12-1998

Fitness Landscape Analysis of Discrete Constrained Optimization Problems

Sai Prasanna Ravichandran
Western Michigan University

FITNESS LANDSCAPE ANALYSIS OF DISCRETE CONSTRAINED OPTIMIZATION PROBLEMS

by

Sai Prasanna Ravichandran

A Thesis
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Master of Science in Engineering
Department of Electrical and Computer Engineering

Western Michigan University
Kalamazoo, Michigan
December 1998

ACKNOWLEDGEMENTS

First and foremost I wish to express my deepest sense of gratitude to Dr. Greenwood, my advisor, but for whom this thesis would not have reached a successful completion. His methodical approach in guiding me from simple to complex problems helped me to conceptualize facts with clarity and yield better results. Besides being an excellent advisor, he has been a great source of help both as a person and as a teacher. It is with immense pleasure that I thank him.

I am grateful to Dr. Sharon Hu, professor Computer Science dept. Notre Dame, with whom Dr. Greenwood and I met periodically to discuss our progress in the research work. Her valuable suggestions during our meetings lead to marked developments in the research. I sincerely thank her for her continuous encouragement and guidance throughout the work.

I thank Dr. Hossein Mousavineshad and Dr. Damon Miller for consenting to be on my thesis committee and providing their feedback on my thesis in a short notice. I also thank Dr. Miller for helping me get over the hassles during the preparation of my thesis.

My sincere thanks are due to the office staff Judith, Jill, Lisa and Adam for all the help they had provided. I thank all my family members for their constant support, encouragement and prayers.

Acknowledgements—Continued

I would like to thank the Defense Advanced Research Project Agency (DARPA) for funding this research work.

Last but not the least I thank God for giving me the strength and courage to complete this thesis successfully.

Sai Prasanna Ravichandran

FITNESS LANDSCAPE ANALYSIS OF DISCRETE CONSTRAINED OPTIMIZATION PROBLEMS

Sai Prasanna Ravichandran, M.S.

Western Michigan University, 1998

The exponentially large size of the solution space of typical optimization problems precludes the use of any deterministic approach to search for an optimal solution. A more efficient and efficacious approach is to use heuristic algorithms based on rules of thumb to guide the search process in the solution space. Associated with every solution in the space is a real number called fitness that signifies the quality of a solution. This space and the fitness values together form the fitness landscapes. Knowledge about the topology of these fitness landscapes is vital for any heuristic search operator to expedite the search process and also, find a good solution.

Complete enumeration of the landscape is usually impractical. Moreover, the landscape is often n-dimensional with $n >> 3$, making it difficult to visualize. This research deals with the development of mathematical and graphical techniques to characterize the structure of fitness landscapes. A new 3-D graphical tool that can depict the topology of high dimensional fitness landscapes has been developed. This graphical approach provides a visual perception of the space to be explored, which can be used to guide and accelerate the search process.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER I

INTRODUCTION

Developing methodologies to solve complex combinatorial optimization problems (COP) is an area of intense research effort.

Combinatorial analysis is the study of the arrangement, grouping, ordering or selection of discrete objects. In combinatorial optimization, one is interested in finding the *best* arrangement. It is assumed that an optimal arrangement exists and the exact number of such arrangements is not important [13]. Unfortunately even moderately sized COPs have an extremely large number of possible solutions that complete enumeration is impractical. In fact most of these problems are NP-Complete. Typical examples include the famous traveling salesman problem [12], frequency assignment problem [7,3,17], hardware software codesign [9] etc.

In general, solving even moderately sized COP's is difficult. To illustrate this difficulty,. consider the traveling salesman problem. One has to find the shortest tour through N cities from an initial city, visiting each city once and returning at the end of the initial city. For as few as 12 cities the number of possible tours is in the millions growing as (N-1)/2! This is the size of the search space, the set of all possible tours, from which the shortest tour has to be found. An exhaustive search will always find the optimal arrangement of the cities. But, due to the exponential increase of the solution space, it may take ages to find it.

Clearly, an exhaustive search is not a feasible approach to find an optimal solution for most COPs. A more efficient approach is to use heuristic algorithms[1] that depend on rules of thumb to guide the search process. Evolutionary algorithms are heuristic algorithms based on stochastic optimization, which utilize the principles of natural evolution. Their working mechanisms rely on the collective learning process within a population of individuals, each of which represents a search point. After an arbitrary initialization, the population evolves towards increasingly better regions of search space by means of a simulated process of selection, mutation and recombination, which are otherwise known as search operators. Better individuals are determined by a measure called *fitness* that quantifies the quality of an individual. This search space together with the fitness values associated with the individual form the *fitness landscape* [20,11]. The objective of any optimization algorithm is to explore the fitness landscape according to a well-defined set of rules to find a globally fit individual. Typically this landscape will be high dimensional.

The distribution of fitness values determines the ruggedness of the fitness landscape. It may be very rugged with many peaks of high fitness and profound sink holes or it may be smooth with low hills and gentle valleys. In this framework, adaptive evolution is a hill climbing process on a landscape. Initially, the process starts

[1] A *heuristic* is a method that produces a good but not necessarily optimal solution to a problem

with a population, which can be thought of as a cluster of individuals located at different points on the landscape. Search operators such as mutation, crossover etc., produces offspring's. From this population of parents and offsprings, highly fit individuals are retained in a manner that moves the population to regions of better fitness on the landscape, which is precisely the hill climbing process. Intuitively the behavior of an adapting population depends on how mountainous the fitness landscape is, on how large the population is and on the operators, which move an individual from one point to another point in the search space. Hence, it is useful to consider a landscape as an abstract mathematical tool, whose characteristics can be analyzed to provide useful information that will largely influence the performance of adaptive evolution in solving COPs. Manderick et al. [14] have shown that knowledge about the structure of the landscape can be used to define effective search operators for the evolutionary algorithms.

Many real world optimization problems are constrained in the sense that problem parameters must reside within certain ranges. Such constraints deform the natural fitness landscape in a way that introduces many more local optima [5]. Methods of characterization that involved taking a random walk on the landscape to collect statistical information that is representative of the entire landscape [8, 19] has been suggested previously. All these works presumed the landscape was statistically isotropic. In other words, independent of where the random walk begins the statistical information is invariant. However, for the class of constrained combinatorial optimization problems statistical isotropy is an incorrect assumption, due to the constraints that deforms the natural fitness landscape [5]. Hence, the landscape for a constrained optimization problem is anisotropic and even more difficult to search.

This thesis discusses the development of mathematical and graphical techniques to characterize the structure of fitness landscapes of such discrete constrained combinatorial optimization problems. Adequate characterization---in particular, visualization---of high dimensional fitness landscapes has proven to be difficult. The major contribution in this thesis, is the development of a new 3-D graphical tool that visualizes the search space to be explored. This graphical approach provides a visual picture of the space to be explored that can be used to guide and accelerate the search process to find the optimal solution.

CHAPTER II

RELATED RESEARCH

Knowledge about the structure of fitness landscapes of COP's can lead to an effective exploration strategy and hence an improvement in the performance of algorithms based on heuristics. Consequently, there has been a considerable increase in research effort to characterize the structure of fitness landscapes.

An analysis of the topology of the landscape determines its smoothness or ruggedness. A landscape is smooth if the neighboring points differ in their fitness values only by a small amount. Conversely, in a rugged landscape, the neighboring points differ markedly in their fitness values. A number of statistical measures like correlation, correlation length, number of local optima etc. exists to infer these characteristics. This section discusses some techniques developed by researchers to characterize the structure of fitness landscapes.

One of the important works in this area is the *NK* model developed by Kauffman [11]. In this model, *N* refers to the number of parts of a system (e.g., in the traveling salesman problem '*N*' is the number of cities). Each part makes a fitness contribution that depends upon that part and upon 'K' other parts among the *N*. In other words, *K* reflects how richly the system is cross-coupled. *K* is also known as epistatic interaction. If $K = 0$ there is no interaction and the landscape is smooth.

Conversely, as K approaches $N - 1$, the landscape becomes increasingly rugged with a large number of local optima. It is a tunable model that can be used to construct fitness landscapes and explore the efficacy of the techniques developed to characterize landscapes.

Weinberger [19] suggested using a random walk to gather statistical information about fitness landscapes. Starting at some random point on the landscape, the walk visits a randomly chosen neighbor. Repeating this process produces a sequence of fitness values $f_0, f_1 \ldots$ Weinberger assumed that there is some underlying distribution of fitness values and a random walk in any direction is sufficient to gather statistics about the landscape. The degree of correlation between any two points s steps apart in a random walk is given by a correlation function $R(s)$

$$R(s) = \frac{\langle f_t f_{t+s} \rangle - \langle f_t \rangle^2}{\sigma_f^{\,2}}$$

where $\langle . \rangle$ is the expected value over all pairs s steps apart and $\sigma_f^{\,2}$ is the variance of f_t. If a high degree of correlation exists, then the landscape is "smooth". In the sense that the neighboring points differ in fitness only by a small amount. Conversely, a low correlation means the landscape is "rugged" since neighboring points differ markedly in their fitness. With this knowledge about the characteristics of the landscapes, a search process can adapt its operators to find a better solution [19].

The autocorrelation function of random walks in fitness landscapes has also been explored by Manderick et al. [14]. This random walk was conducted by using rather sophisticated search operators. The goal here is to explore the strong relationship between the performance of different search operators of genetic algorithms on a fitness landscape and the statistical features of that landscape. They have examined the autocorrelation function, correlation length of the landscapes and the fitness correlation coefficients of corresponding search operators (The correlation length gives an indication of the largest distance between two points at which the value of one point can still provide some information about the expected value of the other point [8]. The correlation coefficient of an operator is related to the correlation length of the underlying fitness landscapes and expresses how correlated the landscape appears to that operator). Manderick et al. [14] have shown that these simple statistical measures can be used to tune different components of genetic algorithms and optimize their performance in solving COP's.

Hordijk [8] used both Weinberger's random walk [19] and a time series analysis known as Box and Jenkins approach [1] to characterize the structure of fitness landscapes. He showed that with the statistics obtained from a random walk on the fitness landscape, one could express the landscape as an autoregressive (AR) model. An autoregressive model of order p is expressed as

$$y_t = y_1 + \alpha_1 y_{t-1} + \ldots + \alpha_p y_{t-p} + \varepsilon_t$$

where the stochastic variable ε_t is white noise. In other words each value y_t in AR (p) process depends on p past values and some stochastic variable ε_t. The stochastic model obtained was shown to express the correlation structure of the fitness landscapes and also to predict the fitness expected during a search on the landscape.

Vassilev [18] proposed a different method of analysis and called it an 'information measure' of the landscape. He considered fitness landscapes as an ensemble of objects with different information characteristics. Viewing the landscape as a directed graph, the objects are a vertex and its neighboring vertices. The vertices in general can be classified as flat points (each vertex together with its neighbors belongs to a plain), isolated points (each vertex higher/lower than all neighbors) or as neither isolated or flat points (see Figure 1). Two information measures are used to specify the degree of ruggedness of an ensemble of basic objects: *information content* and *information stability*. Information content---specified by Shannon entropy reflects the number of local optima contained in the landscape. Information stability is determined from the difference between fitness values seen while conducting a random walk. More details for both these measures can be found in [18].

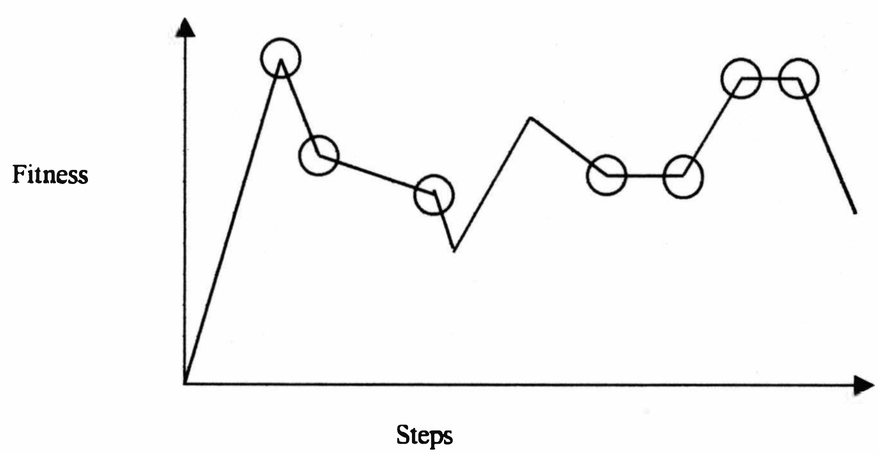

Figure 1. Landscape Path as an Ensemble of Seven Kinds of Objects.

All the above methods of analysis assumed the landscape to be statistically isotropic. In other words, it is assumed that the statistics obtained from a sufficiently long random walk is representative of the entire landscape. Greenwood and Hu [5] have shown that the landscapes of most constrained COP's are anisotropic, which means long random walk are incapable of giving accurate statistic information. In these problems, the goal is to optimize an objective function subject to a set of parametric constraints. Solutions are considered feasible, only if they do not violate constraints.

One of the most popular methods of handling infeasibility in any search process is by using a penalty function [15]. A penalty function artificially decreases a solution's fitness if constraints are violated. Usually, a penalty function is quite severe such that there is a low probability of survival during the evolutionary search process. The fact to be observed here is that an otherwise smooth landscape can now have numerous "sink holes" of extremely low fitness values. Moreover, these sink holes are not isolated points but instead are entire subregions of the fitness landscapes. The sink holes destroy any notion of an underlying distribution as the actual fitness value is artificially decreased for any infeasible solution. Consequently, the correlation present in the landscape can differ dramatically depending on where the initial starting point of a random walk is located. This can be seen in Figure 2. The shaded regions represent infeasible solutions on the anisotropic landscape. Two random walks beginning from different points A and B see completely different correlation on this landscape. Even repeating a random walk from the same point can produce completely different correlation depending on which regions of the landscape is traversed during the random walk. Both adaptive and random walks on landscapes with infeasible regions are analogous to runs from a non-stationary random process; statistics gathered from a single random walk will not be indicative of the true underlying correlation. Therefore, the techniques described earlier cannot accurately characterize the landscapes of constrained COP's.

Greenwood and Hu [6] suggested a method to characterize the fitness landscapes of constrained COP's. Their approach suggests conducting a random walk from a selected number of solutions, within a confined neighborhood and forming a composite picture of the landscape from these observations.

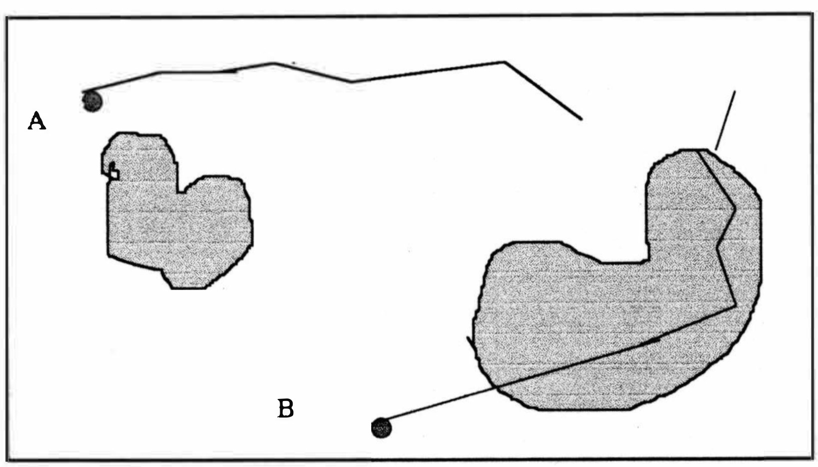

Figure 2. Random Walks on an Anisotropic Landscape.

In the research papers discussed earlier [8, 14, 18, 19], fitness landscape is viewed as a graphical plot of fitness versus solutions visited in a random walk, which was generated from search operators such as crossover, mutation etc. Hence, it has been claimed that the landscape is a function of operators. However, there are some disadvantages with this approach as some solutions may never get visited (except for mutation) and the information one can get out of this is limited.

A new graphical three-dimensional tool has been developed to represent high dimensional fitness landscapes. No assumption is made about the structure of the landscape. The graphical tool developed can assist in viewing the landscape at different levels of detail. Initially, one can view the landscape at a lower level of detail (without complete enumeration) and zoom into the regions of interest for finer details. The parts of solutions responsible for sink holes can be read directly from the gray code equivalent of the graph axis labels. Using this knowledge about the structure of the landscape, it is proposed to constrain the operators of heuristic algorithms such that they avoid the regions of sink holes or try to remain in the regions of high fitness on the landscape during the search process for an optimal solution. Incorporating this knowledge into the operators can expedite the search for an optimal solution.

CHAPTER III

PRELIMINARIES

This chapter provides an overview of the background required, for the discussions that will follow in the later chapters. The next section provides graph theoretical definitions, while the following sections describe gray codes and networks. The intent of this chapter is only to give a brief overview, detailed descriptions can be found in the references cited. Readers who are familiar with these topics can elect to skip this chapter.

Graph Theory

The material presented in this section, with a few modifications is taken from a book by Lawler [13].

A graph $G = (N, A)$ is a structure consisting of a finite set N of elements called nodes and a set A of unordered pairs of nodes called arcs. A directed graph or digraph is similar to a graph except that each arc is an ordered pair, giving it direction from one node to another. Nodes are also referred to as vertices or points and arcs as edges or lines.

In many applications in the physical, biological, social and engineering sciences graphs or digraphs have numerical values attached to their nodes or arcs of a graph to represent construction costs, flow capacities, probabilities of destruction etc. In general any graph to which such additional structure has been added is called a network.

For both directed and undirected graphs, an arc from node i to node j is denoted by *(i, j)*. Most problems deal with undirected graphs and at most one arc between a given pair of nodes i, j. Thus if $|N| = n$ and $|A| = m$, it follows that $m \leq n(n-1)/2$. In the case of directed graphs both *(i, j)* and *(j, i)* are permitted, so $m \leq n(n-1)$.

An undirected graph with four nodes and four interconnecting edges is shown in Figure 3.

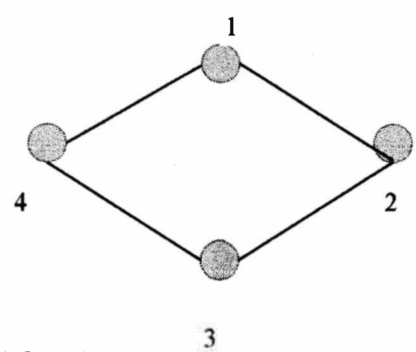

Figure 3. An Undirected Graph.

If there exists an arc *(i, j)* we say the nodes are *adjacent* or *neighbors*. By definition no node is *adjacent* to itself. For an undirected graph, the adjacency matrix $A = a\{ij\}$ is defined as follows:

$$a_{ij} = \begin{cases} 1 \text{ if there exists an arc } (i, j) \text{ between nodes } i \text{ and } j \\ 0 \text{ otherwise} \end{cases}$$

Consider the graph in Figure 3, the adjacency matrix for this graph is

$$A = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \end{pmatrix}$$

Gray Codes

A *Gray code sequence* is a sequence of strings in which successive strings differ in one and only one bit position. Let $b_1\ b_2\ \ldots\ b_n$ be a n-bit binary string where $b_i \in \{0,1\}$. There are 2^n unique binary patterns that can be formed with n-bit binary strings. A sequence of length L contains L distinct n-bit binary strings. This sequence is a gray code sequence if any two successive strings differ in only one bit position. Note that this requirement holds for the first and the last n-bit strings in the sequence. For example, {00,01,10,11} and {00,01,11,10} are both valid strings but only the second sequence is a gray code sequence. The nodes in the lattice and the m-ary n-cube (see next section in this chapter) are labeled in a gray code sequence. In other words all nodes in these networks are arranged as one mutant neighbors.

Networks

Networks are graphs with numerical values attached to their nodes. There are various topologies of networks. Of particular interest to this thesis are the lattice, torus and the *m*-ary *n*-cube.

A lattice is an M X L network with nodes arranged in M rows and L columns. Figure 4 shows an 8 X 8 lattice. Note that the rows and columns are labeled in gray code sequence. This ensures that any two nodes connected by an edge differ in only one bit value. In other words every node in the network has a 1- mutant neighbor. A torus is also a lattice, but with wrap around connections on each row and column.

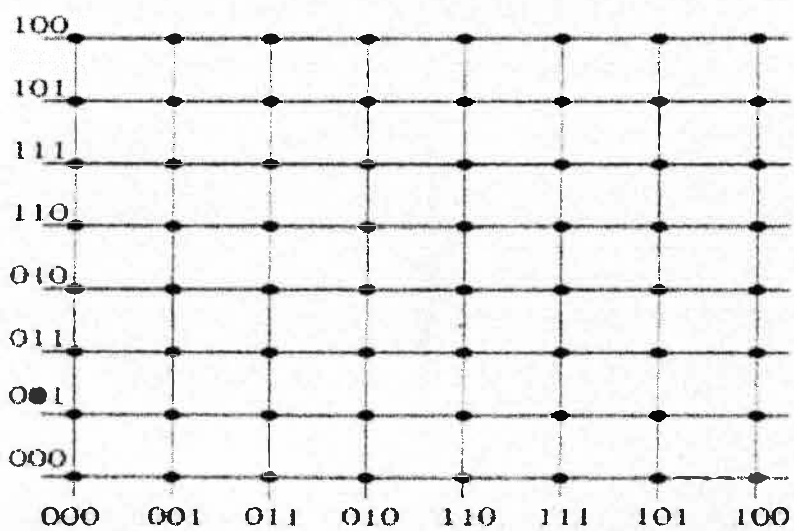

Figure 4. An 8X8 Lattice.

Lattice, torus and other networks such as ring, binary n-cubes and omega networks are topologically isomorphic[2] to a family of m-ary n-cube networks. The parameter n is the dimension of the cube and m is the radix or the number of nodes (multiplicity) along each dimension. The total number of nodes in the network is given by $N = m^n$

A node in the m-ary n-cube can be identified by an n-digit radix-m label $a_0a_1a_2 \ldots a_n$ where a_i represents the node's position in the i-th dimension. The nodes in the m-ary n-cube are also labeled in gray code sequence. Figure 5 shows the m-ary n-cube network with $m = 4$ and $n = 3$; hidden nodes and connections are not shown [10].

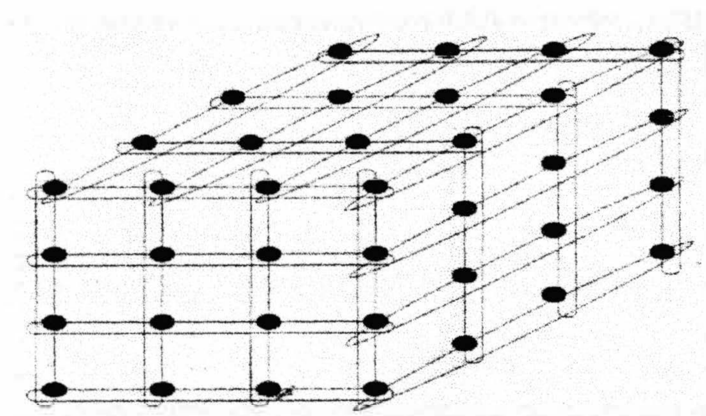

Figure 5. A m-ary n-cube With $m = 4$ and $n = 3$.

[2]two networks are topologically isomorphic if nodes adjacent in one network are adjacent in the other network

CHAPTER IV

DISCUSSION

Landscape is indeed a picturesque term. Literally, it refers to a picture representing some attribute. In *fitness landscapes*, the abstract solution space of COP's is visualized as landscape structures. These structures are fitness functions derived from a form

$$f : s \to \Re$$

where the function f maps a solution $s \in S$ (S is the space of all possible solutions) to a real number line called *fitness*. This measure signifies the quality of a solution. Higher the value of fitness, higher is the quality of that solution. It should be emphasized that an effective search operator for an optimization algorithm based on heuristics, requires some knowledge about the topology of these fitness landscapes. In this section a new 3-D tool for characterizing the structure of fitness landscapes is described. Using this tool, the effect of epistatic interactions on the ruggedness of fitness landscapes as shown by Kauffman's *NK* model is analyzed. The inadequacy of the *NK* model in modeling epistatic interactions for an important class of combinatorial optimization problems is discussed. Most notably is the limitation in modeling the epistatic relationships that exist in many real world problems.

For COP's with a single problem parameter, one can easily visualize the landscape as a two-dimensional plot. Consider a case where the objective of an optimization problem is to find the minimum value of an attribute, the landscape for which is shown in Figure 6. One dimension represents the single problem parameter and the other dimension represents the fitness or the attribute. For a landscape such as this, gradient search method works extremely well. However, gradient search fails for a landscape shown in Figure 7 due to the characteristics of this landscape. The search process can get trapped at a local optima and may never be able to find the global optimum. Clearly, the efficacy of the method used for searching the landscape depends on the characteristics of the landscape. There is no single search strategy that performs well for all kinds of problems. The best strategy is problem dependent---depending on how the landscape looks like for that problem. Hence, characterizing the topology of these landscapes is vital in determining effective search operators capable of finding an acceptable solution.

The fitness landscape for a two dimensional COP can be visualized as a 3-D plot---two dimensions for the problem parameters and one for the fitness. Figure 8 shows the landscape for such a problem.

The landscape for a COP with *N* problem Parameters may be visualized with an *N+1* dimensional landscape where *N* dimensions corresponds to the *N* problem parameters and the additional one dimension represents the fitness.

Real world problems generally deal with a large number of problem parameters. Visualizing landscapes of such high dimensions is certainly very difficult.

Though Weinberger's approach [19] helps to characterize the landscape in two dimensions, it cannot be applied to constrained COP's as described earlier in Chapter II.

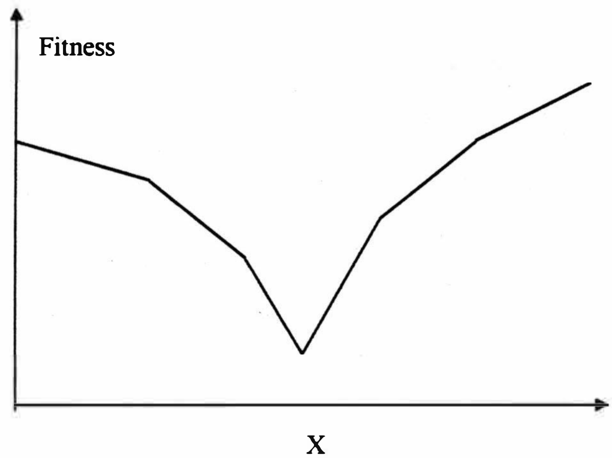

Figure 6. A 2-D Landscape - Gradient Search Works Well.

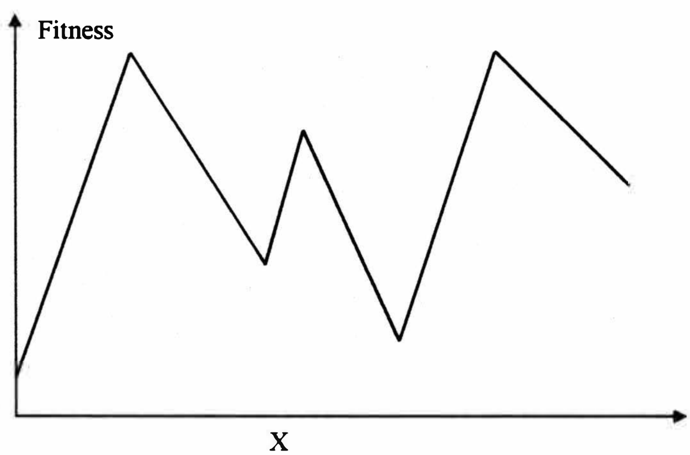

Figure 7. A 2-D Landscape - Gradient Search Does Not Work Well.

A new 3-D graphical tool has been developed to view high dimensional landscapes of COP's, in particular constrained COP's. Although this technique has the same shortcomings as the 2-D techniques---losing information in other dimensions, it provides more information than does a simple correlation and in addition allows one to visualize the ruggedness of the landscape. The basic concept relies on the ability to isomorphically embed a lattice into an *m*-ary *n*-cube.

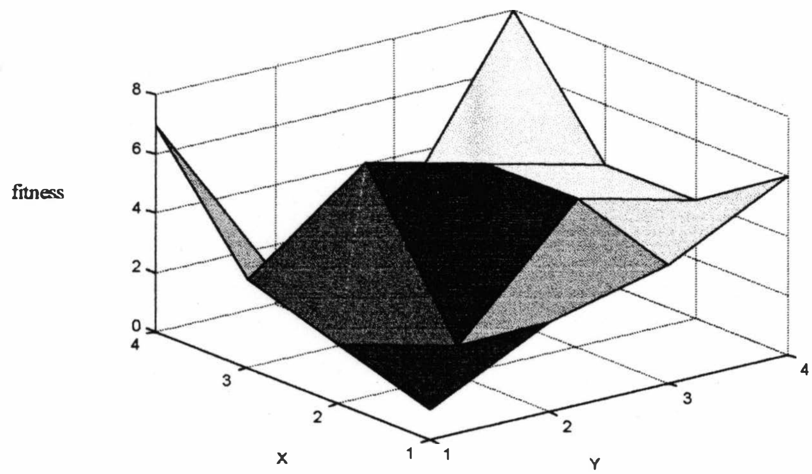

Figure 8. A 3-D Landscape for a COP With Two Problem Parameters.

First consider the entire solution space S for a COP represented on an *m*-ary *n*--cube. Each of the nodes represent a distinct solution s. Without loss of generality, assume that m is an integer power of two. Each node in the *m*-ary *n*-cube is labeled with a $n \lg m$ bit binary label [3]. The labeling is in gray code sequence such that any two nodes connected by an edge (see Figure 5) differ in only one bit position. In other words, the labels identify one mutant neighbors.

Let A and B be integer powers of two. Now, we can show that an A X B lattice can be isomorphically embedded into the *m*-ary *n*-cube in the following way. The

[3] lg denotes a base-2 logarithm

binary label associated with each node in the *m*-ary *n*-cube can be partitioned into two parts, which is written in the form

$$\underbrace{b_1 \ldots b_r}_{\lg A}\underbrace{b_{r+1} \ldots b_{n \lg m}}_{\lg B}$$

If the first lg A bits of the binary label represents the X coordinate and the least significant lg B bits represents the Y coordinate on the X-Y plane of the A X B lattice, then we have an arrangement were the resultant A X B lattice is isomorphically embedded in the *m*-ary *n*-cube.

In effect, this embedding has "unfolded" a high dimensional cube into a two-dimensional lattice. This unfolding process does break edges in the cube, retaining only a few of the *n* neighbors, and thus many neighbors in the cube are no longer neighbors in the lattice. This relates to the shortcoming mentioned earlier about this approach-losing information in higher dimensions. However, these old neighbors in the cube are distributed somewhere else in space and are not completely out of the picture. Now, the landscape consists of an X-Y plane (the lattice) distributed with all possible solutions. Adding the third dimension Z as fitness to this X-Y plane results in a 3-D fitness landscape, in which each point represents a solution and its corresponding fitness.

To assist in understanding the development of our graphical technique, it is advantageous to discuss it in the context of a simple albeit difficult graph theoretical problem---the frequency assignment problem.

Frequency Assignment Problem (FAP)

The radio spectrum is a vital but a limited natural resource and the demand for frequencies is outpacing the increase in the usable spectrum as technology changes. It is therefore vital that the spectrum be managed in the most effective way possible. This requires that the frequencies are assigned in an optimal or nearly optimal manner. Deterministic methods to solve problems of this kind works well only when the size is small. In fact these problems are NP-complete [7,17]. Algorithms based on heuristic techniques perform well for this class of problems. The performance of these algorithms can be improved, as stated before, by incorporating the knowledge of the topology of the fitness landscape, into its search operators. Hence this problem is chosen to illustrate our 3-D technique.

In general the FAP deals with the assignment of frequencies to a large number of transmission sites (N) operating in a region. The assignment of frequencies taken from a discrete set (F) is subject to a large number of constraints. There are several constraints that have to be considered for the FAP, of which cosite interference is one. This constraint states that, when any radio transmission site is located geographically within a distance Δd from another site, they cannot be assigned the same frequency as there will be interference. This FAP is equivalent to the generalized graph-coloring problem, which is known to be NP-complete [7,17].

More formally let N be the number of radio transmission sites in a given geographical area, where the distance between any two sites is known. Each site is to be assigned a single frequency (from a finite frequency set F) to transmit all messages. Interference between sites is possible, so it is necessary to place a constraint on the frequency assignments. Specifically, two sites within a distance Δd of each other cannot be assigned the same transmission frequency. The objective is to find the minimum cardinality subset of F such that all sites are assigned a transmission frequency, without violating the constraints.

Let s represent any solution in the search space S. Each $s \in S$ will be encoded as a bit string of length $k \cdot N$, where N equals the number of frequency sites and k bits represent one of the 2^k possible assignments.

Let η represent the number of different frequencies used in a solution s. Thus η is a measure of fitness for the solution s. A low value corresponds to a high fitness. Typically this problem requires $N+1$ dimensions, N dimensions to depict all possible solutions and one to indicate fitness. However this high dimensional landscape can be depicted on a 3-D landscape using the technique described in the previous paragraphs.

First we apply Kauffinan's *NK* model concepts to the problem. The '*N*' in the model corresponds to the number of sites and *K* refers to the constraint on the assignment of frequencies. For example, if every site is within Δd distance from any other site and this applies to all the sites then the epistatic interaction *K* for this case is one. This is shown in Figure 9. Thus the constraint in the FAP determines the epistatic interaction *K* among the sites.

Consider a specific instance where $N = 6$ is the number of sites and four frequencies can be used. Let the set of frequencies be $F = \{0, 1, 2, 3\}$ and the set of sites

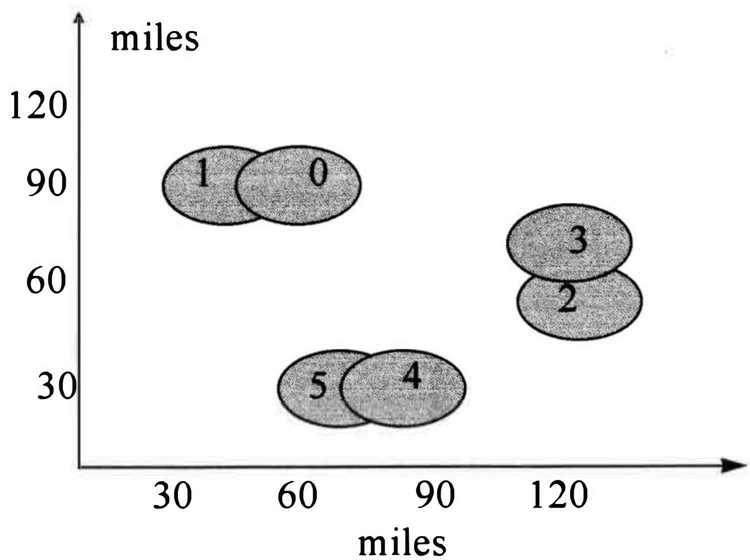

Figure 9. Geographical Distribution of Sites for the Case of $K = 1$.

be $C = \{C_0, C_1, C_2, C_3, C_4, C_5\}$. Now this problem can be encoded and represented on a 4-ary 6-cube. Each node requires a 12-bit label ($n \lg m$) and each site requires two bits ($\lg m$) to identify the frequencies assigned to it. This 4-ary 6 cube is unfolded into a $A X B$ lattice. Partitioning the 12 bit label into two equal parts of six bit labels each, the rows of the lattice can now be represented by the most significant six bits and the columns by the least significant six bits, resulting in a lattice of dimensions $A = B = 64$.

Let the rows in the lattice represent the X axis and the columns represent the Y-axis in a graph. In effect the lattice represents the X-Y plane for the landscape to be constructed. The axis numbering in the X and Y axes of the lattice is labeled in a natural order. However the numbering in a lattice are arranged as one mutant neighbors.

The difference between natural order and gray code order is shown here.

0,1,2,3,4,5,6,7 natural order

0,1,2,3,6,7,5,4 gray code order

The lattice does use the gray code sequence to generate the solution *s* at every node by using a function that converts the sequence of natural numbers on the X and Y axes to its gray code equivalent. An example will clarify this concept. Let *G (i)* be a function that converts a natural number *i* to its gray code equivalent (i.e. *G (5)* = 7). Then a point *(x, y)* in the X-Y plane will have coordinates (*G* (*x*), *G* (*y*)) in the lattice. Code that implements *G* (*i*) and its inverse is widely available (e.g. see page 896 [16]). Once the lattice coordinates are computed they can be expressed in binary form to describe the solution.

Consider a solution for this problem located at (29,50) in the X-Y plane of the lattice. The gray code equivalent coordinates are (19,43). Juxtaposing the binary equivalent of these converted *x* and *y* coordinates results in

$$\underbrace{010011}_{19}\ \underbrace{101011}_{43}$$

In order to identify the actual solution this 12 bit binary label is split into n = 6 segments of width lg m = 2 bits.

$$\underbrace{01}_{1}\ \underbrace{00}_{0}\ \underbrace{11}_{3}\ \underbrace{10}_{2}\ \underbrace{10}_{2}\ \underbrace{11}_{3}$$

In effect this partitioning gives the actual solution s that represents the frequency assignments to the six sites. The above solution '103223' indicates that site C_0 is assigned frequency '1' and site C_1 is assigned frequency '0', sites C_2 and C_5 are assigned frequency '3' and sites C_3 and C_4 are assigned frequency '2'.

If any two solutions in space are separated by a distance Δd, their frequency assignments must be different or a constraint will be violated. The parameter η has already been defined as the number of frequencies in a solution s. Then the fitness function of the solution s is defined as

$$fitness(s) = \begin{cases} -\eta & \text{no constraints violated} \\ \beta & \text{otherwise} \end{cases}$$

Without any constraint violations, the fitness values will range from a low of $-|F|$ to a high of -1. Conversely, the fitness is forced to a fixed constant $\beta < -\eta$ if one or more constraint violations exist---creating a deep sink hole in the landscape. The algorithm to generate the 3-D landscape is shown in Figure 10.

The above problem was tested for different values of epistatic interaction K. For the case of K = 0---no epistatic interactions---the six sites are distributed as shown

1. Define m and n in the m-ary n-cube.
2. Compute the number of binary bits required to represent each node and identify the length of the bit string s (solution)

$$\begin{aligned} b_n &= n \lg m \\ n &= \text{length of } s \end{aligned}$$

3. Determine number of bits required to represent the X axis - x_{bits} and Y axis - y_{bits}.

$$x_{bits} = \left\lceil \frac{b_n}{2} \right\rceil$$

$$y_{bits} = b_n - x_{bits}$$

4. Compute lattice dimensions as $x_{max} = 2^{x_{bits}}$ and $y_{max} = 2^{y_{bits}}$ (i.e. the maximum value of X and Y axis on the landscape)
5. Declare a structure - soln$[x_{max}][y_{max}]$- that will store fitness and the solution s it represents on the X-Y plane
6. Let $x_{index} = 0$
7. While $(x_{index} < x_{max})$
   {
   (a) Compute x_{gray} gray code equivalent of x_{index}
   (b) Let $y_{index} = 0$;
   (c) While$(y_{index} < y_{max})$
      i. Compute y_{gray} gray code equivalent of y_{index}
      ii. Compute s for the coordinate (x_{index}, y_{index})
         A. Convert x_{gray}, y_{gray} to equivalent binary code x_{bin}, y_{bin} respectively
         B. Juxtapose $x_{bin} :: y_{bin}$ to get the sol_{bin}
         C. Partition sol_{bin} into n, $\lg m$ bits and convert each partitioned set to equivalent decimal to identify the values of frequency.
         D. Store the result in the variable $soln[x_{index}][y_{index}]$
      iii. Compute fitness for the solution computed in the previous step and store in the variable $soln[x_{index}][y_{index}]$
      iv. $y_{index} = y_{index} + 1$;
   (d) $x_{index} = x_{index} + 1$;
   }
8. Plot the fitness versus solutions now arranged in the two-d array variable $soln$

Figure 10. Algorithm to Generate 3-D Landscape

in Figure 11. The circles around each site shows regions within which that site will interact with another site. Kauffman has shown that $K = 0$ results in a smooth landscape as there is no interaction. This is verified in the landscape shown in Figure 12 that has been generated using the new 3-D technique.

Consider the case where the sites are distributed as shown in Figure 13. (This figure is same as Figure 9 referred earlier. It is shown here again for convenience). The epistatic interactions for this case is $K = 1$. Every site is affected only by one other site (i.e. site 0 interacts with site 1 and so on). The landscape generated for this case shows (see Figure 14) the relative increase in ruggedness due to interaction, as shown earlier by Kauffman [11]. Kauffman's *NK* - model and the effect of epistatic interactions on fitness landscapes is easily visualized using the new 3-D technique for depicting higher dimensional landscapes.

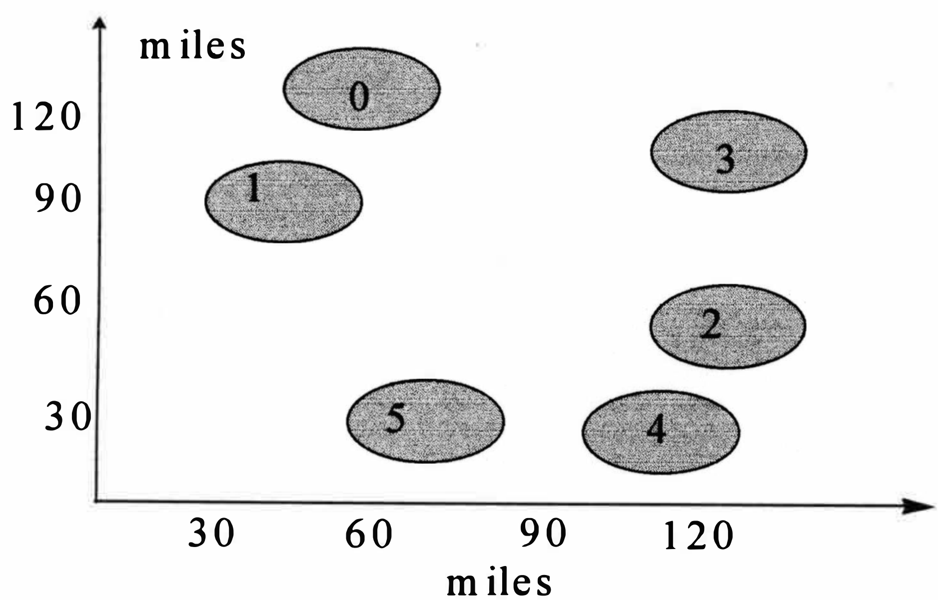

Figure 11. Geographical Distribution of Sites With $K = 0$.

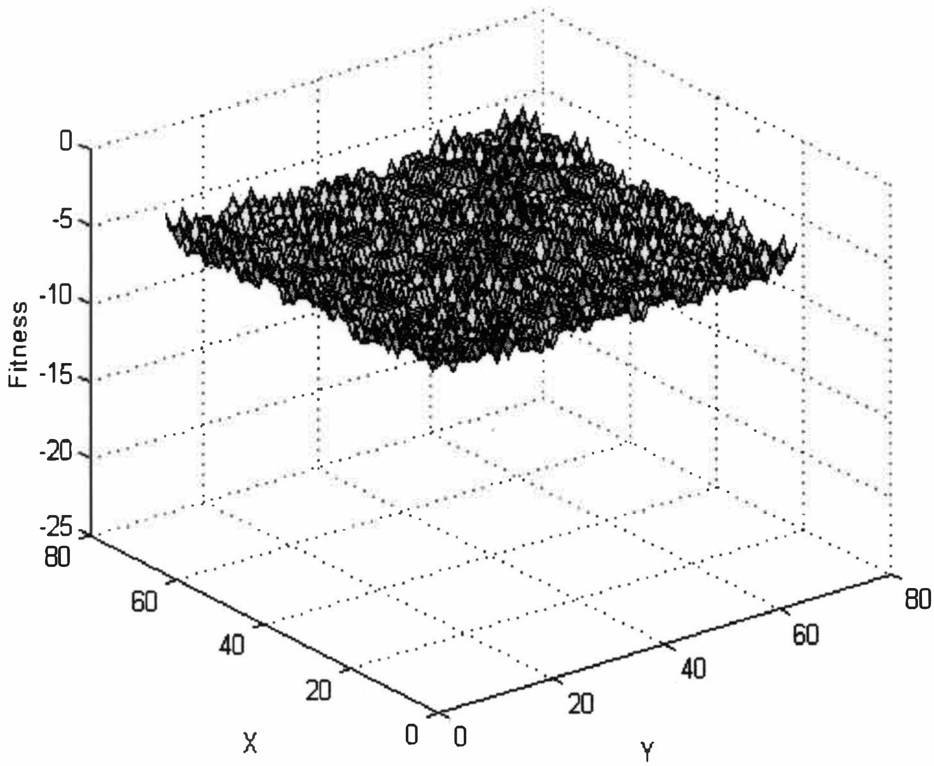

Figure 12. Fitness Landscape for $K = 0$.

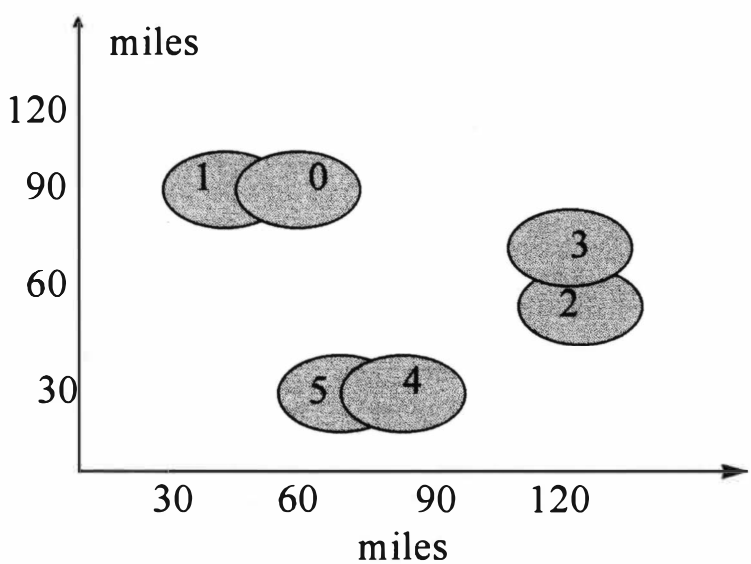

Figure 13. Geographical Distribution of Sites With $K = 1$.

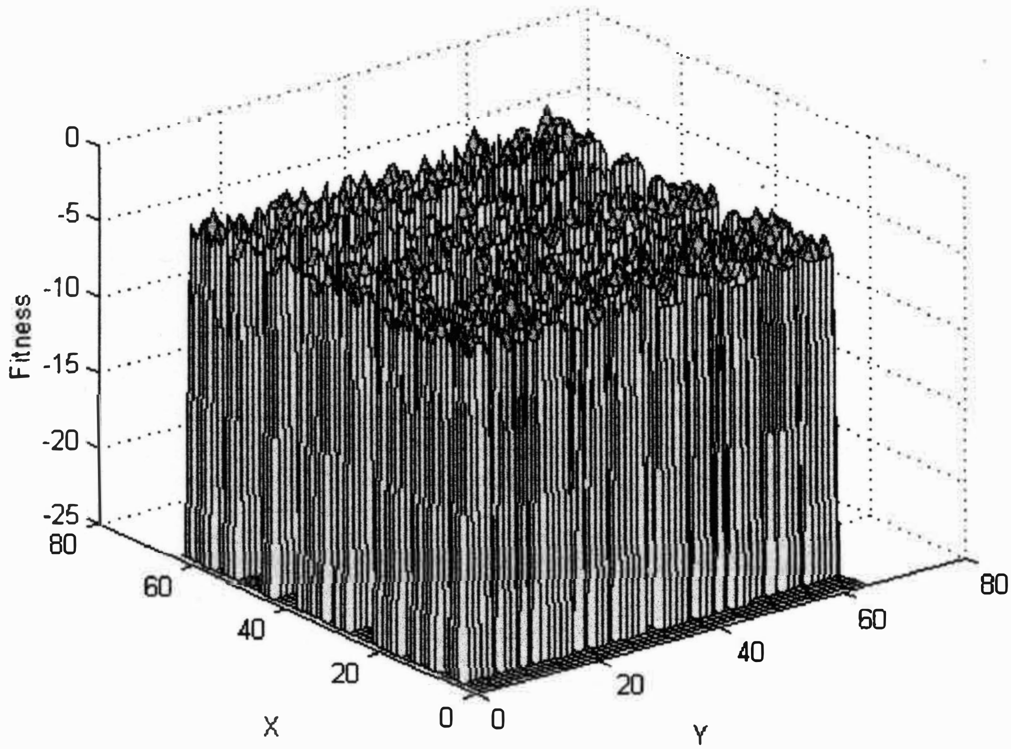

Figure 14. Fitness Landscape for $K = 1$.

The above test cases for $K = 0$ and $K = 1$ were conducted such that K was uniform for all sites. But, in practice an uniform epistatic interaction among the different sites is a rare occurrence. It may be possible for only two sites to interact and the rest may not. A typical example of such non-uniform interaction is shown in Figure 15. Kauffman assumed that K is uniform for all parts comprising the COP, which is not the case in most real world problems. The fitness landscape for this non-uniform interaction K is shown in Figure 16.

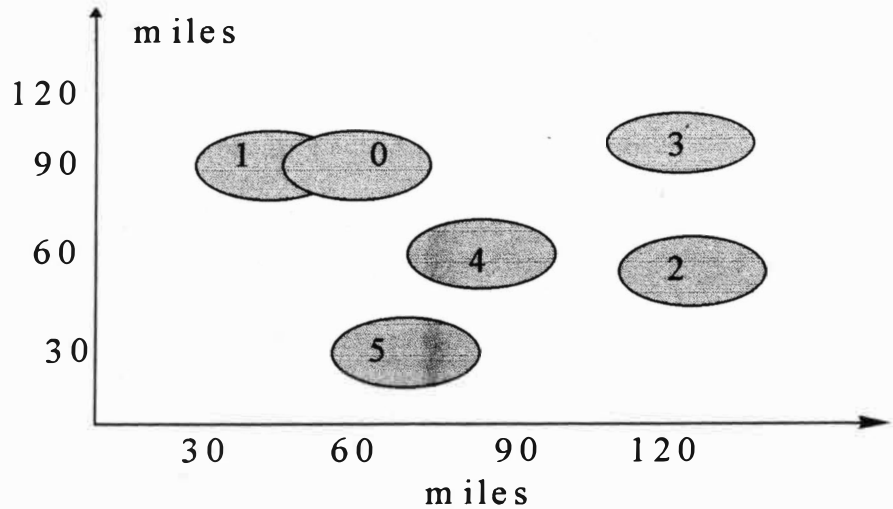

Figure 15. Geographical Distribution of Sites With Non-uniform K.

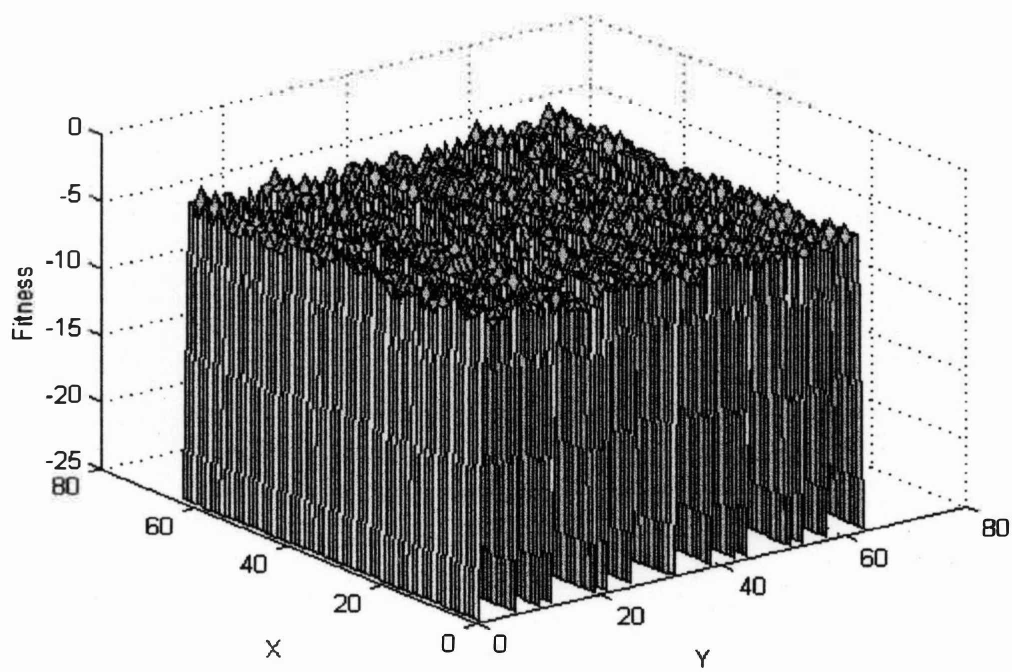

Figure 16. Fitness Landscape for Non–uniform K.

A comparison of the results for $K = 1$ and non-uniform K show that some of the sink holes in the landscape for $K = 1$ have disappeared for the non-uniform K landscape. Indeed, complete enumeration of the solutions found that 57.8 % of the uniform K landscape was composed of sink holes, whereas only 25% of the non-uniform K landscape was composed of sink holes, as the interaction level for this example of non-uniform K is less.

High Dimensional COP's

The FAP problem used to illustrate the 3-D technique in the previous section was small, with only 4096 solutions in its solution space S. Hence, it was possible to completely enumerate and analyze its fitness landscape. So, how does this technique work for high dimensional COP's with a very large size of S (typical of most COP's)? It is impossible to completely enumerate and study the characteristics of fitness landscapes for such high dimensional problems.

Nevertheless, the 3-D technique developed proves to be universal. In particular, the scalable property of this technique helps it to address high dimensional COP's. It can be applied to any kind of COP in any field. Many constrained COP's have multiple objects to be optimized while satisfying various constraints. This results in a highly complex COP. The 3-D technique can be used to ease the difficulty in optimizing such COP's by visualizing the landscape and providing valuable guidance to the heruistic technique used to optimize it.

The universal nature of this technique is illustrated in this section with reference to an instance of a hardware/software codesign problem. This problem is a typical example of a complex, constrained, high dimensional COP.

Hardware Software Codesign Problem (HSC)

The objective of hardware software codesign is to build computer systems that have an ideal balance of hardware and software components, which work together to satisfy a given specification. Hardware/software partitioning is a critical step towards achieving this balance. For years, designers have manually partitioned systems into hardware and software components. With this approach, designers make architectural decisions early in the design process and are often forced to revisit them. The result is a struggle to make a less than optimal architecture meet the design goals. Efforts have been made by several researchers and electronic design automation vendors to automate and optimize this process of partitioning. The success of any approach however, depends on the level of abstraction adopted for specifying the system to be designed. Most approaches deal with pure hardware or software specifications at the behavioral level. These approaches are more specific and are not flexible enough to handle tradeoffs between multiple attributes such as power consumption, cost etc.

Hu and D'Ambrosio [4] adopted a system level abstraction to evaluate various partitioning schemes. All system specifications are modeled as a set of functions, where each function has one or more performance constraints (e.g. feasibility). Functions can be implemented in either hardware or software. A system implementation is constructed by assigning functions to components selected from a hardware library (containing microprocessors, application specific integrated circuits (ASIC) etc.) in an optimal way. The decision of assigning the functions to hardware or software is made with respect to satisfying design constraints as well as some form of trade off between cost, power consumption and other attributes---an instance of a multiple objective optimization problem. Associated with every assignment is a real number that reflects the fitness or the quality of a solution. The space of all possible assignments and its associated fitness is the fitness landscape for this problem.

The space of all possible hardware-software partitions is very large and thus an exhaustive search is not a pragmatic approach. In fact the system design problem is shown to be NP- complete [2]. Hence some type of heuristic technique is required to optimize the problem. In general all heuristic search operators take current solutions and perturb the problem parameters in some stochastic manner to produce new solutions. A more effective approach is to construct an 'intelligent' search operator that is constrained to search in specific regions of the landscape. Incorporating intelligence into the operator expedites the search process and also finds a better solution. This can be illustrated as follows.

The fitness landscape for this problem can be viewed as a three dimensional plot by applying the 3-D technique. From this plot the user identifies regions on fitness landscape to avoid or regions to explore thoroughly. This information is provided to the search algorithm, which automatically designs search operators that are constrained to search (or avoid) regions identified by the user.

However the 3-D technique cannot portray the entire solution space for the problem, since it is a high dimensional multimodal entity with an exponential number of points. Nevertheless, the 3-D technique allows one to view the landscape with varying levels of granularity. A user can specify a quantization degree, which plots only a subset of points, resulting in a quantized landscape. For example the user can plot every 5000th point or 10000th point and all missing points assume the same value. For example if every 5000th point is chosen, points 5001,5002,…,9999 are assumed to have the same fitness as point 5000. This forms a piecewise continuous landscape that is quantized. This quantized landscape illustrates only the most prominent features. Nevertheless, often this will be a sufficient level of detail to identify regions of low fitness, which should be avoided during the search process. Furthermore, if the user is interested in investigating some regions in greater detail he/she can zoom into the selected region by simple specifying the (x, y) coordinates of that region and the degree of quantization (in other words ‘scale factor’). Thus, the technique is highly scalable.

The efficacy of this technique is further enhanced by following some guidelines that can keep the fitness of 1-mutant neighbors from being drastically different. For instance processors come in different versions, but the significant difference may be the clock speed or amount of on chip memory. One can try to keep processors with similar clock speeds as neighbors and then order them according to on chip memory. In other words, modules that have similar attributes should be placed as neighbors. In effect, this results in a smoother fitness landscape, which is much easier to search than a rugged landscape.

To illustrate the efficiency of this technique a specific instance of an automotive problem that deals with the design of an engine control module is discussed. This problem is taken from a research paper by Hu and D'Ambrosia [4] and consists of nine tasks whose properties are given in Table 1. Deadline and activation values are in μ sec and the number of instructions is based on a generic instruction set.

Table 2 lists the 16 possible hardware modules that are available for task assignments along with their power consumption. The hardware modules include microcontrollers (MC), general-purpose processors with no I/O (P), application specific integrated circuits (ASIC) and standard peripherals (PIO). Note that microcontrollers MC3 and MC4 have multiple distinct versions where the same core processor is used, but there are variations in the architecture (e.g., amount of on chip memory). The objective is to determine the tasks that are implemented by hardware or software and the modules to be included in the final system. Note that certain tasks may only be assigned to specific modules.

For simplicity consider the fitness landscape constructed for power consumption alone. The constraint for this attribute states all feasible solutions should not exceed a power consumption of 20 W. Lower power consumption implies higher fitness. If the power exceeds the power limit constraint the solutions are assigned a constant value of 0, regardless of the actual value.

Table 1

Specifications of Tasks for the Automotive Problem

Name	Deadline	Period	Activation	Instructions
Digitalfilter1 (DF1)	46.00	104.17	0.00	64
Digitalfilter2 (DF2)	10000.00	10000.00	9895.83	32
DecodeSPUB (DSB)	83.00	208.33	0.00	30
Read CAM (RC)	416.67	10000.00	0.00	30
ServiceRoutine (SR)	208.33	416.67	0.00	20
FuelCalc (FC)	1333.33	2500.00	833.33	480
SparkCalc (SC)	2500.00	2500.00	1666.67	100
ReadMap (RM)	312.50	416.67	0.00	40

Figure 17 depicts the fitness landscape based on a quantization factor of 5000 for both X and Y coordinates. Notice there are numerous regions that contain sink holes, (regions corresponding to a violation of constraint---power limit 20 W, with a fitness value of 0).

Table 2

Power Consumption of Modules Used

Module	Power Consumption in W
MC1-H	5.00
MC-2H	14.00
MC3a-H	1.30
MC3b-H	1.00
MC4a-H	0.30
MC4b-H	1.50
MC4c-H	1.50
p1-H	14.00
P2-H	13.00
P3-H	10.00
P4-H	5.00
ASIC1-H	1.65
ASIC2-H	1.20
ASIC3-H	1.80
PIO1-H	1.00
PIO2-H	1.30

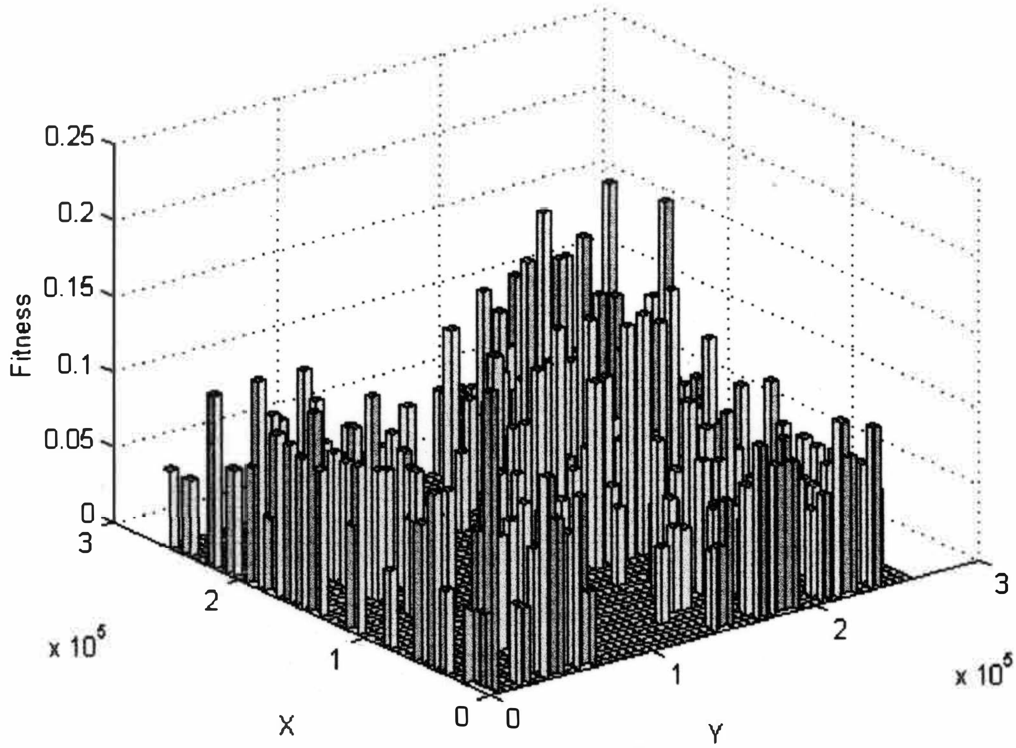

Figure 17. Fitness Landscape for the HSC Problem - With Low Level of Detail.

From this picture of the landscape, it is possible to identify the regions of sink holes as $0.5 \text{ X } 10^{5} < Y < 1.5 \text{ X } 10^{5}$ and the entire X-axis. Figure 18 shows the landscape zoomed into this region. One can use this information to constrain the search operators to avoid sink holes. If it in a sink hole the operator could discard that search point and generate another point that is not in the sink holes region, and proceed with the search. This method of constraining search operators is both simple and universal.

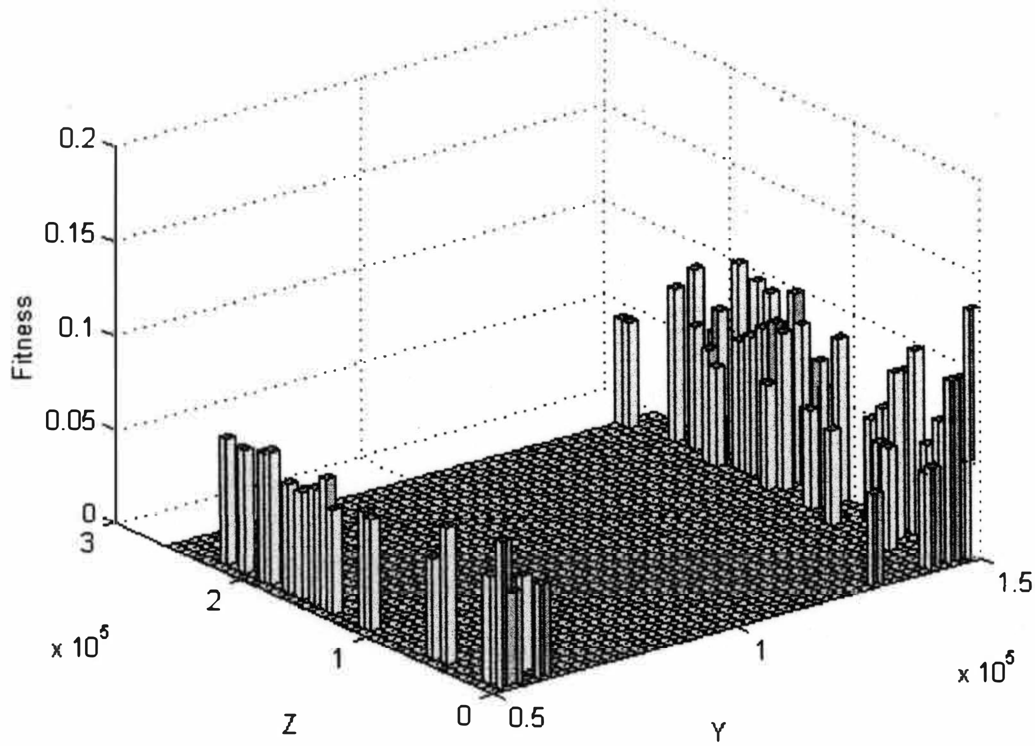

Figure 18. Zoomed in Fitness Landscape – Higher Level of Detail.

CHAPTER V

CONCLUSIONS AND FUTURE WORK

Analyzing the characteristics of multipeaked fitness landscapes is vital to the performance of search algorithms based on heuristics. The new 3-D graphical tool presented here maps high dimensional fitness landscapes to a three dimensional surface for easier visualization. By using an appropriate scale factor the 3-D fitness landscape can be readily generated and depicted. It is proposed to use this landscape information to constrain the search space for the COP. The user is able to identify regions on landscapes to explore or to avoid during a search process from this 3-D plot.

This approach is quite general. It is independent of the search mechanism used and is readily applicable to several different stochastic search techniques. (e.g., evolutionary and simulated annealing algorithms). Furthermore, for multiple objective optimization, one can generate more than one landscape which captures different constraints (e.g., in the HSC problem---optimizing power and cost). Eventually one needs to just specify forbidden regions in each landscape. The search process can quickly eliminate the solutions in those regions, and continue to search for better solutions in better regions of the landscape.

The 3-D technique developed here is highly versatile and can be applied to any COP. It has several advantages:

1. Simple and easy to use.

2. Provides visualization of the problem under consideration.

3. Gives information about whether or not a potential solution is in a particular region or not.

4. A user can identify regions of interest by reading off the coordinates from the X and Y axes of the landscape.

5. No restrictions on whether the problem is a constrained COP or not. In other words, it can characterize the landscape be it isotropic or anisotropic.

6. The method is scalable.

7. A valuable aid to define efficient search operators used in algorithms based on heuristics (e.g., Monte Carlo or evolutionary programs).

The relationship between ordering the different parts of a solution (N) and the epistatic interaction (K) and its influence on the structure of fitness landscapes needs to be investigated further. This indirectly defines the neighbor relations on fitness landscapes. Though this appears to be entirely problem dependent, with future investigations particularly for the case of non-uniform K, it is believed that one can arrive at some general guidelines. These guidelines should yield a landscape, wherein neighboring fitness values do not differ drastically. This would be a highly attractive feature for search algorithms as smoother landscapes are easier to search. However the 3-D technique will continue to act as a powerful tool for characterizing fitness landscapes of high dimensional COP's---constrained or not.

The two papers that resulted from this research work are

1. X. Hu, S. Ravichandran and G. Greenwood, "Modeling Epistatic Interactions in Fitness Landscapes", Computational Statistics and Data Analysis (submitted).

2. X. Hu, G. Greenwood, S. Ravichandran and G. Quan, "A Framework for User Assisted Design Space Exploration", Design Automation Conf., 1999. (submitted).

BIBLIOGRAPHY

[1] G. E. P. Box and G.M. Jenkins, Time series analysis, forecasting and control, San Francisco, CA: Holden Day, 1970.

[2] N. Chapman and J. Rozenbilt, The system design problem is NP-Complete, IEEE Conf. Sys. Man and Cyber., pages 1880-1884, 1994

[3] W. Crompton, S. Hurley and N. M Stephens, A parallel genetic algorithm for frequency assignment problems, IMACS International Symposium on Signal Processing Robotics and Neural Networks Conf. Proc., Pages 81-84, 94.

[4] J. G. D'Ambrosio and X. Hu, Configuration-level hardware/software partition for real time embedded systems, Proc. of the third Intl. Workshop on Hardware-Software Co-Design, pages 34-41, 1994.

[5] G. Greenwood and X. Hu. Are landscapes for constrained optimization problems statistically isotropic? Physica Scripta, 57:321-32, 1998.

[6] G. Greenwood and X. Hu, On the use of random walks to estimate correlation in fitness landscapes, Computational Statistics and Data Analysis, 98

[7] W. Hale, Frequency Assignment: Theory and Applications, Proc. of IEEE, 68, 1980

[8] W. Hordijk, A measure of landscapes, Evolutionary Computation, 4(4): 335-360, 1997

[9] X. Hu and G. Greenwood, Evolutionary approach to hardware/software partitioning, IEE Proc. Comput. Digit. Tech, 145:203-209, 1998.

[10] K. Hwang, Advanced Computer Architecture: Parallelism , Scalability, programmability, Mc.Graw Hill, 93.

[11] S. Kauffman, The Origins of Order, Oxford University Press NY, 1993.

[12] E. Lamagna, Infeasible Computation: NP-Complete Problems, Abacus, 4:18-33,1987

[13] E. L. Lawler, Combinatorial optimization: Networks and Matroid, Holt, Rinehart and Winston, 76.

[14] B. Manderick and P. Spiessons, How to select genetic operators for combinatorial optimization by analyzing their fitness landscapes, Computational Intelligence: Imitating life, edited by J. Zurada, R. Marks and C. Robinson, pages 170-180, 1994.

[15] Z. Michaelwicz and M. Schoenauer, Evolutionary algorithms for constrained optimization problems, Evolutionary Comp., 4:1-32, 1996.

[16] W. Press, S. Teukolsky, W. Vetterling and B. Flannery, Numerical Recipes in C, Cambridge University Press, 2nd edition, 1992

[17] D. H. Smith, S. Hurley and S. U. Thiel, Frequency assignment algorithms, Report ref: rccm070 edition, 1996.

[18] V.K. Vassilev, An information measure of landscapes, Proc. of seventh Intl. Conf. on Genetic Algorithms, MSU, pages 49-55, 1997.

[19] E. Weinberger, Correlated and uncorrelated fitness landscapes and how to tell the difference, Biological Cybernetics, 63:325-336, 1997.

[20] S. Wright, The role of mutation, inbreeding, crossbreeding and selection in evolution, Proc. of 6th Conf. on Genetics, 1:356,32.