Functions as Permutations: Implications for No Free Lunch, Walsh Analysis and Statistics

D. Whitley

Computer Science Department
Colorado State University
Fort Collins, CO 80523
whitley@cs.colostate.edu

Abstract

Permutations can represent search problems when all points in the search space have unique evaluations. Given a particular set of N evaluations we have N! search algorithms and N! possible functions. A general No Free Lunch result holds for this finite set of N! functions. Furthermore, it is proven that the average description length over the set of N! functions must be O(N lg N). Thus if the size of the search space is exponentially large with respect to a parameter set which specifies a point in the search space, then the description length of the set of N! functions must also be exponential on average. Summary statistics are identical for all instances of the set of N! functions, including mean, variance, skew and other r-moment statistics. These summary statistics can be used to show that any set of N! functions must obey a set of identical constraints which holds over the set of Walsh coefficients. This also imposes mild constraints on schema information for the set of N! functions. When $N = 2^L$ subsets of the N! functions are related via Gray codes which partition N! into equivalence classes of size 2^L.

1 Functions as Permutations

Consider a set composed of N unique values; these N values can be mapped to a set V of evaluations for some objective function. Let X be the set of points and V the set of evaluations; we then define a one-to-one objective function f:

$$f(x) = v, \qquad v \in V, x \in X.$$

Construct a set Π of all permutation over the values in V. In this case, the set Π represents all objective functions which can be constructed over V. We will also use Π to represent all search algorithms which can be applied to the set of evaluation functions which can be constructed over V. We will represent an algorithm by the order in which it samples the values in V. If different search procedures enumerate the points in V in the same order, then they form part of an equivalence class. Resampling of points is ignored. Hence, in this context, algorithms as well as functions are permutations. It is easy to show that a specialization of the "No Free Lunch" theorem result holds [14] [9] [2]. *On average, no algorithm is better than random enumeration in locating the global optimum. If algorithms are executed for only* **m** *steps, every algorithm find the same set of best so-far solutions over all functions.*

Focusing attention on a well defined set of permutations of finite length, allows one to make more detailed comments about the No Free Lunch result as it pertains to this set.

Observation: The set of N! functions corresponding to Π have a description length of O(N lg N) bits on average, where N is the number of points in the search space.

Proof: The proof follows the well known proof demonstrating that the best sorting algorithms have complexity O(N lg N). First, since we have N! functions, we would like to "tag" each function with a bit string which uniquely identifies that function. We then make each of these tags a leaf in a binary tree. The "tag" acts as an address which tells us to go left or right at each point in the tree in order to reach a leaf node corresponding to that function. But the "tag" also uniquely identifies the function. The tree is constructed in a balanced fashion so that the height of the tree corresponds to the number of bits needed to tag each function. Since there are N! leaves in the tree, the height of the tree must be O(lg(N!)) = O(N lg N). Thus O(N lg N) bits are required to uniquely label each function. **QED**.

Note that the number of bits required to construct a full enumeration of any permutation of N elements is also O(N lg N) bits, since there are N elements and lg N bits are needed to distinguish each element. Thus, most of these functions have exponential description. To be NP-Complete, the description length must be polynomial. This means that an NP-Complete problem class *cannot* be used to generate all N! functions. This includes NK-Landscapes [8] and MAXSAT, for example.

This is, of course, one of the major concerns about No Free Lunch results. Do "No Free Lunch" results really apply to sets of functions which are of practical interest? Yet this same concern is often overlooked when theoretical researchers wish to make mathematical observations about search. For example, proofs relating the number of expected optima over all possible functions [11], or the expected path length to a local optimum over all possible functions [13] under local search are computed with respect to the set of N! functions.

1.1 Walsh Analysis

It is next shown that the set of Walsh coefficients are constrained with respect to the set of all possible members of the set of N! functions. From this it follows that summary statistics such as variance, skew, kurtosis are not useful for guiding search.

Every real-valued function f over an L-bit string can be expressed as a weighted sum of a set of 2^L orthogonal functions called **Walsh functions**.

$$f(x) = \sum_{j=0}^{2^L-1} w_j \psi_j(x)$$

where the Walsh functions are denoted $\psi_j : \mathcal{B}^L \rightarrow \{-1, 1\}$. The weights $w_j \in \mathcal{R}$ are called **Walsh coefficients**.

The indices of both Walsh functions and coefficients may be expressed as binary strings or the equivalent integer. Treating the indices x, j as binary vectors, we can compute the Walsh function as follows:

$$\psi_j(x) = (-1)^{j^\top x}.$$

The 2^L Walsh coefficients can likewise be computed by a Walsh transform:

$$w_j = \frac{1}{2^L} \sum_{i=0}^{2^L-1} f(i)\psi_j(i)$$

The calculation of Walsh coefficients can be thought of in terms of matrix multiplication. Let $\vec{f}$ be a column vector of 2^L elements where the i^{th} element is the evaluation of function $f(i)$. Similarly define a column vector $\vec{w}$ for the Walsh coefficients. If M is a $2^L \times 2^L$ matrix where $M_{i,j} = \psi_j(i)$, then:

$$\vec{w} = \frac{1}{2^L} \vec{f}^\top M$$

For example, if we have a 3 bit function with the 2^3 function evaluations labeled $f_0..f_7$, then the Walsh coefficient calculation would be:

$$\vec{w} = \frac{1}{8} \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \end{bmatrix}^\top \begin{bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\ 1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\ 1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\ 1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \end{bmatrix} \quad (1)$$

An important property of Walsh coefficients is that w_j measures the contribution to the evaluation function by the interaction of the bits indicated by the positions of the 1's in j. We number the bit positions from right to left, starting at 0. Thus, w_{0001} measures the linear contribution to the evaluation function associated with bit position 0, while w_{0101} measures the nonlinear interaction between the bits in positions 0 and 2. This nonlinearity is often considered to be an important feature in determining problem difficulty for stochastic search algorithms [3, 4, 12].

2 Summary Statistics for Problem Instances

Walsh analysis can be used to compute summary statistics for **fitness distributions** of discrete optimization problems. Note that the fitness distribution is the distribution formed by evaluating all possible inputs to a problem. So, for a problem defined over 2^L possible inputs, the distribution would be comprised of all 2^L evaluations of the inputs. Goldberg and Rudnick [5] have used Walsh coefficients to calculate fitness variance for fitness distributions and schemata. Heckendorn, Rana and Whitley [6] show how higher order statistics such as skew and kurtosis can be also be computed from the Walsh coefficients using a general formula for computing the r^{th} moment for any fitness distribution; they also show that for certain special types of functions, the Walsh coefficients and summary statistics can be computed in polynomial time.

Since $\psi_0(x) = 1$ for all inputs, the w_0 coefficient is the mean of all fitnesses. Given the mean, the formula used to compute the r^{th} moment for a discrete random variable X is:

$$\mu_r = E[(X-\mu)^r] = \sum_{x \in X} (x-\mu)^r p(x)$$

For our purposes, the function $p(x) = \frac{1}{2^L}$ since we are enumerating a function over binary strings and each point occurs $\frac{1}{2^L}$ times. The function then becomes:

$$\mu_r = \sum_{x \in X} \frac{(x-\mu)^r}{2^L}$$

The r^{th} moment over the distribution of fitness for all 2^L possible input strings is:

$$\mu_r = \frac{1}{2^L} \sum_{x=0}^{2^L-1} (f(x) - \mu)^r$$

Since $f(x) = \sum_{i=0}^{2^L-1} w_i \psi_i(x)$,

$$\mu_r = \frac{1}{2^L} \sum_{x=0}^{2^L-1} \left(\sum_{i=0}^{2^L-1} w_i \psi_i(x) - \mu \right)^r$$

Since $\mu = w_0$, and $\psi_0(x) = 1 \; \forall x$:

$$\mu_r = \frac{1}{2^L} \sum_{x=0}^{2^L-1} \left(\sum_{i=1}^{2^L-1} w_i \psi_i(x) \right)^r$$

Now create a set of r indices a_j for $1 \le j \le r$, and expand the formula:

$$\mu_r = \frac{1}{2^L} \sum_{x=0}^{2^L-1} \left(\sum_{a_1=1}^{2^L-1} w_{a_1} \psi_{a_1}(x) \right) \left(\sum_{a_2=1}^{2^L-1} w_{a_2} \psi_{a_2}(x) \right) \ldots \left(\sum_{a_r=1}^{2^L-1} w_{a_r} \psi_{a_r}(x) \right)$$

By simple algebra these equations simplify to the following:

$$\mu_r = \sum_{a_1 \oplus a_2 \oplus \ldots \oplus a_r = 0} w_{a_1} w_{a_2} \ldots w_{a_r}, \qquad a_i \neq 0 \tag{2}$$

To summarize this formula, given the set of nonzero Walsh coefficients, we can compute the r^{th} moment for the fitness distribution using products of the Walsh coefficients such that the exclusive-or of the indices is zero.

This formula allows us to compute the variance, skew and kurtosis for any fitness distribution provided we are given the Walsh coefficients.

$$variance = \mu_2 = \sigma^2 \qquad skew = \frac{\mu_3}{\sigma^3} \qquad kurtosis = \frac{\mu_4}{\sigma^4}$$

For example, since $a_1 \oplus a_2 = 0$ if and only if $a_1 = a_2$ the variance for any function is given by

$$\sum_{i=1}^{2^L-1} w_i w_i$$

Heckendorn, Rana and Whitley [6] show that in the special case of *Embedded Landscapes* [7] there are only a polynomial number of nonzero Walsh coefficients and we can identify and compute them in polynomial time. MAX3SAT and NK-Landscapes are instances of an Embedded Landscape. Thus, an exact Walsh analysis can be done in polynomial time. And as a result all the summary statistics can be computed exactly from equation 2 for all of these classes of problems in polynomial time.

It at first seems surprising that one can characterize NP-Complete problems in such statistical detail. Since we can exactly compute these summary statistics about NP-Complete problems in polynomial time, is there any useful information which can be extracted from these summary statistics to guide search? We have previously noted [6] that if $P \neq NP$, then in the general case the r^{th} moment summary statistics cannot provide any information which can be used to guide any search algorithm to an optimal solution, or even to an above average solution in the search space. The current paper decouples the usefulness of these summary statistics from the question of P vs NP.

2.1 On Summary Statistics and Search

We again consider the set Π of all permutations over the set of values V.

Observation: The set of summary statistics, such as variance, skew and other higher moment summary statistics, provide no useful information for selecting one search algorithm over another, or for guiding any particular search algorithm, when searching a specific objective function constructed from V.

Proof: The set of summary statistics are identical for the set V. Since all permutations, Π, are constructions over V, and these permutations describe all possible functions over V, No Free Lunch holds over Π. This implies that summary statistics cannot be used to guide search, since the summary statistics are identical over a set for which a No Free Lunch result holds. **QED**.

This also places interesting restrictions on the information which can be extracted from Walsh coefficients about objective functions constructed from V, since the values of the Walsh coefficients themselves are directly constrained by the summary statistics.

2.2 Statistics, Neighborhood and Walsh Analysis

We next look at the relationship between summary statistics, neighborhood structure and Walsh analysis.

A Gray code is a wrapped construction which forms a circuit such that all adjacent integers are Hamming Distance 1 neighbors and endpoints are also neighbors. This circuit can be "shifted" such that any value can be assigned to the string of all zero bits; the result is still a Gray code. Normally, when bit strings are "DeGrayed" they are converted to binary strings, then converted to integers, and finally mapped to real space. "Shifting" [10] can be applied after the bits have been converted to integers, but before the integers are mapped to real values. Any integer constant between 0 and $2^L - 1$ $(mod\ 2^L)$ can be added to the decoded integer generated from the bit string. We may view "shifting" as a change in representation for a single function. But it also changes the permutation in Π and thus also results in a new function. Thus, when $N = 2^L$ then 2^L of the N! functions form an equivalence class; one function can be transformed into another via a change in representation using a shifted Gray Code: this equivalence class corresponds to different Gray codes of the same real valued function. When the real-valued space is modeled as a 1-dimensional circuit, the real valued space is unchanged under this change in representation.

We have recently shown that repeating neighborhood structures exists under shifting. Since a reflected Gray code is a symmetric reflection, flipping the leading bit of each string does not change the Hamming distance-1 neighborhood. This corresponds to a shifts by 2^{L-1}. This means that the neighborhood structure repeats.

Theorem: In any reflected Gray code representation of a 1-dimensional function, or parameter of a multidimensional function, there are only 2^{L-2} unique shifts.

For a proof see Barbulescu, Watson and Whitley [1]. The neighborhood structure repeats every quartile of the parameter space under shifting. For example, in a 4-bit neighborhood defined over 16 points, there are only 4 unique Hamming distance-1 neighborhoods.

Let $\pi \in \Pi$ be that permutation where the values from V are in sorted order from small to large. To compute the Walsh coefficients for any other function in the set Π, we can apply a reordering permutation to either π to generate the new function/permutation, or we can apply the same reordering permutation to the rows of the Walsh Transform matrix. Note that under matrix multiplication, there is no difference between permuting the function/permutation vector and permuting the rows of the matrix. Let π' be the permuted function (in column vector form) and M' the corresponding matrix where the permutation π' is used to reorder the rows of M.

$$\textbf{Observation:} \quad \vec{w} = \left[\frac{1}{2^L}(\vec{\pi'})^\mathsf{T} M\right] = \left[\frac{1}{2^L}\vec{\pi}^\mathsf{T} M'\right]$$

We can also view the Walsh Matrix is as a recursively defined structure. A Walsh Transform of dimension 2^d can be expressed as a composition of four 2^{d-1} dimensional transforms. The following notation denotes a $2^(d-i)$ transform as W_i. In addition, because we wish to manipulate the rows of the Walsh matrix, the four "rows" of the recursively decompose Walsh matrix can be expressed as M x N matrices; these are labeled using $W_{2A}, W_{2B}, W_{2C}, W_{2D}$ as shown:

$$W = \begin{bmatrix} W_1 & W_1 \\ W_1 & -W_1 \end{bmatrix} = \begin{bmatrix} W_2 & W_2 & W_2 & W_2 \\ W_2 & -W_2 & W_2 & -W_2 \\ W_2 & W_2 & -W_2 & -W_2 \\ W_2 & -W_2 & -W_2 & W_2 \end{bmatrix} = \begin{bmatrix} W_{2A} \\ W_{2B} \\ W_{2C} \\ W_{2D} \end{bmatrix}$$

Now, we can shift the rows of the W_2 form of the Walsh Matrix, and this exactly corresponds to shifting the input function by 2^{L-2}. However this shift is in Binary space. The shift in Gray space leaves the Hamming neighborhood structure unchanged. Under Binary, a shift of 2^{L-1} leaves the neighborhood unchanged, but an examination of any small space (e.g. 2 to 10 bits) shows that a shift of 2^{L-2} changes the neighborhood under Binary. But there are similarities between the two representations. Study of the Gray shifts shows that the decomposed Walsh matrix is reversed at various points under shifting. This is consistent with the fact that the Gray code is a reflected code. This motivates the following observation.

Let $\equiv_n$ denote a binary relation between Walsh Matrices, such that the resulting set of Walsh coefficients corresponds to functions with an identical neighborhood structure in Hamming space. The decomposed Walsh matrix can be shifted by 2^{L-1} and the decomposed Walsh matrix can be reversed without changing the neighborhood structure. Thus,

$$\begin{bmatrix} W_{2A} \\ W_{2B} \\ W_{2C} \\ W_{2D} \end{bmatrix} \equiv_n \begin{bmatrix} W_{2C} \\ W_{2D} \\ W_{2A} \\ W_{2B} \end{bmatrix} \equiv_n \begin{bmatrix} W_{2B} \\ W_{2A} \\ W_{2D} \\ W_{2C} \end{bmatrix} \equiv_n \begin{bmatrix} W_{2D} \\ W_{2C} \\ W_{2B} \\ W_{2A} \end{bmatrix} \tag{3}$$

These reordering can be done by applying exclusive-or between the encoding an a target string and by reordering the first two bits. Neither of these operators have any impact on the neighborhood structure. There are of course other reorderings of the Walsh matrix that leave the neighborhood structure unchanged, including some shifts of the Gray representation.

2.3 Empirical Examples from Walsh Analysis

The point of the last section is that there are several different changes in representation that do not change the structure of the Hamming neighborhood. Does

Gray	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}
0	8.5	0	0	0	0	0	0	0	-4	0	0	0	-2	0	-1	-.5
4	8.5	0	0	0	-4	0	0	0	0	0	0	0	2	0	-1	-.5
8	8.5	0	0	0	0	0	0	0	4	0	0	0	-2	0	-1	-.5
12	8.5	0	0	0	4	0	0	0	0	0	0	0	2	0	-1	-.5
1	8.5	-1	-1	-1	-1	-1	-1	-1	-3	1	1	1	-1	1	0	.5
5	8.5	-1	1	1	-3	1	-1	-1	1	1	-1	-1	1	-1	0	.5
9	8.5	-1	-1	-1	1	1	1	1	3	-1	-1	-1	-1	1	0	.5
13	8.5	-1	1	1	3	-1	1	1	-1	-1	1	1	1	-1	0	.5
2	8.5	0	-2	0	-2	0	-2	0	-2	0	2	0	0	0	1	-.5
6	8.5	0	2	0	-2	0	-2	0	2	0	-2	0	0	0	1	-.5
10	8.5	0	-2	0	2	0	2	0	2	0	-2	0	0	0	1	-.5
14	8.5	0	2	0	2	0	2	0	-2	0	2	0	0	0	1	-.5
3	8.5	1	-1	-1	-3	1	-1	-1	-1	-1	1	1	1	-1	0	.5
7	8.5	1	1	1	-1	-1	-1	-1	3	-1	-1	-1	-1	1	0	.5
11	8.5	1	-1	-1	3	-1	1	1	1	1	-1	-1	1	-1	0	.5
15	8.5	1	1	1	1	1	1	1	3	1	1	1	-1	1	0	.5

Binary	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}
0	8.5	-.5	-1	0	-2	0	0	0	-4	0	0	0	0	0	0	0
8	8.5	.5	0	1	-1	1	1	1	3	-1	-1	-1	-1	-1	-1	-1
0^R	8.5	.5	1	0	2	0	0	0	4	0	0	0	0	0	0	0
8^R	8.5	-.5	0	1	1	1	1	-1	-3	-1	-1	1	-1	1	1	-1

Table 1: *Walsh coefficients for the 16 shifts of the permutation [1 .. 16].*

this preservation of neighborhood show up under Walsh analysis? And what relationship do these transformations have to one another? This question is only partly answered, but empirical data derived from Walsh analysis suggests there are very strong similarities in these neighborhoods.

Let $\pi = (1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16)$ be the sorted permutation composed of the integers from 1 to 16. (This permutation is the Binary ordering. The permutation has be mapped to Gray-coded strings, then those strings must be reordered according to their Binary interpretation for the Walsh analysis to be correct.) Table 1 gives the Walsh coefficients for all possible shifts of this function under Gray code. There are really only 4 different Hamming neighborhood structures over the 16 possible shifts. Shifts at i+0, i+4, i+8, i+12 are identical for i = 1 to 4. There are strong similarities between the Walsh coefficients for these identical neighborhoods. Table 1 also shows the Walsh coefficients for 4 equivalent neighborhoods under Binary corresponding to equation 3.

Table 2 gives four randomly selected permutations and the corresponding Walsh coefficients. The Walsh coefficients are now more diverse. Nevertheless, the summary statistics always remain the same for all possible permutations. In this example, a simple calculation shows that the variance is 21.25 for all func-

Permutations A, B, C, D

A	5	14	4	9	1	12	2	11	8	16	6	13	3	15	7	10
B	11	5	1	4	13	2	15	9	6	3	14	7	10	8	12	16
C	7	12	1	3	14	9	15	10	2	16	5	8	13	4	11	6
D	10	9	13	6	11	3	12	1	14	2	5	8	16	4	15	7

Walsh Coefficients for Permutations A, B, C, D

ID	w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}	w_{14}	w_{15}
A	8.5	-4	0.75	-1.0	.875	.375	.625	.375	-1.25	-.25	0	.25	-.125	.375	.125	-.625
B	8.5	1.75	-1.25	1.0	-2.125	-.125	1.125	-.375	-1.0	0.75	1.5	0.75	-.125	-1.625	1.375	.875
C	8.5	0	1.125	-.625	-1.75	-3.0	1.375	-1.125	.375	.375	.5	.25	-1.375	.875	0.75	0.75
D	8.5	3.5	.125	.625	-.125	-1.375	.25	0.5	-.375	-.125	0	-1.75	1.5	0	-.375	-.875

Table 2: *Walsh coefficients for random permutations of the elements [1 .. 16].*

tions/permutations. Again, this is because the summary statistics are computed with respect to V and not with respect to which member of the permutation set Π is involved.

3 Discussion and Conclusions

In as much as Walsh analysis can be used to compute information about schema fitness information, constraints on what we can learn about a function from Walsh analysis has some impact on what we can or cannot learn about a function from schema fitness information.

Often Walsh analysis is dismissed as not being of practical interest. However, Heckendorn, Rana and Whitley [7] have shown that it is possible to do Walsh analysis in polynomial time for all functions which can be modeled as Embedded landscapes. Embedded landscapes include not only NP-Complete functions such as MAX3SAT and NK-Landscapes, but also common separable test functions such as Rastrigin's functions. Any function that can be decomposed into a polynomial number of nonlinear interactions is an Embedded landscape. Heckendorn, Rana and Whitley [6] also show that for all Embedded landscapes it is possible to compute summary statistics in polynomial time.

This paper specifically looks at functions as permutations. It is observed that the No Free Lunch results hold for this finite set, that the average description length for this finite set of problems must be exponential on average, and that the summary statistics must be identical for all members of this set. This places constraints on the set of values which can be possible Walsh coefficients. More concretely, it is shown that summary statistics cannot be used to guide search, since No Free Lunch results holds over a set for which the summary statistics are identical. Also different representations of the same function can have identical neighborhood structure; this also appears to be reflected in the similarity of the Walsh coefficients. But more work is needed to fully understand this relationship.

ACKNOWLEDGEMENT: The observation that summary statistics are identical under shifting of Gray codes was made by Soraya Rana.

References

[1] L. Barbulescu, J.P. Watson, and D. Whitley. Dynamic representations and escaping local optima. In *17th National Conf. on Artificial Intelligence (AAAI)*, 2000.

[2] J. Culberson. On the Futility of Blind Search. *Evolutionary Computation*, 6(2):109–127, 1999.

[3] D. Goldberg. Genetic Algorithms and Walsh Functions: Part I, A Gentle Introduction. *Complex Systems*, 3:129–152, 1989.

[4] D. Goldberg. Genetic Algorithms and Walsh Functions: Part II, Deception and its Analysis. *Complex Systems*, 3:153–171, 1989.

[5] D. Goldberg and M. Rudnick. Genetic algorithms and the variance of fitness. *Complex Systems*, 5(3):265–278, 1991.

[6] R. Heckendorn, S. Rana, and D. Whitley. Polynomial time summary statistics for a generalization of MAXSAT. *GECCO-99*, Morgan Kaufmann, 1999.

[7] R. Heckendorn, S. Rana, and D. Whitley. Test Function Generators as Embedded Landscapes. In *Foundations of Genetic Algorithms FOGA-5*. Morgan Kaufmann, 1999.

[8] S.A. Kauffman. Adaptation on Rugged Fitness Landscapes. *Lectures in the Science of Complexity*, pages 527–618. Addison-Wesley, 1989.

[9] N.J. Radcliffe and P.D. Surry. Fundamental limitations on search algorithms: Evolutionary computing in perspective. *Lecture Notes in Computer Science 1000*. Springer-Verlag, 1995.

[10] S. Rana and D. Whitley. Bit Representations with a Twist. *7th International Conf. on GAs (ICGA)*, Morgan Kaufmann, 1997.

[11] S. Rana and D. Whitley. Search, representation and counting optima. *IMA Workshop on Evolutionary Algorithms*. Springer-Verlag, 1998.

[12] C. Reeves and C. Wright. An Experimental Design Perspective on Genetic Algorithms. In *FOGA - 3*, pages 7–22. Morgan Kaufmann, 1995.

[13] C.A. Tovey. Hill climbing and multiple local optima. *SIAM Journal on Algebraic and Discrete Methods*, 6(3):384–393, July 1985.

[14] D.H. Wolpert and W.G. Macready. No free lunch theorems for optimization. *IEEE Transactions on Evolutionary Computation*, 4:67–82, 1997.