

Copyright
by
Matthew Edward Alden
2007

The Dissertation Committee for Matthew Edward Alden
certifies that this is the approved version of the following dissertation:

**MARLEDA: Effective Distribution Estimation Through
Markov Random Fields**

Committee:

Risto Miikkulainen, Supervisor

Chandrajit Bajaj

Robin Gutell

Raymond Mooney

Bruce Porter

**MARLEDA: Effective Distribution Estimation Through
Markov Random Fields**

by

Matthew Edward Alden, B.S.

DISSERTATION

Presented to the Faculty of the Graduate School of
The University of Texas at Austin
in Partial Fulfillment
of the Requirements
for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

December 2007

MARLEDA: Effective Distribution Estimation Through Markov Random Fields

Publication No. _____

Matthew Edward Alden, Ph.D.
The University of Texas at Austin, 2007

Supervisor: Risto Miikkulainen

Many problems within the biological sciences, such as DNA sequencing, protein structure prediction, and molecular docking, are being approached computationally. These problems require sophisticated solution methods that understand the complex natures of biological domains. Traditionally, such solution methods are problem specific, but recent advances in generic problem-solvers furnish hope for a new breed of computational tools. The challenge is to develop methods that can automatically learn or acquire an understanding of a complex problem domain.

Estimation of Distribution Algorithms (EDAs) are generic search methods that use statistical models to learn the structure of a problem domain. EDAs have been successfully applied to many difficult search problems, such as circuit design, optimizing Ising spin glasses, and various scheduling tasks. However, current EDAs contain ad hoc limitations that reduce their capacity to solve hard problems.

This dissertation presents a new EDA method, the Markovian Learning Estimation of Distribution Algorithm (MARLEDA), that employs a Markov random field model. The model is learned in a novel way that overcomes previous ad hoc limitations. MARLEDA is shown to perform well on standard benchmark search tasks. A multiobjective extension of MARLEDA is developed for use in predicting the secondary structure of RNA molecules. The extension is shown to produce high-quality predictions in comparison with several contemporary methods, laying the groundwork for a new computational tool for RNA researchers.

Table of Contents

List of Figures	ix
List of Tables	xi
Chapter 1. Introduction	1
1.1 Motivation	2
1.2 Approach	3
1.3 Results	6
1.4 Overview of the Dissertation	8
Chapter 2. Background	10
2.1 Combinatorial Optimization	10
2.2 Genetic Algorithms	13
2.3 Estimation of Distribution Algorithms	15
2.4 Markov Random Fields	19
2.5 Pearson's χ^2 Test	21
2.6 Conclusion	24
Chapter 3. The MARLEDA Method	26
3.1 Selection	26
3.2 The MRF Model	27
3.3 Generating Chromosomes	31
3.4 Replacement	32
3.5 Computational Complexity	33
3.6 Conclusion	34

Chapter 4. Benchmark Experiments	35
4.1 Experimental Design	35
4.2 OneMax	37
4.3 Deceptive Trap Functions	39
4.4 The Rosenbrock Function	44
4.5 Ising Spin Glasses	48
4.6 Lattice Proteins	53
4.7 Conclusion	61
Chapter 5. RNA Structure Prediction	62
5.1 RNA in Molecular Biology	62
5.2 RNA Structure Prediction	64
5.3 Multiobjective MARLEDA	73
5.4 Experimental Design	77
5.5 Results	83
5.6 Conclusion	93
Chapter 6. Discussion & Future Work	95
6.1 Learning Statistical Models	95
6.2 Using Hand-Crafted Statistical Models	99
6.3 Reviving Mutation	99
6.4 Scalability & RNA Secondary Structure Prediction	101
6.5 RNA Target Statistics	103
6.6 Conclusion	104
Chapter 7. Conclusion	105
7.1 Contributions	105
7.2 Conclusion	106
Appendix A. Bacterial 5S rRNA Statistics	108
A.1 Nucleotide Pairings	109
A.2 Pairing Patterns of Nucleotide 1-tuples	109
A.3 Pairing Patterns of Nucleotide 2-tuples	110

A.4	Pairing Patterns of Nucleotide 3-tuples	111
A.5	Double-helix Lengths	113
A.6	Double-helix Simple Spans	113
A.7	Double-helix Conditional Spans	114
A.8	Hairpin Loop Lengths	114
A.9	Non-hairpin Loop Lengths	115
	Bibliography	116
	Vita	127

List of Figures

2.1	Example learned gene-dependency structures for a fictional combinatorial optimization task with eight parameters.	16
2.2	An example undirected graph model for a fictional combinatorial optimization task with eight parameters.	20
4.1	Learning curves of the median best population member in 100 independent trials for an instance of OneMax with 300 bits.	38
4.2	Learning curves of the median best population member in 100 independent trials for 300-bit instances of a deceptive trap function.	42
4.3	Learning curves of the median best population member in 100 independent trials for Rosenbrock instances of 32 and 64 bits.	46
4.4	Representative learning curves of the median best fitness in 100 independent trials for an instance of an Ising spin glass system of 400 spins and 900 spins.	51
4.5	Two examples of lattice proteins embedding in a two-dimensional square lattice.	54
4.6	The ten lattice protein sequences from the Harvard vs. UCSF challenge.	56
4.7	Median best energy scores in 50 independent trials over the Harvard vs UCSF lattice protein set.	58
4.8	Four views of the best conformation for lattice protein #1 discovered by MARLEDA ^{+model}	60
5.1	Structural comparison of RNA and DNA.	63
5.2	Secondary structure and tertiary structure of the 5S subunit of ribosomes in <i>Escherichia coli</i>	66
5.3	Linear representation of the secondary structure of E. coli 5S rRNA.	69
5.4	Example minimization of a simple multiobjective problem, $f(x, y) = (x, y)$	76
5.5	The mMARLEDA encoding of the secondary structure of a fragment of RNA.	78

5.6	Two-dimensional projections of the Pareto optimal set from two mMARLEDA trials.	85
5.7	The best E. coli 5S rRNA secondary structure predictions by mMARLEDA when using the four prescribed target statistics and only two target statistics.	86
5.8	The best E. coli 5S rRNA secondary structure predictions by the mMARLEDA, Pfold, RNAalifold, RNAfold, and Mfold algorithms.	90

List of Tables

5.1	Two summary statistics for the 5S subunit of bacterial ribosomes.	73
5.2	Promising potential sets of target statistics for bacterial 5S rRNA, referenced by appendix entry.	81
5.3	Prediction accuracy of nucleotide pairs in the secondary structures of 22 bacterial 5S rRNA references.	89
5.4	Multiobjective fitness scores for the true secondary structure of E. coli 5S rRNA and several mMARLEDA predictions.	92

Chapter 1

Introduction

This dissertation focuses on a difficult problem within computational biology, that of predicting the secondary structure of RNA molecules. This problem is a simplification of RNA folding; rather than addressing how RNA molecules acquire their three-dimensional shape, this problem focuses on identifying the intra-molecular bonds that contribute to a molecule's shape. Determining an RNA molecule's secondary structure is an important step in determining its three-dimensional shape, which in turn defines the molecule's function. Accurate predictions are therefore critical to our understanding of these fundamental molecules of life.

RNA secondary structure is the product of complex physical processes that are not yet fully understood. However, as more and more secondary structures are discovered, it becomes feasible to approach structure prediction through computational search. The dissertation develops a new search method designed to learn and exploit the complexities of RNA secondary structures. The remainder of this chapter briefly describes the motivation, approach, and experimental results of this new method.

1.1 Motivation

In the past two decades computers have become increasingly indispensable tools for researchers in many fields. Some disciplines utilize computers due to the vast quantities of data involved, such as those available in the genetic library produced by the human genome project. Other fields need sophisticated algorithmic processing, such as robotic exploration of alien environments. Still others use computers for simulation, recreating their subjects in virtual environments that allow new observational and experimental methods. Computational science (or scientific computing) transforms physical and mathematical problems into computational problems. As computers become further integrated into the myriad branches of science, increasingly difficult problems are being solved computationally.

Computational biology is a computational science that promises tangible, life-altering benefits. Research in fields such as genomics, protein structure prediction, and molecular docking, finds application in pharmacology and gene therapy. Consequently, day-to-day medicine and health have been improved, with the greatest breakthroughs still to come. These fields directly depend on advances in computer hardware and software, and there is great demand for effective computational problem-solving methods.

Research in computational biology produces computational models of real-world objects or phenomena, such as molecular interactions or phylogenetic trees. The computational models can be studied, adjusted, and experimented with to generate predictions about their real-world counterparts. In

turn, real-world experiments help form the basis for new computational models. There is an interplay between real-world and computational research; each informs and refines the other. While a computational formulation of real-world problems is useful and convenient, there are drawbacks.

The key difficulty of many computational problems, such as DNA sequence alignment, selecting key attributes for data mining, or optimizing antenna design, is that no analytic solution methods exist. There are also problems that do have analytic solutions, but those methods are too computationally expensive to be practical. Without the ability to construct an ideal solution efficiently, we are often reduced to a procedure of guess-and-check. Such a procedure, formally known as *search*, tries to identify the best solution(s) among a set of candidates by systematically evaluating alternatives. Search algorithms are, in many cases, a relatively easy and effective means of producing near-optimal solutions to difficult problems. However, in complex domains search is often inefficient, taking too long to be practical. This dissertation develops a search method that handles complexity better by statistically modeling the problem domain.

1.2 Approach

Fortunately, even when analytical methods are not available, for many computational problems it is relatively easy to test the “correctness” of a potential solution. For example, the hypothesized three-dimensional structure of a protein can be evaluated on the energetic stability of that structure. It

is then reasonable to search for structures that are highly stable. This sort of feedback allows search algorithms to produce high-quality solutions without evaluating all possible solutions.

A variety of search methods exist, each suitable for some search problems and unsuitable for others. This dissertation is concerned with evolutionary search methods, which seek to harness the creativity and problem-solving ability of biological evolution. These methods are easily adapted to new domains, and have been successfully applied to numerous problems, such as cryptography, hardware design, and data compression. For example, the classic genetic algorithm (GA) [17, 32], modeled after natural evolution, combines simple operations such as crossover and mutation to form a generic search system. However, nature retains two important advantages over such an algorithm: massive parallelism and deep time. Simple operations may be sufficient to discover complex solutions in such a luxurious environment, but generating comparable results with human-reasonable resources requires greater sophistication.

Estimation of distribution algorithms (EDAs) [4, 11, 46, 50] are a new and powerful approach to search. The main idea is to combine statistical modeling with evolutionary search. These algorithms exploit statistically identifiable structure within a search domain to produce better solutions or to produce solutions more quickly. The statistical models can be defined a priori, injecting prior information into the search process, or learned as part of the algorithm's operation. The power of these algorithms therefore depends on two factors:

(1) how appropriate the statistical model is to the domain, and (2) how well the system can learn it.

The majority of EDAs incorporate models that organize domain structure in directed acyclic graphs (DAGs), such as Bayesian networks [34, 51]. There are many established learning mechanisms for DAGs and simple methods for sampling the resulting model. However, directed graph models are not necessarily the most natural representation of domain structure for all problems. For example, a DAG cannot represent, by definition, bi-directional or cyclic dependencies.

Several researchers have proposed using undirected graph models in EDAs [24, 60, 63]. In particular, Markov random fields (MRFs) have been shown to be a promising basis for EDA models. However, learning and sampling MRFs is more difficult than for DAGs, and has so far constrained their implementation.

The main contribution of this dissertation is the development of a new EDA, the Markovian Learning Estimation of Distribution Algorithm (MARLEDA), that can learn and use a general Markov random field model. The MRF model is constructed to reflect the interactions of the search domain's parameters. Awareness of these interactions allows MARLEDA to identify good solutions intelligently, making search efficient. Previous EDAs have used constrained forms of MRFs, which made their models easier to learn and sample, but also reduced their ability to efficiently search complex domains. MARLEDA uses a standard statistical hypothesis test, Pearson's χ^2 test, to

learn an unconstrained MRF, thus retaining the model’s full potential. This potential translates into superior search capability when compared to other methods.

1.3 Results

The effectiveness of the MARLEDA method is evaluated on five benchmark search problems, OneMax, deceptive trap functions, the Rosenbrock function, Ising spin glasses, and lattice proteins. MARLEDA performs well in these domains compared to a standard genetic algorithm and a state-of-the-art EDA, the Bayesian Optimization Algorithm (BOA) [53]. Beyond establishing MARLEDA’s competence, these experiments explore two interesting facets of the MARLEDA method, the role of mutation and the use of hand-crafted MRF models.

EDAs are based on the presumption that the structure of a problem domain can be represented by a statistical model, at least sufficiently well to support an effective search. If the model adequately captures the domain, then there is no need for a “primitive” search operation such as mutation, which randomly perturbs an algorithm’s exploration of the problem domain. Consequently, most EDAs do not include mutation.

However, the problem domain is only partially observable to an EDA. Partial information coupled with even a perfect model can still lead to incorrect conclusions about the composition of good solutions, thus reducing the search’s effectiveness. Since there is a non-zero chance that such an error will occur,

there is a non-zero chance that an alternate, randomly chosen decision will correct it. In light of this possibility, MARLEDA retains the classic mutation operation. MARLEDA’s empirical results on the benchmark problems show that mutation is still useful in complex search domains.

For many search problems, there is partial or complete knowledge of the relationships among the problem’s parameters. The statistical models used by EDAs can be hand-constructed to include such information, thus improving search effectiveness. In four of the five benchmark experiments, MARLEDA is evaluated when using a fixed, hand-crafted model based on the known structure of the problem domain. In all cases, MARLEDA’s effectiveness is improved by providing it with this additional knowledge. Furthermore, the improvement is more dramatic in more complex search domains. This makes human knowledge increasingly valuable as EDAs transition to complex real-world search problems.

The final proving ground for MARLEDA is the RNA secondary structure prediction problem. To determine the correctness of a hypothesized secondary structure, MARLEDA compares the secondary structure to a set of statistics collected from a database of known RNA secondary structures. MARLEDA then searches for secondary structures that match the target statistics. In order to compare secondary structures along several axes, a multiobjective extension to MARLEDA is developed, based on well-established multiobjective techniques.

The set of target statistics must be chosen with care. Each statistic

must be informative but not redundant. To maintain efficiency, the set of statistics should be as small as possible. Based on these criteria, four statistics are chosen for the target set, each describing a different aspect of RNA secondary structure. Using this target set, MARLEDA makes high-quality predictions for a specific class of small RNA molecule. In future research, MARLEDA will be scaled-up to larger RNA molecules.

1.4 Overview of the Dissertation

This dissertation is divided into five main parts: background (chapter 2), the MARLEDA method (chapter 3), evaluation (chapter 4), application (chapter 5), and discussion & conclusion (chapters 6 & 7).

Chapter 2 reviews previous work on EDAs and presents and basic theory behind Markov random fields and Pearson’s χ^2 test.

Chapter 3 describes the MARLEDA method in detail. Particular emphasis is placed on the learning and sampling mechanisms for MARLEDA’s MRF model.

Chapter 4 compares MARLEDA with a genetic algorithm and the Bayesian Optimization Algorithm on five benchmark search problems. The experiments demonstrate MARLEDA’s success in both learning an MRF model and using a hand-crafted MRF model.

Chapter 5 describes the application of MARLEDA to the RNA secondary structure prediction problem. A multiobjective extension of MARLEDA

is developed that allows secondary structures to be evaluated by several metrics simultaneously. This extension enables MARLEDA to make high-quality predictions when compared to several contemporary prediction methods.

Chapter 6 discusses the results of the MARLEDA experiments and their implications. Several directions for future research are presented, focusing on extending the MARLEDA method and improving its application in the RNA domain.

Chapter 7 reviews the major contributions of this research and concludes this dissertation.

Chapter 2

Background

The study of search and search algorithms is extremely broad, covering a large array of concepts, disciplines, and computational techniques. This dissertation addresses one particular class of search problem, combinatorial optimization, that contains many interesting and important computational problems. The next section defines and motivates combinatorial optimization problems. Sections 2.2 & 2.3 review two approaches to combinatorial optimization, genetic algorithms and estimation of distribution algorithms. Lastly, the basic theory behind MARLEDA's statistical model, Markov random fields and Pearson's χ^2 test, is presented.

2.1 Combinatorial Optimization

A finite-valued combinatorial optimization task consists of a candidate solution space, \mathbf{X} , and an objective function, $f : \mathbf{X} \rightarrow \mathbb{R}$. The goal is to find a solution, $\mathbf{x}^* \in \mathbf{X}$, that either maximizes or minimizes the objective function, depending on the task. Candidate solutions are composed of many individual parameters, each with a finite number of possible values, thus the size of the solution space is combinatorial in the number of task parameters.

A wide variety of problems can be formulated as combinatorial optimization tasks, ranging from classic computer science problems such as bin packing and path finding to real-world problems like antenna design and robot navigation. Of particular interest to this dissertation are the challenges of computational biology, such as protein folding and molecular docking, whose effective solutions will have profoundly positive effects on our future health and well-being. These challenges are also among the most difficult facing researchers today and require the development of sophisticated computational methods.

Difficult combinatorial optimization tasks often lack analytic solution methods, thus solutions can only be discovered via search. However, because the solution spaces of interesting combinatorial optimization tasks are very large, systematic search is computationally intractable. It is sometimes possible, however, to solve such tasks approximately using probabilistic search methods.

Probabilistic search algorithms forgo systematic exploration of a candidate solution space in favor of randomized search strategies. Guided by an objective function, such techniques attempt to explore only those regions of the solution space in which high-quality solutions are likely to be found. However, because probabilistic search algorithms do not search the entire space, they are generally not *complete*, i.e. they are not guaranteed to identify optimal solutions. In spite of this weakness, by searching only a fraction of the solution space probabilistic search algorithms are practical methods for producing

near-optimal (and occasionally optimal) solutions quickly.

Two of the simplest probabilistic heuristic search methods are hill climbing and simulated annealing. Hill climbing is a local search method that explores a solution space via a path of progressively improving solutions. As the name suggests, this process is analogous to climbing a hill; taking small steps to increase one’s elevation (solution quality) until arriving at the peak (optimum). Starting from an initial point within the solution space, the algorithm evaluates a neighborhood of “adjacent” solutions. If a superior solution is found within the neighborhood, that solution becomes the initial point for the next iteration of the algorithm. This process continues until no further improvement is possible. Simple optimization tasks can be solved by hill climbing algorithms effectively. However, interesting optimization tasks generally have solution spaces with many local optima (multiple hills) that trap such methods into suboptimal solutions.

Simulated annealing [35] is inspired by the annealing of metals, a process of temperature variation designed to improve a metal’s crystalline structure. Similar to hill climbing, simulated annealing searches a solution space via a path of neighboring solutions. Unlike hill climbing, the path need not contain strictly improving solutions, thus simulated annealing can “escape” local optima. The probability of traversing a solution of lower quality is parametrized by a temperature. At high temperatures, the probability of traversing an inferior solution is high, while at low temperatures the probability is low. Simulated annealing systems typically use an initially high temperature that

decays over the course of the search. Consequently, the algorithm initially explores the solution space in an unconstrained fashion, similar to a random search, but then settles on one local optimum. If the annealing takes place slowly enough, it is likely that the local optimum is also the global optimum. Though simulated annealing avoids the greatest weakness of hill climbing, it is not an efficient search. More sophisticated methods are needed for complex domains.

2.2 Genetic Algorithms

Genetic algorithms (GAs) [17, 32] are a well-established search method that has been successfully applied to a wide range of computational problems, such as planning, engineering design, and control. The basic principle underlying GAs makes them well suited for combinatorial optimization tasks. GAs perform a parallel search, maintaining a *population* of candidate solutions that evolve over time. Evolution is guided by the objective function of the task, commonly called the *fitness function* or *fitness metric*. In keeping with GA conventions, “high-fitness” refers to desirable fitness function scores, i.e. high or low values depending on the direction of optimization. Similarly, “low-fitness” refers to undesirable fitness scores. The GA search mechanism is designed to visit candidate solutions of ever improving fitness at each iteration of the algorithm.

Following the biological analogy, candidate solutions are encoded in artificial *chromosomes*. A chromosome is composed of a set of *genes*, each

representing a parameter of the optimization task (in practice there may be an additional mapping between a chromosomal encoding and a candidate solution, but for notational simplicity the space of chromosomes and the space of candidate solutions are assumed to be the same in this chapter). The value of each gene, its *allele*, is one of the possible values for the associated task parameter.

The canonical genetic algorithm proceeds as follows:

-
1. The initial population of chromosomes, $\mathcal{P}(0)$, is uniformly sampled from \mathbf{X} .
 2. At iteration t , a subcollection, $\mathcal{P}'(t) \subseteq \mathcal{P}(t)$, of high-fitness chromosomes is selected.
 3. The members of $\mathcal{P}'(t)$ are recombined to form a collection of new chromosomes, $\mathcal{C}(t)$.
 4. The members of $\mathcal{C}(t)$ have some of their genes mutated.
 5. A subcollection, $\mathcal{R}(t) \subseteq \mathcal{P}(t)$, of low-fitness chromosomes is selected.
 6. $\mathcal{P}(t+1) \leftarrow \mathcal{P}(t) - \mathcal{R}(t) + \mathcal{C}(t)$.
 7. Unless termination criteria are met, return to step 2.
-

The classic selection, recombination, and mutation operations govern the search behavior of GAs. These operators work well, provided high-fitness chromosomes are located “near” other high-fitness chromosomes or their recombinations. GAs perform very well on tasks where such assumptions are true, i.e. tasks with *building blocks* [17]. However, GAs are less effective on

tasks in more structured domains where combinations of genes must be correctly set to achieve high fitness.

2.3 Estimation of Distribution Algorithms

Estimation of distribution algorithms (EDAs) [4, 11, 46, 50] address the building block problem by statistically inferring dependencies among genes. These dependencies are expressed in a statistical model, which can then be sampled to produce a new population. Through the sampling process, EDAs are likely to preserve high-fitness combinations of alleles, making the search process more efficient.

EDAs operate similarly to GAs, but sampling of the statistical model replaces the recombination and mutation operations:

-
1. The initial population of chromosomes, $\mathcal{P}(0)$, is uniformly sampled from \mathbf{X} and the model, $\mathcal{M}(0)$, is initialized.
 2. At iteration t , a subcollection, $\mathcal{P}'(t) \subseteq \mathcal{P}(t)$, of high-fitness chromosomes is selected.
 3. $\mathcal{M}(t)$ is created to model the members of $\mathcal{P}'(t)$.
 4. A collection of new chromosomes, $\mathcal{C}(t)$, is produced by sampling $\mathcal{M}(t)$.
 5. A subcollection, $\mathcal{R}(t) \subseteq \mathcal{P}(t)$, of low-fitness chromosomes is selected.
 6. $\mathcal{P}(t+1) \leftarrow \mathcal{P}(t) - \mathcal{R}(t) + \mathcal{C}(t)$.
 7. Unless termination criteria are met, return to step 2.
-

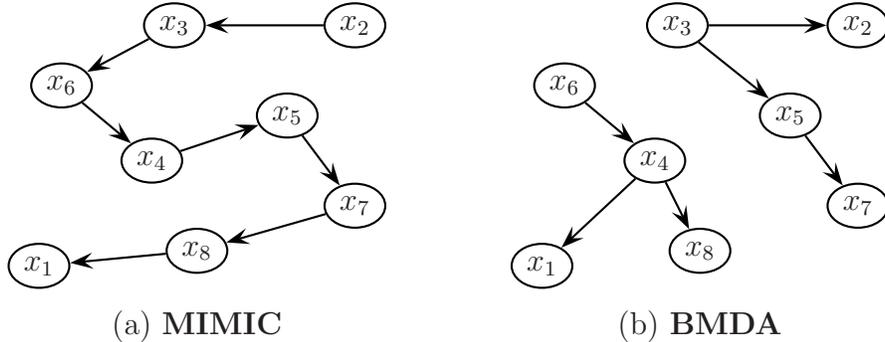


Figure 2.1: Example learned gene-dependency structures for a fictional combinatorial optimization task with eight parameters. While both DAG-based methods, MIMIC and BMDA enforce very different global organization of dependencies. Domains that do match the assumed organization will be difficult to optimize, therefore the choice of statistical model is critically important when addressing a specific optimization task.

The simplest EDAs employ univariate models [3, 4, 11, 26, 46, 65] that do not identify gene dependencies. These models essentially record the marginal frequency of alleles for every gene. Let x_i be the i^{th} gene of chromosome \mathbf{x} . New chromosome are constructed by sampling the modeled distribution

$$P(\mathbf{x}) = \prod_{i=1}^n P(x_i). \quad (2.1)$$

Bivariate EDAs [4, 5] model dependencies between pairs of genes. Formally, the model is

$$P(\mathbf{x}) = \prod_{i=1}^n P(x_i | \text{parent-of}(x_i)). \quad (2.2)$$

Bivariate EDAs are distinguished by the restrictions placed on the parent-child relationship. For example, the Mutual Information Maximization for

Input Clustering algorithm (MIMIC) [11] learns a chain of gene dependencies, while the Bivariate Marginal Distribution Algorithm (BMDA) [55] learns a forest of tree dependencies, as illustrated in figure 2.1.

Multivariate EDAs [1, 15, 47, 48, 49, 57] model dependencies among larger sets of genes. One of the most successful multivariate EDAs is the Bayesian Optimization Algorithm (BOA) [53] and its hierarchical extension (hBOA) [52]. BOA and hBOA use a Bayesian network as the basis for their statistical model, capturing dependencies organized into a directed acyclic graph (DAG). The Bayesian network model is constructed to approximately minimize the Bayesian-Dirichlet difference [9, 28] between the model and the chromosome population, thus promoting an accurate model. The combination of a flexible model and a strong learning mechanism has made BOA an outstanding algorithm in its class. Consequently, BOA is used for comparison with MARLEDA on the benchmark experiments presented in chapter 4.

The above bivariate and multivariate EDAs are based on directed graph models (or equivalent). Undirected graph models have been explored in a number of algorithms including the Extended Compact Genetic Algorithm (EcGA) [24], the Markov Network Factorized Distribution Algorithm (MN-FDA) [60], the Markov Network Estimation of Distribution Algorithm (MN-EDA) [61], and Distribution Estimation Using Markov Random Fields (DEUM) [62, 63]. Undirected graph models have been shown to be superior to their directed graph model counterparts for many optimization tasks, making them strong candidates for further research.

The greatest challenge in using undirected graph models is that it is difficult to learn and sample them. The Markov random field models favored by recent EDAs and used in MARLEDA have been previously applied to problems in physics and image processing [42, 68]. In such applications, they are traditionally not learned. In the few cases where learning is involved, it is only applied to the conditional probabilities composing the MRF, not the MRF neighborhood system itself (as defined in the next section). However, EDAs must be able to learn the neighborhood system as well. Consequently, heuristic schemes have been developed for learning both aspects of Markov random fields to that they can be used with EDAs [60, 61, 62]. While these schemes draw on statistical methods, they generally lack the rigorous theoretical foundation available to directed graph methods.

Sampling undirected graph models is difficult because the modeled variables are treated as inherently homologous. Directed graph models, and the conditional probability distributions encoded therein, define a natural ordering of the nodes of a DAG, such as the ordering achieved by “tracing” the dependency chain in figure 2.1(a). This ordering makes it possible to evaluate the model efficiently via standard graph traversal algorithms. In contrast, undirected graph models express relations that hold across many sets of variables simultaneously and provide no natural node ordering. Consequently, learning and sampling undirected graph models is considerably more costly than directed graph models.

To make these processes tractable, existing algorithms artificially con-

strain their undirected graph models. These constraints typically take the form of limits on the complexity of the model or conversions of the model to simpler forms. For example, the DEUM algorithm uses a univariate MRF model and its extension, Is-DEUM, employs a bivariate MRF model [62]. The MN-FDA and MN-EDA methods use junction graph and Kikuchi approximations to factorize the structure of their MRF models [60, 61]. In effect, the MRF models are simplified or mixed with directed relations to support less intensive MRF processing.

Such constraints make it practical to learn and sample the MRF models. However, they make the models less flexible and the overall search process potentially less effective. Therefore, the main contribution of this dissertation is to develop mechanisms for learning and sampling an unconstrained multivariate undirected graph models, such as Markov random fields.

2.4 Markov Random Fields

A Markov random field defines the joint probability distribution of a set of random variables in terms of *local characteristics*, i.e. joint or conditional probability distributions of subsets of the random variables. Let $\{X_1, \dots, X_n\}$ be a set of finite random variables and let \mathbf{X} be the space of *configurations* of $\{X_1, \dots, X_n\}$, i.e.

$$\mathbf{X} = \prod_{i=1}^n X_i. \quad (2.3)$$

A probability measure, P , is a *random field* with regard to \mathbf{X} if it is strictly

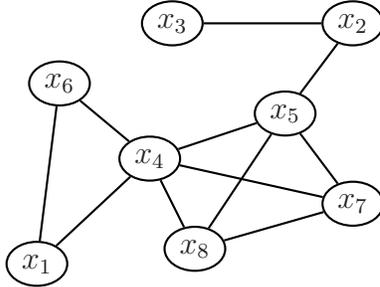


Figure 2.2: An example undirected graph model for a fictional combinatorial optimization task with either parameters. Each variable (node) statistically depends on the variables adjacent to it. In contrast to the DAG models of figure 2.1, this undirected graph model has no natural ordering of its nodes, making learning and sampling the model more difficult.

positive and normalized, i.e.

$$\begin{cases} P(\mathbf{x}) > 0, \quad \forall \mathbf{x} \in \mathbf{X} \\ \sum_{\mathbf{x} \in \mathbf{X}} P(\mathbf{x}) = 1. \end{cases} \quad (2.4)$$

A *neighborhood system*, ∂ , defines a set of neighbors for each random variable such that

$$\begin{cases} i \notin \partial(i) \\ i \in \partial(j) \iff j \in \partial(i). \end{cases} \quad (2.5)$$

The Markov property then induces an MRF on P given ∂ such that

$$P(x_i | x_j, i \neq j) = P(x_i | x_k, k \in \partial(i)). \quad (2.6)$$

The neighborhood system and Markov property together imply that each random variable is statistically dependent on all random variables inside its neighborhood, and statistically independent of all random variables outside

its neighborhood (given the neighborhood set). The neighborhood system can be interpreted as an undirected graph, such as the one in figure 2.2.

MRFs can be equivalently defined in terms of a set of potential functions over cliques in the neighborhood system. However, the MRF learning and sampling mechanisms described in sections 3.2 & 3.3 are more readily understood in terms of the above definitions. In particular, this definition also allows the MRF to be readily used as an EDA model. First, it is defined in terms of conveniently computable conditional probabilities, like those commonly used in EDAs. Second, each gene x_i is conveniently modeled by a random variable X_i , which makes the candidate solution and configuration spaces equivalent (\mathbf{X}). Third, the neighbor relation can apply to any pair of random variables; there are no ad hoc constraints on the structure of the neighborhood system. This dissertation further shows that the MRF neighborhood system can be learned using a metric for statistical dependence, χ^2 , which will be described next.

2.5 Pearson's χ^2 Test

Statistical hypothesis tests are commonly used to measure significance of statistical comparisons. For example, Student's t-test [56] determines if the means of two Gaussian distributions are statistically distinct. The test results in a *confidence level* (or *p-value*) that is usually compared with a desired value, e.g. 0.95 or 0.99, to decide whether the difference in means is statistically significant.

The confidence level can also be used as a direct measure of statistical dependence. This approach leads to Pearson's χ^2 test, which is a non-parametric statistical hypothesis test that measures the degree of similarity between two distributions of nominal (orderless) data. When used with data sampled from nominal random variables, the test measures the degree of dependence among the random variables. Pearson's χ^2 test compares two frequency distributions (FDs), typically an observed FD and an expected FD:

$$\chi^2 = \sum_{\mathbf{x} \in \mathbf{X}} \frac{(\mathbb{F}_{\text{obs}}(\mathbf{x}) - \mathbb{F}_{\text{exp}}(\mathbf{x}))^2}{\mathbb{F}_{\text{exp}}(\mathbf{x})}. \quad (2.7)$$

If \mathbb{F}_{exp} is constructed to represent the null hypothesis (of independence) among a set of random variables $\{X_m, \dots, X_t\}$ and differs notably (confidence level near 1) from \mathbb{F}_{obs} , then \mathbb{F}_{obs} is presumed to be the product of dependencies among $\{X_m, \dots, X_t\}$.

Like most statistical hypothesis tests, the confidence level of Pearson's χ^2 test depends on the degrees of freedom of the hypothesis in addition to the χ^2 value. Degrees of freedom are generally defined as the number of estimate variables minus the number of independent constraints on those variables. Consider an example contingency table showing the joint frequency distribution of two nominal random variables, X_i and X_j , each with four possible values:

$X_i \backslash X_j$	α	β	γ	δ
α	4	5	10	3
β	18	5	3	9
γ	9	7	7	2
δ	34	8	12	15

Modeling this frequency distribution requires 16 estimate variables, one for each cell of the table. The sum of samples in each row and column provides eight constraints on those estimate variables. However, only seven of the constraints are independent, since the total number of samples can be obtained by summing the number of samples across all rows or, equivalently, across all columns. There are therefore $16 - 7 = 9$ degrees of freedom in the model.

More generally, the degrees of freedom, δ , for any two-dimensional contingency table with r rows and c columns can be calculated as

$$\delta = rc - (r + c - 1) = rc - r - c + 1 = (r - 1)(c - 1). \quad (2.8)$$

This derivation is valid when the frequency distribution is well covered, and extends naturally to joint frequency distributions of more than two random variables.

However, derivation (2.8) overestimates the true degrees of freedom of systematically sparse frequency distributions, like those typically occurring during population-based searches. Consider the following modification of the previous contingency table:

$X_i \backslash X_j$	α	β	γ	δ
α	4	5	0	0
β	18	5	0	0
γ	9	7	0	0
δ	34	8	0	0

In this example, two possible values for X_j , γ and δ , are missing from the frequency distribution. In effect, insufficient sampling has systematically reduced the domain of X_j . Modeling the frequency distribution no longer requires 16 estimate variables, but only eight. The other “missing” eight no longer need to be modeled because their value is systematically zero. The degrees of freedom of the model are reduced accordingly, from nine to three. Any comparisons between frequency distributions that are *both* missing the two rightmost columns should be based on three degrees of freedom, otherwise the χ^2 test might fail to identify correlation within the data. This solution is included in the MARLEDA method (described in the next chapter), making it possible to use χ^2 to construct an accurate MRF model despite sparse sampling.

2.6 Conclusion

Statistical models are effective representations for structure within complex optimization domains. State-of-the-art EDAs employ flexible models able to represent many different types of domain structure. When matched with effective learning techniques, statistical models become a foundation for powerful search algorithms. In the next chapter, a combination of Markov random

fields and statistical hypothesis testing is shaped into the MARLEDA method.

Chapter 3

The MARLEDA Method

The MARLEDA search algorithm is designed to overcome the limitations of previous EDAs. By using a more general and flexible model, learned in an efficient way, MARLEDA has the potential to be a more effective search method. MARLEDA still follows the procedural framework for EDAs outlined in section 2.3. The following sections detail the major components of MARLEDA, with parameters of the MARLEDA algorithm shown in **high-lighted** font, and analyze the computational complexity of the system.

3.1 Selection

Each chromosome in MARLEDA’s population is composed of a set of genes, each corresponding to a parameter of the combinatorial optimization task. The number of genes is the same for all chromosomes and fixed at **Genes**. All concrete statistics regarding genes are calculated using members of the current population, $\mathcal{P}(t)$, where $|\mathcal{P}(t)| = \mathbf{PopSize}$. To bias the statistics toward favorable population members, a subcollection $\mathcal{P}'(t) \subseteq \mathcal{P}(t)$ of the current population is chosen via tournament selection [18]. The top **Parents** · **PopSize** (where **Parents** $\in (0, 1]$) high-fitness chromosomes compete in tournaments of

size **TournSize**. A total of **PopSize** chromosomes are selected for membership in $\mathcal{P}'(t)$.

3.2 The MRF Model

MARLEDA uses a set of nominal random variables, $\{X_1, \dots, X_n\}$, to model the nominal genes, $\{x_1, \dots, x_n\}$, of a combinatorial optimization task. Statistical dependencies among the random variables, and therefore among the genes, are recorded in a neighborhood system, thus forming an MRF model.

The neighbor relation between any two random variables is grounded in an observable statistical dependence within the members of $\mathcal{P}'(t)$. Like many EDAs, MARLEDA tests for these dependencies to learn its model. Consider a “partial” MRF whose neighborhood system does not yet fully capture all the observable dependencies. Let X_i and X_j be non-neighbors in the current neighborhood system, each with their own “partial” set of neighbors. If a dependence between X_i and X_j is observable, the neighborhood system should be updated to make X_i and X_j neighbors. Conversely, if X_i and X_j began as neighbors and a dependence is not observable, they should become non-neighbors.

Pearson’s χ^2 test is used to compute the confidence level of dependence between two genes. The two frequency distributions compared are

$$F_{\text{obs}} = F(x_i, x_j | x_k, k \in \partial(i)), \quad \text{and} \quad (3.1)$$

$$F_{\text{exp}} = \frac{F(x_i | x_k, k \in \partial(i)) \cdot F(x_j | x_k, k \in \partial(i))}{|F(x_k, k \in \partial(i))|}, \quad (3.2)$$

where F_{obs} is the joint frequency distribution of x_i and x_j , given x_i 's neighbors, as observed within $\mathcal{P}'(t)$. F_{exp} is the joint frequency distribution of x_i and x_j , given x_i 's neighbors, under the assumption that x_i and x_j are independent, i.e. the product of the marginal FDs of x_i and x_j as observed within $\mathcal{P}'(t)$. (Note: when using binary chromosomes χ^2 is adjusted using Yates' correction [69].)

Intuitively, the above procedure measures how much information is gained by making x_i and x_j neighbors. If F_{obs} and F_{exp} differ, x_i depends on x_j and the two should be made neighbors. Similarly, if x_i and x_j began as neighbors, the gain in remaining neighbors can be computed by temporarily removing their neighbor status and performing the same test. (Note: although the MRF neighbor relation is symmetrical, F_{obs} and F_{exp} are not symmetrical about x_i and x_j . Ideally, the reciprocal test should also be performed, with only two successes or two failures suggesting a change in neighbor status. A single test is performed in MARLEDA for simplicity, and it works well in practice.)

MARLEDA constructs the MRF neighborhood system via a greedy search approach starting from a trivial neighborhood system, $\partial(i) = \emptyset$. At each iteration **ModelAdds** pairs of non-neighbor genes are tested. If the confidence level of the pair is at least **ModelAddThresh**, the model is updated to make the pair neighbors. Similarly, **ModelSubs** pairs of neighbors are tested, and if the confidence level is below **ModelSubThresh** the pair is made non-neighbors. The order of all tests is randomized.

The two threshold values, **ModelAddThresh** and **ModelSubThresh**, represent how strict of the neighbor relation is within the MRF neighborhood system. While statistical hypothesis tests are typically used with very strict confidence levels, 0.95, 0.99, or higher, in MARLEDA it is possible to use more relaxed confidence levels, since even partial correlations in the data are beneficial to the sampling process. During preliminary experimentation it was determined that **ModelAddThresh** = 0.8 and **ModelSubThresh** = 0.6 work well across a spectrum of optimization tasks.

As mentioned in section 2.5, a degrees of freedom term, δ , must be computed for each χ^2 test between a pair of genes x_i and x_j . Let \mathbf{A} be the set of alleles possible for each (and all) genes. Derivation 2.8 in section 2.5 defines the degrees of freedom parameter as

$$\delta(i) = (|\mathbf{A}| - 1)^{|\partial(i)|+2}. \quad (3.3)$$

However, the calculation is adjusted in two situations.

First, when one or more alleles for a gene are not represented in $\mathcal{P}'(t)$, that gene no longer contributes a full $|\mathbf{A}| - 1$ degrees to the above calculation. The actual number of alleles represented for each gene and adjusted degrees of freedom are

$$\mathbf{A}(i) = \{a \in \mathbf{A} : \exists \mathbf{x} \in \mathcal{P}'(t) : x_i = a\}, \quad (3.4)$$

$$\delta(i, j) = \prod_{k \in \{i, j\} \cup \partial(i)} \max(|\mathbf{A}(k)| - 1, 1). \quad (3.5)$$

Second, the degrees of freedom term naturally grows exponentially in the size of the neighborhood. The minimum χ^2 value necessary to demonstrate dependence grows approximately exponentially as well. However, the candidate solution space is generally poorly represented by the chromosomes in $\mathcal{P}'(t)$, i.e. $|\mathcal{P}'(t)| \ll |\mathbf{X}|$. There are a finite number of samples in $\mathcal{P}'(t)$ from which F_{obs} , F_{exp} , and consequently χ^2 , are computed. This constraint places an upper limit on the computable χ^2 value, thus truncating its exponential growth and making it increasingly difficult to expand the neighborhood system. This restriction provides a natural limit to the growth of the neighborhood system, making it easier to avoid ungrounded correlations.

However, when $|\mathbf{A}| = 2$ the degrees of freedom calculation trivially collapses to one. Instead of exponential growth in the size of a gene’s neighborhood, there is no growth. The χ^2 term, however, continues to grow, and this mismatch between degrees of freedom and χ^2 makes it too easy to “pass” Pearson’s χ^2 test. Consequently, neighborhoods can expand and become maximal without true statistical support. To combat this problem, when $|\mathbf{A}| = 2$ the computed degrees of freedom is artificially inflated to restore exponential growth as follows

$$\delta(i, j) = \left\lceil \prod_{k \in \{j\} \cup \partial(i)} \max(|\mathbf{A}(k)| - 0.25, 1) \right\rceil. \quad (3.6)$$

All genes but one, x_i , contribute at most 1.75 “virtual” degrees to the calculation. Values ranging from 1.5 to 1.75 worked well in preliminary experiments.

Pearson’s χ^2 test provides a convenient method for learning the local neighbor relations of a Markov random field. Though this greedy construction procedure is not guaranteed to build an optimal MRF, it captures the high-fitness features of the population. Constructing the model is only the first step; the model must then be sampled to combine the high-fitness features into new chromosomes.

3.3 Generating Chromosomes

New chromosomes are created in MARLEDA by sampling the MRF model. Sampling is performed via a Markov chain Monte Carlo process:

-
1. $\mathbf{x}^{\text{new}} \leftarrow$ a random chromosome from $\mathcal{P}'(t)$.
 2. Randomly select a gene x_i^{new} .
 3. Compute $P(x_i|x_k, k \in \partial(i))$.
 4. $x_i^{\text{new}} \leftarrow$ sample from $P(x_i|x_k, k \in \partial(i))$.
 5. Unless termination criteria are met, return to step 2.
-

Ideally, the sampling process continues until the allele distribution of the new chromosome stabilizes. The number of iterations needed before convergence depends on the specifics of the joint probability distribution encoded by the MRF and thus may not be known a priori. However, a good rule of thumb is to allow the sampler to “burn-in” for at least several thousand iterations. In MARLEDA, the sampling process is truncated after **Montelters** iterations. After termination, genes are mutated with probability **Mutation**.

The complete random field on the configuration space \mathbf{X} is not available, hence $P(x_i|x_k, k \in \partial(i))$ is sometimes undefined. In such cases, a “relaxed” conditional probability is used. In effect, undefined regions of the configuration space are approximated by nearby well-defined regions. Under normal conditions,

$$P(x_i|x_k, k \in \partial(i)) = \frac{F(x_i|x_k, k \in \partial(i))}{|F(x_k, k \in \partial(i))|}. \quad (3.7)$$

When $F(x_k, k \in \partial(i))$ contains no samples, a first-order relaxation is calculated, incorporating all subsets of $\partial(i)$ of size $|\partial(i)| - 1$:

$$P^{(1)}(x_i|x_k, k \in \partial(i)) = \frac{\bigcup_{\partial'(i) \subset \partial(i)} F(x_i|x_k, k \in \partial'(i), |\partial'(i)| = |\partial(i)| - 1)}{\sum_{\partial'(i) \subset \partial(i)} |F(x_k, k \in \partial'(i), |\partial'(i)| = |\partial(i)| - 1)|}. \quad (3.8)$$

If the first-order relaxation is also undefined, the second-order relaxation incorporating all subsets of $\partial(i)$ of size $|\partial(i)| - 2$ is evaluated, and so on, until a valid probability distribution is found. In the worst case, the entire neighborhood $\partial(i)$ is ignored:

$$P^{(|\partial(i)|)}(x_i|x_k, k \in \partial(i)) = P(x_i), \quad (3.9)$$

and the current sampling iteration degenerates to sampling from the marginal distribution of x_i , as univariate EDAs do.

3.4 Replacement

The **Replaced · PopSize** (where **Replaced** $\in (0, 1]$) chromosomes in the population with the lowest fitness are replaced by newly created chromosomes.

This step implements an elitist strategy by which the top $(1 - \mathbf{Replaced}) \cdot \mathbf{PopSize}$ chromosomes are preserved between iterations.

3.5 Computational Complexity

MARLEDA’s computational complexity is dominated by the MRF neighborhood learning procedure and the MRF sampler. MRF neighborhood learning primarily involves constructing the frequency distributions used with Pearson’s χ^2 test, i.e. equations (3.1) and (3.2). In the current implementation of MARLEDA, these frequency distributions are constructed by sorting and then traversing the population. Using a standard comparison-based sorting algorithm, each comparison between two chromosomes takes time proportional to the size of the MRF neighborhood involved, $|\partial(i)|$. In the worst case, $|\partial(i)| = \mathbf{Genes} - 1$. A single iteration of the learning procedure is dominated by the sorting step, leading to a computational complexity of $O(\mathbf{Genes} \cdot \mathbf{PopSize} \log \mathbf{PopSize})$. All iterations together have a complexity of $O((\mathbf{ModelAdds} + \mathbf{ModelSubs})\mathbf{Genes} \cdot \mathbf{PopSize} \log \mathbf{PopSize})$.

The MRF sampler involves repeated calculation of the conditional probability in equation (3.7) and occasionally equation (3.8). At each iteration of the sampler, both calculations involve traversing the population to identify those chromosomes contributing to the appropriate frequency distributions, leading to a computational complexity of $O(\mathbf{Genes} \cdot \mathbf{PopSize})$. All iterations together have a complexity of $O(\mathbf{Montelters} \cdot \mathbf{Genes} \cdot \mathbf{PopSize})$. However, in the current implementation of MARLEDA each unique conditional probability is

calculated only once and cached, thus the MRF sampler’s performance can be better in practice than the worst case scenario. In the benchmark experiments performed in the next section, the caching scheme produced 4–10× speedup.

Together, the MRF neighborhood learning procedure and the MRF sampler give MARLEDA a computational complexity of $O(((\mathbf{ModelAdds} + \mathbf{ModelSubs}) \log \mathbf{PopSize} + \mathbf{Montelters}) \mathbf{Genes} \cdot \mathbf{PopSize})$. The complexity is linear in the major algorithm parameters, except $\mathbf{PopSize}$, and therefore quite practical. The use of more sophisticated data structures (particularly regarding storage of frequency and probability distributions) could improve the computational complexity of MARLEDA further. For instance, the logarithmic $\mathbf{PopSize}$ factor in the MRF neighborhood learning procedure could be removed by replacing the comparison-based sort with a hash.

3.6 Conclusion

The processes described in this chapter are the essential components of the MARLEDA algorithm. MARLEDA is a combination of classic probabilistic search techniques, such as tournament selection and mutation, and advanced statistical methods. MARLEDA’s efficient use of a Markov random field model should enable it to search complex domains more effectively than other probabilistic methods. In the next chapter, MARLEDA’s performance is tested on several combinatorial optimization tasks and found to perform well compared to the standard GA and an advanced EDA.

Chapter 4

Benchmark Experiments

In this chapter MARLEDA’s performance is tested on five combinatorial optimization tasks. The first three tasks, OneMax, deceptive trap functions, and the Rosenbrock function, are standard artificial benchmark tasks for optimization algorithms. The fourth task, optimization of Ising spin glass systems, is a difficult optimization task from statistical physics. The fifth task, optimization of three-dimensional lattice proteins, is a simplified form of the protein folding problem from computational biology. MARLEDA’s performance is compared against two optimization suites with publicly available source code: the GENESYs [2] implementation of a standard GA, to provide an expected performance baseline, and the Bayesian optimization algorithm (BOA) [53] with decision graphs, to provide a comparison with a state-of-the-art search method.

4.1 Experimental Design

In addition to the GENESYs, BOA, and MARLEDA algorithms, two variants of MARLEDA and one variant of BOA were evaluated when appropriate to the task. The first MARLEDA variant, MARLEDA^{-mutation},

disables mutation. Most EDAs lack a traditional mutation operator, thus MARLEDA^{-mutation}'s performance illuminates mutation's contribution to MARLEDA and demonstrates how useful it is for EDA methods. Similarly, the BOA^{+mutation} variant adds mutation to standard BOA to test the same effect. The last variant, MARLEDA^{+model}, disables MRF learning and uses a fixed MRF neighborhood system based on the known domain structure of the optimization task. This variant evaluates MARLEDA's ability to successfully exploit a hand-crafted model designed from human knowledge of the task.

In order to gauge each algorithm's search capability fairly, all algorithms were limited to the same fixed number of fitness function evaluations during each experiment. So that each algorithm could best use this limited resource, each algorithm's parameters were tuned to optimize final solution quality (exception: MARLEDA's **ModelAddThresh** and **ModelSubThresh** parameters were fixed at 0.8 and 0.6, respectively). Tuning proceeded via simple hill climbing: Beginning with reasonable or documented parameter settings, slightly perturbed settings were evaluated, continuing until no further improvement in solution quality was achieved. In cases where multiple parameter settings resulted in equivalent solution quality, preference was given to those producing more rapid progress. While this procedure does not guarantee that the resulting parameter settings are optimal, they reliably lead to good performance.

4.2 OneMax

The OneMax problem is a simple optimization task for binary strings. The goal is to maximize the the number of “on” bits, i.e. maximize

$$f(\mathbf{x}) = \sum_{i=1}^n x_i.$$

There are no dependencies among genes for this task, thus OneMax is not a particularly interesting problem for EDAs. It is included only to provide a performance baseline for the next optimization task, deceptive trap functions.

In this experiment, binary strings of 300 bits were optimized, with each algorithm limited to 10,000 fitness function evaluations. The following algorithm parameters were used:

GENEsYs : population size = 200, full population selection, uniform crossover, elitism, Whitley rank selection with $\alpha = 2.0$, mutation rate = 0.004, crossover rate = 1.0, and generation gap = 1.0.

BOA : population size = 200, offspring percentage = 10, tournament size = 1, and max incoming links = 1.

MARLEDA : **PopSize** = 150, **Parents** = 0.6, **TournSize** = 3, **ModelAdds** = 3000, **ModelSubs** = 2000, **Montelters** = 1500, **Mutation** = 0.01, and **Replaced** = 0.1.

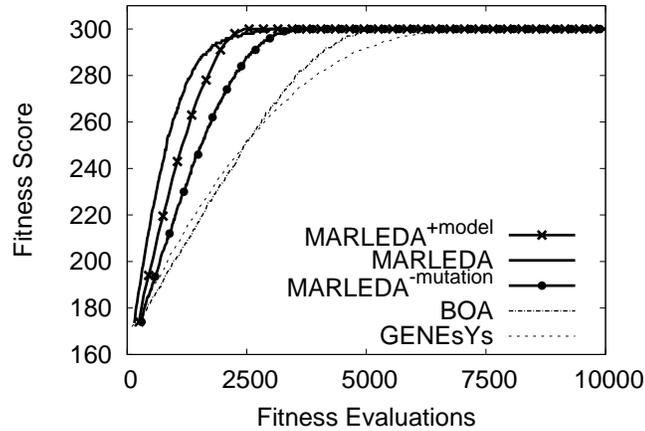


Figure 4.1: Learning curves of the median best population member in 100 independent trials for an instance of OneMax with 300 bits. All algorithms easily find the optimum solution, with MARLEDA demonstrating a moderate advantage in learning time.

MARLEDA^{-mutation} : **PopSize** = 300, **Parents** = 0.65, **TournSize** = 2, **ModelAdds** = 3000, **ModelSubs** = 2500, **Montelters** = 1500, and **Replaced** = 0.05.

MARLEDA^{+model} : **PopSize** = 250, **Parents** = 0.3, **TournSize** = 3, **Montelters** = 750, **Mutation** = 0.0, and **Replaced** = 0.4.

Figure 4.1 shows the median best fitness score during the course of evolution over 100 independent trials of each algorithm. While all algorithms easily found the optimal OneMax solution, the effect of mutation on the three instances of MARLEDA is interesting to note. The parameter tuning process discovered that standard MARLEDA performed best with a small amount of mutation. That is, MARLEDA^{-mutation}'s rate of progress is slightly worse than

that of standard MARLEDA. Interestingly, when MARLEDA was provided with the true univariate model (i.e. no dependencies among genes) of the domain (i.e. MARLEDA^{+model}), it performed best without mutation.

MARLEDA^{+model} was able to find the optimum solution with several hundred fewer evaluations than MARLEDA. However, the rate of progress was initially worse than that of MARLEDA. The coincidental dependencies between bits that standard MARLEDA identifies initially boost its performance but then hinder it as “off” bits are mistakenly preserved. Without an ideal model, mutation contributes to MARLEDA’s performance on this, albeit simple, task.

4.3 Deceptive Trap Functions

Deceptive trap functions [12] are multimodal functions designed such that local gradient information will tend to lead optimization algorithms toward local optima and away from global optima. Search algorithms must therefore have the capacity to escape local optima in order to identify global optima. For EDAs, this means learning the deceptive elements in order to avoid local “traps.”

A standard class of trap function is a variant of OneMax where blocks of bits have two local optima, only one of which contributes to the global

optimum of the function. Let α be the block size of the trap. Then

$$\begin{aligned}
 u(\mathbf{x}, k) &= \sum_{i=\alpha(k-1)+1}^{\alpha k} x_i, \\
 f_\alpha(\mathbf{x}, k) &= \begin{cases} \alpha - u(\mathbf{x}, k) - 1 & \text{if } u(\mathbf{x}, k) < \alpha \\ \alpha & \text{if } u(\mathbf{x}, k) = \alpha, \end{cases} \quad \text{and} \\
 f(\mathbf{x}) &= \sum_{i=1}^{\frac{n}{\alpha}} f_\alpha(\mathbf{x}, i).
 \end{aligned}$$

Within each block of α bits, only one of the 2^α possible bit combinations is part of the global optimum. All other bit combinations guide search toward local trap optima. As the trap size increases the components of the global optimum become more rare and thus more difficult for search to discover. For small trap sizes this task presents an interesting challenge for optimizers, which must weigh the abundant evidence of trap optima against the scarce evidence of the global optimum.

In this experiment, binary strings of 300 bits were optimized for traps of three bits and five bits. Each algorithm was limited to 20,000 fitness function evaluations. For traps of three bits the following algorithm parameters were used:

GENEsYs : population size = 250, full population selection, uniform crossover, elitism, Whitley rank selection with $\alpha = 2.0$, mutation rate = 0.0, crossover rate = 1.0, and generation gap = 1.0.

BOA : population size = 250, offspring percentage = 10, tournament size = 1, and max incoming links = 1.

MARLEDA : **PopSize** = 450, **Parents** = 0.85, **TournSize** = 3, **ModelAdds** = 3000, **ModelSubs** = 2000, **Montelters** = 1500, **Mutation** = 0.0, and **Replaced** = 0.15.

MARLEDA^{+model} : **PopSize** = 400, **Parents** = 0.65, **TournSize** = 4, **Montelters** = 1500, **Mutation** = 0.01, and **Replaced** = 0.6.

For traps of five bits the following algorithm parameters were used:

GENEsYs : population size = 100, full population selection, uniform crossover, elitism, Whitley rank selection with $\alpha = 2.0$, mutation rate = 0.005, crossover rate = 0.5, and generation gap = 1.0.

BOA : population size = 600, offspring percentage = 10, tournament size = 5, and max incoming links = 3.

MARLEDA : **PopSize** = 300, **Parents** = 0.1, **TournSize** = 3, **ModelAdds** = 3000, **ModelSubs** = 2000, **Montelters** = 1500, **Mutation** = 0.015, and **Replaced** = 0.65.

MARLEDA^{+model} : **PopSize** = 1400, **Parents** = 0.7, **TournSize** = 3, **Montelters** = 2500, **Mutation** = 0.005, and **Replaced** = 0.5.

Figure 4.2 shows the median best fitness score during the course of evolution over 100 independent trials of each algorithm. In the case of traps of three bits, all algorithms routinely discovered the optimum solution, though the rate of progress was notably slower than in OneMax.

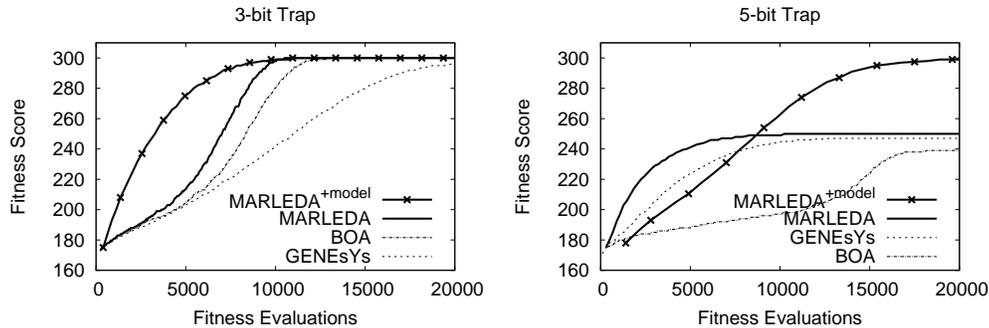


Figure 4.2: Learning curves of the median best population member in 100 independent trials for 300-bit instances of a deceptive trap function. The differences in median best fitness at the end of evolution are statistically significant (as computed by the Wilcoxon rank-sum test with a confidence greater than 99%) in the 5-bit trap domain for all algorithm pairs except GENEsYs/MARLEDA. In the 3-bit trap scenario, the modeling capabilities of the EDAs allow them to outperform the standard GA. In the 5-bit trap scenario, the EDAs only achieved marginal solution gains over the GA. However, when provided an accurate model of the domain, MARLEDA^{+model} readily found the optimal solution in the majority of trials. As domain complexity increases and the potential to become trapped in local optima rises, utilizing an accurate model is critically important.

For traps of five bits, all algorithms ignorant of the true structure of the domain were unable to find the globally optimal solution. The algorithms instead identified trap optima, where each block of five bits has converged to its second local optimum. Resulting fitness scores are at least $300 \cdot \frac{4}{5} = 240$, because a few of the blocks have been correctly optimized by chance. However, when MARLEDA was provided the true structure of the domain (MARLEDA^{+model}), where each bit is dependent on the four other bits in its block, avoiding the trap optima and identifying the global optimum ceased to be a problem.

The comparatively slow improvement of MARLEDA^{+model} is the result of the larger population used in that experiment. With a larger population, more fitness function evaluations were performed each generation, thus making it appear that progress was slow. However, the larger population was necessary to solve the 5-bit trap problem since more samples (chromosomes) were needed to accurately cover each trap block. This observation is consistent with previous work where the necessary population size was shown to increase exponentially with the size of the trap [19, 25]. Consequently, the ignorant algorithms would be likely to find the optimum solution if permitted additional fitness function evaluations by an order of magnitude or more. Without such evaluations, the parameter tuning process discovered that those algorithms with mutation operators (GENEsYs and MARLEDA) were best served by maximizing the total number of generations of evolution, by minimizing population size, and relying on mutation for search. This process resulted in some unintuitive parameter shifts, such as the *decrease* in population size for MARLEDA from 450 on traps of three bits to 300 on traps of five bits.

When provided sufficient fitness evaluations, the EDAs learned and exploited the domain structure to increase search efficiency. However, when evaluations were limited, as was forced upon MARLEDA and BOA in the 5-bit trap experiment, MARLEDA’s performance degraded the most gracefully. As in the OneMax task, the parameter tuning process discovered that mutation contributed to MARLEDA’s search capability. However, unlike the

OneMax domain mutation remained useful even when the true domain structure was known. The results of these and the OneMax experiments suggest that mutation is a useful operation in EDA methods.

4.4 The Rosenbrock Function

The Rosenbrock function [59] is a numerical function definable for any number of dimensions. In two dimensions, the goal is to minimize the function

$$f(x, y) = (1 - x)^2 + 100 (y - x^2)^2 .$$

The function has a global minimum at $f(1, 1) = 0$, but when x and y are encoded in binary, the resulting discretization of the domain produces many local minima near the curve $y = x^2$. In addition, there are many overlapping low-order dependencies among the bits of x and y . Since this domain is much less deceptive than trap functions, EDAs should be able to exploit the domain structure to perform search efficiently and distinguish themselves from GAs.

The parameters x and y are encoded in binary chromosomes as fixed-point numbers in the range $[0, 4]$ whose values are then translated to the target domain $[-2, 2]$. The experiments include chromosomes of 32 bits (16 bits each for x and y) and 64 bits (32 bits each for x and y). Each run of the experimental algorithms was limited to 20,000 fitness function evaluations in the 32-bit Rosenbrock experiments and 30,000 fitness function evaluations in the 64-bit Rosenbrock experiments. The following algorithm parameters were used with the 32-bit Rosenbrock experiment:

GENEsYs : population size = 200, full population selection, uniform crossover, elitism, Whitley rank selection with $\alpha = 1.9$, mutation rate = 0.007, crossover rate = 0.9, and generation gap = 1.0.

BOA : population size = 800, offspring percentage = 80, tournament size = 2, and max incoming links = 10.

BOA^{+mutation} : population size = 800, offspring percentage = 75, tournament size = 3, max incoming links = 10, mutation rate = 0.01.

MARLEDA : **PopSize** = 400, **Parents** = 0.85, **TournSize** = 4, **ModelAdds** = 3000, **ModelSubs** = 2000, **Montelters** = 1000, **Mutation** = 0.0, and **Replaced** = 0.7.

The following algorithm parameters were used with the 64-bit Rosenbrock experiment:

GENEsYs : population size = 220, full population selection, uniform crossover, elitism, Whitley rank selection with $\alpha = 1.9$, mutation rate = 0.005, crossover rate = 1.0, and generation gap = 1.0.

BOA : population size = 650, offspring percentage = 70, tournament size = 2, and max incoming links = 12.

BOA^{+mutation} : population size = 350, offspring percentage = 10, tournament size = 2, max incoming links = 10, mutation rate = 0.02.

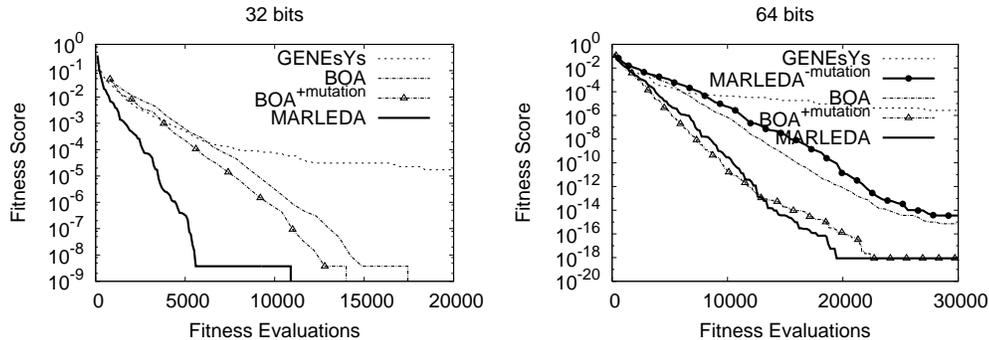


Figure 4.3: Learning curves of the median best population member in 100 independent trials for Rosenbrock instances of 32 and 64 bits. The differences in median best fitness at the end of evolution are statistically significant (as computed by the Wilcoxon rank-sum test with a confidence greater than 99%) in the 32-bit domain for GENE_sYs/BOA and GENE_sYs/MARLEDA. Differences are statistically significant in the 64-bit domain for all algorithm combinations except MARLEDA^{-mutation}/BOA. In this domain of relatively low deception, the EDAs were able to learn and exploit domain structure, allowing them to find solutions vastly superior to more localized GA search.

MARLEDA : **PopSize** = 450, **Parents** = 0.8, **TournSize** = 4, **ModelAdds** = 3000, **ModelSubs** = 2000, **Montelters** = 1000, **Mutation** = 0.03, and **Replaced** = 0.8.

MARLEDA^{-mutation} : **PopSize** = 700, **Parents** = 0.6, **TournSize** = 3, **ModelAdds** = 3000, **ModelAddThresh** = 0.8, **ModelSubs** = 2000, **ModelSubThresh** = 0.6, **Montelters** = 1000, and **Replaced** = 0.8.

Figure 4.3 shows the median best fitness score found by each algorithm over 100 independent trials. Both MARLEDA and BOA performed well on this task, with MARLEDA demonstrating a distinct advantage in both learning rate and final quality. In the 32-bit domain, MARLEDA’s and BOA’s

median best chromosome quickly approached the second lowest fitness score possible, represented by the flat regions of the fitness curves near 15,000 evaluations. This score corresponds to the most deceptive of the local minima in the domain, with a Hamming distance of 29 bits from the global minimum. The vertical segment of the fitness curves shows the point where the median best chromosome was the global optimum. Due to the logarithmic scale of the graphs, the learning curves appear to “fall off” the graph.

BOA^{+mutation} benefited slightly from mutation. Though both MARLEDA and BOA exploited the domain structure to dramatically outperform GENEYSs, the introduction of random noise to BOA’s search helped narrow the gap between BOA and MARLEDA by helping BOA^{+mutation} escape local optima during its search. However, the parameter turning process discovered that mutation was unnecessary for MARLEDA (that is, MARLEDA is equivalent to MARLEDA^{-mutation} in this particular case), demonstrating MARLEDA’s superior learning capabilities in this domain.

The 64-bit domain shows even greater separation between MARLEDA, BOA, and GENEYSs. The encoding of the two numerical coordinates presents a significant hurdle for local search methods such as GENEYSs. While the fitness landscape of the Rosenbrock function is smooth in numerical space, it is quite rough in configuration space: A small change in numerical space may result in a large change in configuration space, and vice versa. The structure-exploiting approach of the EDAs allowed them to find better solutions, with MARLEDA performing significantly better than BOA. Its ability to exploit

the structure of the configuration space to correctly optimize many bits at once was crucial to good performance.

It is interesting to note that mutation results in more efficient search. MARLEDA and BOA^{+mutation}, which both utilize mutation, discovered better solutions than MARLEDA^{-mutation} and BOA, which have no mutation operator. The small search gain provided by mutation in the 32-bit domain compared to the relatively large gain in the 64-bit domain suggests that there is a domain complexity threshold beyond which some component of the EDAs interferes with search. Mutation helps compensate for such deficiencies.

The structure of the Rosenbrock domain cannot be described as concisely as that of OneMax or deceptive trap functions. In fact, the structure varies across the domain. For this reason, no experiments involving MARLEDA^{+model} were performed.

4.5 Ising Spin Glasses

Ising spin glasses [33] are a model of magnetic materials developed in statistical physics; they have been extensively used as EDA benchmarks. A set of *spins*, $\{s_1, \dots, s_n\}$, exist in one of two states, +1 or -1, and each spin is coupled to a set of neighboring spins. An instance of an Ising spin glass system is defined by a set of coupling constants, $\{J_{i,j} : i, j \in \{1, \dots, n\}\}$, that encapsulate the neighbor relationships. A coupling constant $J_{i,j}$ is non-zero if s_i and s_j are neighbors. In these experiments, coupling constants are restricted to values of +1, 0, and -1. The goal is to find a set of spin states

that minimizes the Hamiltonian of the system:

$$H = - \sum_{i,j=1}^n J_{i,j} s_i s_j.$$

Minimizing the Hamiltonian implies that neighboring spins should tend to exist in the same state if their coupling constant is $+1$ and in differing states if their coupling constant is -1 . However, conflicting coupling constants among groups of spins prevent this rule from being applied fully. Consequently, groups of spins may have locally optimal states that are quite different from the globally optimal spin states, or *ground states*. This property makes spin glass systems a difficult partially deceptive search task.

To test the scalability of MARLEDA, the Ising spin glass systems used in these experiments contain the most parameters of any optimization experiment presented in this chapter. Though there is plenty of domain structure for EDAs to exploit, the volume of information will likely make learning an effective model difficult. Five hundred instances of Ising spin glass systems with 400 spins and 900 spins were randomly generated. Each instance was arranged in a two-dimensional square lattice (20×20 or 30×30) with periodic boundary conditions. Each spin was neighbored by the adjacent spin above, below, to the left, and to the right. The coupling constants for neighboring spins were uniformly sampled from $\{+1, -1\}$, with all other coupling constants set to 0, thus each instance was drawn from the region of spin glass system space known to contain a comparatively high density of difficult instances [43, 44]. The set of spin states was encoded in a binary chromosome with one bit per

spin state.

Each run of the experimental algorithms was limited to 20,000 fitness function evaluations in the 400 spin domain and 60,000 fitness function evaluations in the 900 spin domain. The following algorithm parameters were used in all trials of 400 spins:

GENEsYs : population size = 100, full population selection, uniform crossover, elitism, Whitley rank selection with $\alpha = 2.0$, mutation rate = 0.006, crossover rate = 1.0, and generation gap = 1.0.

BOA : population size = 500, offspring percentage = 13, tournament size = 3, and max incoming links = 2.

MARLEDA : **PopSize** = 400, **Parents** = 0.85, **TournSize** = 4, **ModelAdds** = 3000, **ModelSubs** = 2000, **Montelters** = 1000, **Mutation** = 0.0, and **Replaced** = 0.7.

MARLEDA^{+model} : **PopSize** = 900, **Parents** = 0.75, **TournSize** = 2, **Montelters** = 2400, **Mutation** = 0.005, and **Replaced** = 0.25.

The following algorithm parameters were used in all trials of 900 spins:

GENEsYs : population size = 130, full population selection, uniform crossover, elitism, Whitley rank selection with $\alpha = 2.0$, mutation rate = 0.0025, crossover rate = 1.0, and generation gap = 1.0.

BOA : population size = 700, offspring percentage = 75, tournament size = 3, and max incoming links = 6.

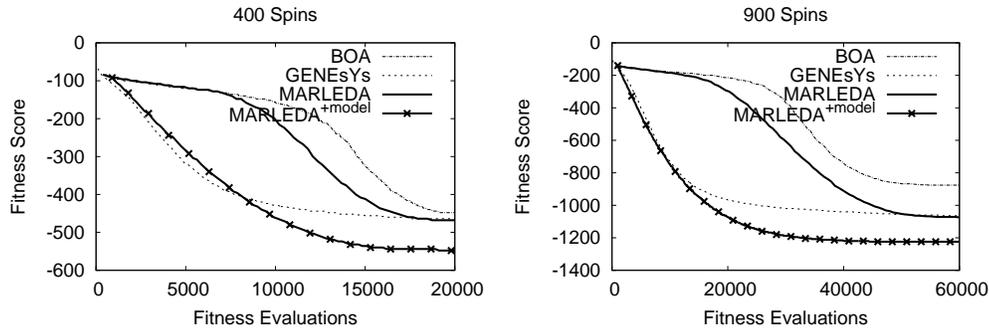


Figure 4.4: Representative learning curves of the median best fitness in 100 independent trials for an instance of an Ising spin glass system of 400 spins and 900 spins. The differences in median best fitnesses at the end of evolution are statistically significant (as computed by the Wilcoxon rank-sum test, with a confidence greater than 99%) for all algorithm combination except GENESys/MARLEDA in both the 400 spin and 900 spin domains. MARLEDA’s Markov random field model naturally represents the relationships in the spin glass domain, resulting in improved performance compared to BOA. In addition, when provided an accurate model of the domain, MARLEDA^{+model} was able to efficiently discover optimal solutions for this difficult optimization task.

MARLEDA : **PopSize** = 700, **Parents** = 0.75, **TournSize** = 3, **ModelAdds** = 3500, **ModelSubs** = 2000, **Montelters** = 1500, **Mutation** = 0.0, and **Replaced** = 0.9.

MARLEDA^{+model} : **PopSize** = 1000, **Parents** = 0.95, **TournSize** = 3, **Montelters** = 2400, **Mutation** = 0.004, and **Replaced** = 0.1.

Each algorithm was run 100 times on each of the 1000 randomly generated spin glass instances. Figure 4.4 shows the median best fitness score over the 100 trials found by each algorithm on *one* particular instance. Nearly all instances resulted in similar learning curves, thus figure 4.4 is representative.

All algorithms ignorant of the true domain structure discovered solutions of nearly the same quality, with the exception of BOA on the system of 900 spins. Unlike the Rosenbrock function, two-dimensional spin glass systems are amenable to local search techniques, shown by GENE_sYs’ good performance. However, local search does not lead to global optima. The optimal fitness score for each spin glass instance was determined using the Spin Glass Ground State Server at the University of Köln [37]. The solutions routinely discovered by MARLEDA and GENE_sYs have fitness scores only 80%–85% of optimal. The deceptive qualities of this domain were not completely overcome by the EDAs’ statistical models.

The EDAs exhibit a curiously slow start, which is caused by poor initial models. The complexity of the domain coupled with the relatively large number of parameters make it difficult for the EDAs to identify dependencies among parameters. The learned models therefore did not promote high-fitness chromosomes during sampling and tended to reproduce low-fitness aspects of the population. However, once the models were sufficiently refined solution quality improved rapidly.

In contrast to BOA and standard MARLEDA, MARLEDA^{+model} performed very well. The lattice structure of the spin glass systems forms a natural MRF neighborhood system. When provided with this system, MARLEDA^{+model} was able to routinely discover the ground state of systems of 400 spins and come to within 1%-2% of the ground state of systems of 900 spins. Though these experiments are constructed differently, the performance

results are consistent with the experiments of [64]; exploiting the structure of spin glass systems is the key to solving them efficiently.

This is an ideal example of injecting human knowledge into the search method and reaping major rewards. The structure of the Ising spin glass domain, while a part of the fitness function, is not directly accessible to the search algorithms. When this structure information was made accessible via an MRF model, MARLEDA^{+model} successfully scaled-up to this large optimization tasks. This result also suggests two parallel lines of improvement for EDAs: (1) the development of models that can easily accommodate human knowledge, and (2) stronger model learning procedures for domains lacking known models.

4.6 Lattice Proteins

Lattice proteins are a theoretical simplification of biological proteins used to explore the basic principles that govern protein folding [39]. Like biological proteins, lattice proteins are chains of residues folded back upon themselves to produce unique structures. However, lattice protein residues are extremely simple and exist within a spatial world of two or three-dimensional lattices (hence the name).

The forces that govern biological protein folding are numerous and complex, whereas lattice proteins are a far simpler first approximation to that domain. Lattice protein residues abstract the complexity of biological residues into a few basic properties. The most common form of lattice proteins ad-

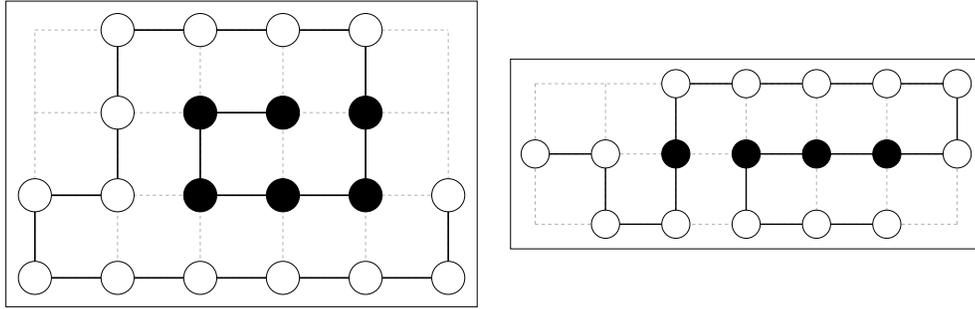


Figure 4.5: Two examples of lattice proteins embedding in a two-dimensional square lattice. The hydrophobic effect causes hydrophobic residues (black) to cluster together, surrounded by hydrophilic (white) residues.

heres to the hydrophobic-hydrophilic (HP) model [13] in which every residue is either hydrophobic (non-polar) or hydrophilic (polar). Consequently, lattice protein folding is guided by the *hydrophobic effect*, the tendency of hydrophobic molecules to form aggregates within an aqueous medium due to the polar nature of water molecules. The resulting lattice protein shapes, or *conformations*, organize hydrophobic residues near the center of the protein and hydrophilic residues near the periphery; a hydrophilic “shell” surrounding a hydrophobic “core” (figure 4.5).

The hydrophobic effect is formalized in an energy function that measures how tightly hydrophobic residues are packed together. Let \mathbf{s} be the sequence of hydrophobic (H) and hydrophilic (P) residues of a lattice protein and let s_i be the i^{th} residue in the sequence. Then

$$\mathbf{s} = (s_1, \dots, s_n : s_i \in \{\text{H}, \text{P}\}).$$

Let v_i be the position vector for the i^{th} residue of the lattice protein and let \mathbf{v}

be a conformation of the lattice protein, i.e. $\mathbf{v} = (v_1, \dots, v_n)$ and $|v_i - v_{i+1}| = 1$.

Then

$$\begin{aligned}
 h(s_i) &= \begin{cases} 1 & \text{if } s_i = \text{'H'} \\ 0 & \text{if } s_i = \text{'P'}, \end{cases} \\
 \text{adj}(v_i, v_j) &= \begin{cases} 1 & \text{if } |v_i - v_j| = 1 \\ 0 & \text{otherwise,} \end{cases} \quad \text{and} \\
 f(\mathbf{v}) &= - \sum_{i=1}^n \sum_{j=i+2}^n h(s_i)h(s_j)\text{adj}(v_i, v_j).
 \end{aligned}$$

This definition of the adjacency function, $\text{adj}(v_i, v_j)$, assumes a regular lattice of unit spacing.

Given a lattice protein \mathbf{s} , the goal is to find a non-self-intersecting conformation \mathbf{v}^* that minimize the energy function f . Intuitively, the energy function counts the number of hydrophobic residue pairs that are spatially adjacent but not adjacent in the residue sequence. For example, in figure 4.5, the lattice protein on the left has two such residue pairs, while the lattice protein on the right has only one. To maximize the number of these pairs, and thus minimize the energy function, hydrophobic residues must form compact clusters, pushing the hydrophilic residues to the periphery.

Optimizing lattice protein conformations is an NP-complete problem in both two and three dimensions [6, 10]. Numerous specialty techniques have been developed to identify optimal and approximate solutions. While advances in computer hardware and software now permit the study of protein models far more complicated than lattice proteins, lattice proteins remain an interesting

limited to binary genes, use binary chromosomes of 141 genes in which groups of three bits encode one positional relation. The chromosomal encoding of a conformation permits self-intersecting conformations, against which the algorithms must select. To penalize self-intersecting conformations, colliding residues (residues occupying the same spatial location) do not contribute to the energy of the conformation.

A fixed MRF neighborhood used with MARLEDA^{+model} exploits the linear nature of lattice proteins. Each residue s_i is dependent on the two residues immediately preceding it, s_{i-2} and s_{i-1} , and the two residues immediately following it, s_{i+1} and s_{i+2} , when present. Broader spans of dependent residues could be used, but using four dependencies was found to be a reasonable balance between solution quality and execution time during preliminary experimentation. While this model is not ideal, since it does not include interactions between well separated genes that may arise for specific conformations, it includes basic knowledge applicable to the entire domain.

Each run of the experimental algorithms was limited to 600,000 fitness function evaluations. The following algorithm parameters were used:

GENEsYs : population size = 1200, full population selection, uniform crossover, elitism, Whitley rank selection with $\alpha = 2.0$, mutation rate = 0.005, crossover rate = 0.5, and generation gap = 1.0.

BOA : population size = 5000, offspring percentage = 50, tournament size = 3, and max incoming links = 8.

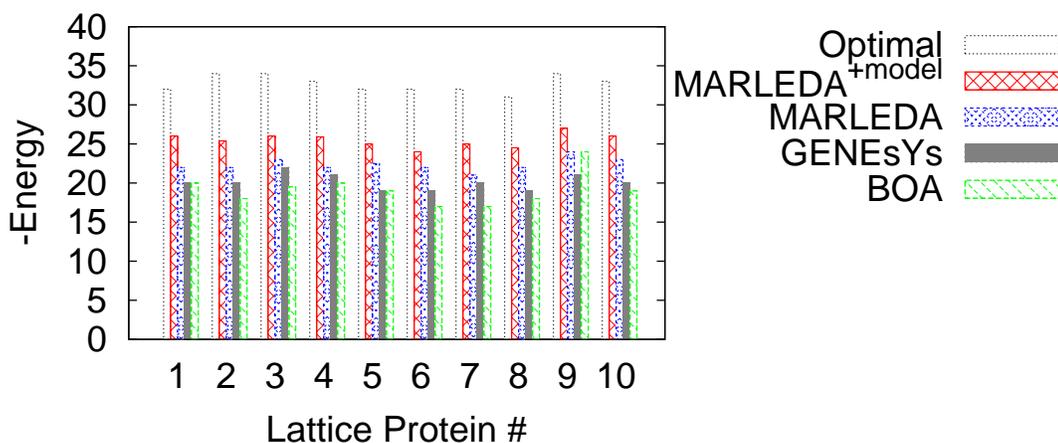


Figure 4.7: Median best energy scores in 50 independent trials over the Harvard vs UCSF lattice protein set. All algorithms produce suboptimal results, though MARLEDA^{+model} demonstrates a 10%–15% margin of improved quality over the other algorithms. Even without the fixed model, MARLEDA found better conformations than GENEsYs and BOA.

MARLEDA : **PopSize** = 4000, **Parents** = 0.2, **TournSize** = 4, **ModelAdds** = 3500, **ModelSubs** = 2000, **Montelters** = 2400, **Mutation** = 0.025, and **Replaced** = 0.45.

MARLEDA^{+model} : **PopSize** = 4000, **Parents** = 0.4, **TournSize** = 3, **Montelters** = 2400, **Mutation** = 0.035, and **Replaced** = 0.5.

Figure 4.7 shows the median best energy score at the end of evolution over 50 independent trials of each algorithm. The optimal energy scores determined by [70] are also included.

Lattice protein optimization is the most difficult of the five optimization tasks presented in this chapter and the results reflect this difficulty. The majority of conformations evolved had less than two-thirds the optimal en-

ergy. MARLEDA produced slightly better conformations than GENE_sYs and BOA, and MARLEDA^{+model} produced notably better conformations than MARLEDA.

Figure 4.8 illustrates four different views of a single conformation for lattice protein #1. As expected, the hydrophobic residues cluster in the center of the conformation and the hydrophilic residues line the periphery or fill in gaps within the conformation. The conformation has an energy of -29 while the optimal conformation energy is -32 , thus the conformation is only slightly suboptimal. The number of optimal conformations for this lattice protein was estimated at above 10^6 [70], hinting that this domain is extremely multimodal. Because there are so many optima, identifying an optimal conformation should be easy, however, any two optima share only 35%–55% of the hydrophobic residue pairs contributing to their energy, thus the optima are well separated within the space of conformations. Even EDAs, which are designed to handle multimodal optimization tasks, have difficulty with this volume of competing optima.

While none of the experimental algorithms demonstrated exemplary performance on this task, the simple step of using an intuitive model with MARLEDA^{+model} increased conformation quality 10%–15%. In fact, all the experiments presented in this chapter have demonstrated this effect; use of a human-provided model consistently improves MARLEDA’s search capability. Furthermore, MARLEDA^{+model} also benefits from classic mutation in every domain, except the extremely simple OneMax domain. These observations

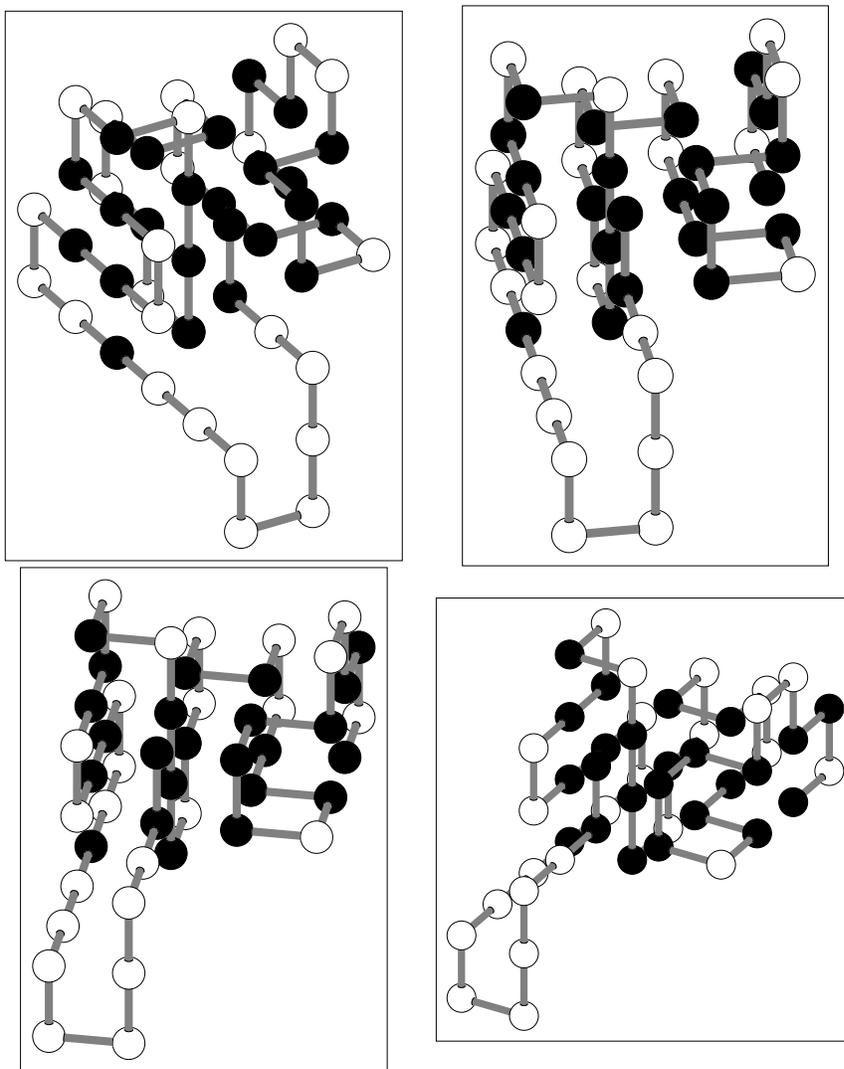


Figure 4.8: Four views of the best conformation for lattice protein #1 discovered by MARLEDA^{+model}. The hydrophobic residues (black) form a single cluster at the center of the conformation, as expected from this simulation of the hydrophobic effect. However, the arrangement of residues is slightly suboptimal. Optimal conformations are significantly different from the shown conformation, demonstrating the multimodal challenge of this domain.

will be utilized in the next chapter, where MARLEDA is applied to a difficult real-world problem.

4.7 Conclusion

The MARLEDA method is a powerful search algorithm in many domains. It can either learn the structure of a problem domain or utilize a provided structure in the form of a Markov random field neighborhood system. In addition, MARLEDA can use mutation to help overcome convergence on local optima. In the next chapter, all these features are utilized to apply MARLEDA to the prediction of RNA secondary structure.

Chapter 5

RNA Structure Prediction

The greatest possible contribution of a new search algorithm such as MARLEDA lays in real-world applications. To that end, I have selected an important problem from computational biology as MARLEDA's first foray into the real world. The prediction of the molecular structure of RNA molecules is both a challenging and important computational problem. In the following sections I briefly describe RNA's role in nature and review current methods for predicting its structure. Section 5.3 covers the adaptation of MARLEDA to this prediction task. Sections 5.4 & 5.5 describe the design of an RNA structure prediction experiment and compare results from MARLEDA and several preexisting methods.

5.1 RNA in Molecular Biology

Ribonucleic acid (RNA) is a nucleic acid polymer similar to the more familiar deoxyribonucleic acid (DNA). RNA and DNA are both chains of nucleotides with a "backbone" of alternating phosphate and sugar residues (ribose sugars in RNA, deoxyribose sugars in DNA) to which a sequence of bases is attached (figure 5.1). The four bases found in RNA are adenine, cytosine,

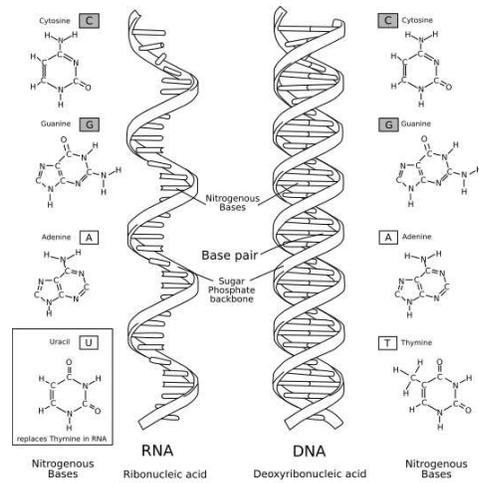


Figure 5.1: Structural comparison of RNA and DNA. Though chemically similar, single-strand RNA and double-strand DNA perform very different biological functions. Image courtesy of *Wikimedia Commons* [66].

guanine, and uracil, with uracil replacing the thymine base found in DNA.

In all but the simplest organisms, RNA and DNA operate together in complementary roles. DNA is the long-term storage medium for genetic information. A pair of DNA strands binds together to form a stable and familiar double-helix. RNA forms much of the molecular machinery necessary for gene expression, i.e. the interpretation of DNA. RNA typically exists as a single strand folded back upon itself, much like a protein, supporting a variety of molecular functions.

RNA is integral to the fundamental process of protein synthesis, whereby a gene's DNA sequence is read and the appropriate protein(s) produced. Protein synthesis involves no less than three classes of RNA molecules. Synthesis

begins with the *transcription* of a gene's DNA sequence. An RNA polymerase enzyme traverses the gene and builds a strand of messenger RNA (mRNA) complementary to the DNA sequence. In this role, mRNA is a temporary carrier of genetic information. The mRNA strand then undergoes *translation* by a ribosome, which constructs the encoded protein one amino acid at a time. Ribosomes are themselves composed predominantly of ribosomal RNA (rRNA), and the transport of amino acids to the emerging protein is performed by transfer RNA (tRNA).

RNA's utility is not limited to protein synthesis. Certain classes of virus use RNA rather than DNA as their genetic material, including familiar viruses like those causing influenza and severe acute respiratory syndrome (SARS). In some higher organisms double-strand RNA (dsRNA) helps regulate gene expression through a process called RNA interference. Recently, it has been discovered that RNA can perform general chemical tasks, functioning as catalysts or enzymes, spawning a new category of RNA called *ribozymes*. Understanding these fundamental molecular processes necessitates study of RNA's basic physical properties, especially the shape of RNA molecules.

5.2 RNA Structure Prediction

Like that of all biomolecules, RNA's role is a product of its composition, its shape, and its surrounding environment. Thanks to decades of gene sequencing research, RNA's composition is well understood. The predominant molecules in RNA's biochemical environment have also been identified. What

remains is to study the mechanics of RNA molecules, which are a product of the molecules' shape. Predicting the shape of RNA molecules is an active area of research with great potential benefits, and the focus of this chapter.

Ideally, we would like to be able to predict the three-dimensional shape, or *tertiary structure*, of an RNA molecule from nothing more than its nucleotide sequence, or *primary structure*. The physical processes that determine an RNA molecule's tertiary structure are very complex, too complex to accurately model with current computational restrictions. However, the number of tertiary structures occurring in nature is very small compared to the number of theoretically possible tertiary structures, and nature seems to utilize those structures that are consistently reproducible. These two observations suggest that it may be possible to predict RNA structure with reasonable accuracy without full physical simulation. In fact, many practical prediction methods exist.

The "gold standard" in structure prediction of biomolecules is crystallography. In this process, a crystallized sample of a molecule is illuminated with radiation, typically X-rays, and the resulting diffraction pattern observed. Analysis of the diffraction pattern provides clues to the composition and structure of the molecule. The process is repeated many times as a hypothesized structure for the molecule is refined.

While crystallography provides high-quality results, it is an expensive and time-consuming process. Difficulties increase as molecule size increases, especially for the wide range of molecules that do not naturally form crystals,

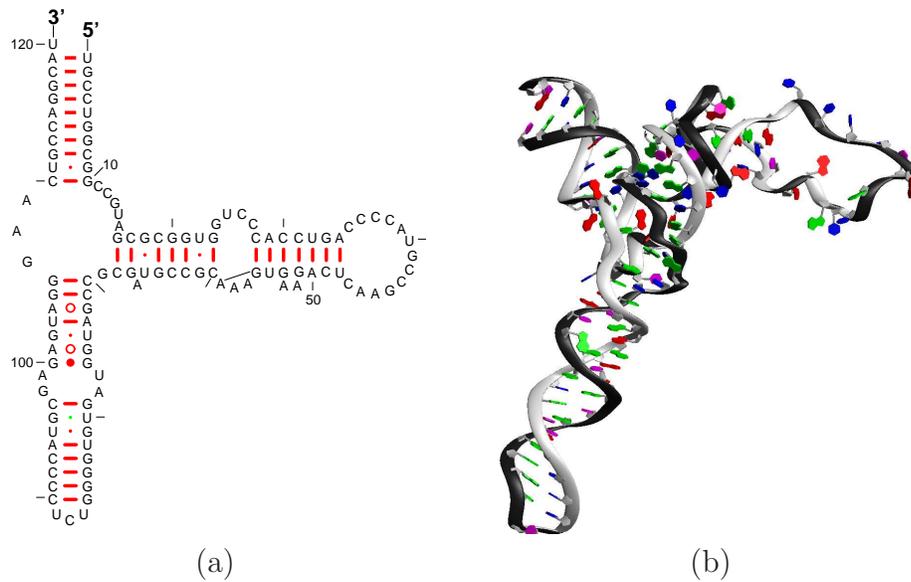


Figure 5.2: Secondary structure (a) and tertiary structure (b) of the 5S subunit of ribosomes in *Escherichia coli*. Bonded nucleotides are connected by line segments, circles, or dots. The tertiary structure’s prominent features are present in the secondary structure, such as the three “branches” of the molecule and the large loop of unpaired nucleotides at the end of the rightmost branch.

including RNA. Nuclear magnetic resonance spectroscopy is an alternative to crystallography, but is presently limited to only small molecules. There has therefore been a great deal of work in predicting RNA structure without resorting to crystallographic methods. In fact, a good prediction can speed subsequent crystallographic verification by providing a strong initial hypothesized structure.

Determining the tertiary structure of biomolecules without crystallography is an extremely difficult task. However, there is a simpler structural form that is still useful, a molecule’s *secondary structure*. The secondary structure

of a molecule is a description of the primary intra-molecule bonds (usually hydrogen bonds) that contribute to the molecule's tertiary structure. The secondary structure of RNA molecules catalogs the nucleotides that have bonded with each other. Figure 5.2 shows the secondary structure (and corresponding tertiary structure) of a piece of ribosomal RNA in the common *E. coli* bacterium. The four nucleotide bases are represented by their initial letters, A, C, G, and U. Canonical bonded nucleotide pairs, A-U and C-G, are connected by red line segments, while non-canonical pairs are connected by circles or dots. The beginning of the RNA sequence is labeled 5' and the end is labeled 3', designations derived from the different exposed sections of the sugars bracketing the sequence. Every tenth nucleotide is marked with a short protruding line, with the 10th, 50th, 100th, and 120th nucleotides labeled explicitly.

Figure 5.2 demonstrates a number of the secondary structure properties that many prediction algorithms exploit. First, each nucleotide is paired with at most one other nucleotide. Second, paired nucleotides tend to exist in contiguous regions. These paired regions give rise to local double-helix structures, which are visible in the tertiary structure. Third, the two halves of a double-helix are formed by anti-parallel regions of the sequence, that is, the sequence of one half read "forwards" is paired with the sequence of the other half read "backwards," according to 5' → 3' directionality. For example, in the first (top-most) double-helix of figure 5.2(a), the first half UGCCUGGCGG (forwards) is paired with CUGCCAGGCA (backwards). Larger RNA molecules may contain double-helices of parallel regions, where both halves match 5' → 3'

directionality, but such helices are rare.

Lastly, any two double-helices are either nested or disjoint, that is, no two double-helices “cross.” More formally, let $\alpha_{5'}$ and $\alpha_{3'}$ be the positions of two nucleotides bonded to each other, where $\alpha_{5'} < \alpha_{3'}$. For example, within the first double-helix of figure 5.2(a) $\alpha_{5'} = 1$ and $\alpha_{3'} = 119$. Let $\beta_{5'}$ and $\beta_{3'}$ be the positions of a nucleotide pair in a different double-helix, such as $\beta_{5'} = 16$ and $\beta_{3'} = 68$. Assuming $\alpha_{5'} < \beta_{5'}$, then either

$$\begin{aligned} \alpha_{5'} < \beta_{5'} < \beta_{3'} < \alpha_{3'} & \text{ (nested) or} \\ \alpha_{5'} < \alpha_{3'} < \beta_{5'} < \beta_{3'} & \text{ (disjoint) but not} \\ \alpha_{5'} < \beta_{5'} < \alpha_{3'} < \beta_{3'} & \text{ (crossed).} \end{aligned}$$

Figure 5.3 illustrates the secondary structure of *E. coli* 5S rRNA in a form where nested (one within another) and disjoint (side-by-side) double-helices are visually apparent. Crossed double-helices, known as pseudoknots, exist in larger RNA molecules but are a small proportion of all double-helices.

The somewhat constrained local features present in secondary structures make it possible to predict tertiary structure from secondary structure. Local patterns of double-helices and loops (contiguous regions of unpaired nucleotides, so named for the curving loop-like structures they produce, e.g. nucleotides 35–47 of figure 5.2(a)) form motifs that tend to be conserved across species. Consequently, though careful comparison the known tertiary structure for an RNA molecule in one organism can be projected to the functionally equivalent molecules of other species. For example, the tertiary structure of

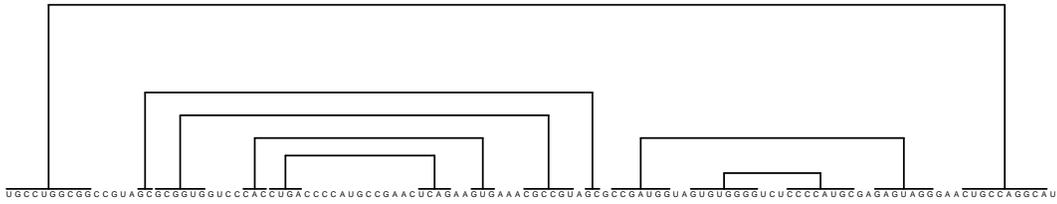


Figure 5.3: Linear representation of the secondary structure of *E. coli* 5S rRNA previously shown in figure 5.2(a). Double-helices are represented by arches above the RNA sequence. Nested and disjoint double-helices are clearly visible. No pseudoknots are present in this molecule, thus no two arches cross one another.

E. coli's 5S ribosomal subunit shown in figure 5.2(b) could be predicted from the tertiary structures of 5S ribosomal subunits of other bacteria. An accurate prediction of secondary structure is a significant step toward determining tertiary structure. The majority of structure prediction algorithms therefore predict secondary structure rather than tertiary structure. The MARLEDA-based prediction method developed in this chapter also predicts secondary structure.

Existing RNA secondary structure prediction algorithms operate using one or more of the following principles, 1) thermodynamics: minimize an energy estimate of the stability of a hypothesized structure, 2) kinetics: simulate simplified RNA folding, or 3) comparison: align and constrain using related RNA sequences. For example, the well known Mfold [71] and RNAfold [31] packages predict secondary structure via dynamic programming minimization of global free energy, estimated from several short-region properties [45] such as nucleotide pairings and tetra-loop energies. An extensive list of active

RNA secondary structure prediction tools is available from the Wikiomics web site [67].

Different prediction algorithms have different strengths and weaknesses, but prediction accuracy varies greatly even for a single algorithm. It is not unreasonable for accuracy to vary between 50%–80% correctly predicted nucleotide pairs. For many algorithms, the predicted “optimal” secondary structure of an RNA molecule is significantly different from the true structure. Consequently, many algorithms also report suboptimal predictions, which may, in fact, be better predictions. This self-deprecating behavior is a clear indicator that existing RNA structure prediction methods leave significant room for improvement.

Of the three classes of prediction principles, comparative analysis forms the basis of the most consistently accurate prediction methods [14, 16]. Such methods frequently use sets of RNA sequences gathered from multiple species to predict a structure common to the entire set. For example, the RNAalifold [30] and Pfold [36] packages use an extension of the Zuker-Stiegler algorithm [73] to determine a consensus structure for a set of aligned RNA sequences.

Of particular interest to this dissertation is the work of Dr. Gutell et al. [8, 22, 20, 21, 40, 41]. Their approach to comparative structure prediction focuses on sequence covariance, i.e. complementary changes in paired nucleotides that permit double-helix structure, and therefore secondary structure, to remain unchanged. Consider the following ribosomal RNA fragments from three

different bacteria:

<i>E. coli</i>	<i>A. tumefaciens</i>	<i>R. capsulatus</i>
UGCCUGGCGG	GACCUUGGUGG	CGUUUGGUGG
ACGGACCGUC	CUGGACCGUC	CCAAACCGCC

These sequence fragments of ten nucleotides are from the previously mentioned 5S ribosomal subunit of their respective organisms. The upper and lower RNA sequences for each bacteria constitute paired region that produce the double-helix immediately adjacent to the 5' and 3' ends of structure shown in figure 5.2(a). Notice that in the majority of spots where the sequences differ both the upper and lower sequence (i.e. both sides of the double-helix) differ. For example, at the second, third, and fourth nucleotides from the left, two organisms have cytosine-guanine pairs, while the third has an adenine-uracil pair. Though each organism shares only approximately 60% sequence identity with either of the other two, these highly differing sequences produce the same functional molecule. Regions of covarying nucleotides among multiple species can give rise to the same local structure, and consequently the same overall structure of the molecule. Analysis of covarying nucleotides therefore permits the prediction of common structure despite major sequence variance.

Covariance analysis has produced some stunningly accurate predictions. For example, a covariance-based comparative analysis of the bacterial 16S (~1540 nucleotides) and 23S (~2900 nucleotides) ribosomal subunits correctly predicted 97%–98% of nucleotide pairs [22] as later verified by crystallography. Covariance analysis is thus a very powerful tool for predicting RNA structure.

While covariance-based RNA prediction is not yet fully automated, decades of comparative analysis research has produced a vast library of RNA structures freely available via the Comparative RNA Website (CRW) [8]. This chapter seeks to answer the following question: Could pure statistical analysis of the CRW database form the basis of a successful RNA structure prediction algorithm?

The dominant RNA structure prediction techniques are based on insights into the RNA domain distilled from decades of research. The approach presented in this chapter is based on a powerful but general-purpose search algorithm, MARLEDA, guided by statistics over known RNA structures. In particular, MARLEDA was used to predict the secondary structure of the 5S ribosomal subunit in several species of bacteria using statistics collected from the known 5S rRNA secondary structures of other bacteria. These statistics consist of several variables, or individual statistics; table 5.1 shows two. Once the set of statistics is chosen, those statistics are considered target statistics, i.e. those to which an optimal secondary structure should conform. During the evolutionary search performed by MARLEDA, the same statistics are computed for each hypothesized secondary structure in the population and compared with the target statistics. The goal is to minimize the difference between the observed and target statistics. Simultaneously minimizing several axes of comparison necessitates a multiobjective form of MARLEDA. The next section describes an extension to MARLEDA for multiobjective optimization.

Base	In loop	In helix	Base	A	C	G	U
A	61.82%	38.18%	A	1.59%	0.91%	5.01%	17.29%
C	32.47%	67.53%	C	-	0.68%	55.29%	2.28%
G	18.28%	81.72%	G	-	-	4.1%	11.26%
U	36.44%	63.56%	U	-	-	-	1.59%

Table 5.1: Two summary statistics for the 5S subunit of bacterial ribosomes. The table on the left shows the proportion of paired (in a double-helix) and unpaired (in a loop) nucleotides. The table on the right shows the distribution of nucleotide pairings. These simple statistics contain moderate to strong biases, such as 81.72% of guanine in double-helices versus 18.28% of guanine in loops, thus helping MARLEDA distinguish between feasible and infeasible secondary structures during search.

5.3 Multiobjective MARLEDA

Multiobjective optimization is the branch of mathematics and computer science devoted to the simultaneous optimization of multiple functions. For evolutionary search algorithms such as MARLEDA, a multiobjective optimization problem is one whose fitness function is a vector function (of at least two dimensions) rather than a scalar function. Typically, a multiobjective fitness function is a composite of individual scalar fitness functions, all to be minimized or maximized. Let k be the dimensionality of the multiobjective fitness function, then

$$\mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})).$$

Multiobjective optimization is interesting when the individual scalar fitness functions have different individual optimums. When all functions cannot be simultaneously satisfied, there is spectrum of “best” solutions representing

trade-offs among the different functions. The core difficulty of multiobjective optimization is the lack of a natural method for comparing such trade-offs. Consequently, it is difficult to order or compare solutions during the selection phase of population-based search algorithms. However, the Pareto dominance relation, originally developed for the study of economic systems, provides a practical solution as will be discussed shortly.

Because genetic algorithms and estimation of distribution algorithms, as described in chapter 2, are well suited for optimizing scalar fitness functions, much research has gone into reformulating multiobjective optimization problems as standard single-objective optimization problems. Such transformations “collapse” the vector function into a scalar function, for example

$$g(\mathbf{x}) = \sum_{i=1}^k \alpha_i f_i(\mathbf{x}), \quad \text{or}$$

$$g(\mathbf{x}) = \prod_{i=1}^k f_i(\mathbf{x}).$$

However, such transformations have drawbacks. Linear combinations require weighting the individual scalar fitness functions, which can be difficult, overly sensitive, and hamper evolution if poorly done. Nonlinear combinations have their own requirements, but the details of the transformation must still be chosen with care. All transformations inherently distort the feedback provided by the vector fitness function, further increasing the difficulty of the optimization problem. These drawbacks make transformations undesirable in all but the simplest domains.

Ad hoc methods also exist for converting a multiobjective problem into a single-objective problem. One simple technique is objective switching, whereby only one dimension of the vector fitness function is optimized at a time. A search algorithm using objective switching might optimize the first dimension for 10 generations, then the second dimension for 10 generations, then the third dimension, etc. Such techniques tend to perform poorly since the search is guided in different, often contradictory, directions at different times, prohibiting overall progress.

The late 18th, early 19th century economist and sociologist Vilfredo Pareto created a formalism for sets that represent trade-offs. Central to his formalism is the notion of Pareto dominance, which has subsequently been used extensively in multiobjective optimization research. Let \mathbf{a} and \mathbf{b} be two solutions to a k -dimensional multiobjective optimization problem. The solution \mathbf{a} *strongly dominates* \mathbf{b} when

$$\mathbf{a} \prec \mathbf{b} \iff \forall i f_i(\mathbf{a}) < f_i(\mathbf{b}),$$

and \mathbf{a} *weakly dominates* \mathbf{b} when

$$\mathbf{a} \preceq \mathbf{b} \iff \forall i f_i(\mathbf{a}) \leq f_i(\mathbf{b}), \exists i : f_i(\mathbf{a}) < f_i(\mathbf{b}).$$

The direction of the inequalities depends on the problem; the above definitions assume a minimization problem.

A solution \mathbf{a} is *weakly nondominated* by a set of other solutions, \mathbf{X} , if no other solution strongly dominates it, that is $\neg \exists \mathbf{x} \in \mathbf{X} : \mathbf{x} \prec \mathbf{a}$. Similarly,

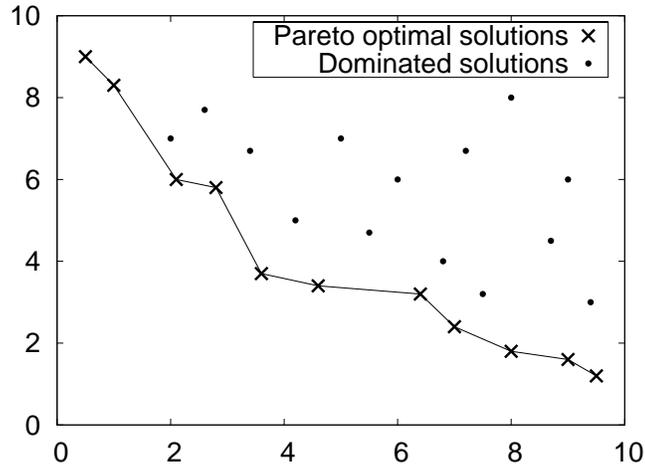


Figure 5.4: Example minimization of a simple multiobjective problem, $\mathbf{f}(x, y) = (x, y)$. Solutions with fitness scores closer to the origin in at least one dimension than all other solutions are Pareto optimal. The set of Pareto optimal solutions forms a spectrum of trade-offs, each solution balancing the competing goals differently.

a solution is *strongly nondominated* if no other solution weakly dominates it, that is $\neg \exists \mathbf{x} \in \mathbf{X} : \mathbf{x} \preceq \mathbf{a}$. Strongly nondominated solutions are also called Pareto optimal. The set of strongly nondominated solutions, $\mathbf{X}^* \subseteq \mathbf{X}$, is called the Pareto optimal set, defined as

$$\mathbf{X}^* = \{\mathbf{x}^* : \neg \exists \mathbf{x} \in \mathbf{X} : \mathbf{x} \preceq \mathbf{x}^*\}.$$

Figure 5.4 illustrates the Pareto optimal set of a simple multiobjective fitness function.

Though Pareto dominance does not define a partial order on a set of solutions, it is sufficient to define a useful ordering for population members [17]. Given a population of chromosomes, $\mathcal{P}(t)$, each chromosome is assigned

a Pareto rank. Rank 1 is assigned to those chromosomes that are strongly non-dominated by the population, i.e. the Pareto optimal set. Rank 2 is assigned to those chromosomes that are strongly nondominated by the population excluding rank 1 chromosomes. Rank 3 chromosomes are strongly nondominated by all but rank 1 & 2 chromosomes, and so forth. This method of generating Pareto rankings can be thought of as providing a simple measure of the distance between a chromosome and the Pareto optimal set of the population.

Using Pareto ranking, the chromosome selection process of multiobjective MARLEDA (mMARLEDA) is slightly different than that of standard MARLEDA (section 3). Chromosomes are ordered by Pareto rank, thus high-ranking chromosomes are considered to be high-fitness and low-ranking chromosomes are considered to be low-fitness. Rank 1 chromosomes are more likely to be members of $\mathcal{P}'(t)$ than rank 2 chromosomes, rank 2 chromosomes are preferred over rank 3 chromosomes, and so on. The order of chromosomes within a rank is arbitrary. Tournament selection can still be employed, but for multiobjective problems where few distinct Pareto ranks exist it is an ineffective method for biasing $\mathcal{P}'(t)$ and should not be used. When tournament selection is disabled in mMARLEDA (by setting **TournSize** = 1), truncation selection is used instead.

5.4 Experimental Design

Two final components are needed before mMARLEDA can predict RNA secondary structures: a chromosomal encoding for secondary structures

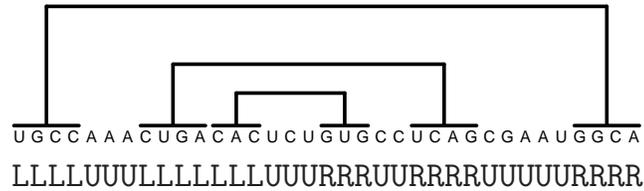


Figure 5.5: The mMARLEDA encoding (at bottom) of the secondary structure of a fragment of RNA. This encoding enforces the common secondary structure restrictions, such double-helices with anti-parallel halves and no pseudoknots.

and a set of target RNA statistics. Secondary structures are encoded in chromosomes using an alphabet of four symbols that classify nucleotides. The four symbols and their interpretations are:

- | | |
|---|--|
| L | A nucleotide on the left-hand (5'-most) half of a double-helix. |
| R | A nucleotide on the right-hand (3'-most) half of a double-helix. |
| U | An unpaired (loop) nucleotide. |
| S | A nucleotide with the same classification as the preceding nucleotide. |

Figure 5.5 illustrates the encoding of the secondary structure of an RNA fragment. Each nucleotide position is classified using one of the four symbols. Each symbol occupies one gene of the chromosomal encoding, thus the encoded secondary structure has the same number of elements as the underlying nucleotide sequence. The MRF neighborhood system used in MARLEDA records those nucleotides whose status (paired or unpaired) are mutually dependent. The encoding scheme is complete, i.e. all possible secondary structures are encodable, and many-to-one, i.e. there are many encodings for each unique secondary structure.

In figure 5.5 the S symbol is unused. However, any repetition of the

other symbols could be equivalently expressed using **S**. For example, the initial symbol series **LLLL** could be rewritten as **LSSS**, **LSLS**, or a number of other variants. While **L**, **R**, and **U** together are sufficient to describe any desired secondary structure, the inclusion of **S** permits small chromosomal changes to affect large structural changes, such as when **LSSS** mutates into **RSSS**. Preliminary experiments demonstrated that this feature significantly reduces the time needed by mMARLEDA to produce high-quality results.

The concrete nucleotide pairings encoded in a chromosome are computed via a simple stack-based method that parses the chromosome left-to-right. When an **L** is encountered the current nucleotide position is recorded on a stack. When an **R** is encountered, a pair is formed between the current nucleotide position and the position on the top of the stack, which is subsequently removed. At the end of parsing, any unpaired **L** and **R** positions remain unpaired, thus all possible chromosomes encode valid secondary structures. This procedure also enforces the common secondary structure restrictions (section 5.2), such as double-helices with anti-parallel halves and no pseudoknots.

Many different statistics can be used as targets for evolution in mMARLEDA. While it is tempting to include as many statistics as possible, including useless or redundant statistics makes the Pareto ranking scheme of section 5.3 ineffective. Each target statistic occupies one dimension of the vector fitness function, and as the number of dimensions increases it becomes less likely that any two chromosomes will have a dominant member. Consequently, the number of chromosomes within each Pareto rank increases

while the number of distinct Pareto ranks decreases, making it more difficult for mMARLEDA to distinguish between high-fitness and low-fitness chromosomes. It is therefore important to select a minimal set of target statistics.

During preliminary experimentation many sets of possible target statistics were explored. Focusing on predicting the 5S ribosomal subunit of bacteria, appendix A describes nine candidate classes of statistics computed from the known 5S rRNA secondary structures of 22 bacterial references. The set of target statistics was chosen from among these nine candidates.

The set of target statistics, while ideally small, must be sufficiently descriptive to distinguish between plausible and implausible secondary structures. Consequently, the set of target statistics should minimize redundancy among its members. For example, the three statistics regarding pairing patterns of nucleotide n -tuples (appendix entries A.2, A.3, and A.4) are highly redundant. The patterns of 1-tuples are the marginal products of the patterns of 2-tuples; all the information contained in the 1-tuple statistics is included in the 2-tuple statistics. Similarly for 2-tuples and 3-tuples, thus only one of these three statistics should be included in the target set.

A few statistics can be reasonably included or excluded from the target set by their very nature. Nucleotide pairing statistics (A.1) describe a fundamental restriction on nucleotide pairs, and are thus included in the target set. Conversely, the statistics on double-helix simple spans (A.6) form an exceptionally large and sparse distribution. Computations involving this distribution would be ill-conditioned and not scale well to larger RNA molecules, thus

A.1, A.4, A.5
A.1, A.4, A.5, A.8 [†]
A.1, A.4, A.5, A.8, A.9
A.1, A.4, A.5, A.7, A.8
A.1, A.4, A.7, A.8
A.1, A.4, A.7, A.8, A.9
A.1, A.4, A.7, A.9

Table 5.2: Promising potential sets of target statistics, referenced by appendix entry. The sets are neither too small, thus lacking descriptive power, nor too large, thus becoming unmanageable for mMARLEDA’s Pareto ranking system. The second set ([†]) of target statistics produced the most accurate secondary structure predictions during preliminary experimentation and was consequently used during final experimentation (section 5.5).

this statistic is excluded from the target set.

Of the remaining candidate statistics, there are many plausible combinations. Table 5.2 lists the sets of potential target statistics that were further evaluated. Of these possibilities, the most accurate predictions were generated by including statistics on nucleotide pairings (A.1), pairing patterns of nucleotide 3-tuples (A.4), double-helix lengths (A.5), and hairpin loop lengths (A.8).

Two of the chosen target statistics are conceptually simple. Nucleotide pairing statistics (A.1) describe the frequency with which each type of nucleotide bonds to another. Not surprisingly, the canonical bonded pairs A-U and C-G dominate this statistic. Double-helix length statistics (A.5) describe the number of nucleotides in each half of a double-helix. The convoluted shape of RNA molecules generally limits the length of double-helices, keeping this

statistic nicely bounded.

The two remaining target statistics are more exotic. Pairing patterns of nucleotide 3-tuples (A.4) describe how groups of three adjacent nucleotides are involved in double-helices and loops. For example, in the RNA fragment of figure 5.5 the first nucleotide 3-tuple, UGC, has the pattern H-H-H (short for helix-helix-helix), since all three nucleotides are part of a double-helix. The second nucleotide 3-tuple, GCC, has the same pattern, while the third 3-tuple, CCA, has a slightly different pattern, H-H-L, since the trailing adenine is part of a loop rather than a double-helix. There are 4^3 possible nucleotide 3-tuples, each with 2^3 possible pairing patterns, for a total of 512 categories in this statistic.

Hairpin loop length statistics (A.8) describe the number of nucleotides belonging to a particular class of unpaired sequence regions. Hairpin loops are loops flanked by the two halves of a single double-helix. The *E. coli* 5S rRNA in figure 5.2 includes two hairpin loops, one at the end of the rightmost “branch” of the secondary structure (nucleotides 35–47) and the other at the bottom of the downward branch (nucleotides 87–89). Hairpin loops tend to be small, containing no more than three or four nucleotides, though bacterial 5S rRNA contains a large hairpin loop of 13 nucleotides.

The A.1 and A.4 statistics are nominal distributions while the A.5 and A.8 statistics are interval and ratio distributions. When used within mMARLEDA’s fitness function, the nominal distributions are compared via Pearson’s χ^2 and the interval distributions are compared via direct cumulative

distribution function (CDF) differencing.

5.5 Results

The primary experiment described in this section is the prediction of the secondary structure of *E. coli*'s 5S ribosomal subunit (~ 120 nucleotides) using target statistics computed from 21 5S rRNA references from 17 other bacterial species. While only the details of this single experiment are presented, an identical experiment was performed for each of the 22 bacterial references available (one-out evaluation). A summary of performance over all 22 experiments is included.

For each experiment, twenty independent trials of mMARLEDA were performed with a limit of 500,000 fitness function evaluations. The following mMARLEDA parameters were used: **PopSize** = 2500, **Parents** = 0.7, **Tournament Size** = 1, **Montelters** = 2400, **Mutation** = 0.03, and **Replaced** = 0.7. For this experiment, Markov random field learning was disabled. Instead, a fixed neighborhood system identical to that of the lattice protein experiments (section 4.6) was used. This simplification reduced the runtime of the mMARLEDA trials without impacting solution quality.

In evaluating the quality of the predicted secondary structures at the end of evolution, the issue of comparing vector fitness scores raised in section 5.3 again emerged. As a population-based search method, mMARLEDA produces a set of alternative secondary structures, each labeled with a four-dimensional fitness score. Unlike the single-objective case, the best prediction

is not obvious. However, it was possible to analyze the populations and devise a method for automatically identifying the best prediction(s).

In all 20 trials, the best secondary structure prediction (as measured by the number of correctly predicted nucleotide pairings) was always a member of the Pareto optimal set of the population. Figure 5.6 shows the six possible two-dimensional projections of the four-dimensional fitness scores within the Pareto optimal set for two of the 20 mMARLEDA trials. The best prediction is highlighted in red. An ideal prediction with no deviation from the target statistics would be located in the lower left corner of each projection. The predictions within the Pareto optimal set, representing a range of trade-offs among the different dimensions, tend to lay near the axes (similar to the function $\frac{1}{x}$), thus all projections are shown in log-log scale for ease of inspection. The domain and range of the axes are irrelevant to this analysis and are omitted.

For most projections, the best prediction does not have a distinctive fitness score; it is lost in a sea of competing fitness scores. However, in the A.4 versus A.1 projection (the first projection within each group of six) the best prediction is an extreme value, or nearly so. The best prediction's fitness score is consistently nearer the origin in the A.4 versus A.1 plane than all but a few other predictions. Consequently, the best prediction(s) can be selected by computing the five or ten population members closest to the origin and identifying the likely candidate among them based on the common structure of the bacterial 5S rRNA references. Such structure can be formulated in a

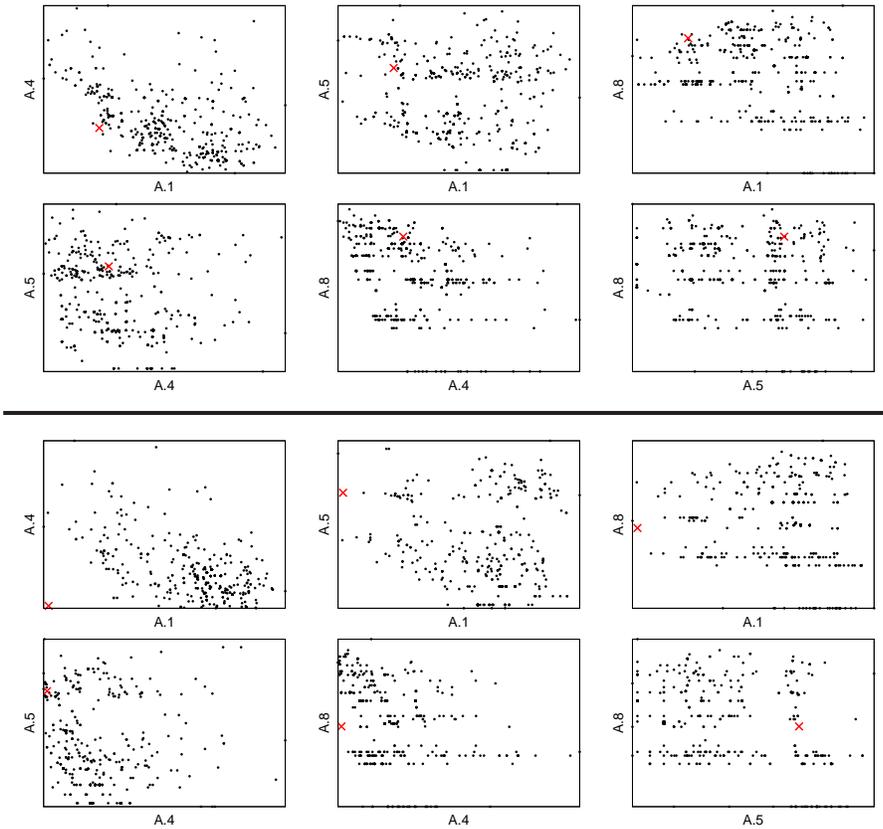


Figure 5.6: Two-dimensional projections of the Pareto optimal set from two mMARLEDA trials. The fitness score of the best secondary structure prediction is highlighted in red. For most projections, the best prediction is not distinctive. However, in the A.4 versus A.1 projection the best prediction is consistently nearer the origin than the majority of other predictions. This phenomenon allows the best prediction to be quickly identified within the Pareto optimal set.

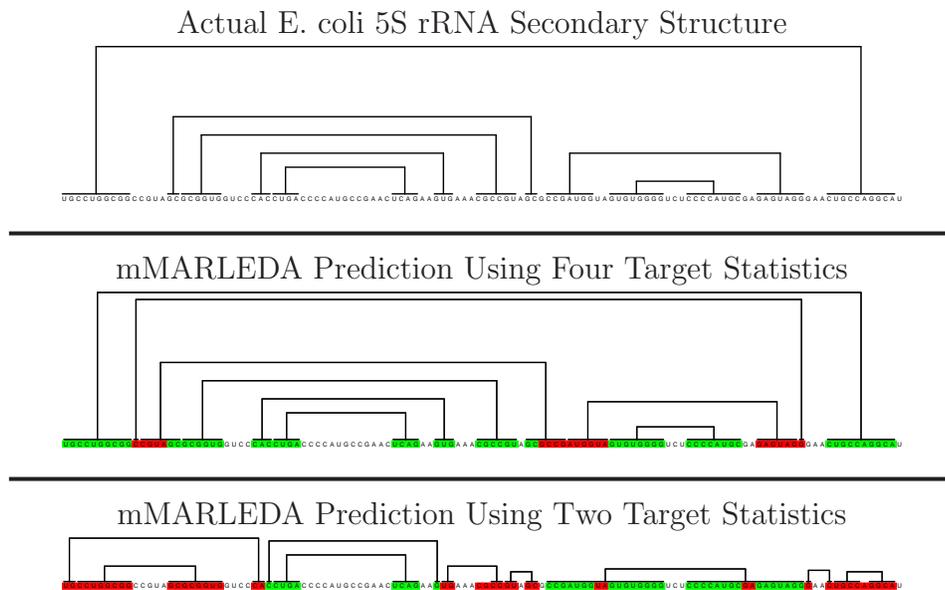


Figure 5.7: The best E. coli 5S rRNA secondary structure predictions by mMARLEDA when using the four prescribed target statistics and only two target statistics. Correctly predicted nucleotide pairs are highlighted in green. Incorrectly predicted or omitted nucleotides pairs are highlighted in red. Using only two target statistics, mMARLEDA identified most of the nucleotides that should form double-helices, but could not predict the exact nucleotide pairings. In contrast, using all four target statistics, mMARLEDA predicted many more nucleotide pairs correctly, resulting in a prediction much closer to the true secondary structure.

complex set of rules; However, since this task is relatively straightforward for a human observer, and often not critical, it was easier for the researcher to simply perform this evaluation manually.

The significance of the nucleotide pairing statistics (A.1) and the pairing pattern of nucleotide 3-tuples statistics (A.4) in identifying the best prediction might lead one to think that mMARLEDA could successfully operate

using only those two statistics. That is not the case. Figure 5.7 shows two of the best predictions evolved by mMARLEDA, one using the four target statistics prescribed in section 5.4 and one using only two target statistics, A.1 and A.4. Using all four target statistics, mMARLEDA correctly predicted the majority of nucleotide pairs. Even in the regions with errors, the general arrangement of double-helices were correct. In contrast, the prediction using only A.1 and A.8 as target statistics was grossly inferior. These two target statistics enabled mMARLEDA to identify most of the nucleotides that should form double-helices, but the organization of the double-helices was largely incorrect. The A.5 and A.8 statistics were therefore necessary for mMARLEDA to discriminate among all the possible arrangements of double-helices that satisfied the two included target statistics.

Four popular and freely available RNA secondary structure prediction tools were selected for comparison with mMARLEDA. Two of the methods, Mfold v3.2 [72] and RNAfold v1.6.4 [29], are based on thermodynamics. For a single RNA sequence, RNAfold generates a single secondary structure prediction, while Mfold generates a family of predictions. The best Mfold prediction, as determined by the accompanying `efn2` tool, was used for comparison purposes.

The two remaining prediction algorithms, RNAalifold v1.6.4 [30] and Pfold [36], form predictions by comparing a set of aligned sequences. These methods use the entire set of 22 RNA sequences to generate a “consensus” structure that is mapped to a secondary structure for each individual sequence.

Several different parameter settings were evaluated for all methods except Pfold, which is parameter-free, and default parameter settings proved best in all cases.

Table 5.3 lists the prediction accuracy of nucleotide pairs for each prediction algorithm across all 22 bacterial 5S rRNA sequences. The reported accuracy for mMARLEDA is the median accuracy of 20 independent trials. To prevent a bias towards similar sequences from the same species, the average accuracy of *A. globiformis* and *G. stearothermophilus* is used in computing the overall prediction accuracy across all 18 species of bacteria.

The comparative prediction methods, Pfold and RNAalifold, and mMARLEDA perform better than the thermodynamics-based algorithms, Mfold and RNAfold. The accuracy of Mfold and RNAfold varies significantly, showing how sensitive these methods are to specific RNA sequences. By incorporating knowledge from multiple sequences, mMARLEDA, Pfold, and RNAalifold can generate consistently superior predictions. While these summary results give a good impression of the relative competence of the algorithms in this one experiment, the structural differences among the predictions are more insightful.

Figure 5.8 show the best secondary structure prediction of each algorithm for the simple case of *E. coli*. The different types of prediction errors made by the various algorithms are quite informative. For example, RNAalifold made only errors of omission, failing to predict nucleotide pairs that exist in the true secondary structure. Those nucleotide pairs RNAalifold did predict

	mMARLEDA	Pfold	RNAalifold	Mfold	RNAfold
<i>A. globiformis</i> #1	87%	82%	74%	64%	38%
<i>A. globiformis</i> #2	81%	79%	74%	56%	31%
<i>A. oxydans</i>	80%	79%	74%	56%	31%
<i>A. tumefaciens</i>	79%	85%	72%	85%	31%
<i>B. subtilis</i>	76%	79%	76%	61%	61%
<i>D. acidovorans</i>	86%	82%	71%	68%	68%
<i>D. radiodurans</i>	68%	80%	72%	78%	68%
<i>E. coli</i>	82%	85%	75%	25%	25%
<i>G. stearothermophilus</i> #1	79%	82%	76%	47%	42%
<i>G. stearothermophilus</i> #2	73%	82%	76%	42%	66%
<i>G. stearothermophilus</i> #3	60%	68%	66%	21%	21%
<i>G. stearothermophilus</i> #4	78%	82%	76%	71%	71%
<i>M. luteus</i>	81%	85%	74%	69%	23%
<i>P. brasiliensis</i>	84%	76%	70%	16%	16%
<i>P. stutzeri</i>	95%	82%	72%	56%	56%
<i>R. capsulatus</i>	92%	82%	47%	74%	74%
<i>S. aureus</i>	81%	81%	76%	49%	76%
<i>S. pasteurii</i>	76%	82%	76%	16%	74%
<i>Synechococcus</i> sp.	81%	77%	74%	69%	79%
<i>T. aquaticus</i>	88%	85%	72%	35%	20%
<i>T. thermophilus</i>	82%	85%	72%	21%	21%
<i>Thermus</i> sp.	79%	82%	28%	77%	62%
Overall	81%	81%	69%	53%	47%

Table 5.3: Prediction accuracy of nucleotide pairs in the secondary structures of 22 bacterial 5S rRNA references. The comparative methods and mMARLEDA consistently perform better than the thermodynamics-based methods. mMARLEDA achieves the same overall accuracy as Pfold, the best of the contemporary prediction tools evaluated.

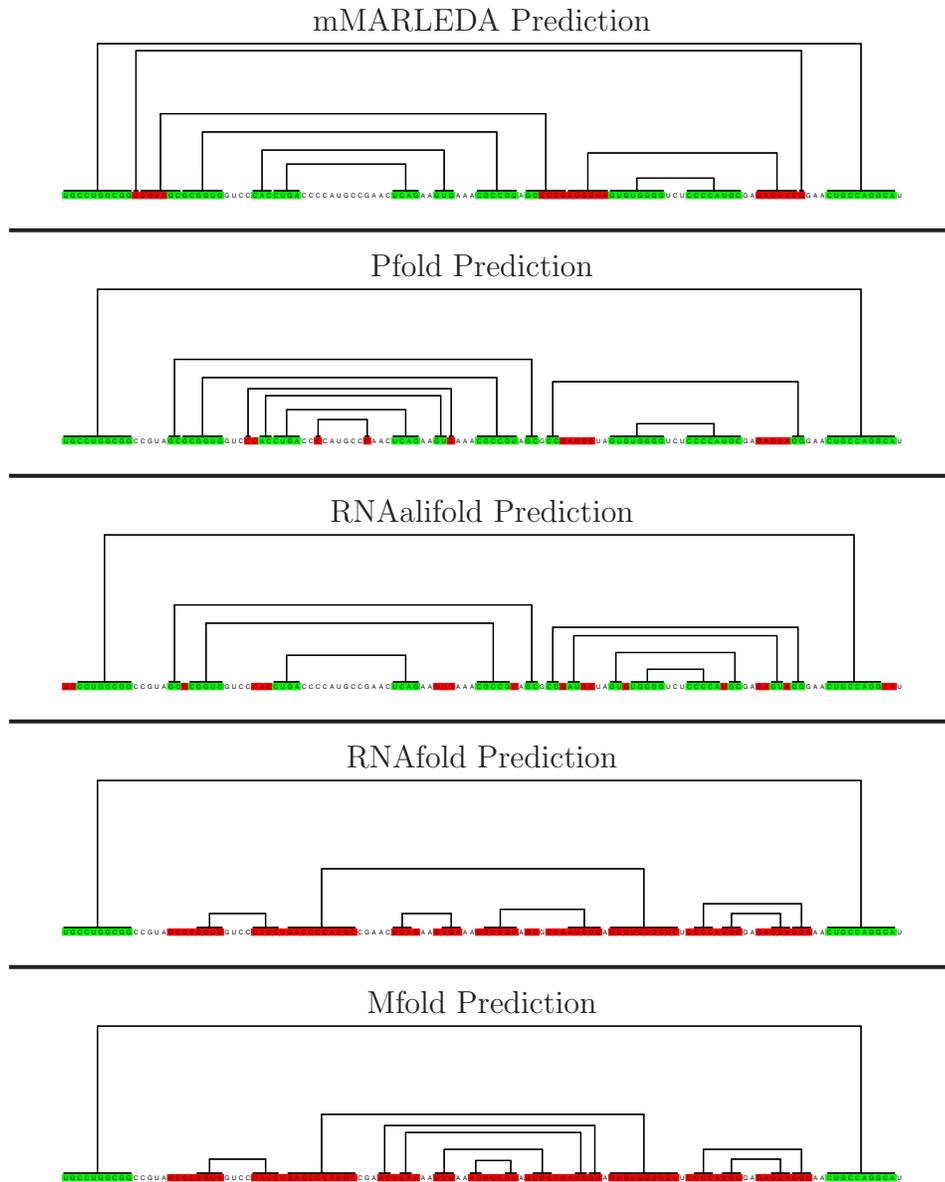


Figure 5.8: The best *E. coli* 5S rRNA secondary structure predictions by the mMARLEDA, Pfold, RNAalifold, RNAfold, and Mfold algorithms.

are correct. “Underpredicting” the total number of nucleotide pairs had the beneficial side-effect of increasing the number of correctly predicted unpaired nucleotides, accounting for RNAalifold’s flawless prediction of all 40 unpaired nucleotides. However, even poor structure predictions can contain many correct unpaired nucleotides, thus paired nucleotides are a far better measure of prediction quality.

The other comparative method, Pfold, also predominantly made errors of omission, though Pfold incorrectly predicted a few nucleotide pairs as well. In contrast to RNAalifold and Pfold, mMARLEDA “overpredicted” the total number of nucleotide pairs; there are no errors of omission. Most of sequence regions forming double-helices were correctly identified by mMARLEDA. However, a few extra erroneous nucleotide pairs precipitated a slight misalignment of one double-helix in the latter half of the sequence, accounting for the majority of mMARLEDA’s errors.

The thermodynamics-based methods, Mfold and RNAfold, performed very poorly on this molecule, making gross errors of every kind. Though E. coli happens to be an especially troublesome molecule for these algorithms, the overall performance of these algorithms is inconsistent with previously reported prediction quality of Mfold [14, 45] and, by extension, RNAfold. Their poor performance in this evaluation is due to two factors. First, in [14] Mfold was evaluated on a set of 90 rRNA sequences. An evaluation using 22 rRNA sequences may therefore be biased towards more difficult sequences and result in poorer overall performance. Second, in these experiments a nucleotide pair

Target fitness score			
(5.15,	274.4,	0.97,	6.33)
Evolved fitness scores			
(7.56,	337.3,	25.76,	19.22)
(3.34,	281.6,	23.46,	6.33)
(18.77,	339.1,	1.65,	17.22)
(29.33,	325.9,	29.22,	6.33)
(8.16,	362.2,	1.15,	3.37)

Table 5.4: Multiobjective fitness scores for the true secondary structure of E. coli 5S rRNA and several mMARLEDA predictions. The target statistics associated with each dimension are A.1, A.4, A.5, and A.8. The first two dimensions are raw χ^2 values and the last two dimensions are unnormalized CDF differences. The majority of evolved structure predictions have fitness scores strongly dominated by the target fitness score of E. coli, indicating that mMARLEDA is not fully optimizing the associated fitness function and needs improvement.

is considered correct if and only if it exists in the known secondary structure. Other evaluations of Mfold allow nucleotide pairs to be shifted left or right by one position and remain “correct” [45]. By taking into account nearly-correct nucleotide pairs, which are beneficial for some evaluation purposes, the method’s apparent accuracy is increased.

Even though mMARLEDA and Pfold perform comparably on this evaluation, mMARLEDA has one crucial advantage. The target statistics used by mMARLEDA are efficiently precomputed and place no theoretical limit on the amount of knowledge that can be included in the system. On the other hand, the current implementation of Pfold does not scale well as the number of RNA sequences evaluated increases. Consequently, mMARLEDA is in a

good position to take advantage of large RNA datasets such as the CRW [8].

While mMARLEDA can form good predictions of secondary structures for this class of RNA molecules, there is room for improvement. In particular, the carefully chosen target statistics are not fully utilized by mMARLEDA. An examination of the fitness scores of mMARLEDA’s best predictions for *E. coli* 5S rRNA (table 5.4) reveals that the best evolved fitness scores are often strongly dominated by the fitness score of the true *E. coli* secondary structure. This disparity indicates that mMARLEDA is not fully optimizing the multiobjective fitness function comprised of target statistics. Had the evolved fitness scores been nondominated by the “ideal” fitness score, mMARLEDA would have been learning as much as it could have from the target statistics; the target statistics would be the limiting factor of the search process. However, this is not the case. Chapter 6 discusses several avenues of improvement for the core MARLEDA and mMARLEDA systems to improve their search capabilities.

5.6 Conclusion

The predictions of the secondary structure of 5S rRNA by mMARLEDA, competitive with the predictions of several existing methods, show that (1) a statistically driven search algorithm can successfully form the basis for a secondary structure prediction tool, and (2) EDAs, and MARLEDA in particular, are ready for deployment on real-world applications. In addition, mMARLEDA is structured to capitalize on the large volumes of biological

data available.

In the RNA domain, the key factor in MARLEDA's success is the intelligent choice of target statistics. An ensemble of complimentary statistics is essential for an effective search. The ideal set of target statistics may not be the same for all classes of RNA molecule, but the cost of experimenting with alternative sets is relatively low and should not prevent the application of MARLEDA to other RNA molecules.

Extensions and further refinement of the MARLEDA algorithm should enable MARLEDA to become a practical tool for RNA researchers. Advancing the MARLEDA method should also promote its use in other difficult real-world domains. The next step in this process is to apply MARLEDA to larger RNA molecules. Predicting the secondary structure of larger molecules will not only demonstrate the power of the MARLEDA method, but also lead to the prediction of novel structures for molecules without crystallographic verification, thereby genuinely contributing to the forefront of RNA research. Predicting the structures of larger molecules will require special attention to scalability, but there are no apparent obstacles to MARLEDA's future success in this domain. The next chapter discusses the work that can make this future a reality.

Chapter 6

Discussion & Future Work

This chapter discusses the insights gained from the benchmark experiments of chapter 4 and MARLEDA’s preliminary success at predicting RNA secondary structure. These insights suggest several opportunities for improving the MARLEDA method and additional avenues of research. In addition, this chapter outlines the steps needed to fully develop the MARLEDA method into a tool for RNA secondary structure prediction.

6.1 Learning Statistical Models

MARLEDA’s strong performance on the five benchmark optimization problems demonstrates that a general Markov random field can form the basis for a successful EDA. Additionally, the MRF-based search algorithm proved superior to both a standard genetic algorithm, GENEsYs, and a sophisticated DAG-based EDA, the Bayesian Optimization Algorithm. However, the advantages of statistical modeling are not free. Compared to simple search methods such as genetic algorithms, EDAs have a much greater startup cost. It costs CPU cycles and fitness evaluations to learn an effective statistical model, and without an effective model EDAs lose their advantage over other methods.

This cost is most evident in the Ising spin glass experiments (section 4.5). The “stress” of that domain, stemming from its structural complexity and the large number of parameters being optimized, elicited very different behaviors from GENEsYs and the EDAs. MARLEDA and BOA initially progressed slowly compared to GENEsYs. During the initial stages of evolution, the first 8,000–13,000 fitness evaluations in the 400 spin domain and the first 20,000–30,000 fitness evaluations in the 900 spin domain, fitness scores improved only slightly (figure 4.4). However, once MARLEDA’s and BOA’s models had been refined, performance improved rapidly. The lengthy refinement period is due to the effect of the large number of spins and spin interactions on the algorithms’ models.

Both algorithms began by optimizing small, localized groups of spins. Consequently, each algorithm’s model was organized around small groups of spins. As evolution progressed, the models adapted to allow optimization across larger sets of spins, forming numerous intricate webs of dependencies. For BOA’s directed graph model, this process caused many learned dependencies supporting locally good fitness to be lost, either to preserve the acyclic property of the graph or to change the direction of dependency. MARLEDA’s MRF model, on the other hand, did not need to destroy learned dependencies in favor of new ones, thus allowing more rapid optimization across the entire system.

In comparison to the Ising spin glass domain, the simpler benchmark problems had far few task parameters and were much easier for the EDAs

to learn. The cost of learning a statistical model was well spent, especially for MARLEDA. The Rosenbrock function experiments, in particular, demonstrate the advantage statistical modeling brings to search. In that domain, the EDAs significantly outperformed GENESYs, with MARLEDA also finding much better solutions than BOA. The relative equality of all three algorithms in the spin glass domain is therefore something of a disappointment. However, the excellent results of MARLEDA^{+model}, which used an ideal model, indicate that it is a deficiency in MARLEDA’s MRF learning mechanism that is inhibiting its performance. Thankfully, there are several ways to improve this mechanism.

The current MRF learning mechanism only evaluates the quality of local structures within the model, specifically gene pairs, and explicitly avoids evaluating the global accuracy of the entire model. A global accuracy estimate would provide a better metric for constructing the model, and several DAG-based EDAs use such metrics like the Kullback-Leibler divergence [38] or minimum description length (MDL) [58], but these metrics are less practical for undirected graph models. The computational complexity of computing such metrics for undirected graph models is far greater than that of DAGs, making them infeasible. Consequently, local quality metrics are the only practical approach for constructing the model.

MARLEDA’s current quality metric, Pearson’s χ^2 test, could be replaced by a new local metric, such as mutual information. However, it is not obvious that any alternative metric would have a positive effect on model ac-

curacy. A survey of alternatives would be worthwhile project, but there is also the intriguing possibility of simply constructing the model differently.

There are numerous MRF construction policies that may perform better than the current greedy policy. As a simple example, a limit could be imposed on the number of neighbors per gene. Such a limit would restrict local complexity within the model, making it easier for Pearson's χ^2 test to accurately measure the confidence of each neighbor relation and subsequently construct a more accurate model. However, such a limit would reduce the potential precision of the model. Whether the increase in accuracy outweighs the decrease in precision would have to be determined experimentally.

An alternative limit might restrict the *total* number of neighbor relations in the model, thus restricting global complexity. It is not difficult to devise more elaborate construction biases, each promoting a different feature within the model. For example, to promote global connectivity within the model, sets of neighbors could be added to the MRF neighborhood system at once rather than individually. Pearson's χ^2 test could be performed for all non-neighbor pairs, effectively producing a set of weighed edges between the genes. The edges forming the maximally weighted spanning forest could then be added to the model. Several iterations of this process would produce a well-connected and somewhat uniform model, perfectly appropriate for domains such as Ising spin glasses. Further investigation is needed to identify those construction policies with broad benefits.

6.2 Using Hand-Crafted Statistical Models

Across all the benchmark experiments, MARLEDA^{+model} consistently produced the best solutions. The simple act of providing MARLEDA with a fixed MRF neighborhood system derived from human knowledge dramatically improved MARLEDA’s search effectiveness. For example, in the Ising spin glass experiments standard MARLEDA routinely produced solutions approximately 80%–85% of optimal, while MARLEDA^{+model} discovered optimal or near-optimal solutions. A priori knowledge of a suitable MRF model is domain specific, but when available, is easily integrated into and used by the MARLEDA method.

There is a curious phenomenon present in the experiments with MARLEDA^{+model}. In all those experiments (excluding the trivial OneMax domain), the algorithm tuning process discovered that mutation was beneficial to the search, even when the corresponding MARLEDA experiments did not use mutation. While overall success of the MARLEDA^{+model} experiments demonstrates that MARLEDA’s MRF sampling procedure working effectively, the use of mutation suggests there is room for improvement.

6.3 Reviving Mutation

An accurate statistical model is useless without an effective sampling procedure. The current Markov chain Monte Carlo sampler is extremely simple yet effective. However, MARLEDA’s population represents a very limited window into the problem domain. Consequently, the domain structure encoded

in the MRF model may not precisely match the structure of high-fitness chromosomes within the population. It is therefore possible for even an accurate model to mistakenly preserve low-fitness attributes of chromosomes. Such errors will not significantly impede the progress of the search if they are rare. However, even a low occurrence of such errors could effect the search enough to justify fixing this problem.

MARLEDA uses mutation to compensate for weakness in its model. The local search performed by mutation implements a form of fine-tuning. The surprisingly good performance of GENE_Ys in the Ising spin glass experiments shows that local search methods, although simple, can be effective in complex domains. The experiments with MARLEDA^{-mutation} and BOA^{+mutation} specifically demonstrate that mutation can contribute to EDA performance. There is therefore reason to believe that hybridizing EDAs with a more sophisticated local search method may result in more efficient search [54].

One strong candidate for hybridization with MARLEDA is hill climbing, which is effectively an aggressive form a mutation. In the combined system, hill climbing occurs after MRF sampling and in place of mutation, further fine-tuning each new chromosome. The MRF model would therefore only need to be accurate enough to generate samples in the vicinity of high-fitness chromosomes, thus reducing MARLEDA's sensitivity to MRF model deficiencies. If successful, such a hybrid algorithm might also lessen the computational demands of the MRF learning and sampling procedures and thereby make it more scalable.

6.4 Scalability & RNA Secondary Structure Prediction

In chapter 5, mMARLEDA proved to be a competitive predictor of RNA secondary structure compared to five other methods. However, the RNA experiments were limited to a set of relatively small molecules of ~ 120 nucleotides. Two issues needed to be addressed before mMARLEDA can be applied to larger and more interesting molecules: utilizing MRF learning in the RNA domain and overall scalability.

In section 5.5, mMARLEDA’s secondary structure predictions were produced without the benefit of MRF neighborhood system learning. In preliminary experiments, MRF learning was tried and it produced predictions equal to the MRF with a hand-crafted neighborhood system, but the learned MRFs were more complicated than the hand-crafted systems and therefore required more runtime to process. While not a large issue for the relatively small 5S rRNA molecules studied, learning is more advantageous for larger molecules and should thus be encouraged.

The learned MRF neighborhoods tended to omit many of the relations encoded in the hand-crafted MRF neighborhoods. Those relations, while logically supported by the linear nature of RNA strands, are not statistically prevalent in every evolved population and are thus not learned. In order to solve this slight overfitting of the model to the population, while still permitting the inclusion of human knowledge, it is reasonable to augment mMARLEDA’s model with partially immutable neighborhood systems. The immutable human knowledge is therefore preserved at all times and mMARLEDA learns

only additional dependencies among nucleotides.

Like most EDAs, the bulk of mMARLEDA's runtime is spent learning and sampling its model. Since mMARLEDA's model is undirected, an even greater proportion of that runtime is devoted to sampling. It is therefore important to find new ways to sample the MRF model efficiently. The current Markov chain Monte Carlo sampler is sufficient for optimization tasks of the scale presented in chapter 4. However, this class of sampler does not scale well as the number of optimization parameters and MRF complexity increases. More efficient alternatives include the Metropolis-Hastings algorithm [27] and Gibbs sampling [7, 23], which have been successfully used in other EDAs. In addition to these algorithmic improvements, modern multi-CPU, multi-core, and clustered computing environments provides an additional solution.

The MRF sampling process is repeated once for every new chromosome constructed by mMARLEDA. However, each sampling is independent of all others. Consequently, each sampling can be performed by a different CPU, thus achieving a significant overall speedup. With only minor modifications, the current implementation of MARLEDA and mMARLEDA could utilize parallel computing hardware via common libraries such as Message Passing Interface (MPI) or Parallel Virtual Machine (PVM). This simple extension will greatly increase the size of RNA molecules that mMARLEDA can process.

6.5 RNA Target Statistics

The four target RNA statistics used in the RNA prediction experiments were each chosen to contribute different and useful information to the secondary structure evaluation process. However, it is possible that other classes of molecule, or molecules of greater size, would be better served by different secondary statistics. For example, the hairpin loop length statistic is relevant to the 5S rRNA domain because hairpin loops are a significant feature of those molecules. For a larger molecule such as 16S rRNA, hairpin loops are proportionally less significant. Statistics over other types of loops may be more useful. The most basic statistics, such as the frequency of nucleotide pairs, are likely useful for all RNA molecules, but once mMARLEDA can be scaled-up to larger molecules alternate statistics can be explored.

The original formulation of mMARLEDA as the basis for an RNA structure prediction method intended to keep the amount of specialist knowledge to a minimum. Only easily computed statistics from raw RNA data are used to drive mMARLEDA. However, existing prediction algorithms encapsulate a significant amount of useful domain knowledge. The ability of mMARLEDA to predict RNA secondary structures might improve if these “foreign” evaluation metrics were also used. A combination of metrics might take the place of several of the target statistics currently used, though probably not all. Thankfully, exploring this possibility does not depend on improving mMARLEDA’s scalability.

6.6 Conclusion

The MARLEDA search method is already an effective search algorithm for combinatorial optimization. It has proven successful in several benchmark domains and a difficult real-world domain. MARLEDA can be made more robust and effective in several ways. First, the MRF model can be constructed in an alternative manner depending on the expected structure of a particular problem domain. Second, the algorithm can incorporate a “backup” search procedure such as hill climbing to help fine-tune its solutions. Third, MARLEDA can be scaled-up to optimization problems with more parameters by utilizing parallel and cluster computing hardware.

Further developing mMARLEDA into an RNA secondary structure prediction tool is primarily a matter of improving the scalability of mMARLEDA. Once this extension is complete, alternative methods for evaluating the quality of proposed secondary structures can be explored. These alternative methods include additional RNA statistics and evaluation metrics from other prediction algorithms. Such improvements should make mMARLEDA a practical tool for RNA researchers.

Chapter 7

Conclusion

EDAs' ability to tackle difficult combinatorial optimization problems makes them strong candidates for application to real-world problems. Domains of daunting complexity, such as those in computational biology, can greatly benefit from techniques that inherently exploit domain structure. The primary contribution of this dissertation is the development of the MARLEDA search algorithm, which has helped make EDAs' potential a reality. This chapter summarizes the important contributions of this research and concludes this dissertation.

7.1 Contributions

The MARLEDA method is a composite of several important ideas and procedures. First and foremost, MARLEDA employs a sophisticated statistical model based on Markov random fields. In chapter 3, two techniques were developed to utilize the model's full potential: (1) a greedy search method for constructing the MRF neighborhood system based on Pearson's χ^2 test, and (2) a modified Markov chain Monte Carlo sampler for constructing new chromosomes.

In chapter 4 MARLEDA was compared to two other search algorithms on a set of standard combinatorial optimization tasks. Not only was MARLEDA shown to be the most successful of these algorithms, but two hypothesized capabilities of MARLEDA were verified. First, the presumption that mutation no longer has a useful role in EDAs was challenged and proved false. The effectiveness of the MARLEDA method was boosted by mutation in several problem domains. Second, MARLEDA's MRF model successfully incorporated human knowledge, dramatically improving its search performance. This last capability will be increasingly useful as MARLEDA is applied to more complex optimization tasks.

Finally, in chapter 5 the MARLEDA method was extended for multiobjective optimization problems and applied to RNA secondary structure prediction. This application demonstrates that RNA structure can be successfully predicted using comparative statistics. Careful selection of informative statistics made MARLEDA an excellent prediction algorithm. Further improvements to MARLEDA will make it a practical tool for RNA researchers.

7.2 Conclusion

MARLEDA has proven to be an effective general purpose search algorithm for combinatorial optimization. It is capable of learning the structure of a problem domain or utilizing a known structure, thus enabling MARLEDA to easily accommodate human domain knowledge. This power and flexibility make MARLEDA the most sophisticated EDA in existence today and an

attractive tool for solving difficult computational search problems.

Computational science depends on the development of methods like MARLEDA. While there may be a temptation to believe that each unique computational problem requires a unique solution method, there do exist common elements of structure and knowledge. These elements allow generic computational methods to solve many types of problem; there is no need to reinvent the wheel for every new car. This generality also means that search algorithms such as MARLEDA must be more self-sufficient, learning about the problem domain in order to implement search effectively. Future improvements to the MARLEDA method will broaden its applicability and increase its utility to researchers. MARLEDA should become a useful tool for scientists and help inspire new and better search technologies.

Appendix A

Bacterial 5S rRNA Statistics

This appendix provides a reference for the nine statistics computed over all 22 bacterial 5S rRNA secondary structures available in the CRW database [8]. All statistics are reported in percentages rather than raw frequencies.

A.1 Nucleotide Pairings

This statistic records the proportional occurrence of all nucleotide pairing combinations, ignoring $5' \rightarrow 3'$ directionality.

Base	A	C	G	U
A	1.59%	0.91%	5.01%	17.29%
C	-	0.68%	55.29%	2.28%
G	-	-	4.1%	11.26%
U	-	-	-	1.59%

A.2 Pairing Patterns of Nucleotide 1-tuples

This statistic records the proportional occurrence of each nucleotide in loops (unbonded) and in double-helices (bonded).

Base	In loop	In helix
A	61.82%	38.18%
C	32.47%	67.53%
G	18.28%	81.72%
U	36.44%	63.56%

A.3 Pairing Patterns of Nucleotide 2-tuples

This statistic records the proportional occurrence of adjacent nucleotides in loops, in double-helices, and in transitions between loops and double-helices. Nucleotide 2-tuples follow 5' → 3' directionality, thus symmetric 2-tuples, e.g. AC and CA, do not have symmetric distributions.

Base	Loop-loop	Loop-helix	Helix-loop	Helix-helix
AA	79.1%	4.48%	1.49%	14.93%
AC	24.84%	28.03%	7.01%	40.13%
AG	11.34%	46.39%	0%	42.27%
AU	48.19%	4.82%	10.84%	36.14%
CA	37.24%	9.66%	17.24%	35.86%
CC	30.98%	4.04%	3.03%	61.95%
CG	18.72%	5.48%	11.42%	64.38%
CU	11.11%	5.05%	12.12%	71.72%
GA	34.83%	2.25%	14.61%	48.31%
GC	6.25%	19.27%	9.38%	65.1%
GG	3.52%	3.52%	13.67%	79.3%
GU	4.76%	3.7%	25.93%	65.61%
UA	49.54%	0.92%	19.27%	30.28%
UC	55%	2.5%	2.5%	40%
UG	6.99%	2.1%	4.9%	86.01%
UU	29.17%	0%	18.06%	52.78%

A.4 Pairing Patterns of Nucleotide 3-tuples

Nucleotide 3-tuples follow $5' \rightarrow 3'$ directionality, thus symmetric 3-tuples, e.g. ACG and GCA, do not have symmetric distributions.

Base	L-L-L	L-L-H	L-H-L	L-H-H	H-L-L	H-L-H	H-H-L	H-H-H
AAA	70.37%	3.7%	0%	18.52%	0%	0%	0%	7.41%
AAC	49.06%	37.74%	0%	0%	1.89%	0%	0%	11.32%
AAG	29.79%	46.81%	0%	2.13%	2.13%	0%	0%	19.15%
AAU	28.57%	28.57%	0%	0%	0%	0%	0%	42.86%
ACA	0%	33.33%	0%	30.77%	10.26%	0%	20.51%	5.13%
ACC	3.77%	18.87%	0%	11.32%	9.43%	0%	0%	56.6%
ACG	11.76%	2.94%	2.94%	38.24%	0%	0%	0%	44.12%
ACU	3.7%	18.52%	0%	44.44%	7.41%	0%	7.41%	18.52%
AGA	0%	9.38%	0%	65.63%	0%	0%	9.38%	15.63%
AGC	6.67%	18.33%	1.67%	66.67%	0%	0%	3.33%	3.33%
AGG	1.96%	5.88%	0%	9.8%	0%	0%	11.76%	70.59%
AGU	0%	0%	0%	45.1%	0%	0%	1.96%	52.94%
AUA	69.23%	0%	0%	0%	19.23%	0%	3.85%	7.69%
AUC	66.67%	0%	0%	20%	13.33%	0%	0%	0%
AUG	16.67%	0%	0%	0%	0%	0%	0%	83.33%
AUU	70%	0%	0%	10%	0%	0%	0%	20%
CAA	21.05%	5.26%	0%	0%	26.32%	0%	0%	47.37%
CAC	0%	13.73%	0%	23.53%	7.84%	1.96%	3.92%	49.02%
CAG	16.22%	8.11%	0%	5.41%	0%	37.84%	0%	32.43%
CAU	86.11%	2.78%	0%	0%	0%	0%	2.78%	8.33%
CCA	41.27%	0%	0%	9.52%	6.35%	1.59%	11.11%	30.16%
CCC	31.78%	0.93%	0%	5.61%	3.74%	0%	0.93%	57.01%
CCG	26.32%	3.16%	0%	0%	0%	0%	6.32%	64.21%
CCU	9.68%	0%	0%	0%	0%	0%	6.45%	83.87%
CGA	43.33%	1.67%	0%	1.67%	1.67%	0%	0%	51.67%
CGC	0%	10.91%	0%	9.09%	3.64%	29.09%	0%	47.27%
CGG	0%	3.13%	0%	0%	6.25%	1.56%	3.13%	85.94%
CGU	2.5%	12.5%	0%	15%	0%	2.5%	0%	67.5%
CUA	15%	5%	0%	0%	15%	0%	20%	45%
CUC	3.7%	3.7%	0%	14.81%	11.11%	3.7%	0%	62.96%
CUG	2.63%	0%	0%	2.63%	0%	2.63%	2.63%	89.47%
CUU	27.27%	0%	0%	0%	18.18%	0%	9.09%	45.45%

Base	L-L-L	L-L-H	L-H-L	L-H-H	H-L-L	H-L-H	H-H-L	H-H-H
GAA	55.22%	4.48%	0%	0%	32.84%	0%	2.99%	4.48%
GAC	0%	3.23%	0%	12.9%	9.68%	0%	29.03%	45.16%
GAG	0%	34.78%	0%	0%	0%	2.17%	0%	63.04%
GAU	12.12%	3.03%	0%	0%	0%	0%	24.24%	60.61%
GCA	14.29%	0%	0%	4.76%	4.76%	0%	33.33%	42.86%
GCC	5.13%	0%	0%	35.9%	7.69%	0%	10.26%	41.03%
GCG	0%	6.45%	0%	0%	4.84%	1.61%	29.03%	58.06%
GCU	4.17%	0%	0%	33.33%	4.17%	0%	16.67%	41.67%
GGA	6.78%	0%	0%	0%	40.68%	0%	27.12%	25.42%
GGC	0%	0%	0%	9.76%	4.88%	4.88%	36.59%	43.9%
GGG	5.33%	1.33%	0%	2.67%	0%	0%	28%	62.67%
GGU	0%	0%	0%	3.7%	8.64%	0%	59.26%	28.4%
GUA	6.98%	0%	0%	0%	39.53%	0%	2.33%	51.16%
GUC	8%	0%	0%	6%	52%	0%	6%	28%
GUG	0%	0%	0%	1.82%	5.45%	0%	10.91%	81.82%
GUU	4.88%	0%	0%	7.32%	7.32%	0%	26.83%	53.66%
UAA	10.53%	5.26%	0%	5.26%	78.95%	0%	0%	0%
UAC	19.05%	57.14%	0%	0%	0%	14.29%	0%	9.52%
UAG	1.56%	48.44%	0%	0%	0%	4.69%	0%	45.31%
UAU	60%	0%	0%	0%	0%	0%	0%	40%
UCA	72.73%	0%	0%	0%	0%	0%	13.64%	13.64%
UCC	67.27%	1.82%	0%	3.64%	0%	0%	0%	27.27%
UCG	34.62%	7.69%	0%	0%	0%	0%	0%	57.69%
UCU	6.25%	0%	0%	6.25%	12.5%	0%	25%	50%
UGA	11.11%	0%	0%	0%	14.81%	0%	25.93%	48.15%
UGC	8.57%	2.86%	0%	5.71%	2.86%	2.86%	0%	77.14%
UGG	0%	3.03%	0%	0%	0%	0%	9.09%	87.88%
UGU	0%	6.67%	0%	6.67%	6.67%	0%	0%	80%
UUA	25%	0%	0%	0%	0%	0%	75%	0%
UUC	34.62%	3.85%	0%	0%	42.31%	0%	0%	19.23%
UUG	5.56%	5.56%	0%	0%	0%	5.56%	0%	83.33%
UUU	42.86%	0%	0%	0%	14.29%	0%	14.29%	28.57%

A.5 Double-helix Lengths

This statistic records the proportional occurrence of double-helix lengths, i.e. the number of nucleotides in each half of the double-helix.

Length	1	2	3	4	5	6	7
	0.64%	12.74%	17.2%	14.01%	0.64%	15.29%	17.83%
Length	8	9	10				
	14.01%	1.27%	6.37%				

A.6 Double-helix Simple Spans

This statistic records the proportional occurrence of double-helix simple spans, i.e. the number of nucleotides between the interior edges of the double-helix. Omitted spans 56–90 are all 0%.

Span	0	1	2	3	4	5	6
	0%	0%	0%	1.91%	11.46%	0.64%	0%
Span	7	8	9	10	11	12	13
	0%	0%	0%	0%	0%	0%	14.01%
	14	15	16	17	18	19	20
	0%	0%	0%	0.64%	0%	0%	0.64%
	21	22	23	24	25	26	27
	0%	3.18%	15.29%	8.28%	0%	0%	0%
	28	29	30	31	32	33	34
	0%	0%	0%	0%	0%	0%	0%
	35	36	37	38	39	40	41
	0%	12.74%	1.27%	0%	0%	0%	0%
	42	43	44	45	46	47	48
	0%	0.64%	0%	0%	0%	0%	0%
	49	50	51	52	53	54	55
	11.46%	1.27%	0.64%	0%	0%	0%	0%

Span	91	92	93	94	95	96	97
	0%	0.64%	0%	0%	0%	0.64%	0%
Span	98	99	100	101	102	103	104
	3.18%	1.91%	6.37%	0.64%	0.64%	0%	0%
	105	106	107	108	109	110	111
	0%	0%	0%	0%	0%	0%	0%
	112	113					
	0%	1.91%					

A.7 Double-helix Conditional Spans

This statistic records the proportional occurrence of double-helix conditional spans, i.e. the number of nucleotides between the interior edges of the double-helix excluding nucleotides in nested double-helices or spanned by nested double-helices.

Span	0	1	2	3	4	5	6
	0%	15.29%	14.01%	2.55%	25.48%	0.64%	0%
Span	7	8	9	10	11	12	13
	13.38%	0.64%	13.38%	0.64%	0%	0%	14.01%

A.8 Hairpin Loop Lengths

This statistic records the proportional occurrence of hairpin loop lengths, i.e. the number of nucleotides in a loop that is bracketed by a single double-helix.

Length	1	2	3	4	5	6	7	8	9
	0%	0%	6.82%	40.91%	2.27%	0%	0%	0%	0%
Length	10	11	12	13					
	0%	0%	0%	50%					

A.9 Non-hairpin Loop Lengths

This statistic records the proportional occurrence of non-hairpin loop lengths, i.e. the number of nucleotides in a loop that is not bracketed by a single double-helix.

Length	1	2	3	4	5
	29.41%	33.19%	18.49%	9.66%	9.24%

Bibliography

- [1] M. Alden, A.-J. van Kesteren, and R. Miikkulainen. Eugenic evolution utilizing a domain model. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, pages 279–286, San Francisco, CA, 2002. Morgan Kaufmann.
- [2] T. Bäck. *A User's Guide to GENEsYs 1.0*. Department of Computer Science, University of Dortmund, 1992.
- [3] S. Baluja. Population-based incremental learning: A method for integrating genetic search based function optimization and competitive learning. Technical Report CMU-CS-94-163, Carnegie Mellon University, Pittsburgh, PA, 1994.
- [4] S. Baluja and S. Davies. Using optimal dependency-trees for combinatorial optimization: Learning the structure of the search space. Technical Report CMU-CS-97-107, Carnegie Mellon University, 1997.
- [5] S. Baluja and S. Davies. Fast probabilistic modeling for combinatorial optimization. In *Proceedings of the 15th National Conference on Artificial Intelligence and the 10th Annual Conference on Innovative Applications of Artificial Intelligence*, Menlo Park, CA, 1998. AAAI Press.

- [6] B. Berger and T. Leighton. Protein folding in the hydrophobic-hydrophilic (HP) model is NP-complete. In *Proceedings of the Second Annual International Conference on Computational Molecular Biology*, pages 30–39, New York, NY, 1998. ACM Press.
- [7] J. Besag. Spatial interaction and the statistical analysis of lattice systems. *Journal of the Royal Statistical Society, Series B*, 36:192–236, 1974.
- [8] J. Cannone, S. Subramanian, M. Schnare, J. Collett, L. D’Souza, Y. Du, B. Feng, N. Lin, L. Madabusi, K. Müller, N. Pande, Z. Shang, N. Yu, and R. Gutell. The comparative RNA web (CRW) site: an online database of comparative sequence and structure information for ribosomal, intron, and other RNAs. *BMC Bioinformatics*, 3:15, 2002.
- [9] G. Cooper and E. Herskovits. A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9:309–347, 1992.
- [10] P. Crescenzi, D. Goldman, C. Papadimitriou, A. Piccolboni, and M. Yannakakis. On the complexity of protein folding. *Journal of Computational Biology*, 5(3):423–466, 1998.
- [11] J. De Bonet, C. Isbell, and P. Viola. MIMIC: Finding optima by estimating probability densities. In *Advances in Neural Information Processing*, volume 9. MIT Press, Cambridge, MA, 1997.
- [12] K. Deb and D. Goldberg. Analyzing deception in trap functions. In

- Foundations of Genetic Algorithms*, pages 93–108. Morgan Kaufmann, San Francisco, CA, 1992.
- [13] K. Dill. Theory for the folding and stability of globular proteins. *Biochemistry*, 24(6):1501–1509, 1985.
- [14] K. Doshi, J. Cannone, C. Cobaugh, and R. Gutell. Evaluation of the suitability of free-energy minimization using nearest-neighbor energy parameters for RNA secondary structure prediction. *BMC Bioinformatics*, 5:105, 2004.
- [15] R. Etxeberria and P. Larrañaga. Global optimization with Bayesian networks. In *Proceedings of the Second Symposium on Artificial Intelligence*, pages 332–339, 1999.
- [16] P. Gardner and R. Giegerich. A comprehensive comparison of comparative RNA structure prediction approaches. *BMC Bioinformatics*, 5:140, 2004.
- [17] D. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, MA, 1989.
- [18] D. Goldberg and K. Deb. A comparative analysis of selection schemes used in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 69–93. Morgan Kaufmann, San Francisco, CA, 1990.
- [19] D. Goldberg, K. Deb, and J. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6:333–362, 1992.

- [20] R. Gutell. Comparative sequence analysis and the structure of 16S and 23S rRNA. In A. E. Dahlberg and R. A. Zimmermann, editors, *Ribosomal RNA. Structure, Evolution, Processing, and Function in Protein Biosynthesis*, pages 111–128. CRC Press, Boca Raton, 1996.
- [21] R. Gutell, J. Cannone, D. Konings, and D. Gautheret. Predicting U-turns in ribosomal RNA with comparative sequence analysis. *Journal of Molecular Biology*, 300:791–803, 2000.
- [22] R. Gutell, J. Lee, and J. Cannone. The accuracy of ribosomal RNA comparative structure models. *Current Opinion in Structural Biology*, 12(3):301–310, 2002.
- [23] J. Hammersley and P. Clifford. Markov field and finite graphs and lattices. *unpublished*, 1971.
- [24] G. Harik. Linkage learning via probabilistic modeling in the EcGA. Technical Report 99010, IlliGAL, University of Illinois at Urbana-Champaign, 1999.
- [25] G. Harik, E. Cantú-Paz, D. Goldberg, and B. Miller. The gambler’s ruin problem, genetic algorithms, and the sizing of populations. *Evolutionary Computation*, 7(3):231–253, 1999.
- [26] G. Harik, F. Lobo, and D. Goldberg. The compact genetic algorithm. In *Proceedings of the IEEE Conference on Evolutionary Computation*, volume 3, pages 523–528, November 1998.

- [27] W. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57:97–109, 1970.
- [28] D. Heckerman, D. Geiger, and M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.
- [29] I. Hofacker. Vienna RNA secondary structure server. *Nucleic Acids Research*, 31(13):3429–3431, 2003.
- [30] I. Hofacker, M. Fekete, and P. Stadler. Secondary structure prediction for aligned RNA sequences. *Journal of Molecular Biology*, 319:1059–1066, 2002.
- [31] I. Hofacker, W. Fontana, P. Stadler, L. Bonhoeffer, M. Tacker, and P. Schuster. Fast folding and comparison of RNA secondary structures. *Monatshefte f. Chemie*, 125:167–188, 1994.
- [32] J. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. University of Michigan Press, Ann Arbor, MI, 1975.
- [33] E. Ising. Beitrag zur theorie des ferromagnetismus. *Zeitschrift fur Physik*, 31:253–258, 1925.
- [34] F. Jensen. *Bayesian Networks and Decision Graphs*. Springer, 2001.

- [35] S. Kirkpatrick, C. Gelatt, and M. Vecchi. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [36] B. Knudsen and J. Hein. Pfold: RNA secondary structure prediction using stochastic context-free grammars. *Nucleic Acids Research*, 31:3423–3428, 2003.
- [37] Spin Glass Ground State Server. http://www.informatik.uni-koeln.de/lis_juenger/research/spinglass/, University of Köln, Germany, 2004.
- [38] S. Kullback and R. Leibler. On information and sufficiency. *Annals of Mathematical Statistics*, 22:79–86, 1951.
- [39] K. Lau and K. Dill. A lattice statistical mechanics model of the conformational and sequence spaces of proteins. *Macromolecules*, 22:3986–3997, 1989.
- [40] J. Lee, J. Cannone, and R. Gutell. The lonepair triloop: A new motif in RNA structure. *Journal of Molecular Biology*, 325:65–83, 2003.
- [41] J. Lee and R. Gutell. Diversity of base-pair conformations and their occurrence in rRNA structure and RNA structural motifs. *Journal of Molecular Biology*, 344:1225–1249, 2004.
- [42] S. Li. *Markov Random Field Modeling in Image Analysis*. Springer, second edition, 2001.

- [43] F. Liers, M. Jünger, G. Reinelt, and G. Rinaldi. Computing exact ground states of hard Ising spin glass problems by branch-and-cut. In *New Optimization Algorithms in Physics*, pages 47–68. Wiley, 2004.
- [44] F. Liers, M. Palassini, A. Hartmann, and M. Jünger. Ground state of the Bethe-lattice spin glass and running time of an exact optimization algorithm. *Physical Review B*, 68(9):094406, 2003.
- [45] D. Mathews, J. Sabina, M. Zuker, and D. Turner. Expanded sequence dependence of thermodynamic parameters improves prediction of RNA secondary structure. *Journal of Molecular Biology*, 288:911–940, 1999.
- [46] H. Mühlenbein. The equation for response to selection and its use for prediction. *Evolutionary Computation*, 5(3):303–346, 1997.
- [47] H. Mühlenbein and T. Mahnig. The factorized distribution algorithm for additively decomposed functions. In *Proceedings of the Congress on Evolutionary Computation*, pages 752–759, 1999.
- [48] H. Mühlenbein and T. Mahnig. FDA - A scalable evolutionary algorithm for the optimization of additively decomposed functions. *Evolutionary Computation*, 7(4):353–376, 1999.
- [49] H. Mühlenbein and T. Mahnig. Evolutionary computation and beyond. In *Foundations of Real-World Intelligence*, pages 123–186. CLSI Publications, Stanford, CA, 2001.

- [50] H. Mühlenbein and G. Paaß. From recombination of genes to the estimation of distributions I. Binary parameters. In *Proceedings of the 4th International Conference on Parallel Problem Solving from Nature (PPSN IV)*, pages 178–187, London, UK, 1996. Springer-Verlag.
- [51] J. Pearl. *Causality*. Cambridge University Press, 2000.
- [52] M. Pelikan and D. Goldberg. Hierarchical Bayesian optimization algorithm = Bayesian optimization algorithm + niching + local structures. In *Optimization by Building and Using Probabilistic Models (OBUPM) 2001*, pages 217–221, 2001.
- [53] M. Pelikan, D. Goldberg, and E. Cantú-Paz. BOA: The Bayesian optimization algorithm. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)*, volume I, pages 525–532, San Francisco, CA, 1999. Morgan Kaufmann.
- [54] M. Pelikan, A. Hartmann, and K. Sastry. Hierarchical BOA, cluster exact approximation, and Ising spin glasses. Technical Report Missouri Estimation of Distribution Algorithms Laboratory (MEDAL) No. 2006002, University of Missouri in St. Louis, St. Louis, MO, 2006.
- [55] M. Pelikan and H. Mühlenbein. The bivariate marginal distribution algorithm. In *Advances in Soft Computing - Engineering Design and Manufacturing*, pages 521–535. Springer-Verlag, London, UK, 1999.

- [56] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, UK, second edition, 1992.
- [57] J. Prior. Eugenic evolution for combinatorial optimization. Master's thesis, Department of Computer Sciences, The University of Texas at Austin, Austin, TX, 1998.
- [58] J. Rissanen. Modelling by shortest data description. *Automatica*, 14:465–471, 1978.
- [59] H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3:175–184, 1960.
- [60] R. Santana. A Markov network based factorized distribution algorithm for optimization. In *Machine Learning: ECML 2003*, pages 337–348. Springer, 2003.
- [61] R. Santana. Estimation of distribution algorithms with Kikuchi approximations. *Evolutionary Computation*, 13(1):66–97, 2005.
- [62] S. Shakya. *DEUM: A framework for an Estimation of Distribution Algorithm based on Markov Random Fields*. PhD thesis, The Robert Gordon University, Aberdeen, UK, April 2006.
- [63] S. Shakya, J. McCall, and D. Brown. Using a Markov network model in a univariate EDA: An empirical cost-benefit analysis. In *Proceedings of the*

- Genetic and Evolutionary Computation Conference (GECCO)*, volume I, pages 727–734, 2005.
- [64] S. Shakya, J. McCall, and D. Brown. Solving the Ising spin glass problem using a bivariate EDA based on Markov random fields. In *Proceedings of the Congress on Evolutionary Computation*, pages 908–915, 2006.
- [65] G. Syswerda. Simulated crossover in genetic algorithms. In *Foundations of Genetic Algorithms*, pages 239–255. Morgan Kaufmann, San Francisco, CA, 1992.
- [66] *Wikimedia Commons*, http://commons.wikimedia.org/wiki/Image:NA-comparedto-DNA_thymineAndUracilCorrected.png.
- [67] RNA secondary structure prediction. *Wikiomics: Open Bioinformatics*, http://wikiomics.org/wiki/RNA_secondary_structure_prediction.
- [68] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods*. Springer, second edition, 2003.
- [69] F. Yates. Contingency table involving small numbers and the χ^2 test. *Journal of the Royal Statistical Society*, Supplement 1:217–235, 1934.
- [70] K. Yue, K. Fiebig, P. Thomas, H. Chan, E. Shakhnovich, and K. Dill. A test of lattice protein folding algorithms. In *Proceedings of the National Academy of Sciences of the United States of America*, volume 92, pages 325–329, 1995.

- [71] M. Zuker. On finding all suboptimal foldings of an RNA molecule. *Science*, 244:48–52, 1989.
- [72] M. Zuker. Mfold web server for nucleic acid folding and hybridization prediction. *Nucleic Acids Research*, 31(13):3406–3415, 2003.
- [73] M. Zuker and P. Stiegler. Optimal computer folding of large RNA sequences using thermodynamics and auxiliary information. *Nucleic Acids Research*, 9:133–148, 1981.

Vita

Matthew Edward Alden was born in Kirkland, Washington on July 17, 1977, the son of Edward Anthony Alden and Francis Lucille Alden. After graduating from Bothell High School, Bothell, Washington in 1996, he entered the University of Washington in Seattle, Washington. In 2000, he graduated cum laude with a Bachelor of Science degree in Computer Science with College Honors and Applied & Computational Mathematical Sciences. That same year he began his graduate studies in computer science at The University of Texas at Austin.

Permanent address: 18929 89th Ave N.E.
Bothell, WA 98011

This dissertation was typeset with \LaTeX^\dagger by the author.

[†] \LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's \TeX Program.