DISSERTATION

AN ANALYSIS OF COMBINATORIAL SEARCH SPACES FOR A CLASS OF NP-HARD PROBLEMS

Submitted by

Andrew M. Sutton

Department of Computer Science

In partial fulfillment of the requirements

for the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Spring 2011

Doctoral Committee:

Advisor: L. Darrell Whitley
Co-advisor: Adele E. Howe

A. P. Willem Böhm
Edwin K. P. Chong

ABSTRACT

AN ANALYSIS OF COMBINATORIAL SEARCH SPACES FOR A CLASS OF NP-HARD PROBLEMS

Given a finite but very large set of states $\mathcal{X}$ and a real-valued objective function f defined on $\mathcal{X}$, *combinatorial optimization* refers to the problem of finding elements of $\mathcal{X}$ that maximize (or minimize) f. Many combinatorial search algorithms employ some perturbation operator to hill-climb in the *search space*. Such perturbative local search algorithms are state of the art for many classes of NP-hard combinatorial optimization problems such as maximum k-satisfiability, scheduling, and problems of graph theory.

In this thesis we analyze combinatorial search spaces by expanding the objective function into a (sparse) series of basis functions. While most analyses of the distribution of function values in the search space must rely on empirical sampling, the basis function expansion allows us to directly study the distribution of function values across regions of states for combinatorial problems without the need for sampling. We concentrate on objective functions that can be expressed as bounded pseudo-Boolean functions which are NP-hard to solve in general. We use the basis expansion to construct a polynomial-time algorithm for exactly computing constant-degree moments of the objective function f over arbitrarily large regions of the search space. On functions with restricted codomains, these moments are related to the true distribution by a system of linear equations. Given low moments supplied by our algorithm, we construct bounds of the true distribution of f over regions of the space using a linear programming approach. A straightforward relaxation allows us to efficiently approximate the distribution and hence quickly estimate the count of states in a given region that have certain values under the objective function.

The analysis is also useful for characterizing properties of specific combinatorial problems. For instance, by connecting search space analysis to the theory of inapproximability, we prove that the bound specified by Grover's *maximum principle* for the Max-Ek-Lin-2 problem is sharp. Moreover, we use the framework to prove certain configurations are forbidden in regions of the Max-3-Sat search space, supplying the first theoretical confirmation of empirical results by others.

Finally, we show that theoretical results can be used to drive the design of algorithms in a principled manner by using the search space analysis developed in this thesis in algorithmic applications. First, information obtained from our moment retrieving algorithm can be used to direct a hill-climbing search across plateaus in the Max-k-Sat search space. Second, the analysis can be used to control the mutation rate on a (1+1) evolutionary algorithm on bounded pseudo-Boolean functions so that the offspring of each search point is maximized in expectation. For these applications, knowledge of the search space structure supplied by the analysis translates to significant gains in the performance of search.

ACKNOWLEDGEMENTS

I am indebted to my advisor Dr. Darrell Whitley and my co-advisor Dr. Adele Howe. Both have been outstanding mentors and have masterfully guided me and fostered my intellectual growth throughout this project. I would also like to thank the rest of my doctoral committee, Dr. Wim Böhm and Dr. Ed Chong, for the constructive feedback and sharp insight they provided to me while reviewing this document. My thanks also to Dr. Ross McConnell who served on my doctoral committee (until a sabbatical conflict) and contributed to valuable discussions and ideas during my preliminary examination. I also gratefully acknowledge my undergraduate advisor Dr. Anne Cable who in the first place suggested that I pursue my doctorate.

I would also like to acknowledge the support of my fellow Ph.D. students at Colorado State University. I specifically would like to thank my friends and colleagues Monte Lunacek and Mark Roberts with whom I have had many engaging discussions and from whom I have received extensive sound advice and moral support.

I owe a great debt of gratitude to my parents who have continually supported my many endeavors and instilled in me a love for learning and science, my brother and sister for their positive encouragement, and to Amanda for her support and limitless patience during the end years of my Ph.D.

Lastly, I would like to acknowledge the financial support I received from the Discrete Mathematics and Optimization Program of the Air Force Office of Scientific

Research, Air Force Materiel Command, USAF, under grant numbers FA9550-07-1-0403 and FA9550-08-1-0422.

This document was prepared and typeset by the author in LaTeX

To Mom and Dad.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

Combinatorial optimization refers to the problem of locating from a set of discrete structures an element that optimizes some value or cost criterion. For example, suppose $\mathcal{X}$ is a finite but very large set of *states*. Let $f : \mathcal{X} \to \mathbb{R}$ be a real-valued function defined on $\mathcal{X}$. We call f the *objective function* and $\mathcal{X}$ the *state set*.

A specific *instance* of a combinatorial optimization problem is thus a state set taken with a specific objective function $(\mathcal{X}, f)$ [AL03]. The problem is to find a globally maximal (resp., minimal) state, that is, an element $x^* \in \mathcal{X}$ such that $f(x^*) \geq f(x)$ (resp., $f(x^*) \leq f(x)$) for all $x \in \mathcal{X}$. In many cases, the problem of finding a globally optimal state of an instance of combinatorial optimization belongs to the family of NP-hard problems. This family contains computational problems that, unless $\mathsf{P} = \mathsf{NP}$, cannot be solved efficiently in the worst case (i.e., in time that scales as a polynomial in the size of the input).

The general computational approach for solving hard combinatorial problems is the *combinatorial search algorithm*: some prescription of iteratively generating states in $\mathcal{X}$ and evaluating them with respect to the objective function f [HS04]. Combinatorial search algorithms can be partitioned into two broad classes. *Constructive* search algorithms examine the space of partial solutions to iteratively build a solution from component parts. In contrast, *perturbative* or *local* search algorithms employ some kind of transformation (e.g., a *move operator* or *mutation operator*) to perform small perturbations to states in order to incrementally "move" through the state set toward improving solutions. In this thesis, we will focus on methods of perturbative search since it is a universal approach to combinatorial problem solving and is often considered state of the art for many NP-hard problems such as maximum k-satisfiability [PB10], problems of graph theory [AL03, JM04], and scheduling problems [NS96, Wat03, BHWR06].

Both constructive and perturbative search algorithms can be *complete*, that is, they are guaranteed to find an optimal solution if one exists given enough computational resources. However, local search algorithms are generally formulated as incomplete algorithms in which no such guarantee exists. Despite this fact, they have received considerable attention in both theoretical and experimental computer science communities due to the fact that they often empirically converge to high quality solutions within low order polynomial time [Yan03] and, for some problem classes, can quickly solve difficult instances that lie beyond the grasp of conventional complete solvers [GW93b] and sometimes scale better than complete solvers [PW96].

These successes have largely been attributed to the fact that perturbative local search algorithms are somehow exploiting underlying structure in the *search space*: the set of all states along with their relationship to one another and their relationship to the objective function. The specific attributes of this inherent structure in the search space and the causality on the behavior of a local search algorithm are generally not well-understood. Moreover, search algorithms are often designed in an ad-hoc manner and are subsequently developed by making incremental modifications without a clear and scientific understanding of the underlying relationship with the search space.

In this thesis, we describe a formal study of the structure of combinatorial search spaces. We will appeal to the tools of Fourier analysis of finite groups. In the same manner that the Fourier decomposition of an arbitrary continuous function can uncover harmonic structure hidden within complicated signals, decomposing a combinatorial objective function into an alternate basis expansion can also reveal useful information about the underlying search space. We employ this basis function decomposition to study the statistical structure of the allocation of objective function values to states that lie in relevant regions of the search space. We concentrate on a class of NP-hard combinatorial optimization problems (i.e., those whose objective functions are real-valued functions over length-n strings from a binary alphabet) with a special focus on instances of maximum k-satisfiability. Formal search space analyses can improve our understanding of the behavior of algorithms and ultimately effectuate more principled algorithm design: an idea explored in the penultimate chapter.

This thesis makes several contributions. We present a polynomial-time algorithm that computes constant-degree statistical moments of any bounded pseudo-Boolean objective function over arbitrarily large regions of the search space. We employ a linear programming approach to construct bounds for the true distribution of objective function values over regions and subsequently relax the approach to devise an approximation of such distributions. This approximation must share its low moments with the true distribution and can be computed without sampling. We also demonstrate an application for the approximation by showing how it can be used to accurately estimate the number of improving states in any region without resorting to sampling.

We also present analyses for specific combinatorial problems that instantiate bounded pseudo-Boolean objective functions. We make a new connection between search space analysis and results from inapproximability theory to prove that a well-known bound on the quality of local maxima in the Max-Ek-Lin-2 search space is sharp. For problems of satisfiability, we appeal to the basis function decomposition to construct bounds on the quality of local maxima and present new proofs that provide a theoretical confirmation of previous empirical observations made by others.

A fundamental goal of this research is to explore how formal analysis of combinatorial search spaces can provide a foundation for principled algorithm design. Toward that end, we also introduce two algorithmic applications that benefit directly from the framework advanced in this thesis. In one application, we employ our moment calculation algorithm to create a surrogate gradient function that directs a simple hill-climbing search algorithm through plateaus in the maximum k-satisfiability search space. We find that this ultimately translates to faster convergence to near-optimal values of the objective function. In another application, we consider the on-line control of the mutation rate parameter of an evolutionary algorithm on nonlinear functions. We establish how the basis function expansion can be used to compute on-line the expected fitness of an offspring of the evolutionary algorithm at any point in the search space. Moreover, we show that it is always possible to solve for the roots of a polynomial of bounded degree in the mutation rate to find the rate that maximizes the expected fitness of the offspring. We demonstrate that this approach results in a significant improvement over the standard recommended rate of $1/n$ early in search.

1.1 Combinatorial Search Space Analysis

The beginnings of perturbative local search algorithms for combinatorial optimization can perhaps be traced back to work at Los Alamos Scientific Laboratory in

1953 where Metropolis et al. [MRR$^+$53] developed an efficient simulation of physical systems cooling to thermal equilibrium. Years later, a number of researchers [KGV83, Čer85] noticed a deep connection between minimizing the objective function of a combinatorial optimization problem and the cooling of a solid to its low-energy ground state. This led to the well-known *simulated annealing* algorithm [vLA87].

By the mid- to late-1950s, several researchers had devised procedures for solving a conventional graph optimization problem called the *traveling salesman problem* by making perturbative exchanges of state elements (in this case, the edges of a graph) [Flo56, Cro58, Boc58]. To put these results in a historical context, such procedures were often introduced for solving combinatorial problems *by hand*. For example, Croes [Cro58] mentions in the concluding section of his paper that the procedure could be automated by a computer with sufficient storage capacity. This perturbative approach has since evolved into many high-performance computer algorithms for treating hard combinatorial optimization problems such as propositional satisfiability [SLM92, SK93, SKC94, SKC96, TH03, PB10], the traveling salesman problem [Lin65, LK73, JM97], the quadratic assignment problem [MF97, MF00], the linear ordering problem [SS03], the vertex cover problem [RHG07, Wit09], the maximal clique problem [PH06], graph bipartitioning [FA86, KS96], scheduling problems [WBWH02, WBHW03, BHWR06] and many others [HS04].

Combinatorial search processes are pervasive in nature. For example, the progression of a physical system through a set of discrete states that seeks to minimize system energy and the evolution of biological structures through adaptation and natural selection are both natural analogues of the processes in which we are interested in this research. Indeed, rigorous analyses of such “natural” search spaces comes from theoretical biology with the so-called *fitness landscape* model [Wri32, EMS88, Kau93, FSBB$^+$93], and from condensed matter physics with the study of disordered magnets [EA75, SK75]. In these cases, analyses focus on identifying certain structural features of fitness landscapes or potential energy surfaces and the dynamics of processes that explore the state set.

Many researchers have since realized the connection between the study of such "natural" search spaces and the study of "synthetic" search spaces of computer algorithms [KT85, FA86, And88, Wei90, SS92, Sta95, RS02]. These connections have led to important developments in the study of combinatorial search spaces. Perhaps one of the most prominent structural characteristics that affects the dynamics of processes exploring the search space is the concept of *ruggedness* or dependence of objective function value on state change [KL87, Wei90]. This concept of ruggedness is treated mathematically with the *autocorrelation coefficient* which can be estimated by random walks. In many cases, it can also be computed exactly using analytical approaches [Sta96, AZ98, AZ00, AZ01, SWH09].

Loosely speaking, higher ruggedness results in more *local optima*, or states that are extremal in their neighborhood. It is generally understood that perturbative local search algorithms are affected negatively by the presence of a large number of local optima since the optima must be *escaped* in order to make progress. Several analyses have concentrated on characterizing the count and distribution of optima [KL87], as well as exploring the mathematical relationship between ruggedness and local optima [SS92, Sta95, GPS97]. The difficulty of search processes escaping local optima depends on the structure of the search space "near" the extremal point and the accessibility of improving states. This structure is coarsely captured by the rigorous concept of *depth* which has been introduced in the theory of simulated annealing. The depth of a local optimum is the minimum disimproving change in the objective function value that must be accepted to escape the optimum. It has been used to prove the existence of an optimal cooling schedule for annealing algorithms (and hence prove their completeness) [Haj88] and has been related to their rate of convergence [TSY88, TSY89]. The study of depth also has implications for general computational complexity since, as Kern has conjectured, characterizing the depth of a combinatorial optimization problem is exactly as hard as solving it [Ker93]. Sharp bounds on depth (and a related concept called *width*) have been derived for certain combinatorial problems such as the $(0, 1)$-knapsack problem and set covering [Rya95].

Related to the concept of depth is the *basin*: the set of all mutually reachable states that lie above (respectively, below) a particular objective function value. The structure of basins and their influence on gradient walks was studied by Flamm et al. [FFHS00] who developed the concept of a *barrier tree* that describes the height of barriers between locally optimal solutions. Barrier trees were initially developed in the context of studying the folding kinetics of RNA sequences, but have since been applied to the search spaces of combinatorial optimization [FFS00, FHSW02, FHSS07].

A related but distinct concept to ruggedness is *neutrality*: the count of neighboring positions in the search space that share an objective function value. Neutrality has been handled mathematically by considering a fitness landscape as an element of an appropriate probability space [RS01] and treating the quantity statistically (a similar treatment was given to ruggedness by Stadler and Happel [SH99]).

High neutrality is a necessary condition for a qualitative search space feature called a *plateau*: a maximal set of mutually reachable states whose image under the objective function is a single value. Neutrality and plateaus arise in a number of common combinatorial search spaces [Hor97, FCS97, Bar98, BBK$^+$00, Smy04, BVCE06]. In order to describe plateaus and study their properties, Hoos and

Stützle [HS04] propose constructing *plateau connection graphs* which are similar to Markov models of a random hill-climbing process in which each state corresponds to a plateau. On k-satisfiability problems in particular, Smyth [Smy04] studied summary statistics for plateau connection graphs (such as vertex degree and depth) on random and structured instances of the propositional satisfiability problem. Furthermore, he experimentally examined the relative frequency of a number of graph theoretic features of plateaus themselves such as size, branching factor, and diameter. All such analyses must employ extensive sampling (or in some cases, exhaustive enumeration of particular small instances) to construct the underlying empirical models.

Various investigations have considered the classification of qualitative features in the search space such as the "plateau taxonomy" of Frank et al. [FCS97] which partitions states into the unique type of plateau to which they belong. Closely related to this taxonomy is the search space *position types* of Hoos and Stützle [HS04] which we will consider in more detail in Chapter 3. Analyses that examine such qualitative features study the relative frequency of occurrence of features and again are based on empirical sampling of search space instances to estimate this frequency distribution.

The relationship between combinatorial problem *hardness* and search space structure has also been studied empirically. In the context of propositional satisfiability, Clark et al. [CFG$^+$96], and later Hoos [Hoo98], studied the relationship between the number of optimal solutions and empirical search cost. The hardness of uniformly generated random constraint satisfaction problems for both local and complete search has been related to the concept of the *phase transition* [MZK$^+$99]: a dependence of the solution character of problem instances on constrainedness. Several researchers have attempted to explore the apparent link between dependency of search cost on constrainedness by empirically studying the distribution of unary prime implicates (sometimes referred to as *backbones*) [PW96, SGS00]. However, in the case of local search and the satisfiability phase transition, the picture is often obfuscated by the common practice of filtering unsatisfiable instances, which is likely to produce unaccounted effects.

For scheduling problems, empirical models that relate search space features to local search runtime were studied extensively by Watson [Wat03] and Watson et al. [WBHW03]. Empirical search space models have proven useful for informing the design of specialized local search for scheduling problems such as the attenuated leap heuristic of Barbulescu et al. [BHWR06] which responds to plateaus and the iterated jump-and-redescend heuristic of Watson et al. [WHW03] which addresses the weakness of attractor basins.

One issue with empirical models is that while they can be richly descriptive, especially for particular applications, they can also be difficult to generalize. In this thesis we will employ formal theoretical tools to make general statements about the entire class of bounded pseudo-Boolean functions (which includes NP-hard problems such as maximum k-satisfiability, NK-landscapes, and the maximum cut problem). Furthermore, we remark here that the analyses contained within can be easily generalized to bounded functions over strings of higher cardinality alphabets.

1.2 Organization

In the next chapter we will construct the foundational framework for the remainder of the thesis in terms of basis function expansions of the objective function. While doing so, we prove the sharpness of bounds on local optima for a particular combinatorial problem. We then focus the discussion on the Fourier analysis of pseudo-Boolean functions which is analogous to the well-known Walsh analysis in the theory of evolutionary computation. In Chapter 3 we concentrate on the search space structure of the maximum 3-satisfiability problem and prove theorems regarding certain forbidden structure.

In Chapter 4 we introduce a tight connection between the basis function expansion and the *moments* of the objective function over regions of the search space for bounded pseudo-Boolean functions and present an efficient algorithm for computing moments. We then relate these moments to the true distribution of values in the image of regions of the search space under the objective function in Chapter 5. In Chapter 6 we use the theoretical framework developed in this thesis to inform principled algorithm design in two algorithmic applications. Finally, in Chapter 7 we summarize the thesis and discuss avenues of future work.

Chapter 2

Expressing Functions in Terms of Neighborhood Graphs

Universal, non-specialized combinatorial search algorithms explore the set of states $\mathcal{X}$ making decisions based on the objective function $f : \mathcal{X} \to \mathbb{R}$. In the case of *local search* algorithms, a neighborhood operator is employed that maps states into each other and thus structures the search of $\mathcal{X}$. The performance of local search algorithms depends on the morphology of the search space which ultimately arises from the relationship between the objective function f and the neighborhood operator. The central focus of this chapter is to mathematically study this relationship by decomposing the objective function into a *basis expansion* that relates it directly to the neighborhood operator. Specifically, we re-express the objective function in terms of the Fourier series of the graph induced by the underlying neighborhood operator. The study of objective functions by expressing them in the Fourier series of highly symmetric graphs was introduced by Peter Stadler [Sta95] and has since been employed for studying various combinatorial optimization problems [KS96, RKHS02, RS02].

In the case of real functions over binary strings, the Fourier series expansion is identical to the *Walsh* basis expansion. The expression of such functions in their Walsh basis expansion has been studied extensively in theoretical work on genetic algorithms since it breaks down the function into components that are pertinent to algorithms that perform implicit hyperplane sampling (i.e., population-based genetic algorithms employing recombination operators) [Gol89, LV91, Gol92, HW97, RHW98, Hec99, HW99, Hec02, HW04]. In this thesis we show that the Walsh basis expansion can also be useful for studying algorithms that perform *local sampling*, such as local search and mutation-only evolutionary algorithms. This appears to be the first application of the Walsh decomposition for studying local search.

For some combinatorial problems, the objective function has a very sparse representation in the basis expansion that relates it to the neighborhood operator. In these cases (the so-called *elementary landscapes*), the *maximum principle* introduced by Grover [Gro92] imposes a bound on the quality of local optima in the space. We will prove that a problem called MAX-Ek-LIN-2 which requires finding quasi-solutions to an inconsistent linear system over a finite field possesses this property. We present an interesting connection between elementary landscapes and inapproximability results that allows us to prove that the bound imposed by the maximum principle is sharp for MAX-Ek-LIN-2.

This chapter introduces formal concepts and lays the groundwork for the remainder of the thesis. In later chapters we will characterize the statistics of the objective function over regions of the search space partitioned with respect to the neighborhood operator. Such an analysis is possible because the set of basis functions into which we decompose the objective function will be in some sense “well-behaved” over the regions in question.

2.1 Preliminaries

We begin by making some preliminary observations about the space of functions on $\mathcal{X}$. For convenience, notation and concepts are compiled in the appendix provided on page 160. The set of all possible objective functions on a state set $\mathcal{X}$

$$\mathscr{F}(\mathcal{X}) = \{f : \mathcal{X} \to \mathbb{R}\}$$

forms a vector space isomorphic to $\mathbb{R}^{|\mathcal{X}|}$. Furthermore, $\mathscr{F}(\mathcal{X})$ is an inner product space with the scalar product

$$\langle f, g \rangle = \sum_{x \in \mathcal{X}} f(x) g(x),$$

for $f, g \in \mathscr{F}(\mathcal{X})$. If we associate with each state $z \in \mathcal{X}$ a standard basis function

$$e_z(x) = [x = z],^{1}$$

then $\{e_z\}$ forms the "standard basis" of $\mathscr{F}(\mathcal{X})$ and

$$f(x) = \langle e_x, f \rangle.$$

Consider any linear operator $\boldsymbol{M} : \mathscr{F}(\mathcal{X}) \to \mathscr{F}(\mathcal{X})$. Such an operator is a *function endomorphism* in the sense that $\boldsymbol{M}f \in \mathscr{F}(\mathcal{X})$ where $\boldsymbol{M}f$ denotes $\boldsymbol{M}$

[1] Throughout this thesis, we will employ the *Iverson bracket* notation [Ive62, Knu92] to denote an indicator function on statements that can be true or false. For a such a statement s,

$$[s] = \begin{cases} 1 & \text{if } s \text{ is true,} \\ 0 & \text{otherwise.} \end{cases}$$

applied to $f \in \mathscr{F}(\mathcal{X})$.

$$\begin{aligned}\boldsymbol{M}f(x) &= \langle e_x, \boldsymbol{M}f\rangle \\ &= \sum_{y\in\mathcal{X}} \langle e_x, \boldsymbol{M}e_y\rangle\langle e_y, f\rangle \\ &= \sum_{y\in\mathcal{X}} \langle e_x, \boldsymbol{M}e_y\rangle f(y). \end{aligned} \tag{2.1}$$

Given a function $f \in \mathscr{F}(\mathcal{X})$ we say f is an *eigenfunction* of a linear operator $\boldsymbol{M}$ if and only if

$$\boldsymbol{M}f(x) = \lambda f(x),$$

for some scalar λ and all $x \in \mathcal{X}$.

Local search algorithms and many evolutionary algorithms operate by moving through the state set by performing minor perturbations on current states to construct similar "neighboring" states. Thus with each state $x \in \mathcal{X}$ we associate a set $N(x) \subseteq \mathcal{X}$ which comprises the neighboring states of x. This neighborhood operator imposes a connectivity on the underlying state set. This concept is formalized by the idea of a *neighborhood graph*: a graph whose vertex set is $\mathcal{X}$ and (possibly directed) edges connecting x to y if and only if $y \in N(x)$.

The structure of this neighborhood graph is determined by adjacency operator $\boldsymbol{A} : \mathscr{F}(\mathcal{X}) \to \mathscr{F}(\mathcal{X})$ defined as

$$\langle e_x, \boldsymbol{A}e_y\rangle = \begin{cases} 1 & \text{if } y \in N(x), \\ 0 & \text{if } y \notin N(x). \end{cases} \tag{2.2}$$

The relationship between an objective function f and its neighborhood graph can be studied by considering $\boldsymbol{A}$ as an endomorphism on real functions over $\mathcal{X}$. The image of f under $\boldsymbol{A}$ is a function $\boldsymbol{A}f : \mathcal{X} \to \mathbb{R}$ that gives the sum of f evaluated over the neighbors of x. This is captured by the following lemma.

Lemma 2.1. *Let $f \in \mathscr{F}(\mathcal{X})$ and $\boldsymbol{A}$ be the adjacency operator of a neighborhood N. The function $\boldsymbol{A}f$, i.e., the image of f under the linear map $\boldsymbol{A}$, evaluates to*

$$\boldsymbol{A}f(x) = \sum_{y \in N(x)} f(y).$$

Proof. By definition we have

$$\begin{aligned}
\boldsymbol{A}f(x) &= \langle e_x, \boldsymbol{A}f \rangle \\
&= \sum_{y \in \mathcal{X}} \langle e_x, \boldsymbol{A}e_y \rangle f(y) && \text{by (2.1),} \\
&= \sum_{y \in N(x)} f(y) && \text{by (2.2).}
\end{aligned}$$

□

2.2 Alternative basis expansions

Reidys and Stadler [RS02] point out that it is often useful to write elements from $\mathscr{F}(\mathcal{X})$ in alternative bases. We can learn more about the structure of the search space if an appropriate choice of basis functions is used. Let $\{\varphi_i\}$ be a set of basis functions that span $\mathscr{F}(\mathcal{X})$. Then

$$f(x) = \sum_i a_i \varphi_i(x),$$

where a_i is a scalar. Furthermore, consider a linear map $\boldsymbol{M}$ applied to f. By the linearity of $\boldsymbol{M}$,

$$\boldsymbol{M}f = \sum_i a_i \boldsymbol{M}\varphi_i.$$

Throughout this thesis, the set of basis functions we choose will be eigenfunctions of an appropriate linear operator. This means the quantity $\boldsymbol{M}\varphi_i(x)$ is efficiently computable given the value of $\varphi_i(x)$ and the corresponding eigenvalue. This is especially useful when f has a *sparse* representation in the basis, that is,

$$|\{a_i : a_i \neq 0\}| \ll |\mathcal{X}|.$$

Furthermore, if the basis functions are *instance independent*, this gives a natural separation of an objective function into components that are instance dependent (i.e., the coefficients) and the components that are instance independent (i.e., the basis functions).

For example, let $f \in \mathscr{F}(\mathcal{X})$ and $X \subseteq \mathcal{X}$. The *expectation*, or arithmetic mean value of f over X is written as $\langle f \rangle_X$ and defined to be the expectation of a random variable that gives the value of f evaluated at a state sampled uniformly at random from X. Since the probability any particular state $y \in X$ is sampled is equal to $\frac{1}{|X|}$, we have

$$\langle f \rangle_X = \frac{1}{|X|} \sum_{y \in X} f(y).$$

Given a functional basis $\{\varphi_i\}$ for any objective function f, we can immediately see that

$$\langle f \rangle_X = \sum_i a_i \langle \varphi_i \rangle_X . \tag{2.3}$$

Therefore, we have the following result, which we will exploit in later chapters.

Remark 2.1. *Given some basis expansion for f, the problem of finding the expectation of f over a set of states X reduces to the problem of finding the expectation of the basis functions over X.*

This is especially useful when the basis functions in the expansion of f depend only on the adjacency defined by N. We explore this now.

2.2.1 The relationship between f and N

As a local search algorithm explores a combinatorial space, it must rely on a "signal" that arises from the relationship between the objective function f and the neighborhood operator N, or more precisely, the neighborhood graph induced by N. A strong relationship between f and the neighborhood graph induced by

N supports local search algorithms since they make progress toward states with improving f values by examining states constructed by N. Our goal is to study this relationship in detail.

2.2.1.1 The adjacency spectrum

When the neighborhood operator satisfies some common constraints, we can derive a number of useful results.

Lemma 2.2. *If N is symmetric, that is,*

$$y \in N(x) \iff x \in N(y),$$

then the adjacency operator $\boldsymbol{A}$ corresponding to the neighborhood graph of N is self-adjoint*. This means*

$$\langle \boldsymbol{A}f, g \rangle = \langle f, \boldsymbol{A}g \rangle.$$

Proof. By the definitions,

$$\begin{aligned} \langle \boldsymbol{A}f, g \rangle &= \sum_{x \in \mathcal{X}} \boldsymbol{A}f(x) g(x) \\ &= \sum_{x \in \mathcal{X}} g(x) \sum_{y \in N(x)} f(y). \end{aligned}$$

But since $y \in N(x) \iff x \in N(y)$,

$$\begin{aligned} &= \sum_{y \in \mathcal{X}} f(y) \sum_{x \in N(y)} g(x) \\ &= \langle f, \boldsymbol{A}g \rangle. \end{aligned}$$

□

Since $\boldsymbol{A}$ is self-adjoint, the finite dimensional spectral theorem (see e.g., [Hal63]) guarantees that $\boldsymbol{A}$ has an orthogonal basis $\{\varphi_0, \ldots, \varphi_{|\mathcal{X}|-1}\}$ such that

$$\langle \varphi_i, \varphi_j \rangle = [i = j],$$

and $\boldsymbol{A}\varphi_i = \lambda_i \varphi_i$. This simply means that φ_i is an eigenfunction of the adjacency operator corresponding to eigenvalue λ_i.

Therefore, when N is symmetric, a natural way to study the relationship between f and N is by writing f as an expansion in the orthogonal eigenbasis $\{\varphi_i\}$ of the adjacency operator of the neighborhood graph induced by N

$$f(x) = \sum_{i=0}^{|X|-1} a_i \varphi_i(x).$$

We impose the following ordering on the eigenvalues of $\boldsymbol{A}$

$$\lambda_0 \geq \lambda_1 \geq \ldots \geq \lambda_{|\mathcal{X}|-1}. \tag{2.4}$$

Lemma 2.3. *If N is regular with degree d, that is, for all $x \in \mathcal{X}$, $|N(x)| = d$, then the function*

$$\varphi_0(x) = 1$$

is an eigenfunction of the adjacency operator of the neighborhood graph of N corresponding to eigenvalue $\lambda_0 = d$.

Proof. Choose an arbitrary eigenvalue λ_i of $\boldsymbol{A}$ and let $\varphi_i \in \mathscr{F}(\mathcal{X})$ be the corresponding eigenfunction. Furthermore, let

$$x^* \in \arg\max_{x \in \mathcal{X}} |\varphi_i(x)|$$

be a global maximum of φ_i. Then we have,

$$\begin{aligned}
|\lambda_i||\varphi_i(x^*)| &= |\lambda_i \varphi_i(x^*)| \\
&= |\boldsymbol{A}\varphi_i(x^*)| \\
&= \left| \sum_{y \in N(x^*)} \varphi_i(y) \right| && \text{by Lemma 2.1,} \\
&\leq |N(x^*)||\varphi_i(x^*)|.
\end{aligned}$$

Since $|N(x^*)| = d$ we thus have

$$\begin{aligned} |\lambda_i||\varphi_i(x^*)| &\leq d|\varphi_i(x^*)|, && \text{and so,} \\ |\lambda_i| &\leq d. && (2.5) \end{aligned}$$

Since we chose λ_i arbitrarily, d is an upper bound on the eigenvalues of $\boldsymbol{A}$. Now let

$$\varphi^{const}(x) = 1$$

be the constant function. Then

$$\begin{aligned} \boldsymbol{A}\varphi^{const}(x) &= \sum_{y \in N(x)} \varphi^{const}(y) && \text{by Lemma 2.1,} \\ &= |N(x)| = d = d\varphi^{const}(x). \end{aligned}$$

So d is an eigenvalue of $\boldsymbol{A}$. Due to the order imposed on the eigenvalues in (2.4), λ_0 is maximal and the bound in (2.5) gives us

$$\lambda_0 = d,$$

and the corresponding eigenfunction is

$$\varphi_0(x) = \varphi^{const}(x) = 1. \qquad \square$$

2.2.1.2 Elementary landscapes

Grover [Gro92] discovered that in many well-studied combinatorial problems with natural neighborhood operators, the objective function is up to an additive constant an eigenfunction of the adjacency operator

$$f(x) = a_0 + a_k\varphi_k(x), \tag{2.6}$$

for some adjacency eigenfunction φ_k. In all cases Grover studied, the neighborhoods are connected and symmetric; so by Lemma 2.3, this can be written as the following trivial basis expansion:

$$f(x) = a_0\varphi_0(x) + a_k\varphi_k(x). \tag{2.7}$$

Since φ_k is an eigenfunction of the adjacency, we can write

$$\boldsymbol{A}f(x) = a_0 d\varphi_0(x) + \lambda_k a_k\varphi_k(x). \tag{2.8}$$

This is a version of a linear difference equation that is typically called Grover's *wave equation* due to its similarity to the wave equation of mathematical physics [CM92, BDD03]. Typically, this equation is stated in terms of the *combinatorial Laplacian* [Gro92, CM92, Sta96, BC01, RS02, BLS07] which is defined for d-regular graphs as $\boldsymbol{L} = d\boldsymbol{I} - \boldsymbol{A}$. In this thesis, however, we will always work with the adjacency operator.

We can write Equation (2.8) in terms of the expectation operator over the neighborhood. This will become useful in Chapter 3 where it will play a role in a probabilistic argument in proofs about forbidden (local) structure in certain search spaces. It is also a special case of the moment constructions we will perform in Chapter 4.

Due to Lemma 2.1 we can write the expectation over the neighborhood as

$$\langle f\rangle_{N(x)} = \frac{1}{d}\boldsymbol{A}f(x).$$

This allows us to write Equation (2.8) in terms of the expectation operator over the neighborhood

$$\begin{aligned}\langle f\rangle_{N(x)} &= a_0\varphi_0(x) + \frac{\lambda_k}{d}a_k\varphi_k(x)\\ &= a_0 + \frac{\lambda_k}{d}\left(f(x) - a_0\right) && \text{by (2.6),}\\ &= \frac{d-\lambda}{d}a_0 + \lambda f(x). && (2.9)\end{aligned}$$

When Equation (2.9) holds, we can immediately infer that there must be a direct relationship between the elements of $(\mathcal{X}, N, f)$. Stadler called these structures *elementary landscapes* [Sta95], the term "landscape" coming from theoretical biology.

Despite the apparent restrictiveness of Equation (2.8), a large number of combinatorial problems along with their natural neighborhood operators have been shown to be elementary. For example, Grover [Gro92] proved that Equation (2.6) holds for graph coloring and not-all-equal satisfiability under the corresponding Hamming neighborhoods, as well as min-cut graph partitioning and weight partitioning under their natural neighborhood operators. The symmetric Traveling Salesman Problem (TSP) under the 2-opt and the 3-exchange neighborhoods [CM92] and the 2-exchange neighborhood [Gro92], the antisymmetric TSP under the 2-opt and 2-exchange neighborhoods [Sta96], the weakly symmetric TSP [SBDA03], and variants of the multiple-TSP [CB00] have also all been shown to satisfy the wave equation.

The general interest in elementary landscapes has flourished in recent years because a number of useful properties can easily be derived from Equation (2.6). For instance, Grover [Gro92] showed that all elementary landscapes obey what is sometimes called the *maximum principle* [BLS07]:

$$f(\hat{x}_{\min}) \leq \bar{f} \leq f(\hat{x}_{\max}),$$

where $\hat{x}_{\min}$ and $\hat{x}_{\max}$ are respectively arbitrary local minima and local maxima of f and $\bar{f} = \langle f \rangle_{\mathcal{X}}$ is the average value of f over $\mathcal{X}$. In other words, there are no local minima (resp. maxima) with higher (resp. lower) than average objective value. We will revisit this maximum principle again in Section 2.3.1 and show that the bound for local maxima is sharp for a particular combinatorial problem.

The elementary property also has broad implications for the statistics of random walks through the search space. Weinberger [Wei90] proposed that different landscapes might be characterized by their random-walk autocorrelation: a time-series autocorrelation of values of $f(x)$ sampled along a random walk on the adjacency induced by N. Stadler [Sta96] showed that the random-walk autocorrelation function decays exponentially if and only if the landscape is elementary. Dimova et al. [DBP05] asserted that an exponentially decaying autocorrelation function is a characteristic of an AR(1) stochastic process. They use this result to show that a landscape is elementary if and only if the time series generated by random walk is consistent with an AR(1) process.

It has also been conjectured (e.g., in [Sta95]) that the autocorrelation properties (specifically, the correlation length) that can be easily derived for elementary landscapes are somehow related to the count of local optima in the search space. This count would be a useful quantity for predicting how hard a particular problem class or instance is for local search algorithms to solve.

2.3 An example: MAX-Ek-LIN-2

We now study a very simple basis decomposition of a combinatorial problem. We will also prove that the decomposition satisfies Equation (2.9) and is therefore an instance of an elementary landscape, as introduced in the previous section. This section thus contributes the first proof that the elementary property holds for this particular problem; however it is very similar to the problem of finding the ground state of a p-spin glass which has been shown to be elementary [dOFS99, RS02]. We will also illuminate an interesting connection to inapproximability that allows us to make assertions about the sharpness of Grover's maximum principle.

Let $\mathbb{Z}_2$ denote the finite field of integers modulo 2. MAX-Ek-LIN-2 is a combinatorial optimization problem in which, given a potentially inconsistent system of linear equations over $\mathbb{Z}_2$, we are interested in finding a "quasi-solution" to the system that maximizes the number of equations that are consistent. Aside from being interesting from a theoretical perspective, the general problem of finding consistent equations of a linear system modulo p also has practical applications in factoring large prime numbers (e.g., for breaking RSA encryption) [HM08] and linear cryptanalysis. In the latter application, given a *cipher* which maps plaintext bits and key bits to ciphertext bits, the objective is to discover linear relationships between the plaintext, key bits, and the ciphertext bits to analyze the cipher. This can be modeled as a set of linear equations over $\mathbb{Z}_2$.

Suppose we have a set of m linear equations of the following form

$$\begin{array}{ccccccccc} z_{11}x_1 & + & z_{12}x_2 & + & \dots & + & z_{1n}x_n & = & b_1, \\ z_{21}x_1 & + & z_{22}x_2 & + & \dots & + & z_{2n}x_n & = & b_2, \\ \vdots & & \vdots & & & & \vdots & & \vdots \\ z_{m1}x_1 & + & z_{m2}x_2 & + & \dots & + & z_{mn}x_n & = & b_m, \end{array}$$

where

- $z_{ij}, x_i, b_i \in \mathbb{Z}_2$.
- There are *exactly* $k \geq 3$ nonzero coefficients z_{ij} in the i^{th} equation.

Put another way, we have the following linear system over $\mathbb{Z}_2$

$$\boldsymbol{Z}\boldsymbol{x} = \boldsymbol{b},$$

where $\boldsymbol{Z} \in \mathbb{Z}_2^{n \times n}$, and $\boldsymbol{b}, \boldsymbol{x} \in \mathbb{Z}_2^n$, and exactly k nonzero entries appear in each row of $\boldsymbol{Z}$. The problem of determining the consistency of this linear system, that is, finding if there exists an $\boldsymbol{x}$ which simultaneously satisfies all m equations, is called Ek-Lin-2.

The Gaussian elimination algorithm for solving systems of linear equations is well-defined over finite fields, so we can apply this procedure to solve for $\boldsymbol{x}$:

$$\boldsymbol{x} = \boldsymbol{Z}^{-1}\boldsymbol{b}$$

in time polynomial in the input size. If the system is *inconsistent*, that is, there is no such solution $\boldsymbol{x}$, the Gaussian elimination algorithm easily detects this by halting with a degenerate system. Thus the decision problem Ek-Lin-2 is in P.

Suppose we are instead interested in finding the quasi-solution $\boldsymbol{x}$ that gives the *maximum* number of consistent equations. In other words, we want to find the largest feasible subsystem of $\boldsymbol{Zx} = \boldsymbol{b}$. This rather straightforward maximization variant, which is called Max-Ek-Lin-2 is NP-hard.

If the system is overdetermined, Gaussian elimination will find some $q > 0$ equations which are inconsistent. In this case, q is dependent on the order in which equations are considered during the elimination procedure. Thus the problem is transformed into finding the order of equations that minimizes q.

A Max-Ek-Lin-2 system of m equations in n unknowns can be solved in $\tilde{O}(2^{\epsilon n})$ time ($\epsilon > 0$) or approximated in polynomial time (to a factor of $\frac{1}{2} + \frac{\epsilon n}{12m}$) using a hybrid heuristic-selection algorithm [VWW06]. Suppose instead we apply the following *local search* algorithm. Start with an initial set $\boldsymbol{x}^{(0)}$ of decision variables for the $\boldsymbol{Zx} = \boldsymbol{b}$ system generated uniformly at random. While stopping criteria are not met, repeat the following.

1. Let S be the set of all states that can be obtained by adding 1 to a single decision variable in $\boldsymbol{x}^{(i)}$.

2. Choose $\boldsymbol{y}$ to be the element in S that has the maximal number of consistent equations in $\boldsymbol{Zy} = \boldsymbol{b}$ (ties broken arbitrarily).

3. If $\boldsymbol{y}$ admits fewer inconsistencies than $\boldsymbol{x}^{(i)}$, then $\boldsymbol{x}^{(i+1)} \leftarrow \boldsymbol{y}$. Otherwise $\boldsymbol{x}^{(i+1)} \leftarrow \boldsymbol{x}^{(i)}$.

We would like to perform an analysis of the search space of this algorithm. The state set $\mathcal{X}$ in this case is the set of all decision variable vectors in $\mathbb{Z}_2^n$. The objective function is the function

$$\begin{aligned} &f : \mathbb{Z}_2^n \mapsto \{0, 1, \dots, m\}, \text{ where} \\ &f(\boldsymbol{x}) = \text{the number of consistent equations of } \boldsymbol{Zx} = \boldsymbol{b}. \end{aligned} \tag{2.10}$$

that represents the count $f(\boldsymbol{x})$ of the number of consistent equations of $\boldsymbol{Zx} = \boldsymbol{b}$. The resulting combinatorial optimization problem is thus $(\mathbb{Z}_2^n, f)$ where f is given by (2.10).

The neighborhood operator used in the above local search algorithm takes a vector in $\mathbb{Z}_2^n$ into a set of vectors that differ by one element from its input. This is the well-known Hamming operator which we will be working with throughout this thesis. In the case of $\mathbb{Z}_2^n$, the induced neighborhood graph is isomorphic to Q_n: the hypercube graph of order n.

Let χ be the indicator function

$$\chi(\boldsymbol{x}, j) = \begin{cases} 1 & \text{if equation } j \text{ is consistent under } \boldsymbol{x}, \\ 0 & \text{if equation } j \text{ is inconsistent under } \boldsymbol{x}. \end{cases}$$

Hence we can write (2.10) as the sum of indicator functions

$$f(\boldsymbol{x}) = \sum_{j=1}^{m} \chi(\boldsymbol{x}, j). \tag{2.11}$$

A Hamming move, by definition, changes the state of exactly one decision variable. Note that an equation that was consistent under $\boldsymbol{x}$ becomes inconsistent only when the state of one of its decision variables with a nonzero coefficient changes. A similar argument holds for inconsistent equations.

Denote as $\Delta(i, j)$ the *change in consistency* of equation j when the state of x_i is changed, i.e.,

$$\Delta(i, j) = \begin{cases} 1 & \text{if equation } j \text{ becomes consistent when 1 is added to } \boldsymbol{x}_i, \\ -1 & \text{if equation } j \text{ becomes inconsistent when 1 is added to } \boldsymbol{x}_i, \\ 0 & \text{if equation } j \text{ is unaffected when 1 is added to } \boldsymbol{x}_i. \end{cases}$$

Thus the sum objective function values evaluated over the neighborhood of $\boldsymbol{x}$ is the value of $f(x)$, plus the gains in consistency, minus the losses in consistency.

This yields the following identity:

$$\begin{aligned}\sum_{\boldsymbol{y}\in N(\boldsymbol{x})} f(\boldsymbol{y}) &= \sum_{\boldsymbol{y}\in N(\boldsymbol{x})} \left(f(\boldsymbol{x}) + \sum_{j=1}^{m} \Delta(i,j) \right) \\ &= nf(\boldsymbol{x}) + \sum_{i=1}^{n}\sum_{j=1}^{m} \Delta(i,j). \qquad (2.12)\end{aligned}$$

Since each equation has exactly k nonzero coefficients, there are exactly k out of the n possible Hamming moves that change its consistency. In other words, for a given equation j, we have the following result.

$$\sum_{i=1}^{n} \Delta(i,j) = \begin{cases} -k & \text{if } j \text{ is consistent,} \\ +k & \text{if } j \text{ is inconsistent.} \end{cases}$$

This can be rewritten in terms of the indicator function

$$\sum_{i=1}^{n} \Delta(i,j) = k - 2k\chi(\boldsymbol{x}, j), \qquad (2.13)$$

and summing over all m equations,

$$\begin{aligned}\sum_{i=1}^{n}\sum_{j=1}^{m} \Delta(i,j) &= \sum_{j=1}^{m}\sum_{i=1}^{n} \Delta(i,j) \\ &= \sum_{j=1}^{m} (k - 2k\chi(\boldsymbol{x}, j)) && \text{by (2.13),} \\ &= mk - 2k\sum_{j=1}^{m} \chi(\boldsymbol{x}, j) \\ &= mk - 2kf(\boldsymbol{x}) && \text{by (2.11),} \\ &= 2k\left(\frac{m}{2} - f(\boldsymbol{x})\right).\end{aligned}$$

Substituting this result into the corresponding term of (2.12) produces

$$\begin{aligned}\sum_{\boldsymbol{y}\in N(\boldsymbol{x})} f(\boldsymbol{y}) &= nf(\boldsymbol{x}) + 2k\left(\frac{m}{2} - f(\boldsymbol{x})\right) \\ &= 2k\frac{m}{2} + (n-2k)f(\boldsymbol{x}). \qquad (2.14)\end{aligned}$$

Dividing by the neighborhood size n we recover Equation (2.9) with $a_0 = m/2$, $d = n$, and $\lambda = n - 2k$, we get

$$\langle f \rangle_{N(\boldsymbol{x})} = \frac{2k}{n}\frac{m}{2} + \frac{n-2k}{n} f(\boldsymbol{x}). \tag{2.15}$$

So the Max-Ek-Lin-2 combinatorial optimization problem under local search by Hamming moves is a so-called elementary landscape. In other words, the objective function f, as defined in (2.10) is (up to an additive constant) an eigenfunction of the adjacency operator $\boldsymbol{A}$ of the Hamming neighborhood graph.

2.3.1 The maximum principle is sharp for Max-Ek-Lin-2

We now show an interesting connection between the maximum principle of Grover [Gro92] and work by Håstad [Hås01] on approximability. A *local maximum* is a state x such that for all $y \in N(x)$, $f(y) \leq f(x)$. Recall from the discussion of elementary landscapes in Section 2.2.1.2 that Grover showed if the objective function and neighborhood obeyed Equation (2.9), then it must be the case that all local maxima (minima) lie above (below) the mean objective function value over the entire state set.

In the case of Max-Ek-Lin-2, this bound is sharp, as we now show. We point out that the average objective function value of Max-Ek-Lin-2 is equal to $m/2$. This is a simple proof using the symmetry of the neighborhood operator and Equation (2.15) and we omit it here.

Lemma 2.4. *For any instance of* Max-Ek-Lin-2*, a local maximum can be found in polynomial time.*

Proof. A simple local search algorithm suffices. Starting from an arbitrary state, move to a neighboring state that has strictly improving value. If no such neighbor exists, the current state is already a local maximum. The number of strictly improving moves from any arbitrary state bounded by m. □

We can also place a lower bound on the objective function value of local maxima in the Max-Ek-Lin-2 search space.

Lemma 2.5. *In the* Max-Ek-Lin-2 *search space, all local maxima are greater than or equal to* $\frac{m}{2}$.

Proof. This is simply a restatement of *Grover's maximum principle* [Gro92]. Let $\hat{\boldsymbol{x}}$ be a local maximum. Thus we have,

$$\begin{aligned} \langle f \rangle_{N(\hat{\boldsymbol{x}})} &\leq f(\hat{\boldsymbol{x}}) \\ \frac{2k}{n}\frac{m}{2} + \frac{n-2k}{n} f(\hat{\boldsymbol{x}}) &\leq f(\hat{\boldsymbol{x}}) && \text{by (2.15)}, \\ f(\hat{\boldsymbol{x}}) &\geq \frac{m}{2}. \end{aligned}$$

□

Grover's maximum principle, restated in Lemma 2.5, gives a lower bound on the objective function evaluation of all local maxima for the Max-Ek-Lin-2 problem. However, it is not immediately clear that the bound might be sharp. We now appeal to a result from algorithmic complexity theory to prove that it is indeed sharp.

Theorem 2.1. *For* Max-Ek-Lin-2*, local search can always find a state* $\hat{\boldsymbol{x}}$ *with* $f(\hat{\boldsymbol{x}}) \geq \frac{m}{2}$ *in polynomial time.*

Proof. This follows immediately from Lemmas 2.4 and 2.5. □

A maximization problem can be *approximated in polynomial time* within a factor of ρ if there is a polynomial-time algorithm that always (correctly) produces a solution to the problem with objective value at least $f(\boldsymbol{x}^*)/\rho$ where $f(\boldsymbol{x}^*)$ is globally maximum.

Theorem 2.2. *For any* $\epsilon > 0$, $k \geq 3$, *it is* NP*-hard to approximate* Max-Ek-Lin-2 *within a factor of* $2 - \epsilon$.

Proof. In [Hås01, Theorem 5.5]. □

Corollary. *Unless* $\mathsf{P} = \mathsf{NP}$*, no polynomial-time algorithm exists that can always find a solution* $\hat{\boldsymbol{x}}$ *with* $f(\hat{\boldsymbol{x}}) \geq \frac{m}{2-\epsilon}$ *for any* $\epsilon > 0$*.*

Proof. This follows directly from Theorem 2.2. Such an algorithm must find a solution $\hat{\boldsymbol{x}}$ with

$$f(\hat{\boldsymbol{x}}) \geq \frac{m}{2-\epsilon} \geq \frac{f(\boldsymbol{x}^*)}{2-\epsilon}.$$

□

Since, by Theorem 2.1, it is possible to find a local optimum in polynomial time, this means unless $\mathsf{P} = \mathsf{NP}$, local optima can become *arbitrarily close* to $\frac{m}{2}$. Given any $\epsilon > 0$, there must always exist some instance of Max-Ek-Lin-2 that has a local optimum $\hat{\boldsymbol{x}}$ with $\frac{m}{2} \leq f(\hat{\boldsymbol{x}}) \leq \frac{m}{2-\epsilon}$ or local search could could always approximate the solution within a factor of $2 - \epsilon$ in polynomial time, due to Lemma 2.5. It immediately follows from this that, for Max-Ek-Lin-2, $\frac{m}{2}$ is a sharp lower bound on the quality of local maxima.

2.4 Sparse representations

We found that the objective function of the combinatorial problem introduced above was (up to an additive constant) an eigenfunction of the search space adjacency operator. We would now like to try to generalize the allowable complexity of this series expansion somewhat. Consider the neighborhood graph on a set of states $\mathcal{X}$ induced by a operator N. If N is symmetric, we know from Lemma 2.2 and the finite dimensional spectral theorem that the adjacency operator $\boldsymbol{A}$ corresponding to N has an orthogonal basis $\{\varphi_0, \ldots, \varphi_{|\mathcal{X}|-1}\}$ of eigenfunctions. This basis spans $\mathscr{F}(\mathcal{X})$ and we can write *any* function $f : \mathcal{X} \to \mathbb{R}$ as

$$f(x) = \sum_{i=0}^{|\mathcal{X}|-1} a_i \varphi_i(x), \tag{2.16}$$

where a_i is a real-valued coefficient.

It is not immediately clear why this particular basis expansion may be useful. In fact, in the general case, the series in (2.16) has $|\mathcal{X}|$ terms, a quantity we have already supposed is intractably large. However, we will see in the remainder of this thesis that many important combinatorial optimization problems have a *sparse* representation in this basis which means that all except $O(1)$ coefficients a_i vanish.

For instance, the trivial function $f(x) = 0$ might be considered as having a "maximally sparse" decomposition since it can be represented in the alternate basis with all zero coefficients $a_i = 0$ for all $i = 0, \ldots, |\mathcal{X}| - 1$. In the previous sections, we discussed (and gave an example) of combinatorial problems whose objective functions had representations in an adjacency basis $\{\varphi_o, \ldots, \varphi_{|\mathcal{X}|-1}\}$ that were somehow maximally sparse while remaining interesting, that is, those in the form of Equation (2.6): in which a_i is only nonzero at the constant function φ_0 and at another single eigenfunction φ_k. Such sparse decompositions make up the so-called elementary landscapes of Stadler [Sta95].

In the rest of this thesis we will concentrate on the more general case where the objective function can be expressed sparsely in the eigenbasis of a natural adjacency, but with $k > 1$ further nonzero coefficients where k is $O(1)$. We will be then able to generalize Equation (2.9) to perform analyses of certain search spaces.

The amenability of search spaces to analysis that employs this basis decomposition approach depends on the fact that the state set $\mathcal{X}$ admits a neighborhood operator N that is symmetric and regular (i.e., the underlying neighborhood graph is a regular, undirected graph). As we have seen above, in this case, the adjacency operator is self-adjoint. Barnes et al. [BDD03] have also discussed generalizing such an analysis to non-regular, asymmetric operators.

Many different combinatorial problems yield different state sets and hence different “natural” neighborhood graphs. Stadler [Sta95] presents various graphs that represent neighborhood graphs of different combinatorial search spaces. For example, in the case of scheduling and permutation problems, the neighborhood graph is a Cayley graph of the symmetric group generated by transpositions or inversions. In the case of bipartitioning problems, the neighborhood graph is the Johnson graph $\mathcal{J}(n, n/2)$ for even n.

For the remainder of this thesis we will concentrate exclusively on the family of combinatorial optimization problems whose objective functions are defined over *Hamming space*, or $\{0,1\}^n$, i.e., the set of strings of length n over a binary alphabet. We will pay close attention to a subset of this family: problems of maximum k-satisfiability.

2.4.1 Pseudo-Boolean functions and Hamming space

We begin by introducing some basic concepts for working with the state set $\{0,1\}^n$. Let $x \in \{0,1\}^n$. Denote the b^{th} element of x as

$$x[b] \in \{0,1\}.$$

Throughout our work in Hamming space, we will often implicitly take advantage of the isomorphism between $\{0,1\}^n$ and the set of integers $\{0,1,\ldots,2^n-1\}$. In particular, we identify each $x \in \{0,1\}^n$ with an integer $a \in \{0,1,\ldots,2^n-1\}$ as follows

$$x \mapsto a; \quad a = \sum_{b=1}^{n} \left(2^{b-1} \times x[b]\right),$$

i.e., $x[1]$ corresponds to the “least significant bit” of the string x.

The most natural neighborhood for $\{0,1\}^n$ is produced by the Hamming neighborhood operator. Given $x \in \{0,1\}^n$, the Hamming neighborhood operator N is defined as

$$N(x) = \Big\{y \in \{0,1\}^n : |\{x[b] \neq y[b]\}| = 1\Big\},$$

for $b = 1, \ldots, n$. In the context of local search (especially when applied to satisfiability problems), this neighborhood is often called the “flip” neighborhood [GW93a] since it consists of the set of all strings derived by “flipping” a bit. In the context of evolutionary computation, the Hamming neighborhood is generated by single point mutations on a binary chromosome.

The search space “closeness” of two binary strings x and y is thus captured by the minimum number of Hamming neighborhood operations required to transform x into y. Of course this gives rise to a natural metric. Given $x, y \in \{0,1\}^n$, the *Hamming distance* between x and y is defined as

$$\mathcal{H}(x,y) = |\{b : x[b] \neq y[b]\}| = \langle x \oplus y, x \oplus y \rangle,$$

where $\oplus$ denotes component-wise exclusive-or. The set $\{0,1\}^n$ taken with the function $\mathcal{H}$ forms a *metric space*. This is exactly the graph theoretic distance between two vertices in the neighborhood graph on $\{0,1\}^n$ induced by the Hamming operator.

Definition 2.1. *Let $x, y \in \{0,1\}^n$. The (string) inner product of x and y is a binary operator*

$$\langle \cdot, \cdot \rangle : \{0,1\}^n \times \{0,1\}^n \to \mathbb{N},$$

defined as

$$\langle x, y \rangle = \sum_{b=1}^{n} x[b]y[b].$$

By this definition, given $x \in \{0,1\}^n$, the quantity $\langle x, x \rangle$ can be interpreted as the *number of nonzero bits* in x. We will often refer to this quantity as the *order* of x.

A *Boolean* function is simply a function over $\{0,1\}^n$ into $\{0,1\}$. When we relax the codomain to the real numbers, we refer to the function as a pseudo-Boolean function.

Definition 2.2. *A pseudo-Boolean function is a function*

$$f : \{0,1\}^n \to \mathbb{R}$$

that takes binary strings (also called bitstrings) of length n *to the real numbers.*

2.4.2 Bounded pseudo-Boolean functions

The simplest pseudo-Boolean functions are *separable* in which the function can be written as a linear sum of subfunctions depending on each bit:

$$f(x) = \sum_{b=1}^{n} h(x[b]),$$

where $h : \{0,1\} \to \mathbb{R}$. Clearly, this function can be optimized in $\Theta(n)$ time since each subfunction can be optimized separately in constant time.

Various search algorithms have been proved to have polynomial complexity on separable pseudo-Boolean functions such as the $(1+1)$ evolutionary algorithm [DJW98], the $(\mu{+}1)$ evolutionary algorithm [Wit06], randomized local search without [GKS99] and with [SY11] memory, and simulated annealing [JW07]. Pseudo-Boolean functions become hard to optimize when they are no longer additively separable. For example, the class of functions of the form

$$f(x) = \sum_{\{b,b'\} \subset \{1,\ldots,n\}} h(x[b], x[b']),$$

where $h : \{0,1\}^2 \to \mathbb{R}$ contains the NP-hard maximum 2-satisfiability problem as a special case. Of course, there are subclasses of this class that can be solved efficiently, for instance pseudo-Boolean polynomials of degree 2 with non-negative coefficients [WW05].

More generally, the objective functions to a large number of well-studied combinatorial problems can be expressed as a sum of subfunctions that depend on at most k input bits where k is a constant with respect to the input size. This family of *bounded* pseudo-Boolean functions is pervasive in many applications. In molecular biology and biophysics for example, bounded pseudo-Boolean functions are often employed to model the evolution of a population of organisms [FL70, KL87, MP89]. In NK-landscape models [Kau93], for instance, the fitness of a genotype (a string over a binary alphabet) is computed as a sum over individual k-ary gene interactions. NK-landscapes have also been employed to simulate landscapes that arise from RNA folding [FSBB$^+$93].

Bounded pseudo-Boolean functions also play an important role in theoretical computer science. The problem of maximizing a k-bounded pseudo-Boolean function is NP-hard, even when $k = 2$ since it is at least as hard as the maximum 2-satisfiability (Max-2-Sat) problem [GJS76]. In general, the objective function for any maximum k-satisfiability (Max-k-Sat) problem can be expressed as a k-bounded pseudo-Boolean function. In subsequent chapters, we will explore Max-k-Sat problems more deeply using the framework introduced here.

We now formally introduce the bounded pseudo-Boolean functions. To do so, we must first introduce the *pack* function of Heckendorn [Hec99]. Note that we can also think of $\{0,1\}^n$ as a vector space over the finite field $\{0,1\}$ which is closed over multiplication and addition modulo 2. This allows us to make a formal algebraic characterization of Heckendorn's pack function.

Definition 2.3. *The Heckendorn Pack Function is defined as*

$$\mathscr{P} : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}^k,$$

where $k \leq n$ such that

$$\mathscr{P}(x, z) = x\boldsymbol{Z},$$

where $\boldsymbol{Z}$ is an $n \times \langle z, z\rangle$ matrix over the finite field $\{0,1\}$ given by

$$\boldsymbol{Z}_{ij} = z[i]\delta_{\langle z, 2^i - 1\rangle, j}.$$

Here, δ is the Kronecker delta function.

Note here that the string inner product $\langle z, 2^i - 1\rangle$ gives the number of nonzero entries from 1 to i. Thus

$$\delta_{\langle z,2^i-1\rangle,j} = \begin{cases} 1 & \text{if there are } j \text{ nonzero entries from 1 to } i, \\ 0 & \text{otherwise.} \end{cases}$$

So $\boldsymbol{Z}_{ij}$ is equal to 1 if and only if i is the j^{th} position of z that is nonzero. Let $y = \mathscr{P}(x, z) = x\boldsymbol{Z}$. Clearly, y is a bitstring of length $\langle z, z\rangle$. The b^{th} element of y is given by

$$y[b] = \sum_i x[i]\boldsymbol{Z}_{ib},$$

and is simply the element in x corresponding to the position with the b^{th} nonzero entry in z. The intuitive meaning of the Heckendorn Pack Function function is that $\mathscr{P}(x, z)$ selects the bits in x and "masks" them with the bitmask given by z and returns a bitstring of length $\langle z, z\rangle$ containing the masked out bits. For example,

$$\mathscr{P}((1, 0, 1, 0, 1), (0, 1, 1, 0, 1)) = (0, 1, 1).$$

Definition 2.4. *A k-bounded pseudo-Boolean function is a pseudo-Boolean function that can be expressed as a sum of subfunctions that each depend on at most k bits, i.e.,*

$$f(x) = \sum_{i=0}^{k} \sum_{z:\langle z,z\rangle=i} g_z\left(\mathscr{P}(x, z)\right).$$

where $g_z : \{0, 1\}^{\langle z,z\rangle} \to \mathbb{R}$.

Each subfunction g_z depends on $\langle z, z\rangle = i$ bits. We define inclusion notation on bitstrings as follows. Given two bitstrings of length n $x, y \in \{0, 1\}^n$, we write

$$x \subseteq y \iff x[b] = 1 \implies y[b] = 1,$$

for all $1 \leq b \leq n$.

2.4.3 Fourier (Walsh) series expansion

Recall from Section 2.2 that we can infer properties of search space structure by studying a given function over $\mathcal{X}$ in an alternative basis given by a suitable set of functions that span the function space $\mathscr{F}(\mathcal{X})$.

A convenient alternative basis for discrete functions comes from the theory of *discrete Fourier analysis* which has existed since at least the eighteenth century [HJB84]. In this case, the basis functions are sine and cosine functions of different frequencies. The discrete Fourier series expansion is the projection of an arbitrary discrete function onto the orthogonal set of sines and cosines. This can be generalized into n dimensions as follows. Let Σ_q denote a finite alphabet of cardinality q. Suppose we are interested in functions over length-n strings from Σ_q. The set of such strings Σ_q^n can be associated with the direct n-product of the additive group of integers modulo q

$$(\mathbb{Z}/q\mathbb{Z})^n = \underbrace{\mathbb{Z}/q\mathbb{Z} \times \mathbb{Z}/q\mathbb{Z} \times \cdots \times \mathbb{Z}/q\mathbb{Z}}_{n},$$

which is a finite Abelian group. We can define the complex trigonometric function

$$\phi_a(x) = \cos\left(\frac{2\pi\langle x, a\rangle}{q}\right) + \sqrt{-1}\sin\left(\frac{2\pi\langle x, a\rangle}{q}\right),$$

which can be expressed as an exponential function (i.e., as a root of unity),

$$= \exp\left(\frac{2\pi\sqrt{-1}\langle x, a\rangle}{q}\right), \tag{2.17}$$

where $x, a \in (\mathbb{Z}/q\mathbb{Z})^n$ and $\langle x, a\rangle$ denotes the corresponding string inner product. Here ϕ_a maps $(\mathbb{Z}/q\mathbb{Z})^n$ to the unit circle. We can write any function $f : (\mathbb{Z}/q\mathbb{Z})^n \to \mathbb{R}$ in its *Fourier series expansion* as

$$f(x) = \sum_{i \in (\mathbb{Z}/q\mathbb{Z})^n} a_i \phi_i(x), \tag{2.18}$$

where

$$a_i = |(\mathbb{Z}/q\mathbb{Z})^n|^{-1} \sum_{x \in (\mathbb{Z}/q\mathbb{Z})^n} f(x)\overline{\phi_i(x)}. \tag{2.19}$$

The overline denotes complex conjugation. The Fourier series expansion can be generalized to complex functions of arbitrary finite groups (see e.g., [Ter99]).

In the case of pseudo-Boolean functions, the Fourier series expansion is often better known as the *Walsh series* expansion. Walsh analysis has been studied extensively in theoretical work on genetic algorithms [Gol89, LV91, Gol92, HW97, RHW98, VW98a, VW98b, Hec99, HW99, Hec02, HW04] because of its usefulness in characterizing *epistasis* or bitwise interaction of fitness functions. Epistasis is a critical component in the analysis of the behavior of genetic algorithms in Hamming space. Moreover, the coefficients of the expansion have been related to the statistics of hyperplanes [Hec02] which are pertinent to algorithms that perform implicit hyperplane sampling (such as genetic algorithms employing recombination).

Our interest in the Walsh expansion is somewhat different. We would instead like to use the expansion to say something about algorithms that employ local neighborhood operators. We will show (in Lemma 2.7 below) that the functions in the Walsh basis expansion are eigenfunctions of the local neighborhood adjacency, thus directly relating the Walsh expansion to the alternative basis expansions introduced in Section 2.2. In later chapters, we will generalize this further in order to characterize the distribution of objective function values over regions of Hamming space.

Joseph L. Walsh [Wal23] introduced the set of orthogonal *Walsh functions* that form a complete orthogonal basis of $\mathscr{F}(\{0,1\}^n)$. Thus any pseudo-Boolean function can be written and analyzed in the Walsh basis. Walsh analysis was first introduced to the evolutionary computation and search community by Holland and Bethke [Hol75, Bet80] and later developed by Goldberg [Gol89] and used primarily for modeling bitwise nonlinearities in fitness functions and analyzing *deception* and *disruption* in the context of genetic search.

The domain of a pseudo-Boolean function, $\{0,1\}^n$, corresponds to the finite Abelian group $(\mathbb{Z}/2\mathbb{Z})^n$. The Walsh functions are the corresponding special cases of the exponential functions ϕ_a in Equation (2.17) and are defined as follows.

$$\psi_i(x) = \exp\left(\pi\sqrt{-1}\langle x, i\rangle\right).$$

Taking advantage of the isomorphism between $\{0,1\}^n$ and $\{0,\ldots,2^n-1\}$, we refer to ψ_i as the i^{th} Walsh function which can be written more succinctly as a real-valued function:

$$\psi_i(x) = \exp\left(\pi\sqrt{-1}\right)^{\langle x,i\rangle} = (-1)^{\langle x,i\rangle}. \tag{2.20}$$

The *order* of the i^{th} Walsh function is $\langle i, i\rangle$, that is, the number of ones in the length-n binary string representation of i.

The Walsh basis is *functionally complete* over $\{0,1\}^n$ [Wal23], that is, any arbitrary pseudo-Boolean function $f : \{0,1\}^n \to \mathbb{R}$ can be written as a linear combination of at most 2^n orthogonal Walsh functions

$$f(x) = \sum_{i=0}^{2^n-1} w_i \psi_i(x), \tag{2.21}$$

where w_i is a scalar called the i^{th} *Walsh coefficient*. Note that this is simply a special case of the Fourier expansion in Equation (2.18).

In Equation (2.21), it is possible to retrieve the value of any Walsh coefficient using the following inversion

$$w_i = \frac{1}{2^n} \sum_{x=0}^{2^n-1} f(x)\psi_i(x). \tag{2.22}$$

This identity is a straightforward specialization of (2.19) and is proved, for example, by Heckendorn [Hec99, Theorem 12].

The following Walsh function identity follows directly from the *pack/unpack equivalency* proved by Heckendorn (i.e., a special case of Corollary 28a to Theorem 28 in [Hec99] or [HW97, Theorem 1]). We state it here for convenience since it will be used in later chapters.

Lemma 2.6. *Let z be an arbitrary bitstring of length n. For any $i \subseteq z$,*

$$\psi_{\mathscr{P}(i,z)}(\mathscr{P}(x,z)) = \psi_i(x).$$

Proof. We have the following identity

$$\begin{aligned}\langle \mathscr{P}(i,z), \mathscr{P}(x,z)\rangle &= \sum_{b=1}^{\langle z,z\rangle} \mathscr{P}(i,z)[b]\mathscr{P}(x,z)[b] \\ &= \sum_{b=1}^{n} i[b]x[b] = \langle i,x\rangle \qquad\qquad \text{since } i \subseteq z.\end{aligned}$$

Thus,

$$\begin{aligned}\psi_{\mathscr{P}(i,z)}(\mathscr{P}(x,z)) &= (-1)^{\langle \mathscr{P}(i,z),\mathscr{P}(x,z)\rangle} \\ &= (-1)^{\langle i,x\rangle} \qquad\qquad \text{by the above identity,} \\ &= \psi_i(x).\end{aligned}$$

□

Formula (2.21) describes a basis expansion for f in the manner of those in Section 2.2. As mentioned above, this basis expansion has recently been useful in the analysis of hyperplane sampling algorithms. We now show that the Walsh basis is also useful for characterizing the relationship between pseudo-Boolean objective functions and the Hamming operator. In particular, we prove that each Walsh function is an eigenfunction of the Hamming neighborhood adjacency.

Lemma 2.7. *Let $\boldsymbol{A}$ be the adjacency operator corresponding to the Hamming neighborhood N. The i^{th} Walsh function is an eigenfunction of $\boldsymbol{A}$:*

$$\boldsymbol{A}\psi_i = (n - 2\langle i,i\rangle)\psi_i.$$

Proof. Let $x \in \{0,1\}^n$ be arbitrary. We have

$$\begin{aligned}
\boldsymbol{A}\psi_i(x) &= \sum_{y \in N(x)} \psi_i(y) && \text{by Lemma 2.1,} \\
&= \sum_{y \in N(x)} (-1)^{\langle i,y \rangle}.
\end{aligned}$$

For each $y \in N(x)$, because x and y differ by a single bit, there exists a unique $0 \leq a \leq 2^n - 1$ for which $x \oplus y = 2^a$. Thus we can make the following case distinction. Let $\wedge$ denote componentwise conjunction in the binary representation. If $i \wedge 2^a = 0$ (i.e., $\langle i, (x \oplus y) \rangle = 0$) then $\langle i, y \rangle = \langle i, x \rangle$ and $\psi_i(y) = \psi_i(x)$. On the other hand, if $i \wedge 2^a = 2^a$ then $|\langle i, y \rangle - \langle i, x \rangle| = 1$ and $(-1)^{\langle i,y \rangle} = -(-1)^{\langle i,x \rangle}$, or equivalently, $\psi_i(y) = -\psi_i(x)$.

Since each Hamming neighbor differs from x in each of the n possible bit positions, there are $n - \langle i, i \rangle$ elements y of $N(x)$ that satisfy the first condition and $\langle i, i \rangle$ that satisfy the second. Hence

$$\begin{aligned}
\sum_{y \in N(x)} \psi_i(y) &= \left((n - \langle i, i \rangle) \, \psi_i(x) - \langle i, i \rangle \psi_i(x) \right) \\
&= (n - 2\langle i, i \rangle) \, \psi_i(x).
\end{aligned}$$

Since we chose x arbitrarily, the property holds for any basis function e_x and we have the general equation

$$\boldsymbol{A}\psi_i = (n - 2\langle i, i \rangle) \, \psi_i,$$

and ψ_i is an eigenfunction of $\boldsymbol{A}$. □

2.4.4 Walsh representation sparsity for bounded functions

Note that we can group each term in (2.21) by its order

$$f(x) = \sum_{p=0}^{n} \Psi_p(x), \tag{2.23}$$

where Ψ_p is defined as

$$\Psi_p(x) = \sum_{i:\langle i,i\rangle = p} w_i \psi_i(x). \tag{2.24}$$

Hence Ψ_p is a linear combination of Walsh functions of order p. In other words, Ψ_p is a component eigenfunction of f that lies in the eigenspace of $\boldsymbol{A}$ corresponding to eigenvalue $n - 2p$. Since there are $\binom{n}{p}$ orthogonal Walsh functions of a given order p, Ψ_p contains at most $\binom{n}{p}$ terms.

We now prove some simple bounds on the order of non-zero Walsh coefficients which will become useful in later chapters for bounding the complexity of computing Walsh coefficients. The following lemma is a slight generalization of the Expansion Theorem of Heckendorn and Whitley [HW97, Theorem 3]. It has also been observed informally in a number of works [HRW98, RHW98, Hec02, KP01].

Lemma 2.8. *Let f be a k-bounded pseudo-Boolean function on $\{0,1\}^n$. For any length-n binary string i,*

$$w_i \neq 0 \implies \langle i,i\rangle \leq k,$$

where w_i is the i^{th} coefficient in the decomposition of f.

Proof. Since f is k-bounded it can be expressed as a sum of subfunctions g_z that each depend on at most k bits.

Denote as $w_i^{(g_z)}$ the i^{th} Walsh coefficient for the subfunction g_z. Since the Walsh transform is linear, the i^{th} Walsh coefficient of f is the sum of the i^{th} Walsh coefficients of the subfunctions, i.e.,

$$w_i = \sum_j w_i^{(g_z)}.$$

Since any g_z depends on at most k bits, if $\langle i,i\rangle > k$ then $\forall g_z, w_i^{(g_z)} = 0$. Thus $\langle i,i\rangle > k \implies w_i = 0$ which gives the contrapositive. □

The following theorem follows directly from the above lemma.

Theorem 2.3 (Decomposition Theorem)**.** *Every k-bounded pseudo-Boolean function f can be written as a linear combination of $k+1$ eigenfunctions of $\boldsymbol{A}$.*

Proof. We can write f in the Walsh representation

$$f(x) = \sum_i w_i \psi_i(x).$$

By the contraposition of Lemma 2.8, w_i is zero for all $\langle i, i \rangle > k$ so we may write

$$\begin{aligned} f(x) &= \sum_{i : \langle i,i \rangle \leq k} w_i \psi_i(x) \\ &= \sum_{p=0}^{k} \Psi_p(x), \end{aligned}$$

where Ψ_p is defined as in (2.24). Each Ψ_p is a linear combination of at most $\binom{n}{p}$ Walsh functions of order p and is thus an eigenfunction of $\boldsymbol{A}$ corresponding to eigenvalue $n - 2p$. □

The sparse representation of k-bounded pseudo-Boolean functions in the Walsh basis supports the tractable computation of certain statistical quantities of the objective function. In Chapter 4, we will explicitly appeal to this sparsity property to construct an efficient algorithm for computing exact objective function statistics over regions of the search space.

In the next chapter we will show the theory developed here is immediately useful for proving bounds on certain structures in the Max-3-Sat search space. In subsequent chapters we will focus on bounded pseudo-Boolean functions over Hamming landscapes and show how this basis expansion can be exploited to provide information about the distribution of function values over regions of the search space.

Chapter 3

Forbidden Structure in the MAX-3-SAT Search Space

We now appeal to the basis function expansion presented in the previous chapter to construct proofs that provide a theoretical confirmation of previous empirical observations by other researchers on problems of maximum 3-satisfiability.[2] Maximum k-satisfiability (MAX-k-SAT) is a generalization of the propositional satisfiability problem [Coo71] in which the objective is to find an assignment to variables that appear in a Boolean formula comprised of clauses of length at most k, such that the cardinality of the set of clauses satisfied under the assignment is maximized.

Though MAX-k-SAT is NP-hard, it can be approximated efficiently by a number of polynomial-time approximation algorithms. Johnson [Joh74] proposed the first such algorithm which guaranteed in polynomial time to produce a state that produced at least 1/2 the maximum number of satisfied clauses for any instance. The approximation bound for Johnson's algorithm was later improved to 2/3 by Chen and Friesen [CFZ97]. Many polynomial-time approximation algorithms for

[2] The work presented in this chapter was initially published in the proceedings of the Second International Workshop on Engineering Stochastic Local Search algorithms [SHW09].

Max-k-Sat have since been proposed, the most recent of which [AW02] guarantees a solution within 0.7846 of the optimal solution. This bound can be improved to 0.8331 supposing the correctness of a conjecture by Zwick [Zwi99].

For the specific case of $k = 3$, Karloff and Zwick [KZ97] gave a polynomial-time algorithm based on semidefinite programming that produces an assignment that is guaranteed to be at least 7/8 of the optimal solution. In the case of $k = 2$, Feige and Goemans proposed an efficient algorithm that constructs an assignment guaranteed to be at least 0.931 of the optimal solution. However, there are theoretical limits to the approximability in these cases. Håstad [Hås01] showed that unless $\mathsf{P} = \mathsf{NP}$, no approximation algorithm can exist that guarantees a solution within $7/8 + \epsilon$ of the optimal of any Max-3-Sat instance and $21/22 + \epsilon$ of the optimal of any Max-2-Sat instance for any constant $\epsilon > 0$. In the case of Max-3-Sat, since a random assignment satisfies any clause with probability 7/8, such an assignment will satisfy in expectation 7/8 of the optimal solution. The implication of the result of Håstad is that Max-3-Sat is polynomial-time inapproximable beyond the expectation of a randomly generated state.

Local search algorithms have also been studied for their approximability. Hansen and Jaumard [Han90] proved that a local optimum in the Max-k-Sat search space is guaranteed to be a solution within 1/2 of the optimum. Mastrolilli and Gambardella [MG05] improved this to 2/3 for a restricted case. In general, local search algorithms have been extremely popular for Max-k-Sat problems. This is due to the fact that, despite the asymptotic worst-case bounds on the Max-k-Sat problem class, they appear to perform empirically well on average. They have been shown to quickly solve difficult problems that lie beyond the grasp of conventional complete solvers [GW93b] and have been found to exhibit superior scaling behavior on soluble problems at the phase transition [PW96].

In this chapter we will use the basis function expansion developed in Chapter 2 to show that two well-studied structural features are forbidden in certain regions of the MAX-3-SAT search space. These two features, *local maxima* and *plateaus*, directly affect the performance of local search algorithms on the MAX-k-SAT problem [FCS97, Smy04, HS04]. We use the basis function expansion developed in this thesis to prove bounds on the location and characteristics of local maxima and plateaus. We show local maxima are forbidden below a certain objective function value. This result gives an instance-dependent threshold that guarantees all local maxima in the MAX-k-SAT search space must have an objective function value that lies strictly above the threshold.

Plateaus are regions of the search space consisting of states that are interconnected by a neighborhood operator and share an objective function value. Hoos and Stützle [HS04] define the *width* of a plateau P: the minimal length path between any state in P and one not in P. For many k-SAT instances, empirical results suggest that plateaus of width greater than one do not exist, or are at least very rare [Hoo98, HS04]. We prove there are regions of the search space that *cannot* contain plateaus of width greater than 1 and show empirically that these regions encompass the majority of the range of the objective function value. To our knowledge, there are no analytical results on the existence (or non-existence) of plateaus of particular width.

3.1 MAX-k-SAT

A Boolean variable is a variable that takes one of two values. Without loss of generality we will say a Boolean variable $v \in \{0, 1\}$. If v is a Boolean variable, we say v and $\neg v$ are *literals*. In this case $\neg v$ denotes the negation of v: the operation that takes v to its complementary value in $\{0, 1\}$. A *clause* is a set of literals.

An instance of MAX-k-SAT consists of a set

$$\mathfrak{V} = \{v_1, v_2, \ldots, v_n\}$$

of n Boolean variables and a collection

$$\mathfrak{C} = \{C_1, C_2, \ldots, C_m\}$$

of clauses on $\mathfrak{V}$ where $|C_j| \leq k$.

An *assignment* is a function $\mathcal{A} : \mathfrak{V} \to \{0,1\}$. A clause is *satisfied* under an assignment when at least one of its constituent literals evaluate to 1. A solution to the MAX-k-SAT problem is an assignment that maximizes the number of satisfied clauses.

There are 2^n unique assignments of $\mathfrak{V}$. The set of all assignments are thus isomorphic to $\{0,1\}^n$ and we can represent each assignment $\mathcal{A}$ on $\mathfrak{V}$ as a unique string $x \in \{0,1\}^n$ where $x[b]$ takes on the value of variable v_b in the assignment. In what follows, we will simply refer directly to strings in $\{0,1\}^n$ as assignments, implicitly applying the bijection.

The objective function $f : \{0,1\}^n \to \{0, \ldots, m\}$ maps an assignment represented by a length-n binary string to the number of clauses satisfied under that assignment.

$$f(x) = |\{C_j \in \mathfrak{C} : C_j \text{ is satisfied under } x\}|. \tag{3.1}$$

The isomorphism allows us to apply the theory of pseudo-Boolean functions developed in Chapter 2.

3.1.1 Basis expansion of the MAX-k-SAT objective function

Note that we can rewrite Equation (3.1) as a sum over indicator functions

$$f(x) = \sum_{j=1}^{m} g_j(x), \tag{3.2}$$

where $g_j(x) = \big[C_j \text{ is satisfied under } x\big]$. We can see that the Max-k-Sat objective function can be expressed as a sum over m subfunctions where each subfunction depends on at most k variables, hence it is k-bounded. Recall that any pseudo-Boolean function $f : \{0,1\}^n \to \mathbb{R}$ has a Walsh basis expansion

$$f(x) = \sum_i w_i \psi_i(x).$$

For Max-k-Sat, it is straightforward to compute the Walsh coefficients using the identity in Equation (2.22). In this section, we will restrict ourselves to the case where $|C_j| = k$ for all $1 \leq j \leq m$. It is trivial to generalize this to the case where $|C_j| \leq k$ and we refer the reader to the work of Rana et al. [RHW98].

Let f be the objective function for a Max-k-Sat instance over a set $\mathfrak{V}$ of n variables and a set $\mathfrak{C}$ of m clauses. We would like to compute the i^{th} Walsh coefficient of f. By Equation (2.22), we have

$$\begin{aligned} w_i &= \frac{1}{2^n}\sum_{x=0}^{2^n-1} f(x)\psi_i(x) &&= \frac{1}{2^n}\sum_{x=0}^{2^n-1}\sum_{j=1}^{m} g_j(x)\psi_i(x) \\ &= \sum_{j=1}^{m}\left(\frac{1}{2^n}\sum_{x=0}^{2^n-1} g_j(x)\psi_i(x)\right) &&= \sum_{j=1}^{m} w_i^{(C_j)}, \end{aligned}$$

where $w_i^{(C_j)}$ denotes the contribution to the i^{th} Walsh coefficient from clause C_j. Thus it is enough to compute the Walsh coefficient contributions from each clause.

The following Lemma was first proved for Max-k-Sat by Rana et al. [RHW98, Theorem 1] and later discussed in several other papers in a more general form [HRW98, HRW99, Hec02]. We restate and prove it here so that it may be expressed in the notational conventions adapted in this thesis.

Lemma 3.1. *Let $C_j \in \mathfrak{C}$. Let $z_j^{\text{var}} \in \{0,1\}^n$ be the bitstring*

$$z_j^{\text{var}}[b] = \begin{cases} 1 & \text{if } v_b \in C_j \text{ or } \neg v_b \in C_j, \\ 0 & \text{otherwise.} \end{cases}$$

Let $z_j^{\text{neg}} \in \{0,1\}^n$ be the bitstring

$$z_j^{\text{neg}}[b] = \begin{cases} 1 & \textit{if } \neg v_b \in C_j, \\ 0 & \textit{otherwise.} \end{cases}$$

Then the contribution to the i^{th} Walsh coefficient from clause C_j is

$$w_i^{(C_j)} = \begin{cases} \frac{2^k-1}{2^k} & \textit{if } i = 0, \\ -\frac{1}{2^k}\psi_i(z_j^{\text{neg}}) & \textit{if } i \neq 0 \textit{ and } i \subseteq z_j^{\text{var}}, \\ 0 & \textit{otherwise.} \end{cases}$$

Proof. We make the following case distinction. Either $i = 0$, $i \subseteq z_j^{\text{var}}$ and $i \neq 0$, or $i \not\subseteq z_j^{\text{var}}$.

Case 1: Suppose $i = 0$. Then by Equation (2.22), we have

$$w_0^{(C_j)} = \frac{1}{2^n}\sum_{x=0}^{2^n-1} g_j(x)\psi_0(x) = \frac{1}{2^n}\sum_{x=0}^{2^n-1} g_j(x) \qquad \text{since } \psi_0(x) = 1. \tag{3.3}$$

Recall that we are assuming $|C_j| = k$. Hence, out of 2^n possible assignments, there are 2^{n-k} assignments x which leave C_j unsatisfied and $2^n - 2^{n-k} = 2^{n-k}(2^k - 1)$ assignments x that leave C_j satisfied. If C_j is unsatisfied, $g_j(x)$ evaluates to 0, otherwise it evaluates to 1 so we can write the equality in (3.3) as

$$w_0^{(C_j)} = \frac{1}{2^n}2^{n-k}(2^k - 1) = \frac{2^k - 1}{2^k}.$$

Case 2: Suppose $i \not\subseteq z_j^{\text{var}}$. In this case, there must exist an integer $0 \leq a < 2^n$ such that $2^a \wedge i \neq 0$ and $2^a \wedge z_j^{\text{var}} = 0$. Note that the latter condition implies $v_a \notin C_j$. We have the following identities.

$$g_j(x) = g_j(x \oplus 2^a), \tag{3.4}$$

since $v_a \notin C_j$, and,

$$\psi_i(x) = -\psi_i(x \oplus 2^a), \tag{3.5}$$

since $2^a \wedge i \neq 0$.

We define the following partition of $\{0,1\}^n$

$$S_a = \{x \in \{0,1\}^n : 2^a \wedge x = 0\}$$

to be the set of all bitstrings x with $x[a] = 0$. By Equation (2.22), we have

$$\begin{aligned}
w_i^{(C_j)} &= \frac{1}{2^n}\sum_{x=0}^{2^n-1} g_j(x)\psi_i(x) \\
&= \frac{1}{2^n}\sum_{x\in S_a} \left(g_j(x)\psi_i(x) + g_j(x\oplus 2^a)\psi_i(x\oplus 2^a)\right) \\
&= \frac{1}{2^n}\sum_{x\in S_a} \left(g_j(x)\psi_i(x) + g_j(x)\psi_i(x\oplus 2^a)\right) && \text{by (3.4),} \\
&= \frac{1}{2^n}\sum_{x\in S_a} g_j(x)\left(\psi_i(x) + \psi_i(x\oplus 2^a)\right) \\
&= \frac{1}{2^n}\sum_{x\in S_a} g_j(x)\left(\psi_i(x) - \psi_i(x)\right) = 0. && \text{by (3.5),}
\end{aligned}$$

Case 3: Finally, suppose $i \subseteq z_j^{\text{var}}$ and $i \neq 0$. In this case, there exists an integer $0 \leq a < 2^n$ such that $2^a \wedge z_j^{\text{var}} = 2^a \wedge i = 0$.

Again, since $v_a \notin C_j$, the equality in (3.4) still holds. However, in this case, we now have

$$\psi_i(x) = \psi_i(x \oplus 2^a), \tag{3.6}$$

since $i \wedge 2^a = 0$. By Equation (2.22), we have

$$\begin{aligned}
w_i^{(C_j)} &= \frac{1}{2^n}\sum_{x=0}^{2^n-1} g_j(x)\psi_i(x) \\
&= \frac{1}{2^n}\sum_{x\in S_a} \left(g_j(x)\psi_i(x) + g_j(x\oplus 2^a)\psi_i(x\oplus 2^a)\right) \\
&= \frac{1}{2^n}\sum_{x\in S_a} \left(g_j(x)\psi_i(x) + g_j(x)\psi_i(x\oplus 2^a)\right) && \text{by (3.4),} \\
&= \frac{1}{2^n}\sum_{x\in S_a} 2g_j(x)\psi_i(x). && \text{by (3.6),}
\end{aligned}$$

Note that there are $n-k$ ways to choose a bit position a outside of z_j^{var} (since $\langle z_j^{\text{var}}, z_j^{\text{var}}\rangle = k$). Thus we can also choose an $a' \neq a$ such that $2^{a'} \wedge z_j^{\text{var}} = 2^{a'} \wedge i = 0$ and let the partition

$$S_{a\cup a'} = \{x \in \{0,1\}^n : 2^a \wedge x = 2^{a'} \wedge x = 0\}.$$

Following the argument above, we have

$$w_i^{(C_j)} = \frac{1}{2^n} \sum_{x \in S_{a\cup a'}} 2 \times 2 g_j(x)\psi_i(x).$$

Continuing this argument for all $n-k$ possible bit positions not contained in z_j^{var} we have

$$\begin{aligned} w_i^{(C_j)} &= \frac{1}{2^n} \sum_{x:\in S_{a\cup a'\cup a''\cup a'''\ldots}} 2 \times 2 \times 2 \times \cdots \times 2 g_j(x)\psi_i(x) \\ &= \frac{1}{2^n} \sum_{x:x\subseteq z_j^{\text{var}}} 2^{n-k} g_j(x)\psi_i(x) = \frac{1}{2^k} \sum_{x:x\subseteq z_j^{\text{var}}} g_j(x)\psi_i(x). \end{aligned}$$

We now define the function $g_j' : \{0,1\}^k \to \{0,1\}$ as $g_j'\left(\mathscr{P}\left(x, z_j^{\text{var}}\right)\right) = g_j(x)$. From above we have

$$\begin{aligned} w_i^{(C_j)} &= \frac{1}{2^k} \sum_{x:x\subseteq z_j^{\text{var}}} g_j(x)\psi_i(x) \\ &= \frac{1}{2^k} \sum_{x:x\subseteq z_j^{\text{var}}} g_j'(\mathscr{P}(x, z_j^{\text{var}}))\psi_{\mathscr{P}(i,z_j^{\text{var}})}(\mathscr{P}(x, z_j^{\text{var}})) \quad \text{by Lemma 2.6,} \\ &= \frac{1}{2^k} \sum_{x=0}^{2^k-1} g_j'(x)\psi_{\mathscr{P}(i,z_j^{\text{var}})}(x). \end{aligned}$$

But g_j' is zero for exactly one length-k bitstring. Let z be that string. Then,

$$\begin{aligned} &= \frac{1}{2^k}\left(\left(\sum_{x=0}^{2^k-1} \psi_{\mathscr{P}(i,z_j^{\text{var}})}(x)\right) - \psi_{\mathscr{P}(i,z_j^{\text{var}})}(z)\right) \\ &= -\frac{1}{2^k}\psi_{\mathscr{P}(i,z_j^{\text{var}})}(z) \\ &= -\frac{1}{2^k}\psi_{\mathscr{P}(i,z_j^{\text{var}})}(\mathscr{P}(z, z_j^{\text{var}})) \\ &= -\frac{1}{2^k}\psi_i(z_j^{\text{neg}}). \end{aligned}$$

The final equality comes from the consequence of Lemma 2.6, and the fact that packing z into the z_j^{var} mask gives exactly the string z_j^{neg}. □

It immediately follows from Lemma 3.1 that the objective function for a Max-k-Sat instance can be written in the Walsh expansion

$$f(x) = \sum_i w_i \psi_i(x),$$

where

$$w_i = \begin{cases} m\frac{2^k-1}{2^k} & \text{if } i = 0, \\ -\frac{1}{2^k}\sum_{j=1}^m \psi_i(z_j^{\text{neg}}) & \text{if } i \neq 0 \text{ and } i \subseteq z_j^{\text{var}}, \\ 0 & \text{otherwise.} \end{cases}$$

Recall from Equation (2.24) that Ψ_p denotes a linear combination of Walsh functions of order p. We will refer to this entity as an order-p *Walsh span element* since it is an element of the linear space $\mathscr{F}(\{0,1\}^n)$ spanned by the Walsh functions of order p. Now we can write the objective function in (3.1) as a sum over each order-p Walsh span element:

$$f(x) = \sum_{p=0}^{k} \Psi_p(x). \tag{3.7}$$

From Chapter 2 recall our notational convention of writing the expectation of a function f over a subset X of its domain is as $\langle f \rangle_X$. Thus $\langle f \rangle_{N(x)}$ denotes the expectation of f over the neighborhood of a point x. We can exploit the basis expansion introduced in Chapter 2 to derive an expression for the exact value of $\langle f \rangle_{N(x)}$. On any Max-k-Sat instance, the expectation of f over the neighborhood is a linear combination of $k+1$ span elements evaluated at x.

$$\begin{aligned}\langle f\rangle_{N(x)} &= \frac{1}{|N(x)|}\sum_{y\in N(x)} f(y)\\ &= \frac{1}{n}\boldsymbol{A}f(x)\\ &= \sum_{p=0}^{k}\frac{1}{n}\boldsymbol{A}\Psi_p(x)\\ &= \sum_{p=0}^{k}\left(1-\frac{2p}{n}\right)\Psi_p(x), \end{aligned}\tag{3.8}$$

since Ψ_p is an eigenfunction of $\boldsymbol{A}$ corresponding to eigenvalue $(n-2p)$. We will later use this expression for expectation of f over the neighborhood in a nonconstructive combinatorial proof that employs the probabilistic method pioneered by Erdös [Erd59] to prove the main result of the chapter.

The following two lemmas will be useful in the next section. First, we show that the zero-order Walsh span element is always a constant that is equal to the mean objective function value over $\{0,1\}^n$.

Lemma 3.2. *Let $\bar{f}$ be the mean objective value over $\{0,1\}^n$,*

$$\bar{f} = \frac{1}{2^n}\sum_{x\in\{0,1\}^n} f(x).$$

For all $x\in\{0,1\}^n$, the zero-order Walsh span element is the constant function

$$\Psi_0(x) = \bar{f}.$$

Proof. Let $x\in\{0,1\}^n$. There is only one Walsh function of order zero: $\psi_0(x)=1$. We have $\Psi_0(x)=w_0\psi_0(x)=w_0$. Note that for $p\neq 0$ we have

$$\frac{1}{2^n}\sum_{x\in\{0,1\}^n}\Psi_p(x) = 0 \tag{3.9}$$

because of the parity of bitstrings of order p. By some algebraic manipulation,

$$\begin{aligned}
w_0 &= \left(\frac{1}{2^n} \sum_{x \in \{0,1\}^n} w_0 \right) \\
&= \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \Psi_0(x) \\
&= \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \Psi_0 + \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \sum_{p=1}^{k} \Psi_p(x) && \text{by (3.9),} \\
&= \frac{1}{2^n} \sum_{x \in \{0,1\}^n} \sum_{p=0}^{k} \Psi_p(x) \\
&= \frac{1}{2^n} \sum_{x \in \{0,1\}^n} f(x) && \text{by (3.7).}
\end{aligned}$$

□

In the next section, we will need to bound the value of Ψ_p over all states $x \in X$. We use the absolute values of the Walsh coefficients w_i to do so.

Lemma 3.3. *For all $x \in X$,*

$$\sum_{\langle i,i \rangle = p} -|w_i| \leq \Psi_p(x) \leq \sum_{\langle i,i \rangle = p} |w_i|.$$

Proof. Let x be an arbitrary state in X. By definition we have

$$\Psi_p(x) = \sum_{\langle i,i \rangle = p} w_i \psi_i(x) = \sum_{\langle i,i \rangle = p} \pm |w_i|,$$

since $\psi_i(x) = \pm 1$ and $w_i = \pm |w_i|$. Clearly, the smallest that each term could be is $-|w_i|$ and the largest is $|w_i|$. □

3.2 The Max-3-Sat search space

Despite their incompleteness, local search algorithms are typically counted as among the state-of-the-art for solving propositional satisfiability (the decision problem) and maximum satisfiability (the optimization problem). In the context of local search, both cases are identical since the *objective function* for local search is nearly always characterized as the number of satisfied clauses under a given assignment. Several local search studies in the past several years have focused on $k = 3$: the smallest case in which the decision problem is NP-complete (though $k = 2$ is NP-hard for the optimization problem). We now specialize our attention to the particular case of $k = 3$.

Since the behavior of local search algorithms closely depends on the underlying structure of the search space, a large number of researchers have conducted empirical investigations on certain structural features of the MAX-3-SAT[3] problem. Hoos and Stützle [HS04] defined *search position* types which correspond to how improving, equal, and disimproving moves behave in the neighborhood of a point. For example, the familiar *strict local maximum (minimum)* where each neighborhood element has a strictly lower (higher) objective function value is a particular search position. They further identify local minima, local maxima (i.e., non strict variance), ledges (improving, disimproving, and equal moves), slopes (no equal moves), and interior plateau points. This concept is roughly illustrated in Figure 3.1. For a collection of MAX-3-SAT problems, they empirically measured the distribution of these positions across states either exhaustively (for very small problems) or using a biased sampling procedure. Most notably, they found that *no interior plateau states were found for any of the instances* [HS04, page 214]. In Section 3.2.2, we will prove rigorously that interior plateau states cannot exist in a large fraction of the search space.

[3] Though many of these works are motivated by the study of local search on the decision problem (i.e., propositional 3-SAT), in all cases the MAX-3-SAT objective function was used. For consistency, and the reasoning outlined above, we have elected to use the single term "MAX-3-SAT" since all results are applicable to the more general optimization problem.

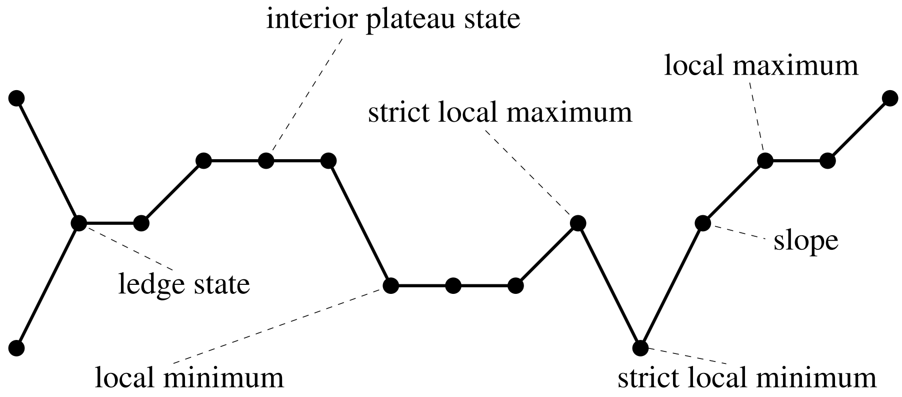


Figure 3.1: The *search position* types of Hoos and Stüztle (illustration adapted from [HS04]).

Clark et al. [CFG$^+$96] studied the relationship between problem hardness (in terms of search cost) and the number of solutions on random 3-Sat problems. In this work, they empirically measured the cost of several local search algorithms and found that search cost tended to be inversely proportional to the count of global optima in the space, and that this relationship changed through the solubility phase transition. Similar to the search positions of Hoos and Stützle, Frank et al. [FCS97] developed a taxonomy of search space *regions* for Max-3-Sat and experimentally probed the space by probabilistic sampling of the space (using a local search algorithm) in order to characterize the empirical distribution of differing types of these regions such as plateaus and local optima.[4] Smyth [Smy04] followed somewhat the approach of Frank et al. to empirically characterize plateaus in the Max-3-Sat search space in an effort to relate their characteristics (such as size and the distribution of incident states with improving objective). Smyth's work also focused on empirically assessing properties (e.g., diameter and branching

[4] Note that while their definition of plateau coincides with ours, their definition of local optima is distinct since they have allowed multiple states to belong to a single local optimum.

factor) of the subgraphs induced by plateau states on a certain level. Finally, Yokoo [Yok97] investigated the dependency of search cost on search space characteristics by exhaustively enumerating the search spaces of some small 3-Sat problems to measure plateau size and studying empirically how local search runtime is related to the size of plateaus.

Despite the large amount of empirical work on characterizing search space structure for Max-3-Sat, very few theoretical analyses exist that can make general statements about every Max-3-Sat search space. We now employ the framework developed in Chapter 2 to make some assertions about the Max-3-Sat search space in general.

3.2.1 Bounding the level of local maxima

A state x is said to be a *local maximum* if, for all $y \in N(x)$, $f(y) \leq f(x)$. We point out that this definition is distinct from studies that allow for multi-state local maxima (e.g., [FCS97]). Our single-state definition coincides with Hoos and Stützle [HS04]. Furthermore, every *global maximum* is also a local maximum.

In Chapter 2 we discussed the *maximum principle* for elementary landscapes [BLS07]. The maximum principle was proved by Grover [Gro92] and states that for *elementary* landscapes no local maxima (minima) lie below (above) the mean value of the objective function over $\mathcal{X}$. This will not necessarily hold for arbitrary functions. However, we show here that the basis decomposition of the Max-3-Sat objective function provides us with a series of eigenfunctions, or elementary *components*. Knowledge of these components and their properties allow us to *bound* the evaluation level of local maxima on Max-3-Sat.

Before we continue, we prove the following lemma that provides an identity for a series expansion (when $k = 3$) that will allow for some algebraic manipulation in the theorems below.

Lemma 3.4. *For the* Max-3-Sat *objective function we have the following identity.*

$$\sum_{p=0}^{3} p\Psi_p(x) = 2f(x) - 2\bar{f} - \Psi_1(x) + \Psi_3(x).$$

Proof. The series is equal to

$$\sum_{p=0}^{3} p\Psi_p(x) = \Psi_1(x) + 2\Psi_2(x) + 3\Psi_3(x).$$

We can group the terms on the right hand side as follows

$$\Big(\Psi_1(x) + \Psi_2(x) + \Psi_3(x)\Big) + \Big(\Psi_2(x) + 2\Psi_3(x)\Big).$$

By the decomposition in Equation (3.7),

$$\Big(f(x) - \Psi_0(x)\Big) + \Big(f(x) - \Psi_0(x) - \Psi_1(x) + \Psi_3(x)\Big).$$

By Lemma 3.2,

$$\Big(f(x) - \bar{f}\Big) + \Big(f(x) - \bar{f} - \Psi_1(x) + \Psi_3(x)\Big),$$

and simplifying gives the result. □

Theorem 3.1. *On any* 3-Sat *instance with n variables and m clauses, there exists a positive real number τ such that for any state x, if $f(x) < \bar{f} - \tau$, then x cannot be a local maximum.*

Proof. We begin by showing if $f(x) < \langle f \rangle_{N(x)}$, it cannot be a local maximum. We will then use the previous results to bound the inequality. Let x be a state such that $f(x) < \langle f \rangle_{N(x)}$. There exists some point y in the neighborhood of x that has an evaluation $f(y) > f(x)$. Thus x cannot be a local maximum. In this case,

$$f(x) < \langle f \rangle_{N(x)} = \sum_{p=0}^{3} \left(1 - \frac{2p}{n}\right) \Psi_p(x) \qquad \text{by (3.8)},$$

allowing us to write

$$f(x) < \sum_{p=0}^{3} \Psi_p(x) - \frac{2}{n} \sum_{p=0}^{3} p\Psi_p(x).$$

The first term on the right hand side is simply the decomposition of $f(x)$ given by Equation (3.7). Thus we can make the following substitution.

$$f(x) < f(x) - \frac{2}{n} \sum_{p=0}^{3} p\Psi_p(x).$$

By Lemma 3.4,

$$f(x) < f(x) - \frac{2}{n} \left(2f(x) - 2\bar{f} - \Psi_1(x) + \Psi_3(x)\right).$$

Simplifying, we have

$$f(x) < \bar{f} + \frac{1}{2} \left(\Psi_1(x) - \Psi_3(x)\right). \tag{3.10}$$

Inequality (3.10) describes a threshold that depends on $\Psi_1(x)$ and $\Psi_3(x)$ such that if $f(x)$ is less than this threshold, x cannot be locally maximum. We now give a threshold that holds *over the entire search space.*

By Lemma 3.3, we have for *any* $x \in X$,

$$\left(\Psi_1(x) - \Psi_3(x)\right) \geq \left(\sum_{\langle i,i \rangle = 1} -|w_i| - \sum_{\langle i,i \rangle = 3} |w_i| \right),$$

and letting

$$\tau = \frac{1}{2} \left(\sum_{\langle i,i \rangle = 1} |w_i| + \sum_{\langle i,i \rangle = 3} |w_i| \right), \tag{3.11}$$

we now have the following bound on the r.h.s. of Inequality (3.10).

$$\bar{f} - \tau \leq \bar{f} + \frac{1}{2} \left(\Psi_1(x) - \Psi_3(x)\right),$$

and thus, for all $x \in X$, if $f(x) < \bar{f} - \tau$, then x cannot be a local maximum. The threshold $\bar{f} - \tau$ is simply computed (in polynomial time) by summing the absolute Walsh coefficients of order 1 and 3 and holds over the entire search space. □

We have thus just proved that, in the MAX-3-SAT search space, all local maxima must lie above $\bar{f} - \tau$.

3.2.2 Bounding the level of unit width

In a similar manner, we can bound the function value at which plateaus of width greater than one can appear.

Definition 3.1. *A plateau is a maximal set P of states such that for all $x, y \in P$ there is a path $(x = x_1, x_2, \ldots, x_t = y)$ of length $t \geq 1$ with $f(x) = f(x_i)$ for $i = 1, 2, \ldots, t$ and, if $t > 1$, $x_{i+1} \in N(x_i)$. The* level *of a plateau P is the evaluation $f(x_p), \forall x_p \in P$.*

We say the neighborhood of a state x is *flat* if, for all $y \in N(x)$, $f(y) = f(x)$, that is, x has the same value as all the states in its neighborhood. A state with a flat neighborhood corresponds to the *interior plateau* search position described by Hoos and Stützle [HS04]. We show that flat neighborhoods cannot exist at certain levels of the objective function.

Theorem 3.2. *On any* 3-SAT *instance with n variables and m clauses, there exists a positive real number τ such that for any state x, if $f(x) < \bar{f} - \tau$ or $f(x) > \bar{f} + \tau$, then x cannot have a flat neighborhood.*

Proof. We prove the equivalent contrapositive. Let x be a state with a flat neighborhood. We have

$$\begin{aligned}
f(x) &= \langle f \rangle_{N(x)} \\
&= \sum_{p=0}^{3} \left(1 - \frac{2p}{n}\right) \Psi_p(x) \\
&= \sum_{p=0}^{3} \Psi_p(x) - \frac{2}{n} \sum_{p=0}^{3} p\Psi_p(x) \\
&= f(x) - \frac{2}{n} \sum_{p=0}^{3} p\Psi_p(x) && \text{by (3.7).}
\end{aligned}$$

Therefore, at such a point x we must have

$$\begin{aligned}
\sum_{p=0}^{3} p\Psi_p(x) &= 0, \\
2f(x) - 2\bar{f} - \Psi_1(x) + \Psi_3(x) &= 0 && \text{by Lemma 3.4.}
\end{aligned}$$

Thus if x has a flat neighborhood, the following must hold.

$$f(x) = \bar{f} + \frac{1}{2}\left(\Psi_1(x) - \Psi_3(x)\right). \tag{3.12}$$

Using Lemma 3.3 we can bound the terms $\Psi_1(x)$ and $\Psi_3(x)$ giving the following

$$\bar{f} - \tau \leq f(x) \leq \bar{f} + \tau,$$

where τ is given by Equation (3.11) in Theorem 3.1. □

Recall the width of a plateau P is the minimal length path between any state in P and one not in P. We have the following corollary.

Corollary. *A plateau P with level less than $\bar{f} - \tau$ or greater than $\bar{f} + \tau$ cannot have width greater than 1.*

Proof. This follows directly from the fact that no flat neighborhoods exist outside of the range $\bar{f} - \tau$ to $\bar{f} + \tau$. Thus, for these points, every state on a plateau P must have at least one neighbor outside P and the width of P is at most 1. □

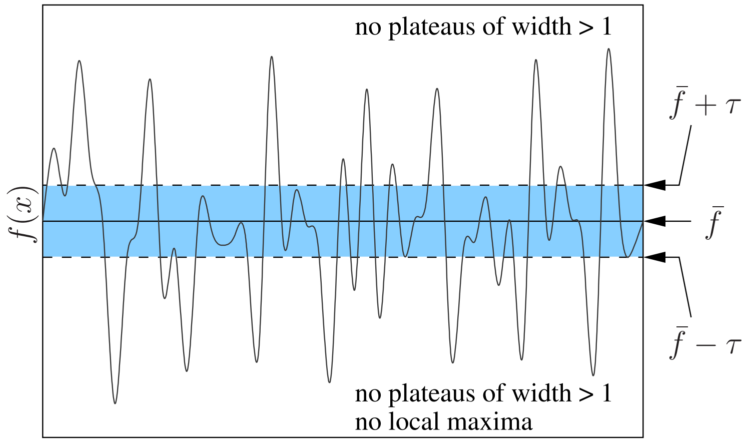


Figure 3.2: An illustration of the proved properties. No plateaus of width strictly greater than one can lie outside the interval. No local maxima can lie below the interval.

3.3 Derived values in practice

We have shown how the average value of the neighborhood can be obtained analytically for any particular state and that a region (τ from $\bar{f}$) can be defined outside of which plateaus of width greater than one cannot exist and certain local optima cannot be found. We illustrate the proved properties in Figure 3.2. In this section, we compute numerical values of the expectation value and of τ on representative instances to show that (1) the expectation value computation is informative, and (2) the region is non-trivial in benchmark problem instances.

3.3.1 The neighborhood expectation value

The neighborhood expectation value computed in Equation (3.8) is useful because it can potentially provide algorithms with higher resolution information about states than the objective function. For example, given two states x and y with $f(x) = f(y)$, it is not necessarily the case that the neighborhood expectation values are equal for both x and y.

Stochastic local search algorithms applied to Max-k-Sat problems often must select a neighboring state from a large set of moves with equal evaluation. This presents a problem for such algorithms due to the lack of gradient information in the neighborhood [HS04]. A collection of states at the same evaluation level are indistinguishable in terms of objective function value. However, we conjecture the expectation value can serve as a predictor of the number of *improving moves* that exit a particular state.

To illustrate this concept, we sampled 100 states at a particular objective function level ($f(x) = 390$) on each of 1000 instances that make up the uf100-430 benchmark set in SATLIB (100 vars, 430 clauses). For each point we calculated the correspondence between the expectation value given by Equation (3.8) and the actual number of improving moves in the neighborhood of the state. These data are plotted in Figure 3.3. A correlation test gives a strong positive correlation value of 0.51 with $p < 2.2 \times 10^{-16}$ indicating that better expectation leads to more potential for improvement. These data indicate that the neighborhood expectation value can provide useful information about the neighborhoods of points even if they are equal in objective function value. We will revisit this phenomenon in Chapter 6 and exploit this information in a local search application on Max-k-Sat.

3.3.2 Numerical values of τ

To demonstrate the region outside the interval is not trivial, we computed the values for τ as a percentage of the objective function range m across 18 benchmark distributions from SATLIB and the 2008 SAT competition. In Table 3.1 we report the mean (μ), standard deviation (σ), minimum, and maximum of the value τ/m over all N problems in each distribution.

The mean value of τ is consistently about 10% of the range m with a relatively low standard deviation. The maximum value of τ does not exceed 13% of the total objective function range over all the problem distributions we tested.

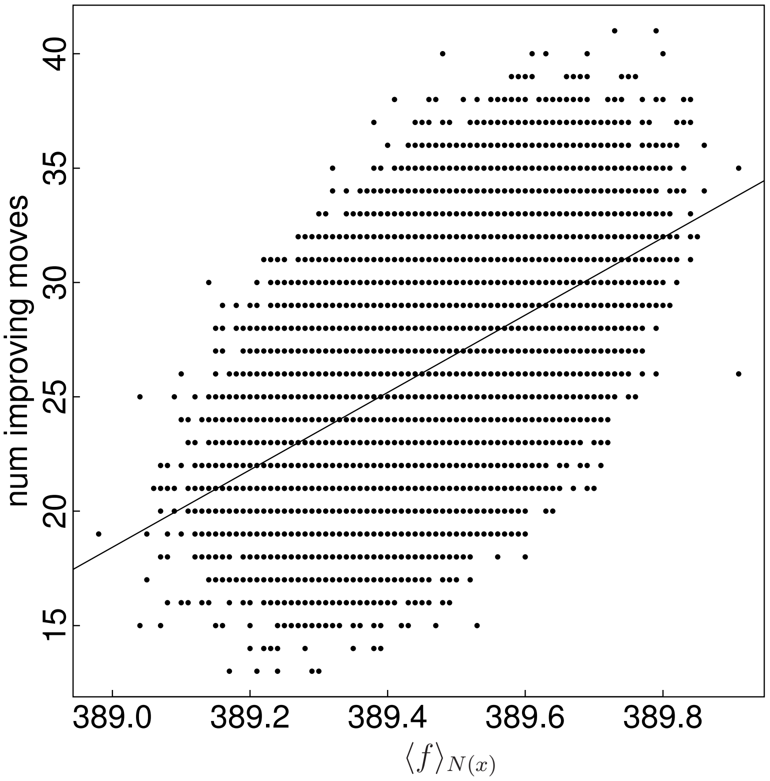


Figure 3.3: Number of improving moves vs $\langle f \rangle_{N(x)}$ at $f(x) = 390$ for 100 points each on 1000 instances of SATLIB benchmark set uf100-430. Line indicates linear best fit.

SATLIB

set	setsize	μ	σ	min	max
uf20-91	1000	0.10252	0.00707	0.08104	0.12775
uf50-218	1000	0.10467	0.00421	0.08945	0.11984
uf75-325	100	0.10487	0.00358	0.09538	0.11231
uf100-430	1000	0.10483	0.00307	0.09680	0.11483
uf125-538	100	0.10477	0.00241	0.09898	0.11245
uf150-645	100	0.10514	0.00221	0.10039	0.11027
uf175-753	100	0.10533	0.00239	0.09910	0.11155
uf200-860	100	0.10469	0.00203	0.09942	0.11047
uf225-960	100	0.10484	0.00194	0.09870	0.10898
uf250-1065	100	0.10478	0.00167	0.10082	0.10986
uuf50-218	1000	0.10131	0.00406	0.08888	0.11640

2008 SAT competition

set	setsize	μ	σ	min	max
v360	10	0.10382	0.00146	0.10046	0.10535
v400	10	0.10370	0.00198	0.10072	0.10651
v450	10	0.10369	0.00162	0.10016	0.10571
v500	10	0.10384	0.00177	0.09947	0.10616
v550	10	0.10366	0.00113	0.10137	0.10494
v600	10	0.10404	0.00107	0.10270	0.10603
v650	10	0.10400	0.00108	0.10293	0.10627

Table 3.1: Computed statistics for τ/m across several benchmark distributions from SATLIB and 2008 SAT competition.

Chapter 4

Efficient Construction of Local Moments

We now investigate how the results in previous chapters can be applied to characterize the distribution of objective function values over partitions of the state set that represent regions in the neighborhood graph.[5] In particular, we consider partitioning $\mathcal{X}$ into *local regions* (i.e., sets of states within a certain graph-theoretic distance) of a given state and studying the *moments* of the distribution of objective function values across the region. We will show that if f is a pseudo-Boolean function that is epistatically bounded by k, the exact moments over local regions can be constructed in polynomial time, even when the size of the region is exponential.

In Chapter 2 we made the assertion (c.f., Remark 2.1) that the problem of finding the *expectation* of the values of any function f evaluated over some set of states X reduces to finding the expectation of basis functions over X from some convenient basis expansion of f. We will now develop this concept to compute moments over local regions of the space. In particular, we will show that computing the c^{th} moment of the distribution of a function f reduces to finding the expectation of

[5] The work presented in this chapter appears in a journal article in *Theoretical Computer Science* [SWH11b].

the c^{th} power of f. Thus, we will show how to compute the moments of f over Hamming regions by first finding a convenient basis function expansion of the c^{th} power of f and then efficiently computing the expectation of the basis functions over the Hamming region using an eigendecomposition equation. This ultimately leads to a polynomial-time algorithm for computing moments over arbitrary Hamming regions in the space.

This result is significant since the cardinality of such regions can be exponential in the problem size. For example, a radius $n/2$ Hamming sphere contains $\Omega(2^{n/2})$ unique states and any Hamming ball of radius εn for $0 \leq \varepsilon \leq 1$ has $\Omega(2^{\varepsilon n})$ unique states. Any k-bounded pseudo-Boolean function is expressible by a sum of m subfunctions that each depend on at most k bits. On such functions, our approach has a time complexity of $O(m^c)$ to calculate the exact value for the c^{th} moment of f over any sphere and a time complexity of $O(rm^c)$ to calculate the exact moment over a ball of radius r. In general, since there can be at most $\binom{n}{k}$ functions on k bits, m is $O(n^k)$ giving a worst case polynomial time bound for our approach of $O(n^{ck})$ and $O(rn^{ck})$ for general bounded pseudo-Boolean functions. However, in many combinatorial optimization problems, m is typically linear (Max-k-Sat, NK-landscapes) or quadratic (MAX-CUT and other graph optimization problems) in n. In these cases, this approach has complexity $O(rn^c)$ and $O(rn^{2c})$, respectively.

This chapter generalizes the work of Heckendorn, Rana, and Whitley [HRW99]. Using a Walsh decomposition, they showed how one could efficiently compute summary statistics (e.g., central moments such as the mean, variance, skewness, and kurtosis) over the entire search space for Max-k-Sat and all k-bounded pseudo-Boolean functions, which they call *embedded landscapes.* Heckendorn [Hec02] also extended this work to prove that moments of the distribution of fitness values over hyperplanes of $\{0,1\}^n$ can be computed in polynomial time for such functions. While hyperplane statistics are important from the perspective of hyperplane sampling for selection-based genetic algorithms that employ recombination [Gol89, LV91, Gol92], processes that explore the search space by making decisions based on the value of nearby states will be strongly influenced by the distribution of fitness values in volumes of the search space that are close by. Toward that end, we further generalized the work of Heckendorn et al. to show how low order moments of the objective function distribution over *local* regions (Hamming balls and spheres) can be computed in polynomial time for k-bounded functions.

4.1 Moments of codomain distributions

A *moment* is a quantitative measure that describes the nature (e.g., location, spread, shape) of a distribution. In this research, we are specifically interested in the distribution of values across the image of a set of some domain elements for a function.

Suppose we have a pseudo-Boolean function

$$f : \{0,1\}^n \to \mathbb{R}.$$

f has an associated cumulative distribution function

$$P : \mathbb{R} \to [0,1].$$

where $P(a) = \Pr\{f(x) \leq a, \forall x \in \{0,1\}^n\}$. The c^{th} moment of the distribution of codomain values of f is defined by the Riemann-Stieltjes integral

$$\mu_c = \int_{-\infty}^{\infty} a^c dP(a).$$

This is equivalent to the distribution of a random variable that gives the value of f evaluated at a point chosen uniformly at random from the domain.

In this chapter we are interested in characterizing the distribution of codomain values over meaningful subsets of $\{0,1\}^n$. Let $X \subseteq \{0,1\}^n$ be a set of points. The c^{th} *moment* of f over X can be defined as moments of a random variable that assumes the value of $f(x)^c$ evaluated at a point x drawn uniformly at random from X. In this case, since each element of X is drawn with equal probability, the probability mass function is $\frac{1}{|X|}$ and we can define

$$\mu_c(X) = \frac{1}{|X|} \sum_{x \in X} f(x)^c \tag{4.1}$$

to be the c^{th} moment of f over the set X. For any nonempty set X, it should be clear that $\mu_0(X) = 1$. The first moment, $\mu_1(X)$, is the mean value of the function f evaluated over each point in X. The variance of f (the second *central moment*) over the set X can be written as

$$\sigma^2 = \mu_2(X) - \mu_1(X)^2.$$

In general, the c^{th} central moment of f over the subset X can be computed as

$$\sum_{i=0}^{c} \binom{c}{i} (-1)^{c-i} \mu_i(X) \mu_1(X)^{c-i}.$$

Higher central moments correspond to statistical quantities such as *skewness* and *kurtosis* which further characterize the shape of the distribution.

4.2 Local regions

The short-term dynamics of local search and mutation-based evolutionary algorithms are influenced by the statistical structure of the search space regions that are near the current search point. The state set taken together with the connectivity of the neighborhood operator form a metric space. In the case of $\{0,1\}^n$, this translates to the well-known Hamming metric.

Recall from Section 2.4.1 the definition of the set of Hamming neighbors of x: $N(x) = \{y : \mathcal{H}(x, y) = 1\}$ and the corresponding Hamming adjacency structure $\boldsymbol{A}$. Consider an arbitrary element x of the state set $\{0, 1\}^n$. We generalize the definition of the Hamming neighborhood in the following manner.

Definition 4.1. *The Hamming sphere of radius r about x is the set*

$$S^{(r)}(x) = \{y : \mathcal{H}(x, y) = r\}.$$

The set of spheres of distinct radius thus form a partition of the set of states $\{0, 1\}^n$ into equivalency classes.

Definition 4.2. *The Hamming ball of radius r about x is the union of all spheres about x with radius at most r*

$$B^{(r)}(x) = \{y : \mathcal{H}(x, y) \leq r\}.$$

The sphere of unit radius is thus equivalent to the Hamming neighborhood. We now discuss the construction of the exact moments of f over these local regions, e.g., $\mu_c(S^{(r)}(x))$ or $\mu_c(B^{(r)}(x))$ for some $c = O(1)$ and some $r = O(n)$. These moments can be calculated directly by enumerating all states in the region. However, in most cases, such regions have intractably large cardinality which prohibits direct computation. For instance, $|B^{(r)}(x)|$ has exponential growth in n when $r = \epsilon n$ for some $0 < \epsilon \leq 1$. The limiting case, of course, is $B^{(n)}(x)$ which covers the entire search space. In these cases, one might resort to sampling in an attempt to obtain an approximation for the moment (e.g., by computing the sample moments from a uniform or biased sample). However, if f is epistatically bounded, we can take advantage of the basis expansions developed in the previous chapter to compute exact (low) moments over regions in polynomial time, even if the cardinality of the region in question has exponential growth.

4.2.1 Using the Walsh basis expansion

To compute the c^{th} moment of f, it will become necessary to work with higher powers of f. Consider the pseudo-Boolean function constructed by taking f to the c^{th} power, that is,

$$f^c : \{0,1\}^n \to \mathbb{R}; \qquad f^c(x) = (f(x))^c, \quad \forall x \in \{0,1\}^n.$$

Since f^c is a pseudo-Boolean function, it can be written in a Walsh basis expansion.

Lemma 4.1. *Let $f : \{0,1\}^n \to \mathbb{R}$ with Walsh basis expansion $f(x) = \sum_i w_i \psi_i(x)$. The function f^c that gives the c^{th} power of f has a Walsh basis expansion*

$$f^c(x) = \sum_j \mathfrak{w}_j \psi_j(x),$$

where

$$\mathfrak{w}_j = \sum_{\substack{i_1,i_2,\ldots,i_c \\ i_1 \oplus i_2 \oplus \cdots \oplus i_c = j}} w_{i_1} w_{i_2} \ldots w_{i_c}.$$

Proof. Since $f^c(x) = f(x)^c$ we can write

$$\begin{aligned}
f^c(x) &= \left(\sum_i w_i \psi_i(x) \right)^c \\
&= \sum_{i_1} w_{i_1} \psi_{i_1}(x) \sum_{i_2} w_{i_2} \psi_{i_2}(x) \ldots \sum_{i_c} w_{i_c} \psi_{i_c}(x) \\
&= \sum_{i_1,i_2,\ldots,i_c} w_{i_1} w_{i_2} \ldots w_{i_c} \psi_{i_1}(x) \psi_{i_2}(x) \ldots \psi_{i_c}(x) \\
&= \sum_{i_1,i_2,\ldots,i_c} w_{i_1} w_{i_2} \ldots w_{i_c} \psi_{i_1 \oplus i_2 \oplus \cdots \oplus i_c}(x) \\
&= \sum_{\substack{i_1,i_2,\ldots,i_c \\ i_1 \oplus i_2 \oplus \cdots \oplus i_c = j}} (w_{i_1} w_{i_2} \ldots w_{i_c}) \, \psi_j(x).
\end{aligned}$$

This yields the claimed result. □

Given a function f and corresponding Walsh basis expansion

$$f(x) = \sum_i w_i \psi_i(x),$$

let

$$\mathcal{W}(f) = |\{i : w_i \neq 0\}|, \quad i = \{0, 1, \ldots, 2^n - 1\}$$

be the count of nonzero Walsh coefficients in the Walsh basis expansion of f. The following proposition is an immediate consequence of Lemma 2.8.

Proposition 4.1. *If f can be expressed as a sum of m subfunctions that each depend on at most k bits, then*

$$\mathcal{W}(f) \leq m2^k.$$

Proposition 4.1 makes precise the observations in Section 2.4.4 that a k-bounded pseudo-Boolean function f has a sparse representation in the Walsh basis. Furthermore, we observe that as long as f is epistatically bounded, we can also bound the number of nonzero terms in the Walsh expansion of f^c.

Lemma 4.2. *If f can be expressed as a sum of m subfunctions that each depend on at most k bits, then*

$$\mathcal{W}(f^c) \leq \binom{c + m2^k - 1}{c}.$$

Proof. Since $f^c(x) = f(x)^c$ we can write

$$f^c(x) = \left(\sum_i w_i \psi_i(x) \right)^c. \tag{4.2}$$

Consider a c^{th} order multinomial sum $(z_1 + z_2 + \ldots + z_b)^c$ in b indeterminates. The number of terms in the expansion of this expression is equal to the number of monomials of degree c on the variables z_i which is $\binom{c+b-1}{c}$. Setting $z_i = w_i\psi_i(x)$, the r.h.s. of (4.2) is such an expression. By Lemma 4.1 there are $b \leq m2^k$ terms in this sum which results in the claimed bound. □

As a corollary to Lemmas 4.1 and 4.2, if f is a k-bounded pseudo-Boolean, i.e., it can be expressed as a sum of m subfunctions that each depend on at most k

bits, then the Walsh basis expansion of f^c

$$f^c(x) = \sum_j \mathfrak{w}_j \psi_j(x) \tag{4.3}$$

has at most $\binom{c+m2^k-1}{c}$ nonzero terms.

4.3 Constructing moments in polynomial time

If f has a sparse representation in the eigenbasis of some linear operator that acts on $\mathscr{F}(\{0,1\}^n)$, then the image of f under that operator can be efficiently computed if its corresponding eigenvalues are known. In this section we will show that f has a sparse representation in the eigenbasis of the general adjacency structures defined above.

We begin by characterizing the adjacency structure for radius-r Hamming spheres. Let x be an arbitrary but fixed point in $\{0,1\}^n$. Consider a vertex y at some distance $\mathcal{H}(x,y)$. All Hamming neighbors of y are either one vertex closer to x or one vertex further away. Define the *approaching set*

$$\alpha(x,y) = \{z \in N(y) : \mathcal{H}(x,z) = \mathcal{H}(x,y) - 1\},$$

and the *retreating set*

$$\beta(x,y) = \{z \in N(y) : \mathcal{H}(x,z) = \mathcal{H}(x,y) + 1\}.$$

Thus the approaching and retreating sets partition the neighborhood set of y and

$$\alpha(x,y) \cup \beta(x,y) = N(y). \tag{4.4}$$

See Figure 4.1 for an illustration.

The set $S^{(r)}(x)$ consists of all strings at Hamming distance r from x: those strings that differ from x in exactly r positions. Hence $|S^{(r)}(x)| = \binom{n}{r}$. Consider

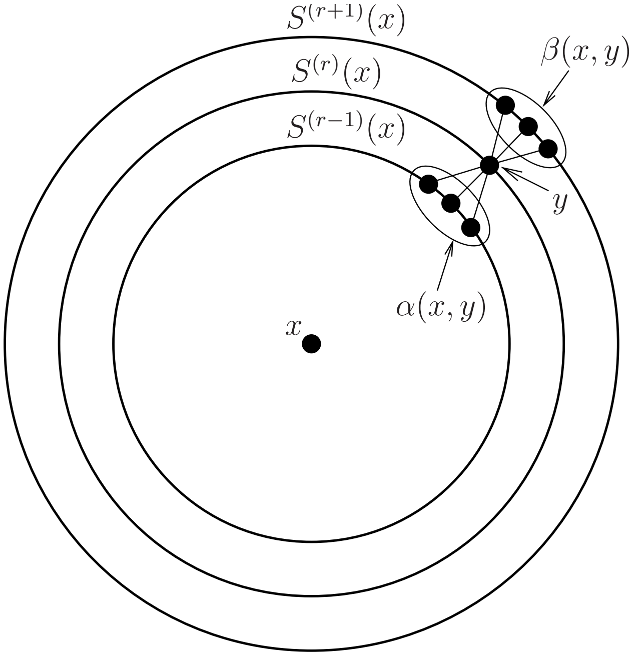


Figure 4.1: Illustration of approaching set $\alpha(x, y)$ and retreating set $\beta(x, y)$. For some y with $\mathcal{H}(x, y) = r$.

a state y on this sphere, that is, $\mathcal{H}(x, y) = r$. Since y differs from x in exactly r positions, there are r Hamming moves that result in some state z_1 with $\mathcal{H}(x, z_1) = r - 1$. Thus we have $|\alpha(x, y)| = r$. Furthermore, there are $n - r$ Hamming moves from y that result in a state z_2 with $\mathcal{H}(x, z_2) = r + 1$. Hence, $|\beta(x, y)| = n - r$.

A generalization of the adjacency matrix $\boldsymbol{A}$ which we will call the *sphere matrix* of radius r we define as

$$\boldsymbol{S}^{(r)}_{xy} = \begin{cases} 1 & \text{if } y \in S^{(r)}(x)\text{, that is, } \mathcal{H}(x, y) = r, \\ 0 & \text{otherwise.} \end{cases}$$

This matrix identifies all vertex pairs in which one is contained in the radius-r sphere of the other. We construct the sphere matrix $\boldsymbol{S}^{(r)}$ of radius r recursively in terms of $\boldsymbol{A}$. In order to do so, we will first prove some useful properties about sphere matrices.

The set $\{0, 1\}^n$ together with the Hamming distance function form a metric space so we have for all $x, y \in \{0, 1\}^n$, $\mathcal{H}(x, y) = \mathcal{H}(y, x)$ and sphere matrices of any radius are symmetric:

$$\boldsymbol{S}^{(r)}_{xy} = \boldsymbol{S}^{(r)}_{yx}. \tag{4.5}$$

Given any two sphere matrices, their product is a matrix that gives the number of elements in the intersection of the spheres they represent. Formally, let $\boldsymbol{S}^{(r)}$ and $\boldsymbol{S}^{(s)}$ be sphere matrices of radius r and s respectively. The product is the matrix

$$\begin{aligned}\left(\boldsymbol{S}^{(r)}\boldsymbol{S}^{(s)}\right)_{xy} &= \sum_z \boldsymbol{S}^{(r)}_{xz}\boldsymbol{S}^{(s)}_{zy} \\ &= \sum_z \boldsymbol{S}^{(r)}_{xz}\boldsymbol{S}^{(s)}_{yz} && \text{by (4.5)}, \\ &= |S^{(r)}(x) \cap S^{(s)}(y)|. && (4.6)\end{aligned}$$

We now characterize the particular matrix product $(\boldsymbol{S}^{(r-1)}\boldsymbol{A})$ which will be used in our recursive expression for $\boldsymbol{S}^{(r)}$.

Lemma 4.3. *Over* $\{0,1\}^n$ *we have for all* $r \in \{1, \ldots, n\}$,

$$(\boldsymbol{S}^{(r-1)}\boldsymbol{A})_{xy} = \begin{cases} r & \text{if } y \in S^{(r)}(x), \\ n-r+2 & \text{if } y \in S^{(r-2)}(x), \\ 0 & \text{otherwise.} \end{cases}$$

Proof. By (4.6) we have

$$(\boldsymbol{S}^{(r-1)}\boldsymbol{A})_{xy} = |S^{(r-1)}(x) \cap N(y)|,$$

since $\boldsymbol{A} = \boldsymbol{S}^{(1)}$ and $N(x) = S^{(1)}(x)$. Consider the neighbor set $N(y)$ of y. Recall from Equation (4.4) that the approaching and retreating sets $\alpha(x,y)$, $\beta(x,y)$ partition $N(y)$.

Suppose $y \in S^{(r)}(x)$. For all $z \in \alpha(x,y)$, $\mathcal{H}(x,z) = \mathcal{H}(x,y) - 1 = r - 1$. The neighbors of y that are in $S^{(r-1)}(x)$ are exactly the approaching set $\alpha(x,y)$. Thus we have

$$\begin{aligned}(\boldsymbol{S}^{(r-1)}\boldsymbol{A})_{xy} &= |S^{(r-1)}(x) \cap N(y)| \\ &= |\alpha(x,y)| \\ &= r.\end{aligned}$$

Now suppose $y \in S^{(r-2)}(x)$. For all $z \in \beta(x, y)$, $\mathcal{H}(x, z) = \mathcal{H}(x, y) + 1 = r - 1$. Thus the neighbors of y that are in $S^{(r-1)}(x)$ are exactly the retreating set $\beta(x, y)$. Thus we have

$$\begin{aligned}(\boldsymbol{S}^{(r-1)}\boldsymbol{A})_{xy} &= |S^{(r-1)}(x) \cap N(y)| \\ &= |\beta(x, y)| \\ &= n - r + 2.\end{aligned}$$

Finally suppose y is in neither sphere $S^{(r)}(x)$ nor $S^{(r-2)}(x)$. Then $\mathcal{H}(x, y) \neq r$ and $\mathcal{H}(x, y) \neq r - 2$. So

$$\begin{aligned}(\boldsymbol{S}^{(r-1)}\boldsymbol{A})_{xy} &= |S^{(r-1)}(x) \cap N(y)| \\ &= |(\alpha(x, y) \cup \beta(x, y)) \cap S^{(r-1)}(x)| \\ &= |\emptyset| \\ &= 0,\end{aligned}$$

since $\mathcal{H}(x, y) - 1 \neq r - 1$ and $\mathcal{H}(x, y) + 1 \neq r - 1$. □

The following lemma uses the above result to provide a matrix expression for the characteristic function of $y \in S^{(r)}(x)$. The expression involves the sphere matrices of radius $r - 1$ and $r - 2$. This will allow us to define $\boldsymbol{S}^{(r)}$ recursively in terms of lower radius sphere matrices.

Lemma 4.4. *Let x and y be arbitrary points in $\{0, 1\}^n$. Given sphere matrices $\boldsymbol{S}^{(r-1)}$ and $\boldsymbol{S}^{(r-2)}$ we have the following identity.*

$$\frac{1}{r}\left((\boldsymbol{S}^{(r-1)}\boldsymbol{A})_{xy} - (n - r + 2)\boldsymbol{S}^{(r-2)}_{xy}\right) = \begin{cases} 1 & \textit{if } y \in S^{(r)}(x), \\ 0 & \textit{otherwise.} \end{cases} \tag{4.7}$$

Proof. We prove this result by cases.

Case 1: $y \in S^{(r)}(x)$. By Lemma 4.3 we have $\boldsymbol{S}^{(r-1)}\boldsymbol{A}_{xy} = r$. Furthermore, since $y \notin S^{(r-2)}(x)$ we have $\boldsymbol{S}^{(r-2)}_{xy} = 0$. Thus Equation (4.7) evaluates to

$$\begin{aligned}\frac{1}{r}\left((\boldsymbol{S}^{(r-1)}\boldsymbol{A})_{xy} - (n-r+2)\boldsymbol{S}^{(r-2)}_{xy}\right) &= \frac{1}{r}(r-0)\\ &= 1.\end{aligned}$$

Case 2: $y \in S^{(r-2)}(x)$. By Lemma 4.3 we have $\boldsymbol{S}^{(r-1)}\boldsymbol{A}_{xy} = (n-r+2)$. Since $y \in S^{(r-2)}(x)$, $\boldsymbol{S}^{(r-2)}_{xy} = 1$ and Equation (4.7) evaluates to

$$\begin{aligned}\frac{1}{r}\left((\boldsymbol{S}^{(r-1)}\boldsymbol{A})_{xy} - (n-r+2)\boldsymbol{S}^{(r-2)}_{xy}\right) &= \frac{1}{r}((n-r+2)-(n-r+2))\\ &= 0.\end{aligned}$$

Case 3: $y \notin S^{(r)}(x)$ and $y \notin S^{(r-2)}(x)$. By Lemma 4.3 we have $\boldsymbol{S}^{(r-1)}\boldsymbol{A}_{xy} = 0$. Furthermore, since $y \notin S^{(r-2)}(x)$ we have $\boldsymbol{S}^{(r-2)}_{xy} = 0$. Thus Equation (4.7) evaluates to

$$\begin{aligned}\frac{1}{r}\left((\boldsymbol{S}^{(r-1)}\boldsymbol{A})_{xy} - (n-r+2)\boldsymbol{S}^{(r-2)}_{xy}\right) &= \frac{1}{r}(0-0)\\ &= 0.\end{aligned}$$ □

Hence, by Lemma 4.4 we can now define the sphere matrix recursively.

$$\boldsymbol{S}^{(r)} = \frac{1}{r}\left(\boldsymbol{S}^{(r-1)}\boldsymbol{A} - (n-r+2)\boldsymbol{S}^{(r-2)}\right). \tag{4.8}$$

We have the two base cases $\boldsymbol{S}^{(1)} = \boldsymbol{A}$ and $\boldsymbol{S}^{(0)} = \boldsymbol{I}$, where $\boldsymbol{I}$ is the $2^n \times 2^n$ identity matrix (this corresponds to the degenerate sphere $S^{(0)}(x) = \{x\}$). We now show that if f is an eigenfunction of $\boldsymbol{A}$ with eigenvalue λ, it is also an eigenfunction of the sphere matrix $\boldsymbol{S}^{(r)}$ with an eigenvalue that is a degree-r polynomial in λ.

Let f be some pseudo-Boolean function. Consider the matrix-vector product

$\boldsymbol{S}^{(r)}f$ evaluated at state x.

$$\begin{aligned}\boldsymbol{S}^{(r)}f(x) &= \langle e_x, \boldsymbol{S}^{(r)}f\rangle \\ &= \sum_{y\in\{0,1\}^n} \boldsymbol{S}^{(r)}_{xy} f(y) \\ &= \sum_{y\in S^{(r)}(x)} f(y), \qquad (4.9)\end{aligned}$$

since $\boldsymbol{S}^{(r)}_{xy} = 1 \iff y \in S^{(r)}(x)$, otherwise it is equal to zero. Lemma (2.1) provides the special case when $r = 1$.

It is now straightforward to show that eigenfunctions of the immediate Hamming neighborhood structure are also eigenfunctions of the radius r Hamming sphere. In particular, if φ_i is an eigenfunction of $\boldsymbol{A}$ with eigenvalue λ_i, it must also be an eigenfunction of $\boldsymbol{S}^{(r)}$ with eigenvalue $\gamma_i^{(r)}$ which is a scalar that is given by a nonlinear recurrence equation. We capture this in the following theorem.

Theorem 4.1. *If φ_i is an eigenfunction of $\boldsymbol{A}$ with eigenvalue λ_i, then φ_i is an eigenfunction of $\boldsymbol{S}^{(r)}$ with eigenvalue $\gamma_i^{(r)}$ given by the recurrence*

$$\gamma_i^{(r)} = \frac{1}{r}\left(\lambda_i \gamma_i^{(r-1)} - (n-r+2)\gamma_i^{(r-2)}\right),$$

with $\gamma_i^{(1)} = \lambda_i$ and $\gamma_i^{(0)} = 1$.

Proof. We proceed by induction on r. We have two base cases,

$$\boldsymbol{S}^{(0)}\varphi_i = \boldsymbol{I}\varphi_i = \varphi_i,$$

$$\boldsymbol{S}^{(1)}\varphi_i = \boldsymbol{A}\varphi_i = \lambda_i\varphi_i.$$

Thus $\gamma_i^{(0)} = 1$ and $\gamma_i^{(1)} = \lambda_i$. Suppose for induction that

$$\boldsymbol{S}^{(r-1)}\varphi_i = \gamma_i^{(r-1)}\varphi_i,$$

and

$$\boldsymbol{S}^{(r-2)}\varphi_i = \gamma_i^{(r-2)}\varphi_i,$$

for scalars $\gamma_i^{(r-1)}$ and $\gamma_i^{(r-2)}$. Thus,

$$\begin{aligned}
\boldsymbol{S}^{(r)}\varphi_i &= \frac{1}{r}\left(\boldsymbol{S}^{(r-1)}\boldsymbol{A} - (n-r+2)\boldsymbol{S}^{(r-2)}\right)\varphi_i && \text{by (4.8),}\\
&= \frac{1}{r}\left(\lambda_i\boldsymbol{S}^{(r-1)}\varphi_i - (n-r+2)\boldsymbol{S}^{(r-2)}\varphi_i\right)\\
&= \frac{1}{r}\left(\lambda_i\gamma_i^{(r-1)} - (n-r+2)\gamma_i^{(r-2)}\right)\varphi_i && \text{by induction,}
\end{aligned}$$

so we have the recurrence

$$\gamma_i^{(r)} = \frac{1}{r}\left(\lambda_i\gamma_i^{(r-1)} - (n-r+2)\gamma_i^{(r-2)}\right).$$

□

By Lemma 2.7, the i^{th} Walsh function ψ_i is an eigenfunction of the Hamming adjacency $\boldsymbol{A}$ corresponding to eigenvalue $\lambda_i = n - 2\langle i, i\rangle$ where $\langle i, i\rangle$ is the string inner product of i with itself: the order of the length-n bitstring representation of i. Thus in the case of the Walsh functions, we have

$$\boldsymbol{S}^{(r)}\psi_i(x) = \gamma_i^{(r)}\psi_i(x),$$

where

$$\begin{aligned}
\gamma_i^{(r)} &= \left(\frac{n-2\langle i,i\rangle}{r}\right)\gamma_i^{(r-1)} - \left(\frac{n-r+2}{r}\right)\gamma_i^{(r-2)},\\
&\vdots\\
\gamma_i^{(1)} &= (n - 2\langle i,i\rangle),\\
\gamma_i^{(0)} &= 1.
\end{aligned}$$

Note that this recurrence is exactly equivalent to an identity for the well-known Krawtchouk polynomials [Kra29]

$$\gamma_i^{(r)} = \mathcal{K}_r(\langle i,i\rangle, n),$$

which has the closed form

$$\gamma_i^{(r)} = \mathcal{K}_r(\langle i,i\rangle, n) = \sum_{j=0}^{r}\binom{\langle i,i\rangle}{j}\binom{n-\langle i,i\rangle}{r-j}(-1)^j. \tag{4.10}$$

The first moment of f^c (i.e., the c^{th} moment of f) over the sphere of radius r around an arbitrary point x can be calculated using a function corresponding to the image of f^c under the linear map $\boldsymbol{S}^{(r)}$.

Theorem 4.2. *Fix c and k. Let f be any k-bounded pseudo-Boolean function. Let $S^{(r)}(x)$ be a sphere of radius r around an arbitrary state x. The quantity $\mu_c(S^{(r)}(x))$ (the c$^{\text{th}}$ moment of f over the sphere) can be expressed as a series containing a polynomial number of terms.*

Proof.

$$\begin{aligned}
\mu_c(S^{(r)}(x)) &= \frac{1}{|S^{(r)}(x)|} \sum_{y \in S^{(r)}(x)} f(y)^c \\
&= \frac{1}{|S^{(r)}(x)|} \boldsymbol{S}^{(r)} f^c(x) && \text{by (4.9),} \\
&= \frac{1}{|S^{(r)}(x)|} \boldsymbol{S}^{(r)} \sum_j \mathfrak{w}_j \psi_j(x) && \text{by (4.3),} \\
&= \frac{1}{|S^{(r)}(x)|} \sum_j \gamma_j^{(r)} \mathfrak{w}_j \psi_j(x) && \text{by Theorem 4.1,}
\end{aligned}$$

and since $\forall x \in \{0,1\}^n, |S^{(r)}(x)| = \binom{n}{r}$,

$$= \binom{n}{r}^{-1} \sum_j \gamma_j^{(r)} \mathfrak{w}_j \psi_j(x).$$

By Lemma 4.2 there are at most $\binom{c+m2^k-1}{c}$ terms in the series. □

Since a Hamming ball of radius r is a union over all spheres of radius at most r, the moment calculation can be easily generalized to Hamming balls.[6]

[6] This result also trivially applies to *any* set of points which is a union of spheres about x.

Theorem 4.3. *Fix c and k. Let f be any k-bounded pseudo-Boolean function. Let $B^{(r)}(x)$ be a Hamming ball of radius r around an arbitrary state x. The quantity $\mu_c(B^{(r)}(x))$ (the c^{th} moment of f over the ball) can be expressed as a series containing a polynomial number of terms.*

Proof.

$$\begin{aligned}
\mu_c(B^{(r)}(x)) &= \frac{1}{|B^{(r)}(x)|} \sum_{s=0}^{r} \sum_{y \in S^{(s)}(x)} f(y)^c \\
&= \left(\sum_{s=0}^{r} \binom{n}{s} \right)^{-1} \sum_{s=0}^{r} \sum_{j} \gamma_j^{(s)} \mathfrak{w}_j \psi_j(x) \\
&= \left(\sum_{s=0}^{r} \binom{n}{s} \right)^{-1} \sum_{j} \mathfrak{w}_j \psi_j(x) \left(\sum_{s=0}^{r} \gamma_j^{(s)} \right).
\end{aligned}$$

Again, by Lemma 4.2 there are at most $\binom{c+m2^k-1}{c}$ terms in the series. □

It immediately follows that central moments of the distribution of f over Hamming regions can be expressed in terms of these series.

Corollary. *Fix c and k. Let f be any k-bounded pseudo-Boolean function. Let X be a Hamming region (sphere or ball) of some radius around an arbitrary state. The c^{th} central moment of f over X can be expressed as a series containing a polynomial number of terms.*

Proof. This follows from the definition of central moments in terms of μ_c.

$$\sum_{i=0}^{c} \binom{c}{i} (-1)^{c-i} \mu_i(X) \mu_1(X)^{c-i}.$$

□

4.3.1 A polynomial-time algorithm for computing moments

We have shown that when a function has a sparse representation in the Walsh basis, the expression for fixed moments of that function over Hamming regions can be characterized as a series containing a polynomial number of terms involving products of the Walsh coefficients. Computationally, this means when given such a function, we can immediately construct the exact fixed moments over any Hamming region in time polynomial in n, even when the cardinality of the Hamming region in question is exponential in n.

Let us compute $\mu_c(B^{(r)}(x))$ for a function f. We first compute the nonzero Walsh coefficients of f and store them in a data structure W which is an array of (bitstring, value) pairs such that, for the j^{th} nonzero Walsh coefficient of f in some arbitrary order, $W[j] = (i, w_i)$. We shall assume that arrays are indexed from zero. Since f is k-bounded, this data structure can be constructed in polynomial time [HRW99]. The eigenvalue $\gamma_p^{(r)}$ corresponds to the closed expression for the Krawtchouk polynomial in Equation (4.10).

The ball moment $\mu_c(B^{(r)}(x))$ is computed as follows.

```
BallMoment($x, r, c, W$)

1  $vol \leftarrow 0$
2  $sum \leftarrow 0$
3  for $s \leftarrow 0$ to $r$
4      do $sum \leftarrow sum + $SphereSum$(x, s, c, W)$
5         $vol \leftarrow vol + \binom{n}{s}$
6  return $sum \,/\, vol$
```

In line 4 we must calculate the term corresponding to the sum of f^c over spheres of radius $s \leq r$. Let the function Tuples(c, d) return the set of all c-tuples over the index set $\{0, 1, \ldots, d-1\}$. The sum of f^c evaluated over a sphere of radius s around x can be computed as

```
SphereSum($x, s, c, W$)
1   if $c = 0$ return 1
2   $sum \leftarrow 0$
3   for each $q \in$ Tuples($c$, $length[W]$)
4        do $\mathfrak{w} \leftarrow 1$
5           $j \leftarrow (000\ldots0)$
6           for $\ell \leftarrow 0$ to $c - 1$
7                do $(i, w_i) \leftarrow W[q[\ell]]$
8                   $j \leftarrow j \oplus i$
9                   $\mathfrak{w} \leftarrow \mathfrak{w} \times w_i$
10          $sum \leftarrow sum + \left(\gamma_j^{(s)} \times \mathfrak{w} \times \psi_j(x)\right)$
11  return $sum$
```

If f can be expressed as the sum of $m \ll n^k$ subfunctions, since k and c are taken as constants, the time complexity of SphereSum is $O(m^c)$, i.e., there are at most $m2^k$ elements in W. Note that since multiplication and exclusive or are commutative operations, there are a large number of symmetries in the sum over all c-tuples. Therefore the efficiency of the outer loop in lines 3 to 10 may be improved further using combinatorial enumeration techniques to remove these symmetries. Since BallMoment must call SphereSum exactly r times, the former algorithm has a time complexity of $O(rm^c)$. These runtime bounds are typical for many combinatorial optimization problems such as Max-k-Sat and evolutionary models such as NK-landscapes where m is $O(n)$. In the general case, there can be at most $\binom{n}{k} = O(n^k)$ nonzero Walsh coefficients of order k so the worst case complexity for k-bounded pseudo-Boolean functions is $O(n^{ck})$ and $O(rn^{ck})$. Thus we have the runtime bounds claimed in the introduction.

This algorithm relies on direct knowledge of the Walsh coefficients. For many combinatorial optimization problems (such as Max-k-Sat, as we saw in Chapter 3) it is fairly simple to compute the Walsh coefficients directly. In a more general setting, if the Walsh coefficients are unknown, but f is still epistatically bounded by a constant k, the Walsh coefficients can be efficiently retrieved deterministically in $O(n^k)$ time [KP01], or stochastically with negligible error in $O(n^2 \log n)$ time [HW04]. If there are m nonzero Walsh coefficients, Choi et al. [CJK08] present an $O(m \log n)$ adaptive randomized algorithm for finding all of them with high probability.

In this chapter we have used the analysis presented in the thesis to characterize the distribution of of codomain values of f across Hamming regions using low moments retrieved by a method that takes advantage of the sparsity of f in an eigenbasis of the Hamming adjacency. In the next chapter, we will study exactly how these moments are related to the actual distribution of f over Hamming regions and use the results to bound and approximate the true distributions of f.

Chapter 5

Characterizing Distributions of Codomain Values over Local Regions

In the previous chapter, we investigated a method for efficiently computing moments of the distribution of codomain values of the objective function over "local" regions of $\{0, 1\}^n$. In this chapter,[7] we will study how these moments relate to the local codomain distribution itself. In particular, we are interested here in the *moment problem* of classical probability theory: given a set of moments, find a probability distribution that exhibits those moments.

András Prékopa [Pré90] originally studied the moment problem of distributions with finite support. Prékopa showed how to compute bounds on such a distribution using a linear programming approach. We will connect this theory to the theoretical work on search space modeling developed in earlier chapters. In particular, we develop a method for computing bounds on the distribution of codomain values of the objective function over local regions for a particular class

[7] Most of the work presented in this chapter was initially published in the proceedings of the Eleventh ACM SIGEVO Conference on Foundations of Genetic Algorithms [SWH11a].

of problems. Moreover, we show that Prékopa's linear programming approach can be adjusted to provide an heuristic approximation of the local region distribution for such problems which can be used to estimate the count of improving moves that lie within a region of the search space.

The distribution of codomain values of an objective function over the entire state set is essentially a measure that provides one with the count of states in a particular problem instance that evaluate to a given value of the objective function. In the Boolean satisfiability research community, this measure is called the *density of states* (borrowing nomenclature from statistical physics) and provides a fine characterization of the structure of a problem instances [EGS10]. For Max-k-Sat, this so-called density of states can be interpreted as the frequencies of codomain values of a function that counts the number of assignments to a given Boolean formula that unsatisfy a particular number of clauses. The density of states also has a strong connection to *model counting* since the extremal values of the distribution of the Max-k-Sat objective function correspond to the number of satisfying assignments to the corresponding Boolean formula. In general, the density is intractable to compute, and for Max-k-Sat is approximated by Monte Carlo methods such as Metropolis sampling [REA96, AER96] or flat histogram based MCMC methods [EGS10].

The distribution of codomain values over Hamming regions that we address in this chapter is a generalization of the density of states in Max-k-Sat since one can construct a Hamming region to cover the entire state set. Recall that our interest in such a *local region distribution* comes from the fact that local algorithms will be influenced by the statistics of the objective function over states that are near the current solution. For instance, consider a Hamming region around a particular state x. The codomain distribution over that region determines quantities such as the count of elements in the region with higher objective value than x, or even the count of the best values in the region. The approximations and bounds we present in this chapter are directly applicable to estimating and bounding these quantities.

5.1 The local region value distribution

Recall the definition of Hamming spheres (Definition 4.1) and Hamming balls (Definition 4.2). We define a *Hamming region* to be any subset of $\{0,1\}^n$ that is a Hamming ball or sphere. Equation (4.1) gives the definition of $\mu_c(X)$: the c^{th} moment of a real function f over a Hamming region $X \subseteq \{0,1\}^n$. In this section we will study how a set of moments is related to the distribution of values from the codomain of f across X. In the remainder of this chapter, we constrain the codomain of the objective function f to a finite set with linearly bounded cardinality:

$$f : \{0,1\}^n \to A,$$

where

$$A = \{a_0, a_1, \ldots, a_{q-1}\}$$

is a finite set of q elements where q is $\Theta(n)$. This asymptotic bound on the codomain size is important to subsequent analysis since we later characterize the distribution of codomain values as the solution to q linear equations in q unknowns.

Without loss of generality, let us impose a total order on A so that $i < j \implies a_i < a_j$. Assuming maximization, let

$$a^* = \max_{x \in \{0,1\}^n} f(x).$$

Given a Hamming region X, we define the measure

$$p_X : A \to [0,1],$$

where $p_X(a_i)$ is the probability that an element x chosen *uniformly at random* from X has objective function value $f(x) = a_i$. In this case, p_X is a *probability mass function* with support A. For a given distribution, the set of q discrete values $\{p_X(a_i)\}$ are called the *impulses* of the probability mass function.

We can thus define the codomain *value distribution* for f over the local region X as a function

$$V_X : A \to \mathbb{N},$$

where $V_X(a_i) = |X|p_X(a_i)$ is the number of states $y \in X$ such that $f(y) = a_i$. In this case, $V_{\{0,1\}^n}$ corresponds to the so-called *density of states* for k-SAT and MAX-k-SAT problems [AER96, REA96, EGS10].

The value distribution exactly characterizes the allocation of objective function values to states in the region X. Under the assumption of maximization, $V_X(a^*)$ is the number of optimal solutions in X, and

$$\sum_{a_i > f(x)} V_X(a_i)$$

is the number of states in the Hamming region with improving objective function value with respect to a point x. Thus, an approximation of this quantity can be used to estimate the number of optimal solutions in X and the number of states in X with improving objective function value.

We now show that when f has the constraints imposed above, the moments of f over X appear in a system of equations that determine the value distribution V_X.

5.1.1 Computing the exact value distribution

Consider a state y drawn uniformly at random from the Hamming region X. The value of f evaluated at y can be modeled as a random variable Z. Since each state

$y \in X$ has a probability $\frac{1}{|X|}$ of being selected, the expectation of Z raised to the c^{th} power can be written as

$$\begin{aligned}\mathbb{E}[Z^c] &= \frac{1}{|X|}\sum_{y \in X} f(y)^c \\ &= \mu_c(X) && \text{by (4.1).} \qquad (5.1)\end{aligned}$$

But $\mathbb{E}[Z^c]$ is, by definition, the c^{th} moment of the distribution of the random variable Z. Note that the distribution of Z is the above defined probability mass function p_X. Hence we can write

$$\mathbb{E}[Z^c] = \sum_{i=0}^{q-1} a_i^c p_X(a_i). \tag{5.2}$$

Putting together Equations (5.1) and (5.2) we have the following identity:

$$\mu_c(X) = \sum_{i=0}^{q-1} a_i^c p_X(a_i). \tag{5.3}$$

In other words, the c^{th} moment of f over X is equal to the c^{th} moment of the probability mass function p_X.

In general, a function f has an infinite number of moments. Let us consider the lowest $|A| = q$ moments of X:

$$\{\mu_0(X), \mu_1(X), \ldots, \mu_{q-1}(X)\}.$$

Using the identity in (5.3) we have the following system of q equations in q unknowns.

$$\sum_{i=0}^{q-1} a_i^j p_X(a_i) = \mu_j(X), \tag{5.4}$$

for $j = \{0, 1, \ldots, q-1\}$. We can construct the q-dimensional column vectors

$$\boldsymbol{p} = (p_X(a_0), p_X(a_1), \ldots, p_X(a_{q-1}))^\top,$$

and

$$\boldsymbol{\mu} = (\mu_0(X), \mu_1(X), \ldots, \mu_{q-1}(X))^\top.$$

If the $q \times q$ matrix

$$\boldsymbol{M}_{i,j} = \left((a_i)^j\right)^\top$$

is nonsingular, there is a unique solution $\boldsymbol{p}$ to

$$\boldsymbol{M}\boldsymbol{p} = \boldsymbol{\mu},$$

which defines the probability mass function for the Hamming region since $p_X(a_i) = \boldsymbol{p}_i$. The value distribution is then given by $V_X(a_i) = |X| p_X(a_i)$.

$\boldsymbol{M}$ belongs to a well-known class of matrices known as Vandermonde matrices. The determinant is

$$\det(\boldsymbol{M}) = \prod_{i<j} (a_j - a_i)\,.$$

The matrix is nonsingular if and only if all the values of a_i are distinct. In our case, since A is a set, all elements $a_i \in A$, by definition, are distinct so $\boldsymbol{M}$ always has an inverse and the above system of equations always has a unique solution. Hence, if we have q moments of the Hamming region, we can obtain exactly the probability mass function over X by solving the system. Since q is $O(n)$, the size of the linear system is polynomial in n, even if $|X|$ is exponential in n.

5.2 Sharply bounding the local distribution

In the foregoing, q moments of f over X are needed to characterize V_X. Thus we must be able to retrieve moments of *arbitrary order*. If $\mathsf{P} \neq \mathsf{NP}$, this is computationally difficult, as is captured by the following theorem.

Theorem 5.1. *In general, the calculation of V_X is #P-hard.*

Proof. Let $\mathcal{F}$ be a propositional 3-Sat formula with n variables and m clauses. Let $f : \{0,1\}^n \to \{0,1,\ldots,m\}$ give the number of clauses satisfied under an assignment. Note that f satisfies the conditions we have imposed.

Let X be a ball of radius n around an arbitrary assignment x. In other words, $X = \{0,1\}^n$ and $V_X(m)$ gives the number of satisfying assignments to $\mathcal{F}$, solving #3-Sat which is #P-complete [Val79]. □

A surprising corollary to Theorem 5.1 is the following.

Remark 5.1. *Unless* P = NP*, no* NP*-hard problem can be represented by an objective function $f : \{0,1\}^n \to A$ such that f is both epistatically bounded and the cardinality of the codomain A of f is $O(1)$.*

This follows immediately from Theorem 5.1 since for such a function f one could compute the moments of f over $\{0,1\}^n$ in time polynomial in n (since in this case c must be asymptotically bounded by a constant) and solve the $|A| \times |A|$ linear system corresponding to $V_{\{0,1\}^n}$. The solution to the problem could then be immediately deduced from $V_{\{0,1\}^n}$.

When q is $\Omega(n)$, given all q moments, it is, in principle, possible to solve the linear system in polynomial time. In this case, the computational intractability must arise in the calculation of the moments themselves. Indeed, one can see that since the calculation of $\mu_c(X)$ in Chapter 4 is exponential in c, the generation of all q salient moments must be exponential in n.

Thus, unless P = NP, we cannot hope to efficiently compute V_X in general. Given the results developed in Chapter 4, it is natural to ask the question, given *some* information from the low moments of the distribution, is it possible to make some mathematically rigorous statements about the distribution? In other words, if we have a set of moments of f over X $\{\mu_0(X), \mu_1(X), \ldots, \mu_{c^{\max}}(X)\}$ where $c^{\max} \ll q$, is it possible to characterize the distribution in some way?

5.2.1 Constructing bounding functions

Given only moments up to $0 < c^{\max} \ll q$ of the objective function f over a Hamming region X, we have the partial Vandermonde system where j in Equation (5.4) runs from 0 to $c^{\max}$. Algebraically, let $\boldsymbol{M}'$ be the $(c^{\max}+1) \times q$ *partial* Vandermonde matrix that consists of the first $c^{\max}+1$ rows of $\boldsymbol{M}$ and a *truncated* moment vector $\boldsymbol{\mu}' = (\mu_0(X), \mu_1(X), \ldots, \mu_{c^{\max}}(X))^\top$ consisting of the lowest $c^{\max}+1$ moments of f over X. Consider the partial Vandermonde system

$$\boldsymbol{M}'\boldsymbol{p} = \boldsymbol{\mu}'. \tag{5.5}$$

This system is underdetermined, so there are potentially infinite solutions. Furthermore, there is no guarantee that a solution to this system gives a valid probability mass function. A solution to (5.5) may contain elements that are meaningless as probabilities, i.e., lying outside of the unit interval.

We can, however, impose a set of reasonable constraints on the decision variables of the system. In fact, it is possible to pose the formulation of a solution to the partial Vandermonde system in Equation (5.5) in terms of a *linear programming problem* subject to constraints that arise from the moment equations.

$$\begin{array}{rl} \max & \boldsymbol{b}^\top \boldsymbol{z} \\ \text{s.t.} & \boldsymbol{M}'\boldsymbol{z} = \boldsymbol{\mu}', \\ & 0 \leq \boldsymbol{z}_i \leq 1. \end{array} \tag{5.6}$$

where $\boldsymbol{b}$ is a length q vector of coefficients and $\boldsymbol{z}$ is a vector of decision variables.

Since the probability mass function in question has finite, preassigned support, if we select the coefficient vector $\boldsymbol{b}$ carefully, we can construct bounds on the impulse values of the probability mass function [Pré90]. We will use this result to construct functions that sharply bound the *true* distribution of f over X. Moreover, these bounds will be valid, even when the set of available moments has cardinality significantly less than q. We now formalize this in the following theorem.

Theorem 5.2. *Let X be a Hamming region. Let $\boldsymbol{z}$ be the solution to the linear program in (5.6) with $\boldsymbol{b}$ being the standard j^{th} basis vector:*

$$\boldsymbol{b}_i = \delta_{ij}, \quad i = 0, 1, \ldots, q-1,$$

where δ is the Kronecker delta function. Then

$$V_X(a_j) \leq |X|\boldsymbol{z}_j.$$

Proof. By definition, $V_X(a_j) = |X|p_X(a_j)$ so it is enough to prove that $p_X(a_j) \leq \boldsymbol{z}_j$.

Suppose for contradiction that $p_X(a_j) > \boldsymbol{z}_j$. In other words, we have $\boldsymbol{p}_j > \boldsymbol{z}_j$. By definition, $\boldsymbol{z}$ is the unique solution that maximizes

$$\boldsymbol{b}^\top \boldsymbol{z} = \boldsymbol{z}_j$$

and satisfies the partial Vandermonde system $\boldsymbol{M'z} = \boldsymbol{\mu'}$. Now, consider the full Vandermonde system $\boldsymbol{Mp} = \boldsymbol{\mu}$. Since all equations in the partial system corresponding to $\boldsymbol{M'}$ are contained in the full system, it follows that $\boldsymbol{M'p} = \boldsymbol{\mu'}$ is also satisfied. But $\boldsymbol{p}_j > \boldsymbol{z}_j \implies \boldsymbol{b}^\top\boldsymbol{p} > \boldsymbol{b}^\top\boldsymbol{z}$, a contradiction that $\boldsymbol{z}$ maximizes the linear program corresponding to the partial Vandermonde system. □

Iteratively maximizing the linear program using the j^{th} standard basis vector for $j = 0, 1, \ldots, q-1$ thus generates an upper bound function for each of the impulses of the distribution function V_X. We define this upper bound function formally as follows.

Definition 5.1. *The upper bound function $UB_X : A \to \mathbb{R}$ is constructed as $UB(a_j) = |X|\boldsymbol{z}_j$ where $\boldsymbol{z}$ is the LP solution to (5.6) using the standard j^{th} basis vector for $\boldsymbol{b}$. By Theorem 5.2 we have*

$$V_X(a) \leq UB_X(a),$$

for all $a \in A$.

A lower bound can be analogously found by solving the corresponding minimization problem using j^{th} standard basis vectors.

Definition 5.2. *The lower bound function* $LB_X : A \to \mathbb{R}$ *is constructed as* $LB(a_j) = |X|\boldsymbol{z}_j$ *where* $\boldsymbol{z}$ *is the LP solution to the minimization version of (5.6) using the standard* j^{th} *basis vector for* $\boldsymbol{b}$*. By a result symmetric to that of Theorem 5.2 we have*

$$V_X(a) \geq LB_X(a),$$

for all $a \in A$*.*

5.2.2 Bounding the cumulative local distribution

The linear program approach to constructing bounding functions for V_X can be modified to bound the cumulative variant of the value distribution of f over X. We define the *cumulative codomain value distribution* of f over X as

$$C_X(a) = \sum_{a_i \leq a} V_X(a_i).$$

Intuitively, $C_X(a)$ counts the states in $y \in X$ such that $f(y)$ is at most a. Moreover, $|X| - C_X(a)$ is the complementary cumulative distribution function that counts the states $y \in X$ such that $f(y)$ is strictly greater than a. We now use the above approach to construct bounding functions UB_X^C and LB_X^C for the cumulative codomain value distribution of f over X.

Theorem 5.3. *Let* X *be a Hamming region. Let* $\boldsymbol{z}$ *be a solution to the linear program in (5.6) with* $\boldsymbol{b}$ *being the coefficient vector*

$$\boldsymbol{b}_i = \begin{cases} 1 & \textit{if } i \leq j, \\ 0 & \textit{if } i > j, \end{cases} \qquad i = 0, 1, \ldots, q-1,$$

for some $0 \leq j < q$*. Then*

$$C_X(a_j) \leq |X|\boldsymbol{b}^\top \boldsymbol{z}.$$

Proof. By definition,

$$C_X(a_j) = |X| \sum_{i=0}^{j} p_X(a_i),$$

so it is enough to prove that

$$\sum_{i=0}^{j} p_X(a_i) \leq \boldsymbol{b}^\top \boldsymbol{z}.$$

Suppose for contradiction that $\sum_{i=0}^{j} p_X(a_i) > \boldsymbol{b}^\top \boldsymbol{z}$. In other words, we have $\sum_{i=0}^{j} \boldsymbol{p}_i = \boldsymbol{b}^\top \boldsymbol{p} > \boldsymbol{b}^\top \boldsymbol{z}$. By definition, $\boldsymbol{z}$ is the unique solution that maximizes $\boldsymbol{b}^\top \boldsymbol{z}$ and satisfies the partial Vandermonde system $\boldsymbol{M'z} = \boldsymbol{\mu'}$. Now, consider the full Vandermonde system $\boldsymbol{Mp} = \boldsymbol{\mu}$. Since all equations in the partial system corresponding to $\boldsymbol{M'}$ are contained in the full system, it follows that $\boldsymbol{M'p} = \boldsymbol{\mu'}$ is also satisfied. But $\sum_{i=0}^{j} \boldsymbol{p}_i > \boldsymbol{b}^\top \boldsymbol{z} \implies \boldsymbol{b}^\top \boldsymbol{p} > \boldsymbol{b}^\top \boldsymbol{z}$, a contradiction that $\boldsymbol{z}$ maximizes the linear program corresponding to the partial Vandermonde system. □

We can also define a corresponding cumulative upper bound function formally as follows.

Definition 5.3. *The upper bound function* $UB_X^C : A \to \mathbb{R}$ *is constructed as* $UB^C(a_j) = |X|\boldsymbol{b}^\top \boldsymbol{z}$ *where* $\boldsymbol{z}$ *is the solution to (5.6) using the coefficient vector*

$$\boldsymbol{b}_i = \begin{cases} 1 & \text{if } i \leq j, \\ 0 & \text{if } i > j, \end{cases} \quad i = 0, 1, \ldots, q-1.$$

By Theorem 5.3 we have

$$C_X(a) \leq UB_X^C(a),$$

for all $a \in A$.

Again, a cumulative lower bound function can be constructed by solving the corresponding minimization problem.

Definition 5.4. *The lower bound function* $LB_X^C : A \to \mathbb{R}$ *is constructed as* $LB^C(a_j) = \boldsymbol{b}^\top \boldsymbol{z}$ *where* $\boldsymbol{z}$ *is the solution to the minimization version of (5.6) using the coefficient vector*

$$\boldsymbol{b}_i = \begin{cases} 1 & \text{if } i \leq j, \\ 0 & \text{if } i > j, \end{cases} \quad i = 0, 1, \ldots, q-1.$$

By a result symmetric to that of Theorem 5.3 we have

$$C_X(a) \geq LB_X^C(a),$$

for all $a \in A$.

5.2.3 Bounding the extremal values of f in X

The bounding functions on V_X given by Definitions 5.1 and 5.2 can be used to bound the maximum (resp. minimum) values of f within X. We call a bounding function *degenerate* if it is zero for all inputs $a \in A$.

Theorem 5.4. *Let* $a^{\max}$ *be the maximal value of* f *in a Hamming region* X. *Assume* UB_X *and* LB_X *are nondegenerate. Let*

$$a_{UB}^{\max} = \max\{i : UB_X(a_i) \neq 0\},$$

and

$$a_{LB}^{\max} = \max\{i : LB_X(a_i) \neq 0\}.$$

Then

$$a_{LB}^{\max} \leq a^{\max} \leq a_{UB}^{\max}.$$

Proof. By Theorem 5.2, $V_X(a)$ is bounded above by zero for all $a > a_{UB}^{\max}$. Hence there are no states x in X with $f(x) > a_{UB}^{\max}$ and thus $a^{\max} \leq a_{UB}^{\max}$. Furthermore, $V_X(a_{LB}^{\max}) > LB_X(a_{LB}^{\max}) > 0$. Thus there exists an $x \in X$ with $f(x) = a_{LB}^{\max}$ so the maximal value of f in X is at least $a_{LB}^{\max}$. Hence, $a_{LB}^{\max} \leq a^{\max}$. □

In the case that UB_X is degenerate, then X must be the empty set since the count of states occupying any codomain value is bounded above by zero. In the case that LB_X is degenerate, then we must resort to the degenerate lower bound on $a^{\max}$: that is $a^{\max}_{LB} = a_0$.

It is straightforward to bound the *minimal* value of f in X by taking the minimal nonzero impulses of UB_X and LB_X. However, in this case, the upper bound function gives the lower bound on the minimal value since we are bounding "from the left" in this case. Similarly, the lower bound function gives the upper bound on the minimal value since it bounds "from the right." See Figure 5.1 for an illustration. We formalize this in the following theorem.

Theorem 5.5. *Let $a^{\min}$ be the minimal value of f in a Hamming region X. Assume UB_X and LB_X are nondegenerate. Let*

$$a^{\min}_{LB} = \min\{i : UB_X(a_i) \neq 0\},$$

and

$$a^{\min}_{UB} = \min\{i : LB_X(a_i) \neq 0\}.$$

Then

$$a^{\min}_{LB} \leq a^{\min} \leq a^{\min}_{UB}.$$

Proof. By Theorem 5.2, $V_X(a)$ is bounded above by zero for all $a < a^{\min}_{LB}$. Hence there are no states x in X with $f(x) < a^{\min}_{LB}$ and thus $a^{\min} \geq a^{\min}_{LB}$. Furthermore, $V_X(a^{\min}_{UB}) > LB_X(a^{\min}_{UB}) > 0$. Thus there exists an $x \in X$ with $f(x) = a^{\min}_{UB}$ so the minimal value of f in X is at most $a^{\min}_{UB}$. Hence, $a^{\min}_{UB} \geq a^{\min}$. □

The degenerate cases are handled accordingly.

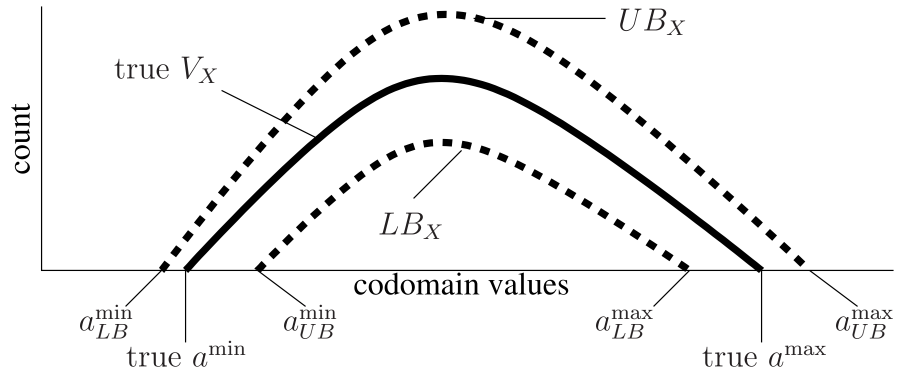


Figure 5.1: Bounding the extremal values of f over X using the nonzero impulses of UB_X and LB_X.

5.3 Estimating the count of improving moves in a local region

In this section we will use the solution of the linear program directly to estimate the number of improving moves in a local region. This quantity is important to local search algorithms because it determines how many states in nearby neighborhoods are candidates for selection. It could potentially be used to provide on-line estimations of the probability of a hill-climbing search or a mutation-only evolutionary algorithm performing a successful move within a certain number of steps. Moreover, it specifies a new way of comparing two arbitrary points in the search space by estimating the relative merits of exploration near each state. Finally, it can provide a sense of how rugged the region is.

We will now consider a general solution of the linear program to be an *approximation* of the true distribution. Without loss of generality, we shall assume maximization and imply the term “improving” is synonymous with “strictly greater objective function value.” Of course, the results also hold for minimization of f by reversing the appropriate inequalities.

Consider again the linear program formulation of the partial Vandermonde system specified in Equation (5.6). Any solution $\boldsymbol{z}$ to this system has a number of desirable properties. First, it can be interpreted as a probability mass function in the sense that its elements lie between 0 and 1 because of the constraints imposed by the linear program. Moreover, the zeroth moment $\mu_0(X) = 1$ corresponding to the first row of $\boldsymbol{M'}$ ensures the elements sum to unity. Thus we can define the *approximated* probability mass function $\hat{p}_X$ over X in terms of $\boldsymbol{z}$: $(\hat{p}_X(a_0), \hat{p}_X(a_1), \ldots, \hat{p}_X(a_{q-1}))^\top$. The approximated probability mass function $\hat{p}_X$ shares low moments with the exact solution to the original system in Equation (5.4). This is captured by the following.

Theorem 5.6. *Let $\boldsymbol{z}$ be a solution to the linear program in (5.6) with an arbitrary coefficient vector $\boldsymbol{b}$. Let $\hat{p}_X$ denote the probability mass function obtained from $\boldsymbol{z}$ as follows:*

$$\hat{p}_X(a_i) = \boldsymbol{z}_i, \quad i = 0, 1, \ldots, q-1.$$

Then $\hat{p}_X$ has the same j^{th} moment as the true distribution p_X for $0 \leq j \leq c^{\max}$.

Proof. Let $0 \leq j \leq c^{\max}$. The j^{th} moment of $\hat{p}_X$ is

$$\begin{aligned}\sum_{i=0}^{q-1} a_i^j \hat{p}_X(a_i) &= \sum_{i=0}^{q-1} \boldsymbol{M'}_{j,i} \boldsymbol{z}_i \\ &= \boldsymbol{\mu}'_j && \text{by (5.6),} \\ &= \mu_j(X).\end{aligned}$$

By (5.3), $\mu_j(X)$ is equivalent to the j^{th} moment of p_X: the true probability mass function over X. □

In other words, since mean and variance depend only on the first and second moments, for $c^{\max} \geq 2$, the approximated probability mass function given by solving the above linear program has the same mean and variance as the true probability mass function of the region. The codomain value distribution over X is approximately

$$\hat{V}_X(a) = |X|\hat{p}_X(a). \tag{5.7}$$

Similarly the cumulative value distribution is approximately

$$\hat{C}_X(a) = |X| \sum_{a_i < a} \hat{p}_X(a). \tag{5.8}$$

We define the function

$$\Xi_X(a) = |\{y \in X : f(y) \geq a\} \tag{5.9}$$

to be the true count of states y in X that have an objective function value strictly greater than a. For example, $\Xi_{B^{(r)}(x)}(f(x))$ is the count of improving states within Hamming radius r of x. It is easy to see that

$$\begin{aligned}\Xi_X(a) &= \sum_{a_i > a} V_X(a_i)\\ &= |X| - C_X(a).\end{aligned}$$

Hence an approximation for the number of improving states in X is given by

$$\hat{\Xi}_X(a) = |X| - \hat{C}_X(a). \tag{5.10}$$

5.3.1 Improving accuracy

The underdeteriminacy of the partial Vandermonde system introduces inherent inaccuracies into the approximation, especially when the number of available moments is very low. We are thus interested in ways to improve the accuracy of the approximation of p_X by $\hat{p}_X$.

5.3.1.1 Choosing the coefficient vector

The premise of Theorem 5.6 does not specify the coefficient vector $\boldsymbol{b}$. The result does not directly depend on the contents of this vector. However, one might expect different choices for $\boldsymbol{b}$ to yield approximations of differing accuracy. Indeed, it is not immediately clear what an appropriate choice for the coefficient vector might be.

We would expect impulse values occurring near the mean (that is, the values of a_i closest to $\mu_1(X)$) to be highest in the probability mass function. Hence a heuristic might be to maximize impulses near $\mu_1(X)$. Let ω be a "window size" parameter. Define also the index of the element nearest to the mean as $\zeta = \arg\min_i |a_i - \mu_1(X)|$ (recall we have imposed a total order on A). We can then define the coefficient vector as

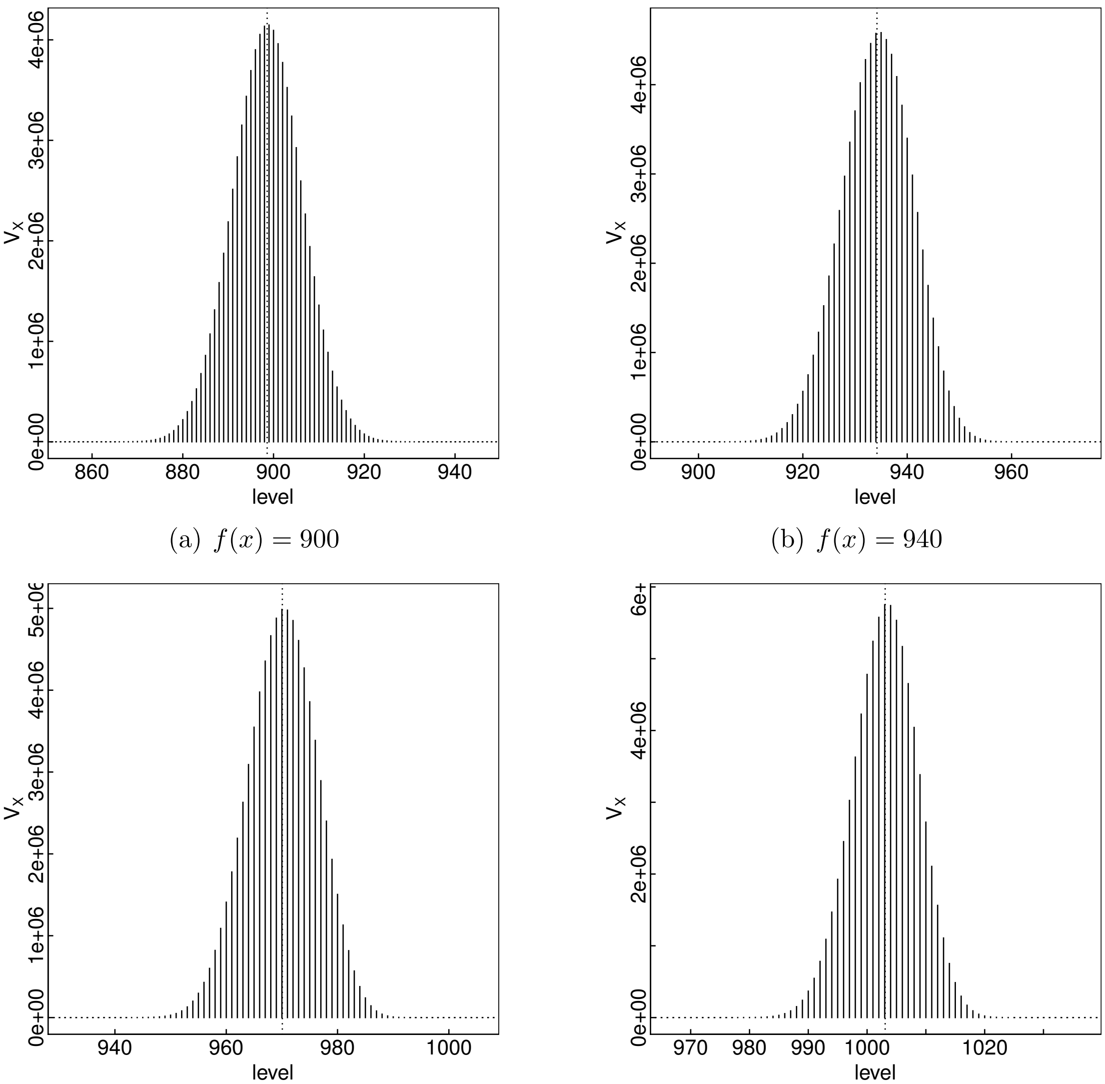
$$\boldsymbol{b}_i = \begin{cases} 1 & \text{if } |\zeta - i| \leq \omega, \\ 0 & \text{otherwise.} \end{cases}$$

Maximizing $\boldsymbol{b}^\top \boldsymbol{z}$ is akin to finding the approximated probability mass function in which impulses lying near the mean value are maximal. Determining more principled values for $\boldsymbol{b}$ remains a direction for future research.

5.3.1.2 Limiting impulse values

Since the linear program is very underconstrained, the above approach tends to result in sparse probability mass functions in which a large amount of mass is allocated to few impulses. Empirical data suggests that the nonzero impulse values tend to be unimodal and "clustered" around the mean, each with a limited mass. As an example, see Figure 5.2 which shows four different true codomain distributions V_X over Hamming regions of radius 5 sampled at different levels of the objective function on a Max-2-Sat instance. To address this and further refine the accuracy of the approximation we introduce an upper limit to the mass contribution of each impulse.

If $A \subset \mathbb{N}$ and $\hat{p}_X$ is reasonably well-behaved, then a suitable continuity correction would allow us to model $\hat{p}_X$ with a continuous distribution. Neglecting higher



(a) $f(x) = 900$

(b) $f(x) = 940$

(c) $f(x) = 980$

(d) $f(x) = 1020$

Figure 5.2: True V_X distributions over Hamming balls of radius 5 around points x sampled at different levels of the objective function. The objective function comes is defined by an instance of Max-2-Sat. In each plot, a broken vertical line denotes the mean value of the distribution.

moments, we note that a normal probability distribution with variance σ^2 has a maximum of $\frac{1}{\sqrt{2\pi\sigma^2}}$. Hence we might limit the maximum value of the impulses in $\hat{p}_X$ by

$$\left(2\pi(\mu_2(X) - \mu_1(X)^2)\right)^{-1/2}$$

to mitigate the sparse distribution of mass in the above approach.

Imposing this heuristic limit does not violate the constraints of the program; hence the result of Theorem 5.6 remains valid, and the solution is still a probability mass function with the same $c^{\max}+1$ moments as the true probability mass function p_X. We will presently find (in Section 5.3.2) that this limit empirically improves the accuracy of the approximated distribution function.

5.3.1.3 Incorporating constraints on higher moments

One advantage to the linear programming approach is that, provided we can compute upper and lower bounds on higher moments, we can incorporate this information as linear constraints in the LP approximation.

Let c be the maximum moment degree available and d be an arbitrary increment. Bounds on moments of higher degree can be added explicitly to the linear program as doubly bounded constraints:

$$LB(\mu_{c+d}(X)) \leq \sum_{i=0}^{q-1} a_i^j p_X(a_i) \leq UB(\mu_{c+d}(X)), \tag{5.11}$$

where $LB(\mu_{c+d}(X))$ and $UB(\mu_{c+d}(X))$ are respectively lower and upper bounds on the moment of order $c+d$. Obtaining the exact moments of higher degrees becomes computationally difficult (and is generally intractable by Theorem 5.1). However, if bounds on higher moments can be efficiently obtained, they may be incorporated into the approximation in this way.

We now impose some mild restrictions on the codomain A of f and show how to calculate some trivial upper and lower bounds on higher moments. First, we will assume $\forall a_i \in A, a_i \geq 0$. Since A is a finite set with cardinality linear in the problem size, we can impose this condition without loss of generality since the evaluation of f can always be shifted by an appropriate constant. Before deriving upper and lower moment bounds, we prove the following preparatory lemma that establishes a useful property for the theorem to follow.

Lemma 5.1. *Let X be a Hamming region. As long as there exist at least two states $x_1, x_2 \in X$ with $f(x_1) \geq 1$ and $f(x_2) \geq 1$, then, for $c, d \geq 1$,*

$$\frac{1}{|X|}\sum_{y\in X} f(y)^c \left(\sum_{z\in X\setminus\{y\}} f(z)^d\right) \geq \mu_c(X).$$

Proof. Since either $x_1 \in X \setminus \{y\}$ or $x_2 \in X \setminus \{y\}$, we have $\sum_{z\in X\setminus\{y\}} f(z)^d \geq 1$. □

The conditions for the lemma are relatively weak since, if necessary, we can shift f without altering the total order on A. We are now ready to give an upper bound on moments of degree $c + d$.

Theorem 5.7. *Let X be a Hamming region with at least two states $x_1, x_2 \in X$ such that $f(x_1) \geq 1$ and $f(x_2) \geq 1$. Let $d \geq 1$. Then,*

$$\mu_{c+d}(X) \leq |X|\mu_c(X)\mu_d(X) - \mu_c(X).$$

Proof.

$$\begin{aligned}
\mu_c(X)\mu_d(X) &= \left(\frac{1}{|X|}\sum_{y\in X} f(y)^c\right)\left(\frac{1}{|X|}\sum_{y\in X} f(y)^d\right) \\
&= \frac{1}{|X|^2}\sum_{y\in X} f(y)^{c+d} + \frac{1}{|X|^2}\sum_{y\in X} f(y)^c \left(\sum_{z\in X\setminus\{y\}} f(z)^d\right) \\
&= \frac{1}{|X|}\mu_{c+d}(X) + \frac{1}{|X|^2}\sum_{y\in X} f(y)^c \left(\sum_{z\in X\setminus\{y\}} f(z)^d\right).
\end{aligned}$$

Rearranging terms and multiplying by the cardinality of X we have

$$|X|\mu_c(X)\mu_d(X) - \frac{1}{|X|}\sum_{y\in X} f(y)^c \left(\sum_{z\in X\setminus\{y\}} f(z)^d\right) = \mu_{c+d}(X),$$

and by Lemma 5.1, $|X|\mu_c(X)\mu_d(X) - \mu_c(X) \geq \mu_{c+d}$. □

We can also derive the following trivial lower bound.

Theorem 5.8. *Let X be a Hamming region such that for all $x \in X$, $f(x) = 0$ or $f(x) \geq 1$. Let $d \geq 1$. Then,*

$$\mu_{c+d}(X) \geq \mu_c(X).$$

Proof. Since for all $x \in X$, $f(x)^{c+d} \geq f(x)^c$ we immediately have

$$\frac{1}{|X|} \sum_{y \in X} f(y)^{c+d} \geq \frac{1}{|X|} \sum_{y \in X} f(y)^c,$$

which proves the claim. □

Again, the conditions for the theorem are relatively weak since 1.) Domains where $A \subset \mathbb{N}$ already satisfy them, and 2.) The elements of A can be appropriately shifted without changing the total order. These bounds can be added to the linear program in the manner mentioned at the beginning of the section.

5.3.2 Numerical results

In order to assess the accuracy of the approximation devised above, it is necessary to perform a number of computational experiments. We divide this section into two parts. In the first part, we observe the dependence of accuracy on the accuracy-improving measures introduced above. In the second part, we compare the predicted number of improving moves with the actual number of improving moves.

We report all results on the maximum k-satisfiability (Max-k-Sat) domain which has, as we have already seen in Chapter 2, a k-bounded pseudo-Boolean objective function. In this case, as we proved in Theorem 5.1, unless $\mathsf{P} = \mathsf{NP}$, it is intractable to generate the true value distribution over all Hamming regions since such a quantity yields a solution to the decision problem.

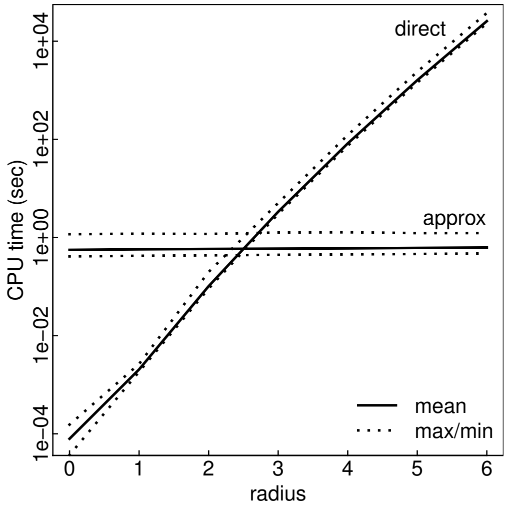


Figure 5.3: A comparison of time (in seconds) to exhaustively compute true distribution and time to perform LP approximation as a function of ball radius. The y-axis is on a logarithmic scale.

Therefore, given a Hamming region, we construct the true distribution by a direct count of states at each objective function value in the region and compare it with the approximated distribution. Of course this limits the comparison to computationally manageable regions. Figure 5.3 illustrates this with a logarithmic plot of CPU time in seconds necessary to compute the true distribution as a function of Hamming ball radius on a 100 variable MAX-k-SAT instance. The required time is directly proportional to the cardinality of the Hamming ball which is exponential in the radius. As a comparison, we also plot in Figure 5.3 the time required to perform the LP approximation of the distribution. To solve Equation (5.6) we used the GNU Linear Programming Kit (GLPK) using a simplex-based LP solver [Mak08]. While the time to compute the true distribution increases to over 20 minutes for each Hamming region, the time to perform the LP approximation remains less than a second on average. This means it becomes intractable to compare the approximation accuracy for all radius values on nontrivial instances. However, we conjecture that, in the case of a Hamming ball, the approximation accuracy remains stable with increasing radius, or even possibly improves.

As a test set, we use the 10 instance `s2v100c1200` Max-2-Sat benchmark set from the MAXSAT-2009 competition.[8] Each instance contains 100 variables and 1200 clauses. Each approximation is evaluated over Hamming balls of fixed radius $r = 5$. Thus the calculations are performed over regions containing 79375496 states.

5.3.2.1 Effects on accuracy

In order to compare how well a particular approximated distribution fits the true distribution, we define the (normalized) measure of absolute error as

$$\varepsilon = \frac{1}{q}\sum_{i=0}^{q-1} |c_X(a_i) - \hat{c}_X(a_i)|.$$

Note that $0 \leq \varepsilon \leq 1$ and measures the extent to which the two cumulative probability distribution functions disagree (see Figure 5.4). The ε metric has a loose similarity to the Kolmogorov-Smirnov statistic which measures the maximum deviation between two (continuous) cumulative distributions.

We plot the actual vs. approximated cumulative distribution function in Figure 5.5(a) for a radius 5 Hamming ball around a random point sampled from a particular instance from the benchmark set (the results are consistent across instances). The approximation is calculated using a truncated moment vector of the first four moments of the region

$$\boldsymbol{\mu}' = (\mu_0(X), \mu_1(X), \mu_2(X), \mu_3(X))^\top,$$

each generated using the algorithm on page 4.3.1 of Chapter 4. The approximation reported here also uses the heuristic impulse limit based on the second moment

[8] `http://www.maxsat.udl.cat/09/`

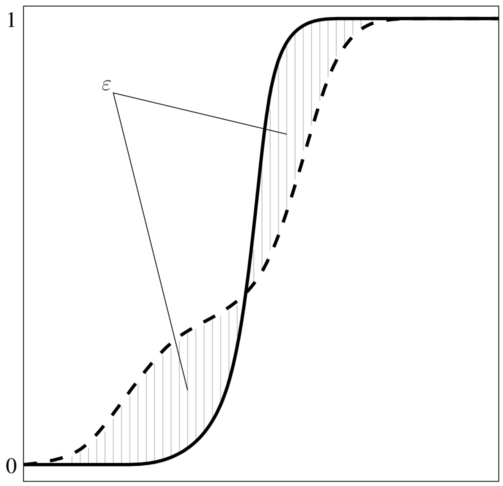
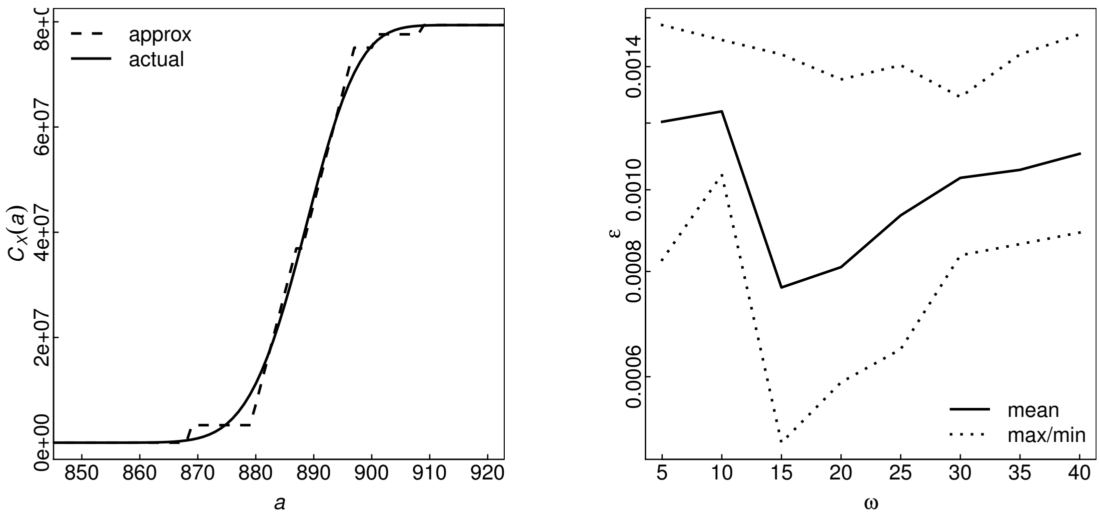


Figure 5.4: Illustration of the ε measure for two hypothetical cumulative distribution functions. ε measures the shaded area: the extent to which two distribution functions disagree.



(a)

(b)

Figure 5.5: (a) Cumulative value distribution on a single Max-2-Sat instance: actual vs. approximated over region of radius 5. (b) Dependence of approximation accuracy on window size for Max-2-Sat benchmark set `s2v100c1200`. The y-axis is on a logarithmic scale.

(discussed above on page 100), and incorporates the upper and lower bounds on moments $\mu_4(X)$, $\mu_5(X)$, and $\mu_6(X)$ (discussed on page 102). The window was set to $\omega = 20$. The measured ε value is approximately 7.47×10^{-5}.

To determine the dependence of approximation accuracy on window size, we varied the window size from

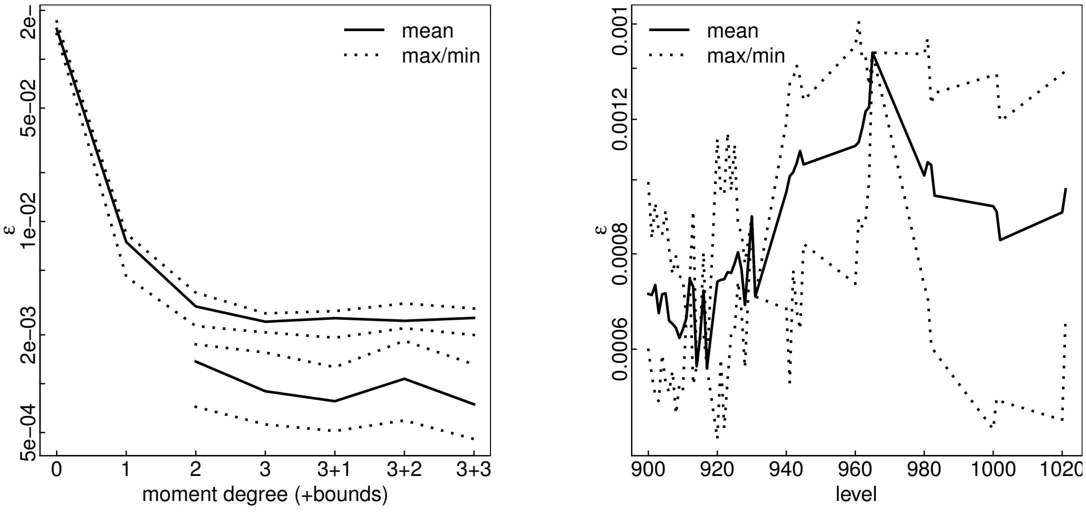
$$\omega = 5, 10, \ldots, 40.$$

For each unique ω value, we sampled 10 states from each of the 10 instances. For each state we compute the ε for the approximation (using the current ω value) with respect to the actual value distribution (obtained exhaustively). Figure 5.5(b) shows that the accuracy as a function of window size appears to tend toward a minimum at $\omega = 15$.

To determine the dependence of approximation accuracy on the length of the moment vector, heuristic impulse limiting (discussed on page 100), and added constraints on higher moments (discussed on page 102), we repeat the experiment, holding the window size at 15 and varying the number of moments used (1 to 4), and the bounds on higher moments. We performed the experiments with and without heuristic impulse limiting. The results are displayed in Figure 5.6(a). As expected, the more moments, the more accurate the approximation. The higher moment bounds, however, do not appear to produce a strong effect. Clearly, the heuristic impulse limit improves the approximation accuracy in this case.

To determine the dependence of approximation accuracy on the value of the centroid, we select a representative instance (`s2v100c1200-1`) and measure the approximation accuracy for a number of different centroid states at varying objective function levels.

Since arbitrarily low objective function values are somewhat extraneous in the Max-k-Sat domain (at least from the perspective of optimization), we limit our investigation to a range of objective function values that run from the average objective function value of the instance to near-optimal values. When each clause is exactly of length k, it is easy to show, by linearity of expectation, a random



(a)

(b)

Figure 5.6: (a) Dependence of approximation accuracy on moment degree (and bounds on higher moments) for Max-2-Sat benchmark set `s2v100c1200`. Top lines are *without* heuristic impulse limit, bottom lines are *with* heuristic impulse limit (note the heuristic impulse limit requires the second moment). The y-axis is on a logarithmic scale. (b) Dependence of approximation accuracy on centroid value for Max-2-Sat instance `s2v100c1200-1`. Expected value of a random solution 900, best value 1031. The y-axis is on a logarithmic scale.

assignment satisfies a clause with probability $(2^k - 1)/2^k$. Therefore, the average objective function value for the 1200 clause Max-2-Sat instance `s2v100c1200-1` is $\frac{3}{4} \times 1200 = 900$. The optimal objective function value (found by a complete solver) of this particular instance is 1031.

In order to focus on pertinent levels of the objective function for this instance, we considered a set of seven target levels: 900, 920, 940, 960, 980, 1000, and 1020, which range from the random expectation value to near-optimal. For each target level, we performed 100 episodes of a local hill-climbing search to generate solutions at or above the target level. Each resulting solution was then used as a centroid in a Hamming ball of radius 5, the true and approximated value distributions were subsequently calculated, and the resultant ε was computed (see Figure 5.6(b)). Due to statistical noise, these results are somewhat hard to interpret. However, we do note that accuracy has a stronger trend toward the boundary values of the target value range.

In all cases, we note the very small ε values. We can thus conclude that the approximation is substantially accurate, at least with respect to the chosen metric.

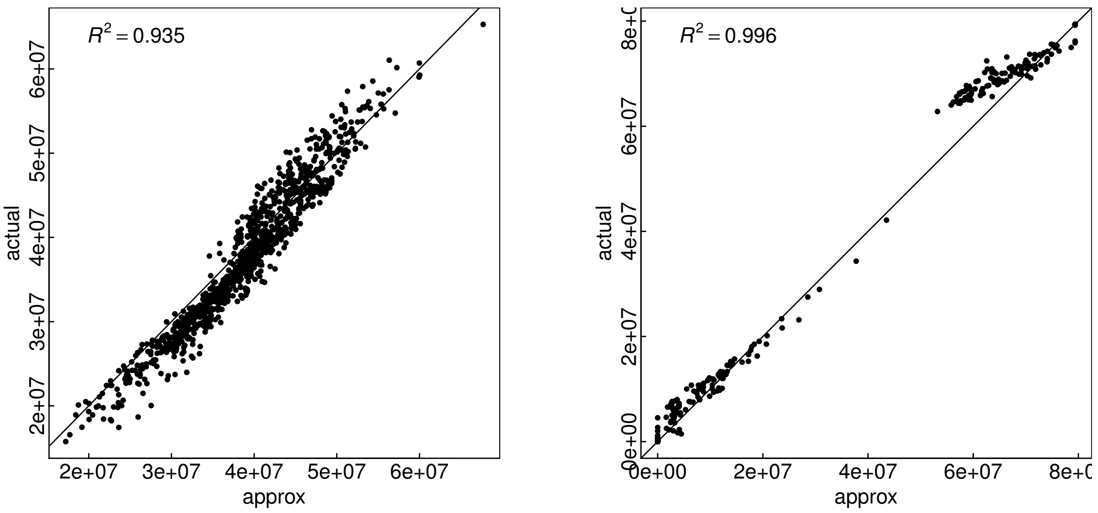
5.3.2.2 Estimation accuracy

To evaluate how well the model predicts the number of improving states in a region, we generated 100 random states on each of the 10 instances (1000 states total). For each generated state x, we computed the true number $\Xi_{B^{(5)}(x)}(f(x))$ of states with improving objective that lie in the Hamming ball of radius $r = 5$ about x. We then compute our approximation of this quantity using $\hat{\Xi}_X$ defined in (5.10). We used exact moments $\mu_0(B^{(5)}(x))$ through $\mu_3(B^{(5)}(x))$ retrieved by the algorithm introduced in Chapter 4. We incorporated further constraints on the linear program by imposing bounds on moments $\mu_4(B^{(5)}(x))$ through $\mu_6(B^{(5)}(x))$ using the method outlined above. We also employed the heuristic impulse limit described above.

We plot the actual number of improving states vs. the number predicted in Figure 5.7(a). Using the above settings, the approximation tends to slightly overpredict for lower values.

To evaluate the approximation for high-quality states, we sampled, using hill-climbing local search, 700 states from a single instance `s2v100s1200-1` whose objective function values lie in the interval $[900, 1020]$ (the global optimum is at 1031). Using each of these states as centroids, we enumerate a radius 5 Hamming ball and count the number of states lying in the ball with objective function value at least 90% of optimal. We compare this with the corresponding count predicted by the approximation in Figure 5.7(b).

The strong correlation and corresponding high empirical R^2 values suggest that



(a) 1000 random regions of radius 5 over s2v100c1200 Max-2-Sat benchmark set.

(b) 700 high-quality (90% of optimal) regions of radius 5 over s2v100c1200-1 Max-2-Sat instance

Figure 5.7: Number of actual improving states vs. number predicted.

the approximation provides a useful estimate of the number of improving states in a region around given states. This estimate is useful to practitioners since it provides a projection of how useful a region is in terms of exploration. In the next chapter, we will explore further how the results of this thesis might be used in practice.

Chapter 6

Two Applications

In this chapter we are interested in applying a number of insights gained from the theoretical framework developed in previous chapters. The central aim of the chapter is to demonstrate that formal analysis of the search space can be used to successfully inform the construction of existing algorithms in a more principled manner. The chapter is divided into two sections.[9] In Section 6.1 we study how the moment algorithm of Chapter 4 can be incorporated into a heuristic that guides hill-climbing search algorithms across plateaus. In Section 6.2 we will show how the framework can be used to control the mutation rate of a (1+1) evolutionary algorithm in such a way that the expected fitness of the resulting offspring is maximized.

6.1 Directing Search Across Plateaus

In Chapter 3, we devised a convenient basis function decomposition of the Max-k-Sat objective function. In Chapter 4, we conceived an algorithm for efficiently

[9] The work in Section 6.1 of this chapter was initially published in the proceedings of the Third International Symposium on Combinatorial Search [SHW10]. The work in Section 6.2 of this chapter has been accepted for publication in the proceedings of the Genetic and Evolutionary Computation Conference (GECCO 2011).

constructing the moments of an objective function given an appropriate basis decomposition. In this section, we will combine this knowledge to design a principled heuristic for guiding hill-climbing local search algorithms. We test the heuristic empirically and find that it improves the performance of GWSAT, a hill-climbing SAT solver, in certain regions of the MAX-k-SAT search space.

Local search algorithms have classically been characterized by iteratively accepting only neighbors with a *strictly* improving objective function evaluation. However, in the case of many combinatorial problems, it can be beneficial to also accept neighbors with *equal* evaluation in the interest of eventually discovering improving states. For example, on instances of k-satisfiability, Selman, Levesque, and Mitchell [SLM92] first discovered that accepting equal moves empirically improved the convergence time of local search. This was later studied in detail by Gent and Walsh [GW93b, GW93a] who concluded that once greedy hill-climbing search reached a state within 97% of the optimal, the majority of the moves become plateau moves. Furthermore, they also remarked that this latter phase of search constitutes, not hill-climbing, but a search for the occasional escape from a plateau. From a theoretical perspective, Mastrolilli and Gambardella [MG05] show that on unweighted MAX-k-SAT allowing local search to take equal moves results in an approximation ratio of 2/3 which is superior to basic local search's approximation ratio of 1/2.

The success of local search largely depends on how quickly it can follow a discrete "gradient" to move to better states. When equal moves are allowed, search must contend with *plateaus*: connected regions of the search space that are equivalent under the objective function. Plateaus pose a challenge to local search since they provide no gradient information to guide search.

Experimental data show that plateaus are a prominent search space feature in

MAX-k-SAT problems [Smy04]. In this section, we use the methods presented in this thesis to develop a "surrogate gradient" heuristic for MAX-k-SAT (on which clause length is bounded by a constant k, e.g., MAX-3-SAT) to help local search navigate plateaus. The goal of this application is to demonstrate that formal analysis of search space structure can be used to direct existing algorithms in a more principled manner than random walks. In particular, we apply the basis function decomposition of the MAX-k-SAT objective function to compute the mean value of states over large volumes of arbitrary radius around plateau states. We then use this statistic to direct search across plateaus. Finally, we empirically assess the utility of the heuristic in a study of its application to MAX-k-SAT problems. We first consider the hypothesis that the guidance offered by the surrogate gradient improves the time to escape a particular plateau. We then apply the heuristic to the plateau phase of a hill-climbing local search algorithm in order to determine whether this ultimately translates to faster convergence. We find the surrogate gradient to be advantageous in directing search through plateaus to near-optimal levels.

6.1.1 Background

Recall that Definition 3.1 designates a plateau P as a maximal set $P \subseteq \mathcal{X}$ such that, for all $x, y \in P$, there is a path $(x = x_1, x_2, \ldots, x_t = y)$ where, if $t > 1$, $x_{i+1} \in N(x_i)$ and $f(x_{i+1}) = f(x_i)$ for $i = 1, 2, \ldots, t$. By this definition, the set of plateaus partitions the state set $\mathcal{X}$ so each state x belongs to a unique plateau. The unique objective function value of all states in P is called the *level* of P.

Broadly speaking, hill-climbing local search algorithms progressively attempt to escape plateaus of constantly improving level. If no state on the plateau has a neighbor on a plateau of improving level, the plateau is *closed*. Assuming maximization, P is closed when

$$\forall x \in P,\ \neg\exists y \in N(x) : f(x) < f(y).$$

Regions of this nature have been called *local optima* [FCS97], *mesas* [Jon95], or *basins* [Yok97]. On the other hand, a plateau that has one or more states with improving neighbors is an *open* plateau and can eventually be escaped. Again, assuming maximization, P is open when

$$\exists x \in P,\ \exists y \in N(x) : f(x) < f(y).$$

Determining whether a plateau P is open or closed takes time proportional to $|P|$ in the worst case. Thus, on extensive plateaus, it is intractable to do so directly.

A hill-climbing local search algorithm can attempt to escape a plateau region by moving across it using a random walk or by performing systematic search. Either approach can be prohibitive since the number of states in a particular plateau can be exponential in the problem size [HK93, FCS97, Smy04]. The distribution of exit states (incident states with strictly improving objective function value) across a plateau further impacts how quickly it is escaped.

Hampson and Kibler [HK93] empirically studied plateau characteristics in MAX-k-SAT focusing on plateaus at near-optimal levels. They found that the number of plateaus in this region grows linearly with n (where n is the number of variables). They also found that the size of plateaus at better values grows exponentially with n, while the density of escapes decreases with n, producing an $O(n)$ increase in waiting time to escape plateaus. Thus the linear growth in waiting time on plateaus along with linear growth in the number of plateaus should produce a growth rate for a single hill-climbing episode that is roughly quadratic.

In an experimental study, Frank et al. [FCS97] found that escape density on open plateaus tends to decrease as plateau level approaches the optimal objective function value. This implies that plateaus nearer to optimal solutions become increasingly difficult to escape. They also found that escape density for near-optimal plateaus *increases* with constrainedness as measured by number of clauses. Smyth [Smy04] performed an empirical analysis of plateau structure on uniform random and structured problems. He found that closed plateaus tend to be smaller than open plateaus and that plateau characteristics correlate with instance hardness for stochastic local search algorithms.

A common strategy for escaping plateaus is to introduce a small amount of noise to the search process. For satisfiability problems, noise is added to the local search process in the form of a biased random walk [SKC94] which gives rise to the high performance WalkSAT algorithm [SKC96]. Algorithms from the WalkSAT family avoid plateaus by inverting variables that belong to unsatisfied clauses even if the move results in a *disimproving* value of the objective function. Within a clause, however, it may be necessary to break ties among a collection of variables if they have equal score. To address this, tie-breaking heuristics are based on the *dynamics* of the algorithm, such as how recently the variables have been flipped, e.g., WalkSAT-tabu, Novelty, and R-Novelty [MSK97]. Adding stochasticity to the latter two results in the probabilistically approximately complete variants: Novelty+ and R-Novelty+ [Hoo99]. Further refinements include diversification (Novelty++), deterministic greedy moves (G^{2}WSAT) [LH05], adaptive noise and combinations of these strategies [LWZ07]. In contrast to these tie-breaking heuristics, the surrogate gradient introduced in the following section is based solely on features of the search space.

6.1.2 A surrogate gradient

When a hill-climbing local search algorithm reaches a state with no improving neighbors, it must either interpret the state as a local optimum, or select among neighbors of equal value, if any exist. In the latter case, we would like the algorithm to make an informed decision about which equal neighbor to choose, in the absence of "gradient" information from the objective function. Recall from Equation (3.1) in Chapter 3 that the objective function for a MAX-k-SAT instance with n variables and m clauses can be written as a k-bounded pseudo-Boolean function $f : \{0, 1\}^n \to \{0, 1, \ldots, m\}$ that counts the number of clauses satisfied under an assignment corresponding to a binary string of length n.

We define the *surrogate gradient* function $\tilde{g}^{(r)} : \{0, 1\}^n \to \mathbb{R}$ to be

$$\tilde{g}^{(r)}(x) = \mu_1(B^{(r)}(x)),$$

or the first moment (i.e., mean value) of f over $B^{(r)}(x)$, or the Hamming ball of radius r around x as defined in Definition 4.2. In Section 4.3.1 we introduced an algorithm for efficiently computing exactly this expression, even when the size of the region under consideration is exponential in n.

An important observation is that $f(x) = f(y)$ *does not necessarily imply* $\tilde{g}^{(r)}(x) = \tilde{g}^{(r)}(y)$ and so $\tilde{g}^{(r)}$ might be used to delineate among a set of states with equal evaluation. For example, in Figure 6.1 we plot the empirical density function of $\tilde{g}^{(5)}$ evaluated over 35 equal neighbors of a particular state sampled at level 1060 (optimal is 1065) from SATLIB instance `uf250-1065-01`.

All other things being equal, a plateau state with escapes within radius r would be expected to have a higher average $\tilde{g}^{(r)}$ value than a state with no escapes within radius r. Thus $\tilde{g}^{(r)}$ could function as a heuristic for choosing more promising states among a set with equal evaluation. In other words, $\tilde{g}^{(r)}$ might be used as a surrogate "gradient" function to guide search across plateaus (see Figure 6.2).

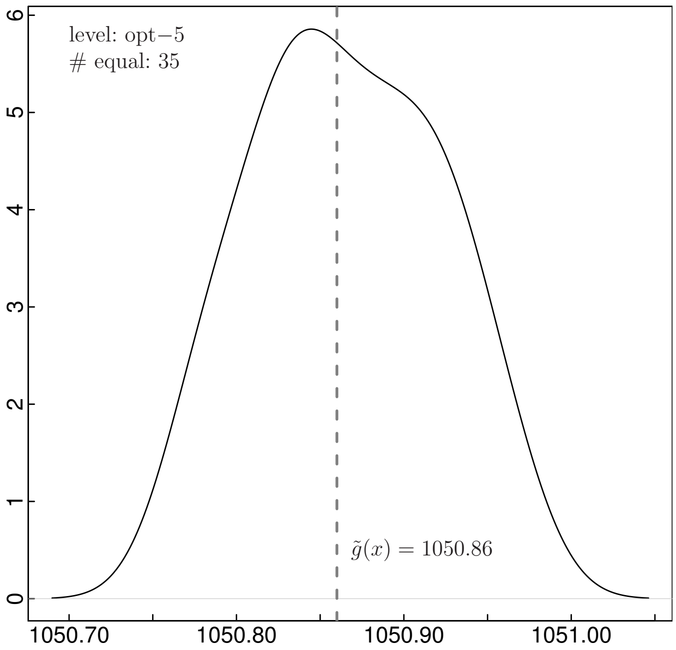


Figure 6.1: Empirical density function of $\tilde{g}^{(5)}$ evaluated over 35 equal neighbors of a plateau state x sampled from SATLIB instance `uf250-1065-01` where $f(x) = 1060$ (value of global optimum is 1065).

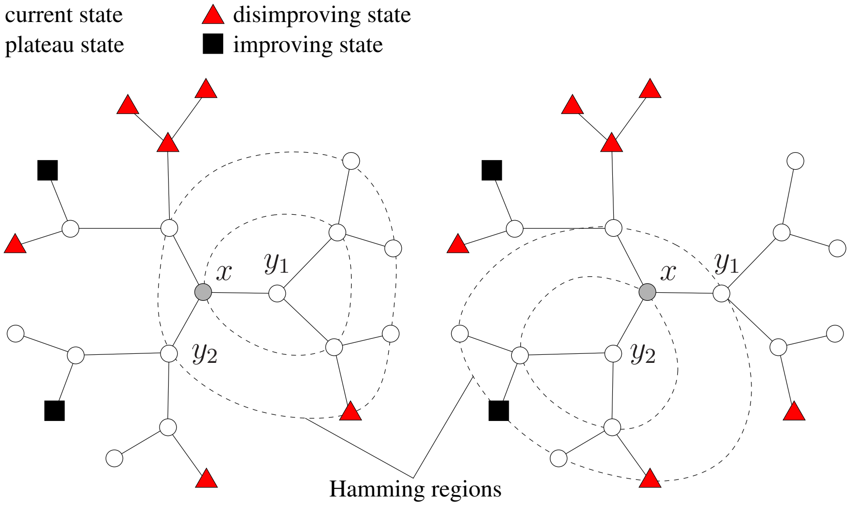


Figure 6.2: Schematic of surrogate gradient heuristic. Hamming ball of radius two (denoted by closed splines) around neighbors y_1 and y_2 of state x. Due to an improving state near y_2, it is likely that $\tilde{g}^{(2)}(y_2) > \tilde{g}^{(2)}(y_1)$.

```
DPS($x, f, \tilde{g}^{(r)}$)
$\tilde{g}_{best} \leftarrow \tilde{g}^{(r)}(x)$
while not done
        do $N \leftarrow \{y : \mathcal{H}(x, y) = 1\}$
           if $\exists y \in N : f(y) > f(x)$
              then return
           $E \leftarrow \{y \in N : f(y) = f(x)\}$
           Choose $x \in \arg\max_{z \in E} \tilde{g}^{(r)}(z)$
           if $\tilde{g}^{(r)}(x) \leq \tilde{g}_{best}$
              then $x \leftarrow$ random element of $E$
              else $\tilde{g}_{best} \leftarrow \tilde{g}^{(r)}(x)$
```

Figure 6.3: Directed plateau search process.

6.1.3 Directing search across plateaus

In this section we test the hypothesis that a surrogate gradient function that computes the mean value of the objective in a volume of search space of a given radius can direct a search algorithm to escape a plateau more quickly than random search alone.

6.1.3.1 Directed plateau search

On non-degenerate plateaus, the objective function is no longer useful for directing search to more promising states. Typically, hill-climbing search algorithms resort to a stochastic process by iteratively selecting equal neighbors at random until the plateau is escaped. We will instead use $\tilde{g}^{(r)}$ as a heuristic to direct search across plateau states by choosing plateau neighbors that lie in regions with lower average f. Given a state x, we perform local search maximizing $\tilde{g}^{(r)}$ using only neighbors with equivalent f values. The directed plateau search process (DPS) is given in Figure 6.3.

Until a plateau exit is found, plateau moves are chosen to maximize the surrogate gradient $\tilde{g}^{(r)}$. However, it is possible to reach a local optimum with respect to $\tilde{g}^{(r)}$. In this case, the search reverts to a random plateau walk by taking random plateau moves without regard to $\tilde{g}^{(r)}$ until a plateau move with an improving $\tilde{g}^{(r)}$ value is found.

6.1.3.2 Plateau escape results

Our hypothesis is that the additional information provided by $\tilde{g}^{(r)}$ allows DPS to escape plateaus more quickly on average than a random plateau walk (RPW). Starting from an initial state x, each plateau escape process generates a sequence of, not necessarily unique, states $(x = x_1, x_2, \ldots, x_t)$, called a *trace*, until a plateau exit is found, or the number of states exceeds some bound. We define $L_{dps}(c)$ to be the *trace length* of DPS on level c: the length of a trace beginning at a state x with $f(x) = c$ until stopping criteria are met and define $L_{rpw}(c)$ to be the length of a trace generated by a random plateau walk with initial state on level c.

If we choose states uniformly at random from a particular level c, we can characterize $L_{dps}(c)$ and $L_{rpw}(c)$ as random variables. So to test our hypothesis we must show that $L_{dps}(c)$ stochastically dominates $L_{rpw}(c)$. Sampling these random variables amounts to performing both a DPS and a random plateau walk from states on a level and measuring their trace lengths. To do so, we sampled 100 states each at levels opt-5, opt-4, opt-3, opt-2, and opt-1 where (opt$-c$ denotes the c^{th} level below the optimal) on all 1000 instances in the SATLIB benchmark set `uf100-430`. For each sampled state, we measured the trace length of both DPS and a random plateau walk. For the radius parameter we used a number of different values: $r = \{1, 2, 5, 10, 20\}$.

Note that if the initial state has an improving neighbor, the plateau can be immediately escaped by both processes (trace of length 1) so such data points are useless to our experiment and are removed from the data. Furthermore, the

r	1	2	5	10	20
mean trace	30.38	29.75	28.88	29.55	29.87
std. dev.	109.33	111.71	104.36	110.16	111.71

Table 6.1: Mean and standard deviation of DPS trace lengths for different radius values on levels opt−4 and opt−5 of `uf100-430`

maximum allowed trace length was set to 2000 states. We say the process *fails* if it does not escape the plateau within the allotted trace length. To control for states that may lie on closed plateaus, we remove from consideration states on which both the random walk and DPS process fail.

To test whether r has an effect on escape time, we measured the mean and standard deviation of the trace lengths on levels 4 and 5 of the `uf100-430` distribution. The results are shown in Table 6.1. The mean trace length for DPS appears not to significantly depend directly on radius. For the experiments in this paper, we will use $r = 5$.

The results for $r = 5$ on the `uf100-430` distribution are shown in Figure 6.4. The data come from a population of 10^5 states (100 states/instance). The distribution of each random variable is heavy-tailed, and follows an overdispersed Poisson distribution. Such a distribution can be modeled by a negative binomial distribution with parameters $\alpha = \beta = \sigma^{-2}$ where σ^2 is the variance. To test for stochastic dominance we perform the (nonparametric) sign test that $L_{rpw}(c) - L_{dps}(c)$ is, on average, greater than zero. For each level and each radius, we compute a p-value of less than 0.0001 when comparing to random plateau walk. We can thus conclude there is a statistically significant effect, and that DPS escapes plateaus more quickly on average than a simple random plateau walk. Statistics for the escape experiments are shown in Table 6.2. The number of times both methods failed (and hence were removed) increases sharply from 0.003% at level opt−5 to over

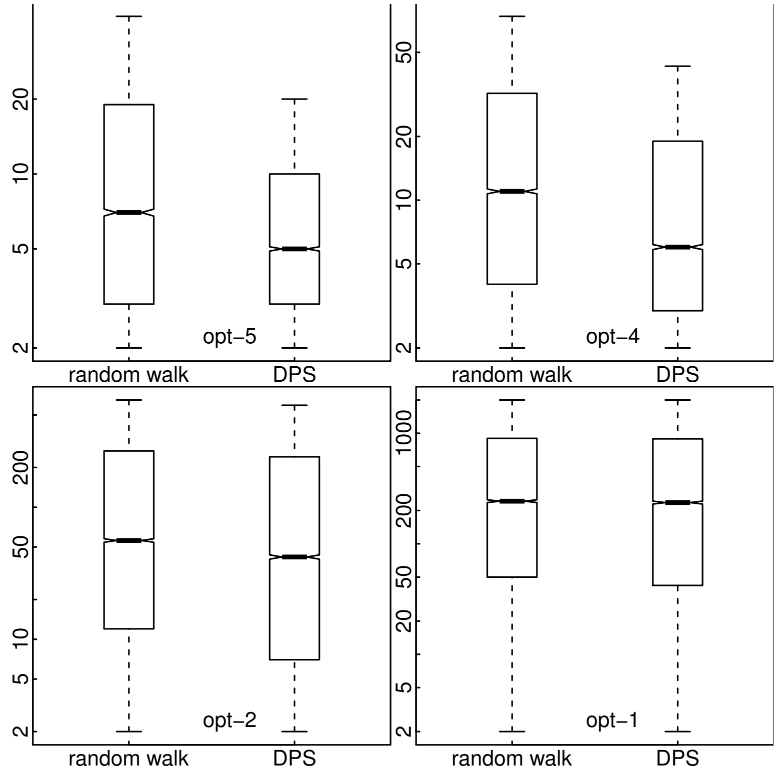


Figure 6.4: Plateau escape experiments (at radius 5) for levels opt−5, opt−4, opt−2, and opt−1 of `uf100-430` distribution. A sign test confirms statistical significance for each with $p < 0.0001$.

60% at the level directly below the optimal (c.f. rightmost column of Table 6.2). This corroborates the empirical findings of Frank et al. [FCS97] that escape density tends to decrease at the optimal level is approached.

6.1.3.3 Timing and efficiency

Recall from Section 4.3.1 the computation of the mean objective function value over $B^{(r)}(x)$ (and thus $\tilde{g}^{(r)}(x)$) requires the Walsh coefficients of f and the sphere eigenvalues given by the Krawtchouk polynomials (4.10). These quantities can be computed off-line and stored. Moreover, if the value of $\tilde{g}^{(r)}(x)$ is already stored when DPS is called, for any $z \in N(x)$ the value of $\tilde{g}^{(r)}(z)$ can be obtained using a difference equation. This equation can be evaluated in time proportional to the

Level	**Alg.**	**Trace statistics**		**Run statistics**		
		med.	mean (sd)	% failed	mean % grad	% both failed
opt−5	random	8	23.19 (76.71)	0.03	-	0.003
	dps	5	17.89 (76.21)	0.05	61.02	
opt−4	random	11	44.23 (131.89)	0.14	-	0.072
	dps	6	34.78 (116.27)	0.07	54.72	
opt−3	random	20	115.58 (291.16)	0.96	-	0.974
	dps	11	103.21 (280.45)	0.87	44.09	
opt−2	random	57	283.34 (504.00)	4.16	-	9.991
	dps	43	271.91 (503.98)	4.15	29.93	
opt−1	random	248	574.92 (679.23)	10.94	-	60.572
	dps	230	561.42 (678.45)	10.69	14.18	

Table 6.2: Results for plateau escape experiments: trace length statistics, percentage of runs each method failed (i.e., reached the cutoff), and (for DPS) mean percentage of steps that utilized the surrogate gradient heuristic. We remove runs in which both methods failed, the percentage of which is listed in the final column.

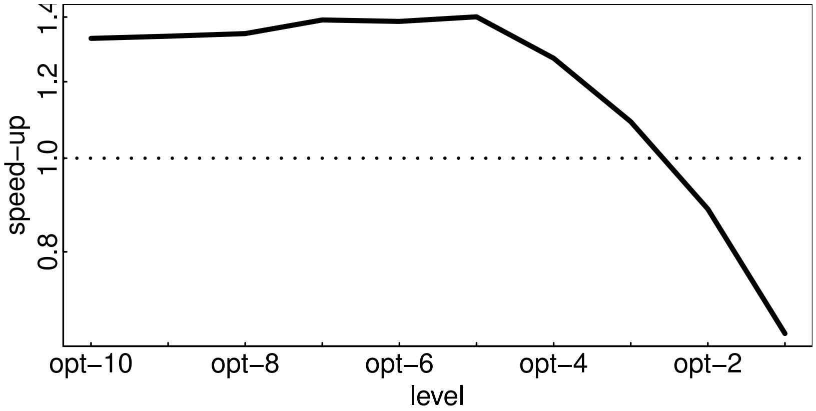


Figure 6.5: Median relative CPU time speed-up (DPS) for escaping best 10 levels of `uf100-430` distribution.

number of literals containing the variable which must be negated to transform x into z. Hence, there is a small overhead associated with calculating the surrogate gradient in each plateau search step. We would expect, however, that shorter trace lengths ultimately translate to faster escape time. To investigate this further, let $T_{dps}(c)$ denote the processing time needed by DPS to escape level c and $T_{rpw}(c)$ be the processing time needed by a random plateau walk. We measure the *relative speed-up* at level c as $\frac{T_{rpw}(c)}{T_{dps}(c)}$. We report the median relative speed-up for levels opt-10 up to opt-1 on the `uf100-430` distribution in Figure 6.5. On lower levels, when there is an improvement in trace length, we see this corresponds to an improvement in processing time. However, as the highest levels are approached, we see that the advantage of DPS is diminished (c.f. Figure 6.4), and the overhead for computing the surrogate gradient translates to a slow-down in processing time.

6.1.4 Improving the performance of hill-climbing search

We now consider the hypothesis that the expedition of plateau search time observed in the above experiment will ultimately translate to faster convergence time for hill-climbing search algorithms. An episode of hill-climbing local search can be seen as a process that escapes plateaus at progressively improving levels. Suppose a hill-climbing process is started from an arbitrary state x_0 and eventually reaches some state x^* with $f(x^*) > f(x_0)$ (recall we are maximizing). The waiting time (in terms of number of evaluations) between these two boundary states can be modeled as a sum over random variables which gives the time spent at each level between $f(x_0)$ and $f(x^*)$. Letting τ_c denote the number of evaluations performed at level c, we have the total waiting time

$$\Lambda = \sum_{c=f(x_0)}^{f(x^*)-1} \tau_c. \tag{6.1}$$

Since not all levels are visited during an episode of hill-climbing, let χ_c be the indicator random variable where

$$\chi_c = \begin{cases} 1 & \text{if search skips level } c, \\ 0 & \text{otherwise.} \end{cases}$$

If the hill-climbing local search randomly selects among equal neighbors, then we have

$$\tau_c = L_{rpw}(c)(1 - \chi_c).$$

By linearity of expectation the expected waiting time is

$$\mathbb{E}[\Lambda] = \sum_{c=f(x_0)}^{f(x^*)-1} \mathbb{E}[L_{rpw}(c)(1 - \chi_c)], \tag{6.2}$$

where $\mathbb{E}[\cdot]$ denotes random variable expectation. Instead of a random walk, if DPS is implemented at each level to direct search across plateaus, we can substitute $L_{rpw}(c)$ with $L_{dps}(c)$ in Equation (6.2). If we assume that the probability of skipping a level is invariant under plateau search dynamics (and we have no reason to believe otherwise), then given the results presented in the previous section, we would expect statistically shorter convergence times for these hill-climbing episodes when compared to the standard random walk. In other words, using DPS to escape plateaus should result in faster plateau escape times which ultimately translate into faster convergence.

6.1.4.1 Hill-climbing search results

To test the hypothesis that directed plateau search can speed up periods of hill-climbing local search, we incorporated the DPS process into the plateau phase of GWSAT: a variant of the randomized greedy hill-climbing local search algorithm in which an unbiased random walk is performed with probability p in each step [SK93]. Our reasoning for using GWSAT is that the random walk element is necessary for probabilistically departing closed plateaus, which still pose a problem to DPS.

Since we assess the effect of DPS on single episodes of hill-climbing, we set the MAX-TRIES parameter to 1. We set the MAX-FLIPS parameter of GWSAT to 10000 and used a walk probability of $p = 0.3$. To observe the effect of DPS on GWSAT's convergence to specific levels of the objective function, we performed both GWSAT and GWSAT with DPS (GWSAT-DPS) targeting different levels.

Run length distributions for levels opt−5, opt−3, opt−1, and opt−0 (level opt−0 being the optimal value) are plotted in Figure 6.6. To generate these run length distributions, we performed both searches 1000 times for each instance in `uf100-430`. This set contains 1000 instances so each empirical cumulative distribution function is generated from 10^6 data points. For these target levels, GWSAT-DPS dominates. In target level opt−0, there is a slight crossover around 2000 evaluations.

The SATLIB benchmark set we used contains filtered random uniform problems generated at the phase transition region (clause to variable ratio ≈ 4.3) and guaranteed to be satisfiable (using a filtering process). Such benchmarks are typically used in the context of the k-SAT decision problem. Typical MAX-k-SAT benchmarks are not satisfiable and are deep into the overconstrained phase. To investigate the impact of adding DPS to a hill-climbing algorithm on a typical MAX-k-SAT benchmark, we generated empirical run length distributions for benchmark

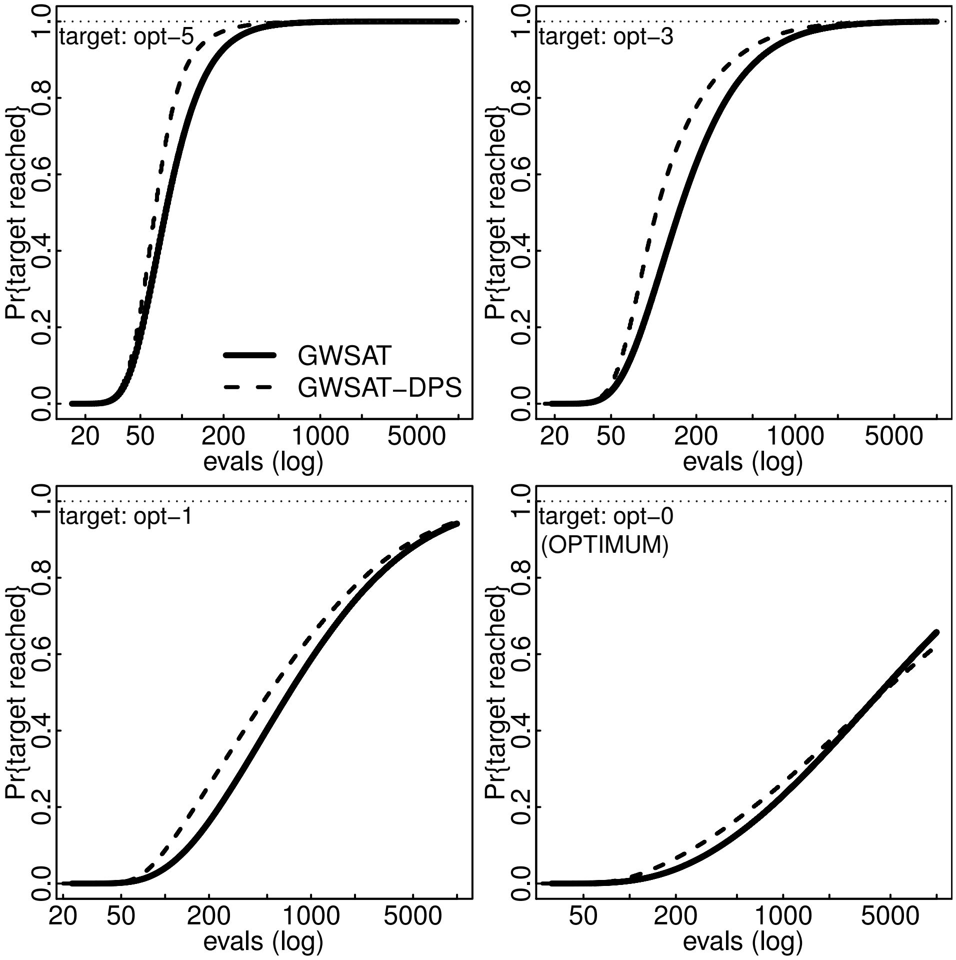


Figure 6.6: Empirical run length distributions for 1000 runs each on 1000 instance `uf100-430` distribution for four different target levels.

instances from the Fourth MAXSAT Evaluation in 2009.[10] We performed both searches 10000 times on each instance in the set `s2v100c850`. This set has exactly two literals per clause, i.e., Max-2-Sat, and a clause to variable ratio of 8.5. We use the same settings as above. Since each instance has a different optimal value, we plot the empirical run length distribution only for 10000 runs on two representative instances: `s2v100c850-01` and `s2v100c850-06`. These instances were selected because, of the ten instances in the set, DPS appears to perform best on the former, and worst on the latter when targeting the optimal solution.

The run length distributions for targeting the optimal solution (opt$-$0, which was found by a complete solver) generated from both instances appear in the upper half of Figure 6.7. As a reference, we also plot the empirical run length distribution for a state-of-the-art WalkSAT algorithm G^{2}WSAT [LH05]. The hill-climbing search with DPS clearly dominates on `s2v100c850-01` (top left of Figure 6.7) but performs relatively poorly on `s2v100c850-06` (top right of Figure 6.7). However, when targeting a suboptimal solution that only differs by one clause from the optimal, i.e., the level opt$-$1, the hillclimber with DPS dominates again for both instances (bottom left and right of Figure 6.7). Determining what causes the prominent discrepancy for targeting the optimal solution between these two instances remains a direction of future research.

6.1.5 Implications for search

The empirical results suggest that GWSAT augmented with DPS tends to dominate until the lowest level plateaus. A closer look at the number of evaluations necessary for convergence to levels opt$-$5 and opt$-$0 is given in Figure 6.8. The

[10] `http://www.maxsat.udl.cat/09/`

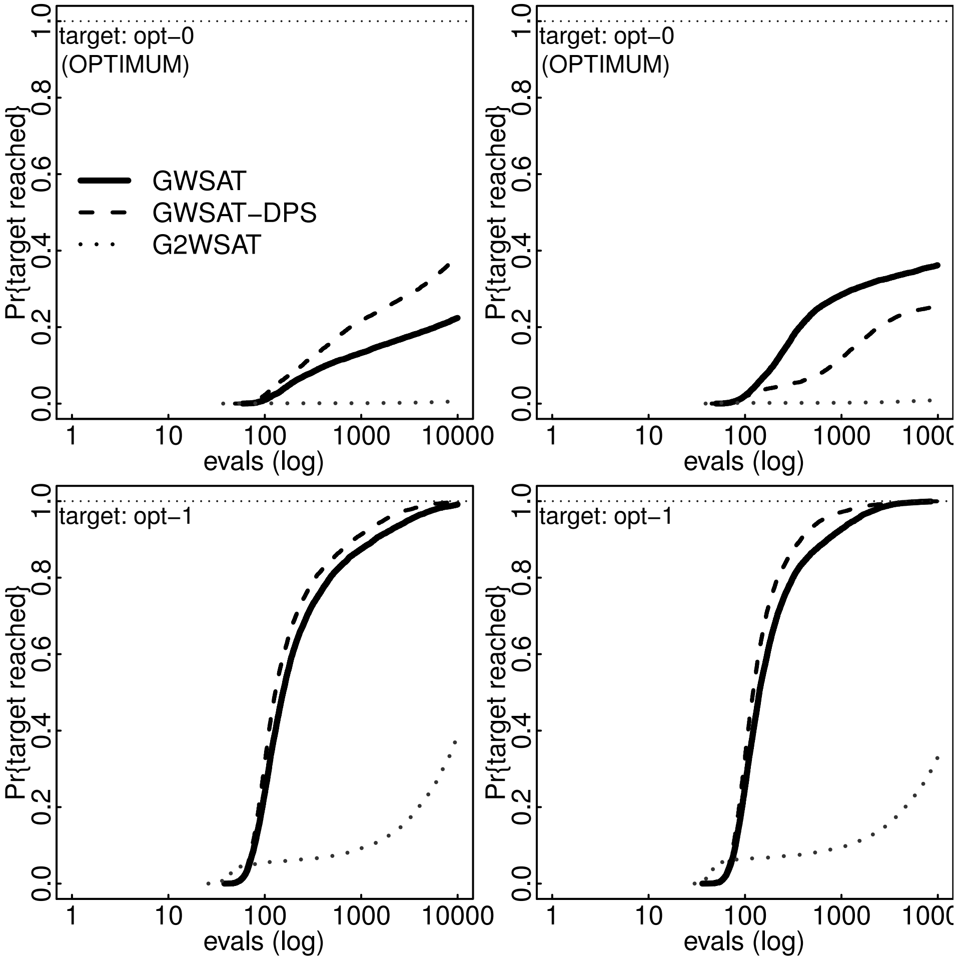


Figure 6.7: Empirical run length distributions for 10000 runs each targeting the optimal solution on `s2v100c850-01` [top left] and `s2v100c850-06` [top right], and targeting a suboptimal solution with a difference of only one clause from the optimal on `s2v100c850-01` [bottom left] and `s2v100c850-06` [bottom right].

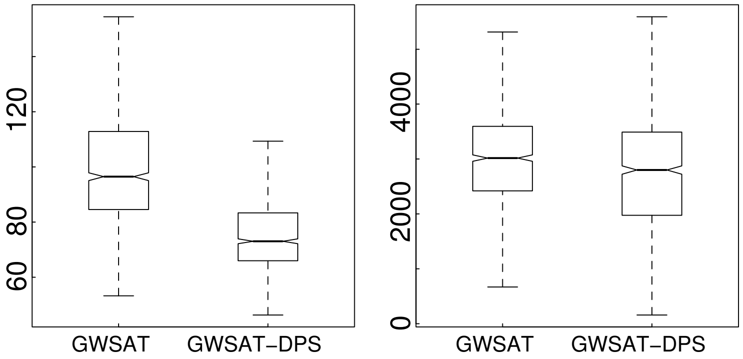


Figure 6.8: Comparison of number of evaluations (log scale) to find level opt−5 (left) and level opt−0 (right) on `uf100-430`.

effect is far less dramatic when converging to the optimal level.

The observed traces for DPS at each level suggest that on near-optimal plateaus, the surrogate gradient is not followed as often. For instance, in Table 6.2, at level opt−1, an average of approximately 14% of the steps are directed whereas at level opt−5, just over 61% of the steps are directed. This is likely due to a loss in resolution of the surrogate gradient at lower levels. As the extremal value of f is approached, the local volume will be composed mainly of states on disimproving levels. Furthermore, many plateaus at this level are closed; GWSAT-DPS will have to rely on GWSAT's walk probability rather than plateau search. These characteristics affect a heavy tailed distribution in waiting time on plateaus at near optimal levels. These factors statistically obscure the gains in convergence obtained by DPS on other levels. In Figure 6.9 we plot the empirical density of the surrogate gradient over equal neighbors for points sampled near the global optimum (at levels opt−4, opt−3, opt−2, and opt−1).

In order to observe the convergence profile more clearly, we plot the mean convergence (Δ from optimal value as a function of evaluation) for GWSAT and GWSAT-DPS over the distribution `uf250-1065` in Figure 6.10. The gains begin occurring around level opt−20 and vanish near level opt−3. as long random walks

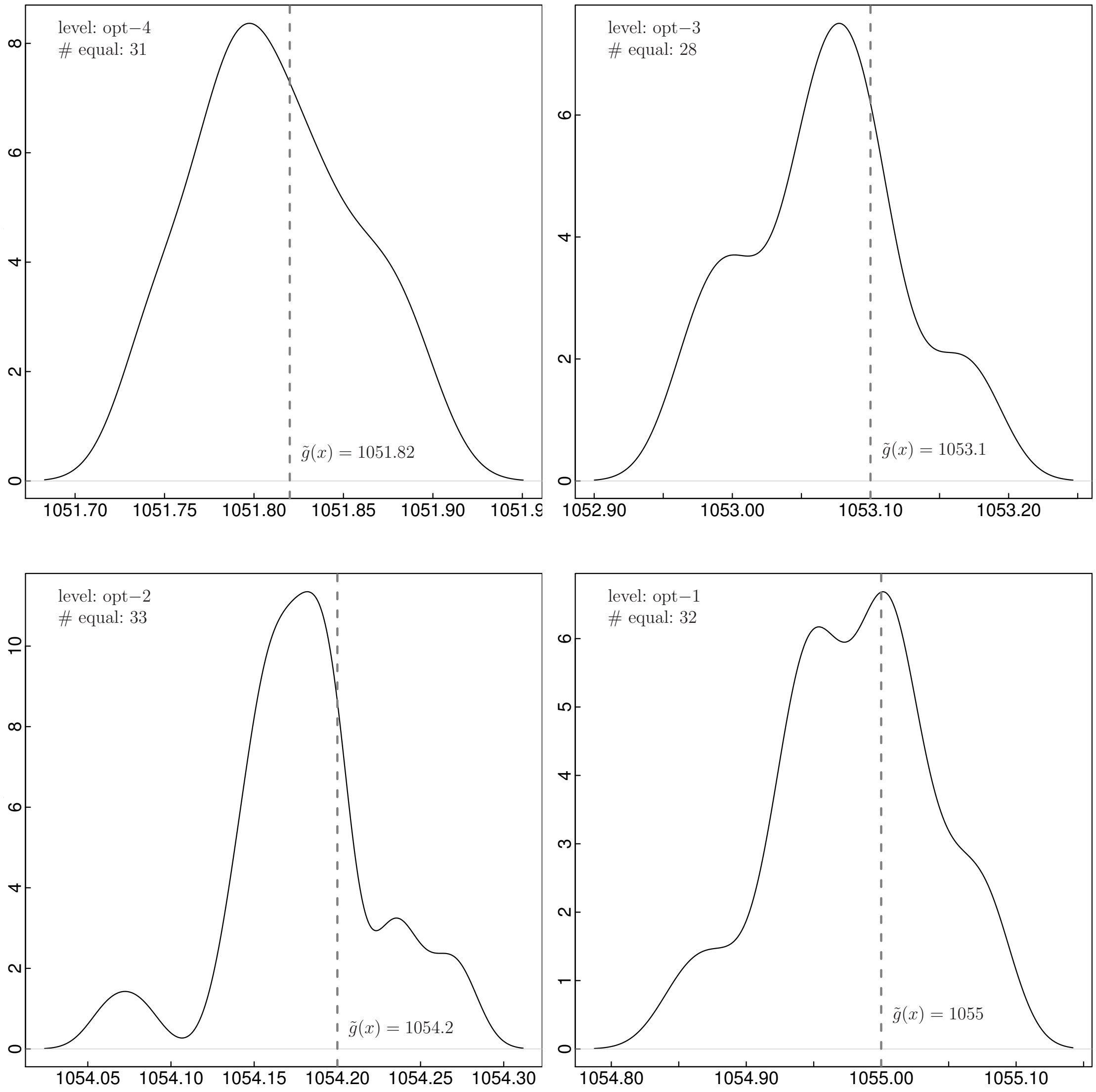


Figure 6.9: Empirical density function of $\tilde{g}^{(5)}$ evaluated over equal neighbors of plateau states x sampled from SATLIB instance `uf250-1065-01` at levels close to the optimal.

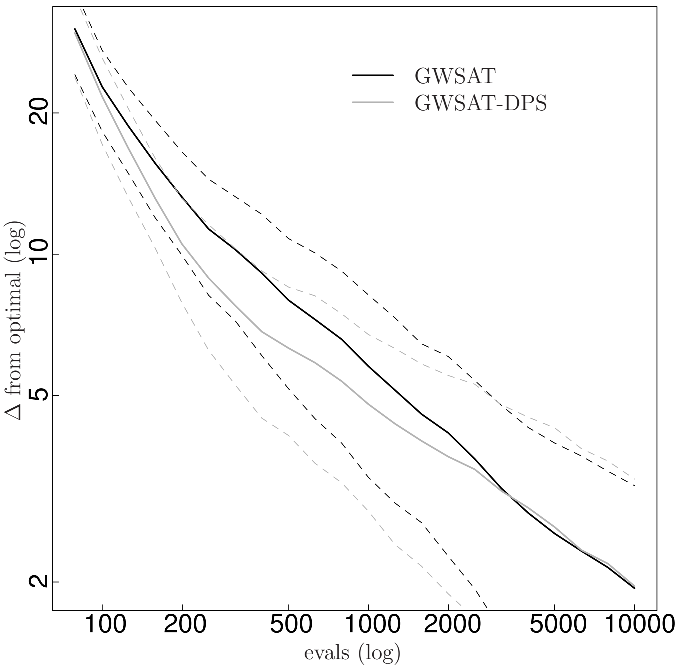


Figure 6.10: Mean convergence (Δ from optimal vs. evaluation) plot for GWSAT and GWSAT-DPS over `uf250-1065` distribution. Dashed lines indicate standard deviation from mean convergence.

on low level plateaus begin to dominate.

To investigate whether this phenomenon translates deep into the overconstrained phase, we also report the percentage of runs (out of 1000) that reached certain levels near and at the optimal level for ten instances in the MAXSAT 2009 benchmark set `s3v80c1000` (in this case the clause to variable ratio is 12.5). The optimal level is found by the complete solver MiniMaxSat [HLO08]. We report the percentage of runs that reach each level from opt−5 (five levels below the optimal), to opt−0 (the optimal value) in Table 6.3. Again we see that the heavy-tail escape behavior obscures the advantage of DPS at the optimal level.

The results demonstrate that search space structure can be used to influence the trajectory of plateau search in such a way that certain plateaus may be escaped more quickly. DPS may also be used to quickly move to better regions of the ob-

		% of runs that reach level					
instance	algorithm	opt-5	opt-4	opt-3	opt-2	opt-1	opt-0
1	GWSAT	96.9	93.2	88.2	83.8	59.0	43.1
	GWSAT-DPS	**97.6**	**94.4**	**90.3**	**87.5**	**61.3**	**51.6**
2	GWSAT	98.3	95.8	**86.3**	71.3	**47.0**	**35.2**
	GWSAT-DPS	**99.2**	**96.8**	85.7	**72.8**	46.8	32.3
3	GWSAT	88.9	70.1	61.3	52.1	**29.5**	**29.4**
	GWSAT-DPS	**90.5**	**70.4**	**64.7**	**53.4**	29.2	28.8
4	GWSAT	92.4	89.1	82.0	78.1	**42.0**	**18.9**
	GWSAT-DPS	**95.0**	**91.4**	**85.2**	**80.8**	41.3	17.1
5	GWSAT	95.4	93.1	86.5	75.5	**42.5**	28.9
	GWSAT-DPS	**97.0**	**96.1**	**89.5**	**77.0**	40.4	**30.7**
6	GWSAT	88.7	73.0	72.7	67.6	58.4	43.6
	GWSAT-DPS	**92.5**	**73.8**	**73.3**	**70.2**	**60.5**	**49.5**
7	GWSAT	98.7	98.3	95.4	89.7	74.6	**73.8**
	GWSAT-DPS	**99.7**	**99.5**	**96.3**	**93.0**	**74.7**	64.0
8	GWSAT	96.3	89.4	68.5	63.8	39.3	**18.7**
	GWSAT-DPS	**98.2**	**92.8**	**75.1**	**71.1**	**47.1**	18.2
9	GWSAT	97.1	95.6	95.0	90.0	86.4	40.0
	GWSAT-DPS	**97.8**	**97.4**	**97.1**	**94.0**	**93.4**	**53.5**
10	GWSAT	92.8	88.8	81.1	80.0	62.1	25.7
	GWSAT-DPS	**95.8**	**94.3**	**88.9**	**87.4**	**69.5**	**32.7**

Table 6.3: Percentage of runs (out of 1000) that reached best six levels in each of the 10 instances from the MAXSAT 2009 `s3v80c1000` benchmark set. Higher percentages are in boldface.

jective function. This could be useful for faster approximations on large problems, or hybridized in the manner of Kroc et al. [KSGS09] to switch to a complete solver when the surrogate gradient is no longer helpful.

We have thus seen that knowledge of simple objective function statistics over local regions can inform the decision process of hill-climbing local search algorithms to improve their behavior in certain regions of the search space. We will now see how this information can be used to solve for promising mutation rates in a specific evolutionary algorithm.

6.2 Controlling Mutation Rates in the (1+1)-EA

Evolutionary algorithms (EAs) are probabilistic direct search methods that are often applied to the task of function optimization. Broadly speaking, this class of algorithms employs operations inspired by *organic evolution* such as *recombination*, *mutation*, and *selection* to navigate the search space in response to a *fitness function*. When applied to a combinatorial optimization problem $(\mathcal{X}, f)$, the *fitness function* of the EA is simply the objective function f of the combinatorial problem; we will use the former term for the remainder of the section.

We will focus on the simple case of the (1+1)-EA which employs mutation but not recombination. Mutation can be seen as a generalized local operator that computes a new state (called an *offspring*) from the current state (called the *parent*) that lies a certain distance away on the neighborhood graph according to some probability.

Evolutionary algorithms operating on $\{0,1\}^n$ often employ a natural mutation operation in which the state of each bit of the state under consideration is inverted with some probability ρ, the so-called *mutation rate*. We show, using the framework developed in Chapter 4, that when the fitness function is a k-bounded pseudo-

Boolean function, it is always possible to efficiently compute the *expected fitness* of a mutation from each string for a given rate. We then show that it is always possible to solve for the mutation rate that maximizes the expected fitness of the offspring for any point.

It is well-understood that the choice of the mutation rate parameter can have a strong impact on the performance of EAs, and a large number of experimental and theoretical investigations have been carried out to determine the optimal mutation rate. For example, many experimental studies have suggested a mutation rate between 0.001 and 0.01 [De 75, Gre86, SCED89]. In many cases, however, mutation rates that cause an EA to perform well on one class of functions may produce poor performance on another class of functions. Indeed, Droste et al. [DJW98] have given theoretical evidence that a mutation rate of $1/n$ guarantees convergence in $O(n \log n)$ time for the (1+1)-EA applied to linear functions. On the other hand, Jansen and Wegener [JW00] have introduced a function for which a mutation rate of $1/n$ leads to superpolynomial runtime of the (1+1)-EA with high probability while a mutation rate of $\frac{\log n}{n}$ leads to expected polynomial-time convergence on the same function. Such results stress the importance of an understanding of the relationship between the mutation rate and the function being searched.

On linear functions, this relationship is well-understood. For instance, in the case of One-Max, it is straightforward to derive an analytical expression for the probability of a successful mutation [Bäc92a]. In the case of general pseudo-Boolean functions, the probability of a successful mutation from any arbitrary point is difficult to know. Furthermore, analytical expressions specifying optimal mutation rates have not previously been derived for epistatically bounded pseudo-Boolean functions.

6.2.1 The (1+1)-EA

We concentrate on the (1+1)-EA applied to the task of maximizing pseudo-Boolean functions. The (1+1)-EA has been subject to a number of theoretical studies [DJW98, GKS99, Müh92, Rud97, JW00]. The algorithm is presented below, parameterized by mutation rate ρ.

```
(1+1)-EA($\rho$)

Choose $x \in \{0,1\}^n$ uniformly at random
while stopping criteria not met
    do
        $y \leftarrow x$
        Flip each bit of $y$ independently with prob. $\rho$
        if $f(y) \geq f(x)$
           then $x \leftarrow y$
```

The mutation rate parameter ρ controls the degree to which each search point is perturbed to produce the next search point. Often, a constant mutation rate of $\rho = 1/n$ is recommended [Bäc93, Müh92], especially for linear functions. On functions with nonlinearity, there is strong evidence that the optimal mutation rate is time-dependent [Hol75, Bäc92a, BS96, HM91].

For some functions, it is possible to compute the exact probability of a successful mutation as a function of *fitness level* and mutation rate [Bäc93, JW00]. This is especially useful in the case of runtime analysis because it allows one to bound the expected number of mutations until a successful offspring is produced. However, in the case of general pseudo-Boolean functions, this probability is difficult to compute. When this probability is not known, one solution is to use *self-adaptation* [Bäc92b] in which each individual is augmented with an encoding of its own mutation rate and the rate is adapted along with the function parameters.

While linear, unimodal functions have a provably optimal mutation rate of $1/n$, Bäck pointed out that when the fitness is multimodal, a search for a dynamically varying mutation rate different from a constant value during search may be worthwhile to overcome local optima [Bäc93]. Hesser and Männer [HM91] presented a theoretical argument that suggested the mutation probability in a population-based GA employing crossover should decrease with time. In this section we also find evidence for this on k-bounded functions. In fact, we find that each state has its own "expectation-best" mutation rate that maximizes the expected fitness of its offspring. This rate changes in response to the relationship between the fitness of the state itself and the expected fitness of states that lie within Hamming distance k.

We now show that on a k-bounded pseudo-Boolean function, even if we cannot recover the optimal mutation rate (in terms of success probability) for a state, we can at least efficiently compute the mutation rate that maximizes the expected fitness of the offspring.

6.2.2 The expected fitness of mutations

We assume that the fitness function $f : \{0,1\}^n \to [0,\infty)$ is k-bounded and has a non-negative real codomain. In the context of function optimization and search, adding an arbitrary constant to satisfy this constraint will not affect the behavior of the algorithm.

Recall from Definition 4.1 on page 69 that the Hamming sphere $S^{(r)}(x)$ of radius r around a point x is the set of points that lie at Hamming distance exactly r from x. Let $x \in \{0,1\}^n$ be the string under consideration. Mutation is a stochastic process that produces an offspring z by changing components of x. Since the process is stochastic, we can characterize $f(z)$ as a random variable. We can calculate the expected value of this random variable as a function of $f(x)$: the fitness of the current state. In other words, we are interested in calculating the first moment of f over a ball of radius n around x, but now the sampling is no longer uniform throughout the region as it was in Equation (4.1). Indeed, the probability mass function of the random variable corresponding to $f(z)$ now depends on Hamming distance from x, which is captured by sphere membership. Applying mutation to an element $x \in \{0,1\}^n$ is analogous to performing n independent Bernoulli trials to determine whether or not to change each bit of x. Thus, the probability that the offspring of x under mutation with rate ρ lies at Hamming distance r from x is distributed binomially.

To produce an offspring z via mutation, each bit of x is flipped with probability ρ. Thus z lies in a sphere of radius r around x with probability $\rho^r(1-\rho)^{n-r}$. The total fitness of all states that lie in the sphere $S^{(r)}(x)$ is equal to

$$\sum_{y \in S^{(r)}(x)} f(y) = \boldsymbol{S}^{(r)} f(x) \qquad \text{by (4.9)},$$

and as we derived in the proof of Theorem 4.1,

$$\begin{aligned} &= \sum_{i: w_i \neq 0} \gamma_i^{(r)} w_i \psi_i(x) \\ &= \sum_{i: w_i \neq 0} \mathcal{K}_r(\langle i, i \rangle, n) w_i \psi_i(x), \end{aligned}$$

where $\mathcal{K}_r(p, n)$ is the Krawtchouk polynomial defined in (4.10) and $\langle i, i \rangle$ is the string inner product of i with itself, i.e., the order of the length-n bitstring representation of i. The contribution to the expectation in a sphere at radius x can be obtained by multiplying this sum by the probability of the offspring lying in the sphere.

$$\rho^r(1-\rho)^{n-r} \sum_{y \in S^{(r)}(x)} f(y) = \rho^r(1-\rho)^{n-r} \sum_{i: w_i \neq 0} \mathcal{K}_r(\langle i, i \rangle, n) w_i \psi_i(x).$$

We denote as $\mathbb{M}_x(\rho)$ the expected fitness of the offspring of x under mutation. Since all spheres around x are disjoint, the expected fitness of the offspring of x

under mutation can be computed as the sum of the expectation contributions from each sphere:

$$\mathbb{M}_x(\rho) = \sum_{r=0}^{n} \rho^r (1-\rho)^{n-r} \sum_{i:w_i \neq 0} \mathcal{K}_r(\langle i, i \rangle, n) w_i \psi_i(x). \tag{6.3}$$

Again, since f is k-bounded, the above series contains a polynomial number of terms. Letting

$$a_r = \sum_{i:w_i \neq 0} \mathcal{K}_r(\langle i, i \rangle, n) w_i \psi_i(x), \tag{6.4}$$

we can immediately compute the expected fitness of the offspring as

$$\mathbb{M}_x(\rho) = \sum_{r=0}^{n} a_r \rho^r (1-\rho)^{n-r} \qquad \text{by (6.3).} \tag{6.5}$$

Intuitively, a_r is the total fitness over all states that lie in sphere $S^{(r)}(x)$. It will be convenient later to express a_r in terms of the first moment of f over $S^{(r)}(x)$ which follows directly from the proof of Theorem 4.2

$$a_r = \binom{n}{r} \mu_1(S^{(r)}). \tag{6.6}$$

We can re-express Equation (6.5) as a degree-n polynomial in ρ as

$$\mathbb{M}_x(\rho) = A_0 + A_1\rho + A_2\rho^2 + \cdots + A_n\rho^n, \tag{6.7}$$

where

$$A_m = \sum_{\ell=0}^{m} a_{m-\ell} \binom{n-m+\ell}{\ell} (-1)^\ell. \tag{6.8}$$

When f is epistatically bounded by a constant, the Walsh coefficients can be found in polynomial time, and the coefficients A_m can be efficiently computed. Later, we will also see that it is possible to further bound the degree of this polynomial.

To find the mutation rate ρ which maximizes the expected fitness of the offspring of x, we simply need to find

$$\arg\max_{0 \leq \rho \leq 1} \mathbb{M}_x(\rho) = \arg\max_{0 \leq \rho \leq 1} A_0 + A_1\rho + A_2\rho^2 + \cdots + A_n\rho^n.$$

The first and second derivatives of the expected fitness are

$$\frac{d}{d\rho}\mathbb{M}_x(\rho) = A_1 + 2A_2\rho + 3A_3\rho^2 + \cdots + nA_n\rho^{n-1},$$

and

$$\frac{d^2}{d\rho^2}\mathbb{M}_x(\rho) = 2A_2 + 6A_3\rho + 12A_4\rho^2 + \cdots + n(n-1)A_n\rho^{n-2}.$$

It is easy to find the stationary points of $\mathbb{M}_x(\rho)$ by numerically solving for the real roots of $\frac{d}{d\rho}\mathbb{M}_x(\rho)$. Of course, we can use the so-called "second derivative test" to test for concavity and solve for the local maxima point set

$$M = \left\{\rho : \frac{d}{d\rho}\mathbb{M}_x(\rho) = 0 \text{ and } \frac{d^2}{d\rho^2}\mathbb{M}_x(\rho) < 0\right\}.$$

The mutation rate that maximizes the expected fitness of the offspring is easily retrieved by finding the point $\rho^\star \in (M \cap [0,1]) \cup \{0,1\}$ such that $\mathbb{M}_x(\rho^\star)$ is maximal.

6.2.3 Degeneracy: when no mutation is "best"

The polynomial defined in (6.7) always has a (possibly non-unique) maximum in the interval $[0,1]$. The degenerate case is when $\mathbb{M}_x(\rho)$ is monotonically decreasing and no stationary points lie within the interval. Moreover, it is possible that any maxima lying within the interval have evaluation strictly less than $\mathbb{M}_x(0)$. In this case, the "optimal" value is $\rho^\star = 0$. Since $\mathbb{M}_x(0) = f(x)$, this means that there is no possible mutation rate (constant across bitstrings) that will produce an offspring whose expectation is greater than $f(x)$: the fitness of the current point. This corresponds to a local optimum in "expectation space", that is, any mutation is disimproving in expectation. In Sections 6.2.4 and 6.2.5, we will find conditions on the fitness of f in which this degeneracy must hold for linear functions and $k > 1$-bounded functions, respectively.

6.2.3.1 Choosing a suitable nonzero mutation rate

When any mutation rate is expected to produce an offspring with lower fitness, the optimal choice to maximize expected fitness is to perform no mutation. Instead, we would like to perform mutations that, in some sense, minimize the expected loss in fitness.

Suppose $\rho^\star = 0$. Let $0 < \rho \ll 1$ be any positive value close to zero. Then we know

$$\mathbb{M}_x(\rho) = (1-\rho)^n f(x) + \sum_{r=1}^{n} a_r \rho^r (1-\rho)^{n-r} < \mathbb{M}_x(0) = f(x).$$

Ignoring the higher order terms we can write

$$(1-\rho)^n f(x) \leq \mathbb{M}_x(\rho) < f(x).$$

Choosing a mutation rate $\rho = k/n$ means that in expectation, k/n bits will be changed. We can recover the "standard" recommended mutation rate of $\rho = 1/n$ by observing from the above inequality,

$$\left(1 - \frac{k}{n}\right)^n f(x) \leq \mathbb{M}_x\left(\frac{k}{n}\right) < f(x).$$

Asymptotically we have

$$e^{-k} f(x) \leq \mathbb{M}_x\left(\frac{k}{n}\right) < f(x).$$

The lower bound on $\mathbb{M}_x\left(\frac{k}{n}\right)$ is maximized when $k = 1$. Thus, when the offspring is expected to be disimproving, the mutation rate $1/n$ maximizes the lower bound on the expectation of the fitness of the offspring under the constraint that we flip at least 1 bit in expectation. Moreover, in this case we know the expected fitness of the offspring of x is asymptotically bounded below by $e^{-1}f(x)$.

The slope of the $\mathbb{M}_x(\rho)$ polynomial at zero tells us how quickly the expected fitness falls off by choosing close-to-zero mutation rates. Interestingly, this slope is exactly n times the difference between the fitness of the current point and the average fitness over the immediate neighbors at Hamming distance 1. This can be derived easily by observing that

$$\frac{d}{d\rho}\mathbb{M}_x(0) = A_1,$$

so by (6.8), the slope of the $\mathbb{M}_x(\rho)$ polynomial at zero is equal to $a_1 - na_0$. From (6.4) it is easy to see that

$$\begin{aligned} a_1 &= \sum_{i:w_i \neq 0} \mathcal{K}_1(\langle i, i\rangle, n) w_i \psi_i \\ &= n\mu_1(S^{(1)}(x)) = n\mu_1(N(x)) && \text{by (6.6)}, \end{aligned}$$

where $N(x)$ is the Hamming neighborhood of x, and,

$$a_0 = \sum_{i:w_i \neq 0} \mathcal{K}_0(\langle i, i\rangle, n) w_i \psi_i = f(x),$$

so the rate of change at zero is equal to $n\left(\mu_1(N(x)) - f(x)\right)$. Of course this makes intuitive sense: the change in expectation of very small mutations is completely determined by the difference between the current point and its immediate neighbors.

6.2.4 Linear functions

Many analyses of the (1+1)-EA (e.g., [Müh92, Bäc93, DJW98]) have focused on the linear (or *separable*) case, that is, $k = 1$-bounded pseudo-Boolean functions. In the case of a linear function, the Walsh basis expansion has only nonzero terms in the zeroth and first order. Therefore,

$$\begin{aligned}
a_r &= \mathcal{K}_r(0,n)w_0 + \sum_{i:\langle i,i\rangle=1} \mathcal{K}_r(1,n)w_i\psi_i(x) && \text{by (6.4)},\\
&= \binom{n}{r}w_0 + \sum_{i:\langle i,i\rangle=1}\sum_{j=0}^{r}\binom{1}{j}\binom{n-1}{r-j}w_i\psi_i(x)\\
&= \binom{n}{r}\left(w_0 + \frac{n-2r}{n}\left(f(x)-w_0\right)\right),
\end{aligned}$$

and since $w_0 = 2^{-n}\sum_{x\in\{0,1\}} f(x) = \bar{f}$ is the average fitness over all bitstrings (see e.g., [Hec99]),

$$= \binom{n}{r}\left(\bar{f} + \frac{n-2r}{n}\left(f(x)-\bar{f}\right)\right).$$

Substituting a_r in (6.8) we get

$$A_m = \sum_{\ell=0}^{m}\left(\bar{f} + \frac{n-2(m-\ell)}{n}\left(f(x)-\bar{f}\right)\right)\times\binom{n}{m-\ell}\binom{n-(m-\ell)}{\ell}(-1)^k,$$

which simplifies significantly to

$$A_m = \begin{cases} f(x) & \text{if } m=0,\\ 2\left(\bar{f}-f(x)\right) & \text{if } m=1,\\ 0 & \text{otherwise.}\end{cases}$$

Thus if f is a linear (1-bounded) pseudo-Boolean function, the expected fitness of an offspring using mutation rate ρ is simply

$$\mathbb{M}_x(\rho) = f(x) + 2\left(\bar{f}-f(x)\right)\rho.$$

So in the case of linear functions, the polynomial terms of order greater than one vanish and the $\mathbb{M}_x(\rho)$ polynomial is always a line with y-intercept $f(x)$ and slope equal to twice the difference between the mean fitness and the fitness of x. Therefore we have recovered the well-known result for linear functions that when $f(x) < \bar{f}$, $\mathbb{M}_x(1)$ is maximal (since the slope is positive) and when $f(x) > \bar{f}$, $\mathbb{M}_x(0)$ is maximal (negative slope). On such functions, large mutations are quickly able to reach the mean value, after which the smallest mutation probability *that still flips at least one bit in expectation*, namely $1/n$ maximizes the expected fitness of the offspring. This agrees somewhat with the result of Droste et al. [DJW98] that on linear functions the (1+1)-EA converges in $O(n \log n)$ steps with this mutation rate, and a constant mutation rate of much larger or much smaller results in provably longer convergence times.

Here we also see a weakness in relying solely on the expected fitness of the offspring to choose a mutation rate. In the case of linear functions, when $f(x) = \bar{f}$, the mutation rate that maximizes the *probability of success* is equal to $\frac{n}{2}$ (see e.g., [Bäc92a]). However, using the $\mathbb{M}_x(\rho)$ polynomial, when $f(x) = \bar{f}$, the A_1 term vanishes and $\mathbb{M}_x(\rho)$ is a constant function: all mutation rates give the same expectation of $f(x)$.

We can thus conclude that success probability, when available, presents better higher resolution information about the *optimal* mutation rate: i.e., that which maximizes the probability of a successful offspring. However, on general k-bounded pseudo-Boolean functions where that probability is unknown or difficult to compute, the expectation of fitness offers a compromise.

6.2.5 Functions of bounded epistasis

Linear functions, while amenable to analysis, are a restricted class of fitness functions. At the other extreme, the entire set of general pseudo-Boolean functions is rather expansive. The class of pseudo-Boolean functions whose epistasis is bounded by some constant k contains fitness functions that can be very difficult for evolutionary algorithms, and includes a collection of NP-hard optimization problems such as maximum k-satisfiability and the unrestricted model of NK-landscapes.

In the last section we saw that the $\mathbb{M}_x(\rho)$ polynomial coefficients A_m vanish for $m > 1$. As expected, one can generalize the result to k-bounded pseudo-Boolean functions.

Proposition 6.1. *Let f be an arbitrary k-bounded pseudo-Boolean function. Consider the $\mathbb{M}_x(\rho)$ polynomial for any $x \in \{0,1\}^n$. If $m > k$ then $A_m = 0$.*

Proof sketch. The proof is based on showing that the sum of the first $k+1$ terms $\ell = 0, \ldots, k$ of A_m in (6.8) is exactly equal to the additive inverse of the sum of the last $m-k$ terms $\ell = (k+1), \ldots, m$. This is mostly tedious manipulation and is thus omitted here. □

It follows from Proposition 6.1 that in general, when f is epistatically bounded by a constant k, the expected fitness of the offspring is a degree k polynomial in the mutation rate. In order to compute the rate $\rho^\star$ that yields the maximal expected fitness, it is enough to solve for the real roots of the degree $k-1$ polynomial $\frac{d}{d\rho}\mathbb{M}_x(\rho)$ as described above.

Proposition 6.1 also asserts that only the fitness of the points that lie within Hamming distance k of any individual completely determine the expected fitness of the offspring since any A_m contains terms involving a_r (and hence, by (6.6), $\mu_1(S^{(r)}(x))$) only for $r < m$. Thus, as in the linear case, it is enough to compute the mean fitness in Hamming spheres out to radius k. It follows that if the mean fitness in spheres of radius one to k are strictly less than the fitness at x, $\mathbb{M}_x(\rho)$ is degenerate and no mutation rate will produce offspring with expected fitness above the fitness at x. We formalize this argument in the following lemma.

Lemma 6.1. *If $\mu_1(S^{(r)}(x)) < \mu_1(S^{(0)}(x))$ for all $0 < r \leq k$, then $\mathbb{M}_x(0)$ is maximal.*

Proof. Choose $0 < \rho \leq 1$. Then

$$\begin{aligned}
\mathbb{M}_x(\rho) &= (1-\rho)^n \mu_1(S^{(0)}(x)) + \sum_{r=1}^{n} \binom{n}{r} \rho^r (1-\rho)^{n-r} \mu_1(S^{(r)}(x)) \qquad \text{by (6.6)}, \\
&< (1-\rho)^n \mu_1(S^{(0)}(x)) + \sum_{r=1}^{n} \binom{n}{r} \rho^r (1-\rho)^{n-r} \mu_1(S^{(0)}(x)) \\
&= \mu_1(S^{(0)}(x)) \left(\sum_{r=1}^{n} \binom{n}{r} \rho^r (1-\rho)^{n-r} \right) \\
&< \mathbb{M}_x(0).
\end{aligned}$$

Since the choice of ρ was arbitrary, and, by Proposition 6.1 all coefficients above k vanish, it follows that $\mathbb{M}_x(0)$ is maximal. □

Clearly, when $k > 1$, k-bounded functions do not have simple linear $\mathbb{M}_x(\rho)$ polynomials as we saw in the previous section. To illustrate this we plot the $\mathbb{M}_x(\rho)$ polynomials for several random points drawn from various k-bounded functions in Figure 6.11.

Heckendorn et al. [HRW98], among others, proposed using NK-landscapes and k-satisfiability problems as test problem domains for evolutionary algorithms. Since both are representative k-bounded pseudo-Boolean functions, we now report the results of a number of numerical simulations that study the $\rho^\star$ mutation rate and compare it to proposed static rates found in the literature.

6.2.5.1 Unrestricted NK-landscapes

The NK-landscape model [KL87] is a stochastic method for constructing fitness functions over binary sequences of length n. The model was developed to study how epistasis affects the ruggedness of the fitness landscape. The fitness function for the NK model is defined as

$$f(x) = \frac{1}{n} \sum_{j=1}^{n} g_j \left(x[j], x[b_1^{(j)}], x[b_2^{(j)}], \ldots, x[b_K^{(j)}] \right),$$

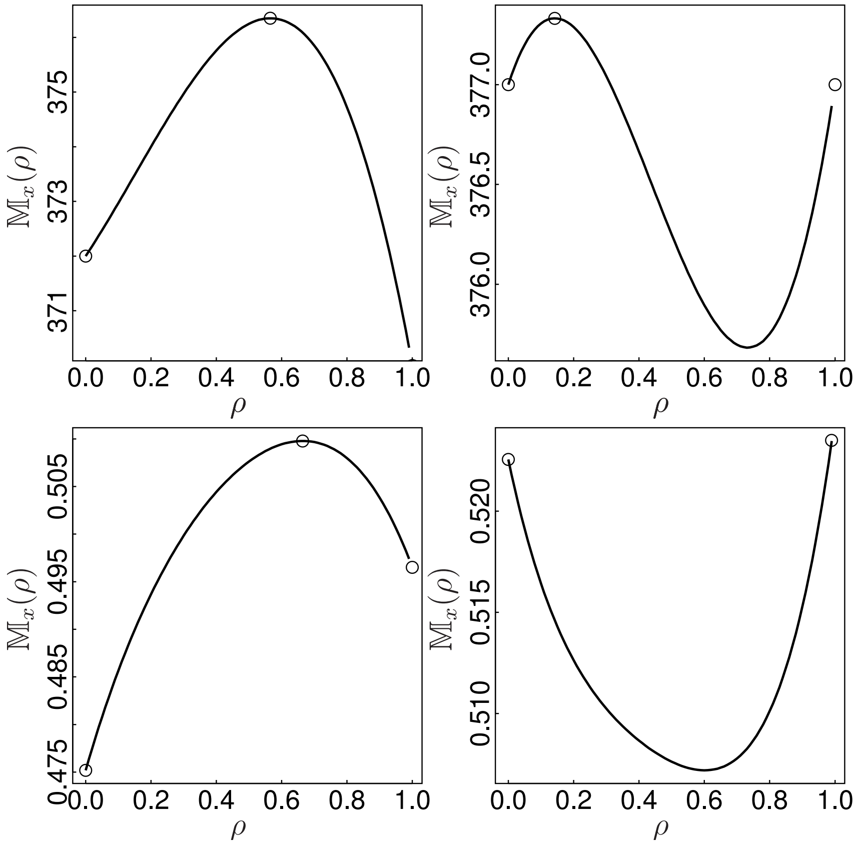
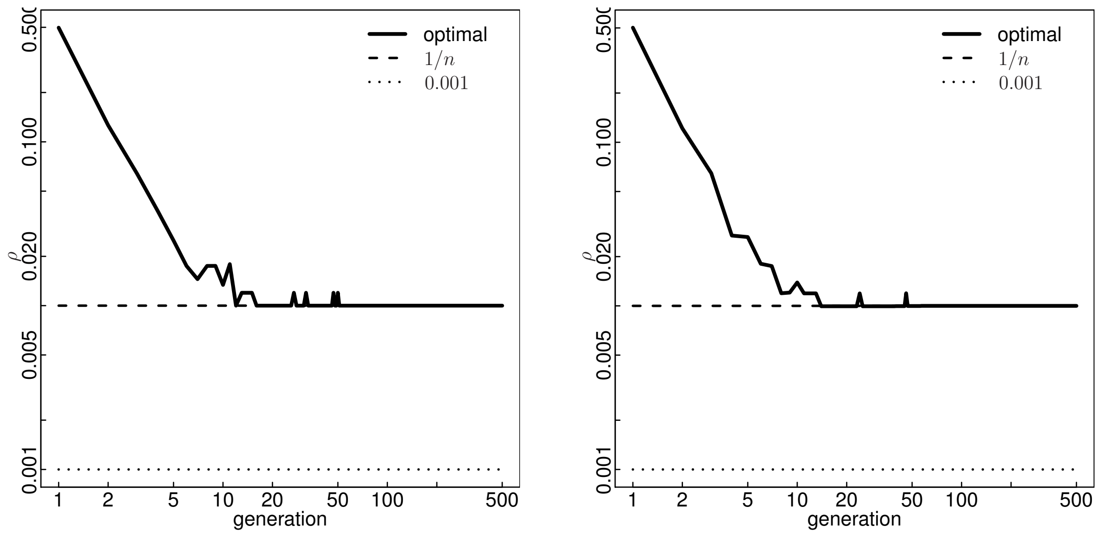


Figure 6.11: $\mathbb{M}_x(\rho)$ polynomials for random points in the Max-3-Sat search space ($n = 100$) [top left and right] and NK-landscapes ($N = 100, K = 3$) [bottom left and right].

where $g_j : \{0,1\}^{K+1} \rightarrow [0,1]$ gives the fitness contribution of the j^{th} bit in x, and K other bits $\{b_i^{(j)}\}$. Typically, the codomain values for g_j are generated uniformly at random and fixed during search.

There are two variants of the NK model. In the *adjacent* model, the set of K bits $\{b_i^{(j)}\}$ that interact epistatically with bit j are adjacent to bit j on the bit string. In the *unrestricted* model (sometimes called the *random* model), the epistatic bit pattern $\{b_i^{(j)}\}$ for the j^{th} bit is drawn randomly (and fixed) from the $n-1$ remaining bits. Thus for each bit j, there are $\binom{n-1}{K}$ possible selections for the set $\{b_i^{(j)}\}$. Since the fitness function is expressed as the sum of n functions each of which depends only on a single bit and the K bits in its epistatic pattern, the function is epistatically bounded by $K+1$. Wright et al. [WTZ00] proved that the problem of finding the global optimum for the adjacent model is in P by giving a P-time dynamic programming solution. Moreover, they proved that the unrestricted model is NP-hard. In this paper, we concentrate on the unrestricted NK model.

To illustrate the behavior of the optimal rate $\rho^\star$, we performed 500 trials of 500 generations each of the (1+1)-EA employing three different mutation rates: 1) the commonly recommended $1/n$, 2) a "hard-wired" rate of 0.001, and 3) the expectation-optimal rate given by the maximum of $\mathbb{M}_x(\rho)$ at each point. The NK-landscape function parameters were $n = 100$ and $K = 3$. In each case, the extra time necessary to solve for the real roots of $\frac{d}{d\rho}\mathbb{M}_x(\rho)$ was negligible since the degree of the polynomial was so low ($k = 4$). When the expectation maximal rate $\rho^\star$ reaches degeneracy (i.e., the maximum is at $\mathbb{M}_x(0)$), we revert to the mutation rate of $1/n$ that we proved in Section 6.2.3 maximizes the expected fitness of the offspring while imposing the constraint that at least some bits are flipped in expectation.

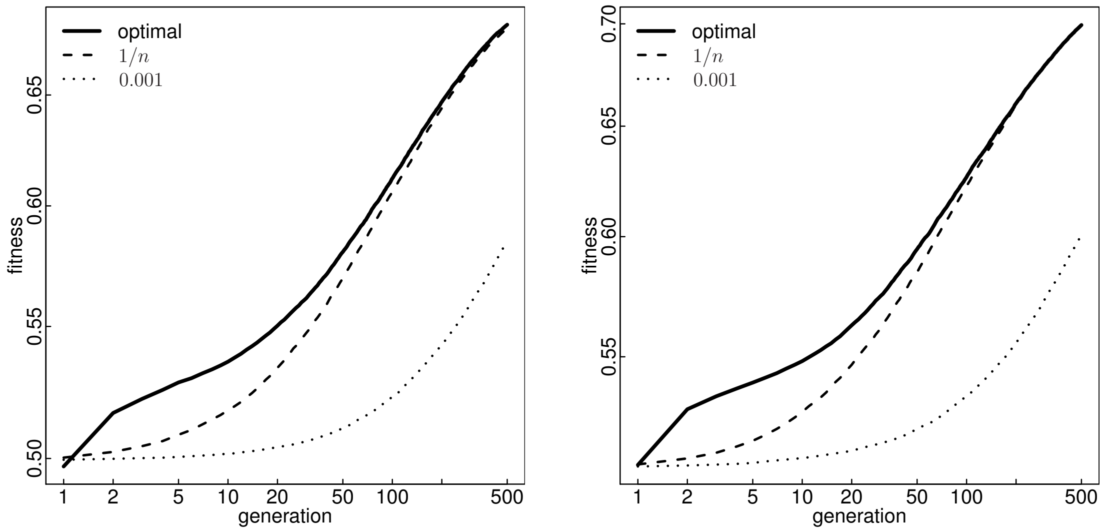


(a) $n = 100$, $K = 2$ (b) $n = 100$, $K = 3$

Figure 6.12: Log-log plot of mean mutation rates for (1+1)-EA on 500 trials of 500 generations each on two unrestricted NK-landscape models.

In Figure 6.12 we plot the average mutation rate ρ as a function of generation. As the fitness of the points remain below the average fitness within Hamming radius k, there is a significant increase from the recommended rate of $1/n$. However, very quickly the fitness of the point exceeds the expectation of the fitness within radius k and $\mathbb{M}_x(\rho)$ reaches degeneracy and the rate reverts to $1/n$. This result somewhat corroborates the claims of others [HM91, Bäc93] that the best mutation rate tends to decrease during search.

On the NK-landscape with bounded epistasis, the optimal rate $\rho^\star$ reverts quickly to the recommended rate of $1/n$. However, it leads to significant gains very early in search as we see in Figure 6.13 where we plot the mean convergence of the (1+1)-EA for a representative NK-landscape using the three mutation rate schemes. Before the $\rho^\star$ rate becomes degenerate, significant gains are made over the static $1/n$ rate.



(a) $n = 100$, $K = 2$

(b) $n = 100$, $K = 3$

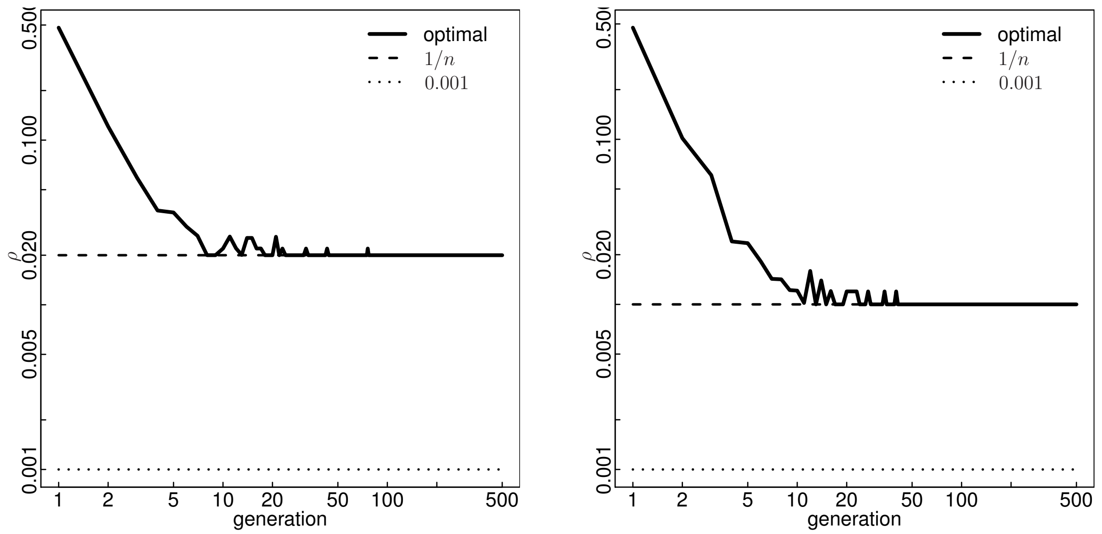
Figure 6.13: Log-log plot of mean fitness of (1+1)-EA on 500 trials of 500 generations each on two unrestricted NK-landscape models.

6.2.5.2 Maximum k-satisfiability

In Chapter 3 we saw that the objective function for Max-k-Sat can be written as a k-bounded pseudo-Boolean function. Hence, a (1+1)-EA tasked to optimize the Max-k-Sat problem can be easily adapted to find the expectation-optimal mutation rate for any string.

We performed 500 trials of 500 generations each of the (1+1)-EA on two randomly generated Boolean formulas with 50 variables and 218 clauses, and 100 variables and 430 clauses. The algorithm employed the three different mutation rates discussed above. Again, when the optimal rate reaches degeneracy, we revert to the rate that maximizes expected fitness while enforcing bits to be flipped in expectation.

In Figure 6.14 we report the ρ values found during search. Again, due to the simplicity of the $\mathbb{M}_x(\rho)$ polynomial, the solution time is negligible to compute $\rho^\star$ in each case. In both instances, we see again the decrease of the expectation-

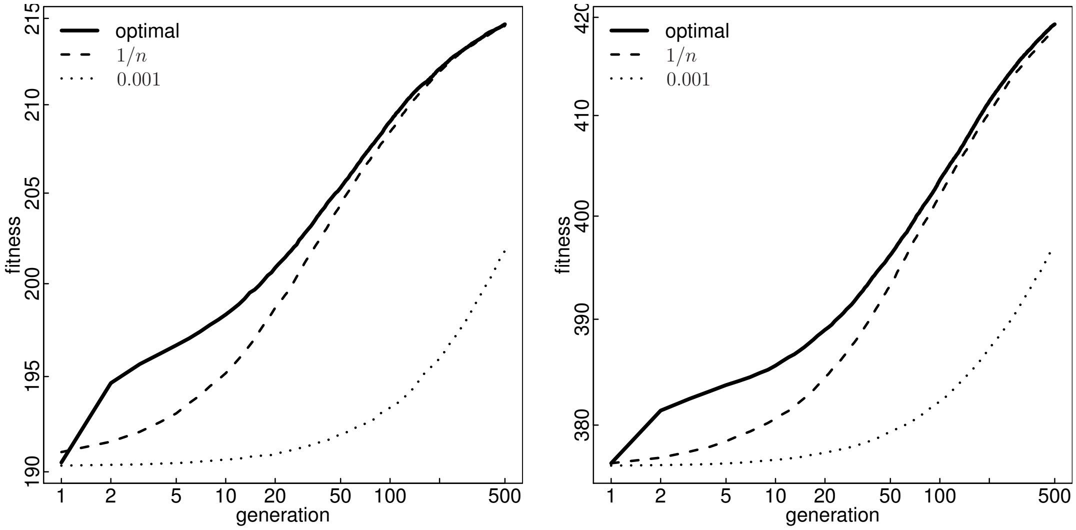


(a) $n = 50$ (variables), $m = 218$ (clauses)

(b) $n = 100$ (variables), $m = 430$ (clauses)

Figure 6.14: Log-log plot of mean mutation rates for (1+1)-EA on 500 trials of 500 generations each on two Max-3-Sat problems.

optimal rate quickly to the degenerate rate where it then reverts to the standard $1/n$ mutation rate around generation 20 to 50. The initially higher expectation-optimal rate shown in Figure 6.14 translates to early gains in search as reported in Figure 6.15 when compared to standard and hard-wired mutation rates.



(a) $n = 50$ (variables), $m = 218$ (clauses) (b) $n = 100$ (variables), $m = 430$ (clauses)

Figure 6.15: Log-log plot of mean fitness of (1+1)-EA on 500 trials of 500 generations each on two Max-3-Sat problems.

Chapter 7

Summary and Future Work

In this thesis we have studied characteristics of combinatorial search spaces in the context of search algorithms that perform perturbative local search by employing some move, mutation, or neighborhood operator. A strong understanding of search space characteristics is important to understanding the behavior of search algorithms. A paucity of fundamental models and theories in the study of incomplete combinatorial search algorithms has resulted in endemic speculation and vague, inchoate ideas about their behavior and performance: a problem that has been articulated by others, e.g., [Hoo96, Wat03]. In many cases, rigorous analyses of search algorithms tend to be outstripped by a somewhat pathological ad-hoc "incremental design" paradigm. In this paradigm, algorithms and heuristics are continually designed to be successful on a finite suite of benchmarks. Accordingly, a large amount of research effort is expended on tweaking and tuning to produce algorithms that are competitive on such benchmark suites without a clear and scientific understanding of the processes that ultimately make combinatorial search successful.

For our analysis we have employed a basis function expansion of the objective function in terms of the neighborhood graph induced by the search operator. We have shown that this formalism provides insight into certain structural relationships. For example, in Chapter 2, we connected the formalism to results from the theory of inapproximability to show that the lower bound on the quality of local maxima introduced by Grover's maximum principle is sharp for Max-Ek-Lin-2. In Chapter 3 we applied the framework to obtain previously unknown bounds on the objective function levels for local maxima and plateau width in the 3-Sat search space.

The results presented in Chapter 4 provide a general approach to computing moments of a k-bounded pseudo-Boolean function f over arbitrary radius regions (Hamming spheres and balls) in polynomial time. This is significant for the following reasons.

1. The calculation is exact, i.e., the moments are not approximated.

2. The calculation is computationally efficient with respect to naïve enumeration since, in general, the size of these regions is exponential in the bitstring length of the domain of f (for instance, spheres of radius $n/2$ or Hamming balls of radius $O(n)$).

Exact calculation of the moments affords opportunities not previously available for heuristic search algorithms that rely on directed sampling. The moments $\{\mu_0(X), \mu_1(X), \mu_2(X), \ldots\}$ characterize the distribution of values in the codomain of f over particular regions X of the landscape. In the context of local and genetic search, an algorithm can exploit this information by computing statistical information about unexplored regions of the landscape to determine how promising such a region might be for further exploration.

Current constructive heuristic search techniques for constraint satisfaction problems such as Max-k-Sat are often able to prune large regions of their decision tree by fixing certain variables during search [Dec03]. These fixed variables transform an objective function to a new function defined over a subset of the original domain. Low moments over these subsets can also be retrieved in polynomial time using the algorithm of Chapter 4. Thus this approach is also useful to constructive algorithms since it can efficiently retrieve moments of a function over leaves in a subtree of the complete search tree.

In Chapter 5 we explored the *moment problem* of classical probability theory to connect the moments over local regions to the actual distribution of codomain values over local regions. We used a linear programming approach to obtain sharp bounds on the true distribution over Hamming regions of k-bounded pseudo-Boolean functions given constant-order moments constructed by the algorithm in Chapter 4. Furthermore, we relaxed the constraints on this approach to develop an approximation of the true distribution of objective function values over Hamming regions. We demonstrated how integrating the distribution function with respect to codomain value supplies a cumulative distribution function over the region that we used to estimate the number of improving moves in the region.

A fundamental goal of this research was to show how a formal analysis of the search space can be used to guide local search algorithms. To address this, in Chapter 6 we developed a surrogate plateau "gradient" function based on a Walsh transform of the Max-k-Sat objective function. This surrogate gradient gives the average objective function value over localized volumes of the search space to provide information to direct search through plateaus more quickly. We have shown that this improves the convergence time of hill-climbing local search on Max-k-Sat problems, especially when targeting near-optimal levels. We have thus shown that it is beneficial to use exact information about the search space structure to escape plateaus, rather than by resorting to blind random walks. We also believe that this approach will provide a rigorous foundation for future algorithmic innovations.

Finally, we showed that the framework makes it possible to efficiently compute the expected fitness of a mutation in the (1+1)-EA for any given mutation rate. Moreover, we showed how to efficiently compute for any point the mutation rate that results in the highest possible expected fitness (supposing that the mutation rate must be constant across the string). We also proved that, for strings with fitness higher than the expectation in Hamming spheres up to radius k, the frequently recommended rate of $1/n$ yields the maximal expected fitness of offspring while imposing the constraint that some bits are flipped in expectation.

7.1 Future Work

In this work we have concentrated on bounded pseudo-Boolean functions in Hamming space. One reason for this is that the Fourier analysis is much easier on Abelian (commutative) groups such as $(\mathbb{Z}/2\mathbb{Z})^n$ which corresponds to the domain of pseudo-Boolean functions. The analyses presented in this thesis will (almost trivially) generalize to bounded functions on higher cardinality alphabets where the underlying group remains Abelian, e.g., $(\mathbb{Z}/q\mathbb{Z})^n$ for $q \geq 2$. This would have immediate implications for other combinatorial problems such as graph coloring and hypergraph coloring.

Extensions to non-Abelian groups such as the symmetric group would have extensive impact on combinatorial problems such as TSP, scheduling, and ordering problems, but would be far less straightforward to develop. In some cases, most notably the TSP, linear ordering problem, and the quadratic assignment problem, Fourier decompositions of the objective function have been studied [Sta95, RKHS02], though the development of general adjacency operators such as those presented in Chapter 4 are not as easily specified. However, we believe many undiscovered connections exist between the work presented here and such problems.

The basis decomposition introduced in Chapter 2 and developed in Chapters 3 through 5 allows us to compute the autocorrelation coefficient of any bounded pseudo-Boolean function (such as the Max-k-Sat objective function) using its Fourier coefficients [SWH09]. A conjecture by Stadler and Schnabl [SS92] states that the autocorrelation is somehow directly related to the number of local optima in the search space. An interesting direction of research would be to connect the autocorrelation on bounded pseudo-Boolean functions to work by Reeves and Eremeev [RE04] who have developed an empirical estimate for the number of optima on combinatorial landscapes. This might allow one to place a confidence interval on the quantity predicted by the correlation length conjecture. For Max-k-Sat, this is directly related to work done by Reeves and Aupetit-Bélaidouni [RAB04] who have empirically estimated the number of optimal solutions for k-Sat problems.

In Chapter 5 we saw that the approximation of the true distribution over Hamming regions made it possible to efficiently estimate the number of improving moves in a region for any bounded pseudo-Boolean function. We conjecture that this approximation might be refined for some combinatorial problems by using *continuous* approximations of the probability mass functions that share the same low moments of the true distribution (e.g., retrieved by the algorithm presented in Chapter 4).

Finally, we mention that the two applications presented in Chapter 6 are only simple examples of algorithm design that is informed by search space analysis. Straightforward extensions of the directed plateau search heuristic described in Section 6.1 would be to incorporate higher moments into the surrogate gradient. This would lead to a slow-down in computation time but could translate to a higher resolution heuristic for escaping plateaus more quickly early in search. It would also be simple and informative to generalize the mutation rate control for evolutionary algorithms presented in Section 6.2 to other mutation-only evolutionary algorithms such as $(\mu + \lambda)$-EAs.

Another potential avenue for extension is to apply the current analysis to different search paradigms. The theoretical analysis in Chapters 4 and 5 partitions the search space into localized regions with respect to a neighborhood metric. As mentioned above, it is also possible to apply our results to partitions of the search space that correspond to decision pivots in complete search algorithms. In this way, we can extend the existing work to compute moments and approximated distributions over subtrees in decision tree space. This information could easily be used to guide branches in complete search algorithms (such as DPLL on k-Sat), or combined with the work on local moments in Chapter 4 to construct a directed hybrid approach.

7.2 Concluding remarks

This research is a first step toward connecting theoretical ideas to form a strong and useful foundation for the analysis of combinatorial search spaces. It provides a "sampling-free" characterization of distributions of function values that make sense in the context of a perturbative combinatorial search paradigm. It also allows for a deeper understanding of the relationship between functions and their distribution across the search space. Finally, this work illuminates the rich potential that lies at the interstices of a formal theory of search spaces and the principled design of heuristic search algorithms.

Over the course of this research we have also obtained a glimpse of why the perturbative combinatorial search paradigm is so successful on the class of NP-hard combinatorial problems instantiated by the k-bounded pseudo-Boolean functions. In the case of Max-3-Sat, we have ruled out certain local search pathologies for a large percentage of the search space. More generally, we have seen that for any k-bounded pseudo-Boolean objective function, the function value at a particular state is closely related to the moments that describe the true distribution of objective function values over states that are nearby in the space. However, many unresolved questions and open problems still remain to be addressed. We hope that this thesis constitutes an important advance in the theory of combinatorial search spaces.

Appendix A

Symbols and Concepts

Following is a list of symbols and concepts used in this thesis. In many cases, a page number is also included that provides a reference to the first use or definition of the corresponding symbol or concept in the text.

$\mathbb{R}$ = the set of real numbers.

$\mathbb{N} = \{0, 1, 2, \ldots\}$.

$\mathbb{Z} = \{\ldots, -2, -1, 0, 1, 2, \ldots\}$.

$[P]$ – square brackets enclosing an expression P should be understood as the *Iverson bracket* [Ive62, Knu92] which is used as an indicator function. In particular, we use square brackets to denote a number that is 1 if the enclosed condition is satisfied, and 0 otherwise:

$$[P] = \begin{cases} 1 & \text{if } P \text{ is true,} \\ 0 & \text{otherwise.} \end{cases}$$

where P is an expression that is either true or false. Note that $[\neg P] = 1 - [P]$, page 13.

δ_{ij}: unless otherwise specified, is the Kronecker delta function, page 35.

$\mathbb{E}[\cdot]$ denotes the expectation of a random variable, page 88.

A vector space is a set V over a field F together with two binary operations: vector addition which takes $v, w \in V$ to $v + w \in V$ and scalar multiplication which takes $a \in F, v \in V$ to $av \in V$ such that for all $u, v, w \in V$, $a, b \in F$:

- $u + (v + w) = (u + v) + w$;
- $v + w = w + v$;
- There exists an element $0 \in V$, called the zero vector, such that $v+0 = v$ for all $v \in V$;
- For all $v \in V$, there exists an element $w \in V$, called the additive inverse of v, such that $v + w = 0$. The additive inverse is denoted v;
- $a(v + w) = av + aw$;
- $(a + b)v = av + bv$;
- $a(bv) = (ab)v$;
- $1v = v$;

where 1 is the multiplicative identity of F.

An inner product space is a vector space V over a field F with an additional structure called an inner product which is a map $\langle \cdot, \cdot \rangle : V \times V \to F$, page 13.

A linear map from a vector space V to a vector space W is a function $\mathcal{F} : V \to W$ such that $f(x+y) = f(x)+f(y)$ and $f(ax) = af(x)$ for all $x, y \in V$ and $a \in F$.

A basis of an n-dimensional vector space V is a set of vectors $v_1, \ldots, v_n \in V$ such that every vector $v \in V$ can be expressed as a linear combination $v = \sum_{j=1}^{n} a_j v_j$ for unique scalars $a_j \in F$.

$\mathcal{X}$ is a state set: a finite but very large set of discrete structures, page 1.

A function $f : \mathcal{X} \to \mathbb{R}$ is an objective function, page 1.

$(\mathcal{X}, f)$ is a combinatorial optimization problem: a set of states along with an objective function $f : \mathcal{X} \to \mathbb{R}$.

$\mathscr{F}(\mathcal{X})$ is the set of all real functions $\{f : \mathcal{X} \to \mathbb{R}\}$. It is a vector space isomorphic to $\mathbb{R}^{|\mathcal{X}|}$, page 13.

$\{\varphi_i\}$ is a set of basis functions for $\mathscr{F}(\mathcal{X})$, page 15.

$\{e_z\}$ is the *standard basis* for $\mathscr{F}(\mathcal{X})$. Each element $z \in \mathcal{X}$ has an associated standard basis function $e_x(x) = [x = z]$, page 13.

$\langle f \rangle_X$ is the average value of a function f over a set X where X is a subset of the domain of f, page 16.

$\bar{f}$ is a more concise way of writing $\langle f \rangle_{\mathcal{X}}$, page 52.

f^c is the c^{th} power of a function f, page 70.

$\mu_c(X)$ is the c^{th} moment of a function f over a subset X of its domain, page 68.

$N : \mathcal{X} \to 2^{\mathcal{X}}$ is a neighborhood operator on $\mathcal{X}$. Here $2^{\mathcal{X}}$ denotes the powerset of $\mathcal{X}$: the set of all subsets of $\mathcal{X}$, page 14.

When N is *regular*, the *degree* of the neighborhood is denoted d, page 18.

Given a neighborhood operator, $\boldsymbol{A}$ is the (algebraic) adjacency operator corresponding to the neighborhood graph induced by N, page 14.

$\mathcal{H}(x, y)$: the Hamming distance between two strings x and y, page 32.

$S^{(r)}(x)$ is a Hamming sphere of radius r about a string x, page 69.

$B^{(r)}(x)$ is a Hamming ball of radius r about a string x, page 69.

$\alpha(x,y)$ and $\beta(x,y)$ are *approaching* and *retreating* sets. They are sets of points in $\{0,1\}^n$ that partition the neighborhood of y which lies at some Hamming distance from x, page 74.

$\boldsymbol{S}^{(r)}$ is the radius r sphere matrix, page 73.

$\{\gamma_i^{(r)}\}$ is the spectrum (i.e., set of eigenvalues) of the radius r sphere matrix, page 77.

$\alpha(x,y)$ is the approaching set with respect to two strings x and y, page 72.

$\beta(x,y)$ is the retreating set with respect to two strings x and y, page 72.

$\langle x, y\rangle$: the string inner product of two strings x and y, page 32.

$\mathscr{P}$: the Heckendorn Pack Function, page 34.

$\oplus$ denotes the logical exclusive-or operation, page 32.

ψ_i is the i^{th} Walsh function. The *order* of ψ_i is the number of elements in the binary string representation of i that are equal to 1, page 38.

w_i denotes the i^{th} Walsh coefficient, page 38.

Ψ_p: a linear combination of Walsh functions of order p, page 41.

$\mathfrak{w}_j$ is the sum of degree c monomials on Walsh coefficients such that the bitwise exclusive disjunction of the binary string representations of the coefficient indexes in each monomial is equal to j, page 70.

$\mathcal{W}(f)$ is the bound of nonzero Walsh coefficients in the Walsh basis expansion of f, page 71.

Q_n denotes the hypercube graph of order n: a regular graph with 2^n vertices, each of which correspond to a unique element of $\{0,1\}^n$, page 25.

$\mathfrak{V}$ denotes a set of Boolean variables, page 46.

$\mathfrak{C}$ denotes a set of disjunctive Boolean clauses, page 46.

$\mathcal{A} : \mathfrak{V} \to \{0,1\}$ is a Boolean assignment, page 46.

τ: a positive real number that defines the objective function range outside of which certain search space structures are forbidden in Max-3-Sat, page 57.

$\mathcal{K}_r(x,n)$ is the order r Krawtchouk polynomial, page 78.

p_X is the frequency distribution of codomain values of a function over a subset X of its domain, page 86.

REFERENCES

[AER96] Torsten Asselmeyer, Werner Ebeling, and Helge Rose. Smoothing representation of fitness landscapes – the genotype-phenotype map of evolution. *BioSystems*, 39(1):63–76, 1996.

[AL03] Emile Aarts and Jan Karel Lenstra, editors. *Local Search in Combinatorial Optimization.* Princeton University Press, 2003.

[And88] Philip W. Anderson. Spin glass hamiltonians: A bridge between biology, statistical mechanics and computer science. In David Pines, editor, *Emerging Synthesis in Science: Proceedings of the Founding Workshops of the Santa Fe Institute.* Santa Fe Institute (Santa Fe, NM), Perseus Books Publishing, L.L.C., 1988.

[AW02] Takao Asano and David P. Williamson. Improved approximation algorithms for MAX-SAT. *Journal of Algorithms*, 42:173–202, 2002.

[AZ98] Eric Angel and Vassilis Zissimopoulos. Autocorrelation coefficient for the graph bipartitioning problem. *Theoretical Computer Science*, 191:229–243, 1998.

[AZ00] Eric Angel and Vassilis Zissimopoulos. On the classification of NP-complete problems in terms of their correlation coefficient. *Discrete Applied Mathematics*, 99:261–277, 2000.

[AZ01] Eric Angel and Vassilis Zissimopoulos. On the landscape ruggedness of the Quadratic Assignment Problem. *Theoretical Computer Science*, 263(1–2):159–172, 2001.

[Bäc92a] Thomas Bäck. The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm. In Reinhard Männer and Bernard Manderick, editors, *Parallel Problem Solving from Nature 2*, pages 85–94, Amsterdam, 1992. Elsevier.

[Bäc92b] Thomas Bäck. Self-adaptation in genetic algorithms. In Francisco J. Varela and Paul Bourgnine, editors, *Proceedings of the First European*

Conference on Artificial Life, pages 263–271, Cambridge, MA, 1992. The MIT Press.

[Bäc93] Thomas Bäck. Optimal mutation rates in genetic search. In Stephanie Forrest, editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 2–8, San Mateo, CA, 1993. Morgan Kaufmann.

[Bar98] Lionel Barnett. Ruggedness and neutrality: the NKp family of fitness landscapes. In *Proceedings of the Sixth International Conference on Artificial Life*, pages 18–27, Cambridge, MA, USA, 1998. MIT Press.

[BBK$^+$00] Christopher L. Barrett, B. W. Bush, Stephan Kopp, Henning S. Mortveit, and Christian M. Reidys. Sequential dynamical systems and applications to simulations. In *Proceedings of the Thirty-third Annual Simulation Symposium (SS 2000)*, pages 245–252, Los Alamitos, CA, USA, 2000. IEEE Computer Society.

[BC01] J. Wesley Barnes and Bruce Colletti. Local search structure in the symmetric traveling salesperson problem under a general class of rearrangement neighborhoods. *Applied Mathematics Letters*, 14(1):105–108, 2001.

[BDD03] J. Wesley Barnes, Boryana Dimova, and Steftcho P. Dokov. The theory of elementary landscapes. *Applied Mathematics Letters*, 16(3):337–343, April 2003.

[Bet80] Albert D. Bethke. *Genetic Algorithms as Function Optimizers*. PhD thesis, University of Michigan, 1980.

[BHWR06] Laura Barbulescu, Adele E. Howe, L. Darrell Whitley, and Mark Roberts. Understanding algorithm performance on an oversubscribed scheduling application. *Journal of Artificial Intelligence Research*, 27:577–615, Dec 2006.

[BLS07] Türker Bıyıkoğlu, Josef Leydold, and Peter F. Stadler. *Laplacian Eigenvectors of Graphs*, volume 1915 of *Lecture Notes in Mathematics*. Springer, 2007.

[Boc58] Frederick Bock. An algorithm for solving "traveling-salesman" and related network optimization problems. In *Bulletin of the Fourteenth National Meeting of the Operations Research Society of America*, volume 897, 1958.

[BS96] Thomas Bäck and Martin Schütz. Intelligent mutation rate control in canonical genetic algorithms. In Zbigniew Ras and Maciek Michalewicz, editors, *Foundations of Intelligent Systems*, volume 1079 of *Lecture Notes in Computer Science*, pages 158–167. Springer Berlin / Heidelberg, 1996.

[BVCE06] William Beaudoin, Sébastien Verel, Philippe Collard, and Cathy Escazut. Deceptiveness and neutrality the ND family of fitness landscapes. In *Proceedings of the Eighth Annual Conference on Genetic and Evolutionary Computation (GECCO-2006)*, 2006.

[CB00] Bruce Colletti and J. Wesley Barnes. Linearity in the traveling salesman problem. *Applied Mathematics Letters*, 13(3):27–32, April 2000.

[Čer85] Vlado Černý. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. *Journal of Optimization Theory and Applications*, 45:41–51, 1985. 10.1007/BF00940812.

[CFG$^+$96] David A. Clark, Jeremy Frank, Ian P. Gent, Ewan MacIntyre, Neven Tomov, and Toby Walsh. Local search and the number of solutions. In *Principles and Practice of Constraint Programming*, pages 119–133, 1996.

[CFZ97] Jianer Chen, Donald K. Friesen, and Hao Zheng. Tight bound on Johnson's algorithms for Max-SAT. In *Proceedings of the Twelfth Annual IEEE Conference on Computational Complexity*, pages 274–281, Los Alamitos, CA, 1997. IEEE Computer Society Press.

[CJK08] Sung-Soon Choi, Kyomin Jung, and Jeong Han Kim. Almost tight upper bound for finding Fourier coefficients of bounded pseudo-Boolean functions. In Rocco A. Servedio and Tong Zhang, editors, *Proceedings of the Twenty-first Conference on Learning Theory (COLT 2008)*, pages 123–134, Helsinki, Finland, July 2008. Omnipress.

[CM92] Bruno Codenotti and Luciano Margara. Local properties of some NP-complete problems. Technical Report TR 92-021, International Computer Science Institute, Berkeley, CA, 1992.

[Coo71] Stephen A. Cook. The complexity of theorem proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, pages 151–158, May 1971.

[Cro58] G. A. Croes. A method for solving traveling-salesman problems. *Operations Research*, 6(6):791–812, 1958.

[DBP05] Boryana Dimova, J. Wesley Barnes, and Elmira Popova. Arbitrary elementary landscapes & ar(1) processes. *Applied Mathematics Letters*, 18(3):287–292, 2005.

[De 75] Kenneth A. De Jong. *An analysis of the behavior of a class of genetic adaptive systems*. PhD thesis, University of Michigan, 1975.

[Dec03] Rina Dechter. *Constraint processing*. Morgan Kaufmann, 2003.

[DJW98] Stefan Droste, Thomas Jansen, and Ingo Wegener. A rigorous complexity analysis of the $(1 + 1)$ evolutionary algorithm for separable functions with boolean inputs. *Evolutionary Computation*, 6(2):185–196, 1998.

[dOFS99] Viviane M. de Oliveira, José F. Fontanari, and Peter F. Stadler. Metastable states in short-ranged p-spin glasses. *Journal of Physics A: Mathematical and General*, 32:8793–8802, 1999.

[EA75] Samuel F. Edwards and Philip W. Anderson. Theory of spin glasses. *Journal of Physics F: Metal Physics*, 5:965–974, 1975.

[EGS10] Stefano Ermon, Carla Gomes, and Bart Selman. Computing the density of states of Boolean formulas. In *Proceedings of the 16th International Conference on Principles and Practice of Constraint Programming*, 2010.

[EMS88] Manfred Eigen, John McCaskill, and Peter Schuster. Molecular quasispecies. *Journal of Physical Chemistry*, 92(24):6881–6891, Dec 1988.

[Erd59] Paul Erdős. Graph theory and probability. *Canadian Journal of Mathematics*, 11:34–38, 1959.

[FA86] Yaotian Fu and Philip W. Anderson. Application of statistical mechanics to NP-complete problems in combinatorial optimization. *Journal of Physics A*, 19:1605–1620, 1986.

[FCS97] Jeremy Frank, Peter Cheeseman, and John Stutz. When gravity fails: Local search topology. *Journal of Artificial Intelligence Research*, 7:249–281, 1997.

[FFHS00] Christian Flamm, Walter Fontana, Ivo L. Hofacker, and Peter Schuster. RNA folding at elementary step resolution. *RNA*, 6(3):325–338, 2000.

[FFS00] Fernando F. Ferreira, José F. Fontanari, and Peter F. Stadler. Landscape statistics of the low autocorrelated binary string problem. *Journal of Physics A*, 33(48):8635–8647, 2000.

[FHSS07] Christoph Flamm, Ivo L. Hofacker, Bärbel M. R. Stadler, and Peter F. Stadler. Saddles and barrier in landscapes of generalized search operators. In *Foundations of Genetic Algorithms IX*, volume 4436 of *Lecture Notes in Computer Science*, pages 194–212, 2007.

[FHSW02] Christoph Flamm, Ivo L. Hofacker, Peter F. Stadler, and Michael T. Wolfinger. Barrier Trees of Degenerate Landscapes. *Zeitschrift für Physikalische Chemie*, 216:155–173, 2002.

[FL70] Ian Franklin and R. C. Lewontin. Is the gene the unit of selection? *Genetics*, 65(4):707–734, August 1970.

[Flo56] Merrill M. Flood. The traveling-salesman problem. *Operations Research*, 4(1):61–75, 1956.

[FSBB^{+}93] Walter Fontana, Peter F. Stadler, Erich G. Bornberg-Bauer., Thomas Griesmacher, Ivo L. Hofacker, Manfred Tacker, Pedro Tarazona, Edward D. Weinberger, and Peter Schuster. RNA folding and combinatory landscapes. *Physical Review E*, 47(3):2083–2099, 1993.

[GJS76] Michael R. Garey, David S. Johnson, and Larry J. Stockmeyer. Some simplified NP-complete graph problems. *Theoretical Computer Science*, 1(3):237–267, 1976.

[GKS99] Josselin Garnier, Leila Kallel, and Marc Schoenauer. Rigorous hitting times for binary mutations. *Evolutionary Computation*, 7(2):167–203, 1999.

[Gol89] David E. Goldberg. Genetic algorithms and Walsh functions. *Complex Systems*, 3:129–171, 1989.

[Gol92] David E. Goldberg. Construction of high-order deceptive functions using low-order Walsh coefficients. *Annals of Mathematics and Artificial Intelligence*, 5(1):35–47, 1992.

[GPS97] Ricardo García-Pelayo and Peter F. Stadler. Correlation length, isotropy and meta-stable states. *Physica D*, 107:240–254, 1997.

[Gre86] John J. Grefenstette. Optimization of control parameters for genetic algorithms. *IEEE Transactions on Systems, Man, and Cybernetics*, 16(1):122–128, 1986.

[Gro92] Lov K. Grover. Local search and the local structure of NP-complete problems. *Operations Research Letters*, 12:235–243, 1992.

[GW93a] Ian P. Gent and Toby Walsh. An empirical analysis of search in GSAT. *Journal of Artificial Intelligence Research*, 1:47–59, 1993.

[GW93b] Ian P. Gent and Toby Walsh. Towards an understanding of hill-climbing procedures for SAT. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 28–33. MIT Press, 1993.

[Haj88] Bruce Hajek. Cooling schedules for optimal annealing. *Mathematics of Operations Research*, 13:311–329, 1988.

[Hal63] P. R. Halmos. What does the spectral theorem say? *The American Mathematical Monthly*, 70(3):241–247, 1963.

[Han90] Pierre Hansen. Algorithms for the maximum satisfiability problem. *Computing*, 44(4):279–303, 1990.

[Hås01] Johan Håstad. Some optimal inapproximability results. *Journal of the ACM*, 48(4):798–859, 2001.

[Hec99] Robert B. Heckendorn. *Walsh Analysis, Epistasis, and Optimization Problem Difficulty for Evolutionary Algorithms.* PhD thesis, Colorado State University, Fort Collins, CO, July 1999.

[Hec02] Robert B. Heckendorn. Embedded landscapes. *Evolutionary Computation*, 10(4):345–369, 2002.

[HJB84] Michael T. Heideman, Don H. Johnson, and C. Sidney Burrus. Gauss and the history of the fast Fourier transform. *ASSP Magazine, IEEE*, 1(4):14 – 21, October 1984.

[HK93] Steven Hampson and Dennis Kibler. Plateaus and plateau search in Boolean satisfiability problems: When to give up searching and start again. *DIMACS Series in Discrete Math and Theoretical Computer Science*, 26:437–453, October 1993.

[HLO08] Federico Heras, Javier Larrosa, and Albert Oliveras. MiniMaxSat: an efficient weighted Max-SAT solver. *Journal of Artificial Intelligence Research*, 31:1–32, 2008.

[HM91] Jürgen Hesser and Reinhard Männer. Towards an optimal mutation probability for genetic algorithms. In Hans-Paul Schwefel and Reinhard Männer, editors, *Parallel Problem Solving from Nature*, volume 496 of *Lecture Notes in Computer Science*, pages 23–32. Springer Berlin / Heidelberg, 1991.

[HM08] Mathias Herrmann and Alexander May. Solving linear equations modulo divisors: On factoring given any bits. In *Advances in Cryptology – ASIACRYPT 2008*, pages 406–424. Springer, 2008.

[Hol75] John H. Holland. *Adaptation in Natural and Artificial Systems.* The University of Michigan Press, 1975.

[Hoo96] John N. Hooker. Testing heuristics: We have it all wrong. *Journal of Heuristics*, 1:33–42, 1996.

[Hoo98] Holger H. Hoos. *Stochastic Local Search – Methods, Models, Applications.* PhD thesis, TU Darmstadt, 1998.

[Hoo99] Holger H. Hoos. On the run-time behaviour of stochastic local search algorithms for SAT. In *Proceedings of the National Conference on Artificial Intelligence*, pages 661–666. John Wiley & Sons, Inc, 1999.

[Hor97] Wim Hordijk. Correlation analysis of the synchronizing-CA landscape. *Physica D*, 107:255–264, 1997.

[HRW98] Robert B. Heckendorn, Soraya B. Rana, and L. Darrell Whitley. Test function generators as embedded landscapes. In Wolfgang Banzhaf and Colin R. Reeves, editors, *Foundations of Genetic Algorithms V*, pages 183–198. Morgan Kaufmann, 1998.

[HRW99] Robert B. Heckendorn, Soraya Rana, and Darrell Whitley. Polynomial time summary statistics for a generalization of MAXSAT. In *Genetic and Evolutionary Computation Conference (GECCO-1999)*, pages 281–288, 1999.

[HS04] Holger H. Hoos and Thomas Stützle. *Stochastic Local Search: Foundations and Applications.* Morgan Kaufman, 2004.

[HW97] Robert Heckendorn and Darrell Whitley. A Walsh analysis of NK-landscapes. In *Proceedings of the Seventh International Conference on Genetic Algorithms*, 1997.

[HW99] Robert B. Heckendorn and L. Darrell Whitley. Predicting epistasis directly from mathematical models. *Evolutionary Computation*, 7:69–101, 1999.

[HW04] Robert B. Heckendorn and Alden H. Wright. Efficient linkage discovery by limited probing. *Evolutionary Computation*, 12:517–545, 2004.

[Ive62] Kenneth E. Iverson. *A Programming Language.* Wiley, New York, 1962.

[JM97] David S. Johnson and Lyle A. McGeoch. The traveling salesman problem: A case study in local optimization. In Emile H. L. Aarts and Jan Karel Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 215–310. John Wiley and Sons Ltd, 1997.

[JM04] David S. Johnson and Lyle A. McGeoch. Experimental analysis of heuristics for the STSP. In Ding-Zhu Du, Panos M. Pardalos, Gregory Gutin, and Abraham P. Punnen, editors, *The Traveling Salesman Problem and Its Variations*, volume 12 of *Combinatorial Optimization*, pages 369–443. Springer US, 2004.

[Joh74] David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Science*, 9(3):256–278, 1974.

[Jon95] Terry Jones. *Evolutionary Algorithms, Fitness Landscapes and Search*. PhD thesis, University of New Mexico, Albuquerque, New Mexico, May 1995.

[JW00] Thomas Jansen and Ingo Wegener. On the choice of the mutation probability for the (1+1) EA. In Marc Schoenauer, Kalyanmoy Deb, Günther Rudolph, Xin Yao, Evelyne Lutton, Juan Merelo, and Hans-Paul Schwefel, editors, *Parallel Problem Solving from Nature VI*, volume 1917 of *Lecture Notes in Computer Science*, pages 89–98. Springer Berlin / Heidelberg, 2000.

[JW07] Thomas Jansen and Ingo Wegener. A comparison of simulated annealing with a simple evolutionary algorithm on pseudo-Boolean functions of unitation. *Theoretical Computer Science*, 386(1-2):73 – 93, 2007.

[Kau93] Stuart A. Kauffman. *The Origins of Order*. Oxford University Press, 1993.

[Ker93] Walter Kern. On the depth of combinatorial optimization problems. *Discrete Applied Mathematics*, 43:115–129, 1993.

[KGV83] Scott Kirkpatrick, C. Daniel Gelatt, and Mario P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

[KL87] Stuart Kauffman and Simon Levin. Towards a general theory of adaptive walks on rugged landscapes. *Journal of Theoretical Biology*, 128:11–45, 1987.

[Knu92] Donald E. Knuth. Two notes on notation. *American Mathematical Monthly*, 99(5):403–422, May 1992.

[KP01] Hillol Kargupta and Byung-hoon Park. Gene expression and fast construction of distributed evolutionary representation. *Evolutionary Computation*, 9(1):43–69, 2001.

[Kra29] Mikhail Kravchuk. Sur une généralisation des polynomes d'Hermite. *Comptes rendus de l'Académie des sciences*, 189(17):620–622, 1929.

[KS96] Bärbel Krakhofer and Peter F. Stadler. Local minima in the graph bipartitioning problem. *Europhysics Letters*, 34:85–90, 1996.

[KSGS09] Lukas Kroc, Ashish Sabharwal, Carla P. Gomes, and Bart Selman. Integrating systematic and local search paradigms: A new strategy for maxSAT. In *Proceedings of the Twenty-first International Joint Conferenance on Artificial Intelligence (IJCAI 2009)*, 2009.

[KT85] Scott Kirkpatrick and Gérard Toulouse. Configuration space analysis of travelling salesman problems. *Journal de Physique*, 46(8):1277–1292, 1985.

[KZ97] Howard Karloff and Uri Zwick. A 7/8-approximation algorithm for MAX 3SAT? In *Proceedings of the Thirty Eighth Annual IEEE Symposium on Foundations of Computer Science*, pages 406–415, Los Alamitos, CA, 1997. IEEE Computer Society Press.

[LH05] Chu Min Li and Wen Qi Huang. Diversification and determinism in local search for satisfiability. In *Proceedings of the Eighth International Conference on Theory and Applications of Satisfiability Testing (SAT-05)*, volume 3569 of *Lecture Notes in Computer Science*, pages 158–172, 2005.

[Lin65] Shen Lin. Computer solutions of the traveling salesman problem. *Bell System Technical Journal*, 44:2245–2269, 1965.

[LK73] Shen Lin and Brian W. Kernighan. An effective heuristic algorithm for the traveling salesman problem. *Operations Research*, 21:498–516, 1973.

[LV91] Gunar E. Liepins and Michael D. Vose. Polynomials, basis sets, and deceptiveness in genetic algorithms. *Complex Systems*, 5:45–61, 1991.

[LWZ07] Chu Min Li, Wanxia Wei, and Harry Zhang. Combining adaptive noise and look-ahead in local search for SAT. In *Proceedings of the Tenth International Conference on Theory and Applications of Satisfiability Testing*, volume 4501 of *Lecture Notes in Computer Science*, pages 121–133. Springer, 2007.

[Mak08] Andrew O. Makhorin. GLPK: GNU Linear Programming Kit [computer software]. Available from `http://www.gnu.org/software/glpk/`, 2000–2008.

[MF97] Peter Merz and Bernd Freisleben. A genetic local search approach to the quadratic assignment problem. In Thomas Bäck, editor, *Proceedings of the Seventh International Conference on Genetic Algorithms (ICGA97)*, San Francisco, CA, 1997. Morgan Kaufmann.

[MF00] Peter Merz and Bernd Freisleben. Fitness landscape analysis and memetic algorithms for the quadratic assignment problem. *IEEE Transactions on Evolutionary Computation*, 4(4):337–352, November 2000.

[MG05] Monaldo Mastrolilli and Luca Maria Gambardella. How good are tabu search and plateau moves in the worst case? *European Journal of Operations Research*, 166:63–76, 2005.

[MP89] Catherine A. Macken and Alan S. Perelson. Protein evolution on rugged landscapes. *Proceedings of the National Academy of Sciences of the United States of America*, 86(16):6191, 1989.

[MRR$^+$53] Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of state calculations by fast computing machines. *The Journal of Chemical Physics*, 21(6):1087–1092, 1953.

[MSK97] David McAllester, Bart Selman, and Henry Kautz. Evidence for invariants in local search. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence (AAAI-97)*, 1997.

[Müh92] Heinz Mühlenbein. How genetic algorithms really work: I. mutation and hillclimbing. In Reinhard Männer and Bernard Manderick, editors, *Parallel Problem Solving from Nature 2*, pages 15–25. Elsevier, Amsterdam, 1992.

[MZK$^+$99] Rémi Monasson, Riccardo Zecchina, Scott Kirkpatrick, Bart Selman, and Lidror Troyansky. Determining computational complexity from characteristic 'phase transitions'. *Nature*, 400:133–137, 1999.

[NS96] Eugeniusz Nowicki and Czeslaw Smutnicki. A fast taboo search algorithm for the job shop problem. *Management Science*, 42:797–813, June 1996.

[PB10] Denis Pankratov and Allan Borodin. On the relative merits of simple local search methods for the MAX-SAT problem. In Ofer Strichman and Stefan Szeider, editors, *Theory and Applications of Satisfiability Testing SAT 2010*, volume 6175 of *Lecture Notes in Computer Science*, pages 223–236. Springer Berlin / Heidelberg, 2010.

[PH06] Wayne Pullan and Holger H. Hoos. Dynamic local search for the maximum clique problem. *Journal of Artificial Intelligence Research*, 25:159–185, 2006.

[Pré90] András Prékopa. The discrete moment problem and linear programming. *Discrete Applied Mathematics*, 27(3):235–254, 1990.

[PW96] Andrew J. Parkes and Joachim P. Walser. Tuning local search for satisfiability testing. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence (AAAI-96)*, pages 356–362. MIT Press, 1996.

[RAB04] Colin R. Reeves and Mériéma Aupetit-Bélaidouni. Estimating the number of solutions for SAT problems. In *Proceedings of the 8th International Conference for Parallel Problem Solving from Nature VIII*, pages 101–110, Birminghan, UK, September 2004.

[RE04] Colin R. Reeves and Anton V. Eremeev. Statistical analysis of local search landscapes. *The Journal of the Operational Research Society*, 55(7):687–693, 2004.

[REA96] Helge Rose, Werner Ebeling, and Torsten Asselmeyer. The density of states: a measure of the difficulty of optimisation problems. In *Proceedings of the Fourth International Conference on Parallel Problem Solving from Nature*, pages 208–217. Springer Verlag, 1996.

[RHG07] Silvia Richter, Malte Helmert, and Charles Gretton. A stochastic local search approach to vertex cover. In *Proceedings of the Thirtieth German Conference on Artificial Intelligence (KI-2007)*, pages 412–426. Springer, 2007.

[RHW98] Soraya Rana, Robert B. Heckendorn, and L. Darrell Whitley. A tractable Walsh analysis of SAT and its implications for genetic algorithms. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence (AAAI-98)*, pages 392–397, 1998.

[RKHS02] Dan Rockmore, Peter Kostelec, Wim Hordijk, and Peter F. Stadler. Fast Fourier transform for fitness landscapes. *Applied and Computational Harmonic Analysis*, 12:57–76, 2002.

[RS01] Christian M. Reidys and Peter F. Stadler. Neutrality in fitness landscapes. *Applied Mathematics and Computation*, 117:321–350, 2001.

[RS02] Christian M. Reidys and Peter F. Stadler. Combinatorial landscapes. *SIAM Review*, 44:3–54, 2002.

[Rud97] Günter Rudolph. *Convergence Properties of Evolutionary Algorithms.* Verlag Dr Kovač, 1997.

[Rya95] Jennifer Ryan. The depth and width of local minima in discrete solution spaces. *Discrete Applied Mathematics*, 56(1):75 – 82, 1995.

[SBDA03] Andrew Solomon, J. Wesley Barnes, Steftcho P. Dokov, and Raul Acevedo. Weakly symmetric graphs, elementary landscapes, and the TSP. *Applied Mathematics Letters*, 16(3):401–407, 2003.

[SCED89] J. David Schaffer, Richard A. Caruana, Larry J. Eshelman, and Rajarshi Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In *Proceedings of the Third International Conference on Genetic Algorithms*, pages 51–60, San Francisco, CA, USA, 1989. Morgan Kaufmann Publishers Inc.

[SGS00] Josh Singer, Ian P. Gent, and Alan Smaill. Backbone fragility and the local search cost peak. *Journal of Artificial Intelligence Research*, 12:235–270, 2000.

[SH99] Peter F. Stadler and Robert Happel. Random field models for fitness landscapes. *Journal of Mathematical Biology*, 38(5):435–478, 1999.

[SHW09] Andrew M. Sutton, Adele E. Howe, and L. Darrell Whitley. A theoretical analysis of the k-satisfiability search space. In Thomas Stützle, Mauro Birattari, and Holger H. Hoos, editors, *Proceedings of the Second International Workshop on Engineering Stochastic Local Search Algorithms (SLS 2009)*, volume 5752 of *Lecture Notes in Computer Science*, pages 46–60, Brussels, Belgium, September 2009. Springer.

[SHW10] Andrew M. Sutton, Adele E. Howe, and L. Darrell Whitley. Directed plateau search for MAX-k-SAT. In *Proceedings of the Third Annual Symposium on Combinatorial Search*, Atlanta, GA, 2010.

[SK75] David Sherrington and Scott Kirkpatrick. Solvable model of a spin-glass. *Physical Review Letters*, 35(26):1792–1796, Dec 1975.

[SK93] Bart Selman and Henry A. Kautz. Domain-independent extensions to GSAT: Solving large structured variables. In *Proceedings of the Thirteenth International Joint Conference on Artificial Intelligence (IJCAI-93)*, pages 290–295, 1993.

[SKC94] Bart Selman, Henry A. Kautz, and Bram Cohen. Noise strategies for improving local search. In *Proceedings of the Twelfth National Conference on Artificial Intelligence (AAAI-94)*, pages 337–343, Seattle, 1994.

[SKC96] Bart Selman, Henry Kautz, and Bram Cohen. Local search strategies for satisfiability testing. In David S. Johnson and Michael A. Trick, editors, *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, volume 26. AMS, 1996.

[SLM92] Bart Selman, Hector Levesque, and David Mitchell. A new method for solving hard satisfiability problems. In *Proceedings of the Tenth National Conference on Artificial Intelligence (AAAI-92)*, San Jose, CA, 1992.

[Smy04] Kevin R. G. Smyth. Understanding stochastic local search algorithms: An empirical analysis of the relationship between search space structure and algorithm behaviour. Master's thesis, University of British Columbia, 2004.

[SS92] Peter F. Stadler and Wolfgang Schnabl. The landscape of the traveling salesman problem. *Physics Letters A*, 161:337–344, 1992.

[SS03] Tommaso Schiavinotto and Thomas Stützle. Search space analysis of the linear ordering problem. In Günther R. Raidl, Jean-Arcady Meyer, Martin Middendorf, Stefano Cagnoni, Juan J. Romero Cardalda, David Corne, Jens Gottlieb, Agnès Guillot, Emma Hart, Colin G. Johnson, and Elena Marchiori, editors, *EvoWorkshops*, volume 2611 of *Lecture Notes in Computer Science*, pages 322–333. Springer, 2003.

[Sta95] Peter F. Stadler. Toward a theory of landscapes. In Ramon Lopéz-Peña, Riccardo Capovilla, Ricardo García-Pelayo, Henri Waelbroeck, and Federico Zertruche, editors, *Complex Systems and Binary Networks*, pages 77–163. Springer Verlag, 1995.

[Sta96] Peter F. Stadler. Landscapes and their correlation functions. *Journal of Mathematical Chemistry*, 20:1–45, 1996.

[SWH09] Andrew M. Sutton, L. Darrell Whitley, and Adele E. Howe. A polynomial time computation of the exact correlation structure of k-satisfiability landscapes. In *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-09)*, 2009.

[SWH11a] Andrew M. Sutton, L. Darrell Whitley, and Adele E. Howe. Approximating the distribution of fitness over Hamming regions. In *Proceedings of Foundations of Genetic Algorithms XI*, 2011.

[SWH11b] Andrew M. Sutton, L. Darrell Whitley, and Adele E. Howe. Computing the moments of k-bounded pseudo-Boolean functions over Hamming spheres of arbitrary radius in polynomial time. *Theoretical Computer Science*, 2011. In press.

[SY11] Chi Wan Sung and Shiu Yin Yuen. Analysis of $(1 + 1)$ evolutionary algorithm and randomized local search with memory. *Evolutionary Computation*, 2011. In press.

[Ter99] Audrey Terras. *Fourier Analysis on Finite Groups and Applications, Cambridge U. Press, Cambridge.* Cambridge University Press, 1999.

[TH03] Dave A.D. Tompkins and Holger H. Hoos. Scaling and probabilistic smoothing: Dynamic local search for unweighted MAX-SAT. In *Sixteenth Conference of the Canadian Society for Computational Studies of Intelligence*, 2003.

[TSY88] Chiang Tzuu-Shuh and Chow Yunshyong. On the convergence rate of annealing processes. *SIAM Journal on Control and Optimization*, 26:1455–1570, October 1988.

[TSY89] Chiang Tzuu-Shuh and Chow Yunshyong. A limit theorem for a class of inhomogeneous Markov processes. *Annals of Probability*, 17:1438–1502, 1989.

[Val79] Leslie G. Valiant. The complexity of computing the permanent. *Theoretical Computer Science*, 8(2):189 – 201, 1979.

[vLA87] Peter J. M. van Laarhoven and Emile H. L. Aarts. *Simulated annealing: theory and applications.* Springer, 1987.

[VW98a] Michael D. Vose and Alden H. Wright. The simple genetic algorithm and the Walsh transform: Part I, theory. *Evolutionary Computation*, 6(3):253–273, 1998.

[VW98b] Michael D. Vose and Alden H. Wright. The simple genetic algorithm and the Walsh transform: Part II, the inverse. *Evolutionary Computation*, 6(3):275–289, 1998.

[VWW06] Virginia Vassilevska, Ryan Williams, and Shan Leung Maverick Woo. Confronting hardness using a hybrid approach. In *Proceedings of the seventeenth annual ACM-SIAM symposium on Discrete Algorithms*, 2006.

[Wal23] Joseph L. Walsh. A closed set of normal orthogonal functions. *American Journal of Mathematics*, 45(1):5–24, 1923.

[Wat03] Jean-Paul Watson. *Empirical Modeling and Analysis of Local Search Algorithms for the Job-Shop Scheduling Problem.* PhD thesis, Colorado State University, Department of Computer Science, 2003.

[WBHW03] Jean-Paul Watson, J. Christopher Beck, Adele E. Howe, and L. Darrell Whitley. Problem difficulty for tabu search in job-shop scheduling. *Artificial Intelligence*, 143(2):189–217, 2003.

[WBWH02] Jean-Paul Watson, Laura Barbulescu, L. Darrell Whitley, and Adele E. Howe. Contrasting structured and random permutation flow-shop scheduling problems: Search space topology and algorithm performance. *INFORMS Journal on Computing*, 14(2):98–123, 2002.

[Wei90] Edward D. Weinberger. Correlated and uncorrelated fitness landscapes and how to tell the difference. *Biological Cybernetics*, 63:325–336, 1990.

[WHW03] Jean-Paul Watson, Adele E. Howe, and L. Darrell Whitley. An analysis of iterated local search for job shop scheduling. In *Proceedings of the Fifth Metaheuristics International Conference*, pages 1101–1106, Kyoto, Japan, 2003.

[Wit06] Carsten Witt. Runtime analysis of the $(\mu+1)$ EA on simple pseudo-Boolean functions. *Evolutionary Computation*, 14(1):65–86, 2006.

[Wit09] Carsten Witt. Greedy local search and vertex cover in sparse random graphs. In Jianer Chen and S. Barry Cooper, editors, *Theory and Applications of Models of Computation*, volume 5532 of *Lecture Notes in Computer Science*, pages 410–419. Springer Berlin / Heidelberg, 2009.

[Wri32] Sewall Wright. The roles of mutation, inbreeding, crossbreeding, and selection in evolution. In *Proceedings of the Sixth Congress of Genetics*, volume 1, 1932.

[WTZ00] Alden H. Wright, Richard K. Thompson, and Jian Zhang. The computational complexity of N-K fitness functions. *Evolutionary Computation*, 4(4):373–379, 2000.

[WW05] Ingo Wegener and Carsten Witt. On the analysis of a simple evolutionary algorithm on quadratic pseudo-Boolean functions. *Journal of Discrete Algorithms*, 3(1):61–78, 2005.

[Yan03] Mihalis Yannakakis. Computational complexity. In Emile H. L. Aarts and Jan Karel Lenstra, editors, *Local Search in Combinatorial Optimization*, pages 19–55. Princeton University Press, 2003.

[Yok97] Makoto Yokoo. Why adding more constraints makes a problem easier for hill-climbing algorithms: Analyzing landscapes of CSPs. In *Principles and Practice of Constraint Programming*, pages 356–370, 1997.

[Zwi99] Uri Zwick. Outward rotations: a tool for rounding solutions of semidefinite programming relaxations, with applications to MAX CUT and other problems. In *Proceedings of the Thirty-first Annual ACM Symposium on Theory of Computing*, pages 679–687, New York, NY, 1999. ACM Press.