a protein design problem is not straightforward. In this section, we consider the problem of designing a protein region (a loop) aiming to facilitate the transition between two stable states and we show how this problem can be approached by the formal formulation in Section 4.1.1.

4.4.1 Problem definition

The problem of protein design for a chosen motion can be formulated as follow: given two protein scaffold conformations q_1 and q_2, find a sequence of n amino acids with (meta-)stable states applying to both structures, and for which a feasible motion between q_1 and q_2 exists. As explained in Section 1.5.4, if multi-state protein design methods allowing to design a protein for states q_1 and q_2 exist, they do not give any information on the existence of a transition between those two states. In order to design a candidate protein for this motion, it is necessary to find a couple (path

between q_1 and q_2, sequence of amino acids) that minimizes energy variation (i.e. the mechanical work) from q_1 to q_2 and from q_2 to q_1. This corresponds to the problem defined in Section 4.1.1.

A parallel can be drawn between the choice of a sequence for the protein and the choice of a design for an articulated linkage: the set of all possible designs $\mathcal{D}$ corresponds to the set of all possible sequences and the design features $f_1, \ldots, f_{20}$ corresponds to the 20 natural amino acids. The bodies $A_1, \ldots, A_n$ to which the design features are assigned are the protein fragments corresponding to each residue. This differs from the initial definition of the bodies $A_1, \ldots, A_n$ since the residues are not simple rigid bodies: they have DOFs corresponding to their backbone, which are similar for each residue, and they have other DOFs corresponding to the side-chains that will depend on the assigned amino acid. The backbone DOFs being independent from the design, it is quite straightforward to extend the problem to include that particular case: the configuration space $\mathcal{C}$ corresponds to the backbone conformational space. The side-chains DOFs are more problematic as they are dependent on the design, but as will be explained later, this problem can be overcome. The problem of finding a motion between protein scaffold conformations q_1 and q_2 corresponds to the problem of finding a collision free path P between the backbone conformations corresponding to q_1 and q_2. Another parallel can be drawn between the notion of cost function, and the notion of energy function: the energy function takes a sequence and a conformation as inputs and associates them a real value representing the potential energy. The main difference with the cost function defined in Section 4.1.1 is that, in this case, the energy function is evaluated for the full protein conformation (including the side-chains) while the configuration space $\mathcal{C}$ only determines the backbone DOFs. There are multiple ways to deal with this problem. We will discuss two of them:

- Let $\mathcal{C}_s^d$ be the set representing the conformations of the side-chains for a design d. Let E be an energy function that associates an energy value to a full conformation. Define $E_d : \mathcal{C} \times \mathcal{C}_s^d \to \mathbb{R}$ as the energy function associated to the design d. We can define the cost function $c : \mathcal{C} \times \mathcal{D} \to \mathbb{R}$ as:

$$c(q, d) = \min_{q_s \in \mathcal{C}_s^d} E_d(q, q_s) \tag{4.6}$$

  This definition of the cost function allows to take side-chain conformations into account for the computation of the cost. Yet, it has multiple drawbacks. First, the continuity of the motion of the side-chains along the path is not guaranteed as only the best side-chain conformation is considered for the computation of the cost. Second, the computation of the optimal side-chain conformation for a specified backbone conformation is already a very hard problem in itself as the number of DOFs can be very high (higher than the number of backbone DOFs).

- A second way to define the cost function is to use a coarse-grained energy function that will not require considering the side-chain conformations. In

20	3	20	3	20	3	20	3
Ala	B	Met	B	Gly	N	Asn	L
Cys	B	Val	B	Ser	N	His	L
Leu	B	Trp	B	Thr	L	Gln	L
Ile	B	Tyr	B	Glu	L	Lys	L
Phe	B	Pro	N	Asp	L	Arg	L

Table 4.1: Correspondance of the 20 natural amino acids (columns 20) with the 3 coarse grained design features (columns 3).

that case, the energy function directly corresponds to the cost function. This approach also loses the information on the continuity of the side-chains motion along the path, but it is very computationally efficient. This is the solution that will be chosen in the following sections.

Finally, the cost function along the path defined in Equation 4.1 is a good candidate to estimate the quality of the path for a chosen sequence as this corresponds to the *mechanical work* criterion as defined in [Jaillet 2010, Devaurs 2016]. Nevertheless, as future work, it would be interesting to investigate other criteria to evaluate path quality in this context.

4.4.2 Additional simplifications

The combinatorial complexity of the problem presented in the previous section is huge. For an "easy" case of protein fragment involving 15 amino acids to design, the number of possible sequences is $20^{15} \approx 3 \times 10^{19}$. In practice, the current implementation of SDAP is unable to treat such a high-combinatorial complexity. Therefore, additional simplifications are required. This section describes a simplified representation of proteins that we have adopted in order to preform preliminary tests of SDAP on protein motion design problems.

In order to reduce the dimension of the design space, a simple coarse grained sequence model is proposed where the 20 natural amino acids are grouped according to their chemical properties to form a 3 letter alphabet. The features $f_1, \ldots, f_n$ that will be assigned to each position in the amino acid sequence are no longer the amino acid types, but the amino acid categories:

- B for the hydrophobic amino acids,
- L for the hydrophilic amino acids,
- N for the neutral amino acids.

Table 4.1 details how amino acids are divided among the different categories. This simplification reduces the cardinality of the design space to 3^n possible design.

This coarse grained sequence model is called the BLN model, and simple potential energy functions have been proposed for it. Each amino acid is considered

as a bead centered on its C_α. Consecutive beads are linked by virtual bonds. To evaluate the energy, we use the energy function developed in [Brown 2003]. It is defined by :

$$E = \sum_{\theta} \frac{1}{2} k_\theta (\theta - \theta_0)^2 + \sum_{\phi} \Bigg[A(1 + \cos\phi) + B(1 - \cos\phi) + C\,(1 + \cos 3\phi) + D\left(1 + \cos\left[\phi + \frac{\pi}{4}\right]\right)\Bigg] + \sum_{i,j \geq i+3} 4\varepsilon_H S_1 \left[\left(\frac{\sigma}{r_{ij}}\right)^{12} - S_2 \left(\frac{\sigma}{r_{ij}}\right)^{6}\right] \quad (4.7)$$

In this equation, ε_H sets the energy scale. The first term is a bond angle energy term where θ is the angle formed by two consecutive virtual bonds, k_θ is a force constant of $20\varepsilon_H/rad^2$, and θ_0 is a reference angle of 105°. The second term is a dihedral angle energy term where ϕ is the dihedral angle formed by three consecutive virtual bonds, where σ is the distance unit (σ = 1Å), and where A, B, C, and D, are parameters varying as a function of the local secondary structure (computed using DSSP [Kabsch 1983, Touw 2015]):

- $A = 0$, $B = C = D = 1.2\varepsilon_H$ for an helical structure,
- $A = 0.9\varepsilon_H$, $B = D = 0$, $C = 1.2\varepsilon_H$ for a strand structure,
- and $A = B = D = 0$, $C = 0.2\varepsilon_H$ for any other structure.

The third term is a pair-wise non bonded interaction term where r_{ij} is the distance between the i^{th} and the j^{th} C_α in the sequence, and S_1 and S_2 are parameters that depend on the designed features associated to the pair of amino acid: $S_1 = S_2 = 1$ for B-B interactions, $S_1 = \frac{1}{3}$ and $S_2 = -1$ for L-L and L-B interactions, and $S_1 = 1$ and $S_2 = 0$ for N-L, N-B, and N-N interactions.

4.4.3 Preliminary experiments

4.4.3.1 Test system

The *Escherichia coli* dihydrofolate reductase enzyme (*ec*DHFR), represented in Figure 4.6, is a protein that uses the cofactor NADPH to reduce 7,8-dihydrofolate (DHF) and form the product 5,6,7,8-tetrahydrofolate (THF). Some studies showed that the loop formed by residues 9–24 (called the Met20 loop) fluctuates between two orientations (open and occluded) at a rate comparable with the production of THF [Falzone 1994] indicating that this flexible loop is implicated in this reaction. Other studies also showed that the replacement of central residues of the Met20 loop (Met16, Glu17,Asn18, and Ala19) by glycines resulted in a 500-fold decrease in the rate of hybrid transfer [Li 1992, Osborne 2001]. In the BLN coarse grained sequence model, these mutations corresponds to a transformation of the sequence of features BLLB to NNNN. If SDAP could show that these mutations indeed decrease the mobility of the Met20 loop, this would be an encouraging result.

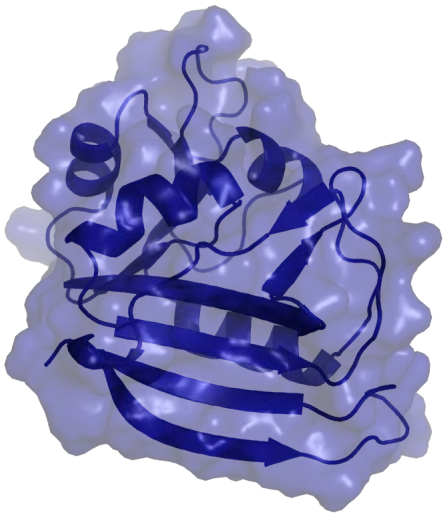


Figure 4.6: Representation of the *ec*DHFR protein.

4.4.3.2 Experiment setup

Similarly to what was explained in Section 2.1, the internal coordinates representation with the rigid geometry assumption is used to model the protein, and the double bonds corresponding to ω dihedral angles are fixed. In order to further reduce the dimension of the problem, other assumptions are necessary. First, only residues 16–19 are associated to multiple design features. The other residues are constrained to keep the design feature B, L, or N, that corresponds to their original amino acid type (see Table 4.1). This reduces the design space to a size of $3^4 = 81$ possible design. Furthermore, to reduce the size of the conformational space to explore, all the protein amino acid residues, except for the residues 8–25, are considered as rigid bodies and are fixed in the space. This reduces the number of DOFs of the system to $18 \times 2 = 36$. Note that this constraint also changes the topology of the conformational space. As the 8^{th} and 25^{th} residues are connected to the 7^{th} and 26^{th} residues, which are fixed, the 8–25 loop must be closed in order to correspond to a valid conformation. This kind of constraint can easily be satisfied within a RRT procedure using the tripeptide decomposition explained in Section 2.2. The loop is decomposed into tripeptides. At each iteration of SDAP, the random configuration q_{rand} is sampled as if there were no constraint. Then, during the *Extend* operation, a tripeptide is chosen at random inside the loop using a uniform distribution. A linear interpolation of the DOFs of the residues outside of this tripeptide is performed from q_{near} to q_{rand} in such a way that the distance between the two configurations is equal to the check step parameter δ. The distance used for the linear interpolation is the Euclidean distance in the internal coordinate

space. The parameter δ was set to 0.015 rad. Finally, IK (see Section 2.3.2) is used on the selected tripeptide to close the loop. The simulation is run at a temperature of 300 K from an initial configuration corresponding to the occluded orientation of the Met20 loop. It is stopped after it performed 50,000 iterations and the furthest configuration is recorded.

4.4.3.3 Results

Figure 4.7 shows the initial conformation and the furthest conformation found by the algorithm. Interestingly, the furthest conformation is very similar to the open orientation of the Met20 loop. Though, SDAP failed to discriminate the different designs for the motion between the open and closed states. First, all the nodes on the path between the initial and the final configuration were accepted by SDAP for all the designs in $\mathcal{D}$. In fact, in the RRT tree built by SDAP, out of the 5,768 nodes, 93% of them were labelled as valid for the 81 possible designs. Second, the *mechanical work* criterion did not allow to discriminate a particular design compared to another. Figure 4.8 (a) shows the mechanical work along the path from the initial to the furthest configuration. The curve is only plotted once because it is roughly identical for all the possible designs. When looking at the energy function described in Equation (4.7), the third term is the only one that is dependent on the design. The exact mechanical work difference between the different designs can be identified by only looking at this term, and more precisely, by only looking at the terms involving the residues 16–19. Figure 4.8 (b) shows these differences for the two target designs BLLB and NNNN. If the relative energies for the BLLB design seems much higher compared to the NNNN design, it should be put in perspective with the energy scale of the mechanical work. Furthermore, this result would indicate that the mechanical work for the BLLB design is higher than the mechanical work of the NNNN design, which would suggest that the motion if favored by the NNNN design. That is in contradiction with the experimental data.

These results clearly indicate that the chosen coarse grained energy function is not well adapted to be used with SDAP. It fails to discriminate sequences during the simulation, and the evaluation of the mechanical work along the solution path cannot differentiate the quality of the sequences. This means that a more detailed energy function should probably be used. Unfortunately, using more accurate all-atom energy function is not possible in practice with the current implementation of SDAP. SDAP does not scale to a sequence space of size 20^3 because this will imply an extremely large number of nodes in the tree that need to be stored in memory. In addition, modifications of the algorithm are required to deal with the side-chains.

4.5 Conclusions and future work

This chapter has formalized a new problem inspired by protein motion design. An algorithm, SDAP, has been proposed to solve this problem. The algorithm simul-

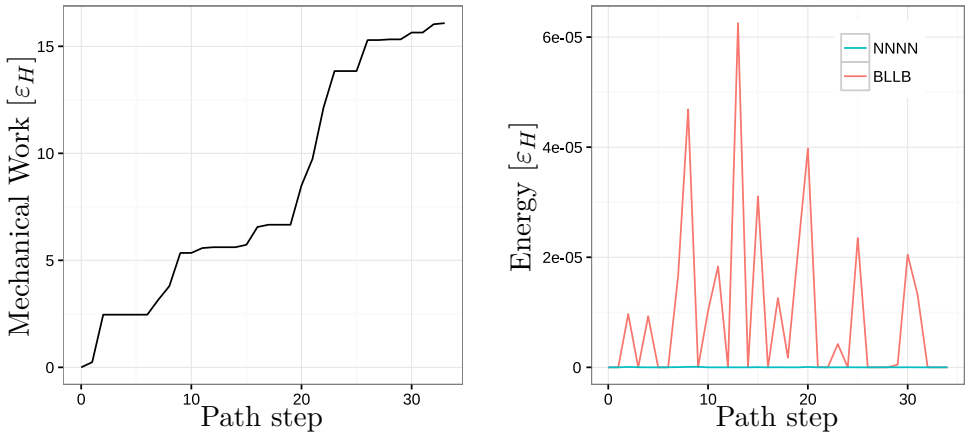


Figure 4.7: Initial conformation of the Met20 loop in green (occluded orientation). Furthest conformation of the Met20 loop after 50,000 iterations of SDAP in red (open orientation).

taneously explores the configuration space for the set of all possible designs. a comparison of SDAP with a brute force approach, in which multiple path planning problems are solved independently for each possible design, shows that the principle of SDAP allows to considerably reduce the time to solve the problem while still identifying the best design. In a second part, the application of SDAP to a simultaneous design and path planning problem involving a protein loop has been investigated. Some simplifications on the protein representation are necessary in order to reduce the number of possible design to a manageable size for SDAP. Those simplifications, based on a BLN coarse-grained model, turned out to be too strong and prevented SDAP to discriminate among the different designs regarding the objective trajectory.

Further improvements of SDAP are required to treat more complex problems, such as the ones posed by protein motion design. SDAP systematically explores the configuration space for all possible designs trying to grow at least one node for each design at each iteration. This causes two different problems. First, for each iteration, there is at least one (and most of the time more than one) energy evaluation for each possible design. Energy evaluation being an expensive operation, this strongly limits the number of possible designs that can be considered. Second, at each iteration, SDAP might add up to one node for every possible design. If the number of possible design is big, storing the exploration tree in memory will be impossible. This memory issue was found to be the most critical limitation in practice.

In order to circumvent these limitations, different approaches can be investigated. First, the exploration tree can be pruned to only store nodes identified as useful. Second, design filtering strategies have to be elaborated, using statistical learning for instance.



(a) (b)

Figure 4.8: a) Mechanical work along the path between the occluded and open positions of the Met20 loop of ecDHFR. b) Design dependent energy values along the path for the BLLB and NNNN designs.

Several applications of SDAP in robotics can also be envisioned. In addition to the design of some robot's features to optimize its motion in a given workspace, it would also be possible to apply the proposed method to optimize the workspace layout for a given robot. One can also imagine applications for helping to the design of modular self-reconfigurable robots.

Conclusion

This thesis has presented several contributions aiming at enhancing computational protein design methods. Firstly, a robotics-inspired protein modeling approach has been presented. It relies on the decomposition of protein in tripeptides, and on the application of 6R inverse kinematic. This model allowed the introduction of a simple and unified approach to design local sampling move classes. The comparison of different move classes implemented using this approach within a MC method showcases the efficiency differences between the different move classes. The overall conclusion of these results is that mixing move classes provides better results than using a single move class, as also suggested by related work on MC methods. Combining different move classes is a straightforward task following the proposed approach.

Secondly, different conformational space exploration algorithms, inspired from robotics, were presented. The T-RRT algorithm, presented in previous work, explores the space using on one side the Voronoï bias, pushing the search toward unexplored regions of the space, and on the other side a transition test with a variable temperature parameter that contains the tree of conformations within the low energy region of the space. Two other algorithms, EST_{kpiece} and EST_{ss}, are based on the principle of the EST algorithm. They rely on heuristics to decide the regions of the conformational space that must be explored in priority. The heuristic used by EST_{kpiece} relies on a grid, built in a low dimensional projection of the conformational space, to estimate the space coverage. The heuristic used by EST_{ss} relies on a score that estimates, for each node, the expected increase of space coverage that will be achieved by sampling the neighborhood of the corresponding conformation. Those two algorithms were used in combination with the T-RRT transition test with a variable temperature. The three algorithms were compared to a simple unbiased MC method. The results suggest that the combination of the transition test of T-RRT with the two EST approaches is inefficient in its current state. T-RRT showed better exploration abilities.

Finally, a new problem combining design and path planning has been formalized. It is inspired by the protein motion design problem. The algorithm SDAP, based on T-RRT, was proposed to solve this problem. SDAP performs a simultaneous exploration of the conformational space for the set of all possible design. An empirical comparison of this algorithm against a naive approach on simple academic scenarios showed the advantages of SDAP. SDAP was then applied to a protein system: the *ec*DHFR protein. The goal was to investigate the effect of some particular mutations in the protein for which experimental results are available. A coarse grained model was used to perform the test. Unfortunately, this model did not allow SDAP to obtain the expected results as the different sequences could not be filtered during the exploration.

Future work

The work presented in this thesis manuscript can be extended in several directions. A first direction to explore is the integration of the move classes developed in Chapter 2 within the EST based algorithms presented in Chapter 3. Indeed, these move classes demonstrated better performances compared to simple random perturbations of all the dihedral angles, as was performed in Chapter 3. Furthermore, the approach, based on particles perturbations, allows to define a large number of move classes using a unique model and makes it possible to mix those different move classes. The integration of those move class into an EST algorithm would permit to combine the efficiency of the mixed local move classes with the smart exploration based on the chosen EST heuristic.

Furthermore, as was mentioned in Chapter 2, the tripeptide-based approach would allow us to easily implement biased moves that deform regions of the protein in a particular direction. This property could be particularly useful to devise local moves that take into account the interaction of a protein with other molecules. This strategy could be particularly interesting for the study of protein-protein or protein-ligand interactions. It could also be useful to optimize a conformation after the introduction of a mutation in a protein sequence. Biased pertubations could also be exploited by conformational space exploration algorithms, speeding up the discovery of new regions, within the framework of the EST_{kpiece} and EST_{ss} algorithms, for instance.

Another direction to explore is to try different, more evolved, heuristics for the EST based algorithms. If those approaches showed bad exploration capabilities compared to T-RRT in Chapter 3, it is probably because the used heuristic were very simple. In the context of exploring a very high dimensional space, EST approaches might have some advantages compared to the T-RRT based algorithms. With the increase of the dimension of the search space, exploring the entire space becomes impossible. Now, although T-RRT confines the expansion of the tree to low energy regions of the space, it spends a huge amount of time trying to explore in unexplored areas of the spaces (ie. regions corresponding to the large Voronoï cells) that can be of low interest. Within an EST-based algorithm, the heuristic determining the node to extend could favor the exploration of regions using more evolved criteria than the simple space coverage. For instance, promising directions could be favored and explored intensively to the detriment of other regions of the space. In summary, the enhancement of the heuristic combined with the possibility to bias perturbations opens many possibilities to improve the exploration capabilities of EST-based algorithms in high dimensional spaces.

The principle of SDAP, which consist in simultaneously exploring the configuration space for all the possible designs, can be applied to other types of algorithms in addition to T-RRT. For instance, other RRT-based algorithms, like RRT*, or AT-RRT, which have the advantage to guarantee the optimality of the found path, could be used as a basis for SDAP. Besides, the application of EST-based algorithms could also be investigated.

The principal limitation of SDAP comes from the combinatorial explosion yielded by the size of the design space. In order to consider the use of SDAP in the context of a real protein design problem, finding a way to limit the number of designs to consider along the conformational exploration is critical. A solution to this problem would be to introduce pruning stages during the exploration, as is done in the SST* algorithm [Li 2014]. Larger design spaces will also require SDAP to employ more sophisticated filters, such as those that incorporate statistical learning, to limit the design search and control the size of the search tree.

Appendix A

Méthodes inspirées de la robotique pour l'aide à la conception de protéines

A.1 Introduction

Les protéines comptent parmi les molécules les plus essentielles à la vie. Présentes dans toutes les cellules vivantes, elles participent à une grande majorité des processus biologiques en remplissant des fonctions aussi variées que la catalyse, la régulation, le signal, le transport, le stockage, et peuvent même avoir des rôles structurels. Elles sont exploitées en pharmacologie, en biotechnologie, ou en tant que composants de nano-systèmes. Pour ces différents domaines, la capacité à créer de nouvelles protéines, ou à améliorer les protéines existantes pour remplir de nouvelles fonctions est un enjeu majeur.

Les protéines sont des chaines d'acides aminés flexibles dont la forme, déterminée par la séquence d'acides aminés, est fortement corrélée à la fonction. Ainsi, concevoir une nouvelle protéine revient à trouver la séquence d'acides aminés qui va correspondre à une structure spatiale objectif. Mais si les techniques expérimentales d'aujourd'hui rendent cette tâche possible en théorie, le nombre de séquences possibles est tellement grand que cette tâche est irréalisable en pratique. Il est nécessaire de faire appel à des méthodes computationnelles. Ces méthodes, appelées *Computational Protein Design* (CPD), permettent de guider le processus de design vers un nombre restreint de séquences candidates sur lesquels vont pouvoir être concentrées les ressources expérimentales.

Les méthodes de CPD sont développées depuis plus d'une dizaine d'années et elles ont déjà permis la création de nouvelles protéines. Les méthodes actuelles reposent sur une approche commune qui consiste à trouver, à partir d'une structure objectif, la séquence d'acide aminée qui va se replier en formant cette structure. Ce problème est formalisé sous la forme d'un problème d'optimisation. La principale difficulté réside dans la grande dimensionnalité de l'espace à explorer: c'est un espace hybride qui a une composante discrète, l'ensemble des séquences d'acides aminés possibles, et une composante continue, l'espace des configurations de la protéine. Ainsi, la résolution de ce problème d'optimisation doit reposer sur des algorithmes qui permettent d'explorer efficacement ces espaces de grandes dimensions. A cet égard, les algorithmes en provenance de la robotique ont montré des capacités prometteuses.

Cette thèse présente un ensemble de contributions s'inscrivant dans l'objectif de résoudre ce type de problèmes d'optimisation dans des espaces hybrides. Ces contributions sont faites à la fois au niveau des techniques d'échantillonnage, ainsi qu'au niveau des stratégies globales d'exploration. Le premier chapitre de cette thèse présente le problème de la conception de protéines dans son contexte. Dans un premier temps, les bases de la modélisation de protéines sont posées. Puis une vue d'ensemble des algorithmes d'exploration du paysage énergétique des protéines est présentée. Pour finir, le problème de la conception de protéine est introduit avec les approches de CPD actuelles et leurs limitations.

Le deuxième chapitre présente un modèle simple permettant d'améliorer l'échantillonnage de l'espace conformationnel des protéines au sein des algorithmes stochastiques tels que les méthodes de Monte Carlo. Ce modèle repose sur le découpage des protéines en tronçons de trois résidus d'acide aminé, ce qui permet de simplifier la conception de nouvelles classes de perturbation locales du squelette protéique.

Le troisième chapitre présente une analyse comparative de quatre algorithmes d'exploration de l'espace conformationnel des protéines: deux existants, l'algorithme T-RRT, et une simple méthode de Monte Carlo, et deux nouveaux, adaptés de l'algorithme de planification de mouvement EST en robotique.

Pour finir, le quatrième chapitre traite d'un problème d'optimisation qui combine conception et planification de mouvement. L'objectif est de trouver le design (parmi un large ensemble de possibilités) qui optimise le mouvement d'un système entre deux configurations. Cela implique de trouver le chemin optimal pour l'ensemble des designs possibles. Un algorithme pour résoudre ce problème est proposé, et ses capacités sont démontrées sur un problème académique simple. Puis la possibilité d'appliquer cette approche pour concevoir une protéine dans le but d'obtenir un mouvement précis est explorée.

A.2 Contexte scientifique

A.2.1 Séquence des protéines et structure

A.2.1.1 Acides aminés, peptides, et protéines

Les acides aminés sont les briques qui constituent les protéines. La figure 1.1 montre la composition d'un acide aminé. La chaine latérale, noté R, est spécifique à chaque acide aminé et lui donne ses propriétés physico-chimique. Il y a vingt différents types d'acides aminés, listés dans le tableau 1.1, qui correspondent à vingt chaines latérales différentes.

La réaction de condensation, décrite dans la figure 1.2, connecte deux acides aminés entre eux en formant une liaison peptidique. Cette réaction peut se répéter pour former des chaines de résidus d'acides aminés plus longues. Ces chaines s'appellent des peptides, ou polypeptides. Lorsque les acides aminés sont impliqués dans un peptide, on parle de résidus d'acide aminé. Une chaine continue de liaison cova-

lente peut être suivi depuis le premier résidu d'acide aminé jusqu'au dernier (voir figure 1.3). On appelle cette chaine le squelette protéique, ou le *backbone* de la protéine. On appelle séquence du peptide la liste des résidus d'acide aminé qui forment la chaine peptidique.

Les peptides sont des molécules flexibles. Sous l'impulsion des interactions des atomes qui composent les résidus, entre eux, ou avec les atomes du milieu extérieur (solvant, ligand, autre peptide), les peptides vont adopter des arrangements spatiaux que l'on appelle des conformations. On peut reconnaitre dans ces conformations des sections localement structurés qui correspondent à des formes stables. Ces formes sont couramment utilisées pour simplifier la représentation des conformations des peptides et permettent de reconnaitre rapidement leur structure (voir figure 1.4). Le terme de protéine est employé pour désigner un polypeptide, ou un conglomérat de polypeptides liés entre eux, qui a une fonction biologique.

A.2.1.2 Relation fonction-séquence

Pour qu'une protéine remplisse sa fonction biologique, elle doit adopter le bon arrangement spatial. On parle d'état natif de la protéine. Le processus de passage d'un état quelconque de la protéine vers son état natif s'appelle le *repliement*. Il est entièrement déterminé par les interactions des atomes qui composent la protéine et par le solvant qui l'entoure. Il est donc dépendent de la séquence de la protéine. Dans les mêmes conditions, deux protéines ayant la même séquence vont, de manière générale, se replier dans la même structure biologiquement active. Beaucoup de petites protéines, lorsqu'elles sont dénaturées, vont tout de même se replier dans leur structure native, biologiquement active [Anfinsen 1972], et on suspecte beaucoup de maladies d'être causées par des mutations de protéines entrainant un mauvais repliement dans un arrangement spatial non fonctionnel [Neudecker 2012, Soto C 2008].

A.2.2 Modélisation des protéines

A.2.2.1 Coordonnées cartésiennes

La manière la plus directe de représenter la configuration d'une protéine est la représentation en coordonnées cartésiennes. Il s'agit, pour une protéine composée de N atomes, de lister les coordonnées cartésiennes de chaque atome dans l'espace. Cette représentation est sans doute la plus utilisée. Le calcul de l'énergie potentielle d'un système moléculaire repose généralement sur ce type de représentation. De plus, c'est le format qui est utilisé par la Protein Data Bank (PDB).

Mais cette représentation a des inconvénients. D'une part, elle nécessite un vecteur de dimension $3N$ pour décrire la configuration de la protéine. D'autre part, les contraintes de positions relatives entre les différents atomes liés ne sont pas représentées. En conséquence, cette représentation est peu efficace dans le cadre d'algorithmes d'exploration de l'espace des configurations.

A.2.2.2 Coordonnées internes

La représentation en coordonnées internes repose sur la connaissance des liaisons atomiques de la protéine. La configuration de la protéine est décrite uniquement grâce aux longueurs de liaisons, aux angles de liaisons, et aux angles de torsions des liaisons (voir Figure 1.5). Une simplification courante, appelée l'hypothèse de géométrie rigide, consiste à considérer que les longueurs et les angles de liaisons sont constants (leurs fluctuations sont de faibles amplitudes) [Scott 1966, Engh 1991]. De cette manière, la configuration de la protéine est entièrement décrite par la liste de ses angles de torsion.

A.2.2.3 Réduction de dimensions complémentaires

Si la représentation en coordonnées interne combinée à l'hypothèse de géométrie rigide permet de réduire drastiquement le nombre de degrés de libertés nécessaires pour décrire la configuration d'une protéine, il est néanmoins courant d'atteindre les 1000 angles de torsion. La section 1.2.3 indique différentes approches pour réduire encore la dimension de l'espace des configurations d'une protéine en utilisant par exemple des connaissances sur le système étudié [Jones 1997, Apostolakis 1998, Pak 2000, Thomas 2013] ou grâce à des méthodes mathématiques (PCA [Fodor 2002], IsoMap [Van Der Maaten 2009], LSDMap [Tenenbaum 2000], NMA [Cui 2005]).

A.2.2.4 Modèles gros grains

Contrairement aux représentations tout-atomes mentionnées précédemment, les modèles gros grains sacrifient les détails structurels en ne décrivant que la position d'un nombre restreint d'atomes. Ces représentations transforment à la fois l'espace à explorer, et la fonction d'énergie potentielle sous-jacente. Elles permettent de réduire la dimension de l'espace des configurations à explorer, et d'accélérer les calculs énergétiques au coût d'une perte de précision pouvant parfois mener à des résultats irréalistes.

Les modèles gros grains existants peuvent représenter plusieurs niveaux de détails. Les modèles les plus simples représentent un seul atome par résidu alors que certains modèles plus complexes utilisent jusqu'à six atomes, parfois virtuels, par résidu (voir section 1.2.4).

A.2.3 Paysage énergétique

A.2.3.1 Théorie physique

Les lois du mouvement décrites par Newton permettent en théorie de prédire la dynamique d'une protéine grâce à son état initial (position et vitesses). Ces lois font notamment intervenir le calcul de l'énergie potentielle. Lorsqu'on regarde l'énergie potentielle comme une altitude, la fonction d'énergie potentielle dessine une hypersurface au-dessus de l'espace des configurations de la protéine. On appelle cette

hypersurface le paysage énergétique [Wales 2004]. Les minima locaux de cette hypersurface correspondent aux configurations théoriquement stables de la protéine. En pratique, le système, en fluctuation constante, va passer d'un minima local à un autre. Les bassins de basse énergie entourés par des régions à haute énergie correspondent aux conformations stables du système. Pour la plupart des protéines, il y a une conformation stable unique correspondant à la conformation native (voir figure 1.6(a)). On parle de paysage énergétique en entonnoir. Certaines protéines peuvent avoir plusieurs bassins énergétiques correspondant à plusieurs états stables, avec éventuellement des chemins de transition à basse énergie permettant la transition du système d'un état à l'autre (voir figure 1.6(b)).

A.2.3.2 Fonctions d'énergie

Obtenir une bonne approximation de l'énergie potentielle n'est pas trivial. Des efforts importants sont faits par les chercheurs pour produire des fonctions d'énergie de plus en plus précises. La section 1.3.2 introduit les fonctions d'énergies potentielles les plus couramment utilisées, comme AMBER [Kollman 1997] ou CHARMM [Brooks 2009].

A.2.4 Méthodes d'exploration de l'espace des conformations des protéines

A.2.4.1 Dynamique moléculaire

La dynamique moléculaire (MD) consiste à simuler la dynamique d'un système moléculaire afin d'obtenir une approximation de sa trajectoire. La dynamique moléculaire, qui repose sur une représentation détaillée de la protéine et sur une fonction d'énergie potentielle, consister à résoudre les lois du mouvement de Newton de manière itérative, avec un pas très faible (voir section 1.4.1). Si la dynamique moléculaire permet d'obtenir des informations très précises, permettant des calculs de physique statistique, elle est néanmoins limitée à la simulation de trajectoires d'une durée d'au plus quelques microsecondes. En pratique, les réactions faisant intervenir des protéines durent quelques millisecondes, pour les plus rapides, et jusqu'à plusieurs secondes pour les plus lentes.

De nombreuses adaptations de la dynamique moléculaire existent pour améliorer les capacités d'échantillonnage de cette technique. On peut citer *replica exchange MD* [Sugita 1999], *steered MD* [Suan Li 2012], ou encore la metadynamique [Laio 2002].

A.2.4.2 Méthodes de Monte Carlo

La méthode de Monte Carlo (MC), décrite dans [Metropolis 1953], est un algorithme stochastique très utilisé dans l'étude des protéines. Elle construit une séquence de configuration $C_1, \ldots, C_n$. A chaque itération, la dernière configuration C_t est perturbée. La nouvelle configuration obtenue $C_{candidate}$ est acceptée avec une prob-

abilité P, décrite par (1.1) dans la section 1.4.2. Ce test s'appelle le Metropolis Criterion. Le paramètre T, pour température, permet de contrôler l'agressivité de l'exploration.

Contrairement à la dynamique moléculaire, les perturbations effectuées à chaque étape sont aléatoires et elles n'ont même pas besoin d'être réalistes du moment qu'elles sont couplés au Metropolis Criterion. Cependant, le choix des perturbations impacte fortement l'efficacité de l'exploration. Ce sujet est traité plus en détail dans le chapitre 2.

Si la méthode de MC a de meilleures capacités d'échantillonnage que la MD, elle ne permet pas d'obtenir une trajectoire réaliste. En revanche, sous certaines conditions (voir section 1.4.2), la distribution des configurations obtenue permet de calculer les propriétés statistiques du système comme l'énergie libre.

Bien que la méthode de MC explore plus efficacement que la MD, les processus de repliement de protéine, ou de transition entre deux conformations sont toujours difficiles à observer. Des méthodes dérivées permettent d'obtenir de meilleurs résultats. On peut citer *replica exchange MC* [Earl 2005], *umbrella sampling* [Torrie 1977], *energy landscape flattening* [Zhang 2002], *basin hopping* [Wales 1997], ou encore *simulated annealing* [Kirkpatrick 1983].

A.2.4.3 Méthodes inspirées de la robotique pour l'exploration de l'espace des conformations

La planification de mouvement, en robotique, est un problème qui fait l'objet d'études intensives depuis près de quarante ans [Latombe 1991, Choset 2005]. Le but de ce problème est de trouver le mouvement pour amener le robot à passer d'une configuration initiale à une configuration finale. L'exploration de l'espace des conformations des protéines est un problème très similaire. En effet, un parallèle peut être fait entre l'espace des conformations des protéines, et l'espace des configurations d'un robot. Et la représentation des protéines en coordonnées interne est très proche de la manière dont sont représentées les chaines articulées en robotique. Ces similarités ont été exploitées dès les années 90 [Parsons 1994] et de nombreuses adaptations des algorithmes utilisés en robotiques ont été faites pour étudier les protéines [Moll 2008, Al-Bluwi 2012]. Ces algorithmes sont majoritairement des variantes de trois algorithmes : Probabilistic Roadmap (PRM), Rapidly-exploring Random Trees (RRT), et Expansive-Spaces Trees (EST).

PRM L'algorithme PRM [Kavraki 1996], fonctionne en deux phases distinctes : la construction de la roadmap, et la phase de requête. La construction de la roadmap, détaillées dans la section 1.4.3.1, construit un graphe de configuration dont les arrêtes représentent la validité de la transition (ou du mouvement) entre deux configurations. Un exemple de roadmap est représenté dans la figure 1.7. En robotique, la vérification de collisions et la planification locale permettent généralement de juger de la validité d'une configuration et de la transition entre deux configurations. Les adaptations de cet algorithme pour l'étude des protéines utilisent

quant à elles l'énergie potentielle et le Metropolis Criterion. La phase de requête consiste à utiliser le graphe construit durant la première phase pour trouver un mouvement entre une configuration initiale et une configuration finale. Pour cela, des algorithmes tels que Dijkstra's shortest path [Dijkstra 1959], ou A* [Hart 1968] sont utilisés.

RRT L'algorithme RRT est basé sur la construction d'un arbre [Lavalle 1998, LaValle 2001]. A partir d'une configuration initiale, l'algorithme itère pour construire un arbre de configurations jusqu'à ce que la configuration finale puisse être connectée à l'arbre. La stratégie de RRT repose sur un biais de Voronoï implicite qui pousse l'exploration vers les régions inexplorées de l'espace [Lindemann 2004]. L'algorithme est décrit rapidement dans la section 1.4.3.2 (voir figure 1.8), puis plus en détail dans le chapitre 3. De nombreuses variantes de RRT ont été créées, dont certaines utilisées en biologies structurelles: ML-RRT [Cortés 2008, Cortés 2010b], et T-RRT [Jaillet 2008, Devaurs 2013b].

EST L'algorithme EST repose également sur la construction d'un arbre de configurations [Hsu 1997, Hsu 2000, Hsu 2002]. A chaque itération, une configuration q est choisie avec une probabilité $P(q)$, puis une configuration q_{rand} est tirée aléatoirement dans le voisinage de q (voir section 1.4.3.1 et figure 1.7). La fonction de probabilité P est un paramètre de l'algorithme, ce qui fait d'EST un algorithme très général. De plus, si q_{rand} est construit à partir d'une distribution uniforme dans la version originale de l'algorithme, cette étape peu facilement être biaisée. Cela est particulièrement intéressant dans le cadre de l'exploration de l'espace à haute dimensions des conformations d'une protéine. Cette approche est utilisée dans le chapitre 3. L'algorithme EST a également quelques variantes. On peut citer notamment KPIECE [Sucan 2012].

A.2.5 Computational Protein Design

La conception de protéine est le processus qui consiste à trouver une séquence d'acides aminés telle que la protéine correspondante soit capable de remplir une certaine fonction. Pour une protéine de taille N, il y a 20^N séquences possibles. Il est impossible de toutes les tester de manières expérimentales. Les méthodes de Computational Protein Design (CPD) ont été développées pour identifier les séquences candidates les plus prometteuses et optimiser le processus de recherche. De gros progrès ont été faits durant les dernières décennies qui ont menés à quelques succès (voir section 1.5).

A.2.5.1 Problème de la conception de protéine

Étant donné que les interactions d'une protéine avec son environnement sont principalement déterminées par son arrangement spatial (voir section 1.1.2), le problème de la conception de protéine peut être formulé comme suit : trouver la séquence

d'acides aminés qui va se replier dans la structure spatiale désirée. La résolution de ce problème se déroule généralement en quatre phases (décrites plus en détail dans la section 1.5.1).

1. La formulation du problème : définition de la structure spatiale, des contraintes, etc...
2. La recherche *in silico* de la séquence qui va se replier dans cette structure spatiale : Les techniques de CPD actuelles traitent une version simplifiée de ce problème où l'on cherche à trouver les séquences qui stabilisent au mieux la structure objectif, sans garantie que le repliement est bien possible.
3. L'analyse *in silico* des résultats : simulations (grâce à la dynamique moléculaire ou aux méthodes de MC) du comportement de la protéine candidate.
4. La validation expérimentale : permet de valider le bon repliement et la bonne activité de la protéine.

Dans la suite de ce travail, le focus est mis sur la seconde phase de ce processus.

A.2.5.2 L'espace de recherche

L'espace à explorer durant la seconde phase du processus de CPD est le produit de deux composantes hétérogènes :

- Une composante discrète qui correspond à l'ensemble des séquences d'acides aminés possibles. Le cardinal de cet espace est de 20^N pour une protéine de longueur N.
- Une composante continue qui correspond à l'espace des conformations correspondant à chaque séquence. La dimension de cet espace correspond au nombre de degrés de liberté de la protéine dans la représentation choisie.

Une partie importante du processus de CPD consiste à réduire la dimension de l'espace de recherche pour le rendre accessible aux méthodes de résolution actuelles. Les simplifications les plus simples consistent à réduire le problème initial en limitant le nombre de positions mutables dans la séquence d'acide aminé, ou en limitant la flexibilité de certaines portions de la protéine. D'autres simplifications doivent généralement être faites. La première est de considérer, pour chaque type d'acide aminé, un nombre restreint de positions de la chaine latérale (on parle de librairie de rotamers). Une deuxième simplification est de considérer que le squelette protéique est rigide [Ponder 1987]. Ainsi, le problème de CPD est réduit à la recherche d'un ensemble de rotamers qui optimise une fonction objectif portant sur la structure spatiale souhaitée.

Bien que ces simplifications aient permis de faire d'énormes progrès dans le domaine du CPD, elles traitent néanmoins un problème éloigné de la réalité. La flexibilité des chaines latérales et du backbone sont primordiales pour juger de la stabilité d'une conformation. Certaines méthodes ont été développées pour prendre en compte, de manière limitée, cette flexibilité [Fung 2008, Kuhlman 2003].

A.2.5.3 Les méthodes de CPD actuelles

Le problème de CPD, même dans sa version la plus simple, est extrêmement complexe (NP-hard [Pierce 2002]). On peut citer deux catégories principales de méthodes de résolution.

Les algorithmes déterministes Le principal algorithme utilisé dans cette catégorie est l'algorithme Dead-End Elimination (DEE) [Desmet 1992] qui réduit progressivement le nombre de rotamers possibles à un ensemble beaucoup plus réduit, et accessible à des algorithmes tel que A* [Hart 1968]. L'algorithme DEE a été amélioré au cours des années pour obtenir des versions toujours plus efficaces [Goldstein 1994, Pierce 2000, Georgiev 2006, Georgiev 2008]. Un second algorithme dans cette catégorie repose sur les Cost Function Network (CFN) et a montré des améliorations significatives comparé au couple d'algorithmes DEE/A* [Allouche 2012, Traoré 2013, Allouche 2014].

Les algorithmes stochastiques Les algorithmes stochastiques, contrairement aux algorithmes déterministes, ne peuvent pas garantir que le résultat trouvé est optimal. Ils offrent néanmoins l'avantage de proposer des solutions candidates en un temps relativement court [Voigt 2000]. L'algorithme le plus utilisé en CPD se base sur les méthodes de MC pour lesquelles le Metropolis Criterion est également utilisé pour valider des changements de séquence [Polydorides 2011]. Mais d'autres algorithmes sont également utilisés. On peut notamment citer des techniques reposant sur les algorithmes génétiques [Weise 2009], ou l'algorithme FASTER, qui combine les approches stochastiques et déterministe [Desmet 2002, Allen 2006].

A.2.5.4 Les challenges du CPD

L'un des principaux challenges du CPD est la réalisation de design multi-objectifs. En effet, de nombreuses protéines font l'objet de changements conformationels lorsqu'elles interagissent avec leur environnement (lors d'une liaison avec un ligand par exemple). Dans ce contexte, il s'agit de construire une protéine stable dans les deux conformations (liée, et non liée). Les méthodes actuelles, dites multi-états, utilisent la fonction objectif pour traduire la stabilité des deux conformations. Elles ne sont pas capables, néanmoins, de garantir que le changement de conformation est possible, c'est à dire qu'il existe une transition entre ces deux conformations qui ne franchisse pas de barrière hautement énergétique dans le paysage énergétique de la protéine.

Ces limitations viennent de la formulation actuelle du problème de CPD qui se base sur la minimisation de la fonction objectif, elle-même basée sur des structures rigides. Les méthodes actuelles cherchent à concevoir la protéine en se basant sur des états statiques, alors que la dynamique de la protéine est fortement impliquée dans de nombreuses fonctions (liaison avec un ligand, avec une autre protéine, libération

d'un produit, transition allostérique). Une procédure de design prenant en compte la dynamique des protéines ouvrirait de nouvelles possibilités pour le CPD.

A.3 Modélisation des protéines et échantillonnage local des conformations

Ce chapitre présente une approche pour améliorer l'échantillonnage locale des configurations au sein des méthodes d'exploration de l'espace des conformations. Elle est basée sur une représentation mécanique des protéines. L'idée générale est de découper la protéine en fragments de trois résidus d'acide aminé, que l'on appelle *tripeptides*. Chaque fragment peut être représenté comme une chaine cinématique, similaire à un bras manipulateur en robotique. Une telle représentation permet de concevoir des méthodes efficaces pour déformer localement la protéine, tout en préservant la géométrie de ses liaisons atomiques, grâce à l'utilisation d'un *solver* de cinématique inverse. Bien que ce chapitre porte principalement sur une application particulière de cette approche pour la conception de classes de mouvement au sein d'une méthode de Monte Carlo, le modèle de découpage en tripeptide peut être exploité dans d'autres contextes.

Une des difficultés principales rencontrée pour appliquer les méthodes de MC aux protéines réside dans la création de classes de mouvement appropriées pour les molécules chaînes. Comme indiqué dans le chapitre 1, une classe de mouvement efficace doit à la fois avoir un bon taux d'acceptation, mais également permettre l'exploration de larges régions de l'espace des conformations. Différents types de classes de mouvement ont été proposées pour améliorer l'efficacité des méthodes de MC appliquées aux protéines. L'approche présentée ici permet de construire un ensemble de types de classe de mouvement qui peuvent être implémentées très simplement au sein d'une seule et unique représentation moléculaire, et d'un seul et unique *solver* de cinématique inverse.

Ce chapitre présente tout d'abord les principes généraux de la représentation mécanique de la protéine et de la décomposition en tripeptides. Puis, il explique comment implémenter différents types de classes de mouvement en se basant sur cette représentation. Les performances de ces classes de mouvement sont ensuite analysées à travers des tests portant sur des protéines de différents types. Ce chapitre est une extension d'un travail préliminaire [Cortés 2012]. Une nouvelle classe de mouvement (la classe *Hinge*) a été implémentée et l'analyse des différentes classes de mouvement a été poussée plus en profondeur avec l'ajout de métriques plus quantitative, comme la fonction RMSD dépendante du temps, ou l'autocorrélation.

A.3.1 Modèle mécanique

Cette section présente en détail le modèle mécanique permettant de représenter une protéine. Ce modèle se base sur la représentation en coordonnées internes,

avec l'hypothèse de géométrie rigide. De plus, les liaisons covalentes doubles sont considérées comme rigides, du fait de leur faible mobilité. Ainsi, les angles de torsions associées aux liaisons peptidiques sont considérées comme rigides réduisant le nombre de degrés de liberté du squelette protéique à deux par résidu (angles ϕ et ψ, voir figure 2.1).

A.3.2 Décomposition en tripeptide

Cette section présente l'idée principale de l'approche décrite dans ce chapitre. Elle consiste à découper la protéine en fragments de trois résidus d'acide aminé, qui sont appelés *tripeptides*. Ces sections ont la particularité d'avoir six degrés de liberté au niveau du backbone de la protéine, ce qui permet d'avoir une liberté complète en position et en orientation entre la base d'un tripeptide et son extrémité. De plus, en connaissant la position et l'orientation de la base et de l'extrémité d'un tripeptide, un *solver* de cinématique inverse 6R permet de calculer les valeurs des angles de torsion correspondants. Un repère orienté est ensuite associé à la base de chaque tripeptide. Les tripeptides étant liés entre eux par les liaisons peptidiques, qui sont rigides, la position et l'orientation de l'extrémité finale d'un tripeptide peut être retrouvée à partir du repère associé à la base du tripeptide suivant. En conséquence, la conformation du squelette protéique dans son ensemble peut être déterminée à partir de la pose (position et orientation) du repère associé à la base de chaque tripeptide. Ces repères orientés sont désignés dans la suite de ce travail sous le terme de *particules*.

A.3.3 Création de classes de mouvement

Grâce à cette représentation, des classes de mouvement qui perturbent uniquement une portion de la protéine peuvent être créées. En perturbant une ou plusieurs particules consécutives, et en utilisant le *solver* de cinématique inverse pour calculer les angles de torsion correspondant à la nouvelle configuration, on crée très simplement une perturbation locale. La création de classes de mouvement découle uniquement de la stratégie de perturbation des particules : une ou plusieurs, avec un mouvement coordonné ou non.

Trois exemples de classes de mouvement génériques sont présentés :

- La perturbation d'une seule particule (mouvement nommé plus tard *OneParticle*)
- La perturbation de plusieurs particules consécutives.
- La perturbation de plusieurs particules consécutives, mais en effectuant un déplacement coordonné, comme si ces particules formaient un bloc rigide qui tourne autour d'une charnière composée de deux tripeptides (mouvement nommé plus tard *Hinge*).

A.3.4 Résultats

Cette section décrit les résultats obtenus en appliquant cette approche à une méthode de Monte Carlo. Quatre classes de mouvement ont été implémentées en utilisant l'approche précédente, toutes basées sur le même modèle mécanique :

- la classe *OneParticle*,
- la classe *Hinge*,
- une classe de mouvement inspirée de la classe *ConRot* [Dodd 1993],
- et une classes de mouvement très simple, et très utilisée, *OneTorsion*, qui n'est pas locale et qui n'utilise pas la cinématique inverse (mais qui peut être implémentée en utilisant le même modèle mécanique).

Ces quatre classes sont évaluées au sein d'une méthode de MC, de manière individuelles, et avec une combinaison des quatre classes entre elles à travers l'introduction de la classe de mouvement *Mixed*. Deux protéines sont utilisées pour faire cette étude comparative : le domaine SH3 de l'*obscurin* (globulaire), et la protéine Sic1 (désordonnée).

Les performances des différentes classes de mouvement sont analysées grâce à différents indicateurs :

- Le premier indicateur, le temps de calcul, indique que les classes de mouvement faisant bouger le moins d'atomes sont plus rapides que les classes de mouvement faisant bouger un nombre d'atome élevé. Cela signifie que le coût de la cinématique inverse est compensé par les bénéfices liés à la localité du mouvement (réduction des temps de calculs de l'énergie, notamment).
- Le second indicateur, la distribution des conformations sur un graphique présentant l'énergie par rapport à la distance RMSD à l'état initial, indique qu'en fonction du type de protéine (ordonnée ou désordonnée), les classes de mouvements ont des performances d'exploration différente. Il ressort en revanche que la classe de mouvement combinée *Mixed* explore plus efficacement que les quatre autres.
- Le troisième indicateur, la fonction RMSD dépendante du temps, permet d'avoir une mesure d'efficacité de l'exploration à court terme. Il indique que les classes de mouvement *ConRot* et *OneParticle* sont plus efficaces que les classes *Hinge* et *OneTorsion*, et ce pour les deux types de protéine. Il indique aussi quand la classe *Mixed* est plus efficace que toutes les autres classes de mouvement prises individuellement.
- Le dernier indicateur, l'autocorrélation, est appliqué à un système plus petit et permet de mesurer la qualité de l'exploration d'un point de vue statistique. Seul trois classes de mouvement (*ConRot*, *OneParticle*, et une version de *Mixed* faisant intervenir une combinaison de ces deux classes) ont pu être

étudiées étant donnée la faible taille du système de test. Cet indicateur montre que la classe de mouvement *Mixed* est encore une fois plus performante que les autres classes de mouvement prises individuellement.

A.4 Exploration du paysage énergétique des protéines

L'efficacité de l'exploration du paysage énergétique d'un système moléculaire n'est pas uniquement une question de classe de mouvement utilisée pour déformer la structure. La stratégie globale utilisée pour échantillonner l'espace des conformations est également décisive. Et si la dynamique moléculaire et les méthodes de MC ont des propriétés très utiles permettant de faire une analyse statistique du paysage énergétique d'une protéine, comme indiqué dans la section 1.4, ce ne sont pas les méthodes les plus efficaces pour découvrir rapidement les chemins de transition possibles entre différentes conformations. A cet égard, des algorithmes inspirés du monde de la robotique ont montré leur grande efficacité. Dans ce chapitre, différents algorithmes sont comparés pour l'exploration de petits peptides très flexibles :

- une méthode de MC simple,
- l'algorithme Transition-based Rapidly-exploring Random Tree (T-RRT),
- une adaptation de l'algorithme Expansive-Search Trees (EST) qui utilise une approche similaire à l'algorithme KPIECE [Sucan 2012] pour la sélection des nœuds,
- et une autre adaptation de l'algorithme EST qui utilise une fonction d'évaluation basée sur le taux d'acceptation des nœuds pour guider l'exploration.

Les deux adaptations d'EST utilisent un test de transition similaire à celui qui est fait par l'algorithme T-RRT pour décider d'accepter ou de rejeter une nouvelle configuration. Ces algorithmes Transition-based EST sont de nouvelles approches développées et analysées dans le cadre de cette thèse. Comme nous le verrons dans ce chapitre, ces approches nécessitent des améliorations avant de pouvoir être considérées comme des alternatives viables à l'algorithme T-RRT.

A.4.1 Le dilemme exploration-exploitation

Cette section présente le dilemme exploration-exploitation auquel on doit faire face lorsqu'on cherche à explorer un espace à très haute dimension. Dans un espace à très haute dimension, comme l'espace conformationel des protéines, explorer de manière exhaustive l'ensemble des configurations est impossible en pratique. Il faut donc faire des choix et privilégier l'exploration de régions spécifiques de l'espace. Les stratégies d'exploration doivent donc décider des régions d'intérêts, et exploiter les ressources au maximum dans ces régions. Il est cependant intéressant de garder une certaine quantité de ressource à l'exploration des autres régions de l'espace,

qui pourraient se révéler à long terme plus intéressantes que les régions identifiées initialement. L'équilibrage des ressources allouées à l'exploration de nouvelles régions et à l'exploitation des régions déjà fortement explorées est ce qu'on appelle le dilemme exploration-exploitation.

A.4.2 Algorithmes

A.4.2.1 T-RRT

Cette section présente l'algorithme T-RRT (voir algorithme 3.1). Cet algorithme, basé sur RRT, utilise le Metropolis Criterion pour décider d'accepter la transition d'une configuration à une autre (voir algorithme 3.2 pour voir le détail du test de transition). La particularité de T-RRT est d'utiliser une température T variable dans le Metropolis Criterion et d'adapter cette température automatiquement au cours de la recherche pour favoriser l'exploration des régions de basse énergie en priorité, mais pour permettre le franchissement de barrières énergétiques afin de faire avancer l'exploration vers de nouvelles régions.

A.4.2.2 Transition-based EST

L'algorithme T-RRT passe un temps considérable à tenter d'explorer l'ensemble de l'espace des configurations. En effet, le biais de Voronoï qui guide sa recherche ne va pas favoriser une région par rapport à une autre. Seule le Metropolis Criterion va permettre de restreindre l'arbre d'exploration aux régions de basse énergie. Mais lorsqu'on cherche à découvrir une transition entre deux conformations, il peut être intéressant de biaiser l'exploration dans une direction particulière. Les algorithmes basés sur EST, grâce à l'introduction d'une heuristique lors du choix de la configuration à perturber, permettent d'introduire ce biais. Dans les sections qui suivent, deux heuristiques différentes sont introduites. Toutes deux sont appliquées à une adaptation d'EST, nommée Transition-based EST (voir algorithme 3.3), qui utilise le même test de transition que T-RRT, avec une température variable, pour accepter ou rejeter les nouvelles configurations.

Heuristique *success score* La section 3.2.2.1 présente les détails de l'heuristique *success score* qui donne lieu à l'algorithme EST_{ss}.

Heuristique *KPIECE like* La section 3.2.2.2 présente l'heuristique *KPIECE like* qui donne lieu à l'algorithme EST_{kpiece}.

A.4.3 Analyse comparative empirique

Dans cette section, l'algorithme T-RRT, et les deux algorithmes EST_{ss} et EST_{kpiece} sont appliqués pour l'exploration de l'espace des conformations de deux petits peptides. Leurs capacités à découvrir des chemins de transitions entre plusieurs états sont comparées à une simple méthode de MC.

A.4.3.1 Systèmes moléculaires

Cette section présente les deux peptides qui seront utilisés pour faire l'analyse comparative. Il s'agit de la *met-enkephalin*, un pentapeptide, et de la *chignolin*, un décapeptide.

A.4.3.2 Méthode expérimentale

Cette section présente la méthode expérimentale utilisée pour effectuer l'analyse comparative, avec les durées des simulations et les paramétrages des algorithmes.

A.4.4 Résultats

Cette section présente les résultats obtenus et les compare de manière empirique. Il en ressort que l'algorithme T-RRT a de bien meilleures capacités d'exploration que les autres algorithmes présentés en terme de couverture de l'espace des configurations. Le mécanisme de réglage automatique de la température ne semble pas parfaitement adapter pour fonctionner avec les deux heuristiques *success score* et *KPIECE like* présentés ici. Néanmoins, il serait intéressant de tenter l'utilisation d'autres heuristiques plus évoluées que celles utilisées dans ce chapitre avant d'abandonner l'utilisation de l'algorithme Transition-based EST.

A.5 Vers la conception de mouvements de protéine

La conception de système et la planification de mouvement sont deux problèmes qui sont généralement abordés de manière indépendante. En robotique, des critères tels que l'espace de travail, la charge de travail, la précision, la robustesse, la raideur, ou d'autres indicateurs de performance sont traités au cours de la conception d'un système [Gosselin 1991, Merlet 2005]. Les problèmes de planification de mouvement, en revanche, sont appliqués à des systèmes dont la géométrie et les propriétés cinématiques sont fixées. Dans ce chapitre, nous proposons une extension du problème de planification de mouvement dans laquelle certaines caractéristiques du système mobile ne sont pas fixées *a priori*. Le but est de trouver le meilleur design (valeurs pour les caractéristiques non définies) afin d'optimiser le mouvement entre deux configurations données.

Une approche brute-force pour résoudre ce problème consisterait à résoudre de manière individuelle des problèmes de planification de mouvement pour chaque design possible, puis de sélectionner le design qui produit le meilleur résultat par rapport à la fonction objectif (fonction du chemin). Cependant, à cause de l'explosion combinatoire, cette approche naïve permet uniquement de résoudre des problèmes faisant intervenir un très faible nombre de caractéristiques variables. Dans ce chapitre, nous proposons une approche plus sophistiquée qui traite de manière simultanée la conception du système, et la planification d'un chemin. Un problème proche, appelé généralement *kinematic synthesis* [McCarthy 2001], consiste à optimiser les paramètres géométriques et cinématiques d'un robot pour effectuer

une trajectoire donnée. Néanmoins, le problème traité dans ce manuscrit est significativement différent, puisqu'il considère que l'ensemble des caractéristiques cinématiques et géométriques possibles du système mobile sont fournies en entrée du problème. Le design concerne un ensemble discret de caractéristiques qui peuvent être associés aux différentes parties du système mobile, telle qu'une forme ou une charge électrostatique, dans le but de trouver le meilleur chemin possible entre deux configuration selon une fonction de coût donnée. Très peu de travaux ont considéré un tel problème qui combine design et planification de mouvement. Un des rares exemples est la méthode récemment proposée pour la planification de mouvement de drone [Rudnick-Cohen 2015] dans laquelle l'algorithme de planification de mouvement optimal considère plusieurs vitesses de mouvement et plusieurs surfaces d'aile de référence pour minimiser le temps et le risque le long du mouvement. Dans ce cas, l'espace des configurations considéré est de dimension deux, et la solution proposée est basée sur une extension de l'algorithme de Dijkstra travaillant une représentation discrète de l'espace de recherche. Ce type d'approche ne peut pas être appliqué en pratique pour des problèmes en plus haute dimension, comme celui que nous adressons ici.

Les algorithmes basés sur l'échantillonnage stochastique ont été développés depuis les années 90 pour la planification de mouvement dans des espaces à haute dimension [Kavraki 1996, LaValle 2006] qui sont hors d'atteinte pour les algorithmes complets et déterministes. Notre travail se base sur cette famille d'algorithme, que nous étendons pour traiter une composante combinatoire dans l'espace de recherche, associée aux designs possibles du système, tout en cherchant un chemin solution. Notre approche a des similarités avec les méthodes qui cherchent à résoudre des problèmes de manipulation [Siméon 2004], ou les problèmes de planification de mouvement multi-modale [Hauser 2010], pour lesquelles l'espace de recherche est également un espace hybride. Tout comme dans ces méthodes, l'algorithme proposé explore simultanément plusieurs sous-espaces dans le but de trouver une solution plus rapidement. Cependant, le problème de design traité ici est complètement différent.

Ce chapitre présente une approche sophistiquée, l'algorithme *Simultaneous Design And Path-planning* (SDAP), qui est basé sur l'algorithme T-RRT [Jaillet 2010]. Comme expliqué dans la section 4.1.2, le choix d'utiliser T-RRT comme base de travail est lié au type de fonction de coût qui est utilisée pour évaluer la qualité d'un chemin. Cependant, d'autres algorithmes basés sur l'échantillonnage stochastique, comme par exemple les algorithmes basés sur EST présentés dans le chapitre 3, peuvent être étendus en suivant la même approche.

Les bonnes performances de SDAP sont évaluées sur des problèmes académiques relativement simples. Ces exemples simples permettent d'appliquer la méthode naïve de recherche exhaustive dont les résultats seront utilisés comme référence pour évaluer les performances et la qualité des solutions proposées par SDAP. Les résultats obtenus montrent que SDAP permet d'identifier les meilleurs pairs chemin-design en un temps largement réduit par rapport à l'approche naïve. Cet avantage augmente avec la dimension du problème.

Bien que le problème traité dans ce chapitre soit formulé de manière abstraite, comme une extension du problème de planification de mouvement, standard en robotique, le but dans un futur proche est de considérer des problèmes de conception de protéine (ou de fragments de protéine) pour un mouvement donné. La section 4.4 explique de quelle manière ces deux problèmes sont similaires et montre comment SDAP peut être utilisé pour résoudre un problème de CPD. Au-delà du CPD, d'autres applications de l'approche proposée peuvent être imaginées en robotique (voir section 4.5).

A.5.1 Définition du problème et approche

A.5.1.1 Définition du problème

Cette section définit de manière formelle le problème combinant conception de système et planification de mouvement. Elle définit notamment l'espace des designs $\mathcal{D}$, et la fonction de coût d'un chemin $c_P(P, d)$, correspondant au travail mécanique. Le problème consiste à trouver la paire (P, d), où P est un chemin solution du problème de planification de mouvement, et où $d \in \mathcal{D}$ est un design du système, qui minimise la fonction de coût c_P.

A.5.1.2 Approche

Cette section décrit l'approche adoptée pour résoudre le problème. Elle consiste à explorer l'espace des configurations simultanément pour l'ensemble des designs possibles. L'algorithme d'exploration de l'espace des configurations se base sur l'algorithme T-RRT qui est justement adapté pour optimiser travail mécanique.

A.5.2 Algorithme

Cette section présente l'algorithme SDAP (voir 4.1) qui, tout en se basant sur T-RRT, explore l'espace des configurations de manière simultanée pour l'ensemble des designs. Elle décrit notamment :

- le mécanisme de sélection des configurations à étendre à chaque itération,
- le mécanisme de contrôle de l'extension de l'arbre de recherche, qui se base sur le Metropolis Criterion, mais pour lequel des températures distinctes sont gérées pour chaque design possible,
- et les propriétés théorique de l'algorithme (complétude probabiliste, et non optimalité du chemin solution à l'instar de T-RRT).

A.5.3 Analyse empirique et résultats

Pour preuve de concept, SDAP appliqué à des problèmes académiques simples.

A.5.3.1 Description des systèmes de test

Cette section décrit les systèmes de test qui seront utilisés pour faire l'analyse empirique. Ces systèmes sont des chaines articulées de corps circulaires. Chaque corps est associé à une charge électrostatique qui peut être positive, négative, ou neutre. Des obstacles circulaires ayant des charges électrostatiques sont également placés dans l'environnement. Une fonction de coût est construite : elle contient deux termes permettant d'assurer qu'il n'y ait pas de collision entre les différents corps du système d'une part, et de représenter le potentiel électrostatique généré par les différents corps chargés d'autre part. L'objectif du problème est de trouver le design et le chemin pour lesquels le travail mécanique le long du chemin sera optimal.

Trois scénarios différents sont construits :

- Un petit scénario dans lequel la chaine articulée a 4 corps.
- Un grand scénario dans lequel la chaine articulée a 10 corps, mais pour lequel les 7 premiers corps ont une charge déterminée (la conception porte uniquement 3 corps)
- Le même grand scénario, mais pour lequel seulement 6 corps ont une charge déterminée (la conception porte sur 4 corps).

Pour les trois scénarios, le problème consiste à trouver le chemin pour passer d'une configuration étendue vers la gauche à une configuration étendue vers la droite (voir figure 4.2).

A.5.3.2 Résultats

Cette section présente les résultats obtenus pour l'ensemble des scénarios. Ces résultats montrent que, dans tous les cas, SDAP a identifié avec succès les designs qui donnaient lieu au chemin de coût le plus faible. SDAP a trouvé des chemins correspondant à ces designs ayant des coûts proches des résultats trouvés par l'approche naïve (T-RRT n'offre aucune garantie d'optimalité). Un gain de temps significatif est accompli par SDAP et ce gain en temps grandi avec la dimension du problème. Les temps de calcul sont améliorés d'un facteur :

- 2 pour le petit scénario,
- 26 pour le grand scénario avec 3 corps à caractéristiques variables,
- et 46 pour le grand scénario avec 4 corps à caractéristiques variables.

A.5.4 Application de SDAP à la conception d'un mouvement de protéine

Dans cette section, nous décrivons les étapes nécessaires à l'application de SDAP à un problème de conception de protéine pour un mouvement donné.

A.5.4.1 Définition de problème

Cette section fait un parallèle entre le problème formel définit dans la section 4.1.1 et le problème de conception d'une protéine pour un mouvement donné. La difficulté principale réside dans le traitement des degrés de liberté des chaines latérales et plusieurs solutions sont proposées pour traiter ce problème.

A.5.4.2 Simplification du problème

Cette section explique comment la combinatoire gigantesque de l'espace des designs d'une protéine peut être réduite en utilisant un modèle gros grain de la protéine : le modèle BLN. L'utilisation de ce modèle permet également de résoudre les problématiques liées aux chaînes latérales, puisqu'elles ne sont pas considérées de manière explicite.

A.5.4.3 Expérimentation préliminaire

Dans cette section, nous tentons d'appliquer SDAP à un problème de conception de boucle de protéine avec l'espoir de retrouver un résultat connu de la littérature. Cette expérimentation tente, en utilisant le modèle BLN, de trouver les classes d'acides aminés des 4 résidus centraux de la boucle Met20 de la protéine *ec*DHFR qui vont optimiser un mouvement d'ouverture de cette boucle.

Les résultats de cette expérience montrent que SDAP a bien trouvé un chemin correspondant au mouvement d'ouverture de la boucle, mais la fonction d'énergie gros grain associée au modèle BLN n'a pas permis de différencier les différents designs d'un point de vue de la qualité du chemin. Ce résultat montre que pour pouvoir appliquer SDAP à un problème de CPD réel, il est nécessaire de subvenir aux limitations qui l'empêchent aujourd'hui de traiter des problèmes ayant une combinatoire trop grande.

A.6 Conclusions

Cette thèse a présenté plusieurs contributions ayant pour but d'améliorer les méthodes de conception de protéine assistée par ordinateur. Tout d'abord, un modèle inspiré de la robotique a été présenté pour les protéines. Il se base sur une représentation mécanique de la protéine, sur un découpage en tripeptides, et sur l'application des méthodes de résolution de cinématique inverse 6R. Ce modèle permet d'introduire une approche simple et unifiée pour construire des classes d'échantillonnage locale des configurations du squelette protéique. Une comparaison de différentes classes de mouvement, implémentées en utilisant cette approche, au sein d'une méthode de MC, a permis de mettre en valeur les différences d'efficacité entre les différentes classes de mouvement. La conclusion générale de ces résultats est qu'utiliser une combinaison des différentes classes de mouvement donne de meilleurs résultats que d'utiliser ces même classes de mouvement toutes seules.

Grâce au modèle proposé, cette combinaison de classes de mouvement au sein d'une même simulation est très facile à réaliser.

Dans un second temps, différents algorithmes d'exploration du paysage énergétique des protéines, inspirés de la robotique, ont été présentés. L'algorithme T-RRT explore l'espace en utilisant le biais de Voronoï pour pousser la recherche vers les régions inexplorées d'une part, et un test de transition a température variable pour contenir l'arbre d'exploration dans les régions de basse énergie d'autre part. Deux autres algorithmes, EST_{kpiece} et EST_{ss} sont basés sur le principe de l'algorithme EST. Ils reposent sur une heuristique pour décider des régions de l'espace des conformations qui doivent être explorées en priorité. L'heuristique utilisée par EST_{kpiece} est basée sur une grille, construite à partir d'une projection en basse dimension de l'espace des conformations, pour estimer la couverture de l'espace. L'heuristique utilisée par EST_{ss} est basée sur une fonction de score qui évalue, pour chaque nœud, l'augmentation de couverture de l'espace attendu en échantillonnant le voisinage de ce nœud. Ces deux algorithmes sont combinés au test de transition à température variable de T-RRT. Les trois algorithmes ont ensuite été comparés à une simple méthode de MC. Les résultats suggèrent que la combinaison du test de transition de T-RRT avec les approches EST n'est pas efficace dans l'état actuel. L'algorithme T-RRT a montré de meilleures capacités d'exploration.

Pour finir, un nouveau problème combinant conception de système et planification de mouvement a été présenté et formalisé. Ce problème s'inspire du problème de conception de mouvement de protéine. Un algorithme, SDAP, basé sur T-RRT, a été proposé pour résoudre ce problème. SDAP effectue une exploration de l'espace des conformations simultanée pour l'ensemble des designs. Une analyse comparative empirique de cette algorithme, contre une approche naïve, a permis de montrer, sur des exemples simples, les avantages de SDAP. SDAP a ensuite été appliqué à une protéine, *ec*DHFR, dans le but d'explorer l'effet de certaines mutations pour lesquelles des résultats expérimentaux existent. Un modèle gros grain a été utilisé pour ces tests. Malheureusement, ce modèle n'a pas permis à SDAP d'obtenir les résultats escomptés, les différentes séquences n'ayant pas pu être filtrées au cours de l'exploration.

Perspectives de travaux futurs

Le travail présenté dans cette thèse peut être étendu dans différentes directions. Une première direction à explorer est l'intégration des classes de mouvement développées dans le chapitre 2 avec les algorithmes basés sur EST présentés dans le chapitre 3. En effet, ces classes de mouvement ont montré qu'elles avaient une efficacité supérieure par rapport à des perturbations aléatoires des angles de torsions, comme utilisé dans le chapitre 3. En outre, cette approche, basée sur des perturbations de particules, permet de définir un grand nombre de classes de mouvement distinctes en utilisant un seul et unique modèle rendant possible la combinaison de plusieurs classes de mouvement. L'intégration de cette approche aux algorithmes de type EST permettrait de bénéficier de l'efficacité des classes de mouvement com-

binées avec l'intelligence de l'exploration permis par le choix d'une heuristique EST.

De plus, comme indiqué dans le chapitre 2, l'approche de découpage en tripeptides permet d'implémenter des mouvements biaisés pour déformer certaines régions de la protéine dans une direction particulière. Cette propriété serait particulièrement utile pour concevoir des classes de mouvement prenant en compte les interactions d'une protéine avec d'autres molécules. Cela serait particulièrement adapté à l'étude des interactions protéine-protéine, ou protéine-ligand. Une autre application serait l'optimisation d'une conformation après l'introduction d'une mutation dans la séquence de la protéine. Les perturbations biaisées pourraient également être exploitées par les algorithmes d'exploration de l'espace des conformations, comme EST_{kpiece} ou EST_{ss}, dans le but d'accélérer la découverte de nouvelles régions.

Une autre direction à explorer est l'utilisation d'heuristiques plus évoluées sur les variantes de l'algorithme EST. Si ces approchent ont montré de mauvaises capacités d'exploration par rapport à l'algorithme T-RRT, dans le chapitre 3, cela vient sans doute de la trop grande simplicité des heuristiques utilisées. Dans le contexte de l'exploration d'espace à très haute dimension, les approches de type EST pourraient en fait avoir un avantage face à l'algorithme T-RRT. Avec l'augmentation de la dimension de l'espace de recherche, l'exploration de l'ensemble de l'espace devient totalement impossible. Et bien que l'algorithme T-RRT confine l'exploration aux régions de basse énergie, il passe néanmoins un temps considérable à tenter de se diriger vers les régions inexplorées de l'espace (qui correspondent à de grandes cellules de Voronoï), même celles qui n'ont pas d'intérêt. Dans le cadre des algorithmes de type EST, l'heuristique qui permet de choisir le nœud à étendre permettrai de favoriser l'exploration de certaines régions en utilisant des critères plus évolués que la simple couverture de l'espace. Par exemple, une direction prometteuse pourrait être favorisée, et explorée intensivement, au détriment des autres régions de l'espace. En résumé, l'amélioration de l'heuristique, combinée avec la possibilité de biaiser les perturbations de configuration ouvrent un grand nombre de possibilités pour améliorer les capacités d'exploration des approches de type EST au sein d'espaces de grande dimension.

Une troisième direction à explorer concerne l'algorithme SDAP, dont le principe est d'explorer l'espace des configurations simultanément pour l'ensemble des designs possibles. Cette approche a été appliquée à l'algorithme T-RRT, mais elle pourrait être appliquée de la même manière à d'autres types d'algorithmes, par exemple à d'autres variantes de RRT, comme RRT*, ou AT-RRT, qui offrent des garanties d'optimalité du chemin solution. L'application de ce principe aux approches de type EST peut également être exploré.

Pour finir, la principale limitation de l'algorithme SDAP vient de l'explosion combinatoire de l'espace des designs avec la taille du système. Afin de pouvoir envisager l'utilisation de SDAP pour résoudre un problème de conception de protéine réel, il est critique de trouver un moyen de limiter le nombre de design à considérer au cours de la recherche. Une première solution à ce problème serait d'introduire des étapes d'élagage au cours de l'exploration, comme ce qui est fait dans l'algorithme SST* [Li 2014]. Il sera aussi nécessaire d'utiliser des méthodes de filtrages plus

sophistiquées, par exemple des méthodes faisant intervenir de l'apprentissage, afin de limiter la recherche à un nombre restreint de design et de contrôler la taille de l'arbre d'exploration.

Bibliography

[Al-Bluwi 2012] Ibrahim Al-Bluwi, Thierry Siméon and Juan Cortés. *Motion planning algorithms for molecular simulations: A survey.* Computer Science Review, vol. 6, no. 4, pages 125–143, 2012. (Cited in pages 18 and 94.)

[Allen 2006] Benjamin D. Allen and Stephen L. Mayo. *Dramatic performance enhancements for the FASTER optimization algorithm.* Journal of Computational Chemistry, vol. 27, no. 10, pages 1071–1075, July 2006. (Cited in pages 25 and 97.)

[Allouche 2012] David Allouche, Seydou Traoré, Isabelle André, Simon De Givry, George Katsirelos, Sophie Barbe and Thomas Schiex. *Computational protein design as a cost function network optimization problem.* In Principles and Practice of Constraint Programming, pages 840–849. Springer, 2012. (Cited in pages 24 and 97.)

[Allouche 2014] David Allouche, Isabelle André, Sophie Barbe, Jessica Davies, Simon de Givry, George Katsirelos, Barry O'Sullivan, Steve Prestwich, Thomas Schiex and Seydou Traoré. *Computational protein design as an optimization problem.* Artificial Intelligence, vol. 212, pages 59–79, July 2014. (Cited in pages 24 and 97.)

[Altis 2007] Alexandros Altis, Phuong H. Nguyen, Rainer Hegger and Gerhard Stock. *Dihedral angle principal component analysis of molecular dynamics simulations.* The Journal of Chemical Physics, vol. 126, no. 24, page 244111, June 2007. (Cited in page 11.)

[Amato 1998] Nancy M. Amato, O. Burchan Bayazit, Lucia K. Dale, Christopher Jones and Daniel Vallejo. *OBPRM: An Obstacle-based PRM for 3D Workspaces.* In Proceedings of the Third Workshop on the Algorithmic Foundations of Robotics on Robotics : The Algorithmic Perspective: The Algorithmic Perspective, WAFR '98, pages 155–168, Natick, MA, USA, 1998. A. K. Peters, Ltd. (Cited in page 18.)

[Amato 2003] Nancy M. Amato, Ken A. Dill and Guang Song. *Using motion planning to map protein folding landscapes and analyze folding kinetics of known native structures.* Journal of Computational Biology: A Journal of Computational Molecular Cell Biology, vol. 10, no. 3-4, pages 239–255, 2003. (Cited in page 18.)

[Anderson 1999] E. Anderson, Z. Bai, C. Bischof, S. Blackford, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney and D. Sorensen. LAPACK Users' Guide: Third Edition. SIAM, 1999. (Cited in page 33.)

[Anfinsen 1972] C B Anfinsen. *The formation and stabilization of protein structure.* Biochemical Journal, vol. 128, no. 4, pages 737–749, July 1972. (Cited in pages 8 and 91.)

[Apostolakis 1998] Joannis Apostolakis, Andreas Plückthun and Amedeo Caflisch. *Docking small ligands in flexible binding sites.* Journal of Computational Chemistry, vol. 19, no. 1, pages 21–37, 1998. (Cited in pages 10 and 92.)

[Auer 2002] Peter Auer, Nicolo Cesa-Bianchi and Paul Fischer. *Finite-time analysis of the multiarmed bandit problem.* Machine learning, vol. 47, no. 2-3, pages 235–256, 2002. (Cited in pages 50 and 51.)

[Bahar 1997] I. Bahar and R. L. Jernigan. *Inter-residue potentials in globular proteins and the dominance of highly specific hydrophilic interactions at close separation.* Journal of Molecular Biology, vol. 266, no. 1, pages 195–214, February 1997. (Cited in page 12.)

[Beck 2007] David A. C. Beck, George W. N. White and Valerie Daggett. *Exploring the energy landscape of protein folding using replica-exchange and conventional molecular dynamics simulations.* Journal of Structural Biology, vol. 157, no. 3, pages 514–523, March 2007. (Cited in page 16.)

[Berman 2000] Helen M. Berman, John Westbrook, Zukang Feng, Gary Gilliland, T. N. Bhat, Helge Weissig, Ilya N. Shindyalov and Philip E. Bourne. *The Protein Data Bank.* Nucleic Acids Research, vol. 28, no. 1, pages 235–242, January 2000. (Cited in page 9.)

[Bernado 2005] P. Bernado, L. Blanchard, P. Timmins, D. Marion, R. W. H. Ruigrok and M. Blackledge. *A structural model for unfolded proteins from residual dipolar couplings and small-angle x-ray scattering.* Proceedings of the National Academy of Sciences, vol. 102, no. 47, pages 17002–17007, November 2005. (Cited in page 36.)

[Betancourt 2005] MR Betancourt. *Efficient Monte Carlo trial moves for polypeptide simulations.* The Journal of chemical physics, vol. 123, no. 17, pages 174905–174905, 2005. (Cited in page 33.)

[Betz 1996] S. F. Betz and W. F. DeGrado. *Controlling topology and native-like behavior of de novo-designed peptides: design and characterization of antiparallel four-stranded coiled coils.* Biochemistry, vol. 35, no. 21, pages 6955–6962, May 1996. (Cited in page 22.)

[Bondi 1964] A_ Bondi. *van der Waals volumes and radii.* The Journal of physical chemistry, vol. 68, no. 3, pages 441–451, 1964. (Cited in pages 13, 36, and 49.)

[Bottaro 2012] Sandro Bottaro, Wouter Boomsma, Kristoffer E. Johansson, Christian Andreetta, Thomas Hamelryck and Jesper Ferkinghoff-Borg. *Subtle*

Monte Carlo Updates in Dense Molecular Systems. Journal of Chemical Theory and Computation, vol. 8, no. 2, pages 695–702, February 2012. (Cited in page 41.)

[Bowman 2009] Gregory R. Bowman and Vijay S. Pande. *Simulated tempering yields insight into the low-resolution Rosetta scoring functions.* Proteins, vol. 74, no. 3, pages 777–788, February 2009. (Cited in page 14.)

[Brooks 2009] B. R. Brooks, C. L. Brooks, A. D. Mackerell, L. Nilsson, R. J. Petrella, B. Roux, Y. Won, G. Archontis, C. Bartels, S. Boresch, A. Caflisch, L. Caves, Q. Cui, A. R. Dinner, M. Feig, S. Fischer, J. Gao, M. Hodoscek, W. Im, K. Kuczera, T. Lazaridis, J. Ma, V. Ovchinnikov, E. Paci, R. W. Pastor, C. B. Post, J. Z. Pu, M. Schaefer, B. Tidor, R. M. Venable, H. L. Woodcock, X. Wu, W. Yang, D. M. York and M. Karplus. *CHARMM: The biomolecular simulation program.* Journal of Computational Chemistry, vol. 30, no. 10, pages 1545–1614, July 2009. (Cited in pages 13 and 93.)

[Brown 2003] Scott Brown, Nicolas J. Fawzi and Teresa Head-Gordon. *Coarse-grained sequences for protein folding and design.* Proceedings of the National Academy of Sciences, vol. 100, no. 19, pages 10712–10717, September 2003. (Cited in page 78.)

[Bruce 2002] J. Bruce and M. Veloso. *Real-time randomized path planning for robot navigation.* In IEEE/RSJ International Conference on Intelligent Robots and Systems, 2002, volume 3, pages 2383–2388 vol.3, 2002. (Cited in page 20.)

[Brunette 2008] Tj Brunette and Oliver Brock. *Guiding conformation space search with an all-atom energy potential.* Proteins: Structure, Function, and Bioinformatics, vol. 73, no. 4, pages 958–972, December 2008. (Cited in page 48.)

[Canutescu 2003] Adrian A. Canutescu and Roland L. Dunbrack. *Cyclic coordinate descent: A robotics algorithm for protein loop closure.* Protein Science, vol. 12, no. 5, pages 963–972, May 2003. (Cited in page 32.)

[Carr 2005] Joanne M. Carr, Semen A. Trygubenko and David J. Wales. *Finding pathways between distant local minima.* The Journal of Chemical Physics, vol. 122, no. 23, page 234903, June 2005. (Cited in page 44.)

[Cavasotto 2005] Claudio N. Cavasotto, Andrew JW Orry and Ruben A. Abagyan. *The challenge of considering receptor flexibility in ligand docking and virtual screening.* Current Computer-Aided Drug Design, vol. 1, no. 4, pages 423–440, 2005. (Cited in page 10.)

[Cheng 2002] Peng Cheng and S. M. LaValle. *Resolution complete rapidly-exploring random trees.* In IEEE International Conference on Robotics and Automation, 2002. Proceedings. ICRA '02, volume 1, pages 267–272 vol.1, 2002. (Cited in page 20.)

[Choset 2005] Howie Choset, Kevin M. Lynch, Seth Hutchinson, George A. Kantor, Wolfram Burgard, Lydia E. Kavraki and Sebastian Thrun. Principles of Robot Motion: Theory, Algorithms, and Implementations. A Bradford Book, Cambridge, Mass, May 2005. (Cited in pages 17 and 94.)

[Clark 2006] Louis A. Clark, P. Ann Boriack-Sjodin, John Eldredge, Christopher Fitch, Bethany Friedman, Karl J.M. Hanf, Matthew Jarpe, Stefano F. Liparoto, You Li, Alexey Lugovskoy, Stephan Miller, Mia Rushe, Woody Sherman, Kenneth Simon and Herman Van Vlijmen. *Affinity enhancement of an in vivo matured therapeutic antibody using structure-based computational design.* Protein Science : A Publication of the Protein Society, vol. 15, no. 5, pages 949–960, May 2006. (Cited in page 22.)

[Clementi 2008] Cecilia Clementi. *Coarse-grained models of protein folding: toy models or predictive tools?* Current Opinion in Structural Biology, vol. 18, no. 1, pages 10–15, February 2008. (Cited in pages 11 and 14.)

[Colubri 2004] Andrés Colubri. *Prediction of protein structure by simulating coarse-grained folding pathways: a preliminary report.* Journal of Biomolecular Structure and Dynamics, vol. 21, no. 5, pages 625–638, 2004. (Cited in page 14.)

[Cortés 2004] J. Cortés, T. Siméon, M. Remaud-Siméon and V. Tran. *Geometric algorithms for the conformational analysis of long protein loops.* Journal of Computational Chemistry, vol. 25, no. 7, pages 956–967, May 2004. (Cited in page 33.)

[Cortés 2005] J. Cortés, T. Siméon, V. Ruiz de Angulo, D. Guieysse, M. Remaud-Siméon and V. Tran. *A path planning approach for computing large-amplitude motions of flexible molecules.* Bioinformatics (Oxford, England), vol. 21 Suppl 1, pages i116–125, June 2005. (Cited in page 2.)

[Cortés 2008] J. Cortés, L. Jaillet and T. Siméon. *Disassembly Path Planning for Complex Articulated Objects.* IEEE Transactions on Robotics, vol. 24, no. 2, pages 475–481, April 2008. (Cited in pages 20 and 95.)

[Cortés 2010a] Juan Cortés, Sergio Carrión, David Curcó, Marc Renaud and Carlos Alemán. *Relaxation of amorphous multichain polymer systems using inverse kinematics.* Polymer, vol. 51, no. 17, pages 4008–4014, August 2010. (Cited in page 33.)

[Cortés 2010b] Juan Cortés, Duc Thanh Le, Romain Iehl and Thierry Siméon. *Simulating ligand-induced conformational changes in proteins using a mechanical disassembly method.* Physical Chemistry Chemical Physics, vol. 12, no. 29, pages 8268–8276, July 2010. (Cited in pages 20 and 95.)

[Cortés 2012] Juan Cortés and Ibrahim Al-Bluwi. *A robotics approach to enhance conformational sampling of proteins.* In ASME 2012 International Design

Engineering Technical Conferences and Computers and Information in Engineering Conference, pages 1177–1186. American Society of Mechanical Engineers, 2012. (Cited in pages 2, 28, and 98.)

[Coutsias 2004] Evangelos A. Coutsias, Chaok Seok, Matthew P. Jacobson and Ken A. Dill. *A kinematic view of loop closure.* Journal of Computational Chemistry, vol. 25, no. 4, pages 510–528, March 2004. (Cited in page 33.)

[Craig 2004] John J. Craig. Introduction to Robotics: Mechanics and Control. Pearson, Upper Saddle River, N.J, 3 edition édition, August 2004. (Cited in page 28.)

[Cui 2005] Qiang Cui and Ivet Bahar, editeurs. Normal Mode Analysis: Theory and Applications to Biological and Chemical Systems. Chapman and Hall/CRC, Boca Raton, 1 edition édition, December 2005. (Cited in pages 11 and 92.)

[Dahiyat 1996] B. I. Dahiyat and S. L. Mayo. *Protein design automation.* Protein Science: A Publication of the Protein Society, vol. 5, no. 5, pages 895–903, May 1996. (Cited in page 22.)

[Dahiyat 1997] B. I. Dahiyat and S. L. Mayo. *De novo protein design: fully automated sequence selection.* Science (New York, N.Y.), vol. 278, no. 5335, pages 82–87, October 1997. (Cited in pages 14 and 22.)

[Das 2006] Payel Das, Mark Moll, Hernán Stamati, Lydia E. Kavraki and Cecilia Clementi. *Low-dimensional, free-energy landscapes of protein-folding reactions by nonlinear dimensionality reduction.* Proceedings of the National Academy of Sciences, vol. 103, no. 26, pages 9885–9890, June 2006. (Cited in page 11.)

[Das 2007] Rhiju Das, Bin Qian, Srivatsan Raman, Robert Vernon, James Thompson, Philip Bradley, Sagar Khare, Michael D. Tyka, Divya Bhat, Dylan Chivian, David E. Kim, William H. Sheffler, Lars Malmström, Andrew M. Wollacott, Chu Wang, Ingemar Andre and David Baker. *Structure prediction for CASP7 targets using extensive all-atom refinement with Rosetta@home.* Proteins, vol. 69 Suppl 8, pages 118–128, 2007. (Cited in page 14.)

[De Simone 2015] Alfonso De Simone, Francesco A. Aprile, Anne Dhulesia, Christopher M. Dobson and Michele Vendruscolo. *Structure of a low-population intermediate state in the release of an enzyme product.* Elife, vol. 4, page e02777, 2015. (Cited in page 26.)

[Denarie 2016] Laurent Denarie, Kevin Molloy, Marc Vaisset, Thierry Siméon and Juan Cortés. *Combining System Design and Path Planning.* In Workshop on the Algorithmic Foundations of Robotics (WAFR), December 2016. (Cited in page 3.)

[Denarie 2017] Laurent Denarie, Ibrahim Al-Bluwi, Marc Vaisset, Thierry Siméon and Juan Cortés. *Segmenting proteins into tripeptides to enhance conformational sampling with Monte Carlo methods.* Journal of Chemical Theory and Computation, vol. Submitted, 2017. (Cited in page 2.)

[Der 2012] Bryan S. Der, Mischa Machius, Michael J. Miley, Jeffrey L. Mills, Thomas Szyperski and Brian Kuhlman. *Metal-Mediated Affinity and Orientation Specificity in a Computationally Designed Protein Homodimer.* Journal of the American Chemical Society, vol. 134, no. 1, pages 375–385, January 2012. (Cited in page 22.)

[Desjarlais 1995] J. R. Desjarlais and T. M. Handel. *De novo design of the hydrophobic cores of proteins.* Protein Science: A Publication of the Protein Society, vol. 4, no. 10, pages 2006–2018, October 1995. (Cited in pages 22 and 25.)

[Desmet 1992] Johan Desmet, Marc De Maeyer, Bart Hazes and Ignace Lasters. *The dead-end elimination theorem and its use in protein side-chain positioning.* Nature, vol. 356, no. 6369, pages 539–542, April 1992. (Cited in pages 24 and 97.)

[Desmet 2002] Johan Desmet, Jan Spriet and Ignace Lasters. *Fast and accurate side-chain topology and energy refinement (FASTER) as a new method for protein structure optimization.* Proteins, vol. 48, no. 1, pages 31–43, July 2002. (Cited in pages 25 and 97.)

[Devaurs 2013a] Didier Devaurs, Léa Bouard, Marc Vaisset, Christophe Zanon, Ibrahim Al-Bluwi, Romain Iehl, Thierry Siméon and Juan Cortés. *MoMA-LigPath: a web server to simulate protein–ligand unbinding.* Nucleic Acids Research, vol. 41, no. W1, pages W297–W302, July 2013. (Cited in page 2.)

[Devaurs 2013b] Didier Devaurs, Marc Vaisset, Thierry Siméon and Juan Cortés. *A multi-tree approach to compute transition paths on energy landscapes.* In Workshop on Artificial Intelligence and Robotics Methods in Computational Biology, AAAI'13, pages pp–8, 2013. (Cited in pages 20, 54, and 95.)

[Devaurs 2014] D. Devaurs, T. Siméon and J. Cortés. *A multi-tree extension of the transition-based RRT: Application to ordering-and-pathfinding problems in continuous cost spaces.* In 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2991–2996, September 2014. (Cited in pages 66 and 69.)

[Devaurs 2015] Didier Devaurs, Kevin Molloy, Marc Vaisset, Amarda Shehu, Thierry Siméon and Juan Cortés. *Characterizing Energy Landscapes of Peptides using a Combination of Stochastic Algorithms.* IEEE Transactions on NanoBioscience, vol. 14, no. 5, pages pp. 545–552, July 2015. (Cited in page 56.)

[Devaurs 2016] D. Devaurs, T. Siméon and J. Cortés. *Optimal Path Planning in Complex Cost Spaces With Sampling-Based Algorithms.* IEEE Transactions on Automation Science and Engineering, vol. 13, no. 2, pages 415–424, April 2016. (Cited in pages 65, 69, 73, and 77.)

[Dijkstra 1959] E. W. Dijkstra. *A Note on Two Problems in Connexion with Graphs.* Numer. Math., vol. 1, no. 1, pages 269–271, December 1959. (Cited in pages 19 and 94.)

[Dodd 1993] L. R. Dodd, T. D. Boone and D. N. Theodorou. *A concerted rotation algorithm for atomistic Monte Carlo simulation of polymer melts and glasses.* Molecular Physics, vol. 78, no. 4, pages 961–996, March 1993. (Cited in pages 31, 34, and 99.)

[Dotu 2011] Ivan Dotu, Manuel Cebrián, Pascal Van Hentenryck and Peter Clote. *On lattice protein structure prediction revisited.* IEEE/ACM transactions on computational biology and bioinformatics / IEEE, ACM, vol. 8, no. 6, pages 1620–1632, December 2011. (Cited in page 11.)

[Earl 2005] David J. Earl and Michael W. Deem. *Parallel Tempering: Theory, Applications, and New Perspectives.* Physical Chemistry Chemical Physics, vol. 7, no. 23, page 3910, 2005. arXiv: physics/0508111. (Cited in pages 17 and 94.)

[Engh 1991] R. A. Engh and R. Huber. *Accurate bond and angle parameters for X-ray protein structure refinement.* Acta Crystallographica Section A Foundations of Crystallography, vol. 47, no. 4, pages 392–400, July 1991. (Cited in pages 10 and 92.)

[Falzone 1994] C. J. Falzone, P. E. Wright and S. J. Benkovic. *Dynamics of a flexible loop in dihydrofolate reductase from Escherichia coli and its implication for catalysis.* Biochemistry, vol. 33, no. 2, pages 439–442, January 1994. (Cited in page 79.)

[Fleishman 2011] Sarel J Fleishman, Timothy A Whitehead, Damian C Ekiert, Cyrille Dreyfus, Jacob E Corn, Eva-Maria Strauch, Ian A Wilson and David Baker. *Computational design of proteins targeting the conserved stem region of influenza hemagglutinin.* Science (New York, N.Y.), vol. 332, no. 6031, pages 816–821, May 2011. (Cited in page 22.)

[Fodor 2002] Imola K. Fodor. A survey of dimension reduction techniques. Technical Report UCRL-ID-148494, Lawrence Livermore National Laboratory, 2002. (Cited in pages 10 and 92.)

[Frenkel 2001] Daan Frenkel and Berend Smit. Understanding Molecular Simulation, Second Edition: From Algorithms to Applications. Academic Press, San Diego, 2 edition édition, November 2001. (Cited in page 15.)

[Fung 2007] Ho Ki Fung, M. S. Taylor and Christodoulos A. Floudas. *Novel formulations for the sequence selection problem in de novo protein design with flexible templates.* Optimisation Methods and Software, vol. 22, no. 1, pages 51–71, 2007. (Cited in page 24.)

[Fung 2008] Ho Ki Fung, Christodoulos A. Floudas, Martin S. Taylor, Li Zhang and Dimitrios Morikis. *Toward Full-Sequence De Novo Protein Design with Flexible Templates for Human Beta-Defensin-2.* Biophysical Journal, vol. 94, no. 2, pages 584–599, January 2008. (Cited in pages 24 and 96.)

[Gaillard 2014] Thomas Gaillard and Thomas Simonson. *Pairwise decomposition of an MMGBSA energy function for computational protein design.* Journal of Computational Chemistry, vol. 35, no. 18, pages 1371–1387, July 2014. (Cited in page 14.)

[Gainza 2012] Pablo Gainza, Kyle E. Roberts and Bruce R. Donald. *Protein design using continuous rotamers.* PLoS computational biology, vol. 8, no. 1, page e1002335, January 2012. (Cited in page 23.)

[Georgiev 2006] I. Georgiev, R. H. Lilien and B. R. Donald. *Improved Pruning algorithms and Divide-and-Conquer strategies for Dead-End Elimination, with application to protein design.* Bioinformatics, vol. 22, no. 14, pages e174–e183, July 2006. (Cited in pages 24 and 97.)

[Georgiev 2007] Ivelin Georgiev and Bruce R. Donald. *Dead-End Elimination with Backbone Flexibility.* Bioinformatics, vol. 23, no. 13, pages i185–i194, July 2007. (Cited in page 24.)

[Georgiev 2008] Ivelin Georgiev, Ryan H. Lilien and Bruce R. Donald. *The minimized dead-end elimination criterion and its application to protein redesign in a hybrid scoring and search algorithm for computing partition functions over molecular ensembles.* Journal of Computational Chemistry, vol. 29, no. 10, pages 1527–1542, July 2008. (Cited in pages 24 and 97.)

[Geraerts 2004] Roland Geraerts and Mark H. Overmars. *A Comparative Study of Probabilistic Roadmap Planners.* In Prof Jean-Daniel Boissonnat, Prof Joel Burdick, Prof Ken Goldberg and Prof Seth Hutchinson, editeurs, Algorithmic Foundations of Robotics V, numéro 7 de Springer Tracts in Advanced Robotics, pages 43–57. Springer Berlin Heidelberg, 2004. DOI: 10.1007/978-3-540-45058-0_4. (Cited in page 18.)

[Geyer 1991] Charles J. Geyer. *Markov chain Monte Carlo maximum likelihood.* Rapport technique, School of Statistics University of Minnesota, 1991. (Cited in page 16.)

[Gō 1970] Nobuhiro Gō and Harold A. Scheraga. *Ring Closure and Local Conformational Deformations of Chain Molecules.* Macromolecules, vol. 3, no. 2, pages 178–187, March 1970. (Cited in page 32.)

[Goldstein 1994] R F Goldstein. *Efficient rotamer elimination applied to protein side-chains and related spin glasses.* Biophysical Journal, vol. 66, no. 5, pages 1335–1340, May 1994. (Cited in pages 24 and 97.)

[Golub 2012] Gene H. Golub and Charles F. Van Loan. Matrix Computations. JHU Press, December 2012. (Cited in page 33.)

[Gosselin 1991] C. Gosselin and J. Angeles. *A Global Performance Index for the Kinematic Optimization of Robotic Manipulators.* Journal of Mechanical Design, vol. 113, no. 3, pages 220–226, September 1991. (Cited in pages 63 and 103.)

[Harada 2011] Ryuhei Harada and Akio Kitao. *Exploring the folding free energy landscape of a β-hairpin miniprotein, chignolin, using multiscale free energy landscape calculation method.* The Journal of Physical Chemistry. B, vol. 115, no. 27, pages 8806–8812, July 2011. (Cited in page 59.)

[Harbury 1995] P B Harbury, B Tidor and P S Kim. *Repacking protein cores with backbone freedom: structure prediction for coiled coils.* Proceedings of the National Academy of Sciences of the United States of America, vol. 92, no. 18, pages 8408–8412, August 1995. (Cited in page 22.)

[Hart 1968] Peter Hart, Nils Nilsson and Bertram Raphael. *A Formal Basis for the Heuristic Determination of Minimum Cost Paths.* IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pages 100–107, 1968. (Cited in pages 19, 24, 94, and 97.)

[Hauser 2010] Kris Hauser and Jean-Claude Latombe. *Multi-modal Motion Planning in Non-expansive Spaces.* The International Journal of Robotics Research, vol. 29, no. 7, pages 897–915, June 2010. (Cited in pages 64 and 104.)

[Hinds 1994] David A. Hinds and Michael Levitt. *Exploring conformational space with a simple lattice model for protein structure.* Journal of Molecular Biology, vol. 243, no. 4, pages 668–682, November 1994. (Cited in page 11.)

[Hinsen 1998] K. Hinsen. *Analysis of domain motions by approximate normal mode calculations.* Proteins, vol. 33, no. 3, pages 417–429, November 1998. (Cited in page 11.)

[Hsu 1997] D. Hsu, J.-C. Latombe and R. Motwani. *Path planning in expansive configuration spaces.* In , 1997 IEEE International Conference on Robotics and Automation, 1997. Proceedings, volume 3, pages 2719–2726 vol.3, April 1997. (Cited in pages 21 and 95.)

[Hsu 2000] David Hsu. *Randomized single-query motion planning in expansive spaces.* PhD thesis, Stanford University, 2000. (Cited in pages 21 and 95.)

[Hsu 2002] David Hsu, Robert Kindel, Jean-Claude Latombe and Stephen Rock. *Randomized Kinodynamic Motion Planning with Moving Obstacles.* The International Journal of Robotics Research, vol. 21, no. 3, pages 233–255, March 2002. (Cited in pages 21 and 95.)

[Hu 2007] Xiaozhen Hu, Huanchen Wang, Hengming Ke and Brian Kuhlman. *High-resolution design of a protein loop.* Proceedings of the National Academy of Sciences, vol. 104, no. 45, pages 17668–17673, 2007. (Cited in page 24.)

[Hurley 1992] J. H. Hurley, W. A. Baase and B. W. Matthews. *Design and structural analysis of alternative hydrophobic core packing arrangements in bacteriophage T4 lysozyme.* Journal of Molecular Biology, vol. 224, no. 4, pages 1143–1159, April 1992. (Cited in page 22.)

[Jaillet 2008] L. Jaillet, J. Cortés and T. Siméon. *Transition-based RRT for path planning in continuous cost spaces.* In 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, pages 2145–2150, September 2008. (Cited in pages 20, 49, and 95.)

[Jaillet 2010] L. Jaillet, J. Cortés and T. Siméon. *Sampling-Based Path Planning on Configuration-Space Costmaps.* IEEE Transactions on Robotics, vol. 26, no. 4, pages 635–646, August 2010. (Cited in pages 3, 64, 65, 67, 68, 69, 77, and 104.)

[Jaillet 2011] Léonard Jaillet, Francesc J. Corcho, Juan-Jesús Pérez and Juan Cortés. *Randomized tree construction algorithm to explore energy landscapes.* Journal of Computational Chemistry, vol. 32, no. 16, pages 3464–3474, December 2011. (Cited in page 49.)

[Janson 2015] Lucas Janson, Edward Schmerling, Ashley Clark and Marco Pavone. *Fast Marching Tree: a Fast Marching Sampling-Based Method for Optimal Motion Planning in Many Dimensions.* The International journal of robotics research, vol. 34, no. 7, pages 883–921, June 2015. (Cited in page 66.)

[Jiang 2008] Lin Jiang, Eric A. Althoff, Fernando R. Clemente, Lindsey Doyle, Daniela Röthlisberger, Alexandre Zanghellini, Jasmine L. Gallaher, Jamie L. Betker, Fujie Tanaka, Carlos F. Barbas, Donald Hilvert, Kendall N. Houk, Barry L. Stoddard and David Baker. *De novo computational design of retro-aldol enzymes.* Science (New York, N.Y.), vol. 319, no. 5868, pages 1387–1391, March 2008. (Cited in page 22.)

[Jones 1994] D. T. Jones. *De novo protein design using pairwise potentials and a genetic algorithm.* Protein Science: A Publication of the Protein Society, vol. 3, no. 4, pages 567–574, April 1994. (Cited in page 25.)

[Jones 1997] G. Jones, P. Willett, R. C. Glen, A. R. Leach and R. Taylor. *Development and validation of a genetic algorithm for flexible docking.* Journal

of Molecular Biology, vol. 267, no. 3, pages 727–748, April 1997. (Cited in pages 10 and 92.)

[Kabsch 1976] Wolfgang Kabsch. *A solution for the best rotation to relate two sets of vectors*. Acta Crystallographica Section A: Crystal Physics, Diffraction, Theoretical and General Crystallography, vol. 32, no. 5, pages 922–923, 1976. (Cited in page 9.)

[Kabsch 1983] Wolfgang Kabsch and Christian Sander. *Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features*. Biopolymers, vol. 22, no. 12, pages 2577–2637, 1983. (Cited in page 78.)

[Karaman 2010] Sertac Karaman and Emilio Frazzoli. *Incremental sampling-based algorithms for optimal motion planning*. Robotics Science and Systems VI, vol. 104, 2010. (Cited in page 20.)

[Karaman 2011] Sertac Karaman and Emilio Frazzoli. *Sampling-based algorithms for optimal motion planning*. The International Journal of Robotics Research, vol. 30, no. 7, pages 846–894, 2011. (Cited in page 66.)

[Kavraki 1996] L. E. Kavraki, P. Svestka, J. C. Latombe and M. H. Overmars. *Probabilistic roadmaps for path planning in high-dimensional configuration spaces*. IEEE Transactions on Robotics and Automation, vol. 12, no. 4, pages 566–580, August 1996. (Cited in pages 18, 63, 66, 94, and 104.)

[Khalili 2004] Mey Khalili, Jeffrey A. Saunders, Adam Liwo, Stanislaw Ołdziej and Harold A. Scheraga. *A united residue force-field for calcium–protein interactions*. Protein Science : A Publication of the Protein Society, vol. 13, no. 10, pages 2725–2735, October 2004. (Cited in page 12.)

[Kirillova 2007] Svetlana Kirillova, Juan Cortés, Alin Stefaniu and Thierry Siméon. *An NMA-guided path planning approach for computing large-amplitude conformational changes in proteins*. Proteins: Structure, Function, and Bioinformatics, vol. 70, no. 1, pages 131–143, July 2007. (Cited in page 11.)

[Kirkpatrick 1983] Scott Kirkpatrick, C. Daniel Gelatt, Mario P. Vecchi and others. *Optimization by simmulated annealing*. science, vol. 220, no. 4598, pages 671–680, 1983. (Cited in pages 17 and 94.)

[Kolinski 1994] A. Kolinski and J. Skolnick. *Monte Carlo simulations of protein folding. I. Lattice model and interaction scheme*. Proteins, vol. 18, no. 4, pages 338–352, April 1994. (Cited in page 11.)

[Kollman 1997] Peter Kollman, Richard Dixon, Wendy Cornell, Thomas Fox, Chris Chipot and Andrew Pohorille. *The development/application of a 'minimalist' organic/biochemical molecular mechanic force field using a combination of ab initio calculations and experimental data*. In Wilfred F. van Gunsteren,

Paul K. Weiner and Anthony J. Wilkinson, editeurs, Computer Simulation of Biomolecular Systems, numéro 3 de Computer Simulations of Biomolecular Systems, pages 83–96. Springer Netherlands, 1997. DOI: 10.1007/978-94-017-1120-3_2. (Cited in pages 13, 35, 55, and 93.)

[Krivov 2009] Georgii G. Krivov, Maxim V. Shapovalov and Roland L. Dunbrack. *Improved prediction of protein side-chain conformations with SCWRL4.* Proteins: Structure, Function, and Bioinformatics, vol. 77, no. 4, pages 778–795, 2009. (Cited in page 36.)

[Kuffner Jr 2000] James J. Kuffner Jr and Steven M. Lavalle. *RRT-Connect: An efficient approach to single-query path planning.* In Proc. IEEE Int'l Conf. on Robotics and Automation, pages 995–1001, 2000. (Cited in page 20.)

[Kuhlman 2003] Brian Kuhlman, Gautam Dantas, Gregory C. Ireton, Gabriele Varani, Barry L. Stoddard and David Baker. *Design of a novel globular protein fold with atomic-level accuracy.* Science, vol. 302, no. 5649, pages 1364–1368, 2003. (Cited in pages 22, 24, and 96.)

[Laio 2002] Alessandro Laio and Michele Parrinello. *Escaping free-energy minima.* Proceedings of the National Academy of Sciences of the United States of America, vol. 99, no. 20, pages 12562–12566, October 2002. (Cited in pages 16 and 93.)

[Latombe 1991] Jean-Claude Latombe. Robot Motion Planning. Springer US, Boston, MA, 1991. (Cited in pages 17 and 94.)

[Lavalle 1998] Steven M. Lavalle. *Rapidly-Exploring Random Trees: A New Tool for Path Planning.* Rapport technique, Iowa State University, 1998. (Cited in pages 20 and 95.)

[Lavalle 2000] Steven M. Lavalle, James J. Kuffner and Jr. *Rapidly-Exploring Random Trees: Progress and Prospects.* In Algorithmic and Computational Robotics: New Directions, pages 293–308, 2000. (Cited in pages 66 and 68.)

[LaValle 2001] Steven M. LaValle and James J. Kuffner. *Randomized Kinodynamic Planning.* The International Journal of Robotics Research, vol. 20, no. 5, pages 378–400, May 2001. (Cited in pages 20 and 95.)

[LaValle 2006] Steven M. LaValle. Planning Algorithms. Cambridge University Press, Cambridge ; New York, 1 edition édition, May 2006. (Cited in pages 63 and 104.)

[Leach 1998] Andrew R. Leach and Andrew P. Lemon. *Exploring the conformational space of protein side chains using dead-end elimination and the A* algorithm.* Proteins: Structure, Function, and Bioinformatics, vol. 33, no. 2, pages 227–239, November 1998. (Cited in page 24.)

[Leach 2001] Andrew Leach. Molecular Modelling: Principles and Applications. Pearson, Harlow, England ; New York, 2 edition édition, April 2001. (Cited in page 10.)

[Lee 1988a] Hong-You Lee and Chong-Gao Liang. *Displacement analysis of the general spatial 7-link 7R mechanism.* Mechanism and Machine Theory, vol. 23, no. 3, pages 219–226, January 1988. (Cited in page 33.)

[Lee 1988b] Hong-You Lee and Chong-Gao Liang. *A new vector theory for the analysis of spatial mechanisms.* Mechanism and Machine Theory, vol. 23, no. 3, pages 209–217, January 1988. (Cited in page 33.)

[Leontidis 1994] E. Leontidis, J. J. de Pablo, M. Laso and U. W. Suter. *A critical evaluation of novel algorithms for the off-lattice Monte Carlo simulation of condensed polymer phases.* In Prof Dr Lucien Monnerie and Prof Dr U. W. Suter, editeurs, Atomistic Modeling of Physical Properties, numéro 116 de Advances in Polymer Science, pages 283–318. Springer Berlin Heidelberg, 1994. DOI: 10.1007/BFb0080202. (Cited in page 31.)

[Li 1992] Luyuan Li, Peter E. Wright, Stephen J. Benkovic and Christopher J. Falzone. *Functional role of a mobile loop of Escherichia coli dihydrofolate reductase in transition-state stabilization.* Biochemistry, vol. 31, no. 34, pages 7826–7833, September 1992. (Cited in page 79.)

[Li 2014] Yanbo Li, Zakary Littlefield and Kostas E. Bekris. *Asymptotically Optimal Sampling-based Kinodynamic Planning.* arXiv:1407.2896 [cs], July 2014. arXiv: 1407.2896. (Cited in pages 87 and 109.)

[Lindemann 2004] S. R. Lindemann and S. M. LaValle. *Incrementally reducing dispersion by increasing Voronoi bias in RRTs.* In 2004 IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04, volume 4, pages 3251–3257 Vol.4, April 2004. (Cited in pages 20 and 95.)

[Lippow 2007] Shaun M. Lippow and Bruce Tidor. *Progress in computational protein design.* Current Opinion in Biotechnology, vol. 18, no. 4, pages 305–311, August 2007. (Cited in page 22.)

[Lozano-Perez 1983] Tomas Lozano-Perez. *Spatial planning: A configuration space approach.* IEEE transactions on computers, vol. 100, no. 2, pages 108–120, 1983. (Cited in page 17.)

[Luo 2002] Peizhi Luo, Robert J. Hayes, Cheryl Chan, Diane M. Stark, Marian Y. Hwang, Jonathan M. Jacinto, Padmaja Juvvadi, Helen S. Chung, Anirban Kundu, Marie L. Ary and Bassil I. Dahiyat. *Development of a cytokine analog with enhanced stability using computational ultrahigh throughput screening.* Protein Science: A Publication of the Protein Society, vol. 11, no. 5, pages 1218–1226, May 2002. (Cited in page 22.)

[Mandell 2009] Daniel J. Mandell and Tanja Kortemme. *Backbone flexibility in computational protein design.* Current Opinion in Biotechnology, vol. 20, no. 4, pages 420–428, August 2009. (Cited in page 44.)

[Manocha 1994] D. Manocha and J. F. Canny. *Efficient inverse kinematics for general 6R manipulators.* IEEE Transactions on Robotics and Automation, vol. 10, no. 5, pages 648–657, October 1994. (Cited in page 33.)

[McCarthy 2001] J. M. McCarthy and L. Joskowitz. *Kinematic Synthesis.* In J. Cagan and E. Antonson, editeurs, Formal Engineering Design Synthesis. Cambridge Univ. Press., 2001. (Cited in pages 63 and 103.)

[Merlet 2005] Jean-Pierre Merlet. *Optimal design of robots.* In Robotics: Science and systems, June 2005. (Cited in pages 63 and 103.)

[Metropolis 1953] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller and Edward Teller. *Equation of State Calculations by Fast Computing Machines.* The Journal of Chemical Physics, vol. 21, no. 6, page 1087, 1953. (Cited in pages 16 and 93.)

[Moll 2008] Mark Moll, David Schwarz and Lydia E. Kavraki. *Roadmap methods for protein folding.* Methods in Molecular Biology (Clifton, N.J.), vol. 413, pages 219–239, 2008. (Cited in pages 18 and 94.)

[Molloy 2014] Kevin Molloy and Amarda Shehu. *A probabilistic roadmap-based method to model conformational switching of a protein among many functionally-relevant structures.* In Intl Conf on Bioinf and Comp Biol (BICoB), Las Vegas, NV, 2014. (Cited in pages 18, 19, and 48.)

[Monticelli 2008] Luca Monticelli, Senthil K. Kandasamy, Xavier Periole, Ronald G. Larson, D. Peter Tieleman and Siewert-Jan Marrink. *The MARTINI Coarse-Grained Force Field: Extension to Proteins.* Journal of Chemical Theory and Computation, vol. 4, no. 5, pages 819–834, May 2008. (Cited in page 12.)

[Mu 2005] Yuguang Mu, Phuong H. Nguyen and Gerhard Stock. *Energy landscape of a small peptide revealed by dihedral angle principal component analysis.* Proteins, vol. 58, no. 1, pages 45–52, January 2005. (Cited in page 11.)

[Mukherjee 2004] Arnab Mukherjee and Biman Bagchi. *Contact pair dynamics during folding of two small proteins: Chicken villin head piece and the Alzheimer protein β-amyloid.* The Journal of chemical physics, vol. 120, no. 3, pages 1602–1612, 2004. (Cited in pages 12 and 14.)

[Murphy 2009] Paul M. Murphy, Jill M. Bolduc, Jasmine L. Gallaher, Barry L. Stoddard and David Baker. *Alteration of enzyme specificity by computational loop remodeling and design.* Proceedings of the National Academy of Sciences, vol. 106, no. 23, pages 9215–9220, 2009. (Cited in page 24.)

[Neudecker 2012] Philipp Neudecker, Paul Robustelli, Andrea Cavalli, Patrick Walsh, Patrik Lundström, Arash Zarrine-Afsar, Simon Sharpe, Michele Vendruscolo and Lewis E. Kay. *Structure of an Intermediate State in Protein Folding and Aggregation.* Science, vol. 336, no. 6079, pages 362–366, April 2012. (Cited in pages 8, 26, and 91.)

[Nguyen 2005] Phuong H. Nguyen, Yuguang Mu and Gerhard Stock. *Structure and energy landscape of a photoswitchable peptide: a replica exchange molecular dynamics study.* Proteins, vol. 60, no. 3, pages 485–494, August 2005. (Cited in page 16.)

[Nilmeier 2008] Jerome Nilmeier and Matt Jacobson. *Multiscale Monte Carlo Sampling of Protein Sidechains: Application to Binding Pocket Flexibility.* Journal of Chemical Theory and Computation, vol. 4, no. 5, pages 835–846, May 2008. (Cited in page 31.)

[Nilmeier 2009] Jerome Nilmeier and Matthew P. Jacobson. *Monte Carlo Sampling with Hierarchical Move Sets: POSH Monte Carlo.* Journal of Chemical Theory and Computation, vol. 5, no. 8, pages 1968–1984, August 2009. (Cited in page 31.)

[Oakley 2011] Mark T. Oakley, David J. Wales and Roy L. Johnston. *Energy Landscape and Global Optimization for a Frustrated Model Protein.* The Journal of Physical Chemistry B, vol. 115, no. 39, pages 11525–11529, October 2011. (Cited in page 12.)

[Okamoto 2004] Yuko Okamoto. *Generalized-ensemble algorithms: enhanced sampling techniques for Monte Carlo and molecular dynamics simulations.* Journal of Molecular Graphics and Modelling, vol. 22, no. 5, pages 425–439, May 2004. (Cited in page 44.)

[Okazaki 2006] Kei-ichi Okazaki, Nobuyasu Koga, Shoji Takada, Jose N. Onuchic and Peter G. Wolynes. *Multiple-basin energy landscapes for large-amplitude conformational motions of proteins: Structure-based molecular dynamics simulations.* Proceedings of the National Academy of Sciences of the United States of America, vol. 103, no. 32, pages 11844–11849, August 2006. (Cited in page 13.)

[Onuchic 1997] J. N. Onuchic, Z. Luthey-Schulten and P. G. Wolynes. *Theory of protein folding: the energy landscape perspective.* Annual Review of Physical Chemistry, vol. 48, pages 545–600, 1997. (Cited in page 13.)

[Onuchic 2004] José Nelson Onuchic and Peter G. Wolynes. *Theory of protein folding.* Current Opinion in Structural Biology, vol. 14, no. 1, pages 70–75, February 2004. (Cited in page 13.)

[Osborne 2001] Michael J. Osborne, Jason Schnell, Stephen J. Benkovic, H. Jane Dyson and Peter E. Wright. *Backbone Dynamics in Dihydrofolate Reductase*

Complexes: Role of Loop Flexibility in the Catalytic Mechanism†. Biochemistry, vol. 40, no. 33, pages 9846–9859, August 2001. (Cited in page 79.)

[Pak 2000] Youngshang Pak and Shaomeng Wang. *Application of a Molecular Dynamics Simulation Method with a Generalized Effective Potential to the Flexible Molecular Docking Problems.* The Journal of Physical Chemistry B, vol. 104, no. 2, pages 354–359, January 2000. (Cited in pages 10 and 92.)

[Park 2004] Sanghyun Park and Klaus Schulten. *Calculating potentials of mean force from steered molecular dynamics simulations.* The Journal of Chemical Physics, vol. 120, no. 13, page 5946, 2004. (Cited in page 16.)

[Parsons 1994] David Parsons and John Canny. *Geometric Problems in Molecular Biology and Robotics.* ResearchGate, July 1994. (Cited in pages 18 and 94.)

[Periole 2007] Xavier Periole and Alan E. Mark. *Convergence and sampling efficiency in replica exchange simulations of peptide folding in explicit solvent.* The Journal of Chemical Physics, vol. 126, no. 1, page 014903, 2007. (Cited in page 16.)

[Pierce 2000] Niles A. Pierce, Jan A. Spriet, Johan Desmet and Stephen L. Mayo. *Conformational splitting: A more powerful criterion for dead-end elimination.* Journal of computational chemistry, vol. 21, no. 11, pages 999–1009, 2000. (Cited in pages 24 and 97.)

[Pierce 2002] Niles A. Pierce and Erik Winfree. *Protein design is NP-hard.* Protein engineering, vol. 15, no. 10, pages 779–782, 2002. (Cited in pages 24 and 96.)

[Polydorides 2011] Savvas Polydorides, Najette Amara, Caroline Aubard, Pierre Plateau, Thomas Simonson and Georgios Archontis. *Computational protein design with a generalized born solvent model: Application to asparaginyl-tRNA synthetase.* Proteins: Structure, Function, and Bioinformatics, vol. 79, no. 12, pages 3448–3468, December 2011. (Cited in pages 25 and 97.)

[Ponder 1987] J. W. Ponder and F. M. Richards. *Tertiary templates for proteins. Use of packing criteria in the enumeration of allowed sequences for different structural classes.* Journal of Molecular Biology, vol. 193, no. 4, pages 775–791, February 1987. (Cited in pages 23 and 96.)

[Renaud 2000] M. Renaud. *A simplified inverse Kinematric model calculation method for all 6R type manipulators.* In Current Advances in Mechanical Design and Production VII, pages 15–25. Elsevier, 2000. (Cited in page 33.)

[Renaud 2006] M. Renaud. *Calcul des modèles géométriques inverses des robots manipulateurs 6R.* Rapport technique 06332, LAAS, 2006. (Cited in page 33.)

[Rodriguez 2006] Rodriguez, Xinyu Tang, Jyh-Ming Lien and N. M. Amato. *An obstacle-based rapidly-exploring random tree.* In Proceedings 2006 IEEE

International Conference on Robotics and Automation, 2006. ICRA 2006., pages 895–900, May 2006. (Cited in page 20.)

[Rohrdanz 2011] Mary A. Rohrdanz, Wenwei Zheng, Mauro Maggioni and Cecilia Clementi. *Determination of reaction coordinates via locally scaled diffusion map.* The Journal of chemical physics, vol. 134, no. 12, page 03B624, 2011. (Cited in page 11.)

[Röthlisberger 2008] Daniela Röthlisberger, Olga Khersonsky, Andrew M. Wollacott, Lin Jiang, Jason DeChancie, Jamie Betker, Jasmine L. Gallaher, Eric A. Althoff, Alexandre Zanghellini, Orly Dym, Shira Albeck, Kendall N. Houk, Dan S. Tawfik and David Baker. *Kemp elimination catalysts by computational enzyme design.* Nature, vol. 453, no. 7192, pages 190–195, May 2008. (Cited in page 22.)

[Roux 1999] Benoît Roux and Thomas Simonson. *Implicit solvent models.* Biophysical Chemistry, vol. 78, no. 1–2, pages 1–20, April 1999. (Cited in page 14.)

[Rudnick-Cohen 2015] Eliot S. Rudnick-Cohen, Shapour Azarm and Jeffrey Herrmann. *Multi-Objective Design and Path Planning Optimization of Unmanned Aerial Vehicles (UAVs).* In Proc. 16th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference. American Institute of Aeronautics and Astronautics, 2015. DOI: 10.2514/6.2015-2322. (Cited in pages 63 and 103.)

[Sánchez 2003] Gildardo Sánchez and Jean-Claude Latombe. *A Single-Query Bi-Directional Probabilistic Roadmap Planner with Lazy Collision Checking.* In Raymond Austin Jarvis and Alexander Zelinsky, editeurs, Robotics Research, volume 6, pages 403–417. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. (Cited in page 18.)

[Satoh 2006] Daisuke Satoh, Kentaro Shimizu, Shugo Nakamura and Tohru Terada. *Folding free-energy landscape of a 10-residue mini-protein, chignolin.* FEBS Letters, vol. 580, no. 14, pages 3422–3426, June 2006. (Cited in page 59.)

[Schön 2009] J. C. Schön and M. Jansen. *Prediction, determination and validation of phase diagrams via the global study of energy landscapes.* International Journal of Materials Research, vol. 100, no. 2, pages 135–152, February 2009. (Cited in page 12.)

[Scott 1966] Roy A. Scott. *Conformational Analysis of Macromolecules. II. The Rotational Isomeric States of the Normal Hydrocarbons.* The Journal of Chemical Physics, vol. 44, no. 8, page 3054, 1966. (Cited in pages 10 and 92.)

[Siegel 2010] Justin B. Siegel, Alexandre Zanghellini, Helena M. Lovick, Gert Kiss, Abigail R. Lambert, Jennifer L. St Clair, Jasmine L. Gallaher, Donald Hilvert, Michael H. Gelb, Barry L. Stoddard, Kendall N. Houk, Forrest E.

Michael and David Baker. *Computational design of an enzyme catalyst for a stereoselective bimolecular Diels-Alder reaction.* Science (New York, N.Y.), vol. 329, no. 5989, pages 309–313, July 2010. (Cited in page 22.)

[Siméon 2000] T. Siméon, J.-P. Laumond and C. Nissoux. *Visibility-based probabilistic roadmaps for motion planning.* Advanced Robotics, vol. 14, no. 6, pages 477–493, January 2000. (Cited in page 18.)

[Siméon 2004] Thierry Siméon, Jean-Paul Laumond, Juan Cortés and Anis Sahbani. *Manipulation Planning with Probabilistic Roadmaps.* The International Journal of Robotics Research, vol. 23, no. 7-8, pages 729–746, August 2004. (Cited in pages 64 and 104.)

[Simons 1999] K. T. Simons, I. Ruczinski, C. Kooperberg, B. A. Fox, C. Bystroff and D. Baker. *Improved recognition of native-like protein structures using a combination of sequence-dependent and sequence-independent features of proteins.* Proteins, vol. 34, no. 1, pages 82–95, January 1999. (Cited in page 14.)

[Singh 1999] A. P. Singh, J. C. Latombe and D. L. Brutlag. *A motion planning approach to flexible ligand binding.* Proceedings / ... International Conference on Intelligent Systems for Molecular Biology ; ISMB. International Conference on Intelligent Systems for Molecular Biology, pages 252–261, 1999. (Cited in page 18.)

[Soto C 2008] Soto C and Estrada LD. *Protein misfolding and neurodegeneration.* Archives of Neurology, vol. 65, no. 2, pages 184–189, February 2008. (Cited in pages 8 and 91.)

[Spong 2005] Mark W. Spong, Seth Hutchinson and M. Vidyasagar. Robot Modeling and Control. John Wiley & Sons, Hoboken, NJ, 2005. (Cited in page 9.)

[Sterpone 2014] Fabio Sterpone, Simone Melchionna, Pierre Tuffery, Samuela Pasquali, Normand Mousseau, Tristan Cragnolini, Yassmine Chebaro, Jean-Francois St-Pierre, Maria Kalimeri, Alessandro Barducci, Yoann Laurin, Alex Tek, Marc Baaden, Phuong Hoang Nguyen and Philippe Derreumaux. *The OPEP protein model: from single molecules, amyloid formation, crowding and hydrodynamics to DNA/RNA systems.* Chemical Society Reviews, vol. 43, no. 13, pages 4871–4893, July 2014. (Cited in page 12.)

[Suan Li 2012] Mai Suan Li and Binh Khanh Mai. *Steered molecular dynamics-a promising tool for drug design.* Current Bioinformatics, vol. 7, no. 4, pages 342–351, 2012. (Cited in pages 16 and 93.)

[Suárez 2008] María Suárez, Pablo Tortosa, Javier Carrera and Alfonso Jaramillo. *Pareto optimization in computational protein design with multiple objectives.* Journal of Computational Chemistry, vol. 29, no. 16, pages 2704–2711, December 2008. (Cited in page 25.)

[Suárez 2009] María Suárez and Alfonso Jaramillo. *Challenges in the computational design of proteins.* Journal of the Royal Society Interface, vol. 6, no. Suppl 4, pages S477–S491, August 2009. (Cited in page 25.)

[Suárez 2010] María Suárez, Pablo Tortosa, Maria M. Garcia-Mira, David Rodríguez-Larrea, Raquel Godoy-Ruiz, Beatriz Ibarra-Molero, Jose M. Sanchez-Ruiz and Alfonso Jaramillo. *Using multi-objective computational design to extend protein promiscuity.* Biophysical Chemistry, vol. 147, no. 1–2, pages 13–19, March 2010. (Cited in page 25.)

[Sucan 2012] Ioan Sucan and Lydia E. Kavraki. *A sampling-based tree planner for systems with complex dynamics.* Robotics, IEEE Transactions on, vol. 28, no. 1, pages 116–131, 2012. (Cited in pages 21, 48, 51, 52, 95, and 101.)

[Sugita 1999] Yuji Sugita and Yuko Okamoto. *Replica-exchange molecular dynamics method for protein folding.* Chemical Physics Letters, vol. 314, no. 1–2, pages 141–151, November 1999. (Cited in pages 16 and 93.)

[Swendsen 1986] Robert H. Swendsen and Jian-Sheng Wang. *Replica Monte Carlo Simulation of Spin-Glasses.* Physical Review Letters, vol. 57, no. 21, pages 2607–2609, November 1986. (Cited in page 17.)

[Taketomi 1975] Hiroshi Taketomi, Yuzo Ueda and Nobuhiro Gō. *Studies on protein folding, unfolding and fluctuations by computer simulation.* International journal of peptide and protein research, vol. 7, no. 6, pages 445–459, 1975. (Cited in page 11.)

[Tama 2001] F. Tama and Y. H. Sanejouand. *Conformational change of proteins arising from normal mode calculations.* Protein Engineering, vol. 14, no. 1, pages 1–6, January 2001. (Cited in page 11.)

[Tenenbaum 2000] Joshua B. Tenenbaum, Vin de Silva and John C. Langford. *A Global Geometric Framework for Nonlinear Dimensionality Reduction.* Science, vol. 290, no. 5500, pages 2319–2323, December 2000. (Cited in pages 11 and 92.)

[Thomas 2005] Shawna Thomas, Guang Song and Nancy M. Amato. *Protein folding by motion planning.* Physical Biology, vol. 2, no. 4, pages S148–155, November 2005. (Cited in page 18.)

[Thomas 2007] Shawna Thomas, Xinyu Tang, Lydia Tapia and Nancy M. Amato. *Simulating protein motions with rigidity analysis.* Journal of Computational Biology: A Journal of Computational Molecular Cell Biology, vol. 14, no. 6, pages 839–855, August 2007. (Cited in page 18.)

[Thomas 2013] Shawna Thomas, Chinwe Ekenna, Hsin-Yi Yeh and Nancy M. Amato. *Rigidity analysis for protein motion and folding core identification.* In

Proc. AAAI Workshop on Artificial Intelligence and Robotics Methods in Computational Biology, 2013. (Cited in pages 10 and 92.)

[Torrie 1977] G.M. Torrie and J.P. Valleau. *Nonphysical sampling distributions in Monte Carlo free-energy estimation: Umbrella sampling.* Journal of Computational Physics, vol. 23, no. 2, pages 187–199, February 1977. (Cited in pages 17 and 94.)

[Touw 2015] W. G. Touw, C. Baakman, J. Black, T. A. H. te Beek, E. Krieger, R. P. Joosten and G. Vriend. *A series of PDB-related databanks for everyday needs.* Nucleic Acids Research, vol. 43, no. D1, pages D364–D368, January 2015. (Cited in page 78.)

[Tozzini 2005] Valentina Tozzini. *Coarse-grained models for proteins.* Current Opinion in Structural Biology, vol. 15, no. 2, pages 144–150, April 2005. (Cited in page 11.)

[Traoré 2013] Seydou Traoré, David Allouche, Isabelle André, Simon de Givry, George Katsirelos, Thomas Schiex and Sophie Barbe. *A new framework for computational protein design through cost function network optimization.* Bioinformatics (Oxford, England), vol. 29, no. 17, pages 2129–2136, September 2013. (Cited in pages 24 and 97.)

[Ulmschneider 2003] Jakob P. Ulmschneider and William L. Jorgensen. *Monte Carlo backbone sampling for polypeptides with variable bond angles and dihedral angles using concerted rotations and a Gaussian bias.* The Journal of Chemical Physics, vol. 118, no. 9, page 4261, 2003. (Cited in page 41.)

[Ulmschneider 2004] Jakob P. Ulmschneider and William L. Jorgensen. *Polypeptide Folding Using Monte Carlo Sampling, Concerted Rotation, and Continuum Solvation.* Journal of the American Chemical Society, vol. 126, no. 6, pages 1849–1857, February 2004. (Cited in page 28.)

[Unger 1993] Ron Unger and John Moult. *Finding the lowest free energy conformation of a protein is an NP-hard problem: Proof and implications.* Bulletin of Mathematical Biology, vol. 55, no. 6, pages 1183–1198, November 1993. (Cited in page 11.)

[Van Der Maaten 2009] Laurens Van Der Maaten, Eric Postma and Jaap Van den Herik. *Dimensionality reduction: a comparative.* J Mach Learn Res, vol. 10, pages 66–71, 2009. (Cited in pages 10 and 92.)

[Voigt 2000] Christopher A. Voigt, D. Benjamin Gordon and Stephen L. Mayo. *Trading accuracy for speed: a quantitative comparison of search algorithms in protein sequence design.* Journal of Molecular Biology, vol. 299, no. 3, pages 789–803, June 2000. (Cited in pages 24 and 97.)

[Wales 1997] David J. Wales and Jonathan P. K. Doye. *Global Optimization by Basin-Hopping and the Lowest Energy Structures of Lennard-Jones Clusters Containing up to 110 Atoms.* The Journal of Physical Chemistry A, vol. 101, no. 28, pages 5111–5116, July 1997. (Cited in pages 17 and 94.)

[Wales 2004] David Wales. Energy Landscapes. Cambridge University Press, Cambridge, 2004. (Cited in pages 12 and 92.)

[Weise 2009] Thomas Weise. *Global optimization algorithms-theory and application.* Self-Published„ pages 25–26, 2009. (Cited in pages 25 and 97.)

[Wernisch 2000] Lorenz Wernisch, Stéphanie Hery and Shoshana J. Wodak. *Automatic protein design with all atom force-fields by exact and heuristic optimization.* Journal of Molecular Biology, vol. 301, no. 3, pages 713–736, August 2000. (Cited in page 24.)

[Wilmarth 1999] S. A. Wilmarth, N. M. Amato and P. F. Stiller. *MAPRM: a probabilistic roadmap planner with sampling on the medial axis of the free space.* In 1999 IEEE International Conference on Robotics and Automation, 1999. Proceedings, volume 2, pages 1024–1031 vol.2, 1999. (Cited in page 18.)

[Wu 1999a] Minghong G. Wu and Michael W. Deem. *Analytical Rebridging Monte Carlo: Application to cis/trans Isomerization in Proline-Containing, Cyclic Peptides.* The Journal of Chemical Physics, vol. 111, no. 14, page 6625, 1999. arXiv: physics/9904057. (Cited in page 31.)

[Wu 1999b] Minghong G. Wu and Michael W. Deem. *Efficient Monte Carlo methods for cyclic peptides.* Molecular Physics, vol. 97, no. 4, pages 559–580, 1999. (Cited in pages 31 and 34.)

[Yosef 2009] Eliyahu Yosef, Regina Politi, Mee H. Choi and Julia M. Shifman. *Computational design of calmodulin mutants with up to 900-fold increase in binding specificity.* Journal of Molecular Biology, vol. 385, no. 5, pages 1470–1480, February 2009. (Cited in page 22.)

[Yue 1995] Kaizhi Yue, Klaus M. Fiebig, Paul D. Thomas, Hue Sun Chan, Eugene I. Shakhnovich and Ken A. Dill. *A test of lattice protein folding algorithms.* Proceedings of the National Academy of Sciences, vol. 92, no. 1, pages 325–329, 1995. (Cited in page 11.)

[Zacharias 2003] Martin Zacharias. *Protein–protein docking with a reduced protein model accounting for side-chain flexibility.* Protein Science : A Publication of the Protein Society, vol. 12, no. 6, pages 1271–1282, June 2003. (Cited in page 12.)

[Zhang 2002] Yang Zhang, Daisuke Kihara and Jeffrey Skolnick. *Local energy landscape flattening: Parallel hyperbolic Monte Carlo sampling of protein folding.*

Proteins: Structure, Function, and Genetics, vol. 48, no. 2, pages 192–201, August 2002. (Cited in pages 17 and 94.)

[Zhang 2005] Wei Zhang, Chun Wu and Yong Duan. *Convergence of replica exchange molecular dynamics.* The Journal of Chemical Physics, vol. 123, no. 15, page 154105, October 2005. (Cited in page 16.)

[Zimmerman 2015] Maxwell I. Zimmerman and Gregory R. Bowman. *FAST Conformational Searches by Balancing Exploration/Exploitation Trade-Offs.* Journal of Chemical Theory and Computation, November 2015. (Cited in pages 48 and 51.)

Abstract: The ability to design proteins with specific properties would yield great progress in pharmacology and bio-technologies. Methods to design proteins have been developed since a few decades and some relevant achievements have been made including *de novo* protein design. Yet, current approaches suffer some serious limitations. By not taking protein's backbone motions into account, they fail at capturing some of the properties of the candidate design and cannot guarantee that the solution will in fact be stable for the goal conformation. Besides, although multi-states design methods have been proposed, they do not guarantee that a feasible trajectory between those states exists, which means that design problem involving state transitions are out of reach of the current methods. This thesis investigates how robotics-inspired algorithms can be used to efficiently explore the conformational landscape of a protein aiming to enhance protein design methods by introducing additional backbone flexibility. This work also provides first milestones towards protein motion design.
Keywords: path planning, protein design, structural biology, robotics, sampling-based algorithms, computational biology