NOVEL COMPUTATIONAL APPROACHES FOR PROTEIN STRUCTURE PREDICTION AND OPTIMIZATION

Università degli Studi di Milano-Bicocca
Dipartimento di Informatica Sistemistica e Comunicazione
Supervisor:
Prof. Giancarlo Mauri
Ph.D candidate:
Andrea Gaetano Citrolo

The reasonable man adapts himself to the world;
the unreasonable one persists in trying to adapt the world to himself.
Therefore all progress depends on the unreasonable man.
— George Bernard Shaw

To my parents...

Abstract

For many important classes of biomolecules such as RNA and proteins, a direct relationship exists between structure and function. On the contrary the relationships between genomic sequences and molecular structures are still poorly understood. The determination of the three dimensional structure of biomolecules on a genome-scale is hence one of the major challenges in modern biology. Indeed, today genomic data are easily achievable, thanks to next generation sequencing technology, while structural data are still obtained through complex experimental protocols. As a result, the disproportion between the available amount of genomic and structural data limits the progress in several fields such as drug discovery and synthetic biology.

The use of computational methods and mathematical optimization in structural biology is fundamental to reduce the amount of data required from experiments speeding up experimental protocols and to define *in silico* protocols for the prediction of three dimensional structures. This thesis introduces novel heuristic approaches to tackle two important problems in structural biology: the *protein structure prediction* (PSP) and the *molecular distance geometry* (MDG) problem. Both these problems are known to have a complex combinatorial structure and are classified as NP-hard. Therefore the proposed approaches are based on *stochastic optimization heuristics* (SOH), which provide a powerful framework to tackle complex combinatorial problems that do not allow for exact approaches.

The PSP problem have been treated in the simplified representation provided by the hydrophobic polar (HP) model; a new perturbation strategy has been introduced to mimic off-lattice approaches and to provide a complementary benchmark to the existing move sets.

Two heuristics, based on the principle of *local landscape mapping*, have been tested on several benchmark instances both in combination with the new perturbation strategy and with standard move sets. The results show that one of the proposed heuristics outperforms state of the art methods on the majority of the considered instances. In the case of the MDG problem, results show that the proposed methodology is able to achieve a performance comparable to the state of the art and to overcome most limitations of the existing approaches.

Keywords: Protein Folding, Combinatorial Optimization, Heuristics.

Riassunto

La determinazione della struttura tridimensionale delle biomolecole su scala genomica è uno dei più importanti obbiettivi della biologia moderna con potenziali ricadute in differenti contesti applicativi che spaziano dalla farmacologia, alla biologia sintetica. Il ruolo dei metodi computazionali ed in particolare dei metodi di ottimizzazione in quest'ambito è fondamentale per l'interpretazione dei dati sperimentali. Inoltre nell'ultimo decennio la predizione computazionale della struttura di importanti classi di biomolecole come RNA e proteine è diventata una prospettiva concreta.
Questa tesi presenta due nuovi metodi di ottimizzazione stocastica progettati rispettivamente per il problema della predizione della struttura delle proteine nel modello idrofobico polare e per il problema della ricostruzione della struttura da dati NMR. Il primo problema consiste nel trovare un assegnamento in un reticolo di una stringa binaria tale da minimizzare una data funzione di costo e senza violare un insieme di vincoli. Il secondo, consiste nel identificare una disposizione di atomi nello spazio tridimensionale che rispetti un insieme di vincoli di distanza. Entrambi questi problemi sono rilevanti dal punto di vista computazionale inquanto è stata dimostrata la NP-completezza del problema di decisione associato. Pertanto essi rappresentano un ottimo banco di prova per le euristiche di ottimizzazione stocastica.
Nel caso della predizione della struttura nel modello idrofobico polare, i risultati ottenuti su una serie di instanze di *benchmark* mostrano che la strategia proposta può essere adattata a differenti modelli di rappresentazione migliorando in alcuni casi la performance rispetto allo stato dell'arte. Per quanto riguarda la ricostruzione di strutture da dati NMR, i risultati, per quanto ancora in fase preliminare, suggeriscono che il metodo proposto sia in grado di raggiungere l'accuratezza richiesta dall'applicazione offrendo altresi numerosi vantaggi in termini di applicabilità rispetto agli approcci esistenti.

Parole chiave: Ripiegamento Proteico, Ottimizzazione Combinatoria, Euristiche.

Contents

Abstract (English/Italian) **i**

List of figures **ix**

List of tables **xi**

1 Introduction **1**
1.1 Context and motivations 1
1.2 Contribution 3
1.3 Structure of the thesis 4

2 Stochastic optimization heuristics **5**
2.1 Optimization problems 5
2.1.1 Continuous optimization problems 6
2.1.2 Combinatorial optimization problems 6
2.1.3 Efficiency in optimization 7
2.1.4 Global optimization problems 8
2.1.5 Implicit constraints 8
2.2 Applicability and general properties 9
2.2.1 The No Free Lunch Theorems 10
2.3 An overview of SOH meta-heuristic 12
2.3.1 Local search based 12
2.3.2 Monte Carlo methods 14
2.3.3 Population based methods 18

3 Biological background **23**
3.1 Proteins 23
3.2 Inter-atomic forces and their modeling 26
3.2.1 Bonded interactions 27
3.2.2 Electrostatic interactions 28
3.2.3 The hydrophobic effect 30
3.2.4 Statistical Potentials 31
3.3 Energetics of protein folding 33

3.3.1 The folding process: from the Levinthal's paradox to the funnel theory . . . 34
3.4 Nuclear Magnetic Resonance . . . 35

4 Problems definition and state of the art **39**
4.1 The protein structure prediction problem . . . 39
4.2 Off-lattice approaches to PSP . . . 40
4.2.1 The FA strategy for PSP . . . 40
4.3 Bravais Lattice . . . 41
4.4 The hydrophobic polar model . . . 43
4.4.1 Move sets . . . 45
4.4.2 The structure of the search space . . . 48
4.4.3 State of the art of the HP model . . . 49
4.5 The MDG problem . . . 51
4.5.1 State of the art of the approaches for the MDG problem . . . 52

5 The local landscape mapping strategy **55**
5.1 A fragment assembly-like representation in the HP model . . . 55
5.2 Fundamental ideas . . . 58
5.3 The memory based biasing strategy . . . 58
5.3.1 The pheromone structure . . . 60
5.3.2 Parameterization and auxiliary procedures . . . 64
5.3.3 Computational results . . . 64
5.4 The local search based biasing strategy . . . 67
5.4.1 The putative hydrophobic core . . . 68
5.4.2 The local search method . . . 69
5.4.3 The optimization step and parameterization . . . 69
5.4.4 Computational results . . . 70

6 The evolutionary springs swarm method **75**
6.1 Fundamental ideas . . . 75
6.2 The objective function . . . 76
6.3 The algorithm . . . 77
6.3.1 GA-layer . . . 77
6.3.2 PSO-layer . . . 80
6.4 Chirality . . . 83
6.5 Parallelization . . . 84
6.6 Results . . . 86
6.6.1 Parameterization . . . 86
6.6.2 Reconstructing the structure of real proteins . . . 88

7 Conclusions and Future Works **91**

A Appendix A **95**
A.1 Benchmark HP sequences used in this thesis 95

Bibliography **116**

Curriculum Vitae **117**

List of Figures

3.1 Primary structure 24
3.2 Secondary structure 25
3.3 Tertiary structure 26
3.4 The hydrophobic effect 32
3.5 The funnel model of protein folding 35

4.1 The fragment assembly strategy 41

5.1 Allowed fragments for the first triplet 56
5.2 Pheromone model for the FA-like perturbation system. 62
5.3 Pheromone-based neighbors selection for the FA-like perturbation system. 63
5.4 Best results on biological instances. 71
5.5 Results on large instances in the FCC. 73

6.1 ESSM overview 77
6.2 Spring-like behavior of restraints in ESSM 81
6.3 The aggregate attractor 82
6.4 Effect of the parameter $v_{\mathtt{MAX}}$ 87
6.5 Effect of the parameter α 87
6.6 Results of ESSM for the MDG problem: 3-peptide 89
6.7 Results of ESSM for the MDG problem: real proteins 90

List of Tables

5.1 Parameters lists and settings 64
5.2 Results of the LLM_{mem} method: fragment assembly-like 65
5.3 Results of the LLM_{mem} method: move sets 66
5.4 Results of the LLM_{LS} method: CBC lattice 1 70
5.5 Results of the LLM_{LS} method: CBC lattice 2 72
5.6 Results of the LLM_{LS} method: FCC lattice 72

6.1 Results of the ESSM: data from real proteins 90

A.1 Harvard instances 95
A.2 Instances from real proteins 96
A.3 F90 instances 96
A.4 S instances 97
A.5 F180 instances 97
A.6 R instances 98

1 Introduction

In this chapter the context of research of this thesis is introduced, motivations and the contribution of this work to the research field are provided along with the structure of the thesis.

1.1 Context and motivations

In the last forty years, the computational techniques have gained a crucial role in biology, spanning from sequence alignment algorithms to inference methods and simulation tools. This has greatly enhanced our understanding of many biological processes, allowing the completion of ambitious task such the *Human Genome Project* [1]. Today high-throughput methods for genome sequencing and analysis allow the completion of a genome analysis in few hours, on-line tools allow the comparison between genes from thousand of different species, and databases have been created to organize all this information and to make it accessible in a meaningful way. Although these results are impressive, most of the information that it is possible to extract and easily process is about biological sequences. This is often referred to as one-dimensional (1D) information. Nevertheless, a significant part of the information needed to understand biological systems and to predict or alter their behavior is the three-dimensional (3D) information. This regards, in particular, two classes of biological macromolecules: proteins and ribonucleic acids (RNAs). The elucidation of the relationship between 1D and 3D information for these macromolecules is so critical that it is sometimes referred to as: *the second half of genetic code*. Unfortunately this relationship is far from trivial and despite the considerable efforts and progress achieved so far, it still remains unclear. In this thesis the discussion focuses on proteins (see [2, 3, 4] for an extended treatment of the RNAs case),a key component of life involved in almost every cellular process. The availability of many different shapes and sizes allows them to cover roles as different as catalysis, signaling, energy production and fiber formation [5]. Due to their active role in life processes takeover, proteins are also the primary therapeutic target. Beyond their biological relevance, proteins are also employed in industrial

processes, in particular in the food industry and pharmaceutical industry. Moreover proteins production and engineering is becoming a field of great interest, promising a broad sets of new applications that spans from fine chemicals to digital memories [6, 7, 8]. Despite this, our ability to determine the structure adopted by a specific protein sequence with atomic accuracy still relies on experimental techniques, in particular crystallographic techniques [9] and nuclear magnetic resonance (NMR) [10]. Both techniques produce rough data and algorithms are needed in order to determine the molecular structure. Today, efficient algorithms for fitting the electron density maps produced by crystallographic techniques are available [11, 12, 13, 14, 15], and most of the improvements in this field rely on the development of new protocols for purification and crystallization, or on the use of new experimental methodologies to reconstruct the phase space [16, 17, 18, 19, 20, 21]. On the other hand, in order to obtain a 3D-structure from NMR, the solution of an instance of the molecular distance geometry (MDG) problem has to be found. This is a difficult optimization problem and existing algorithmic strategy are not considered completely satisfactory. For this reason, the definition of improved methods for MDG problem has been object of intense research in recent years [22, 23, 24, 25, 26].

An alternative to experimental techniques, could be the prediction of protein structures based only on the amino-acidic sequence. Indeed, for reason explained in Chapter 4, the prediction of protein structure can be formulated as a search problem and both simulation and optimization algorithms can be used to perform the search. This is a very ambitious challenge and, although many important steps have been made toward its solution, it is still an open problem. The efforts of the scientific community in this direction have increased significantly in the last 20 years; in particular, after the introduction of high-throughput methods for genome sequencing and analysis, that have pushed the sequence discovery rate further beyond the experimental structure determination rate, resulting in an enormous gap between known sequences and known structures. The high number of research groups and the interest of pharmaceutical companies in this research led to the creation of the Critical Assessment on Protein Structure Prediction (CASP) [27], a biennial event held since 1995 that includes both theoretical and experimental scientists. This event gave to the researchers the opportunity to compare the accuracy of different prediction algorithms on a set of benchmark protein sequences with a known but unpublished structure, leading to a significant improvement in the accuracy of the prediction and to the development of better metrics to assess the quality of the models. According to CASP results [28, 29], today the most successful computational approaches for protein structure prediction (PSP) exploit the evolutionary relations between proteins. These methods allow the prediction of a protein structure only when a closely related sequence (homologous) with known structure is available and their accuracy strongly depends on the quality of the sequence alignment between the *target* sequence and the *template* sequences. Nevertheless, almost a third of the known sequences has no known homologous [30] and it is expected

that a significant number of folds (general structural organization shared between different protein families) have not yet been discovered or characterized [31]. The development of a fast and reliable method to predict a protein structure using only the information contained in the sequence is therefore extremely valuable in computational structural biology.

1.2 Contribution

The focus of this thesis is on the development of improved *stochastic optimization heuristics* (SOH) for *protein structure prediction* (PSP) problem and for the MDG problem. Although both problems are linked to the same applicative contest and the proposed approaches are rooted in the common ground of SOH, they differ significantly in the mathematical formalization and consequently in the proposed heuristics. For this reason, in this thesis, they will be discussed separately.

In the case of, PSP the hydrophobic polar (HP) model [32] have been taken into account since it offers a simplified representation. Consequently, it allows the evaluation of the the performance of new methods using a reasonable amount of computational resources. The PSP problem in the HP model consists in finding a self avoiding lattice chain that maximizes the number of lattice contacts between positions with a specific label in a given string. Although simple in terms of representation, the decision problem associated to PSP in the HP model has been proved to be NP-complete [33]. Due to the relevance of this problem in the field of structural biology, a great number of SOHs have been proposed for HP-PSP [34, 35, 36, 37, 38, 39, 40, 41]. The first contribution of this thesis, is the definition of a new perturbation system that mimics off-lattice approaches and to provide a complementary benchmark to the existing move sets.

Moreover, a new optimization strategy for PSP is introduced, based on the idea of combining a variable neighborhood structure with the rejection sampling scheme of a typical Monte Carlo method. This strategy includes some of the main concepts introduced in recent literature such as the use of memory support structures [34, 38, 37, 41, 39, 40] and specialized local search procedures [42, 34]. Two different implementations of this strategy have been developed: the first is based on an auxiliary memory structure [43, 44], the second (manuscript in preparation) is based on a specialized local search procedure. The main novelty of the memory based approach is that it focuses on the structural level instead of the energy level. Results suggest that this method is more suitable to threat the problem in the new representation systems with respect to other well established SOHs. In the case of the local search based approach, the main novelty is represented by the definition of problem-related variable structure that act as a target for the specialized local search, providing an effective diversification strategy. Results show that this method outperforms state of the art approaches on the majority of the benchmark instances considered in this thesis.

The MDG problem consists in reconstructing the three dimensional structure of a biomolecule using a sparse set of distance restraints obtained through a nuclear magnetic resonance (NMR) experiment. The decision problem associated to the MDG problem has been proved to be NP-complete [45]. Although NP-hard problems are the natural field of application for SOHs, to the best of our knowledge, the approach proposed in this thesis [46] is the first attempt to solve the MDG problem by exploiting evolutionary techniques alone. The underlying idea of the proposed method is that of using the constraints arising from experimental data as a springs system, this system is integrated in a hybrid genetic algorithm-particle swarm optimization heuristics in order to perform the search for satisfactory structures. Results obtained on synthetic data show that the method is able to reconstruct the protein structures with atomic resolution.

1.3 Structure of the thesis

This thesis is organized as follows:

- Chapter 2 provides a generic definition of continuous and discrete optimization problems and introduces the main characteristic of stochatic optimization techniques. Applicability condition are also taken into account along with some limitations inherent to these methods. In addition, an overview of the most successful heuristics is provided.
- Chapter 3 gives a quick overview of the biological background. In particular, the chemico-physical properties of the protein molecules are described and some thermodynamical aspects of folding process are mentioned. Moreover, principles of the NMR technique are discussed to give a better understanding of the data used as input to solve the MDG problem.
- Chapter 4 introduces a formal definition of both PSP and MDG problems along with an overview of the state art approaches and representations.
- Chapter 5 describes the proposed *Local Landscape Mapping* method for PSP problem in the HP model. The performance of the method is evaluated through a comparison with state of the art heuristics on several benchmark instances and representation systems.
- Chapter 6 describes the proposed *Springs Swarm Method* for the MDG problem, discusses the results achieved on synthetic data from real protein structures and provides a qualitative comparison with state of the art methods.
- Finally, Chapter 7 includes conclusions about the whole work and some possible directions for the extension of the presented approaches.

2 Stochastic optimization heuristics

As anticipated in the previous chapter, the work presented in this thesis is rooted in the context of stochastic optimization heuristics (SOH). This chapter provides an overview of some of the major families of SOH and introduces some general concepts on their applicability along with basic informations about optimization problems and the notations that will be used throughout this thesis. An extended treatment of this SOH would exceed by far the scope of this thesis. For this reason, an effort has been made to include only concepts and techniques essential to understand how the SOH can be used to tackle complex problems such as protein structure prediction in the HP model and molecular distance geometry problem.

2.1 Optimization problems

An optimization problem can be informally defined as the problem of finding the most desirable assignment of a variable defined in some space, called the *search space*, according to some evaluation (or objective) function. A commonly used classification of optimization problems is based on the structure of search space. In particular an optimization problem is said to be *continuous* if the search space can be expressed as a combination of continuous variables; it is said to be *discrete*, or *combinatorial*, if the search space can be expressed as a combination of discrete variables; while it is said to be a *mixed integer* problem if the search space is a mixture of discrete and continuous variables. In this thesis only continuous and combinatorial problem will be considered.

2.1.1 Continuous optimization problems

A widely used [47] formal definition of a continuous optimization problem $\mathcal{P}$ is the following:

$$
\begin{aligned}
&\textit{find } x^* \in \left\{ \underset{x \in \mathcal{A}_p}{\arg\min}\, f_0(x) \right\}, \\
&s.t. \\
&\quad f_i(x) \leq 0, \qquad i \in \{1, \ldots, m\}, \\
&\quad h_j(x) = 0, \qquad i \in \{m+1, \ldots, n\}, \\
&f_0 : \mathcal{A}_0 \to \mathbb{R},\ f_i : \mathcal{A}_i \to \mathbb{R},\ h_j : \mathcal{A}_j \to \mathbb{R},\ \mathcal{A}_p = \left\{ \bigcap_{k=0}^{n} \mathcal{A}_k \right\}.
\end{aligned}
\tag{2.1}
$$

In the equation above, x denotes the optimization variable defined over the search space $\mathcal{A}_p$, the function f_0 is the objective function that is used to measure the desirability of a specific assignment of x, x^* denotes a solution of $\mathcal{P}$, while f_i and h_j are called respectively inequality and equality constraints functions used to restrict $\mathcal{A}_p$ to the so called feasible region of the search space. It is important to notice that the definition in Eq.(2.1), also referred to as standard form of optimization problems, includes also maximization problems since they can be expressed by switching the sign of f_0. A continuous optimization problem for which it holds $f_0(x) = k$, $\forall x \in \mathcal{A}_p$ is called *feasibility* problem: these are problems in which the objective is to find a value of x that satisfies all the constraints. Clearly, for some optimization problem no solution x^* can be found. This can happen for one of the following two reasons:

1. there is no assignment of $x \in \mathcal{A}_p$ that satisfies all the constraints, and the problem is said to be *infeasible*;
2. the optimal value of f_0 is $-\infty$ as a consequence x^* is undefined, and the problem is said to be *unbounded below*.

2.1.2 Combinatorial optimization problems

For what concerns combinatorial optimization problems, this thesis follows the definition given by Ausiello in [48]. According to this formalism, a discrete optimization problem $\mathcal{P}$ is represented with a quadruple $(I_p, SOL_p, m_p, goal_p)$, where:

1. I_p is the set of instances of $\mathcal{P}$;
2. $SOL_p : I_p \to \mathbb{N}^n$ is a function that associates to any input instance $\iota \in I_p$ the set of feasible assignments of ι;
3. $m_p : \{I_p, \mathbb{N}^n\} \to \mathbb{N}$ is the measure function, corresponding to f_0 in the continuous

case, defined for pairs (ι, x) such that $\iota \in I_p$ and $x \in SOL_p(\iota)$;

4. $goal_p \in \{MIN, MAX\}$ specifies if $\mathcal{P}$ is a minimization or a maximization problem.

Using this formalism, the value $x^* \in SOL_p$ is a solution for the instance ι of $\mathcal{P}$ if:

$$m_p(\iota, x^*) = goal_p\left\{v \mid v = m_p(\iota, z) \wedge z \in SOL_p(\iota)\right\}. \tag{2.2}$$

Since in many cases the adaptation of the evaluation function m_p to different instances of the problem $\mathcal{P}$ is trivial, in the remaining of this thesis the notation $m_p(x)$ will be used in place of $m_p(\iota, x)$. It is important to notice that the choice of a formalism to represent an optimization problem is arbitrary; therefore according to the context, different definitions can be used. For instance, it is not uncommon to define a discrete optimization problem using the standard form.

In this thesis, the notation O_p will be used, both for continuous and discrete optimization problems, to denote the solutions set whenever it exists.

2.1.3 Efficiency in optimization

Similarly to other fields in computer science and applied mathematics, the *efficiency* of the algorithms is a major concern in the optimization field. This means that, given a model of computation (i.e., the Turing machine), an algorithm A and a generic instance ι of an optimization problem $\mathcal{P}$, it is important to consider the amount of computational resources; namely, time and space (memory), required in order to achieve a solution for ι using algorithm A, which is called the *computational cost* of A. In the case of discrete problems, it is always possible to define a function $|| : I_p \to \mathbb{N}$, such that $|\iota|$ measures the size of the instance ι of $\mathcal{P}$ (i.e., the bit length of an assignment x). The algorithm A is considered efficient if it has polynomial complexity both in time and space with respect to $|\iota|$; that is the upper bound of the computational cost of A over all the possible instances of $\mathcal{P}$ is at most a polynomial function of $|\iota|$. In the case of continuous problems, giving a definition of *efficiency* is not trivial [49, 50]. For the purpose of this thesis, it is enough to say that A is an efficient algorithm for a continuous optimization problem $\mathcal{P}$, if, on average, it is able to find $z \in \mathcal{A}_p$ such that: $f_0(z) - f_0(x^*) \leq \epsilon, \epsilon \in \mathbb{R}_+$ with a computational cost that has a polynomial dependence on both ϵ^{-1} and the dimensionality d of $\mathcal{A}_p$. Both continuous and discrete problems for which no efficient algorithm is known, are said to be *intractable*. Some of the sufficient conditions that determine tractability are known; consequently it is possible (but often not trivial) to know if an efficient algorithm is available to solve a given problem. For example, given an optimization problem $\mathcal{P}$ in standard form, if all the $\mathcal{A}_k$ are convex sets and the following inequality is satisfied by both the objective and constraints functions:

$$f(\theta x + (1-\theta)y) \leq \theta f(x) + (1-\theta) f(y), \qquad \theta \in [0,1],\ x, y \in \mathcal{A}_p \tag{2.3}$$

then $\mathcal{P}$ is said to be a convex problem and it can be solved by means of efficient techniques [51, 52]. Note that often problems that are not convex in their *naive* definition, might be reduced (mathematically manipulated) to an equivalent convex problem. Problems in standard form that are not reducible to a convex problem usually cannot be solved efficiently [53]. Similarly, in the discrete domain there is a class of optimization problems called NP-hard problems [54, 55, 56, 57], for which no efficient algorithm is known. Several problems of practical interest, such as the traveling salesman problem (TSP) [55], bin packing problem [56], boolean satisfiability [54] (SAT), molecular distance geometry (MDG) problem [45] and the PSP in the hydrophobic polar model (HP-PSP)[33], just to name but a few, fall in this class.

2.1.4 Global optimization problems

The main point that emerges from the previous section is that many optimization problems of practical interest can not be solved exactly in a reasonable amount of time. These problems are usually grouped under the name of *global optimization* (GO) problems, since, roughly speaking, they are characterized by the presence of two types of minimal points in the search space: global minima and local minima . A local minimum in a continuous optimization problem is an assignment y of x such that:

$$\begin{aligned} &\exists k \in R_+ : \quad f_0(y) \leq f_0(z) \quad \forall z \in \mathcal{N}, \\ &y \notin O_P, \mathcal{N}_y^k = \left\{ z, y \in \mathcal{A}_p | \, \|z - y\| < k \right\}. \end{aligned} \tag{2.4}$$

The set $\mathcal{N}_y^k$ is called k-*neighborhood* of y. An equivalent definition of local minimum can also be given for discrete optimization problems; in this case the neighborhood $\mathcal{N}_y^k$ of the assignment y is defined as:

$$\mathcal{N}_y^k = \left\{ z, y \in SOL_P(\iota) | \lambda(y,z) \leq k \right\}, k \in \mathbb{N}, \tag{2.5}$$

where $\lambda(y,z)$ is the minimum number of discrete operations (i.e., bit switch) required to transform y in z (discrete neighborhood relationships are symmetric only if the adopted discrete operations are symmetric). The maximal neighborhood $\mathcal{N}_y^{\hat{k}}$ in which the point y is the only minimal point is called *basin of attraction* of y. A huge number of approaches is available today to tackle global optimization problems; among them only SOHs will be discussed in some detail in this thesis. The reader interested in deterministic approaches to GO is referred to [58, 59, 60, 61, 62, 63, 64] and references therein.

2.1.5 Implicit constraints

In the previous section it has been assumed that the constraints of a generic GO problem can be handled explicitly and that the optimization methods evaluate only feasible

assignments. That is to say, some efficient algorithm exists to define the set $\mathcal{A}_p$ in the continuous case or to compute the function $SOL(\iota)$ in the discrete case. Nevertheless, for some optimization problems and depending on the chosen optimization method, the explicit handling of the constraints could be inefficient. In this situation, it is always possible to define a modified evaluation function f_p such that the set O_p is preserved and that unfeasible assignments are associated to an arbitrary poor value of f_p. For example, in the continuous case, this can be obtained with a function of the form:

$$f_p(x) = \begin{cases} f_0(x) & \text{if } x \text{ is feasible,} \\ \infty & \text{otherwise.} \end{cases} . \tag{2.6}$$

This strategy is often referred to as *implicit constrain treatment* and is commonly used by SOH to include constraints and to tackle feasibility problems.

2.2 Applicability and general properties

The SOH are a wide and heterogeneous group of methods for GO problems. Although different SOH methods define different strategies to explore the search space, their common feature is the use of a *biased random sampling*. This *bias* is designed to drive the search toward regions of the search space that exhibit good values of the objective function. Another general feature of SOH is that the average quality of the returned assignment x^t, measured between independent executions, increases as a function of t, where t denotes the computational resources allocated. This observation is supported by the formal proofs of asymptotic convergence obtained for most of these methods [65, 66, 67, 68], often these proofs adopt a probabilistic approach and come in the form:

$$\lim_{t \to +\infty} P_A\left(x^t \in O_p\right) = 1, \tag{2.7}$$

where $P_A\left(x^t \in O_p\right)$ is the probability that the SOH A returns a value in the solutions set of the target problem P using t resources; where t usually denotes a number of iterations of A, since all SOHs are iterative methods. In a comparative setting, the most appealing features of SOH with respect to deterministic approaches can be summarized in the following four points:

1. they allow a fine tuning of the amount of computational resources allocated to achieve a *satisfactory* assignment of x;
2. they can always perform the optimization using the natural definition of the problem;
3. they do not require lower bound estimate, derivatives, or the availability of other mathematical properties;

4. they allow a realistic modeling of the physical behavior of some systems.

Due to their flexibility, SOH have been intensively applied and occasionally introduced in different fields as automation [69], logistic [70], matter physics [71], finance [72], just to name but a few. The most important limitation of SOH is that they give no guarantee on the quality of the returned assignment of the search variable x. In other words, the value that is returned at the end of the execution of a SOH could be outside of O_p. Moreover, no information is achieved on the optimal value in terms of lower bounds or approximation constants. Another aspect, linked to the stochastic nature of these methods, that can be considered unpleasant in some contexts, is the lack of reproducibility of single executions, meaning that two identically parameterized execution of the same SOH on the same instance of an optimization problem often return a different assignment of x. For these reasons there are contexts in which the use of a deterministic GO approach should be preferred. This holds, in particular, if many information are available on the mathematical structure of the problem, if the dimensionality of the search space is low or when the evaluation of the objective function is extremely demanding. Anyhow, for many interesting real life problems, the only way to choose between the available approaches is to test them over a set of benchmark instances. For this reason, the use of SOH is often justified by a *good enough* reasoning. This means that for a given problem, a SOH is chosen if it is able to find values of x that satisfies some practical needs and no deterministic method is able to produce better or comparable results with the given amount of computational resources.

2.2.1 The No Free Lunch Theorems

The selection of a specific optimization heuristic for a given problem is itself challenging since it has been proved in a seminal work by Wolpert and Macready [73] that, at least for combinatorial problems, the performance of any couple of heuristics methods averaged over all the possible optimization problems is the same. This result was a milestone in the theoretical study of the performance of the optimization heuristics, leading to a series of implications that are collectively called *no free lunch* (NFL) theorems for optimization. NFL theorems give a probabilistic view of the relation between the problems space and the heuristic space. They are based on three important assumptions:

1. the search space and problems space are both finite;

2. the algorithms sample each assignment of the search variable at most once;

3. the probability distribution of the problems is uniform.

The first assumption holds for all discrete problems of practical interest, while the second can be imposed to SOH but it is misleading when considering the real performance of these class of algorithms. The formal definition of the main NFL theorem Eq. (2.8), considers the execution of an algorithm A as a trajectory $\mathbf{v}_t^x$ of *unique* assignments of x :

$$\sum_{\mathcal{P}} P(\mathbf{v}_t^y | t, A_1, m_p) = \sum_{\mathcal{P}} P(\mathbf{v}_t^y | t, A_2, m_p), \quad \forall A_1, A_2, t. \tag{2.8}$$

In the equation above, the sum is defined over the space of all optimization problems, t is a given number of functions evaluation, A_1 and A_2 are a generic couple of algorithms, $\mathbf{v}_t^y$ is the vector of the values of the objective function m_p associated to the trajectory $\mathbf{v}_t^x$. The most important implication of Eq.(2.8) is that it is not possible to define a *universal* optimization heuristic that outperform all the others independently of the considered problem. An interesting interpretation of the NFL is that the performance of a given algorithm depends on the alignment of that algorithm with the problem in the problems space. To understand this aspect, one can consider the probability that a specific algorithm A' produces a specific trajectory $d_t^{y'}$ (i.e., a trajectory with low values of m_p):

$$P(d_t^{y'} | t, A') = \sum_{\mathcal{P}} P(d_t^{y'} | t, A', m_p) P(m_p), \tag{2.9}$$

where $P(m_p)$ is the probability of a problem with evaluation function m_p. This sum can be seen as dot product between vectors in the problems space. Consequently, the optimization heuristics lies on a cone in the problem space surrounding the diagonal vector that represents the uniform probability over the problems. If a non-uniform probability distribution over the problem space is taken into account, i.e., we are interested in solving a specific family of problems, it is possible to design *specialized* algorithms that are aligned with the target family of problems. This is the reason why most of the applied research in SOH field is dedicated to the development of specialized variants of well known optimization schemes called *meta-heuristic*. This is also the main argument of this thesis, since both the proposed methods are specializations that combine several features of existing SOH in order to be maximally aligned with the family of problems corresponding to PSP and MDG. However, it is important to notice that this specialization does not imply that the resulting method will be useless in other context, indeed it is possible that apparently unrelated problems exhibit similar orientation in the problems space. Moreover the NFL theorems leave the door open to head to head min-max behavior between couple of algorithms; this means that algorithm A_1 can be superior of algorithm A_2 of a value k according to some performance measure on a certain set of problems, but for no other problem A_2 outperform A_1 of value equal or greater then k. Therefore, in a weak sense a A_1 is a better general purpose *black box* optimizer.

2.3 An overview of SOH meta-heuristic

A brief description of some important classes of SOHs is given in this section; most of these algorithms have been applied to the PSP or to the MDG problem, or are somehow ancestors of some of the specialized techniques that will be discussed in the next chapters. The notation used to describe each technique will reflect the context in which they have been introduced or applied in this thesis. Basing on the method used to generate new assignments during the optimization process, it is possible to divide SOH in two main classes: *neighborhood based* and *population based*. The neighborhood based SOH exploit a generic neighborhood definition such those in Eq. (2.4) and Eq.(2.5), to generate new assignments of x after that a first assignment has been obtained somehow, then some acceptance criterion is applied to choose between neighbor assignments and direct the search. Population based methods on the contrary exploit the possibility to create new assignments using some sort of *communication* (i.e., exchange of partial assignments, or shared memories) between a large number of initial assignments, representing *the population* itself. A second important distinction is between *constructive* and *perturbation-based* approaches. As the name suggests, a constructive approach *builds* assignments of x through the incremental extension of partial assignments (i.e., an assignment of x in a smaller instance of $\mathcal{P}$). Each extension step involves some kind of choice and can be used to bias the search toward new assignments by means of specific criteria and operators. On the contrary, a perturbation-based method proceeds modifying the value of complete assignments of x and then chooses whether to accept or not the resulting assignments as new starting points for the perturbation.

2.3.1 Local search based

A Local search (LS) method is one designed to find the closest optimal point with respect to a given starting point. Since, for many interesting optimization problems, a single optimal point exists, the number of local optimization algorithm is huge including, among the others, the simplex method, the steepest descent algorithm, the greedy algorithms, the Newton method and the hill climbing algorithm. Many SOH are based on the idea to combine a local search method with some strategy to prevent it from being trapped when a local minimum is found.

Random restart local search (RRLS) [74] is the simplest algorithm based on local search. Basically, it consist in restarting the local search algorithm from a random point every time that a minimum has been found. This algorithm achieve good results if local optima are uniformly distributed over the search space; however, for many problems of practical interest such as TSP and PSP problem, empirical observations [75, 76] indicate that interesting regions of the search space are grouped in clusters.

Iterated local search (ILS) [74, 77] is a powerful meta-heuristic, that have been successfully applied to several combinatorial problems [78, 79]. The original version of this heuristic [80] has been developed as a branch of Markov Chain Monte Carlo (MCMC) methods described below, but its modern interpretation focuses more on the definition of recursive neighborhood and sequential local search. The basic idea of ILS is to iteratively apply a stochastic perturbation to the locally optimal assignments found by means of a local search method. The algorithm starts from a random assignment x^0 of the search variable; then, each iteration is divided in two steps. In the first step , a local search method is applied to the current assignment x^t and a locally optimal assignment $\hat{x}^t$ is found. In the second step, a perturbation is applied to $\hat{x}^t$ to generate a new assignment x^{t+1}. The perturbation strategy is the critical aspect of ILS, since small perturbations could lead to re-sampling of the local optimum; while, on the other hand, large perturbations could result in a RRLS-like behavior. A more sophisticate ILS strategy exploits the search history [77] to bias the generation of x^{t+1}; for example, an archive of the best assignments found so far can be stored and used instead of $\hat{x}^t$ during the perturbation step.

Greedy randomized adaptive search procedures (GRASP) [81, 82] are yet another strategy aimed to generate good starting point for local search methods. In this case a randomized greedy heuristic is used to allow the generation of a large number of different starting assignments; then a local search method is applied to each of this starting assignments. GRASP is an iterative procedure consisting of two phases, a construction phase and a local search phase. In the construction phase an assignment is builded from scratch, adding one component at a time. At each step of the construction phase, the components that define the set of possible extension of the partial assignmet are ranked according to some greedy function and a number of the best-ranked components are included in a restricted candidate list; typical strategies of deriving the restricted candidate list are either to take the best g% of the components or to include all the components that have a greedy value within some d% of the best-rated component. Then the choice between components of the restricted candidate list is performed randomly, according to a uniform distribution. Once a full assignment is achieved, it is used as starting point for the local search phase. The use of restricted components lists prevent the generation of some assignments and consequently preclude the possibility to find assignments in O_p for some problems [75]. For this reason, variants of the GRASP method have been proposed in which a the parameter g% is handled in order to leave a chance for all the components during each constructive step.

Tabu search (TS) [83, 84] is a neighborhood-based heuristic that exploits a simple criterion to prevent the local search from halting; moreover, it adds a memory structure to prevent the creation of cycles during the search process. During each iteration a neighborhood of the current assignemt x^t is generetad and the best neighbor is selected

to be the new assignment x^{t+1} also in the case of a worsening in the value of m_p. A memory structure called *tabu list* stores each of the visited states for a certain number of iterations and remove this states from the neighborhood of the current assignment. More complex version of TS include also a focusing strategy, that allow promising assignments to be excluded from the *tabu list* in order to allow a broader exploration of their neighborhood [85].

2.3.2 Monte Carlo methods

This huge family of SOH was named after the notorious Monte Carlo Method (MCM) for the numerical estimation of integrals value [86]. This method, in its most simple version, can be outlined as follows:

- a uniform random sampling over the integration hyper-volume is performed and the values of the integrand function for each of the samples are stored;
- the mean of the stored values is computed and the estimate of the integral is obtained by multiplying this mean by the integration hyper-volume.

The method was very useful to extend the applicability of numerical integration to high-dimensional functions, indeed the error on the estimates decreases with the square root of the number of samples and it is independent from the number of dimensions. The most important contribution of MCM to SOH was the introduction of the idea of random sampling that is at the basis of all the SOH.

The Boltzmann distribution, Equation (2.10), is at the basis of statistical mechanics and the key component of many of the heuristics in the Monte Carlo family. According to the relation derived by Ludvig Boltzmann for discrete systems, the probability of a system to be in a state with a specific energy E, at a given temperature T, is related to the energy itself, in particular:

$$P(E) = \frac{1}{Z} \cdot e^{-\frac{E}{\kappa T}}, \quad Z = \sum_{E} e^{-\frac{E}{\kappa T}}. \tag{2.10}$$

In the equation above κ is the Boltzmann constant. Subsequent studies conducted by Maxwell showed that this energy-dependent exponential decay law applies to a significant number of physical properties. Consequently, in order to simulate molecular systems, a trajectory of states distributed according to Equation (2.10) must be generated. This can be achieved by means of a trial and rejection strategy using the so called Metropolis-Hastings criterion:

$$P_a(x^{t+1} = x') = \begin{cases} 1 & \text{if } \Delta E < 0, \\ e^{-\frac{\Delta E}{\beta}} & \text{otherwise;} \end{cases} \tag{2.11}$$

where $P_a(x^{t+1} = x')$ is the probability to include a new state x', generated somehow, in the trajectory, and $\Delta E = E' - E^t$ is the difference between the energy of x' and the energy of the last state included in the trajectory x^t; β is the product between the temperature and the Boltzmann constant. The resulting strategy guarantees that asymptotically the sampling will focus on low energy states. Moreover, it is also able to escape local minima since it allows the worsening of the values of E'. For this reasons it can be exploited to solve GO problems; in this case the value of E represent the value of the objective function f_0 of the given problem.

The Metropolis method (MeM) [87] has been probably the first SOH to be defined. It was introduced in the context of computer based simulations of molecular systems that was an emerging research field during the same period in which MCM was developed. It is based on the principle of trial and rejection sampling and works iteratively. The algorithm starts with a random assignment x^0 of the search variable, in this context called the *state of the system* in analogy with physical systems. Each iteration of the MeM can be conceptually divided in two phases: *neighbor generation* and *acceptance*. During the neighbor generation, a perturbation is applied to the current assignment x^t to generate a neighbor assignment x'. Then, in the acceptance phase the Metropolis-Hastings criterion is applied to choose between x' and x^t. If x' satisfies the criterion it becomes the new state of the system x^{t+1}. Otherwise, it is discarded and the algorithm proceeds without updating the state. Samples from two probability distributions are required to carry out this procedure: the first distribution is defined over the neighborhood of x^t, $P_n : \mathcal{N}^k_{x^t} \to [0,1]$; it gives the probability to generate an assignment x' from x^t. The second distribution is given by the Metropolis-Hastings criterion: $P_a : R \to (0,1]$; it gives the probability that x' is accepted as new assignment of the system. The use of a neighborhood structure is aimed to reduce the difference between the current state and the candidate assignments in terms of values of the objective function. This has a great impact on the performance of MeM, indeed, for many GO problems, after a certain number of iterations the average quality of assignments generated through random sampling would be very poor with respect to that of the current state and, consequently, the search process would become slow due to the high rejection rate. From a theoretical point of view, when applied to a discrete problem, the MeM method simulates the evolution of an homogeneous Markov chain over the states of the system, and this allowed to prove the asymptotic convergence of this method assuming the ergodicity of the chain [66].

Simulated Annealing (SA) [88] is one of the most studied and applied neighborhood based heuristics belonging to the class of Monte Carlo methods. The basic idea of SA was borrowed from the tempering process used in metallurgy to increase the toughness of iron-based alloys. The process itself consists in two phases, first the alloy is heated in order to allow atomic components to move inside the solid structure, then a slow cooling

phase is performed in order to let them re-organize in a new stable configuration. The heating temperature and the cooling rate are key determinants of the properties of the resulting configuration of the alloy. The SA applies the tempering process to the MeM method, allowing the parameter β in the Metropolis-Hastings criterion to vary during the optimization. The procedure starts from a high value β_0 (corresponding to high temperature) of the control parameter β, that is then decreased following some predefined cooling scheme. The most commonly used scheme is a logarithmic scheme of the type:

$$\beta_t = \frac{c}{1 + log(t)}, \qquad c \in \mathbb{R}. \tag{2.12}$$

Consequently the acceptance probability distribution varies as a function of t and the optimization process can be modeled with an in-homogeneous Markov Chain over the states of the system. In the matter of the optimization, the rationale of this behavior is to create a balance between global and local search. When the value of β is high, the sampling resemble a random walk allowing the method to move fast in high cost regions of the search space and eventually to escape local minima. On the contrary, when β decreases, the sampling becomes progressively more selective in order to converge to an optimal or near-optimal assignment of x. The asymptotic convergence of SA have been proved [89, 90] and several studies [65, 91] are dedicated to understand how the available information about the target problem can be exploited to choose an intial optimal value of the parameter β.

Replica exchange Monte Carlo (REMC) [92] is one of the most successful parallelization of the SA method (see [93] for a review). In the REMC method several MeM optimization, the *thermal baths*, are run in parallel at different temperatures. The state in REMC is therefor a set of MeM states -the *replicas*- and each replica represents an assignment of the optimization variable. Each bath is simulated independently for a given number of MeM steps. After that, the characterizing step of the REMC is performed, consisting in the exchange between two replicas according to the following probabilistic criterion:

$$P(swap(i, j)) = \min(1, e^{\frac{E_i - E_j}{\beta_j - \beta_i}}) \tag{2.13}$$

Where i and j represent two different replicas and the remaining notations are the same as in Equation (2.11). The execution proceeds alternating the exchange step and the parallel executions of thermal baths. The main idea of REMC is to perform exploration and exploitation in parallel, high temperature replicas perform exploration, while low temperature ones perform exploitation. If some high temperature replica enters a promising region of the search space, it will easily move to lower temperature baths according to equation (2.13), resulting in a quasi-local optimization. Specularly, a low temperature replica that is trapped in a local optima, have more chance to escape once

it is moved to higher temperature baths. This strategy provides an excellent compromise between exploration and exploitation showing to be very effective for several optimization problem [93] and in particular in the field of biomolecular simulations [94, 35, 95]. Moreover, it is easily portable on parallel architecture.

Wang-Landau sampling (WLS) [71, 96] is one of the most recent members of the Monte Carlo family of methods. Properly speaking, it has not been designed for optimization, but to provide estimates of the density of states in particle physics and related fields [97]. Nevertheless, in the case of HP-PSP problem, this method showed a good performance also for what concern the search of low energy states. An important difference of WLS with respect to the others Monte Carlo methods previously discussed, optimization in WLS is not a markovian process, since a memory structure is added to bias the probability to select new states. The basic idea of WLS it to perform a uniform sampling over the energies range. At the beginning of the execution, a histogram structure h and an array of densities are initialized to store the number of samples in each energy level and the associated density. If the number and the values of the energy levels associated with the system are not known a priori, a dynamic scheme can be applied; each time a new level is found, a bin is added to the histogram and the minimum number of samples between the already known levels is assigned to it; in the case of the densities, the value of the new bin is set to 1. The search is performed as in MeM, but the Metropolis-Hastings criterion is used to compare the number of samples in each of the two energy levels instead of the energy itself. Consequently, the acceptance probability given in Equation (2.11), is replaced by:

$$P(x^{t+1} = x') = e^{\frac{g(E')-g(E^t)}{\beta}}, \tag{2.14}$$

where $g(E^t)$ denotes the density associated to the energy value of the last assignment in the trajectory and $g(E')$ the density associated to the energy of the candidate assignment. If x' is accepted, then $h(E')$ is increased by one and $g(E')$ is increased by a value f. Otherwise the same increments are applied to $h(E^t)$ and $g(E^t)$. The search proceeds by checking periodically the degree of flatness of the histogram h. Once a user defined value is reached, h is reinitialized and f if reduced, usually according to the scheme $f^{n+1} = \frac{1}{2}f^n$. The execution terminates once f has dropped below a user defined threshold. The strategy implemented in WLS is very effective in escaping local minima and preventing re-sampling. Moreover, if the number of low energy states is low compared to the number expected assuming a uniform distribution of states over the energy values, the WLS works like an optimization algorithm oversampling the low energy region.

2.3.3 Population based methods

Population based methods are always defined bio-inspired heuristics since they usually mimic the behavior of some biological system, nevertheless this heuristics are based on mathematical basis and for many of them the asymptotic convergence have been proved [67, 68].

Genetic algorithms (GAs) were introduced by Holland in 1975 [98, 99] as methodology to tackle GO problems. They mimic the process of natural evolution theorized by C. Darwin. GAs exploit a population Pop^0 composed of n randomly created chromosomes, also called individuals, that, in the original form, are defined as fixed-length strings (over a binary alphabet) that represent assignments of the search variable. The individuals of the population undergo an iterative process whereby two genetic operators (crossover, mutation) are combined with a selection strategy based on an objective function to generate a new population Pop^1. During the selection process, individuals from Pop^t are chosen and inserted into a termporany population Pop' using some fitness-dependent sampling procedure [100]. The fitter the chromosome, the more times it is likely to be selected to reproduce. The crossover operator is applied to a user defined portion of Pop'. Each time the crossover is applied a couple of chromosome in Pop', called the *parent chromosomes*, is replaced by two new chormosomes, called the *offspring chromosomes*. Offpring chromosomes are generated randomly choosing one or more locuses and exchanging the subsequences before and after that locus or, in the case of more locuses, the segments delimited by the selected locuses, between the parent chromosomes. Once the desired number of chrossover events has been accomplished, the mutation operator randomly changes some of the locus in the chormosomes of Pop', allowing a further exploration of the search space. Mutation can occur at each position in a chromosome with a user defined probability. After the application of genetic operators, Pop' becomes Pop^{t+1} and the process iterates until a halting criterion is met, e.g., after a fixed number of generations. The resulting method simulate the evolution process that takes place in a biological systems subjeted to a *selective pressure* and the quality of the population, measured according to the objective function, increases along the iterations.

Ant Colony Optimization (ACO) [75] is a bio-inspired meta-heuristics to approach hard combinatorial problems in which a colony of simple agents (artificial ants) interact to efficiently explore the search space. The general idea of ant inspired systems is that of combining a constructive strategy with a global evaluation. Moreover, a memory structure, called *pheromone*, is used to store the relations between components in high-quality assignments and to trace the explored regions of the search space. This memory is then used to bias the choice during subsequent constructive steps. Each iteration of the meta-heuristics is composed of three main phases: *construction*, *evaluation* and

update. These phases vary slightly according to the specific ACO heuristic, for a review of the available strategies the interested reader is referred to [75], in what follows the $Max\text{-}Min$ Ant System heuristics [101] is described; this strategy is characterized by the fact that the quantity of pheromone in each position of the pheromone matrix is bounded in a range. The construction phase resembles a probabilistic greedy algorithm and its performed using a population of v ants. Each ant builds an assignment of x by means of a sequence of choices; each choice extends the previous partial assignment and is taken according to:

$$P_c(x_j = i) = \frac{\theta\ t_{ij} + (1-\theta)\ \eta_i}{\sum_b(\theta\ t_{bj} + (1-\theta)\ -\eta_b)} \tag{2.15}$$

where $P_c(x_j = i)$ is the probability of making the choice i during the j_{th} constructive step, b is an index running over set of choices available for the j_{th} constructive step, t_{ij} is the pheromone value associated with the i_{th} choice in the j_{th} step, η_i is the heuristic cost of the i_{th} choice computed according to some function that evaluates partial assignments, θ is a parameter of the algorithm used to balance between the two contributions. The construction phase ends when each ant have built a complete assignment. In the evaluation phase, assignments built from the population of ants generally undergo local optimization before being evaluated. The best assignment of the current interation x^{temp} is stored until the end of the iteration. Moreover, the algorithm keeps track of the best assignment found so far x^{best}. The update phase is itself composed of two sub-phases *evaporation* and *release*. During the evaporation sub-phase all the pheromone values are reduced according to eq .2.16.

$$\tilde{t}_{ij} = (1-\rho)\, t_{ij} + \rho\, t_{min}, \tag{2.16}$$

where $\tilde{t}_{ij}$ is the pheromone value associated to choice i_{th} in the j_{th} constructive step after the update, ρ is a parameter of the algorithm, and t_{min} is the lower bound of the range of variability of the pheromone values.

During the release sub-phase, an assignment is chosen between x^{temp} and x^{best} with a probability proportional to the ratio of their cost:

$$P(x^u = x^{temp}) = \begin{cases} 1 & \text{if } m_p(x^{temp}) < m_p(x^{best}), \\ 0.5\frac{m_p(x^{best})}{m_p(x^{temp})} & \text{otherwise.} \end{cases} \tag{2.17}$$

The pheromone values of the choices defining the selected assignment x^u are updated according to:

$$\hat{t}_{ij} = \begin{cases} \tilde{t}_{ij} + \frac{opt}{x^u} & \text{if } \tilde{t}_{ij} < t_{max}, \\ t_{max} & \text{otherwise.} \end{cases} \tag{2.18}$$

In Equation (2.18), $\hat{t}_{ij}$ is the value of pheromone matrix at positions i, j after the release, while opt is an estimate of the optimal value for the given instance. The algorithm starts with a uniform level of pheromone over all choices (generally the value t_{max} is used for this purpose) and the constructive phase behaves exactly as a probabilistic greedy algorithm. The parameter ρ controls the learning rate affecting the number of iterations required to reach a significant pheromone bias. When the pheromone bias becomes significant, the *collective memory* of ant agents is the main determinant during the constructive steps. So doing, the population of ants is induced to explore promising regions of the search space. An appropriate choice of learning rate is crucial in order to prevent undesired behaviors, in particular:

1. if the bias becomes strong when the quality of the releasing assignments is still poor, the method is trapped in a local minimum;

2. if the bias is too weak, a very long time is required to converge.

Although recently extended to continuous problems [102] the ACO meta heuristic has been applied mainly to combinatorial problems.

Particle Swarm Optimization (PSO) is a heuristic inspired by the collective movement of birds and fishes [103]. One of the great advantages of the PSO is that, along with Monte Carlo methods, it is one of the few approaches that can be easily applied to continuous GO problem in its native form. PSO exploits a set (the *swarm*) of n candidate assignments (the *particles*), which move inside a bounded search space in a collective effort to find a solution to a specified GO problem. At each iteration step t of the PSO, each particle is characterized by two vectors: the position $\mathbf{x}_i(t) \in \mathbb{R}^M$ and the velocity $\mathbf{v}_i(t) \in \mathbb{R}^M$. In its most common formulation, the movement of the i-th particle is a consequence of two attractors: the best position found by the swarm ($\mathbf{g}$) and the best position found by the particle itself ($\mathbf{b}_i$). Both attractions are perturbed by means of vectors of random numbers ($\mathbf{r}_1$ and $\mathbf{r}_2$) sampled with uniform distribution in [0,1], in order to avoid the entrapment in local minima; in addition, they are multiplied by two constants called *social* (c_{soc}) and *cognitive* (c_{cog}) factors. Hence, the velocity update formula for PSO is:

$$\mathbf{v}_i(t+1) = w \cdot \mathbf{v}_i(t) + c_{soc} \cdot \mathbf{r}_1 \circ \big(\mathbf{g} - \mathbf{x}_i(t)\big) + c_{cog} \cdot \mathbf{r}_2 \circ (\mathbf{b}_i - \mathbf{x}_i(t)), \tag{2.19}$$

where $w \in \mathbb{R}^+$ is an inertia weight factor, used to damp the velocity. Moreover, the intensity of the velocity is generally clamped to a maximum value $v_{\texttt{MAX}} \in \mathbb{R}^+$, before the particles positions are updated according to:

$$\mathbf{x}_i(t+1) = \mathbf{x}_i(t) + \mathbf{v}_i(t+1). \tag{2.20}$$

Thanks to the collective movement of particles, PSO eventually converges to an solution of the considered problem. The algorithm is stopped when a halting criterion is met, e.g., after a fixed number of iterations.

3 Biological background

This chapter describes the biological background of the thesis, focusing only on relevant aspects in order to provide a clear understanding of the treated problems. A detailed analysis of the biological and physical aspects of protein folding and NMR has been left out since it exceeds by far the scope of this thesis.

3.1 Proteins

Proteins are linear chains composed of small organic molecules, the amino-acids, connected each other in a head to tail fashion. This linear organization allows us to represent proteins as strings in which each character codes for one of the amino-acids in the sequence. Only 20 amino-acids, shown in fig. 3.1, are found in naturally occurring proteins. All the amino-acids share a common structural portion, the *backbone*, that is used to assembly the linear chain and to define the global geometry of the protein structure. The remaining part of the molecule, the *side-chain*, varies significantly between different amino-acids. The portion of the structure composed of the connected backbones is called the *main-chain*. The side-chains protrude on the sides of the main-chain and are able to interact between each others and with the main-chain depending on the conformation adopted by the protein. Since in a protein sequence each amino-acid can be repeated several times, biochemists use the name *residue* to identify a generic amino-acid in a specific position in the protein sequence; in this thesis the same convention has been adopted. Most of the proteins found in living beings adopt a well defined three dimensional structure referred as *native state*. Actually, the native state is not a single structure but a set of closely related structures that rapidly interchanges each others. Nevertheless for the purposes of this thesis and PSP problem in general we can consider it as a single structure. The proteins structure organization can be described using a three levels hierarchy as follows: *primary structure*, *secondary structure*, *tertiary structure*. This hierarchical view is useful to describe the main forces involved in the stabilization of protein structures and, at the same time, provides an useful way to identify the information that can be used to define a PSP protocol, i.e. input data,

building blocks of the combinatorial representation and output data.

The primary structure is the information that can be obtained combining the protein sequence with the chemical composition and topological structure of the amino-acids. It can be processed to achieve bounds on relative displacement of atoms, to define simplified models that aggregate groups of atoms and to assign chemical and physical properties to both atoms and residues. Two of these properties are of particular interest to understand the work in this thesis: *polarity* and *hydrophobicity*. Polarity is chemical property that indicates an unequal distribution of the bonding electrons between atoms of the same molecule resulting in the creation of electrical multipole or permanent charge separation. In fig. 3.1 a commonly used classification of the amino-acid based on polarity is shown. Hydrophobicity is often considered the inverse of polarity, but, as the name suggests, it is mainly related with the possibility for a compound to create favorable interactions with water. PSP methods that exploit only primary structure as input are generally defined *de novo* approaches.

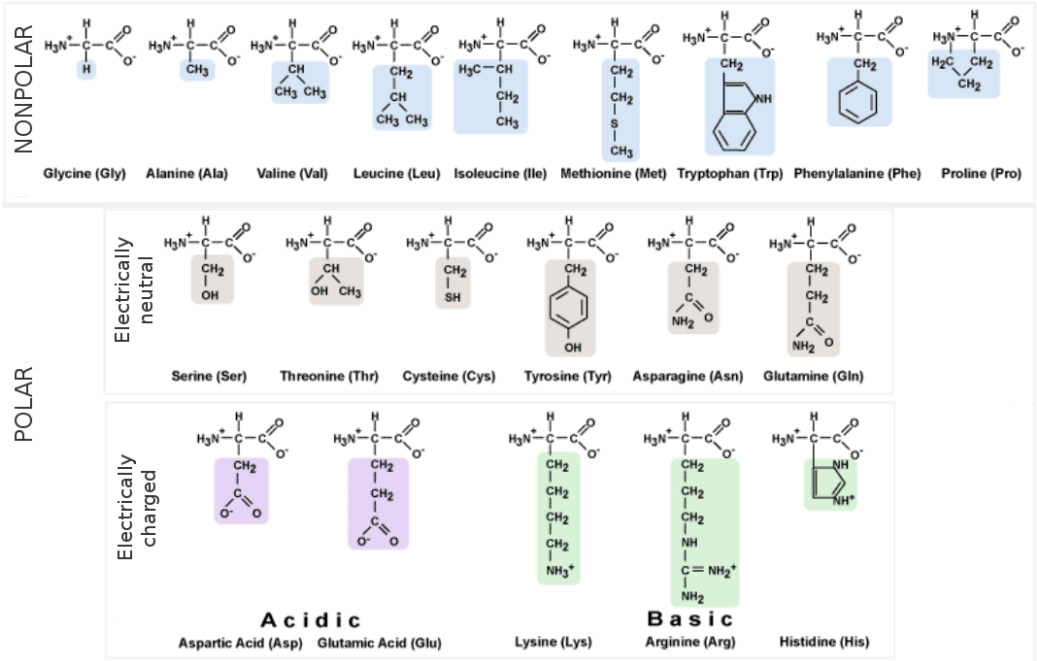


Figure 3.1: topological structure and composition of the 20 amino-acids occurring in natural proteins. The classification of amino-acid based on polarity is represented, non-polar (hydrophobic) amino-acids are shown in the upper panel, polar amino-acid in the bottom panel. The latter are further divided depending on the presence of electrical charge in physiological conditions.

The secondary structure describes the local organization of subsets of consecutive residues in the linear chain. Secondary structure is partially determined by the interactions between the backbone atoms in the considered subset of residues. Two main patterns, shown in figure 3.2, the *helix* and the *strand*, define the local structural organization of a significant portion of proteins structures, while the remaining part is organized in a random coil fashion referred as *loop*. This limited variability arise from the constraints due to the steric interactions (clashes) between the backbone and side-chains and to a stabilization effect described in section 3.2.2. The relation between sequence and secondary structure is complex, since the latter depends both on local and non-local interactions, nevertheless approaches based on sequence comparison and machine learning from the known proteins structures, provide in most cases reliable predictions of both helix and strand regions of a protein sequence [104, 105].

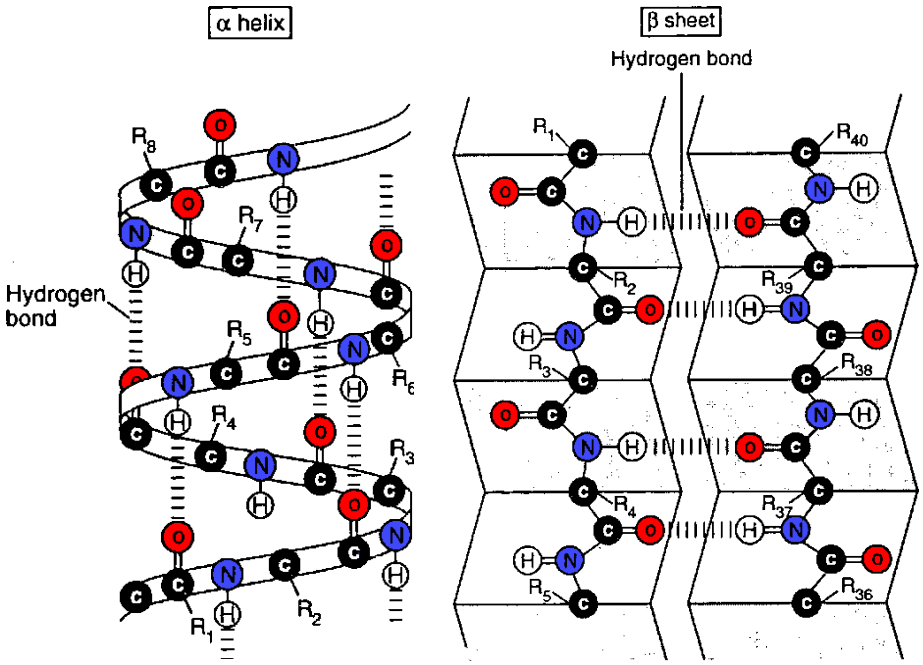
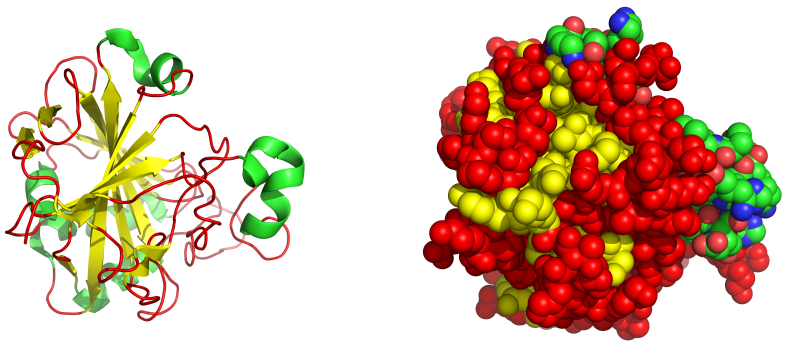


Figure 3.2: schematic representation of the two common secondary structure patterns and associated hydrogen bond networks, only main-chain atoms are shown using the CPK coloring system. α-helix (left), is characterized by a period of 3.6 residues per turn with an hydrogen bond between the carbonyl group of residue i and the amide group of residue $i+4$. β-sheet (right), is a tertiary structure pattern generated by the interaction of two or more β-strand through an hydrogen bonds network.

The tertiary structure describes the overall geometry of the main-chain and depends mainly on the packing of the side-chains in the interior of the structure as shown in figure 3.3. This is the information that is expected as output of a PSP method. As a

consequence of the evolutionary process, that is based on gene duplication, mutation and fusion, in many cases larger proteins are composed of many independent folding units called *domains* or *folds*. Each domain defines a particular tertiary structure motif. Similar domain can be grouped into families in order to provide a classification [106, 107]. A great variability of tertiary organization have been observed in the known proteins structures but a significant number of the known protein domains are distributed over few families while, on the other side, many domains seems to be unique [30]. For these reasons a great number of technique have been developed to perform domain recognition [108, 109] and template based modeling [110, 111, 112].



(a) (b)

Figure 3.3: two different representation of the tertiary structure of the human carbonic anhydrase II, PDB id 3KS3, this is representative of the homonymous super-family according to CATH classification [107]. This family is characterized by a roll structural domain. (a) Cartoon representation, this representation focuses on the main-chain atoms and emphasizes the disposition of secondary structure patterns: helix (green), strands (yellow) and loops (red) . (b) Balls representation, emphasizing, the close packing of atoms associated to protein folding that leads to the creation of van der Waals interactions, see details in the text, coloring as in (a). Images generated using PyMOL.

3.2 Inter-atomic forces and their modeling

In his work, basing on the evidence of spontaneous refolding of small globular proteins, Anfinsen [113] postulated that, at least for this class of proteins, folding is a spontaneous process that leads to the lowest energy conformation. This is called the thermodynamic hypothesis and is one of the major assumptions in the field of structural biology. Consequently, the primary requirement in order to understand the relationship between sequence and structure and to solve PSP problem is the definition of an accurate model that captures the main physical interactions involved in the stabilization of protein

structures. In this section an overview of the major interactions involved in the folding process is provided along with some simple examples of the way they can be modeled to perform computation, the interested reader is referred to the Israelachvili's book [114] and Leach's book [115] for a detailed account of inter-atomic interactions and modeling techniques. The most commonly used representation of physical interactions in protein folding simulations is based on *molecular mechanics* force fields [116, 117, 118]. This gives a classical approximation of internal energy of molecular systems that are too large to be treated with quantum mechanics. It is based on several assumptions, the most importants of which are the following:

1. electrons are uniformly distributed on the surface of a sphere centered in the nucleus of each atom;

2. covalent bonds can be represented with harmonic potentials and can not be broken.

The first assumption is based on the fact that most of the atoms that compose biological molecules falls in the first three periods of the Mendeleev table, where a tight association between electrons and nuclei is observed. The second relies on the fact that in the temperature range in which life is observed the thermal energy available is not enough to break covalent bonds. In order to represent polarization, without considering the movement of electrons, both integer and partial electrical charges are permanently assigned to the atoms.

3.2.1 Bonded interactions

The bonded interactions involve covalently bonded atoms that are close in terms of topological connections in the primary structure. From a physical point of view they are the result of a combination of several forces related to the quantum mechanical properties of chemical bonds. In molecular mechanics these interactions are modeled associating a simple cost function to the displacement of some structural descriptor, i.e. distance or angle, from an equilibrium value that depends on the chemical species involved . The importance of bonded interactions is due to the fact that they are directly associated to the internal degree of freedom of the protein. Three main terms describe the energy variation related to the bonded interactions, stretching energy, bending energy and torsion energy. The stretching energy U_{str} is associated to the variation in the distance between two bonded atoms. The bending energy U_{bnd} is associated to the variation in the angle between two consecutive bonds in a chain. While, the torsion energy U_{trs} is associated to the rotation around the central bond in a chain of three consecutive bonds. U_{str} and U_{bnd} are generally modeled with a harmonic potential as

follows:

$$U(x) = \frac{k_i}{2}(x - x_i)^2. \tag{3.1}$$

Where x is the value of the considered degree of freedom, distance or angle, k_i and x_i are both empirical parameter representing the force constant and the equilibrium distance or angle for the considered set of atoms. Stretching and bending interactions are both associated to high force constants, this means that small deviations from the equilibrium value result in high energy variations. Consequently, stretching and bending interactions are more useful to provide constraints to limit the conformational space, than to evaluate the reliability of a structural model: indeed any physically reasonable conformation has a minimal stress over these degrees of freedom. In molecular mechanics U_{trs} is represented with a cosine expansions of the form:

$$U_{trs}(\omega) = \sum_{n=0}^{N} \frac{V_n}{2}\left[1 + \cos(n\omega - \gamma)\right]. \tag{3.2}$$

Where ω is the torsion angle, N is the number of expected energy minima for the modeled system, V_n is the depth of the n-th minimum and γ is a parameter used to manage the offset of the minima. Special cases such double bonds to cyclical structures are handled through an additional term, called improper torsion. The torsional degrees of freedom are the most important in conformational sampling because the associated energy is the lowest among the bonded interactions and unfavorable torsional energy can be balanced from the creation of favorable electrostatic interactions discussed below. For these reason the majority of PSP methods limit the conformational search [119, 120] only to the torsional degrees of freedom freezing bond angles and length to ideal values.

3.2.2 Electrostatic interactions

Electrostatic interactions interest atoms that are spatially close, independently from their topological connection in terms of covalent bonds. The computational modeling of this kind of interactions is challenging due both to the lack of unifying theoretical model and to issues related to computational complexity, for this reason it is still object of intense research [121, 122, 123, 124, 125]. Depending on the accuracy required, they can be modeled with a single potential or with several specialized potentials. Conceptually it is possible to divide these kind of interactions in three class, those involving charged atoms commonly referred as ionic bonds, those involving polar groups, and those involving non polar groups.

The ionic bonds is the interaction between electrically charged atoms and the energy $w(r)$ of two charged species at distance r is obtained integrating the Coulomb law, and

represented as follows:

$$U_{Coulomb}(r) = \int_{\infty}^{r} \frac{Q_1 Q_2}{4\pi\epsilon_0\epsilon r^2} dr. \tag{3.3}$$

Where Q_1 and Q_2 are the net charge of the two species involved, while ϵ_0 and ϵ are the values of the dielectric constant in vacuum and in the considered medium respectively. Since water has an high dielectric constant and charged side-chains are exposed to the solvent both in the unfolded and folded conformations, the contribution of intra-molecular ionic bonds in the change of free energy associated to the folding is limited although not completely negligible [126]. However, the interaction between charged side-chains and the water is very important and must be modeled in order to prevent the creation of artifacts. This can be done: i.e. considering the difference in the Born energy equation (3.4) between an exposed charged group and an internal charged group. The Born energy, is the energy required to charge a particle of radius r with a charge Q and is defined as follows:

$$U_{Born}(Q) = \int_{0}^{Q} \frac{q dq}{4\pi\epsilon_0\epsilon r}. \tag{3.4}$$

Since the dielectric constant in the interior of the protein is lower than in the water, the value of Born energy will be higher resulting in an unfavorable transition.

Van der Waals interactions involve permanent or instantaneously induced dipoles. They apply to any kind of atoms both polar and non polar. The energy associated to these interactions is very low and they strongly depend on the distance between involved species becoming relevant only when the atoms are placed in close proximity. Nevertheless, they become relevant when the number of closely packed atoms is high, this is the case of the apolar side-chains packing in the core of globular proteins. For this reason van der Waals interactions give a significant contribute to the stabilization of protein structures. The modeling of van der Waals interactions is accomplished by means of the empirical Lennard-Jones potential as shown in the equation below:

$$U_{vdW}(r) = D_0 \left[\left(\frac{r_{min}}{r} \right)^{12} - 2 \left(\frac{r_{min}}{r} \right)^{6} \right]. \tag{3.5}$$

Where D_0 is a parameter that sets the minimum value of the interaction energy at the equilibrium distance r_{min} for the involved species. It is important to notice that this potential includes also a strong repulsive term that accounts for Pauli exclusion principle and short distance Coulomb interactions that prevents the clashes between atoms.

The hydrogen bond is a special case of van der Waals interaction between permanent dipoles that, due to the reduced size of the hydrogen atom, is particularly strong. It is

the most common interaction between polar groups observed in proteins and also the main determinant of the peculiar properties of water that are fundamental for the self assembly and function of biomolecules such proteins [127]. Consequently, it plays an important role in the determination of protein structure. The formation of a hydrogen bond requires two particular chemical groups, the hydrogen donor that includes a polarized hydrogen atom with a partial positive charge, and the acceptor that includes an electronegative atom with a partial negative charge. If some constraints on distances and angles between the two groups are satisfied the polarized hydrogen creates an electrostatic bridge between the two groups. The creation of a single hydrogen bond is associated with a small reduction in the internal energy, but due to the high number of hydrogen donors and acceptors available in the amino-acids and in the water, the folding process is associated with the creation of an extended network of hydrogen bonds that gives a relevant contribution to the total energy. From a structural point of view, the most evident effect of hydrogen bond networks is the stabilization of the secondary structure patterns as shown in 3.2. The energy associated to hydrogen bond formation U_{HB} is modeled through an empirical potentials similar to the Lennard-Jones potential with the addiction of a directional dependency as shown in the equation below:

$$U_{HB}(r,\theta) = D_1 \left[5\left(\frac{r_{min}}{r}\right)^{12} - 6\left(\frac{r_{min}}{r}\right)^{10} \right] \cos^4(\theta) \tag{3.6}$$

Where D_1 and r_{min} are parameters analogous to D_0 and r_{min} in the Lennard-Jones potential, while θ is the angle between the bond vector in the donor and the vector that points from the polarized hydrogen to the acceptor.

3.2.3 The hydrophobic effect

When an apolar compound is placed in a polar solvent, able to create an internal network of hydrogen bonds, it has the tendency to aggregate (i.e. oil drops in the water) in order to minimize the surface that is exposed to the solvent, this effect, shown in fig. 3.4, is particularly evident in the case of the water and for this reason it is called hydrophobic effect. In the case of globular proteins, the hydrophobic effect interests the apolar part of the side-chains and it is responsible for the creation of compact cores in which these side-chains are closely packed. According to the current energetic description [128, 127], the hydrophobic effect is the leading force involved in the folding process. The physical basis of the hydrophobic effect are still object of debate but it is now clear that it arises at least from two different contributions, these are: the increase in the solvent entropy, and the creation of the van der Waals interactions in the core of the protein. In order to evaluate the energy variation associated to the hydrophobic effect U_{asa} in protein folding, most potentials include an empirical term based on the multiplication of residue-specific hydrophobicity coefficients and accessible surface area (ASA). This kind of energy term was introduced by Eisemberg and MacLachlan

[129], and has the following form:

$$U_{asa} = \sum_i \sigma_i A_i \tag{3.7}$$

Where A_i and σ_i are respectively the ASA and an empirical coefficient measuring the hydrophobicity relative to the i-esime atom. The ASA is the area of each atom or residue that is accessible to water molecules and is computed using variants of the algorithm introduced by Shrake and Rupley [130]. It is important to notice that U_{asa} considers both an internal energy contribution and an indirect estimate of the entropic contribution relative to the variation in the entropy of the water. This is a significant difference between this term and the other components of the evaluation function previously described.

3.2.4 Statistical Potentials

An alternative method for protein model evaluation is the use of statistical potentials [131, 132, 133, 134, 135]. The fundamental idea at the base of statistical potentials is that chemical objects (such residues, or atoms), have both preferential and unfavorable states for at least some *observables* in a defined environment. If a big enough dataset of observation is available, it is possible to capture these preferences and to turn them into scores. For what concerns proteins structure evaluation, the information needed to build statistical potentials can be gained from non-redundant subsets of the PDB database. The concept of observable is very broad and includes everything we can directly or indirectly measure, for example pairwise distance between residues, dihedral preference, geometrical organization of secondary structures, chemical environment and so on. Since, in the case of folding, one is interested in the relationships between sequence and structure, commonly used statistical potentials always associate one or more sequence descriptors to one or more structural descriptors. Although different statistical models can be used to derive potentials, a very generic form of statistical potentials has been introduced by Rooman et al. [136] and it is shown in the equation below:

$$\Delta W(c_1, c_2, ..., c_n) = -kT \log \frac{P(c_1, c_2, ..., c_n)}{P(c_1)P(c_2), ..., P(c_n)}. \tag{3.8}$$

Where ΔW is the variation of score associated to a given set of value of the observables relative to the reference state, $c_1, c_2, ..., c_n$ are the values of the considered observables and $P(c_i)$ denotes the probability of the value c_i while $P(c_1, c_2, ...c_n)$ denotes the joined probability of the given combination of values. One interesting properties of observables in statistical potentials is that they can be easily combined leading to complex descriptors. The main consequence of this is that, in some cases, the physical interpretation of the statistical potentials remains obscure. Statistical potentials are broadly used in PSP methods [119, 120, 95], sometime in combination with molecular mechanics.

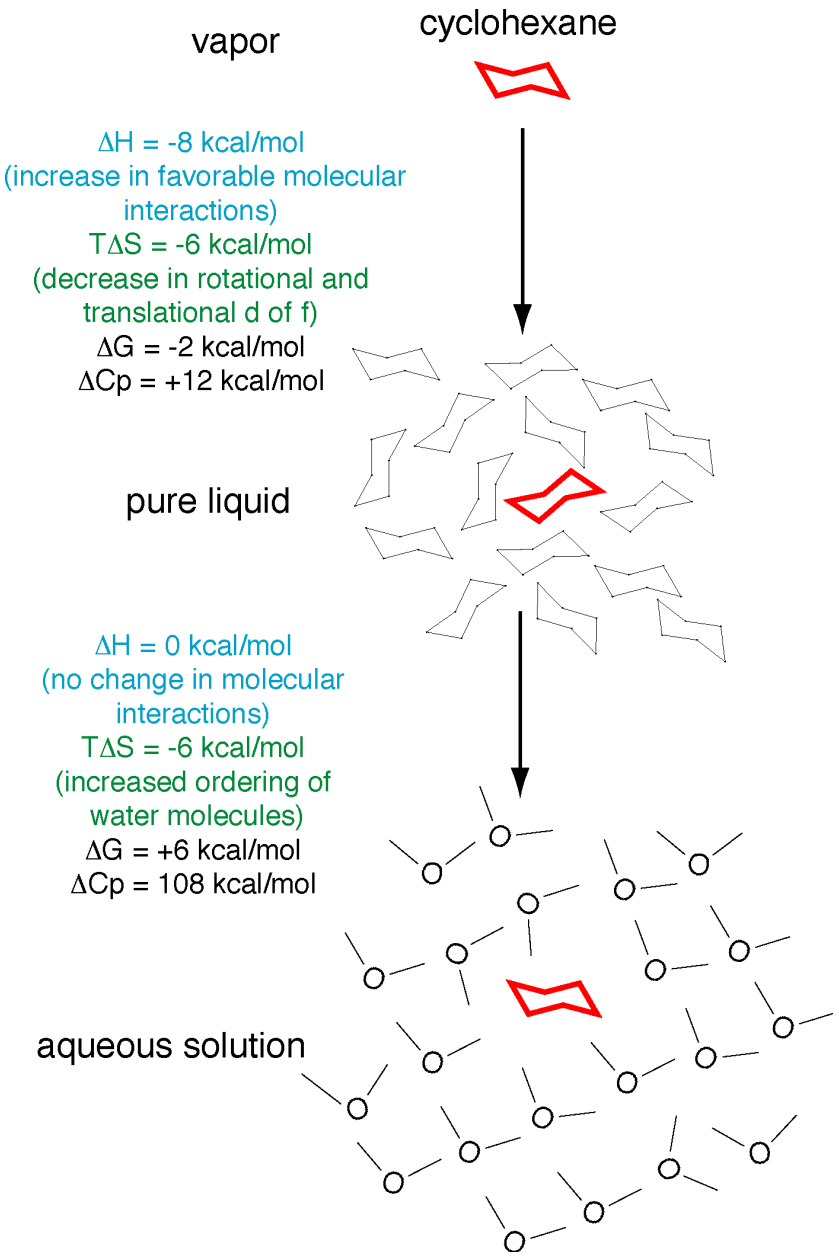


Figure 3.4: schematic representation of the hydrophobic effect for a small organic molecule. The transition from vapor to the pure liquid (comparable to the interior of a protein) is energetically favorable since the creation of van der Waals interactions balance the loss in rotational and translational entropy. The transition from the pure liquid to the water solution is unfavorable since there is no gain in van der Waals contacts but there is a reduction in the rotational and translational entropy of water molecules.

3.3 Energetics of protein folding

In the last forty years a huge number of experimental [137, 138, 139, 140] and theoretical [141, 142, 143, 144, 145, 146, 147] studies investigated the role played by the different kind of inter-atomic interactions in order to elucidate the energetics of the folding process. According to the thermodynamic hypothesis, the folding process is a spontaneous and reversible chemical reaction involving the protein and the solvent. The energy associated to chemical reactions is expressed in terms of variation of the Gibbs free energy ΔG_T between reagents and products:

$$\Delta G_T = \Delta H - T\Delta S. \tag{3.9}$$

Where H is the enthalpy of the system, that is a function of the system internal energy U, S is the entropy, that is a function of the number of accessible states of the system, and T is the temperature. The Gibbs free energy can be used to determine equilibrium constants for reactions that take place at constant temperature and pressure (both reasonable assumptions for biological systems). Spontaneous reactions at a given temperature T are characterized by a negative variation in Gibbs free energy $\Delta G_T < 0$. In order to illustrate all the different aspects that contribute to the folding process, it is useful to further decompose the energy in four different terms and then look at their variation during the process:

1. protein internal energy U_p, that considers all the interactions involving atoms of the protein;
2. solvent internal energy U_{sol}, that considers the interactions involving atoms of the solvent;
3. protein conformational entropy S_p, that considers the number of different conformations available to the protein, roughly speaking the freedom of motion under the effect of thermal energy;
4. solvent entropy S_{sol}, that considers the number of states available to the solvent, roughly speaking measure the rotational and translational freedom of the solvent molecules under the effect of thermal energy.

The first deduction arising from this decomposition is that the folding process involves a reduction in S_p, since compact structures have a reduced freedom of motion with respect to random coil. Moreover, theoretical studies and computer simulations suggest that the variation in U_{sol} associated to the folding process is negligible, since solvent can easily reorganize itself in order to preserve favorable interactions [128]. Therefore, to explain the negative ΔG_T observed, the reduction in S_p must be coupled with a greater reduction in U_p and/or with an increase in S_{sol}. The internal energy component U_p depends on the interactions that take place between protein and solvent and

can be modeled with molecular mechanics. The entropic components, on the other side, can only be approximated without an extensive sampling of the conformational space. As explained in the previous section, the gain of S_{sol} associated to folding is somehow included in the potentials using the ASA approximations. The estimates of the S_p are more challenging since they would require a computationally unfeasible amount of conformational sampling and these terms have been often neglected. In first approximation, one can assume that the loss of S_p is somehow proportional to the compactness. Recent evidences [148] contrast this assumption, suggesting that the difference in conformational entropy can be significant also between structures with similar compactness. Although, conformational entropy plays a significant role in the determination of both secondary and tertiary structure [149, 150] the inclusion of reliable estimates of conformational entropy in potential energy functions remains a major challenge.

3.3.1 The folding process: from the Levinthal's paradox to the funnel theory

One of the most impressing aspects of the folding process is the apparent simplicity and speed that characterizes the transition from the unfolded state to the native state. For some protein sequences this transition requires just a few μs [151]. This is surprising if one considers the astonishing dimension of the conformational space. One of the first theories of protein folding postulated by Cyrus Lenvinthal [152], analyzed what, at that time, seemed to be a paradox, resulting in the pathway model. This description was based on the idea that the *energy landscape* associated to the folding problem is dominated by a flat hyper-surface, representing the huge number of random coil conformations, with a single steep potential well, corresponding to the native structure. This kind of energy landscape is often referred as the *golf-course* landscape. In this scenario it would be impossible for the protein to perform the search over the flat region in order to find the potential well in a reasonable amount of time. To justify the observed folding rates, a pathway model was introduced. According to this model there is a well defined pathway leading from the denatured conformation to the native conformation describing a specific trajectory in the conformational space. The main problem of the pathway theory was the fact that it assumes that the denatured state has itself a specific conformation, while experiments performed in different denaturing conditions, determining different unfolded conformations, lead to similar folding rates [153]. For this reason a new model was introduced, called the funnel model [154, 155]. According to this model the energy landscape of the folding process is funnel shaped, with a strong gradient that points toward compact structures. At the beginning of the process, the hydrophobic effect and the creation of intra-molecular contacts drive the creation of compact structures called *molten globules* opposing the loss in conformational entropy. This behavior is independent from the conformation of the unfolded protein. The rate limiting step of the process seems to be the conformational search in the space of compact structures when some local minima appear [76]. This step is driven by the

creation of van der Waals interactions in the core of the protein and, possibly, by a further increase of the internal hydrogen bonding network due to the increase in regular secondary structure. Figure 3.5 summarize the energetics of folding according to the funnel model.

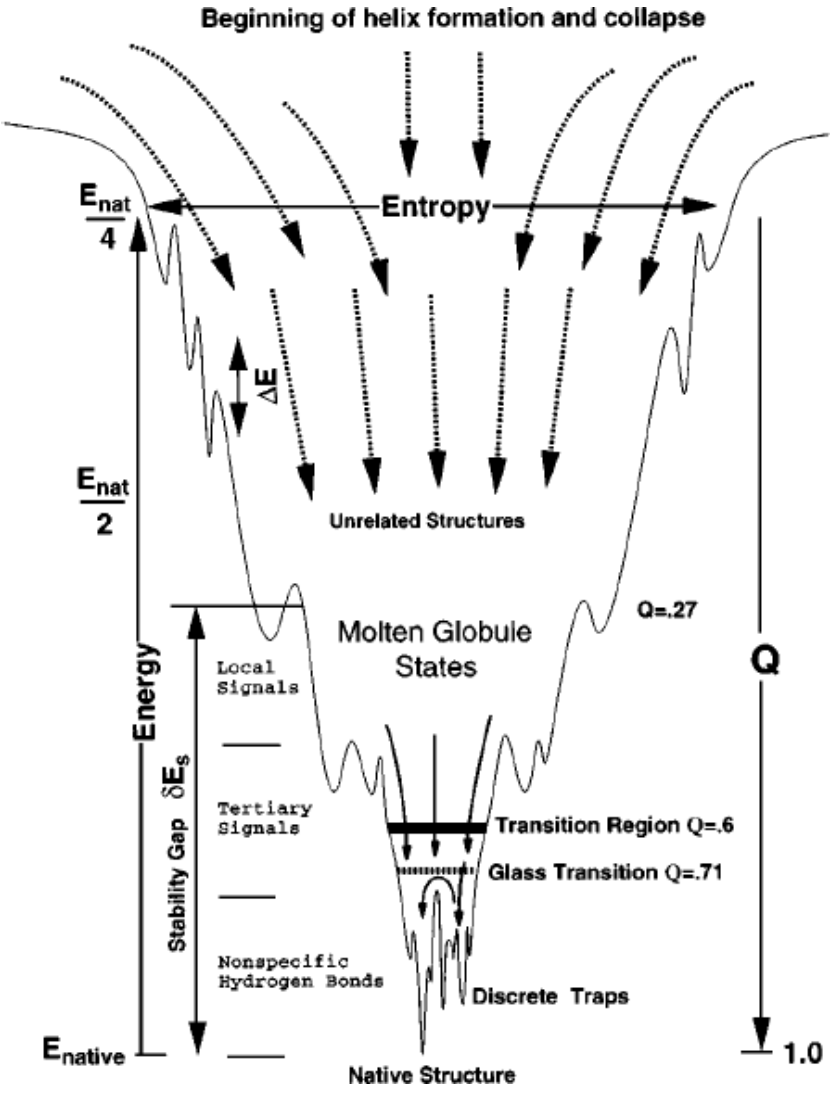


Figure 3.5: the funnel model of protein folding. **Q** denotes the reaction coordinate of folding, that is the fraction of contacts in a given conformation that are the same as in the native structure. Image from *Onuchic et al.* [146].

3.4 Nuclear Magnetic Resonance

This spectroscopic technique is, along with X-ray crystallography, one of the only two experimental methods available to characterize the three dimensional structure of large biomolecules such proteins. Although it still presents some limitations concerning

the size of the analyzed molecule, it offers three important advantages over X-ray crystallography:

- It does not require the crystallization of the target protein. Crystallization is indeed a complex and time demanding process;
- It does not require high energy electromagnetic waves and connected complex facilities.
- It allows to investigate the conformational variability of the target protein allowing a deeper understanding of the mechanism associated to the protein function.

NMR spectroscopy is itself a broad research field and more then a dozen of different protocols are routinely used for the determination of a single protein structure, for an in-depth description of principles and techniques the interested reader is referred to [156], in what follows just a quick overview is given in order to understand the restraints to the available data in the MDG problem.

The NMR technique is based on physical properties of fermions called *spin*. The spin is the quantum mechanics equivalent of the angular momentum in classical physics and when it is associated to electrical charge, as in the nuclei of atoms, it generates a magnetic field. Differently, from angular momentum spin is quantized and for a system with spin value S only $2S+1$ configurations are possible. The nucleus of atoms with both even number of neutron and protons has spin zero, integer values of spin arise when the total number of nucleons is even, otherwise half integer values are observed. Isotopes with spin value $|S| = 1/2$ can have only two different spin configurations and are the only used in NMR applications. The most important isotope for structural studies of protein through NMR is the hydrogen H_1. This is the most common isotope in nature and is abundant in all the biomolecules. Other two isotopes, routinely used in the determination of protein structures, are the nitrogen N_{15} and carbon C_{13}, these are both radioactive isotope that require the labeling of the target protein.

During a NMR experiment a strong magnetic field is applied to the sample. In the absence of the external magnetic field an atoms population will distribute itself uniformly between the two spin states, but if an external magnetic field is applied, the population will align itself to the magnetic field. This means that one spin state become preferential and the distribution of the atoms population between the two states will follow the Boltzmann equation. If an electromagnetic pulse with the appropriate frequency is then applied to the aligned system, the energy from the pulse can be used to promote some of the nuclei to the higher energy state. After a given amount of time, the excited nuclei will decay back to the low energy state, releasing the energy in the form of an electromagnetic wave of characteristic frequency that reflect the energy difference between the two spin states. If the electromagnetic environment is identical for all the nuclei in the system a

single signal will be emitted after the pulse. Otherwise a mixture of waves with different frequencies will be emitted, this signal can be decomposed in the different frequency components using Fourier transform. The difference in the frequencies associated to a particular type of isotope are measured with respect to the value observed in a reference compounds and are called *chemical shifts*. In complex structures such protein we expect that each hydrogen atom experience a slightly different environment and consequently the ideal spectra should contain a different peak for each hydrogen.

In practice this situation is unrealistic since the resolution of the method is limited and many overlaps are observed. To reduce this problem advanced NMR methodology are based on the use of sequence of pulses at different frequency in order to study the coupling between different atoms, i.e. in terms of correlation between their relaxation signals. These methods are called multidimensional NMR and are aimed to increase the resolution by adding dimensions. Moreover, when the protein is labeled, it is possible to study more than one isotope simultaneously and to label only one amino-acid at time further increasing the resolution. Two different objectives have to be reached in order to allow the structure determination through an NMR: the assignment of atoms and the generation of structural restraints. The assignment phase involves the association of each peak to a specific hydrogen atom. This is a complex process, in which starting from a few characteristic chemical shifts that can be assigned easily several multidimensional NMR protocols are used to observe the polarization transfer between topological close atoms. The resulting information is combined with the information from the primary structure in order to assign the other chemical shift to different atoms. Once assignment are generated the Nuclear Overhauser effect (NOE) is used to obtain restraints. The NOE is a particular type of spatial transfer of polarization that can be observed with specific NMR experiment such NOESY and HOESY, this effect decreases rapidly with the distance and in general is detectable for couples of atoms within 5 Å of separation. The result of an NMR experiment is thus a collection of distance and sometimes backbone angular restraints between the subset of the atoms that are in close proximity in the three dimensional structure.

4 Problems definition and state of the art

This chapter introduces the definitions of the protein structure prediction and molecular distance geometry problems and provides a review of the state of the art. Both off-lattice and lattice approaches to PSP are considered with a particular focus on the hydrophobic polar model and the related literature. Moreover, the MDG problem is presented in the standard form and the special case of exact distance restraints is discussed along with an overview of existing approaches.

4.1 The protein structure prediction problem

The protein structure prediction (PSP) problem is defined as the problem of finding the lowest energy three dimensional conformation of a given protein sequence. As discussed in the previous chapter, the conformation adopted by a specific protein depends on both the intra-molecular interactions between the protein atoms and the interactions between the protein and the surrounding environment (the solvent). It has been shown that these interactions can be quantified, with different levels of approximation, using functions that depend only on the internal coordinates. This estimate of the energy can be used to assess the relative quality of different structural models for a given sequence. From the point of view of optimization this means that we can define an evaluation function for the PSP problem. Clearly, this evaluation function depends on the level of accuracy used to represent the structural models and the underlying physics. For this reason, a general definition of the PSP problem requires an agnostic point of view with respect to the representation and the scoring function. A generic instance ι of the PSP problem is given by a string in the amino-acidic alphabet: $\iota \in \Sigma_{aa}^*$. A representation is a function $g : \Sigma_{aa}^* \rightarrow \mathcal{A}_{\iota,}$ that maps the input string into a given number of three dimensional objects, each associated with specific features and a local topology. The objective function $f_{\iota,g} : \mathcal{A}_{\iota,g} \rightarrow \mathbb{R}$ is defined over the space of the three dimensional coordinates of the objects given by the selected representation and considers also the instance-dependent information (i.e., chemical and physical properties inferable from the primary structure). Finally, the constraints are aimed to

prevent the overlaps between the objects and to preserve the chain integrity. The PSP problem can thus be written in the standard form as follows:

$$
\begin{aligned}
&find\ x^* \in \underset{x \in \mathcal{A}_\iota}{arg\,min}\, f_{\iota,g}(x),\\
&s.t.\\
&\quad c_{ij} - \|x_i - x_j\|_2 \leq 0, \quad \forall i,j \in \{1,2,...,g(\iota)\}, c_{ij} \in \mathbb{R}_+\\
&\quad \|x_i - x_b\|_2 - c_{bi} \leq 0, \quad \forall i \in \{1,2,...,g(\iota)\}, \forall b \in B_i, c_{bi} \in \mathbb{R}_+\\
&\quad \mathcal{A}_\iota = \mathbb{R}^{g(\iota)\times 3}.
\end{aligned}
\tag{4.1}
$$

In the above definition, x_i denotes the three dimensional coordinate vector of the i-th object according to the representation of the given input sequence given by g, $\|\|_2$ is the euclidean norm and B_i is the set of objects that are topologically linked to x_i according to the chosen representation.

4.2 Off-lattice approaches to PSP

The increase of computational power and in particular the development of distributed [157, 158] and parallel [159] computing along with the introduction of specialized hardware [160], allowed, in recent years, the first realistic simulations of the protein folding process [161, 162, 163, 164]. These approaches are often referred as *ab initio*, since they rely only on the physical description of the process, and are based on molecular dynamics [163] or Monte Carlo sampling schemes such as REMC [162]. Although these pioneering studies are opening the door to the appealing perspective of simulating the whole folding process gaining insights also on the thermodynamical aspects, they are still limited to very small systems and require an astonishing amount of computational resources. For these reasons, their are unlikely to provide a genomic scale protocol to PSP in the next future [165]. According to CASP results [166, 95, 167] the most effective algorithms for PSP on large scale are based on the *fragments assembly* (FA) strategy.

4.2.1 The FA strategy for PSP

The FA strategy was introduced by Baker and coworkers [168] and it is based on a discretization of the search space. It uses structural fragments derived from know proteins as combinatorial building blocks. Consequently, a structural model in FA strategy is represented as a combination of structural fragments. Each fragment defines the local structure of a segment in the input sequence. The elementary operation used during the optimization process in a FA based scheme is the fragment exchange. Specialized procedures are used to reduce the impact of each fragment insertion on the global structure acting on torsional degrees of freedom at the boundaries of the insertion point [119]. In order to limit the conformational space, only a predefined number of

fragments is allowed to represent a given sequence window, as shown in figure 4.1. The selection is based on the local sequence similarity between the target sequence and the fragment source, and on the agreement between the predicted secondary structure in the target and that observed in the fragment. Although fragments assembly approaches, are able to deal with larger protein sequences with respect to *ab initio* approaches they still require a considerable amount of computing resources. For this reason a great number of the studies [36, 38, 40, 41, 42] aimed to develop new optimization protocols for PSP is performed on representations that allows a further decrease in the computational requirements.

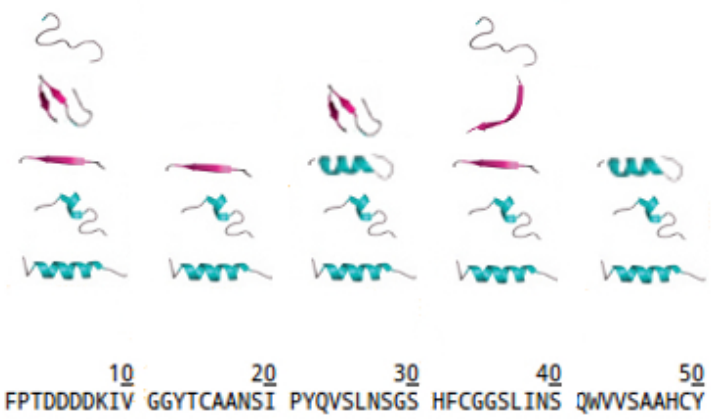


Figure 4.1: The fragment assembly strategy
An example of the discretization adopted in the fragment assembly strategy. For the sake of clarity overlapping fragments have not been considered in this example, nevertheless they are commonly used in PSP methods.

4.3 Bravais Lattice

Beyond the fragment assembly strategy, the most used simplified representation of protein structure [131, 32, 169] relies on a regular discretization of the search space based on the use of Bravais lattices. A Bravais lattice is a discretization of the euclidean space, that defines a regular distribution of points [170]. This means that the global structure of the lattice can be inferred from the local structure at each lattice point. A Bravais lattice is completely determined by a set of direction vectors $\mathcal{D}_{main}$. These vectors have all the same length $||u||$ and define the local structure of the space, representing the permitted directions of movement from a lattice point to one of its nearest neighbors. The cardinality of $\mathcal{D}_{main}$ is called *coordination number* of the lattice. Given $\mathcal{D}_{main}$, any

lattice point l_p can be obtained from a combination of the direction vectors as follows:

$$l_p = n_1\mathbf{u}_1 + n_2\mathbf{u}_2 + \cdots + n_{|\mathcal{D}_{main}|}\mathbf{u}_{|\mathcal{D}_{main}|}, \quad n_i \in \mathbb{N}, \mathbf{u}_i \in \mathcal{D}_{main}. \tag{4.2}$$

Two lattice points l_i, l_j are said to be adjacent if $l_i - l_j \in \mathcal{D}_{main}$. The most used lattice types, in the context of PSP, are the cubic and the triangular lattices both in two and three dimensions. When polymers are represented using lattices, the energy model generally considers the interactions only between adjacent objects; for this reason these representations are often called *contact based*. The suitability of a particular lattice to represent the structure of a class of polymers is thus evaluated by comparing, the coordination number of the lattice with the number of energetically relevant contacts in the target class of polymers. In the case of proteins, a contact between two residues is considered energetically relevant if the distance between the central atom of their backbone falls below a meaningful threshold [170, 171]. It has been observed [170] that depending on the threshold used to define a contact, the coordination number of both the three-dimensional cubic (3DC) lattice and the face centered cubic (FCC) lattice resembles the number of contacts between residues in known protein structures. Indeed, it has been shown that by using the FCC lattice or the 3DC lattice, it is possible to reconstruct the backbone of natural protein with high accuracy [172, 173, 174]. The use of higher dimensional lattices has also been proposed [170], but it is still unclear how results obtained within these representations can be related to the structural properties observed in real proteins. In this thesis the focus is only on the 3DC lattice and the FCC lattice. The basis vectors of the 3DC lattice are the orthonormal basis of the three dimensional euclidean space:

$$\mathbf{u}_c^1 = \begin{bmatrix}1\\0\\0\end{bmatrix}; \quad \mathbf{u}_c^2 = \begin{bmatrix}0\\1\\0\end{bmatrix}; \quad \mathbf{u}_c^3 = \begin{bmatrix}0\\0\\1\end{bmatrix}. \tag{4.3}$$

The set of direction vectors of the 3DC lattice is just the set composed of the union of the base vectors and their inverses:

$$\mathcal{D}_{3DC} = \left\{ \bigcup_{i=1}^{3} \pm\mathbf{u}_c^i \right\}. \tag{4.4}$$

An undesirable feature of the cubic lattices, concerning to the representation of linear chain polymers, is that only residues with different parity are allowed to occupy adjacent lattice points; this is often referred as the *parity problem*. The parity problem does not affects the FCC lattice. The basis vectors of the FCC lattice are the following:

$$\mathbf{u}_f^1 = \begin{bmatrix}1\\1\\0\end{bmatrix}; \quad \mathbf{u}_f^2 = \begin{bmatrix}0\\1\\1\end{bmatrix}; \quad \mathbf{u}_f^3 = \begin{bmatrix}1\\0\\1\end{bmatrix}. \tag{4.5}$$

The set of the direction vectors is then defined as:

$$\mathcal{D}_{FCC} = \left\{ \bigcup_{i=1}^{3} \pm \mathbf{u}_f^i, \bigcup_{i \neq j} \mathbf{u}_f^i - \mathbf{u}_f^j \right\}. \tag{4.6}$$

4.4 The hydrophobic polar model

Among the simplified formalizations of the PSP problem, the hydrophobic polar (HP) model postulated by Dill [32], is probably the most studied and the one with the major theoretical impact since the decision problem associated to the HP-PSP problem has been proved to be NP-complete [33, 175]. In the HP model, proteins are represented as strings in a binary alphabet. Each of the twenty amino-acids that compose natural proteins, is classified according to its chemical properties in one of the two classes, *hydrophobic* or *polar*, and represented as a single character in the input string. By convention, in what follows the value 1 is used to represent the hydrophobic residues while the value 0 to represent the polar residues. Moreover, the conformational space is restricted to a Bravais lattice. The following constraints have to be satisfied in order to consider a lattice assignment of the HP string as a valid assignment in the HP model:

1. the function that maps residues to lattice positions must be injective to guarantee that assignments are *self-avoiding walks* (SAW);

2. consecutive positions in the HP string represent stable topological connections and must be in contact in the lattice structure. This is sometimes referred to as the connectivity constraint.

Two different lattice representations of an HP string are possible: the main-chain-only and the explicit side-chains representation; in the former each residue is assigned to a single lattice point, while in the latter each residue occupies two adjacent lattice points, one for the backbone and the other for the side-chain. This thesis focuses on the main-chain-only representation since the use of side-chain model does not increase the modeling accuracy [173] and results from the field of approximation algorithms suggest that it somehow simplifies the computational task of optimization [171]. The energy function of the HP model considers the contacts between non-consecutive hydrophobic positions in the sequence. It is important to notice that, in this representation, *internal residues* have two topological connections and so the maximum number of contacts for each internal residue is $|D_{main}| - 2$ while *terminal residues* have only one topological connection and consequently can be involved in $|D_{main}| - 1$ contacts. The goal of the optimization in HP model is to find a valid lattice assignment of the sequence ι that minimizes the value of the energy function. The HP model can be formalized in the

standard form according to the Definition 4.1, as follows:

$$\begin{aligned}
&find\ x^* \in \underset{x \in \mathcal{A}_\iota}{arg\,min}\, f(x,\iota),\\
&s.t.\\
&\quad -\|x_i - x_j\|_2 < 0, \quad \forall i,j \in \{1,2,...,|\iota|\},\ i \neq j\\
&\quad \|x_i - x_{i+1}\|_2 - \|u\|_2 = 0, \quad \forall i \in \{1,2,...,|\iota|-1\}\\
&\Sigma_{hp} = \{0,1\},\ \iota \in \Sigma_{hp}^*,\ \mathcal{A}_\iota = \mathbb{Z}^{|\iota|\times 3},
\end{aligned} \tag{4.7}$$

where,

$$f(x,\iota) = \sum_{i=1}^{|h|-2} \sum_{j=i+2}^{|h|} -\kappa(x_i,x_j)\cdot\lambda(\iota_i,\iota_j); \tag{4.8}$$

x is the optimization variable that assigns each residue to a lattice point, x_i represents the lattice point assigned to i-th residue; ι is the given HP sequence, ι_i represents the type of the i-th residue; $\lambda(\iota_i,\iota_j)$ is the logical conjunction; and $\kappa(x_i,x_j)$ is binary function defined as:

$$\kappa(x_i,x_j) = \begin{cases} 1 & \text{if } x_i - x_j \in \mathcal{D}_{main} \\ 0 & \text{otherwise.} \end{cases} \tag{4.9}$$

The description of the folding process provided by the HP model includes coarse-grain torsional degree of freedoms and focuses on the hydrophobic effect; therefore, it is an oversimplification of the real energetics described in the previous chapter. The fixed length of the topological connections (due to the lattice representation) represents the high energy barrier associated to bonded interactions, while the self-avoiding constraint is the discrete equivalent of the repulsive term in a Lennard-Jones potential. For this reason, the HP model is too simple to be used directly for predicting the structure of real proteins, nevertheless it is considered a valid benchmark to test new heuristics for the PSP problem. Indeed, the heuristics that have shown good performance in the HP model, such as REMC and SA [35, 176] have proven to be very effective also in off-lattice models [168, 95]. Moreover, some methods for off-lattice PSP [177, 120] combine lattice and off-lattice representations using a more detailed energy representation. The HP model has been also widely used to investigate thermodynamic [40] and kinetic [76] aspects of the folding process. One of the key elements at the basis of the success of the HP model as benchmark for the PSP problem, has been the introduction of *move sets* that allow an efficient exploration of the search space.

4.4.1 Move sets

A move set is a set of rules and atomic operations that allows an explicit treatment of the constraints in the HP-PSP problem by defining a neighborhood relationship between feasible assignments of the optimization variable. Three aspects determine the usefulness of a move set:

- *completeness*, a move set is complete if it guarantees that any feasible assignment of the optimization variable is reachable with a finite number of moves independently from the starting assignment;
- *efficiency*, a move set is efficient if the computational resources required to find and perform a feasible move are in the order of $O(|\iota|)$;
- *full reversibility*, a move set is fully reversible if given a generic couple of neighbor assignments y and z of the search variable and a generic feasible move $mv_i(y) = z$, it is always possible to find a second move in the move set such that $mv_j(z) = y$.

It is important to notice that the efficiency depends also on the number of feasible moves available on average with respect to the theoretical number of moves available and depends on the considered conformation. For this reason, an important empirical metric of performance is the acceptance rate measured over trial simulations. The acceptance rate is the average ratio between the number of feasible moves and the number of attempts to perform a move during the optimization process.

Considering a chronological classification, commonly used move sets can be divided in two generations. The first generation of move sets has been proposed in the early days of lattice simulations of polymers [178, 179, 180] before the introduction of the HP model and includes the *single bead moves*, the *bend moves* and the *pivot moves*. The common characteristic of these move sets is that they require to verify the occupancy of a number of lattice points equal to the number of residues involved in the move.

The single bead moves, proposed in [178], allow the displacement of a single residue r_i to a target lattice point l_{target}. This set includes only two moves: the *corner flip* and the *terminal flip*. A corner flip is feasible for any non-terminal residue r_i if there is a free lattice point l^{target}, such that l^{target} is adjacent to both residue r_{i+1} and residue r_{i-1}. A terminal flip is feasible for a terminal residue x_t, if there is a free lattice point l^{target} adjacent to r_{nhi}. Where $r\_$ denotes, in this case, the residue topologically connected to a terminal residue in the chain.

The bend moves [179], also called crankshaft moves, allow the rotation/reflection of a substructure called bend, in which residues r_i and $r_{i+n_{bm}}$ are in contact. The value of n_{bm} depends on the lattice type, e.g., in cubic lattices $n_{bm} = 3$, while in the FCC lattice $n_{bm} = 2$. The result of a bend move at residue r_i is the displacement of $n_{bm} - 1$ residues

starting at $r_i + 1$ to target lattice points. In a cubic lattice, a bend move is feasible if it is possible to find a couple of free lattice points l^1_{target} and l^2_{target} such that: l^1_{target} is adjacent to x_i and l^2_{target} is adjacent to x_{i+3}. In a triangular lattice, a bend move is feasible if it is possible to find a free lattice position l_{target} adjacent to both x_i and x_{i+2}.

The pivot moves [180], defined by the symmetry group of the considered lattice involve the displacement of a subchain. Each pivot move is performed choosing a pivot residue r_i and applying a symmetry operator to the shortest part of chain departing from r_i. The feasibility is evaluated a posteriori by checking the creation of overlaps.

Both single bead moves and bend moves are fully reversible. On the contrary it is trivial to show that bend move are not complete; for this reason the they must be coupled with other move sets to perform conformational search. The main advantage of these move sets is that they lead to a very small conformational change and, consequently, they keep a relatively high acceptance rate also in the case of compact conformations. It is important to notice that this high acceptance rate come at the price of a slow movement in the conformational space. The pivot moves, on the other hand, are fully reversible, complete and allow a fast movement in the conformational space; unfortunately, the evaluation of their feasibility is computationally expensive and their acceptance rate is very low in the case of compact structures and long sequences [40].

The second generation move sets include *pull moves* [34, 181] and *bond-rebridging moves* [182]. Pull moves can be divided in two subsets, *internal* and *terminal*, depending on whether an internal or a terminal residue is selected to perform the move. An internal pull, involves the displacement of $n_{bm} - 1$ residues to the target lattice points and the pulling of a subchain. One residue r_i is initially selected, as well as one subchain departing from it, either the forward or the backward one (this is the subchain to be "pulled"), the residue topologically connected to r_i on the selected subchain is denoted with r_s while the one on the unselected subchain with r_p. In cubic lattices, the move is feasible if it is possible to find a couple of free lattice points l_1 and l_2 such that: l_1 is either the position x_s of r_s or a free lattice point adjacent to r_i and l_2 is a free lattice point adjacent to r_p. In the first case, the move is completed moving residue r_i to l_2 as in a corner flip. Otherwise residues r_i and r_s are moved to l_2 and l_1 respectively. If r_s is still topologically connected to the following residue along the selected subchain, the move is complete. Otherwise, the connectivity of the chain has been broken and r_s is now a terminal residue. Therefore, the move proceeds and the remainder of the subchain is displaced one residue per time. Each displaced residues extends the chain occupying a position freed by an upstream residue involved in a previous displacement; consequently the break point is pushed each time toward one of the chain terminus. The move is complete when the chain integrity is restored. In triangular lattices, an internal pull move is feasible if it is possible to find a free lattice point l_1 adjacent to both x_i and x_p. The move is performed by moving residue r_i to l_1. If r_i is still topologically connected with r_s the move in complete. Otherwise, it proceeds as in the case of cubic

lattice until the chain integrity is restored. The terminal moves work analogously to internal ones, except from the fact that, in this case, the entire chain is pulled. A terminal move is feasible for the terminal residue r_t, in cubic lattice, if it is possible to find a couple of free lattice points l_1 and l_2 such that l_1 is adjacent to r_t and l_2 is adjacent to l_1. It is then performed moving residue r_t to l_2 and the residue r_s topologically connected to r_t to l_1; the creation of breaks is handled as in the case of internal moves. In order to make the pull move set fully reversible in cubic lattices, terminal moves involving the creation of a bend structure have been removed from the set [183]. A terminal move is feasible, in triangular lattices, if it is possible to find a free lattice point l_1 adjacent to a terminal residue r_t, the creation of breaks is handled as in the case of internal move. The key feature of the pull moves is the completeness, moreover, they produce a global conformational change requiring only a local occupancy verification and this results in a high acceptance rate compared to other move sets. For this reasons pull moves are by far the most used move set in modern SOH approaches to the HP-PSP problem.

A bond-rebridging move involves the modification of the topological connections of the HP chain followed by a relabeling of the residues and requires the alignment between segments of the chain. Three different moves can be performed depending on the relative orientation of the aligned segments and on the residues involved: *parallel*, *anti-parallel*, *terminal*. A parallel move is feasible if it is possible to find two couples of residues (r_i, r_{i+1}) and (r_j, r_{j+1}) such that: x_i is adjacent to x_j and x_{i+1} is adjacent x_{j+1} and $i < j$. The move is performed breaking the topological connections between the residues in each couple and creating two new topological connection between r_i and r_j and between r_{i+1} and r_{j+1}. In the new structure the substring of the HP sequence between r_i and r_j is reversed and for this reason a relabeling is applied to restore the input HP sequence. A anti-parallel move is feasible if it possible to find four couples of residues (r_i, r_{i+1}), (r_j, r_{j+1}), (r_v, r_{v+1}) and (r_k, r_{k+1}) such that: r_{i+1} is adjacent to r_j, r_i is adjacent r_{j+1}, r_v is adjacent to r_{k+1}, r_{v+1} is adjacent to r_k and $i < v < j < k$. The move is performed by breaking the topological connection between residues in each couple and creating new topological connections between adjacent residues of different couples. In the case of anti-parallel bond-rebridging, the relabeling is applied to the segment between i and $k+1$. A terminal bond-rebridging move is feasible if it is possible to find two residues r_t and r_i such that: r_t is a terminal residue and r_i is an internal residue adjacent to r_t. Denoting with r_s the residues topologically connected to r_i in the subchain between r_t and r_i, the move is performed by breaking the topological connection between r_i and r_s and creating a new topological connection between r_i and r_t. In the new structure the subchain between r_t and r_i is reversed and consequently the relabeling procedure is applied to complete the move. Similarly to the pull moves, the bond-rebridging moves allow global conformational changes with a constant number of occupancy verifications. Moreover, they can be implemented efficiently as relabeling operations [40], and their acceptance rate increases in compact structures when the acceptance rate of other move sets decreases. The main limitation of this move set is

the lack of completeness and the high rejection rate it exhibits when applied to loose conformations.

4.4.2 The structure of the search space

The mathematical representation of the search space for the HP-PSP problem depends on the strategy used to perform the search. Here we focus on two major approaches: the neighborhood based approach, and the constructive approach. As explained in Section 2.3, in the neighborhood based approach, it is required the definition of a neighborhood structure that holds for each point in the search space. A convenient way to satisfy this requirement is the use of a neighborhood relationship $\mathcal{N}^k$. In the case of the HP-PSP problem, this relationship considers the number of operations required to move from one state of the system to another, as follows:

$$\mathcal{N}^k(y,z) = \begin{cases} 1 & \text{if } \lambda(y,z) \leq k, \\ 0 & \text{otherwise;} \end{cases} \qquad (4.10)$$
$$y \neq z, \;\; y,z \in \mathcal{A}_\iota, \;\; k \in \mathbb{N}$$

where y and z are two assignments of the optimization variable, while k is a threshold and λ is the minimum number of elementary operations required to transform y in z. In the case of the HP-PSP problem, one or more move sets are used to define the elementary operations at the base of the neighborhood relationship. Given a neighborhood relationship $\mathcal{N}^k$, it is possible to represent the search space of the HP-PSP problem with an undirected graph $G = (V,E)$ such that: each vertex $v \in V$ is in a bijective relationship with an assignments of the optimization variable x, namely $\forall\ x \in \mathcal{A}_\iota\ \exists!\ v \in V$; and that an edges $e \in E$ represent a neighborood relationship between two generic assignments y and z, namely $e \in E\ \exists$ iff $\mathcal{N}_k(y,z) = 1$. Using this representation, the k-neighborhood $\mathcal{N}_y^k$ of a generic assignment y, is the set $\mathcal{V}_y \in 2^{\mathcal{A}_\iota}$ of states whose images in G are connected to v_y, and its size is given by the degree of v_y.

In the case of the constructive approach, the key point is that the search is performed moving over a series of sub-states of increasing size to finally reach a complete state. In this context, the elementary operations must be defined as extensions of a sub-state (impairing the use of move sets). The search space of a constructive method can be represented using a k-parted graph $G' = (V',E')$ such that: $\forall\ x \in \mathcal{A}_\iota,\ \exists!\ L_x \in 2^{V'}$: $L \bigcap \mathbb{V}'_i| = 1, \forall\ i \in \mathbb{N}, i \leq k$, where $\mathbb{V}'_i$ denotes the set of vertex in the i_{th} partition of G'. The overall structure of G' depends on the representation system and lattice used. The most commonly used policy in constructive approaches for the HP-PSP problem is to use direction vectors to extend the partial assignment adding one monomer at time. In this case partitions in G' represent sequence positions in s, and each partition has a number of vertexes equal to the coordination number of the lattice used. The extension is a sequential procedure; therefore G' is directed and has edges only between

the partitions corresponding to consecutive positions.

4.4.3 State of the art of the HP model

In recent years, a great number of methods have been applied to the PSP problem in the HP model using different types of lattice and optimization techniques.

GAs have been applied with success to the HP model until the introduction of second generations move sets [184]. The general scheme of GAs was based on an internal coordinates representation in which each chromosome represented a sequence of direction vectors defining the structure [185]. This representation has the major drawback of requiring an explicit check for the self avoiding constraint and consequently additional terms in the energy function. In recent years, several evolutionary algorithms have been proposed [186, 187] to tackle HP-PSP problem without showing any significant advantage with respect to existing neighborhood based methods.

A hybrid approach based on Evolutionary Monte Carlo method has been proposed by Liang and Wong [188], this was based on the use of Boltzmann distribution for the selection step and the Metropolis-Hasting criterion for the acceptance of the offspring population; moreover, they combined traditional genetic operators with first generation lattice moves.

The Pruned-enriched Rosenbluth method (PERM) [189, 190] has been extensively applied to the HP-PSP problem [191, 192] and is one of the best performing methods in the 3DC lattice in the case of short sequences [35, 40]. The PERM is a constructive Monte Carlo method, also referred as *chain growth* method. Each constructive step of the PERM adds a single residue to one terminal position t_e of the growing HP chain selecting the position between the free lattice points adjacent to that terminus. The probability to select the j-th lattice point is given by its relative Boltzmann weight w_j defined as:

$$w_j = \frac{1}{Z} e^{-\Delta E_j / \kappa_B T}, \quad Z = \sum_{j=1}^{k_{free}} e^{-\Delta E_j / k_B T}. \tag{4.11}$$

In the equation above, k_{free} is the number of free lattice points adjacent to the t_e in the partial assignment during the i-th constructive step, ΔE_j is the energy variation associated to the selection of the j-th position. The Boltzmann weights W_n associated to a chain of length n is the product of the relative Boltzmann weights associated to the positions selected during each extension step:

$$W_n = \prod_{i=1}^{n} w_i \tag{4.12}$$

Two thresholds $W_t^<$ and $W_t^>$ are used to decide whether a partial conformation has to be

enriched or pruned. If $W_i < W_t^<$ a random number $R \in [0,1]$ is sampled and if $R < 1/2$, then the partial assignment is discarded (pruned); otherwise, its weight is doubled. On the contrary if $W_i > W_t^>$ the partial assignment is copied, creating a branching point from which the extension proceeds independently for each branch. In particular, the number of branches is proportional to $W_i / W_t^>$ and the weight of each copy is divided by the number of copies. The algorithm proceed by completing a *round*, this means that all the branches have reached a complete assignment or have been pruned. A depth first policy is used to perform the extension of each branch during a round. After each round, the values $W^<$ and $W^>$ are updated according to:

$$W_{t+1}^> = C\left(\bar{W}/W_0\right)(C_n/C_0) \tag{4.13}$$

and:

$$W_{t+1}^< = 0.2W^>, \tag{4.14}$$

where C, W_0 and C_0 are parameters of the algorithm while $\bar{W}$ and C_n are respectively the mean weight and the total number of the complete assignments achieved so far by the algorithm. In recent versions of the PERM, pruning and enrichment operations are based on unbiased [191] or biased [192] estimators of the utility of the upcoming extension step instead that on W_i; moreover the weights are not adjusted after the enrichment/pruning operations.

An ACO approach has been proposed in [193]. The canonical ACO scheme described in Section 2.3.3 was applied; during each construction step the chain is extended by choosing between the direction vectors to select a lattice point close to a terminal position. If a dead end (a partial assignment in which all the lattice points available for extension are occupied) is created during one of these steps, a roll back procedure is applied and the construction restarts from one of the previous partial assignments.

After the introduction of second generation move sets, several perturbation based approaches have been applied to the HP-PSP problem. Thachuk and coworkers [35] proposed a very efficient REMC algorithm for 3DC lattice combining pull moves and bend moves. A detailed analysis of convergence conditions and optimal parametrization of the SA method in the 3DC lattice using pull moves is provided in [176].

Dotu and coworkers [38] defined a large neighborhood search (LNS) method [194] based on constraint programming to refine structures obtained by means of a local search heuristic and tested this approach on several FCC instances.

Variants of the TS method have also successfully applied to 2DC lattice [34] and FCC lattice [41, 42]. Moreover, in [41, 42] a modified scoring function and specialized procedures have been introduced to promote the creation of hydrophobic cores. These methods achieved the best results reported so far for many FCC sequence of significant

length.

The WLS method has been applied along with a combination of all the second generation move sets and pivot moves leading to very promising results on 3DC lattices [195, 196, 197, 198, 40]. Another non-Markovian Monte Carlo approach called heuristic energy landscape paving has been applied to the HP-PSP problem in 3DC lattice [199]. In this case a memory structure is used to reduce the acceptance probability of assignments with highly sampled energy values. The main difference with respect to WLS is that this method preserves a bias toward low energy states in its acceptance criterion.

Zang and coworkers [36] combined SA and PERM, leading to an effective method named fragment regrowth via energy-guided sequential sampling (FRESS). In this approach the PERM method is used in place of a move set to generate the candidate assignment x' at each iteration of SA. In particular, the PERM is used to regrowth an internal segment of the chain. A roll back procedure, with depth first policy is used to handle the creation of dead ends and to guarantee the chain closure. This method leaded to very good results on several instance in the 3DC lattice.

Finally, it is important to notice that Backofen and Will [200] defined an exact constraint programming approach, called HPstruct, built on top of a database of precomputed H-cores. H-cores are compact lattice assignment that represent all the possible positionings that allows a given number of hydrophobic residues to create a target number of contacts. The constraint programming method performs an exhaustive search to verify the capability of the input sequence to accommodate a given H-core. This method is sequentially applied to H-cores with decreasing number of contacts until a fitting H-core is found, providing a solution. The HPstruct method is able to efficiently find solutions for HP-PSP instances of moderate size ($|\iota| < 100$) on both 3DC and FCC lattices and it has been extensively used to assess the converge of SOHs for the HP-PSP problem. Nevertheless the computation of H-cores is itself a NP-hard problem limiting the applicability of this approach.

4.5 The MDG problem

As explained in the previous chapter NMR exploits the magnetic properties of the nucleus of isotopes (as ^{1}H, ^{13}C and ^{31}P) to identify spatial neighborhood relationships between chemical groups i.e., given in the form of a matrix of inter-atomic distances. When this technique is applied to molecules of significant size and with complex 3D shape, such proteins, the resulting distance matrix is both sparse and noisy due to the distance dependency of the Overhauser effect and to technical limitations, described in Section 3.4. The MDGP consists in reconstructing the 3D structure of a molecule starting from its (sparse) distance matrix; the MDGP problem is a special case of the Distance Geometry Problem (DGP) [22] in which the distance matrix is obtained from

NMR analysis. The peculiarity of the MDGP is the availability of additional constraints concerning the distances in the 3D structure, which can be assumed considering the chemical and physical properties of the class of molecules under investigation. The MDGP can be formulated in the standard form as a feasibility problem:

$$\begin{aligned} &\textit{find } x \\ &s.t. \\ &\|x_i - x_j\|_2 - e^u_{ij} \leq 0, \quad \forall e^u_{ij} \neq 0 \\ &e^l_{ij} - \|x_i - x_j\|_2 \leq 0, \quad \forall e^l_{ij} \neq 0 \\ &x \in \mathbb{R}^{|\iota|\times 3},\ E^u \in \mathbb{R}_+^{|\iota|\times|\iota|},\ E^l \in \mathbb{R}_+^{|\iota|\times|\iota|}. \end{aligned} \tag{4.15}$$

In the definition above, x is a matrix of atomic three dimensional coordinates, E^l and E^u are two sparse matrices containing respectively upper and lower bounds for the distance restraints obtained in a NMR experiment. In this context, the zeros in the restraint matrices denote the lack of information. This problem was proven to be NP-hard [45], by reducing a 1-dimensional MDG problem to the SUBSETSUM problem. Most of the methodological studies that deals with the MDG problem focuses on the special case in which exact distance restraints are available [201]. If this special case is considered, $E^l = E^u$ so a single restraint matrix E is used and the Definition (4.15) can be simplified replacing each couple of inequality constraints with a single equality constraint. This problem is still NP-Hard if the restraints matrix E is sparse.

4.5.1 State of the art of the approaches for the MDG problem

Several approaches to MDGP have been proposed in recent years. Dong and Wu introduced a linear time algorithm, called "geometric buildup", to solve the 3D-DGP when the exact value of distances between all pairs of atoms are given [202]; recently, this approach has been extended in order to obtain an approximate solution for the MDGP with noisy distance values and sparse matrices [203, 23]. The main limitation of the geometric buildup strategy is that in the case of sparse matrices and, in particular, when some atoms have less than four distance constraints, this method is unable to find any solution. However, to overcome this limitation, it is possible to consider additional distance constraints arising from structural features of proteins, or using optimization algorithms [25], to reconstruct the complete molecular structure from a partial substructure obtained with the geometric buildup algorithm.

A branch and prune algorithm was proposed in [204, 24]: by exploiting additional constraints about the protein structures, this method considers a discrete search space in which the amino-acids can be placed only in two different positions with respect to their precursor in the protein structure. In general, this algorithm has an exponential time complexity; however, it is able to efficiently find solutions for some instances that satisfy particular structural properties.

There exist two approaches based on graph embedding [205] in 3D Euclidean space, able to deal with both noisy data and sparse matrices. The first one, called ABBIE algorithm [206], exploits a divide and conquer strategy and structural rigidity [207]; this method first identifies subproblems (i.e., subsets of nodes) that can be solved with an exact algorithm, then it applies a global optimization algorithm to combine partial solutions. The second one, called EMBED algorithm [208], uses the measured distances to derive a set of lower and upper bounds for all other distances; this requires the identification of the shortest path between each couple of nodes in a particular bigraph in order to derive triangle inequality limits [209]. A local optimization strategy is then applied to refine the solution obtained from the complete bounds set.

Finally, the DGSOL algorithm [201, 210] combines a methodology to select good starting points for the optimization process with the Gaussian smoothing and continuation strategy [211], a technique used to reshape the objective function during the optimization. So doing, a gradient minimization can be applied to the obtained smooth function in order to refine the structure and to minimize the constraints violation. A memory function is used to select new starting points when it is not possible to further improve a solution with the procedure described above. The main limitation of DGSOL is that it provides only approximate solutions in presence of noisy information and sparse matrices.

5 The local landscape mapping strategy

This chapter provides a detailed description of two new SOHs for the HP-PSP problem developed in our laboratory, both these methods are based on the same fundamental idea, referred to as *local landscape mapping* (LLM) strategy. The first section of the chapter, introduces a new perturbation system in the HP model, in particular, this consists in a FA-like perturbation and requires the definition of a specialized energy function in order to select legal HP assignments. The second section of the chapter discusses the fundamental ideas of the LLM strategy. The third section describes a first implementation of the LLM strategy, LLM_{mem}. The results of LLM_{mem} on small benchmark instances of HP-PSP problem in 3DC lattice are presented both using the FA-like perturbation system and a standard move sets. To evaluate the significance of this results they are compared with those achieved by means of SA and ACO. The last part of the chapter describes a different implementation of the LLM strategy,LLM_{LS} . The results of LLM_{mem} for several benchmark instances of the HP-PSP problem using pull moves are provided both in the 3DC and FCC lattices and compared with those achieved by state of the art methods.

5.1 A fragment assembly-like representation in the HP model

In Section 4.4, two different strategies to generate assignments in the HP model have been discussed: the move sets used in neighborhood based methods and the selection of lattice main directions vectors used in constructive approaches. In this section, a new system is introduced, with the purpose of providing an HP benchmark framework closer to the FA strategy used for off-lattice PSP. The representation we propose is based on the use of small embeddings that we call fragments in analogy to off-lattice models. These fragments are combined during the search process in order to produce a complete embedding of the input sequence. In particular, the 150 fragments that cover all the non-overlapping conformations of an HP string of length four in the 3DC lattice were computed. Each fragment determines the local structure of a triplet of residues starting at the insertion point and also the position of the first residue in the subsequent triplet.

An HP sequence of length N is thus encoded as a sequence of $\frac{N-1}{3}$ integer indexes, in this case each index identifies the fragment used to represent the structure of the i_{th} triplet in the HP sequence. Special fragments of reduced length are exploited to represent the last portion of the sequence if $N-1$ is not a multiple of three. To exclude global translational and rotational degrees of freedom and mirror image duplicates from the search space only the six fragments in figure 5.1 have been admitted at the first position.

This FA-like system allows both the overall structure and the energy value of an assignment to change significantly after that a single perturbation is applied; the same holds also for the fragment assembly in off-lattice models since many atomic positions are modified in a single fragment exchange move [119] and the energy function is often a highly non-linear function of the distances and angles between atoms or pseudo-atoms [95]. On the contrary, the self-avoiding neighborhood structure defined with the move sets (excluding pivot moves and bong-rebridging moves) induces a smooth energy landscape in which only a reduced number of hydrophobic contacts is altered after each move [34, 176]. Although some important improvements have been achieved with empirical scoring functions [212], such a smooth landscape is not reproducible in off-lattice models of protein folding due both to the strongly perturbative behavior of the fragment assembly procedure and to the intrinsic nature of the folding process [213].

It is important to notice that, opposite to the case of move sets, there is no guarantee that the assignments generated through the fragments exchange procedure are feasible, since the creation of overlaps between residues of different triplets is possible. Moreover every fragment insertion that modifies the starting position of the upstream fragment implies a translation of all the upstream fragments.

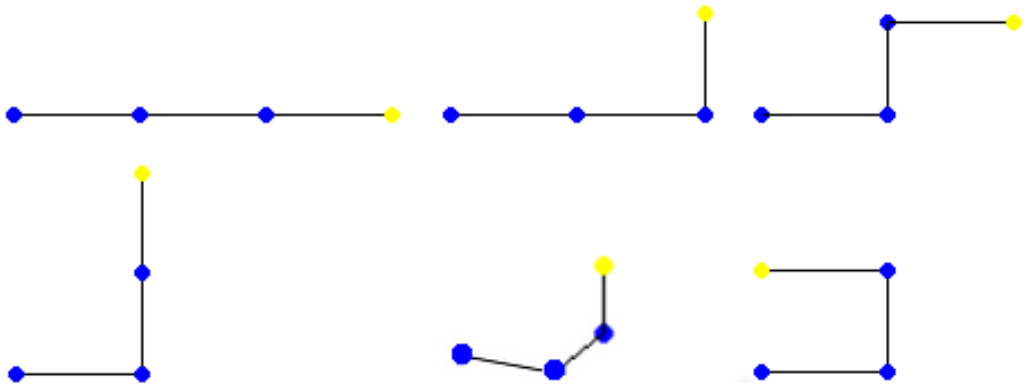


Figure 5.1: Allowed fragments for the first triplet
The six fragments allowed to represent the first triplet of a sequence in the FA-HP representation system. The position of the first residue in the second triplet is shown in yellow.

In order to promote the selection of feasible assignments when the FA-HP representation is used we defined a specific scoring function, shown in Equation 5.1; this discourages the generation of overlapping walks and improves the performance of methods that require heuristic choices.

$$E_{FA}(x,\iota) = -\left[\left(\frac{S_{HH}}{1+S_{OV}}\right)^2 + \frac{1}{1+S_{HP}} + \frac{2N_{HH}}{1+S_{HD}}\right], \tag{5.1}$$

where x is the optimization variable, ι represent the target HP string, N_{HH} is the upper bound of the number of hydrophobic contacts defined in [214]; while S_{HH} represents the canonical objective function of the HP problem, defined in Equation (4.8). The other terms indicated as S_x are scoring functions with the general form defined below:

$$S_x(\cdot) = \sum_{i=1}^{N-3} \sum_{j=i+3}^{N} f(\cdot) \tag{5.2}$$

where f is a function that defines the specific type of score. S_{OV} uses $f = \omega$ shown in Equation 5.3; it counts the number of overlapping positions:

$$\omega(x_i, x_j) = \delta(x_i - x_j), \tag{5.3}$$

where δ denotes the Dirac delta function. To consider the overlaps at the boundary between subsequent fragments the ω function is also computed between residues i and $i+2$ at the boundary positions. S_{HP} uses $f = \phi$ shown in Equation (5.4); it counts the contacts between hydrophobic and polar positions (H-P contacts), it was introduced to prevent the formation of undesired H-P contacts; a similar term was used in [191].

$$\phi(x_i, x_j, \iota_i, \iota_j) = \kappa(x_i, x_j)\,\delta(\iota_i \oplus \iota_j), \tag{5.4}$$

where κ has been defined in Equation (4.9), while $\oplus$ denotes the logical exclusive disjucntion. Finally the S_{HD} uses $f = \zeta$ in Equation (5.5) and it was introduced to bias the early steps of constructive methods toward compact solutions.

$$\zeta(x_i, x_j, \iota_i, \iota_j) = \begin{cases} \|x_i - x_j\|_2 & \text{if } \gamma_{ij} \cdot \lambda(\iota_i \iota_j) \neq 0 \\ 0 & \text{otherwise} \end{cases}, \tag{5.5}$$

where,

$$\gamma_{ij} = \begin{cases} 1 & \text{if i and j have different parity} \\ 0 & \text{otherwise} \end{cases} \tag{5.6}$$

while λ denotes, once again, the logical conjunction. An interesting aspect of the scoring function in Equation (5.1) is the presence of a quadratic term. Quadratic and higher degree energy terms are often exploited in scoring functions of off-lattice methods for

PSP [119, 215].

5.2 Fundamental ideas

By analyzing the state of the art of the PSP problem presented in Section 4.1, two general considerations arise. First, neighborhood based methods [35, 40, 38, 41, 95] seem to perform better than both population based methods [193, 187] and methods based on constructive approaches [191, 193]; second approaches that favor exploration, like those based on the WLS and methods that keep memory of good solutions to prevent re-sampling [38, 41, 42], perform better than methods that rely only on the direct search for ground states [36, 199]. Moreover, all the best performing methods in the 3DC lattice [191, 35, 36, 40] are in the Monte Carlo family, while in the FCC lattice the best performers are based on a combination of TS and large neighborhood search (LNS). The use of a large neighborhood structure allows a method to escape local minima through the generation of neighbors assignments outside of the basin of attraction of the trapping minimum. This is particularly appealing in the case of the PSP problem since, as discussed in section 3.3.1, the majority of the search effort in this problem is spent in escaping local minima associated to non-native compact structures [76]. The formation of these structures is often associated to a significant reduction in the energy [40] and consequently methods based only on the Metropolis-Hasting criterion often get trapped in this sub-optimal assignments. Unfortunately, also the LNS strategy has a major drawback: indeed, as a consequence of the reduction in the conformational entropy associated to the folding process, the number of assignments within a given energy bin E_i decreases exponentially with the energy itself [184, 40]. Therefore, the probability to find a good assignment in the extended neighborhood using a blind approach is very low. The approach found in literature [38] exploits constraint programming to efficiently explore the large neighborhood. The idea at the basis of the LLM strategy proposed here, is to combine a variable neighborhood structure with the rejection sampling scheme of a typical Monte Carlo method i.e., SA. In particular, the method works extending the neighborhood of the SA when the acceptance rate become low. Moreover, a biasing strategy is adopted to perform the search in the extended neighborhood in order to increase the quality of generated neighbor assignments. In this thesis two biasing strategies have been tested, memory based and local search based. The first uses a memory structure that associates a desirability value to the elementary operations that define the neighborhood relationship. The second is based on a specialized local search algorithm designed to search over the set of compact structures.

5.3 The memory based biasing strategy

The first version of the LLM method [43, 44] uses a memory structure borrowed from ACO meta-heuristic to bias the search over the extended neighborhood, for this reason

it will be referred as LLM_{mem}. This is a population-based iterative method that follows the general scheme of SA and uses a pheromone structure to build a desirability map of the neighborhood learning from the rejected samples. The structure of the pheromone matrix depends on the considered problem, on the type of elementary moves used to perform the search and on the amount of computational resources available. The key point, to keep in mind in what follows, is that, in LLM_{mem}, the pheromone values considers the relationships between the elementary operations and the current assignment x^t. Two examples of how this relationship can be represented are given in the next section.

The method starts with a random assignment x^0 of the search variable and the uniform initialization of the pheromone values. Then, each iteration is divided into three phases: *expansion*, *evaluation* and *update*. During the expansion phase, a fixed number of neighbors of x^t, specified in the parameter ν and referred to as the *candidate assignments*, are independently generated. In LLM_{mem} two different policies can be alternatively applied to generate the candidate assignments:

- the uniform sampling in the 1-neighborhood;
- the pheromone based sampling in the extended neighborhood.

In the first case, each of the candidate assignments is selected according to:

$$P_n(\hat{x} = z) = \begin{cases} \frac{1}{|\mathcal{N}^1_{x^t}|} & \text{if } z \in \mathcal{N}^k_{x^t} \\ 0 & \text{otherwise} \end{cases} \tag{5.7}$$

where $|\mathcal{N}^1_{x^t}|$ is the cardinality of the 1-neighborhood of x^t. In the second case, the candidate assignments are generated according to a biased distribution based on pheromone values. In particular, a sequence of elementary operations o is used to generate neighbors in the extended neighborhood. The parameter $\sigma_{\texttt{MAX}}$ controls the cardinality of o. A value σ in the interval $[2, \sigma_{\texttt{MAX}}]$ is selected with uniform probability; then σ choices are sequentially performed to obtain o. Each elementary operation is selected according to the following criterion:

$$P_d(o_i = j) = \frac{\theta\ (t_j) + (1-\theta)\ - \eta_j}{\sum_{b \in Op} \theta\ (t_{br}) + (1-\theta)\ - \eta_b} \tag{5.8}$$

where $P_d(o_i = j)$ is the probability that the j_{th} elementary operation is selected in the i_{th} position of the sequence o and Op represent the set of operation that define the neighborhood, i.e. pull moves, single bead moves or fragments exchange. The choice between the two procedures for neighbors generation is probabilistic and depends on the number of rejections achieved in previous iterations as follows:

$$P_{phe}(x^i) = \frac{1}{1 + e^{-a}} \tag{5.9}$$

where $P_{phe}(x^i)$ is the probability of using the pheromone criterion to generate the i_{th} neighbor state, and a is defined as follows:

$$a = -6\alpha + \frac{\#rej}{\alpha} \tag{5.10}$$

where $\#rej$ is the number of iteration that have been spent without accepting a new state and $\alpha \in \mathbb{N}$ is a parameter of the algorithm. The evaluation phase of LLM is composed of an optional local optimization step and an acceptance step. In the local optimization steps candidate assignments are used as starting point for a local optimizer analogously to what described in Section 2.3 for ACO and GRASP. The acceptance step considers only the best of the candidate assignments $\hat{x}^*$ and it is based on the Metropolis-Hastings criterion in Equation (2.11). Finally, during the update phase, the pheromone values corresponding to elementary operation that have been selected to generate each candidate assignment $\hat{x}^i$, are modified according to:

$$\tilde{t}_{ij} = \frac{1}{2} t_j (1-\rho) + \frac{1}{2} t_j (1 + c(x^i, x^t)\ \rho), \tag{5.11}$$

where $\tilde{t}_{ij}$ is the updated pheromone value, ρ is a parameter controlling the learning rate, and $c(x^i, x^t)$ is the *relative value* of x^i with respect to x^t, that in the case of HP-PSP has been defined as follows:

$$c(x^i, x^t) = \frac{E(m^i)}{-1 + E(m^t)} \tag{5.12}$$

If a candidate assignment has been accepted during the evaluation phase, then, all the pheromone values are updated as follows:

$$\hat{t}_{ij} = w\ \tilde{t}_j + (1 - w) \tag{5.13}$$

where w is a parameter of the algorithm that controls the propagation of the pheromone bias between subsequent samples in the trajectory. The update phase described above creates the local energy landscape map associating each elementary operation to a pheromone level proportional to the relative value of the assignment that arise from the application of that elementary operation. In addition, each accepted assignment inherits a portion of the bias from its precursor in the trajectory basing on the assumption that the neighbor assignments share at least a portion of their local landscape.

5.3.1 The pheromone structure

As anticipated in the previous section in LLM_{mem} the pheromone is used to associate a desirability value to elementary operations used to define the neighborhood. The elementary operation in the FA-like framework is the exchange of a single fragment. The peculiarity of LLM_{mem} is that it considers desirability with respect to the current

assignment. Therefore, a pheromone value has been assigned to each couple of fragments, except for fragments used to represent the same triplet (since they are mutually exclusive); this allows to assess the desirability of an incoming fragment with respect to each possible combination of fragments in the current assignment. A graphical representation of this pheromone model is provided in Figure 5.2. In particular, the desirability t_j of fragment j used to represent the i_{th} triplet is computed as follows:

$$t_j = \sum_{p \neq i} \tau_{jp}, \tag{5.14}$$

where τ_{jp} is the pheromone value associated to the couple composed of the fragment j and the fragment representing the p_{th} triplet in the current assignment. A visual representation of this "aggregate" desirability is provided in Figure 5.3. With the aim of reducing the computational cost of pheromone based fragment selection in LLM_{mem}, the selection criterion in Equation (5.8) was restricted to select only between the fragments associated to a randomly chosen triplet.

In the case of move sets, a pheromone value was associated to each possible move and the pheromone based move selection is performed between all the feasible moves.

An important difference between the two pheromone models discussed above is that, in the case of the FA-like representation, the fragments are combinatorial blocks used to generate assignments; consequently, the pheromone model is associated to the graph in Figure 5.2 that represents the search space of the problem. On the contrary, move sets define only the neighborhood of a specific assignment; consequently the pheromone model is associated to a transient structure.

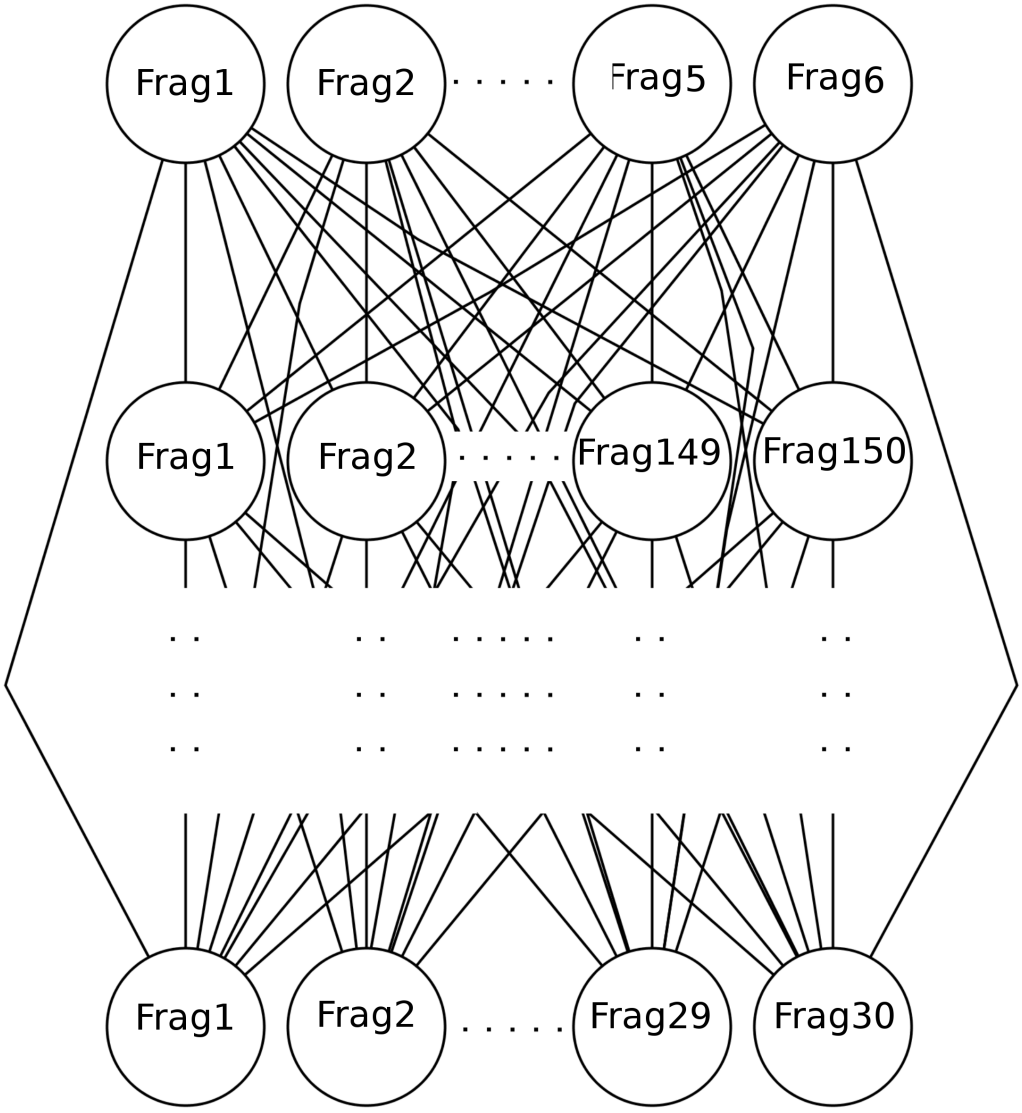


Figure 5.2: Pheromone model for the FA-like perturbation system.
A graph-based representation of the pheromone used for the FA-like perturbation system. The complete k-parted graph represents the search space HP-PSP based on FA-like perturbation system. Each partition represents a triplet of positions. Each node represents a fragment. Assignments of x are paths that include exactly one node from each partition. A pheromone value τ_{ij} is associated to each edge in this graph.

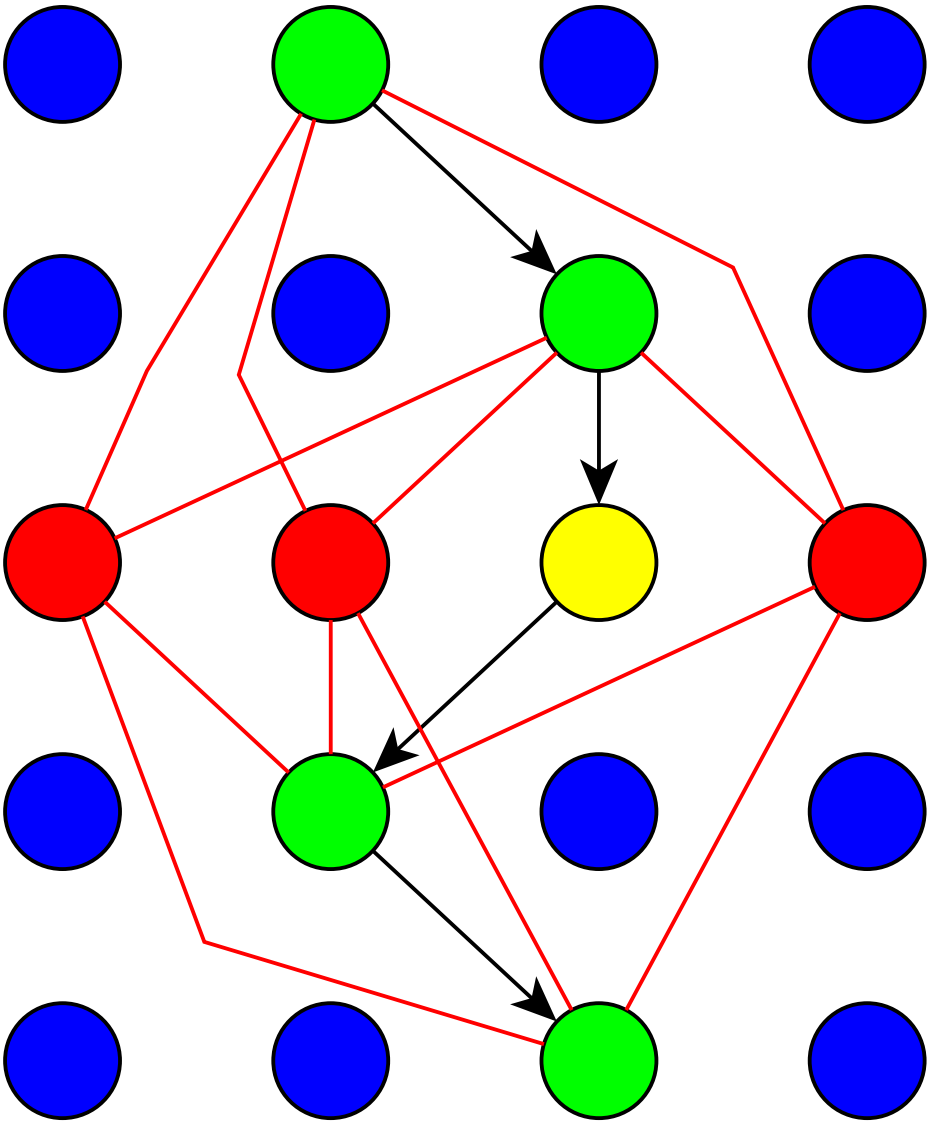


Figure 5.3: Pheromone-based neighbors selection for the FA-like perturbation system. For the sake of clarity, only four fragments have been represented in each partition. The path defining the current assignment is represented with arrows. The yellow node represents the fragment that will be replaced during the considered perturbation step. Green nodes represent fragments in the current assignment that will not be modified. The red nodes represent candidate fragments for the this perturbation step. The red edges represent all the pheromone values considered in equation (5.8). Each red edges is associated to a pheromone value τ, while the aggregate pheromone value t_j of node j is the sum of all the edges connecting node j with the current assignment.

5.3.2 Parameterization and auxiliary procedures

The parameter sets for all the methods were selected tuning each parameter independently on a random sequence of length 48. It is important to notice that a detailed analysis of the optimal value of the parameter T_0 of SA method in the pull move framework is provided in [176]. Nevertheless, it has been observed empirically that due to the limited amount of time considered in our tests a lower value of T_0 with respect to the one proposed [176] leads to better results (data not shown). The values for all parameters are reported in Table 5.1.

The heuristic function used in the pheromone based selection of both ACO and $LLM_(mem)$ is computed according to Equation (5.1). No heuristic function was used in LLM_{mem} in the case of the move sets. The move sets used to perform the tests includes pull moves and bend moves. Only feasible moves are considered during each neighbor generation step.

A local optimizer has been included in both LLM_{mem} and ACO in the case of the FA-like perturbation system. In particular, this method performs an exhaustive search over the 1-neighborhood of the input assignment, if the best assignment found in the neighborhood improves the value of cost function with respect to the input assignment, it is used as new input for a new round of local search; otherwise the method returns the best assignment found.

	FA-like			Move sets	
Parameter	LLM_{mem}	SA	ACO	LLM_{mem}	SA
T_0	5.0	5.0		3.5	3.5
ν	5		5	1	
σ	[2,4]			1	
α	1			5	
θ	0.5		0.5	1	
ρ	0.2		0.1	0.5	
w	0.9			0.2	
local opt	yes	no	yes	no	no

Table 5.1: Parameters lists and settings
Parameterization used to perform computational tests in with the FA-like perturbation scheme and in the HP-PSP using move sets.

5.3.3 Computational results

This section reports the results achieved by LLM_{mem} on a broadly used [188, 193, 35, 36, 176, 39, 40] benchmark set for HP model in the 3DC lattice. This set of instances, often referred to as *Harvard instances*, was introduced in [216] and it is available at [217]. All the sequences in the benchmark set a have length of 48, and the global optimum

for each of them was computed by means of the hpstruct method [218]. Two different computational tests have been performed: the first was aimed to assess the significance of LLM_{mem} using the FA-like perturbation system; the second was aimed to analyze how the FA-like perturbation system affects the performance of different methods with respect to the move sets. In the first test, results of LLM_{mem} have been compared to those obtained with two well established SOHs, in particular SA and ACO. In the second test, LLM_{mem} and SA have been applied using move sets. It is important to notice that ACO is a constructive heuristic and for this reason it cannot be included in test 2 (see Section 4.4). In both tests a time elapsed termination criterion have been applied setting a time limit of $2'$ of CPU time for the execution of each run. The performance measure used in this test is the average best energy computed over several independent runs. This is computed storing the best energy value computed in each independent run and then computing the average of all the stored values. All the tests were carried out on a desktop machine equipped with an Amd Phenom II X6 1090T processor and 4GB of RAM. For each run we stored as result the lowest energy structure found.

Test 1. In Table 5.2 are shown the best and the average results obtained over 50 run for each instance using the FA-like perturbation system.

ID	E_{min}	SA	ACO	LLM$_{mem}$
H1	-32	**-26**(-23.7 ±1.23)	**-29**(-26.9 ±0.79)	**-31** (-28.5 ±0.92)
H2	-34	**-27**(-23.9 ±1.60)	**-29**(-27.0 ±0.97)	**-32** (-29.5 ±1.18)
H3	-34	**-28**(-25.1 ±1.86)	**-28**(-26.7 ±0.67)	**-32** (-29.6 ±1.38)
H4	-33	**-28**(-24.1 ±1.27)	**-29**(-26.4 ±0.89)	**-31** (-29.1 ±0.97)
H5	-32	**-28**(-25.1 ±0.84)	**-29**(-26.5 ±0.73)	**-31** (-28.6 ±0.95)
H6	-32	**-24**(-23.0 ±1.37)	**-28**(-25.6 ±0.96)	**-30** (-27.6 ±0.99)
H7	-32	**-26**(-23.6 ±1.83)	**-28**(-26.5 ±0.81)	**-31** (-28.4 ±0.90)
H8	-31	**-27**(-24.1 ±1.40)	**-29**(-26.0 ±0.95)	**-29** (-27.9 ±0.71)
H9	-34	**-28**(-25.2 ±1.88)	**-30**(-27.8 ±0.76)	**-32** (-29.8 ±0.93)
H10	-33	**-27**(-24.3 ±1.88)	**-29**(-27.3 ±0.98)	**-31** (-29.6 ±0.95)

Table 5.2: Results of the LLM_{mem} method: fragment assembly-like
Results of different heuristics in the 3DC lattice using the FA-like perturbation system. The optimal energy values E_{min} for each benchmark sequence are shown in the second column. In bold is reported the best value obtained in the fifty runs. In round brackets are shown means and standard deviations of the best energy value obtained in each run.

The LLM_{mem} outperforms competitors on all the instances, while ACO performs better than SA. The local optimizer plays an important role in determining the good performance of both ACO and LLM but nonetheless, LLM shows an improved capability in generating starting points with respect to ACO. Moreover the energy function introduced for FA-like proved to be very effective in pruning invalid states since no invalid state has

been found as final result of a run in this test. The low performance of SA method in the case of the FA-like perturbation system is probably due to the large neighborhood size induced by the fragments or to the increased energy differences between neighbor states resulting from the modified energy function.

Test 2. In Table 5.3 are shown the best and the average results obtained over 50 run for each instance using move sets.

ID	E_{min}	SA	LLM$_{mem}$
H1	-32	**-32** (-30.62 ±0.69)	**-32** (-30.60 ±0.69)
H2	-34	**-33** (-31.66 ±0.66)	**-33** (-31.66 ±0.56)
H3	-34	**-34** (-32.18 ±0.74)	**-34** (-32.22 ±0.76)
H4	-33	**-32** (-31.38 ±0.56)	**-33** (-31.40 ±0.76)
H5	-32	**-32** (-30.78 ±0.54)	**-32** (-30.38 ±0.56)
H6	-32	**-32** (-30.84 ±0.56)	**-32** (-30.88 ±0.59)
H7	-32	**-31** (-29.95 ±0.56)	**-31** (-30.13 ±0.54)
H8	-31	**-30** (-29.38 ±0.60)	**-31** (-29.56 ±0.54)
H9	-34	**-34** (-32.04 ±0.60)	**-33** (-32.00 ±0.57)
H10	-33	**-33** (-31.20 ±0.60)	**-32** (-31.12 ±0.65)

Table 5.3: Results of the LLM_{mem} method: move sets
Results of LLM_{mem} and SA in the 3DC lattice using move sets. The optimal energy values E_{min} for each benchmark sequence are shown in the second column. In bold we present the best value obtained in the fifty runs. In round brackets are shown means and standard deviations of the best energy value obtained in each run.

The optimal values for many instances have been obtained by both LLM_{mem} and SA. On the contrary, none of the methods was able to reach the global optimum for any instance using the FA-like perturbation system in the considered amount of time. Moreover mean values obtained with move sets are always better that those obtained with the FA-like perturbation system. These results suggest that the FA-like perturbation system provides somehow a more challenging benchmark than the canonical HP-PSP. This results are not unexpected since the possibility to generate overlapping structures extend the search space. Moreover, the neighborhood size and the energy differences between neighbor solutions are also increased in the case of the FA-like perturbation system. For what concerns SA both these variations are known to affect negatively the performance [65, 176]. It is important to notice that the performance of SA and LLM_{mem} in this test is the same. In our opinion, the lack of improvement of LLM_{mem} method in this case depends on two main factors: the change in set of feasible moves and the small energy difference between neighbor solutions. The former reduces the utility of information obtained from the previous states since when a pull move is applied, the

set of available moves in the resulting assignment is significantly different from the initial one; this is particularly relevant in the case of compact structures in which only a reduced number of moves is feasible. The latter affects the generation of pheromone bias since, with small energy differences between neighbor solutions, more iterations are required to create significant difference in pheromone levels.

5.4 The local search based biasing strategy

Although result obtained by the LLM_{mem} using the FA-like perturbation system are promising, this approach presents several limitations:

- it requires the tuning of a high number of free parameters;
- the pheromone based neighbor selection can be computationally expensive depending on the selected pheromone model and heuristic function;
- the pheromone model is not able to collect information if the neighborhood has a flat energy landscape.

Moreover, the lack of performance improvement with respect to SA observed in the test based on move sets cannot be underestimated. Indeed, this is the standard framework for the PSP problem. Consequently, a new method has been designed aimed to overcome these limitations while keeping the fundamental idea of *LLM*.

In a recent paper [42], Rashid and coworkers introduced a new method, called *spiral search*, that is based on the concept of *hydrophobic-core directed local search*. The basic idea of this local search strategy, is to assign a priority to moves that reduce the euclidean distance of hydrophobic residues from the geometric center of the hydrophobic residues, called hydrophobic-core center (HCC). In particular, during each iteration, all the moves that reduce the distance between an hydrophobic residue and the HCC are listed. Then, the move associated to the residue farther from the HCC is performed. The process iterates updating the HCC after each move, until no further reduction of the distances between hydrophobic residues and the HCC is possible. Only corner flip moves are used to perform the local search. Several diversification strategies based on pull moves are applied in the spiral search method to prevent stagnation of the local search method. This strategy proved to be effective, indeed the method achieved excellent performance on several benchmark instances of the HP model in the FCC lattice [42].

A similar idea has been exploited in the definition of the local search-based LLM (LLM_{LS}). The LLM_{LS} is based on three components: a *putative core generator*, a *local search method*, and a SOH. An high level description of LLM_{LS} is the following:

1. the putative core generator is used to provide a target *putative hydrophobic core* for the LS procedure;
2. the LS is used to modify the current assignment in order to promote, as much as possible, the creation of the hydrophobic contacts specified in the putative hydrophobic core;
3. the MCM method is applied in order to improve the quality of the assignment produced by the local search method.

The three steps reported above are iterated until a termination criterion is reached, i.e., a given number of iteration has been performed.

5.4.1 The putative hydrophobic core

The concept of putative hydrophobic core (PHC) is the major innovation of our approach with respect to existing techniques. A PHC is an object specifying the couples of hydrophobic residues that creates contacts in a putative compact structure. Denoting with I_{HH} the set of indexes of hydrophobic residues in the given instance ι, a PHC is defined as follows:

- it is a vector with a length equal to the number of hydrophobic position in the given instance ι;
- each element of a PHC is a subset of I_{HH} with an arity bounded in the range $[0, |D_{main}| - 2]$;
- the i_{th} element of the PHC is associated to the i_{th} hydrophobic position in ι.

In other words, the PHC associates a set C_h of indexes to each hydrophobic residue r_h and each index in C_i corresponds to a hydrophobic positions in ι. The relationships defined in the PHC are symmetric, this means that if the index j is in C_i then, the index i is in C_j. The indexes of residues $i-1$ and $i+1$ are implicitly included in C_i since they are in contact with residue i in any feasible assignment. Due to the parity problem, in the case of cubic lattices, an index j can be included in C_i only if i and j have a different parity. A set C_i in the PHC is said to be *full* when it includes the maximum number of indexes $|C_i| = |D_{main}| - 2$. The putative core generator is used to define random PHCs. In particular, the following scheme is adopted:

1. initialize the set I^*_{HH} of available hydrophobic residues including all the hydrophobic residues;
2. if I^*_{HH} is not empty, then sample a random hydrophobic residue r_i from I^*_{HH} with uniform probability; otherwise the PHC is complete;

3. if it exists a residue $r_j \in I^*_{HH}$, allowed to interact with r_i, such that $j \notin C_i$, then proceed to the next step; otherwise remove r_i from I^*_{HH} and go to step 2;
4. j is inserted in C_i and i is inserted in C_j;
5. if C_j is full, then remove r_j from I^*_{HH};
6. if C_i is full, then remove r_i from I^*_{HH} and go to step 2; otherwise go to step 3;

5.4.2 The local search method

The local search algorithm receives in input an assignment x^0 of the optimization variable and a random PHC. It is iteratively applied in order to minimize the distances between the hydrophobic residues that are expected to create contacts according to the PHC. In particular, during each iteration an exhaustive search over the 1-neighborhood of the current assignment x^t is performed and the elementary operation that leads to the greater reduction of the function in Equation (5.15) is selected to generate the assignment of the next iteration x^{t+1}.

$$f_{LS}(x, \iota, PHC) = \sum_{i \in I_{HH}} \sum_{j \in C_i} \|x_i - x_j\|_2. \tag{5.15}$$

The local search iterates until it is not possible to further reduce the value of f_{LS}. It is important to notice that there is no guarantee that the contacts specified by the PHC can be realized in the considered instance, since constraints arising from the chain structure and the presence of polar residues are completely ignored in the PHC definition.

5.4.3 The optimization step and parameterization

In LLM_{LS}, the SA method is used for the optimization step. In particular, a re-annealing protocol has been applied since this proved to be more effective in finding solutions for small HP sequence with respect to a single annealing cycle of increased length (data not shown). The only parameters required by the LLM_{LS} concern the SOH method used in the optimization step. In the case of SA with re-annealing we have only two parameters: the starting temperature of each cycle T_0 and the number of annealing cycles N_{cycles}. For all the results presented in the next section this values have been set to 5.0 and 2 respectively. The use of the value $T_0 = 5.0$ instead of $T_0 = 3.5$ previously adopted is due to a change in the implementation of the neighbor generation step in SA. Indeed, the results of the LLM_{mem} were achieved considering only the feasible moves during neighbors generation, while the results of LLM_{LS} have been obtained following the suggestions in [40]. Therefore unfeasible moves are also considered in neighbors generation and, in the case that an unfeasible move is selected, the iteration terminates. As a consequence, in the new implementation the temperature decrease is not always associated to the sampling of a neighbor assignment, resulting in a “faster cooling“.

5.4.4 Computational results

This section presents the results achieved by LLM_{LS} on several sets of HP instances both in 3DC and FCC lattices using only the pull moves. The results have been grouped according to the lattice used and to the length of the considered sequences. All the benchmark sequences used are available in appendix A. In order to assess the significance of our results, they have been compared with those achieved by several state of the art methods. Two different kind of performance measure have been considered in order to match published material: the average first hit time and the average best energy (described before). The average first hit time measures the average time, computed over several independent runs, required to reach an assignment with an energy value equal or better than a target value. The computation of the average first hit time for values close to minimum energy of a given instance is extremely time demanding in the case of long sequences; therefore, this measure has been used only to evaluate results for short sequences (Test 1). For statistical significance the interquartile mean of first hit time have been reported instead of the mean as suggested in [40]. All the tests were carried out on a desktop machine equipped with an Amd Phenom II X6 1090T processor and 4GB of RAM.

Test 1. Table 5.4 shows the performance in term of average first hit time on the Harvard instances [216] in the 3DC lattice achieved with REMC [35], WLS [40] and LLM_{mem}.

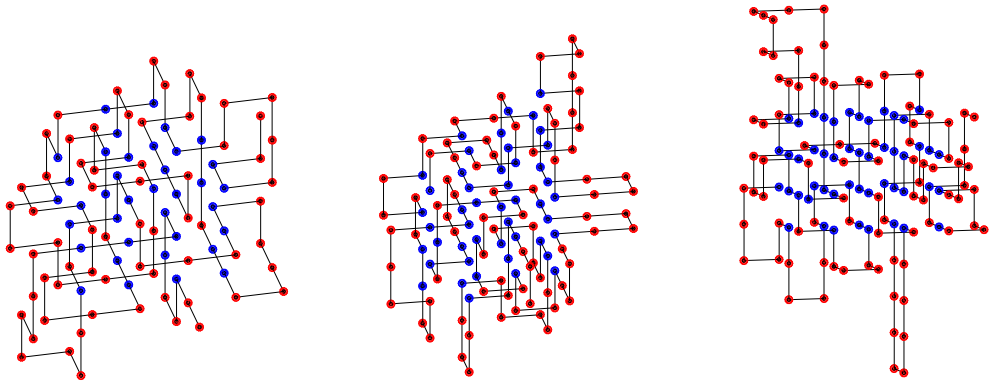
ID	E_{min}	REMC	WLS	LLM_{LS}
H1	-32	0.22	0.32	**0.08**
H2	-34	1.82	0.84	**0.34**
H3	-34	0.71	0.68	**0.24**
H4	-33	0.53	0.59	**0.20**
H5	-32	0.28	0.23	**0.10**
H6	-32	0.40	0.39	**0.13**
H7	-32	1.24	1.58	**0.68**
H8	-31	0.56	0.58	**0.18**
H9	-34	2.64	3.10	**1.66**
H10	-33	0.44	0.98	**0.20**

Table 5.4: Results of the LLM_{LS} method: CBC lattice 1
Average first hit time in minutes needed to reach the optimal energy value of the Harvard instances in the 3DC lattice. In bold is reported the best value obtained for a given instance. WLS: results reported from [40], values computed over 20 independent runs. REMC: code from [35] tested on our machine, values computed over 100 independent runs. LMM_{LS}: values computed over 100 independent runs.

The code for the REMC is freely available [219] under the GNU general public license version 2. Consequently, in order to provide an unbiased comparison it has been

compiled from the source and executed on our local machine. No change to the parameterization reported in [35] has been applied. The results reported for WLS have been taken from the manuscript, in this case some of the difference in the performance can be due to the difference in the hardware. Another aspect that should be taken into account is that WLS is not designed specifically for the search of low energy states. Nevertheless, LLM_{LS} is the best performing methods in all the instances considered in this test and in many cases it is more than 2 times faster than the other methods.

Test 2. Table 5.5 shows the results for longer benchmark sequences introduced in [220] on the 3DC lattice. These instances have been widely used as benchmark in previous studies [36, 40]. PDB identifiers are used to label this sequences since they were generated converting real protein sequence in the HP alphabet basing on chemical properties of residues [220]. The results reported for WLS have been taken from the



(a) (b) (c)

Figure 5.4: Best results on biological instances.
The best models achieved over the 30 runs for each of the biological instances considered in test 2 are shown. (a) An assignment with energy -57 for 3CYT. (b) An assignment with energy -73 for 7RSA. (c) An assignment with energy -80 for 2SNS. Images obtained with MATLAB™.

manuscript [40]. Due to time limitations, only LLM_{LS} and WLS have been considered in this comparison. Moreover, first hit times for LLM_{LS} have been extrapolated from the results of runs performed with the time elapsed termination criterion, fixing the time limit to 10 minutes. The reported values have been computed dividing the overall running time allocated for a given sequence by the number of runs that visited an assignment with energy equal or lower than the target. In the case of 7RSA all the runs visited an assignment with energy equal or lower of the target, consequently the reported time value is an overestimate of the average first hit time. The best results achieved for each sequence in this set are shown in Figure 5.4.

ID	E_{targ}	WLS	LLM_{LS}
3CYT	-57	0.93	0.55
7RSA	-71	0.06	<0.17
2SNS	-80	2.98	1.66

Table 5.5: Results of the LLM_{LS} method: CBC lattice 2
Average first hit time in minutes needed to reach the target energy value on HP instances obtained from biological sequences [220] in the 3DC lattice. WLS: results reported from [40] values computed over 20 independent runs. LMM_{LS}: results extrapolated from 30 independent runs, see details in the text.

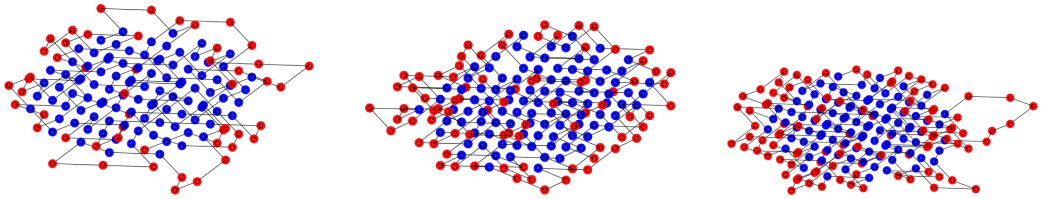
Test 3. Table 5.6 shows the results achieved on the last set of benchmark instances [38] in the FCC lattice.

ID	E_{min}	SS-tabu	LS-tabu	time/run	LLM_{LS}	time/run
F90_1	-168	-166 (-166)	-164 (-160)		**-168** (-167.7 ±0.52)	
F90_2	-168	-164 (-164)	-165 (-158)		**-168** (-167.9 ±0.35)	
F90_3	-167	-165 (-165)	-165 (-159)		**-167** (-166.9 ±0.25)	
F90_4	-168	-165 (-165)	-165 (-159)		**-168** (-167.8 ±0.76)	
F90_5	-167	-165 (-165)	-165 (-159)	120	**-167** (-166.9 ±0.40)	
S1	-357	-355 (-347)	-351 (-341)		**-356** (-355.0 ±1.05)	
S2	-360	-354 (-347)	-355 (-343)		**-360** (-358.3 ±1.25)	
S3	-367	-359 (-350)	-355 (-340)		**-366** (-361.3 ±3.20)	10
S4	-370	-358 (-350)	-354 (-343)		**-368** (-364.4 ±2.92)	
F180_1	-378	-357 (-340)	-338 (-327)		**-368** (-361.9 ±2.94)	
F180_2	-381	-359 (-345)	-345 (-334)		**-374** (-366.7 ±3.11)	
F180_3	-378	-362 (-353)	-352 (-339)	300	**-373** (-369.8 ±2.64)	
R1	-384	-359 (-345)	-332 (-318)		**-372** (-365.6 ±3.84)	
R2	-383	-358 (-346)	-337 (-324)		**-374** (-365.4 ±4.11)	
R3	-385	-365 (-345)	-339 (-323)		**-375** (-368.4 ±3.40)	

Table 5.6: Results of the LLM_{LS} method: FCC lattice
Average best fitness and best overall energy for several instances in the FCC lattice. For each instance the best overall energy value and the average best fitness (in round brackets) are shown. In bold is reported the best value obtained for a given instance. SS-tabu: results reported from [42], values computed over 50 independent runs. LS-tabu: results reported from [42], values computed over 50 independent runs. LLM_{LS}: values computed over 30 independent runs.

In this case, the results achieved with LLM_{LS} have been compared to those achieved with the spiral search (SS) method [42] and the large neighborhood search (LS-tabu) method [38]. Results for both these method have been taken by the recent manuscript of Rashid et al. [42]. In this test the performance of all the methods was assessed basing on the average best energy measure and the overall lower energy found. The LLM_{LS} method

achieve the best performance in all the considered instances both in terms of average best fitness and overall lower energy. Moreover, it is the only method that is able to find an assignment in the solution set at least for some of the instances. Unfortunately, no mention to the hardware used to achieve the results can be found in [42]. Nevertheless, in the paper of [38] similar results to those presented in [42] were achieved with LS-tabu using a system comparable to the one used in this study. Moreover, the time limit for the LLM_{LS} method was fixed to 10', while the time limit for the other methods was of 120' or 300' depending on the length of the target sequence. Basing on these observations, it is possible to say that the LLM_{LS} represents an improvement with respect to state of the art methods for PSP in the FCC lattice. The best result obtained for some of the instances considered are shown in Figure 5.5.



(a) (b) (c)

Figure 5.5: Results on large instances in the FCC.
The best models achieved over the 30 runs for some of the large instances($|\iota| > 150|$) considered in the FCC lattice are shown. (a) An assignment with energy -360 for S2. (b) An assignment with energy -373 for F180_3. (c) An assignment with energy -374 for R2. Images obtained with MATLAB™.

6 The evolutionary springs swarm method

This chapter provides a detailed description of the *evolutionary spings swarm* method (ESSM) for the MDG problem with exact distance constraints, developed in our laboratory. The first section of the chapter is dedicated to the description of fundamental ideas considered to design the method. The second section provides a detailed description of the algorithm, while remaining of the chapter is spent discussing the applied parallelization, the parametrization and the results obtained on 9 synthetic instances of the MDG problem.

6.1 Fundamental ideas

As discussed in Section 4.5.1, several different approaches to the MDG problem have been proposed in recent years, but they all suffer from limitations. For instance, the geometric buildup [202] is unable to find a solution to the problem for some cases of sparse distance matrices; the branch and prune algorithm [204, 24] has an exponential computational time; ABBIE [206], EMBED [208] and DGSOL [210] algorithms allow to obtain only approximate solutions to the MDG problem. Moreover, all these approaches strongly relies on the mathematical structure of the MDG problem and, for this reason, they hardly can be adapted to include other sources of information in the optimization process.

This kind of versatility is generally provided by SOHs since they rely only on the definition of an objective function that allows the incorporation of additional terms and the evaluation of their effect on the performance. To the best of our knowledge, the ESSM is the first attempt to solve the MDG problem by means of SOHs alone. It has been designed to overcome some of the limitations of the existing approaches and, at the same time, to provide a versatile optimization scheme that can be easily extended to allow the use of model evaluation terms.

As discussed in Section 4.5, an assignment x of the optimization variable of the MDG

problem can be encoded as a matrix of 3D coordinates, representing the positions of all atoms of protein in the Euclidean space. This representation can be exploited by a traditional evolution-based methodology such as GAs, whose operators are specifically designed to work on candidate solutions encoding real values [221]. Even though GAs might be feasible methodology for MDG problem, swarm intelligence techniques like PSO are generally more suitable than GAs, since they natively optimize real-valued problems [103]. Nevertheless, the crossover operator of GAs – which exchanges the genetic material of two promising individuals to create an improved offspring generation – is an elegant and powerful means to obtain a recombination of individuals and a better exploration of the search space. For these reasons, ESSM combines the swarm-based optimization of PSO with the crossover capabilities of GAs.

6.2 The objective function

As discussed in Section 2.1.5, the definition of an objective function is needed in order to tackle feasibility problems by means of SOHs. This function is used to quantify the degree of constraint violations in a candidate assignment associating an optimal value to feasible assignments. The objective function $f_\varepsilon \mathbb{R}^{|\iota|\times 3} \times \mathbb{R}^{|\iota|\times|\iota|} \rightarrow \mathbb{R}$ adopted in ESSM for the MDG problem, measures the *error* associated to a structural model in terms of average constraint violations; it is defined as follows:

$$f_\varepsilon(x,E) = \sum_{i=1}^{|\iota|} \frac{\sum_{j=1}^{|\iota|} g_{ij}}{|k_i|}, \tag{6.1}$$

where

$$g_{ij} = \begin{cases} 0 & \text{if } e_{ij} \text{ is not given} \\ |\delta_{ij}| & \text{otherwise,} \end{cases} \tag{6.2}$$

and k_i is the set of indexes of atoms with non-null entry e_{ij} in the distance matrix E, $|\dot{|}$ denotes the absolute value, while $\delta_{ij} = \|x_i - x_j\|_2 - e_{ij}$. According to this objective function, feasible assignments of the MDG problem have a value of zero, while unfeasible ones have a positive value that increases proportionally to the degree constraint violations. The same function has been used in [25] and a closely related is at base of the DGSOL method [201, 210]. It is important to notice that this objective function applies only to the special case of the MDG problem with exact distance restraints, discussed in Section 4.5. This is the only case of the MDG problem considered in this thesis, but a potential extension of our method to the general case will be discussed in the next chapter.

6.3 The algorithm

The ESSM is an hybrid population-based SOH that combines aspects of GAs and PSO; each individual in the population represents an assignment of the optimization variable, that, in the case of the MDG problem, is a candidate structure for the target protein. The ESSM is organized on two layers as shown in figure 6.1. The outer layer (the *GA-layer*) manage the exchange of partial assignments between individuals in the population by means of the crossover operator and propagates promising assignments by means of selection. The inner layer exploits the self-organizing behavior of swarms (hereby called the *PSO-layer*) to modify individuals generating new assignments. The pseudo-code of the ESSM is shown in Algorithm 1.

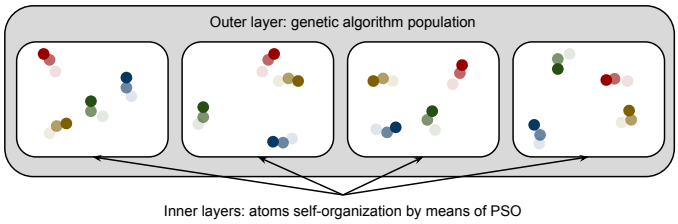


Figure 6.1: ESSM overview
Schematization of the two-layer hybrid methodology. In the outer layer, a population of candidate solutions exchange promising substructures exploiting GAs crossover, while each candidate solution evolves in the inner layer by means of a PSO-like scheme.

6.3.1 GA-layer

The GA-layer of ESSM instantiates Q independent assignments of the optimization variable, which constitute the population of the GA. The peculiarity of the GA-layer in ESSM, is that it does not exploit the mutation operator, which is conceptually realized by the PSO-layer, as described in the next section. Consequently, it performs only initialization, tournament selection and crossover.

The initialization procedure assigns a position to each atom in each individual. In the case of the MDG problem, we defined a strategy that allows to tune the dimension of the search space according to the number N_{aa} of amino-acids in the target protein (which is known *a priori* from the sequence). This is possible since the upper bound of the radius of gyration was identified as $N_{aa}^{3/5}$ Å[222]. Starting from this observation, the best setting for the size of the search space was empirically found to be $4 \cdot N_{aa}^{3/5}$Å for each dimension in the 3D Euclidean space. Consequently, the individuals are initialized by sampling $|\iota| - 1$ random points within a sphere of radius $2 \cdot N_{aa}^{3/5}$Å centered at the origin of the Cartesian axes. The value $(0,0,0)$ is, instead, assigned to vector x_1 and kept fixed during the optimization. For reasons that will be clarified in the next section, this choice provide an anchor point and reduce the probability to have useless translations of the

Algorithm 1 ESSM

Require: input: Q, q, $I_{\mathtt{MAX}}$, I_χ, $v_{\mathtt{MAX}}$, α;
1: $P_0 \leftarrow$ *initialize_population*(Q);
2: *iterations* $\leftarrow 0$;
3: **while** *iterations* $< I_{\mathtt{MAX}}$ **do**
4: **for all** $p \in P_0$ **do**
5: $p \leftarrow$ *execute_PSO* $-$ *layer*$(p, I_\chi, v_{\mathtt{MAX}}, \alpha)$;
6: **end for**
7: $p_{\mathtt{BEST}} \leftarrow$ *find_best_individual*(P_0);
8: $P_1 \leftarrow \{0\}$;
9: $P_1 \leftarrow P_1 \bigcup \{p_{\mathtt{BEST}}\}$;
10: **while** $|P_1| < |P_0|$ **do**
11: *iterations* $\leftarrow$ *iterations* $+ 1$;
12: $p_{\mathtt{DON}} \leftarrow$ *tournament_selection*(P_0, q);
13: $P_1 \leftarrow P_1 \bigcup \{p_{\mathtt{DON}}\}$;
14: $p_{\mathtt{RAND}} \leftarrow$ *random_uniform_selection*(P_0);
15: $\sigma \leftarrow$ *select_substructure*$(p_{\mathtt{DON}})$;
16: $p_{\mathtt{OFF}} \leftarrow$ *insert_substructure*$(\sigma, p_{\mathtt{RAND}})$;
17: $P_1 \leftarrow P_1 \bigcup \{p_{\mathtt{OFF}}\}$;
18: **end while**
19: $P_1 \leftarrow$ *adjust_chirality*(P_1);
20: $P_0 \leftarrow P_1$;
21: *iterations* $\leftarrow$ *iterations* $+ 1$;
22: **end while**
23: **return** *find_best_individual*(P_0)

entire structure, reducing also the computational effort needed to manage boundary conditions.

Among the existing strategies, in this work we exploit the "tournament selection", line 13 Algorithm 1, in which a subset of $2 \leq q \leq Q$ individuals of population Pop^0 is randomly chosen and the individual having the best fitness is copied into the new population Pop^1.

The crossover operator, defined in ESSM, implements the exchange of substructures among individuals. A substructure $\sigma = i, \ldots, k$, $|\sigma| \leq |\iota|$, in our definition, is a set of indexes of atoms in x, the associated geometry can be specified by the set of coordinates vectors x^σ. It is important to notice that the positions selected to define σ do not need to be consecutive rows in x. Moreover, the geometry of the substructure depends only on relative positioning of vectors in x^σ and is invariant under rigid body transformations. In ESSM, a directional crossover has been applied to preserve good individuals. One of the parents, the *donor*, denoted with $p_{\mathtt{DON}}$, is selected through the tournament method and used to define the geometry $x^{\sigma\mathtt{DON}}$ for a given set of indexes σ. The other parent, the *acceptor*, denoted with $p_{\mathtt{ACC}}$, is selected random and used to define the geometry

in the remaining part of the structure. The offspring is thus obtained by replacing the geometry of the acceptor $x^{\sigma\mathtt{ACC}}$ in the positions specified by σ with the geometry of the donor $x^{\sigma\mathtt{DON}}$.

The crossover operator has been divided in two steps, corresponding to line 15 and 16 of Algorithm 1.

The first step, coded by the *select_substructure* procedure, is used to select a promising substructure from $p_{\mathtt{DON}}$. Denoting with $f(x^\sigma, E^\sigma)$ the error of x^σ with respect to the subset of distance restraints E^σ relative to atoms in σ, we want that $f_\varepsilon(x^\sigma, E^\sigma) < f_\varepsilon(x, E)$. In order to satisfy this requirement, the following greedy algorithm has been used:

1. create the set σ^1 composed of a single coordinates vector x_i randomly chosen from x;
2. extend σ^t by adding at each iteration the vector $x_{\mathtt{m}} = \underset{i\in\{1,\ldots,|\iota|\}}{\arg\min} \max_{s\in\sigma^t} g_{is}$;
3. if $\max_{s\in\sigma^t} g_{ms}$ exceeds a given threshold $\varphi_{\mathtt{min}} < f_\varepsilon(x, E)$, go to step 4, otherwise, go to step 2.
4. return σ^t .

The value $\varphi_{\mathtt{min}}$ is defined as $\varphi_{\mathtt{min}} = \min\{\varphi_i \mid i = 1, \ldots, N\}$, where $\varphi_i = \frac{1}{N}\sum_{j\neq i} g_{ij}$.

The second step, coded by the *select_substructure* procedure, is aimed to replace $x^{\sigma\mathtt{DON}}$ with $x^{\sigma\mathtt{ACC}}$ reducing the potentially deleterious impact of crossover between individuals with different global orientation. This problems arises from the fact that individuals are independently modified in the PSO-layer. Consequently, two assignments y and z can be completely different in terms of Cartesian coordinates even in the case of individuals coding for the same geometry. A simple copy of the substructure coordinates, from the donor to the acceptor, has a low probability to succeed and often leads to the creation of artifacts, i.e. the atoms of the exchanged substructure are placed far away from the receiving structure. For this reason, a rigid body structural alignment is performed between the exchanged substructures by means of a local optimization method [1]. This guarantees that the quality of the offspring individual depends on how the geometry of the substructure provided by the donor "fits" with that of the remaining part of the structure and not on the relative orientations of the parents. The procedure *select_substructure* is based on the minimization of the root mean square distance

[1] Global optimization methods coupled to local search are called memetic algorithms [223], thus in the paper describing the algorithm [46] we defined our methodology as a Memetic Hybrid PSO plus GAs (MemHPG). We believe that that the term, *evolutionary springs swarm method*, provides a better understanding of the behavior of the proposed method, and for this reason it has been adopted in this thesis.

(RMSD) between the geometry of the compared substructures:

$$RMSD(x^{\sigma\texttt{DON}}, x^{\sigma\texttt{ACC}}) = \sqrt{\frac{1}{|\sigma|}\sum_{i=1}^{|\sigma|} \| x_i^{\sigma\texttt{DON}} - x_i^{\sigma\texttt{ACC}} \|_2^2}, \tag{6.3}$$

and it can be summarized as follows:

1. copy $x^{\sigma\texttt{DON}}$ in $\tilde{x}^{\sigma}$ and compute the centroids $\tilde{c}^{\sigma}$ and $c^{\sigma\texttt{ACC}}$, where $c^{\sigma} = \frac{1}{|\sigma|}\sum_{i=1}^{|\sigma|} x_i^{\sigma}$;
2. apply to each vector in $\tilde{x}^{\sigma}$ a translation $TRS(\tilde{x}_i^{\sigma}) = \tilde{x}_i^{\sigma} - \tilde{c}^{\sigma} + c^{\sigma\texttt{ACC}}$ in order to have $\tilde{c} = c^{\sigma\texttt{ACC}}$
3. select among the six global rotational and translational degrees of freedom the one associated to the greatest infinitesimal variation in $RMSD(\tilde{x}^{\sigma}, x^{\sigma\texttt{ACC}})$ and the direction that reduce the RMSD along it; set the step size η^i to a predefined value $\eta_{\texttt{MAX}}$;
4. move $\tilde{x}^{\sigma}$ with a fixed step size η^i along the selected degree of freedom and direction;
5. if the RMSD has decreased after step 4, repeat step 4, otherwise go to step 6;
6. if a given time has elapsed, go to step 7, if $|\eta^i|$ is lower than a threshold λ, go to step 3, otherwise, go to step 4 and set $\eta^{i+1} = -\eta^i/2$;
7. replace $x^{\sigma\texttt{ACC}}$ with $\tilde{x}^{\sigma}$ in $p_{\texttt{ACC}}$.

The ESSM stops when a user-defined termination criterion is met, i.e., after a fixed number of iterations $I_{\texttt{MAX}}$.

6.3.2 PSO-layer

The PSO-layer is the most peculiar part of the ESSM. It has been designed to manage the movement of atoms in each individual and it is based on the concepts of inertial movement and communication between simple agents, resembling in this the PSO method; the main difference is that in ESSM each particle represents just a portion of the optimization variable and not a complete assignment as in PSO. In the case of MDG, in particular, at any given time t, each particle a_i is associated to the position $x_i(t)$ of a single atom in the structure and to a velocity $v_i(t)$. Consequently, each particle represents a solution for the sub-problem of identifying an optimal spatial positioning of a single atom relatively to the others. The restraint matrix is used to define a set of connections between couples of atoms. Connected atoms are pushed toward the direction that reduce the violation of the associated constraints modifying their velocity, Figure 6.2 provides two examples of this mechanism, represented in the x-y projection

plane for the sake of simplicity. We named *aggregate attractor*, denoted by $\mathbf{h}_i(t) \in \mathbb{R}^3$, the direction resulting from the combined effect of all the springs acting on a given particle a_i. The value $\mathbf{h}_i(t)$ is calculated as follows:

$$\mathbf{h}_i(t) = \sum_{j \neq i} \gamma_{ij}(\mathbf{x}_j(t) - \mathbf{x}_i(t)), \tag{6.4}$$

where

$$\gamma_{ij} = \begin{cases} 0 & \text{if } e_{ij} \text{ is not given} \\ \frac{\delta_{ij}}{\|x_i(t) - x_j(t)\|} & \text{otherwise} \end{cases}. \tag{6.5}$$

In Equation (6.4), γ_{ij} is used to weight the attraction/repulsion between couple of atoms. In our definition, of $\gamma_i j$ more weight is given to the short distance constraints. This assigns a priority to the creation of the local structure leading to better result with respect to a purely proportional approach. Such a choice can also be justified by the fact that the precision of NMR measure decreases with the distances between considered atoms. It is worth noting that the aggregate attractor $\mathbf{h}_i$ can be seen as a linear combination of $|k_i|$ "global" attractors of the canonical PSO scheme, each one acting on particle $\mathbf{a}_i$ with a different social factor equal to δ_{ij}. Figure 6.3 provide a visual representation of the aggregate attractor for an atom with two distance restraints.

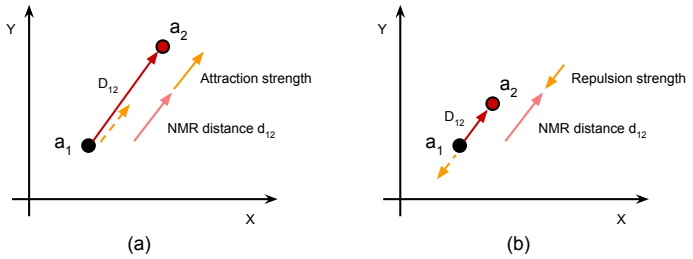


Figure 6.2: Spring-like behavior of restraints in ESSM
Example of the attraction/repulsion mechanism of our modified PSO. For the sake of clarity, only the vectors for particle $\mathbf{a}_1$ are shown. (a) When the distance between two atoms (the red arrow between $\mathbf{a}_1$ and $\mathbf{a}_2$) is larger than the one measured by NMR (pink arrow), the atoms attract each other (dashed yellow arrow). (b) When the distance between the two atoms is smaller than the distance measured by NMR, the atoms act as repulsers.

The new velocity $\hat{v}_i(t+1)$ for particle a_i, in ESSM, is given by:

$$\hat{\mathbf{v}}_i(t+1) = w \cdot \mathbf{v}_i(t) + \mathbf{r} \circ \mathbf{h}_i(t), \tag{6.6}$$

where $\mathbf{r}$ is a vector of random numbers uniformly sampled in [0,1] and w is the inertia of the particle. The Equation (6.6) resembles the velocity update of a canonical PSO shown in Equation 2.19, the only difference being that in ESSM there is a single attractor. An

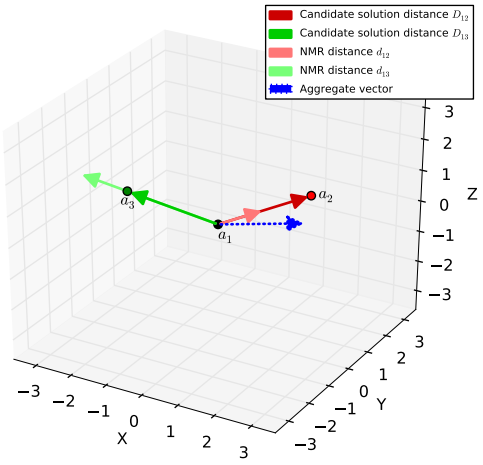


Figure 6.3: The aggregate attractor

Example of calculation of the aggregate attractor for particle $\mathbf{a}_1$, in a 3-atoms system. The length of the red arrows represents the distance between particles $\mathbf{a}_1$ and $\mathbf{a}_2$ according to the candidate solution (dark red) and to NMR data (light red): since the latter is shorter than the former, $\mathbf{a}_2$ acts as an attractor for $\mathbf{a}_1$. The length of the green arrows represents the distance between particles $\mathbf{a}_1$ and $\mathbf{a}_3$ according to the candidate solution (dark green) and to NMR data (light green): since the latter is longer than the former, $\mathbf{a}_3$ acts as a repulser for $\mathbf{a}_1$. The resulting aggregate attractor $\mathbf{h}_1$ is represented by the blue vector. The same process is applied to particles $\mathbf{a}_2$ and $\mathbf{a}_3$ (not shown here).

idealized representation of the PSO-layer is that of a stochastic springs system. In this representation we have that each couple of atoms for which a distance restaints exists is connected by a spring. The equilibrium length of each spring is given by the value of the observed distance for the considered couple of atoms. The aggregate attractor is the resultant of the forces percieved by each atom due to the action of the springs connected to it. The inertia is used to prevent the creation of a chaotic behavior, while the randomness **r** allows particles to escape local optima and break periodic behaviors.

Analogously to PSO, a maximum value $v_{\texttt{MAX}}$ is used in ESSM to clamp particles velocity. Consequently after each velocity update the putative velocity $\hat{\mathbf{v}}_i(t+1)$ is adjusted as follows:

$$\mathbf{v}_i(t+1) = \frac{\hat{\mathbf{v}}_i}{\|\hat{\mathbf{v}}_\mathbf{i}(t+1)\|_2} \min(\|\hat{\mathbf{v}}_i(t+1)\|_2, v_{\texttt{MAX}}).$$

The new position $x_i(t+1)$ is finally obtained by adding $v_i(t+1)$ to $x_i(t)$. During the last generations of ESSM, the finer positioning of atoms in the candidate structures requires smaller and more controlled movements with respect to the initial phases. For this reason, our methodology self-adapts the $v_{\texttt{MAX}}(t)$ value as follows:

$$v_{\texttt{MAX}}(t) = \begin{cases} \alpha \cdot v_{\texttt{MAX}}(t-1) & \text{if } \varepsilon^*(t) > \varepsilon^*(t-1) \\ v_{\texttt{MAX}}(t-1) & \text{otherwise,} \end{cases}$$

where

$$\varepsilon^* = \min_i \frac{\sum_{j=1}^{|\iota|} g_{ij}}{|k_i|}$$

and $\alpha \in (0,1)$ is the velocity adaptation factor. The iterative update of velocity vectors, calculated according to the aggregate attractor, allows the set of atoms to self-organize in a single optimal position.

During each iteration of the GA-layer the PSO-layer is executed a fixed number of times I_χ for each individual. It is important to notice that, the velocities are preserved between subsequent executions of the PSO-layer since they are stored in the individual. The behavior of the PSO-layer is summarize in Algorithm 2. The ESSM allows communication of the two layers through the modification of both the positions and the velocities and this communication is bidirectional. Indeed, after each crossover operation the velocities for the atoms in the inserted σ are set to zero.

6.4 Chirality

An important aspect that must to be considered in ESSM concerns the *chirality* of the target protein. An object in a metric space is said to be chiral if it lacks mirror symmetry [224]; namely, it cannot be aligned with itself using rotations and translations after

Algorithm 2 execute_PSO-layer

Require: input: p, I_χ, $v_{\texttt{MAX}}$, α;
1: $iterations \leftarrow 0$;
2: **while** $iterations < I_\chi$ **do**
3: **for all** $a_i \in p$ **do**
4: $h_i \leftarrow compute\_aggregate\_attractor(a_i, p)$;
5: $update\_velocity(a_i, h_i, v_{\texttt{MAX}})$;
6: $update\_position(a_i)$;
7: $store\_updated\_particle(a_i, p)$;
8: **end for**
9: $velocity\_adaptation(p, v_{MAX}, \alpha)$;
10: $iterations \leftarrow iterations + 1$;
11: **end while**
12: **return** p;

that an odd number of reflections has been performed [225]. Many organic molecules, including the amino-acids, are chiral. In particular, chirality arises when a carbon atom is bound to four different chemical groups[226], this carbon atom is called stereo-center. Each stereo-center can be configured in two different but isometric conformations, called *enantiomers*, that can not be exchanged without breaking chemical bonds.

In the case of amino-acids the two different enantiomers are labeled, L and D. If all the composing amino-acids share the same label, as in the case of natural proteins where only L-amino-acids are found [5], the resulting protein adopts one of the two possible isometric conformations. Since the information contained in the distance matrix is not sufficient to discriminate between a correctly reconstructed molecule and a molecule with a different chirality, a specific procedure is required to impose the specific L conformation to each amino-acid. This procedure was applied during each generation of the GA-layer, line 19 in Algorithm 1. The subset of atoms that can lead to a wrong reconstruction can be identified *a priori* by analyzing carbon atoms and their bound chemical groups. For each candidate solution we identify the substructures whose chirality is not correct, and we modify them by means of matrix operations implementing a single reflections with respect to a specific plane.

6.5 Parallelization

One of the most appealing aspect of ESSM is that it can be implemented in a highly parallel way. Among the available choices, the parallel version was implemented to leverage Graphic Processing Units (GPUs) horsepower, by using Nvidia's CUDA (Compute Unified Device Architecture) [227]. CUDA relies on a single instruction multiple data (SIMD) paradigm. The CUDA framework is one of the few available for general purpose GPU (GP/GPU) programming. It offers an API programmable in the

C/C++ languages that handles the compilation of programs producing binaries for both CPU and GPU.

In order to exploit the parallel architecture, a program in CUDA must be organized following a specific hierarchy. At the top level of the hierarchy there is the *kernel* that represent the whole program. The kernel is organized in a *grid*, the grid is divided in several execution units called *blocks*. Finally, each block is itself composed of *threads*. At the beginning of the execution this logical structure is used to distribute the program in the available hardware resources. In particular the blocks that compose the program are distributed in several hardware units, called *streaming multiprocessors*. Each streaming multiprocessor is thus responsible for the independent execution of a given number of blocks and is itself divided in several computing unit called *cores*. This organization allows a two-folds parallelization both at the level of the blocks and at the level of the threads inside each block. The synchronization required to perform SIMD operations acts at the level of a single block. Nevertheless the number of threads that can be executed simultaneously in each block is limited by the number of cores in the streaming multiprocessor. Conditional jumps break this synchronization determining the serialization of the execution of the blocks, for this reason, in order to achieve good performance this kind of operation should be avoided whenever possible.

An important aspect that should be taken into account in order to design an efficient CUDA program is the memory management. In the Nvidia architecture different memories are available, most notably: the *global memory*, the *local memory*, the *shared memory*, *registers* and the *constant memory*. The global memory has large storage capacity, high latency and it is accessible from any part of the kernel program. The local memory is a segment of the global memory that is assigned to a specific thread. The shared memory is accessible only to threads in the same block, has limited storage capacity with respect to the local memory but exhibits a lower latency. The registers are a private memory assigned to each thread with reduced storage capacity and low latency. The constant memory is a small amount of memory, instantiated at the beginning of the kernel execution, that behave like a read only memory and has a low latency. A well designed CUDA kernel should takes advantage of the memory hierarchy of the GPU, by allocating the most frequently updated data in the shared memory, and the unvarying data in the constant memory.

In order to have a logical mapping between ESSM's entities and the CUDA execution hierarchy, a block is assigned to each GA individual; each thread of the block is responsible for the update of a specific atom of that individual and performs the calculations of the PSO-layer. Following this execution model, a CUDA kernel was implemented to perform a single iteration of the PSO-layer, in which all atoms of all individuals are updated in parallel, reducing the computational complexity for the single protein update from $\mathcal{O}(|\iota|^2)$ of the sequential algorithm to $\mathcal{O}(|\iota|)$. A second CUDA kernel is responsible for the parallel calculation of the objective function, Equation (6.1), which

is performed by means of a parallel reduction algorithm. This strategy reduces the computational complexity from $\mathcal{O}(|\iota|)$ down to $\mathcal{O}(\log_2 |\iota|)$. Moreover the shared memory is used to accumulate the partial results of the reduction algorithm, avoiding the high latencies due to the global memory. The GA-level has not been parallelized yet, so that it is currently performed serially on the CPU. The implementation described above represents an elegant and efficient alternative to a serial counterpart. Nevertheless, CUDA limits the number of threads in a block to 1024, so that ESSM is currently limited to proteins characterized by at most 1024 atoms. An improved and block unaligned version of the algorithm, free from any protein size limitation, is currently under development.

6.6 Results

In this section we present the results obtained by ESSM for the reconstruction of the 3D structure of different proteins. We first performed several tests to determine the influence of the values of the parameters on the reconstruction process, to the aim of finding the best settings of ESSM that were then exploited in all experiments.

6.6.1 Parameterization

These tests consisted in the variation of a single parameter at a time in the optimization process of an *in silico* generated 3-peptide molecule with a length of $N = 56$ atoms. Each test was repeated 30 times, and the average value of the objective function in the resulting models was used to evaluate the influence of each parameter. All preliminary tests, were performed with $I_{\texttt{MAX}} = 2000$, unless otherwise specified.

As a first test we analyzed the impact of the population size Q, by considering the following values: 32, 64, 128, 256 individuals. As expected, the average smallest error achieved decreases as the population size increases (data not shown); however, for $Q > 32$, the improvement of the solutions quality is so slight that it does not justify the larger use of computational resources that it would require. Therefore, the value used in all consecutive tests was set to $Q = 32$.

In the second test, we analyzed the impact of different values for the particles initial maximum velocity $v_{\texttt{MAX}}$, expressed as a function of the diagonal length of the search space $D_{\texttt{MAX}}$. As shown in Figure 6.4, the best results were achieved when $v_{\texttt{MAX}} = D_{\texttt{MAX}}/10$, while smaller values and higher values lead to worse results.

The third test consisted in varying the adaptive velocity factor α. In Figure 6.5 we show the average smallest error obtained with 30 runs of ESSM with several values of factor α. In this test, where $I_{\texttt{MAX}} = 4000$, the best results were obtained with $\alpha = 0.999$, even if smaller values of this factor allowed a faster convergence.

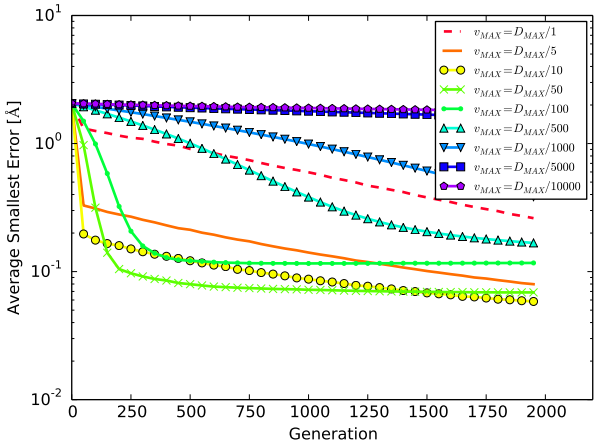


Figure 6.4: Effect of the parameter $v_{\mathtt{MAX}}$
Average smallest error computed over 30 runs of ESSM varying the coefficient μ in $v_{\mathtt{MAX}} = D_{\mathtt{MAX}}/\mu$. The best results were achieved with $\mu = 10$; note that both high and small values for the maximum velocity of particles lead to higher values of the average smallest error.

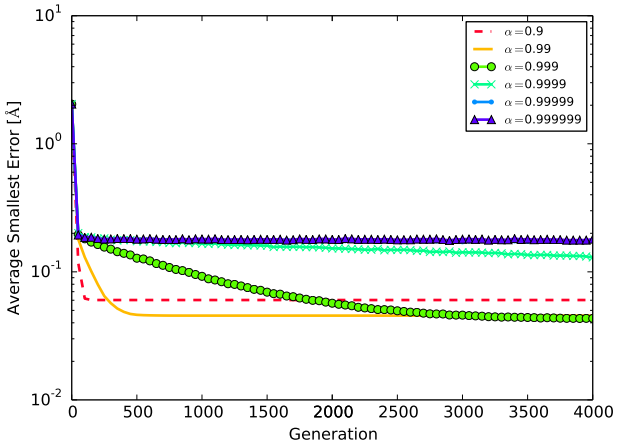


Figure 6.5: Effect of the parameter α
Average smallest error computed over 30 runs of ESSM varying the adaptive velocity factor α. Even though for α equal to 0.9 or 0.99 we obtained a faster convergence, the value $\alpha = 0.999$ allowed to achieve the best results.

A further test concerned the influence of the inertia weight on the particles velocity; in particular, we varied the w value in the range $[0,1]$ and the best result was achieved with $w = 0.4$. Similarly to the case of $v_{\mathtt{MAX}}$, both higher and smaller values of the inertia weight lead to higher values of the average smallest error (data not shown).

The last three tests aimed at finding the best setting for the tournament size, the crossover frequency and the maximum length allowed for a substructure involved in the crossover operation. The best tournament size value was identified around 10% of the population size Q, this value represent a good compromise between the selection pressure and the population diversity throughout the generations. The crossover number of PSO steps for each generation of GA has been set to $I_\chi = 50$. We observed that, despite the crossover improves the average quality of the candidate solutions, increasing its application frequency worsen the objective of individuals (data not shown). Finally, the maximum length allowed for a substructure involved in the crossover operation was set to $size_{\mathtt{MAX}} = 15\%$ of the total number of atoms in the protein (for higher values better results can be achieved, but the improvement of the objective is not enough to justify the larger use of computational resources that it would require).

The results of these preliminary tests led to the following best parameter settings for ESSM:

- population size $Q = 32$ individuals;
- initial $v_{\mathtt{MAX}} = D_{\mathtt{MAX}}/10$;
- adaptive velocity factor $\alpha = 0.999$;
- inertia weight $w = 0.4$;
- tournament size $q = 4$ individuals;
- crossover interval $I_\chi = 50$ generations;
- $size_{\mathtt{MAX}} = 0.15|\iota|$;

To test the validity of ESSM settings we first reconstructed the 3-peptide molecule by using incomplete information. This was realized by removing from matrix **d** the distance values d_{ij} that are above a given cutoff. As shown in Figure 6.6, the average smallest error of the structures reconstructed by ESSM is below 10^{-4}Å also in the case of matrix **d** where $d_{ij} < 6$Å, for all $i, j = 1, \ldots, N$.

6.6.2 Reconstructing the structure of real proteins

To show the effectiveness of our methodology, in Table 6.1 we present the results obtained for the reconstruction of the structure of 9 proteins of increasing length –

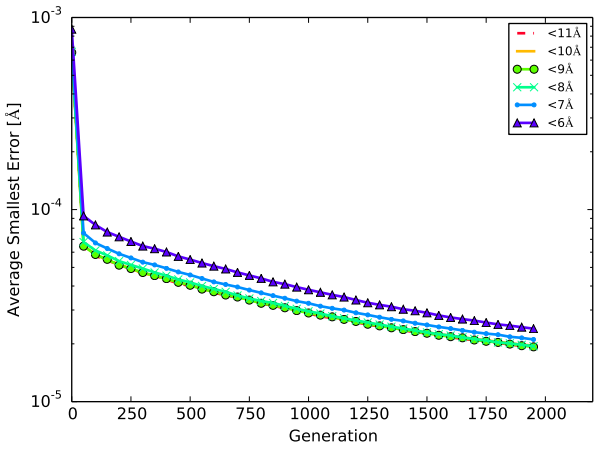


Figure 6.6: Results of ESSM for the MDG problem: 3-peptide
Average smallest error of solutions to the 3-peptide molecule obtained in different optimization processes with incomplete information of inter-atomic distances. Note that, by exploiting only distances $d_{ij} < 6$Å we still achieved an error lower than 10^{-4}Å with respect to the original structure.

taken from the PDB database [25, 228, 229] – using only inter-atomic distances $d_{ij} < 6$Å or $d_{ij} < 7$Å. Each of the structural models obtained has been aligned with the structure in the pdb file using the software TM-align [230]. In particular, for each protein, the error f_ε (defined in Equation (6.1)) and the RMSD associated to the alignment [231] of the best structures found by ESSM after $I_{\texttt{MAX}} = 20000$ iterations are reported. These results highlight the robustness of our method since the f_ε value is low in all cases and, in addition, the RMSD is always lower than 3.5Å, a value that is considered to be indicative of a good reconstruction of proteins' structure [232].

In Figure 6.7, we show the structural alignment, realized with PyMOL [233], of the protein structures obtained with ESSM (using inter-atomic distances below the 6Å cutoff) with the structures available in the PDB database. In the case of proteins 1AX8, 1HOE and 1CTF we obtained a perfect alignment between the protein structure; however, concerning protein 1F39, there is a slight discrepancy between the correct structure and the one obtained with ESSM, probably due to an error in the reconstruction of a small portion of the protein (as better explained in the caption of Figure 6.7), while the overall structure is preserved also in the unaligned region.

Table 6.1: Results of the ESSM: data from real proteins

Results of the reconstruction of proteins' structure with ESSM using only distances $d_{ij} < 6$Å or $d_{ij} < 7$Å. † The reported number considers only atoms from the PDB file, in case of proteins with multiple chains only the atoms in chain A has been considered.

		$d_{ij} < 6$Å		$d_{ij} < 7$Å	
PDB ID	N†	ε [Å]	RMSD [Å]	f_ε [Å]	RMSD [Å]
1PTQ	402	0.152	1.23	0.019	0.08
1CTF	487	0.180	1.46	0.037	0.18
1RGD	548	0.149	1.24	0.014	0.04
1HOE	558	0.172	1.63	0.130	1.7
1LFB	641	0.206	2.21	0.254	2.08
1F39	767	0.278	3.25	0.090	0.93
1PHT	814	0.291	2.02	0.123	1.86
1POA	914	0.056	0.99	0.074	1.26
1AX8	1003	0.092	2.27	0.075	1.59

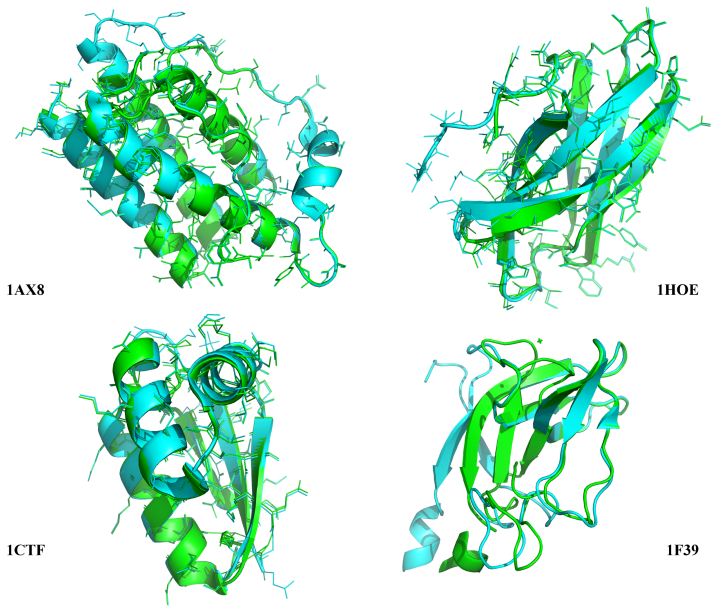


Figure 6.7: Results of ESSM for the MDG problem: real proteins
Examples of the structural alignment between the structures available in the PDB database (*cyan*) and protein structures reconstructed by ESSM, using distance matrices **d** with $d_{ij} < 6$Å (*green*). The alignments are correct, even though, in the case of protein 1F39, there is a slight discrepancy between the correct structure and the one obtained with ESSM, probably due to an error in the reconstruction of a small portion of the protein connecting two major structural motifs, while the overall structure is preserved also in the unaligned region. This kind of errors can arise in portions of the proteins with extended structure, when a very low number of inter-atomic distances are available. Images obtained with PyMOL [233].

7 Conclusions and Future Works

In this thesis two hard combinatorial problems connected to computational prediction and reconstruction of protein structures have been analyzed: the protein structure prediction problem and the molecular distance geometry problem, in both cases effective heuristics approaches have been proposed. In the case of the PSP problem the simplified representation provided by the HP model has been adopted. Two novel heuristic methods have been proposed for PSP in the HP model both based on the idea of extending the neighborhood structure of the SA method when a high rejection rate have been reached. Both the proposed methods rely on the use of a biasing strategy ("the map") to search over the extended neighborhood and for this reason the general strategy has been called local landscape mapping. In the first method, the LLM_{mem}, the biasing strategy is represented by a memory structure, that stores the desirability associated to neighbors assignments. In the second, the LLM_{LS}, the biasing strategy is a specialized local search procedure. Moreover, a new perturbation system for the optimization in the HP model has been introduced that closely resemble the one applied for the prediction in off-lattice models. The results obtained in the new framework suggest that it provides a challenging benchmark, indeed the performance of all the tested heuristic is reduced using this perturbation system with respect to that achieved using "canonical" move sets. Nevertheless, one of the proposed heuristic LLM_{mem} showed to be particularly suitable for this framework outperforming two well established SOHs such as SA and ACO in all the benchmark instances taken into account. Although this results are encouraging the LLM_{mem} method has several relevant limitations:

- it requires the tuning of a high number of free parameters;
- the selection of neighbors assignment can be computationally expensive;
- the biasing strategy based on pheromone is not able to collect information if the neighborhood has a flat energy landscape.

Moreover, results obtained in the 3DC lattice with move sets suggest that the performance improvement over SA vanishes when the memory structure is associated to a volatile neighborhood structure such as the one defined by move sets. The development of a different pheromone structure to better map pull moves-induced neighborhood will be the subject of future works along with the testing of the LLM_{mem} efficacy in off-lattice fragment assembly. The LLM_{LS} has been designed to overcome the limitations described above and to verify the general utility of the LLM strategy. Therefore, the results obtained with this method on several benchmark instances both in 3DC and FCC lattices have been compared with those achieved by several state of the art methods. LLM_{LS} resulted the best performing method in most of the comparisons. Moreover, to the best of our knowledge, LLM_{LS} is the only method that has been successfully applied to both 3DC and FCC instances. In their complex, results presented in this thesis support the idea of combining a biased large neighborhood search and simulated annealing. The use of a variable objective function in the local search step seems the most promising implementation of the this strategy.

In the case of the Molecular Distance Geometry problem, only the case of incomplete information about exact inter-atomic distances has been taken in to account. Our methodology, called evolutionary spring swarm method, is a memetic algorithm that combines swarm intelligence and evolutionary computation along with a local search aimed at improving the effectiveness of the crossover operator. ESSM works at two different levels: the PSO-layer is used to move particles in the 3D space, where each particle encodes the coordinates of an atom of the protein structure to be reconstructed; the GA-layer is exploited to select individuals, and to recombine them by means of a crossover operator that exchanges substructures between individuals. The crossover – aided by a local search method used to identify the best roto-translation of the exchanged substructure – is followed by a chirality correction, in which the correct orientation of amino-acids is verified and adjusted.

The ESSM, was tested on a set of proteins having a number of atoms ranging from 402 to 1003; in all cases we obtained a correct 3D structure, as confirmed by the values of RMSD (see Table 6.1). Indeed, the results indicate that the accuracy achieved by ESSM is comparable to the accuracy achieved by state-of-the-art methods [23, 26]. Although several improvements, discussed below, can be applied to ESSM, the results obtained are remarkable, since they confirm that ESSM successfully extended the domain of application of SOHs to the previously unexplored MDG problem. Moreover, two additional qualities of our method reside in its intrinsic stochasticity and extensibility: on the one hand, the various reconstructed structures (with low error values) that can be obtained in each run of the ESSM are useful to represent the structural variability observed in biological molecules, which is a source of noise in NMR data; on the other hand, the ESSM can be easily improved by including a molecular force field in the scoring function during the final stages of the optimization process, in order to select structural models that are more realistic from a physical point of view. The extension of

ESSM to the general MDG problem with noisy distance restraints will be the subject of future works. An efficient non-sequential implementation of our crossover mechanism is far from trivial and currently under investigation.

A Appendix A

A.1 Benchmark HP sequences used in this thesis

Table A.1: Harvard instances

ID	Length (H)	E_{min}	Sequence
H1	48(24)	-32	HPHHPPHHHHPHHHPPHHPPHPHHHPHPHHPPHHPPPHPPPPPPPPHH
H2	48(24)	-34	HHHHPHHPHHHHHPPHPPHHPPHPPPPPPPHPPHPPPHPPHHPPHHHPH
H3	48(24)	-34	PHPHHPHHHHHHPPHPHPPHPHHPHPHPPPHPPHHPPHHPPHPHPPHP
H4	48(24)	-33	PHPHHPPHPHHHPPHHPHHPPPHHHHHPPHPHHPHPHPPPPHPPHPHP
H5	48(24)	-32	PPHPPPHPHHHHPPHHHHPHHPHHHPPHPHPHPPHPPPPPPHHPHHPH
H6	48(24)	-32	HHHPPPHHPHPHHPHHPHHPHPPPPPPPHPHPPHPPPHPPHHHHHHPH
H7	48(24)	-32	PHPPPPHPHHHPHPHHHHPHHPHHPPPHPHPPPHHHPPHHPPHHPPPH
H8	48(24)	-31	PHHPHHHPHHHHPPHHHPPPPPPHPHHPPHHPHPPPHHPHPHPHHPPP
H9	48(24)	-34	PHPHPPPPHPHPHPPHPHHHHHHPPHHHPHPPHPHHPPHPHHHPPPPH
H10	48(24)	-33	PHHPPPPPPHHPPPHHHPHPPHPHHPPHPPHPPHHPHHHHHHPPHH

Table A.2: Instances from real proteins

ID	Length (H)	E_{min}	Sequence
3CYT	103(37)	-58	PPHHPPPPPPHHPPHHPHPPHPPPPPPPPHPPPHHPHHPPPPPPPHPPHPHP PHPPPPPHHHPPPPHHPHHPPPPPPHHPPPPHHHHPHPPPPPPPPHHHH HPPHPP
7RSA	124(47)	-75	PPPHHHPHPPPPHPPPPPPHHPPPPHHPPHHPPPPHPPPPHPPHPPHHPP PHHPHPHHHPPPPHHHPPPPPPPHHPPHPPHPHPPHPPPPPPPPHPPHHH PPPPHPPPHHHHHHPPPPHHPHPHPHPH
2SNS	136(50)	-83	HPPPPPHPPPPPHPHHPHHPPPPPHPHHHPPPPHPHPHHHHPPPPPPPPP PPHPPHPPPHPHHPPPHHPPHPPHPHPHPPPPPPPPHPPPHHHHHHPP PHHPPHHHPPPHHPHHHHHHPPPPPPPPPHPPPPHPHPPPP

Table A.3: F90 instances

ID	Length (H)	E_{min}	Sequence
F90_1	90(50)	-168	PPHHHPPPHHPPPPHHPHHHHHHPHPHPHHPHHHHHPHHHPHPHHHHPH HPPPPHHHPHPHPPHHHPHHPHPHPPHHHPPPPHHPPHPPP
F90_2	90(50)	-168	PHHPPHPHHPHHHPHHHPPHHHHHHPPHPHPPPPHHHPHPPHHHHPHHH HPHHHPHHPPPPPHHPPPPHPHPHPHPHHPPHHHPPPHHHP
F90_3	90(50)	-167	HPHPHHHPHHHHPHHHPPPHPPPHPPPPHHHPPHPPPPHHHPPPPPPPP HPHHPHHHHPHHHPHPHHPPHHHHHPHHPPHHPHHHHHHPH
F90_4	90(50)	-168	PHHHPPHPPHPHPPPPHPPPHPHPPHPHHPHPPPHHHPHHHPPHHHPPH PPPPHPHHHPPHHPPHHHPPHHHHHHPHHHHHHHPHHHHPH
F90_5	90(50)	-167	PPPHPHHHHHHHPPPHPPHHHHHPHHPPHHPPHHHHHPHPHPHHPPHHPP PPHPPPHHHPHPHHHHHHHPHHPHPPHHPPPHHHPHPPHPP

Table A.4: S instances

ID	Length (H)	E_{min}	Sequence
S1	135 (100)	-357	HHHHPHHHHHHPPHHPHHHHHHHHHPHHPHHHHHHHHHHPPHHPPPPPHH
			PHHHHHHHHPHPPHHPPPPHHHHHHHHHPHHHHHHPPHHHHHHHPHPPHHH
			HHHHHHPPHHPPPPHHHHHHHPHHPHHHHHHHPPHHHH
S2	151 (100)	-360	HHPPHPHHHHHHHHHHPHPPPPHHHPPPHHHHHPPHHHHHPPHHHHPPHH
			HHPPHHHHHHPHHHHPPPHHPPPHHHHHHHHPHPPHHHPPPHHHHHPPH
			HHHHHHPPPHHPPHHHHHPPPHHHHHHHHHPHPPHHHHHHHPPPHHHPP
			HHP
S3	162 (100)	-367	HHHPPPHHPHHPPPPPHHHHHHHHPHPPHHPHHPHHHHHPPPHHHHHHHH
			HPPHPHPPHPHPPHHHPHPPHPHPPPHHHHHHPHHHHPPPHHHPPPPHHP
			PPHHHPPHHHHPHHHHHPPHHHHHHPPPHHHHHHPPPHPPHHHHPHHHH
			HHPPHHPPHHH
S4	164 (100)	-370	HHPPHPHHHHHHHPPHPHPPHPHPPPPHHHPPPHHPHPHHPPHHHHHPH
			HHHPPHHHHPPHHHHHHPHHHHPPPHHPPPHHHHHHHHPHPPHHHPPPHH
			HHHPPHHHHHHPHPPPHHPPHHHPHHPPPHPHHHHHHHPHPPHPPHHHH
			HPHPPPHHHPPHHP

Table A.5: F180 instances

ID	Length (H)	E_{min}	Sequence
F180_1	180(100)	-378	HHPPHHHHHPHHHPPPHHHPPHHHPHPPHHHHHPPPHHHPPPHPHH
			PPPPPHHPPHHPHHPHPHHPPPPPHHHPPPPHPHHHPPHPPPHHHP
			HHHHPPHHPHPHHHHPHHHHPPHHPHHPHHPHHHPHPPHPHHPHPH
			HPHHHPHHPPHPPPHPPPPPPPHHHPHHHHHPHHHHHPPHPP
F180_2	180(100)	-381	PHHPHPPPHPPHHPHHHPHPHHPHHHPHHHPPPHHPPHPHPHHPHH
			HHPPHHPHPHHHHHPHHPPPPHPHPHPPHHHHPHHHHPHHHHHPPH
			PHHHPPPHPHPPHHPPPHHPHPHPPPPPHPHHPHHHPHPPPPHHPH
			HHHHPPPHHHHHHHHPHHPPPPPHPPPHPPHPPPHHPHHHHH
F180_3	180(100)	-378	HHHPHPPHHPPPHPPPHPHPHPPHHHHPPHHHHHHPHPHHPPPPPH
			PPHHPHHHHHHHHHHHHPPHPPHPPHHHHHHHHHPPPPHPPHHHHHPP
			HHHPPHHPPHHHHHPPPHHHHHHPHHHPPPHHPPHPPPPHPPHPPPH
			PPPPHHHPPHHPHPPHHHPHHPPHHPHHPHPHPHPHPHPHHP

Table A.6: R instances

ID	Length (H)	E_{min}	Sequence
R1	200(100)	-384	PPPHPHHPHHPPPHPHPPPPHPHHPPHPHHHHHPPHHPPHHHHHHPPHPP
			HHPPHPHPHHHHHPHHPHHHPPPHHHPHHPPHPHPPHPPPHPPHPPHPPH
			HHPHHHPHPPHPHHPHHHHPHPHHHPHHHPPPPPPHHHHHHPPPPPPPPHH
			HPPHPHPPPHPHPHPHHPPHHPPPPHHHHHHPPPHHPPPPPHPPPHHPP
R2	200(100)	-383	HPHHPPHPPPPPHHPHPHPHHPPHPPPPHHHHHHPPPHPPHHHPPHPPPP
			HHPPHHHPHPHHHPPHPHHPPHPHHPPPPHHPPHPPHHHHPPPPPHHHPP
			PPHPPPPPPHPPHHPHHHHPHHHHHHHHHPPHHPPPHPHHHPHHHHHPHHP
			HHHPHPHHPPPPHPHHPHHHPHPPPPHPPPPPPHPHHHHHPHHPPPHPPH
R3	200(100)	-385	HPHHHPHHPHPHPPPHHHHHPHPHPHHHHPPPHHPPPPPPHHPPPPHPHH
			HPPPPHPPPHHPHHPPPHPPHPPPHHHHPHHPHPPPPHHPPPHHPPHPPP
			HPPHHHPHHHPHPPHPHHHHPPHHPPPPHHHPHHPPHPPHHHHPPHPHPP
			HPHPPPPPHPHPHHHHHHHPHPHHHHHHPHHPPPPHPPPPHPPPHHHPHH

Bibliography

[1] Francis S Collins, Michael Morgan, and Aristides Patrinos. The human genome project: lessons from large-scale biology. *Science*, 300(5617):286–290, 2003.

[2] D H Turner, N Sugimoto, and S M Freier. Rna structure prediction. *Annual Review of Biophysics and Biophysical Chemistry*, 17(1):167–192, 1988.

[3] Shi-Jie Chen. RNA folding: conformational statistics, folding kinetics, and ion electrostatics. *Annual review of biophysics*, 37:197–214, January 2008.

[4] MatthewG. Seetin and DavidH. Mathews. Rna structure prediction: An overview of methods. In Kenneth C. Keiler, editor, *Bacterial Regulatory RNA*, volume 905 of *Methods in Molecular Biology*, pages 99–122. Humana Press, 2012.

[5] Donald Voet and Judith G Voet. *Biochemistry*. W.H. Freeman, 2011.

[6] RR Birge. Protein-based optical computing and memories. *Computer*, 10:56–67, 1992.

[7] D N Bolon and S L Mayo. Enzyme-like proteins by computational design. *Proceedings of the National Academy of Sciences of the United States of America*, 98(25):14274–9, December 2001.

[8] Justin Ashworth, JJ Havranek, and CM Duarte. Computational redesign of endonuclease DNA binding and cleavage specificity. *Nature*, 441(June):10–13, 2006.

[9] Jan Drenth. *Principles of Protein X–Ray Crystallography*. Springer, 2006.

[10] Kurt Wüthrich. NMR studies of structure and function of biological macromolecules (Nobel lecture). *Angewandte Chemie (International ed. in English)*, 42 (29):3340–63, July 2003.

[11] Isabel Usón and George M Sheldrick. Advances in direct methods for protein crystallography. *Current Opinion in Structural Biology*, 9(5):643 – 648, 1999.

[12] Santosh Panjikar, Venkataraman Parthasarathy, Victor S. Lamzin, Manfred S. Weiss, and Paul A. Tucker. *Auto-Rickshaw*: an automated crystal structure determination platform as an efficient tool for the validation of an X-ray diffraction experiment. *Acta Crystallographica Section D*, 61(4):449–457, Apr 2005.

[13] Paul D. Adams, Pavel V. Afonine, Gábor Bunkóczi, Vincent B. Chen, Ian W. Davis, Nathaniel Echols, Jeffrey J. Headd, Li-Wei Hung, Gary J. Kapral, Ralf W. Grosse-Kunstleve, Airlie J. McCoy, Nigel W. Moriarty, Robert Oeffner, Randy J. Read, David C. Richardson, Jane S. Richardson, Thomas C. Terwilliger, and Peter H. Zwart. *PHENIX*: a comprehensive Python-based system for macromolecular structure solution. *Acta Crystallographica Section D*, 66(2):213–221, Feb 2010.

[14] George M. Sheldrick. Experimental phasing with *SHELXC*/*D*/*E*: combining chain tracing with density modification. *Acta Crystallographica Section D*, 66(4):479–485, Apr 2010.

[15] Frank DiMaio, Nathaniel Echols, Jeffrey J Headd, Thomas C Terwilliger, Paul D Adams, and David Baker. Improved low-resolution crystallographic refinement with Phenix and Rosetta. *Nature methods*, 10(11):1102–4, November 2013.

[16] Raymond Hui and Aled Edwards. High-throughput protein crystallization. *Journal of Structural Biology*, 142(1):154 – 161, 2003.

[17] Helen Song, Delai L. Chen, and Rustem F. Ismagilov. Reactions in Droplets in Microfluidic Channels. *Angewandte Chemie International Edition*, 45(44):7336–7356, 2006.

[18] Masahiko Hiraki, Ryuichi Kato, Minoru Nagai, Tadashi Satoh, Satoshi Hirano, Kentaro Ihara, Norio Kudo, Masamichi Nagae, Masanori Kobayashi, Michio Inoue, Tamami Uejima, Shunichiro Oda, Leonard M. G. Chavas, Masato Akutsu, Yusuke Yamada, Masato Kawasaki, Naohiro Matsugaki, Noriyuki Igarashi, Mamoru Suzuki, and Soichi Wakatsuki. Development of an automated large-scale protein-crystallization and monitoring system for high-throughput protein-structure analyses. *Acta Crystallographica Section D*, 62(9):1058–1065, Sep 2006.

[19] Masatoshi Maeki, Yuki Teshima, Saori Yoshizuka, Hiroshi Yamaguchi, Kenichi Yamashita, and Masaya Miyazaki. Controlling protein crystal nucleation by droplet-based microfluidics. *Chemistry – A European Journal*, 20(4):1049–1056, 2014.

[20] Gwyndaf Evans and Gérard Bricogne. Triiodide derivatization and combinatorial counter-ion replacement: two methods for enhancing phasing signal using laboratory Cu$K\alpha$ X-ray equipment. *Acta Crystallographica Section D*, 58(6 Part 2): 976–991, Jun 2002.

[21] Judit É. Debreczeni, Gábor Bunkóczi, Qingjun Ma, Heiko Blaser, and George M. Sheldrick. In-house measurement of the sulfur anomalous signal and its use for phasing. *Acta Crystallographica Section D*, 59(4):688–696, Apr 2003.

[22] Andrea Grosso, Marco Locatelli, and Fabio Schoen. Solving molecular distance geometry problems by global optimization algorithms. *Comput. Optim. Appl.*, 43 (1):23–37, 2009.

[23] Atilla Sit and Zhijun Wu. Solving a generalized distance geometry problem for protein structure determination. *B. Math. Biol.*, 73(12):2809–2836, 2011.

[24] Carlile Lavor, Leo Liberti, Nelson Maculan, and Antonio Mucherino. The discretizable molecular distance geometry problem. *Comput. Optim. Appl.*, 52 (1):115–146, 2011.

[25] Levente Fabry-Asztalos, Istvan Lorentz, and Razvan Andonie. Molecular distance geometry optimization using geometric build-up and evolutionary techniques on GPU. In *2012 IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*, pages 321–328, 2012.

[26] Michael Souza, Carlile Lavor, Albert Muritiba, and Nelson Maculan. Solving the molecular distance geometry problem with inaccurate distance data. *BMC Bioinformatics*, 14 Suppl 9(Suppl 9):S7, 2013.

[27] The CASP website. URL http://predictioncenter.org/.

[28] John Moult, Jan T Pedersen, Richard Judson, and Krzysztof Fidelis. A large-scale experiment to assess protein structure prediction methods. *Proteins: Structure, Function, and Bioinformatics*, 23(3):ii—-iv, 1995.

[29] Andriy Kryshtafovych, Krzysztof Fidelis, and John Moult. Casp10 results compared to those of previous casp experiments. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):164–174, 2014.

[30] Lukasz Jaroszewski, Zhanwen Li, S Sri Krishna, Constantina Bakolitsa, John Wooley, Ashley M Deacon, Ian a Wilson, and Adam Godzik. Exploration of uncharted regions of the protein universe. *PLoS biology*, 7(9):e1000205, September 2009.

[31] Benoît H. Dessailly, Rajesh Nair, Lukasz Jaroszewski, J. Eduardo Fajardo, Andrei Kouranov, David Lee, Andras Fiser, Adam Godzik, Burkhard Rost, and Christine Orengo. PSI-2: Structural Genomics to Cover Protein Domain Family Space. *Structure*, 17(6):869–881, June 2009.

[32] KF Lau and KA Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22(10):3986–3997, 1989.

[33] WE Hart and S Istrail. Robust proofs of NP-hardness for protein folding: General lattices and energy potentials. *Journal of Computational Biology*, 4(1):1–22, 1997.

[34] Neal Lesh, Michael Mitzenmacher, and Sue Whitesides. A complete and effective move set for simplified protein folding. *Proceedings of the seventh annual international conference on Computational molecular biology - RECOMB '03*, pages 188–195, 2003.

[35] Chris Thachuk, Alena Shmygelska, and Holger H Hoos. A replica exchange Monte Carlo algorithm for protein folding in the HP model. *BMC bioinformatics*, 8:342, 2007.

[36] Jinfeng Zhang, S C Kou, and Jun S Liu. Biopolymer structure simulation and optimization via fragment regrowth Monte Carlo. *The Journal of chemical physics*, 126(22):225101, June 2007.

[37] Thomas Wüst and David P. Landau. Versatile approach to access the low temperature thermodynamics of lattice polymers and proteins. *Phys. Rev. Lett.*, 102(4):178101, Apr 2009.

[38] Ivan Dotu, Manuel Cebrian, Pascal Van Hentenryck, and Peter Clote. On lattice protein structure prediction revisited. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 8(6):1620–1632, 2011.

[39] Jingfa Liu, Gang Li, Jun Yu, and Yonglei Yao. Heuristic energy landscape paving for protein folding problem in the three-dimensional HP lattice model. *Computational biology and chemistry*, 38:17–26, June 2012.

[40] Thomas Wüst and David P Landau. Optimized wang-landau sampling of lattice polymers: Ground state search and folding thermodynamics of hp model proteins. *The Journal of chemical physics*, 137(6):064903, 2012.

[41] Swakkhar Shatabda, M a Hakim Newton, Mahmood a Rashid, Duc Nghia Pham, and Abdul Sattar. The road not taken: retreat and diverge in local search for simplified protein structure prediction. *BMC bioinformatics*, 14 Suppl 2(Suppl 2): S19, 2013.

[42] Mahmood A Rashid, MA Hakim Newton, Md Tamjidul Hoque, Swakkhar Shatabda, Duc N Pham, and Abdul Sattar. Spiral search: a hydrophobic-core directed local search for simplified psp on 3d fcc lattice. *BMC bioinformatics*, 14(Suppl 2):S16, 2013.

[43] Andrea G. Citrolo and Giancarlo Mauri. A hybrid monte carlo ant colony optimization approach for protein structure prediction in the hp model. In *Wivace*, pages 61–69, 2013.

[44] Andrea Citrolo and Giancarlo Mauri. A local landscape mapping method for protein structure prediction in the HP model. *Natural Computing*, 13(3):309–319, 2014.

[45] James B Saxe. *Embeddability of weighted graphs in k-space is strongly NP-hard*. Carnegie-Mellon University, Department of Computer Science, 1980.

[46] M.S. Nobile, A.G. Citrolo, P. Cazzaniga, D. Besozzi, and G. Mauri. A memetic hybrid method for the Molecular Distance Geometry Problem with incomplete information. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 1014–1021, July 2014.

[47] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[48] Giorgio Ausiello. *Complexity and approximation: Combinatorial optimization problems and their approximability properties*. Springer Science & Business Media, 1999.

[49] A. Nemirovskii and D. Yudin. *Problem complexity and method efficiency in optimization*. Wiley-Interscience, 1983.

[50] Erich Novak and Henryk Woźniakowski. *Tractability of Multivariate Problems: Standard information for functionals*, volume 2. European Mathematical Society, 2010.

[51] Stephen Boyd, Lieven Vandenberghe, and Michael Grant. Efficient convex optimization for engineering design, 1994.

[52] Y. Nesterov and A. Nemirovskii. *Interior-Point Polynomial Algorithms in Convex Programming*. Society for Industrial and Applied Mathematics, 1994.

[53] StephenA. Vavasis. Complexity issues in global optimization: A survey. In Reiner Horst and PanosM. Pardalos, editors, *Handbook of Global Optimization*, volume 2 of *Nonconvex Optimization and Its Applications*, pages 27–41. Springer US, 1995.

[54] Stephen A Cook. The complexity of theorem-proving procedures. In *Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158. ACM, 1971.

[55] Richard M Karp. *Reducibility among combinatorial problems*. Springer, 1972.

[56] David S Johnson. The np-completeness column: An ongoing guide. *Journal of Algorithms*, 3(2):182–195, 1982.

[57] Pierluigi Crescenzi, Viggo Kann, and Magnús Halldórsson. A compendium of np optimization problems, 1995.

[58] Donald R Jones, Cary D Perttunen, and Bruce E Stuckman. Lipschitzian optimization without the lipschitz constant. *Journal of Optimization Theory and Applications*, 79(1):157–181, 1993.

[59] Costas D Maranas and Christodoulos A Floudas. Global optimization in generalized geometric programming. *Computers & Chemical Engineering*, 21(4):351–369, 1997.

[60] Donald R Jones, Matthias Schonlau, and William J Welch. Efficient global optimization of expensive black-box functions. *Journal of Global optimization*, 13(4):455–492, 1998.

[61] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[62] Arnold Neumaier. Complete search in continuous global optimization and constraint satisfaction. *Acta numerica*, 13:271–369, 2004.

[63] André I Khuri and Siuli Mukhopadhyay. Response surface methodology. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(2):128–149, 2010.

[64] LuisMiguel Rios and NikolaosV. Sahinidis. Derivative-free optimization: a review of algorithms and comparison of software implementations. *Journal of Global Optimization*, 56(3):1247–1293, 2013.

[65] Bruce Hajek. Cooling schedules for optimal annealing. *Mathematics of operations research*, 13(2):311–330, 1988.

[66] G.O. Roberts and A.F.M. Smith. Simple conditions for the convergence of the gibbs sampler and metropolis-hastings algorithms. *Stochastic Processes and their Applications*, 49(2):207 – 216, 1994.

[67] Günter Rudolph. Convergence analysis of canonical genetic algorithms. *Neural Networks, IEEE Transactions on*, 5(1):96–101, 1994.

[68] Walter J Gutjahr. Aco algorithms with guaranteed convergence to the optimal solution. *Information processing letters*, 82(3):145–153, 2002.

[69] N Buniyamin, N Sariff, WAJ Wan Ngah, and Z Mohamad. Robot global path planning overview and a variation of ant colony system algorithm. *International journal of mathematics and computers in simulation*, 5(1):9–16, 2011.

[70] John E Bell and Patrick R McMullen. Ant colony optimization techniques for the vehicle routing problem. *Advanced Engineering Informatics*, 18(1):41–48, 2004.

[71] Fugao Wang and David P Landau. Efficient, multiple-range random walk algorithm to calculate the density of states. *Physical review letters*, 86(10):2050, 2001.

[72] Ma Guadalupe Castillo Tapia and Carlos A Coello Coello. Applications of multi-objective evolutionary algorithms in economics and finance: A survey. In *IEEE congress on evolutionary computation*, volume 7, pages 532–539, 2007.

[73] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *Evolutionary Computation, IEEE ...*, 1(1):67–82, April 1997.

[74] Helena R Lourenço, Olivier C Martin, and Thomas Stutzle. Iterated local search. *arXiv preprint math/0102188*, 2001.

[75] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. MIT Press, 2004.

[76] Martin Karplus. How does a protein fold? *nature*, 369:19, 1994.

[77] Helena R Lourenço, Olivier C Martin, and Thomas Stützle. *Iterated local search*. Springer, 2003.

[78] Richard K Congram, Chris N Potts, and Steef L van de Velde. An iterated dynasearch algorithm for the single-machine total weighted tardiness scheduling problem. *INFORMS Journal on Computing*, 14(1):52–67, 2002.

[79] David Applegate, William Cook, and André Rohe. Chained lin-kernighan for large traveling salesman problems. *INFORMS Journal on Computing*, 15(1):82–92, 2003.

[80] Olivier Martin, Steve W Otto, and Edward W Felten. Large-step markov chains for the tsp incorporating local search heuristics. *Operations Research Letters*, 11(4): 219–224, 1992.

[81] Thomas A Feo and Mauricio GC Resende. A probabilistic heuristic for a computationally difficult set covering problem. *Operations research letters*, 8 (2):67–71, 1989.

[82] Thomas A Feo and Mauricio GC Resende. Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109–133, 1995.

[83] Fred Glover. Tabu search-part i. *ORSA Journal on computing*, 1(3):190–206, 1989.

[84] Fred Glover. Tabu search—part ii. *ORSA Journal on computing*, 2(1):4–32, 1990.

[85] Fred Glover and Manuel Laguna. *Tabu search*. Springer, 1999.

[86] Nicholas Metropolis and Stanislaw Ulam. The monte carlo method. *Journal of the American statistical association*, 44(247):335–341, 1949.

[87] 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*, 21(6):1087–1092, 1953.

[88] S Kirkpatrick, DG Jr., and MP Vecchi. Optimization by simulated annealing. *science*, 220(4598):671–680, 1983.

[89] Tzuu-Shuh Chiang and Yunshyong Chow. On the convergence rate of annealing processes. *Stochastic Processes and their Applications*, 26:214, 1987.

[90] TS Chiang and Y Chow. A limit theorem for a class of inhomogeneous Markov processes. *The Annals of probability*, 17(4):1483–1502, 1989.

[91] John Bertsimas, Dimitris; Tsitsiklis. Simulated annealing. *Statistical Science*, 8(1): 10—-15, 1993.

[92] Robert H Swendsen and Jian-Sheng Wang. Replica monte carlo simulation of spin-glasses. *Physical Review Letters*, 57(21):2607, 1986.

[93] Yukito Iba. Extended ensemble monte carlo. *International Journal of Modern Physics C*, 12(05):623–656, 2001.

[94] Ayori Mitsutake, Yuji Sugita, and Yuko Okamoto. Generalized-ensemble algorithms for molecular simulations of biopolymers. *Peptide Science*, 60(2):96–123, 2001.

[95] Dong Xu and Yang Zhang. Ab initio protein structure assembly using continuous structure fragments and optimized knowledge-based force field. *Proteins: Structure, Function, and Bioinformatics*, 80(7):1715–1735, 2012.

[96] DP Landau, Shan-Ho Tsai, and M Exler. A new approach to monte carlo simulations in statistical physics: Wang-landau sampling. *American Journal of Physics*, 72(10):1294–1302, 2004.

[97] Chenggang Zhou, Thomas C Schulthess, Stefan Torbrügge, and DP Landau. Wang-landau algorithm for continuous models and joint density of states. *Physical review letters*, 96(12):120201, 2006.

[98] J. H. Holland. *Adaptation in Natural and Artificial Systems*. The University of Michigan Press, 1975.

[99] David E. Goldberg. *Genetic algorithms in search, optimization, and machine learning*. Number 2. Addison-Wesley, Reading, MA, 1989.

[100] T. Bäck. Selective pressure in evolutionary algorithms: A characterization of selection mechanisms. In *Proc. of the First IEEE Conference on Evolutionary Computation*, volume 1, pages 57–62. IEEE, 1994.

[101] Thomas Stützle and Holger H. Hoos. Max–Min Ant System. *Future Generation Computer Systems*, 16(8):889–914, 2000.

[102] Krzysztof Socha and Marco Dorigo. Ant colony optimization for continuous domains. *European Journal of Operational Research*, 185(3):1155–1173, March 2008.

[103] J. Kennedy and R.C. Eberhart. Particle swarm optimization. In *Proc. of the IEEE International Conference on Neural Networks*, volume IV, pages 1942–1948, Piscataway, NJ, 1995.

[104] Liam J McGuffin, Kevin Bryson, and David T Jones. The psipred protein structure prediction server. *Bioinformatics*, 16(4):404–405, 2000.

[105] Julia Koehler Leman, Ralf Mueller, Mert Karakas, Nils Woetzel, and Jens Meiler. Simultaneous prediction of protein secondary structure and transmembrane spans. *Proteins: Structure, Function, and Bioinformatics*, 81(7):1127–1140, 2013.

[106] Alexey G Murzin, Steven E Brenner, Tim Hubbard, and Cyrus Chothia. Scop: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536–540, 1995.

[107] Frances Pearl, Annabel Todd, Ian Sillitoe, Mark Dibley, Oliver Redfern, Tony Lewis, Christopher Bennett, Russell Marsden, Alistair Grant, David Lee, et al. The cath domain structure database and related resources gene3d and dhs provide comprehensive domain family information for genome analysis. *Nucleic acids research*, 33(suppl 1):D247–D251, 2005.

[108] Johannes Söding, Andreas Biegert, and Andrei N Lupas. The hhpred interactive server for protein homology detection and structure prediction. *Nucleic acids research*, 33(suppl 2):W244–W248, 2005.

[109] Sitao Wu and Yang Zhang. Lomets: a local meta-threading-server for protein structure prediction. *Nucleic acids research*, 35(10):3375–3382, 2007.

[110] Andrej Šali and Tom L Blundell. Comparative protein modelling by satisfaction of spatial restraints. *Journal of molecular biology*, 234(3):779–815, 1993.

[111] Marc A Martí-Renom, Ashley C Stuart, András Fiser, Roberto Sánchez, Francisco Melo, and Andrej Šali. Comparative protein structure modeling of genes and genomes. *Annual review of biophysics and biomolecular structure*, 29(1):291–325, 2000.

[112] Konstantin Arnold, Lorenza Bordoli, Jürgen Kopp, and Torsten Schwede. The swiss-model workspace: a web-based environment for protein structure homology modelling. *Bioinformatics*, 22(2):195–201, 2006.

[113] CB Anfinsen. Principles that govern the folding of protein chains. *Science*, 181 (4096):223–230, 1973.

[114] Jacob N Israelachvili. *Intermolecular and surface forces: revised third edition*. Academic press, 2011.

[115] Andrew R. Leach. *Molecular modelling: principles and applications*. Pearson Education, 2001.

[116] Alexander D MacKerell, Donald Bashford, MLDR Bellott, RL Dunbrack, Jeffrey D Evanseck, Martin J Field, Stefan Fischer, Jiali Gao, H Guo, Sookhee Ha, et al. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *The Journal of Physical Chemistry B*, 102(18):3586–3616, 1998.

[117] Yong Duan, Chun Wu, Shibasish Chowdhury, Mathew C Lee, Guoming Xiong, Wei Zhang, Rong Yang, Piotr Cieplak, Ray Luo, Taisung Lee, et al. A point-charge force field for molecular mechanics simulations of proteins based on condensed-phase quantum mechanical calculations. *Journal of computational chemistry*, 24(16): 1999–2012, 2003.

[118] Chris Oostenbrink, Alessandra Villa, Alan E Mark, and Wilfred F Van Gunsteren. A biomolecular force field based on the free enthalpy of hydration and solvation: The gromos force-field parameter sets 53a5 and 53a6. *Journal of computational chemistry*, 25(13):1656–1676, 2004.

[119] Carol a Rohl, Charlie E M Strauss, Kira M S Misura, and David Baker. Protein structure prediction using Rosetta. *Methods in enzymology*, 383(2003):66–93, January 2004.

[120] Yang Zhang and Jeffrey Skolnick. Automated structure prediction of weakly homologous proteins on a genomic scale. *Proceedings of the National Academy of Sciences of the United States of America*, 101(20):7594–7599, 2004.

[121] Christina L Vizcarra and Stephen L Mayo. Electrostatics in computational protein design. *Current opinion in chemical biology*, 9(6):622–626, 2005.

[122] Nathan A Baker. Improving implicit solvent simulations: a poisson-centric view. *Current opinion in structural biology*, 15(2):137–143, 2005.

[123] Patrice Koehl. Electrostatics calculations: latest methodological advances. *Current opinion in structural biology*, 16(2):142–151, 2006.

[124] Jianhan Chen, Charles L Brooks, and Jana Khandogin. Recent advances in implicit solvent-based methods for biomolecular simulations. *Current Opinion in Structural Biology*, 18(2):140–148, 2008.

[125] Jaydeep P Bardhan. Nonlocal continuum electrostatic theory predicts surprisingly small energetic penalties for charge burial in proteins. *The Journal of chemical physics*, 135(10):104113, 2011.

[126] Haipeng Gong and Karl F Freed. Electrostatic solvation energy for two oppositely charged ions in a solvated protein system: salt bridges can stabilize proteins. *Biophysical journal*, 98(3):470–477, 2010.

[127] Yaakov Levy and José N Onuchic. Water mediation in protein folding and molecular recognition. *Annu. Rev. Biophys. Biomol. Struct.*, 35:389–415, 2006.

[128] Noel T Southall, Ken A Dill, and ADJ Haymet. A view of the hydrophobic effect. *The Journal of Physical Chemistry B*, 106(3):521–533, 2002.

[129] David Eisenberg and Andrew D McLachlan. Solvation energy in protein folding and binding. *Nature*, 1986.

[130] A Shrake and JA Rupley. Environment and exposure to solvent of protein atoms. lysozyme and insulin. *Journal of molecular biology*, 79(2):351–371, 1973.

[131] Sanzo Miyazawa and Robert L Jernigan. Estimation of effective interresidue contact energies from protein crystal structures: quasi-chemical approximation. *Macromolecules*, 18(3):534–552, 1985.

[132] Manfred J Sippl. Calculation of conformational ensembles from potentials of mena force: an approach to the knowledge-based prediction of local structures in globular proteins. *Journal of molecular biology*, 213(4):859–883, 1990.

[133] Manfred J Sippl. Knowledge-based potentials for proteins. *Current opinion in structural biology*, 5(2):229–235, 1995.

[134] Sanzo Miyazawa and Robert L Jernigan. Residue–residue potentials with a favorable contact pair term and an unfavorable high packing density term, for simulation and threading. *Journal of molecular biology*, 256(3):623–644, 1996.

[135] Min-yi Shen and Andrej Sali. Statistical potential for assessment and prediction of protein structures. *Protein science*, 15(11):2507–2524, 2006.

[136] Yves Dehouck, Dimitri Gilis, and Marianne Rooman. A new generation of statistical potentials for proteins. *Biophysical journal*, 90(11):4010–4017, 2006.

[137] George I Makhatadze and Peter L Privalov. Contribution of hydration to protein folding thermodynamics: I. the enthalpy of hydration. *Journal of molecular biology*, 232(2):639–659, 1993.

[138] Peter L Privalov and George I Makhatadze. Contribution of hydration to protein folding thermodynamics: Ii. the entropy and gibbs energy of hydration. *Journal of molecular biology*, 232(2):660–679, 1993.

[139] Benjamin Schuler, Everett A Lipman, and William A Eaton. Probing the free-energy surface for protein folding with single-molecule fluorescence spectroscopy. *Nature*, 419(6908):743–747, 2002.

[140] Christopher M Dobson. Experimental investigation of protein folding and misfolding. *Methods*, 34(1):4–14, 2004.

[141] K a Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24 (6):1501–9, 1985.

[142] Ken A Dill. Dominant forces in protein folding. *Biochemistry*, 29(31):7133–7155, 1990.

[143] Ken A Dill, Sarina Bromberg, Kaizhi Yue, Hue Sun Chan, Klaus M Ftebig, David P Yee, and Paul D Thomas. Principles of protein folding—a perspective from simple exact models. *Protein Science*, 4(4):561–602, 1995.

[144] Themis Lazaridis, Georgios Archontis, and Martin Karplus. Enthalpic contribution to protein stability: insights from atom-based calculations and statistical mechanics. *Advances in protein chemistry*, 47:231–306, 1995.

[145] George I Makhatadze and Peter L Privalov. Energetics of protein structure. *Advances in protein chemistry*, 47:307–425, 1995.

[146] José Nelson Onuchic, Zaida Luthey-Schulten, and Peter G Wolynes. Theory of protein folding: the energy landscape perspective. *Annual review of physical chemistry*, 48(1):545–600, 1997.

[147] Zhiping Weng, Charles Delisi, and Sandor Vajda. Empirical free energy calculation: comparison to calorimetric data. *Protein science*, 6(9):1976–1984, 1997.

[148] Robert L Baldwin and George D Rose. Molten globules, entropy-driven conformational change and protein folding. *Current opinion in structural biology*, 23(1): 4–10, 2013.

[149] Andrew J Doig and Michael JE Sternberg. Side-chain conformational entropy in protein folding. *Protein Science*, 4(11):2247–2251, 1995.

[150] J Alejandro D'Aquino, Javier Gómez, Vincent J Hilser, Kon Ho Lee, L Mario Amzel, and Ernesto Freire. The magnitude of the backbone conformational entropy change in protein folding. *Proteins: Structure, Function, and Bioinformatics*, 25 (2):143–156, 1996.

[151] Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, and David E Shaw. How fast-folding proteins fold. *Science*, 334(6055):517–520, 2011.

[152] Cyrus Levinthal. Are there pathways for protein folding. *J. Chim. phys*, 65(1): 44–45, 1968.

[153] Ken A Dill and Hue Sun Chan. From levinthal to pathways to funnels. *Nature structural biology*, 4(1):10–19, 1997.

[154] Peter E Leopold, Mauricio Montal, and José N Onuchic. Protein folding funnels: a kinetic approach to the sequence-structure relationship. *Proceedings of the National Academy of Sciences*, 89(18):8721–8725, 1992.

[155] Joseph D Bryngelson, Jose Nelson Onuchic, Nicholas D Socci, and Peter G Wolynes. Funnels, pathways, and the energy landscape of protein folding: a synthesis. *Proteins: Structure, Function, and Bioinformatics*, 21(3):167–195, 1995.

[156] John Cavanagh, Wayne J Fairbrother, Arthur G Palmer III, and Nicholas J Skelton. *Protein NMR spectroscopy: principles and practice*. Academic Press, 1995.

[157] David P Anderson. Boinc: A system for public-resource computing and storage. In *Grid Computing, 2004. Proceedings. Fifth IEEE/ACM International Workshop on*, pages 4–10. IEEE, 2004.

[158] Vijay S Pande, Ian Baker, Jarrod Chapman, Sidney P Elmer, Siraj Khaliq, Stefan M Larson, Young Min Rhee, Michael R Shirts, Christopher D Snow, Eric J Sorin, et al. Atomistic protein folding simulations on the submillisecond time scale using worldwide distributed computing. *Biopolymers*, 68(1):91–109, 2003.

[159] Mark S Friedrichs, Peter Eastman, Vishal Vaidyanathan, Mike Houston, Scott Legrand, Adam L Beberg, Daniel L Ensign, Christopher M Bruns, and Vijay S Pande. Accelerating molecular dynamic simulation on graphics processing units. *Journal of computational chemistry*, 30(6):864–872, 2009.

[160] David E Shaw, Martin M Deneroff, Ron O Dror, Jeffrey S Kuskin, Richard H Larson, John K Salmon, Cliff Young, Brannon Batson, Kevin J Bowers, Jack C Chao, et al. Anton, a special-purpose machine for molecular dynamics simulation. *Communications of the ACM*, 51(7):91–97, 2008.

[161] Bojan Zagrovic, Christopher D Snow, Michael R Shirts, and Vijay S Pande. Simulation of folding of a small alpha-helical protein in atomistic detail using worldwide-distributed computing. *Journal of molecular biology*, 323(5):927–937, 2002.

[162] Jed W Pitera and William Swope. Understanding folding and design: Replica-exchange simulations of"trp-cage"miniproteins. *Proceedings of the National Academy of Sciences*, 100(13):7587–7592, 2003.

[163] Vincent A Voelz, Gregory R Bowman, Kyle Beauchamp, and Vijay S Pande. Molecular simulation of ab initio protein folding for a millisecond folder ntl9 (1-39). *Journal of the American Chemical Society*, 132(5):1526–1528, 2010.

[164] David E Shaw, Paul Maragakis, Kresten Lindorff-Larsen, Stefano Piana, Ron O Dror, Michael P Eastwood, Joseph A Bank, John M Jumper, John K Salmon, Yibing Shan, et al. Atomic-level characterization of the structural dynamics of proteins. *Science*, 330(6002):341–346, 2010.

[165] Ken a Dill and Justin L MacCallum. The protein-folding problem, 50 years on. *Science (New York, N.Y.)*, 338(6110):1042–6, November 2012.

[166] Srivatsan Raman, Robert Vernon, James Thompson, Michael Tyka, Ruslan Sadreyev, Jimin Pei, David Kim, Elizabeth Kellogg, Frank DiMaio, Oliver Lange, et al. Structure prediction for casp8 with all-atom refinement using rosetta. *Proteins: Structure, Function, and Bioinformatics*, 77(S9):89–99, 2009.

[167] Yang Zhang. Interplay of i-tasser and quark for template-based and ab initio protein structure prediction in casp10. *Proteins: Structure, Function, and Bioinformatics*, 82(S2):175–187, 2014.

[168] K T Simons, C Kooperberg, E Huang, and David Baker. Assembly of protein tertiary structures from fragments with similar local sequences using simulated annealing and Bayesian scoring functions. *Journal of molecular biology*, 268(1): 209–25, April 1997.

[169] Martin Karplus and Andrej Šali. Theoretical studies of protein folding and unfolding. *Current opinion in structural biology*, 5(1):58–73, 1995.

[170] Alessio Bechini. On the characterization and software implementation of general protein lattice models. *PloS one*, 8(3):e59504, 2013.

[171] Sorin Istrail, Fumei Lam, et al. Combinatorial algorithms for protein folding in lattice models: a survey of mathematical results. *Communications in Information & Systems*, 9(4):303–346, 2009.

[172] B H Park and M Levitt. The complexity and accuracy of discrete state models of protein structure. *Journal of molecular biology*, 249(2):493–507, June 1995.

[173] Martin Mann, Rhodri Saunders, Cameron Smith, Rolf Backofen, and Charlotte M Deane. Producing high-accuracy lattice models from protein atomic coordinates including side chains. *Advances in bioinformatics*, 2012:148045, January 2012.

[174] Swakkhar Shatabda, MA Newton, Mahmood A Rashid, Duc Nghia Pham, and Abdul Sattar. How good are simplified models for protein structure prediction? *Advances in bioinformatics*, 2014, 2014.

[175] P Crescenzi, D Goldman, C Papadimitriou, a Piccolboni, and M Yannakakis. On the complexity of protein folding. *Journal of computational biology : a journal of computational molecular cell biology*, 5(3):423–65, January 1998.

[176] A A Albrecht, A Skaliotis, and K Steinhöfel. Stochastic protein folding simulation in the three-dimensional HP-model. *Computational biology and chemistry*, 32(4): 248–55, 2008.

[177] Andrzej Kolinski and Jeffrey Skolnick. Reduced models of proteins and their applications. *Polymer*, 45(2):511–524, 2004.

[178] Peter H Verdier and WH Stockmayer. Monte carlo calculations on the dynamics of polymers in dilute solution. *The Journal of Chemical Physics*, 36(1):227–235, 1962.

[179] HJ Hilhorst and JM Deutch. Analysis of monte carlo results on the kinetics of lattice polymer chains with excluded volume. *The Journal of Chemical Physics*, 63 (12):5153–5161, 1975.

[180] Neal Madras and Alan D Sokal. The pivot algorithm: a highly efficient monte carlo method for the self-avoiding walk. *Journal of Statistical Physics*, 50(1-2): 109–186, 1988.

[181] Hans-Joachim Böckenhauer, Abu Zafer M Dayem Ullah, Leonidas Kapsokalivas, and Kathleen Steinhöfel. A local move set for protein folding in triangular lattice models. In *Algorithms in Bioinformatics*, pages 369–381. Springer, 2008.

[182] JM Deutsch. Long range moves for high density polymer simulations. *The Journal of chemical physics*, 106(21):8849–8854, 1997.

[183] Dániel Györffy, Péter Závodszky, and András Szilágyi. "Pull moves" for rectangular lattice polymer models are not fully reversible. *IEEE/ACM transactions on computational biology and bioinformatics*, 9(6):1847–9, 2012.

[184] Ron Unger and John Moult. Genetic algorithms for protein folding simulations. *Journal of molecular biology*, 231(1):75–81, 1993.

[185] A. Piccolboni and G. Mauri. Application of evolutionary algorithms to protein folding prediction. In Jin-Kao Hao, Evelyne Lutton, Edmund Ronald, Marc Schoenauer, and Dominique Snyers, editors, *Artificial Evolution*, volume 1363 of *Lecture Notes in Computer Science*, pages 123–135. Springer Berlin Heidelberg, 1998.

[186] Md Tamjidul Hoque, Madhu Chetty, and Abdul Sattar. Protein folding prediction in 3d fcc hp lattice model using genetic algorithm. In *Evolutionary Computation, 2007. CEC 2007. IEEE Congress on*, pages 4138–4145. IEEE, 2007.

[187] Jyh-Jong Tsay and Shih-Chieh Su. An effective evolutionary algorithm for protein folding on 3d fcc hp model by lattice rotation and generalized move sets. *Proteome science*, 11(Suppl 1):S19, 2013.

[188] Faming Liang and Wing Hung Wong. Evolutionary Monte Carlo for protein folding simulations. *The Journal of Chemical Physics*, 115(7):3374, 2001.

[189] Peter Grassberger. Pruned-enriched rosenbluth method: Simulations of θ polymers of chain length up to 1 000 000. *Physical Review E*, 56(3):3682, 1997.

[190] Hsiao-Ping Hsu and Peter Grassberger. A Review of Monte Carlo Simulations of Polymers with PERM. *Journal of Statistical Physics*, 144(3):597–637, 2011.

[191] HP Hsu, Vishal Mehra, and Peter Grassberger. Growth algorithms for lattice heteropolymers at low temperatures. *Physical review E*, 68, 2003.

[192] Wenqi Huang, Zhipeng Lü, and He Shi. Growth algorithm for finding low energy configurations of simple lattice proteins. *Physical Review E*, 72(1):016704, 2005.

[193] Alena Shmygelska and Holger H Hoos. An ant colony optimisation algorithm for the 2D and 3D hydrophobic polar protein folding problem. *BMC bioinformatics*, 6:30, 2005.

[194] Paul Shaw. Using constraint programming and local search methods to solve vehicle routing problems. In *Principles and Practice of Constraint Programming—CP98*, pages 417–431. Springer, 1998.

[195] Thomas Wüst and DP Landau. The hp model of protein folding: A challenging testing ground for wang–landau sampling. *Computer Physics Communications*, 179(1):124–127, 2008.

[196] Thomas Wüst and David P Landau. Versatile approach to access the low temperature thermodynamics of lattice polymers and proteins. *Physical review letters*, 102(17):178101, 2009.

[197] Ying Wai Li, Thomas Wüst, and David P Landau. Monte carlo simulations of the hp model (the "ising model" of protein folding). *Computer physics communications*, 182(9):1896–1899, 2011.

[198] Thomas Wüst, Ying Wai Li, and David P Landau. Unraveling the beautiful complexity of simple lattice model polymers and proteins using wang-landau sampling. *Journal of Statistical Physics*, 144(3):638–651, 2011.

[199] Jingfa Liu, Gang Li, Jun Yu, and Yonglei Yao. Heuristic energy landscape paving for protein folding problem in the three-dimensional hp lattice model. *Computational biology and chemistry*, 38:17–26, 2012.

[200] Rolf Backofen and Sebastian Will. A constraint-based approach to fast and exact structure prediction in three-dimensional protein models. *Constraints*, 11(1): 5–30, 2006.

[201] Jorge J Moré and Zhijun Wu. Global continuation for distance geometry problems. *SIAM J. Optimiz.*, 7(3):814–836, 1997.

[202] Qunfeng Dong and Zhijun Wu. A linear-time algorithm for solving the molecular distance geometry problem with exact inter-atomic distances. *J. Global Optim.*, 22(1-4):365–375, 2002.

[203] Atilla Sit, Zhijun Wu, and Yaxiang Yuan. A geometric buildup algorithm for the solution of the distance geometry problem using least-squares approximation. *B. Math. Biol.*, 71(8):1914–33, 2009.

[204] Carlile Lavor, Leo Liberti, Antonio Mucherino, and Nelson Maculan. On a discretizable subclass of instances of the molecular distance geometry problem. In *Proc. of the 2009 ACM symposium on Applied Computing*, pages 804–805. ACM, 2009.

[205] J Reiterman, V Rödl, and E Šinajová. Geometrical embeddings of graphs. *Discrete Math.*, 74:291–319, 1989.

[206] Bruce Hendrickson. The molecule problem exploiting structure in global optimization. *SIAM J. Optimiz.*, 5(4):835–857, 1995.

[207] L. Roth and B. Asimow. The Rigidity of Graphs. *Trans. Amer. Math. Soc.*, 245: 279–289, 1978.

[208] G. M. Crippen and T. F. Havel. Distance geometry and molecular conformation. *J. Comput. Chem.*, 11(2):265–266, 1990.

[209] Timothy F. Havel. Distance geometry: Theory, algorithms and chemical applications. In *Encyclopedia of Computational Chemistry*, pages 723–742. John Wiley & Sons, 1998.

[210] JJ Moré and Zhijun Wu. Distance geometry optimization for protein structures. *J. Global Optim.*, 15(3):219–234, 1999.

[211] Lucjan Piela, Jaroslaw Kostrowicki, and Harold A. Scheraga. On the multiple-minima problem in the conformational analysis of molecules: deformation of the potential energy hypersurface by the diffusion equation method. *J. Phys. Chem.*, 93(8):3339–3346, 1989.

[212] Jeffrey Skolnick. In quest of an empirical potential for protein structure prediction. *Current opinion in structural biology*, 16(2):166–71, April 2006.

[213] Ruti Kapon, Reinat Nevo, and Ziv Reich. Protein energy landscape roughness. *Biochemical Society transactions*, 36(Pt 6):1404–8, December 2008.

[214] Sorin Istrail and Fumei Lam. Combinatorial algorithms for protein folding in lattice models: A survey of mathematical results. *Communications in Information and Systems*, pages 1–40, 2009.

[215] Yoshimi Fujitsuka, Shoji Takada, Zaida a Luthey-Schulten, and Peter G Wolynes. Optimizing physical energy functions for protein folding. *Proteins*, 54(1):88–103, January 2004.

[216] 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*, 92(1):325–329, 1995.

[217] http://www.cs.sandia.gov/tech_reports/compbio/tortilla-hp-benchmarks.html.

[218] Martin Mann, Sebastian Will, and Rolf Backofen. CPSP-tools–exact and complete algorithms for high-throughput 3D lattice protein studies. *BMC bioinformatics*, 9: 230, 2008.

[219] Source code for the REMC method. URL http://www.cs.ubc.ca/labs/beta/Projects/REMC-HPPFP/.

[220] Eaton E Lattman, Klaus M Fiebig, and Ken A Dill. Modeling compact denatured states of proteins. *Biochemistry*, 33(20):6158–6166, 1994.

[221] Francisco Herrera, Manuel Lozano, and Jose L. Verdegay. Tackling real-coded genetic algorithms: Operators and tools for behavioural analysis. *Artif. Intell. Rev.*, 12(4):265–319, 1998.

[222] Liu Hong and Jinzhi Lei. Scaling law for the radius of gyration of proteins and its dependence on hydrophobicity. *J. Polym. Sci. Pol. Phys.*, 47(2):207–214, 2009.

[223] Natalio Krasnogor and James Smith. A tutorial for competent memetic algorithms: model, taxonomy, and design issues. *IEEE T. Evolut. Comput.*, 9(5):474–488, 2005.

[224] LC Cross and W Kline. *Rules for the Nomenclature of Organic Chemistry Section E-stereo-chemistry*. Perg. Press, 1976.

[225] Michel Petitjean. CHIRALITY IN METRIC SPACES. *Symmetry: Culture and Science*, 21:27–36, 2010.

[226] AB Harris, RD Kamien, and TC Lubensky. Molecular chirality and chiral parameters. *Rev. Mod. Phys.*, 71(5):1745–1757, 1999. ISSN 0034-6861.

[227] CUDA Nvidia. Programming guide, 2015.

[228] Helen M Berman, John Westbrook, Zukang Feng, Gary Gilliland, TN Bhat, Helge Weissig, Ilya N Shindyalov, and Philip E Bourne. The protein data bank. *Nucleic acids research*, 28(1):235–242, 2000.

[229] RCSB Protein Data Bank. www.rcsb.org.

[230] Yang Zhang and Jeffrey Skolnick. Tm-align: a protein structure alignment algorithm based on the tm-score. *Nucleic acids research*, 33(7):2302–2309, 2005.

[231] Forbes J. Burkowski. *Structural Bioinformatics: An Algorithmic Approach*. Chapman & Hall, 2009.

[232] Tamar Schlick. *Molecular modeling and simulation: an interdisciplinary guide*. Springer, 2010.

[233] The PyMOL Web Page. http://www.pymol.org.

Andrea Gaetano Citrolo

B.Sc., M.Sc.

Curriculum vitæ

PERSONAL DETAILS

Birth	24/04/1985
Address	Via Bolzano 30, I-20127 Milan, Italy
Affiliation	Dipartimento di Informatica Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, Viale Sarca 336, Milan, 20126, Italy
Contact	✉ andrea.citrolo@disco.unimib.it ☎ (+39) 02 64487879 (+39) 328 2871525 http://www.disco.unimib.it/

RESEARCH INTERESTS

Computational Structural Biology, Combinatorial Optimization, Machine Learning, Synthetic Biology

EDUCATION

Ph.D. in Computer Science 2012–present

Università degli Studi di Milano-Bicocca, Italy

Supervisor: Prof. Giancarlo Mauri

Thesis title: Novel Computational Approaches for Protein Structure prediction and Optimization

M.Sc. in Bioinformatics 2009–2011

Università degli Studi di Milano-Bicocca, Italy

Supervisors: Prof. Luca De Gioia, Prof. Leonardo Vanneschi

Thesis title: Genetic programming applied to statistical potentials optimization for protein structure evaluation

Final grade: 110/110 magna cum laude

B.Sc. in Biotechnology 2005–2008

Università degli Studi di Milano-Bicocca, Italy

Supervisor: Prof. Luca De Gioia

Thesis title: Applications of computational design to protein stability and binding specificity

Final grade: 110/110 cum laude

PUBLICATIONS

1. Citrolo, A. G. and Mauri, G. (2014). A local landscape mapping method for protein structure prediction in the hp model. *Natural Computing*, 13(3):309–319
2. Nobile, M., Citrolo, A., Cazzaniga, P., Besozzi, D., and Mauri, G. (2014). A memetic hybrid method for the molecular distance geometry problem with incomplete information. In *Evolutionary Computation (CEC), 2014 IEEE Congress on*, pages 1014–1021. IEEE
3. Citrolo, A. and Mauri, G. (2013). A hybrid monte carlo ant colony optimization approach for protein structure prediction in the hp model. *Electronic Proceedings in Theoretical Computer Science*, 130:61–69
4. Bianco, S. and Citrolo, A. G. (2013). High contrast color sets under multiple illuminants. In *Computational Color Imaging*, pages 133–142. Springer

COMPUTER SKILLS

Languages	MATLAB, C++, PYTHON
Libraries	GSL, Boost, NumPy
Development	Git, Gprof, Valgrind, Code::Blocks
OS	Windows, Linux
Bioinformatics	PYMOL, MOE, GROMACS, Modeller, web portals
Database	Toad, MySQL, Oracle DB
Editing	LaTeX, MS Office

TEACHING

1. 2013 and 2014, invited to give advanced lectures for the course Computational Biology, Master Degree in Computer Science, Università degli Studi di Milano-Bicocca.
2. 2013 and 2014, Lecturer for the course Introduction to Computer Science, Bachelor's Degree in Foreign Languages, Università degli Studi di Bergamo.
3. 2012, Lecturer and examiner for the course Computational Biology, Master Degree in Biology, Università degli studi di Milano-Bicocca.

PROFESSIONAL EXPERIENCE

Software Developer 2011

List S.p.A.

My role was the development of software for the management of middle and back office data in financial institutions. In particular, I was involved in the design and implementation of data structures and algorithms for the analysis, processing and storage of large amount of data and in the maintenance of the existing software infrastructure.

HONORS & AWARDS

MIUR doctoral fellowship 2012–2015

Università degli Studi di Milano-Bicocca

This fellowship is issued by the Italian Ministry of Education, Universities and Research (MIUR) basing on a competitive examination for qualified candidates.

Excellence Diploma 2009

Collegio di Milano

This award is issued by the Collegio di Milano (an institution recognized by MIUR) to undergraduate and graduate students basing on GPA and involvment in advanced courses.

REFERENCES

Prof. Giancarlo Mauri

Dipartimento di Informatica, Sistemistica e Comunicazione, Università degli Studi di Milano-Bicocca, I-20126 Milan, Italy.

✉ mauri@disco.unimib.it

Prof. Luca De Gioia

Dipartimento di Biotecnologie e Bioscienze, Università degli Studi di Milano-Bicocca, I-20126 Milan, Italy.

✉ luca.degioia@unimib.it