A Free Lunch Proof for Gray versus Binary Encodings

D. Whitley
Computer Science Department
Colorado State University
Fort Collins, CO 80523
email: whitleycs.colostate.edu

Abstract

A measure of complexity is proposed that counts the number of local minima in any given problem representation. A special class of functions with the maximum possible number of optima is also defined. A proof is given showing that reflected Gray code induce more optima than Binary over this special class of functions; by the No Free Lunch principle, reflected Gray codes therefore induces fewer optima over all other remaining functions.

1 INTRODUCTION

Over all possible functions, Gray codes and Standard Binary codes are equal in that they both cover the set of all possible bit representations [4]. In spite of this No Free Lunch result [5], applications oriented researchers have often argued for the use of Gray codes [1]. The debate as to whether Gray coding is better than Binary representations has been a classic example of where theory and practice clash. The results in this paper bring theory and practice closer together and yields new insights into the role of representation during search.

A measure of complexity is proposed that counts the number of local minima in any given problem representation. On average, functions that have fewer minima are assumed to be less complex than functions with more minima.

The set of functions which are considered are such that they can be remapped so that the range of the functions is 0 to $2^L - 1$ and the domain the values 0 to $2^L - 1$. This restriction of the domain and range has the advantage of mapping all functions that can be discretized and represented by an L bit encoding onto a well defined finite set of functions. We will refer to this set of functions as F, and f_j as a function in F.

Two neighborhood structures over F will be defined. Let $\mathbb{F}$ represent the set of wrapped neighborhoods for functions in F. For all integers, i, $i-1$ and $i+1$ are neighbors and the addition and subtraction operations are $mod\ 2^L - 1$ so that endpoint are also neighbors.

Let $\mathcal{F}$ represent the set of non-wrapping neighborhoods for functions in F. For all integers, i, $i-1$ and $i+1$ are neighbors except that endpoints have a single neighbor; thus 0 and $2^L - 1$ are not neighbors. This is a relatively minor difference, but one that turns out to be useful for proving properties about Gray and Binary representations.

The main argument of this paper is as follows.

Typically, when functions are encoded as bit strings, they are first mapped to $\mathcal{F}$ or $\mathbb{F}$. Note this neighborhood preserves local optima in the original function under search operators that move to adjacent points in the space. The neighborhoods $\mathcal{F}$ and $\mathbb{F}$ preserve the connectivity of the original function. After mapping to $\mathcal{F}$ or $\mathbb{F}$, a bit representation is introduced. Assuming that the original function has limited complexity (in that sense that it has fewer than the maximum possible number of local optima), it can be proven that Gray codes on average provide theoretical advantages over standard Binary encodings. In this situation, a special family of Gray codes (called "reflected" Gray codes) will in expectation induce fewer total optima than the corresponding set of Binary representations.

2 BIT REPRESENTATIONS

Let R denote the set of all bit representations over functions in F; R includes the string representation and the associated evaluations of those strings. When the parameters in the domain of every function in F are converted into their Binary representation, a one-to-one and onto mapping is created between F and R; Gray coding also induces a one-to-one and onto mapping between F and R, but the mapping is different. Hence, a No Free Lunch result holds: over all functions in F, both Gray and Binary encodings produce all possible representations in R.

2.1 "REFLECTED" GRAY CODES

Usually "Gray code" refers to Standard Reflected Binary Gray code [2]. A Gray code is any representation where for any integer i the bit representation of i is Hamming distance 1 away from the bit representations of the integers $i+1$ and $i-1$. The addition and subtraction operators are mod the number of points in the set from which i is drawn.

To define what is meant by "reflected," consider any Gray code for a function defined over 2^{L-1} points. We then construct a *reflected* Gray code for L bits as follows. Define the permutation π_{L-1} to be the ordered set of 2^{L-1} strings. Let $R(\pi_{L-1})$ reverse the order of the set of strings. Concatenate a 0 onto each string in π_{L-1} to generate Π and concatenate a 1 onto every string in $R(\pi_{L-1})$ to generate $R(\Pi')$. Then concatenate the two permutations Π and $R(\Pi')$ to generate π_L. Since π_{L-1} is a Gray code, the new permutation π_L is also a Gray code.

2.2 DEFINITIONS AND CONCEPTS

When referring to functions in "F" the neighborhood can be either $\mathbb{F}$ or $\mathcal{F}$.

Let FQ be the set of functions with less than or equal Q optima in F.

Let $\overline{FQ}$ be FQ-complement, the set of functions with more than Q optima in F.

Let RQ be the set of functions with less than or equal Q optima in R.

Let $\overline{RQ}$ be RQ-complement, the set of functions with more than Q optima in F.

Let $C(G, FQ)$ be the count of all optima for Gray coded functions in class FQ.

Let $C(G, \overline{FQ})$ be the count of all optima for Gray coded functions in class $\overline{FQ}$.

Let $C(B, FQ)$ be the count of all optima for binary coded functions in class FQ.

Let $C(B, \overline{FQ})$ be the count of all optima for binary coded functions in class $\overline{FQ}$.

Note that every bit function in R is the Gray coding for some function in F; it is also the Binary encoding for some function in F. Thus, over all functions in F, the total number of local optima under each representation is the same. Forall Q,

$$C(G, FQ) + C(G, \overline{FQ}) = C(B, FQ) + C(B, \overline{FQ})$$

It also follows that :

$$C(G, FQ) < C(B, FQ) \text{ IFF } C(G, \overline{FQ}) > C(B, \overline{FQ}).$$

A number of properties that relate to Gray codes have previously been proven by Whitley and Rana [4].

THEOREM 1 (The Gray-Compactness Theorem): *Let X be the number of local optima induced by a local neighborhood search operator over the bits in any Gray coded representation of any function in F. Let Y be the number of local optima induced under the neighborhood of the same function in $\mathbb{F}$ or $\mathcal{F}$: $X \leq Y$.*

For specific values of Q, one can construct functions in FQ that have a Binary representation in $\overline{RQ}$. However, when Gray encoding is involved, the following Theorem holds.

THEOREM 2: *Every function in FQ has a Gray representation, r_g in RQ. Every representation, $r_{g'}$ in $\overline{RQ}$ is the representation of a Gray encoding of a function in $\overline{FQ}$.*

Theorem 2 following directly from the Gray-Compactness Theorem: every function in FQ has Q or fewer minima and must have a Gray representation in RQ; every representation in $\overline{RQ}$ is the Gray representation of some function that must have more than Q optima and hence must be in $\overline{FQ}$.

THEOREM 3: *For all discrete functions in F over with bit representation of any length L, for $Q = 1$, the set of all Gray coded representation for functions in FQ induce fewer total minima than the set of all Binary coded representations for functions in FQ.*

Proofs for all of the above theorems are given by Whitley and Rana [4]. A corollary of Theorem 3 when $Q = 1$ is that $C(B, \overline{FQ}) < C(G, \overline{FQ})$ which means that Binary is better over $\overline{FQ}$ when $Q = 1$. However, consider the classification for all function in F for bit strings of length 3 given in Table 1. In this case there are $(2^3)!$ functions, all of which can be enumerated. The functions are decomposed into sets of functions with exactly "K" minima.

The Gray code in this case is the commonly used Standard Reflected Binary Gray. This empirical data illustrates a number of interesting properties. First for $K = 1, 2$ and 3, the Gray code induces fewer minima than Binary. For K = 4, Binary codes induce

FOR WRAPPING FUNCTIONS						
	#F	# of Min		Wins		
K	K Min	Gray	Binary	Gray	Binary	Ties
1	512	512	960	448	0	64
2	14,592	23,040	27,344	6384	2176	6032
3	23,040	49,152	49,392	7088	6704	9248
4	2,176	7,936	2,944	0	2160	16
Sum	40,320	80,640	80,640	13,920	11,040	15,360
FOR NON-WRAPPING FUNCTIONS						
	#F	# of Min		Wins		
K	K Min	Gray	Binary	Gray	Binary	Ties
1	128	128	264	120	0	8
2	7,680	11,904	15,016	3944	1016	2720
3	24,576	48,384	51,160	8720	6144	9712
4	7,936	20,224	14,200	1136	3880	2920
Sum	40,320	80,640	80,640	13,920	11,040	15,360

Table 1: A comparison of the number of minima under Gray and Binary encodings for all 3-bit functions of F with 1, 2, 3 and 4 minima.

fewer minima than Gray. The implications of No Free Lunch can be seen in the totals: the total number of optima in Gray and Binary is equal. However, there is also a "MiniMax" difference between Gray and Binary: when asking which representation is better in a one-to-one comparison, Gray is better more often. This is because Gray can be somewhat better on many functions, while being much worse than Binary on a few functions. However, the total number of optima under Gray and Binary stay the same. Wolpert and Macready [5] speculated that such MiniMax difference were possible. Radcliffe and Surry [3] explore properties of the MiniMax relationship. This example proves MiniMax differences do exist. But exactly on what functions is the Gray representation much worse than Binary?

The empirical evidence suggests that Gray is much worse than Binary on functions that have the maximum possible number of optima. On these functions, it is never possible to take more than 1 step in the search space before becoming stuck at a local optimum. By the No Free Lunch principle, then, Gray codes will induce fewer optima than Binary on all remaining functions.

Note that one consequence of Theorems 1 and 2, is that under Gray encoding, a bit representation with the maximum number of optima can only be produced by Gray coding a function in F with the maximum number of optima.

3 WORST CASE FUNCTIONS

A "worse case function" in F results if every other point is a minima (i.e. even numbered points can be minima and odd numbered points maxima, or vice versa). An equivalent worse case function in R occurs if all strings with even numbers of bits are minima (or maxima) while all strings with odd numbers of bits are maxima (or minima). Thus, the maximum number of minima in $\mathcal{F}$ or $\mathbb{F}$ is $MAX = 2^{L-1}$.

The remainder of this paper proves the following theorem.

THEOREM 4: *For all discrete functions in F with bit representation of any length L, for Q = MAX-1, the set obtained using any Reflected Gray Coded representation for functions in FQ induce fewer total minima than the set obtained using the Binary coded representation for functions in FQ; this holds for both neighborhoods $\mathbb{F}$ and $\mathcal{F}$.*

Before presenting a proof, several preliminary results are needed. Define $\mathcal{F}^{max}$ and $\mathbb{F}^{max}$ to be the set of functions under neighborhood $\mathcal{F}$ and $\mathbb{F}$ to be the set of functions with the maximum number of local optima: 2^{L-1}.

LEMMA 1. The Subset Lemma: $\mathbb{F}^{max} \subset \mathcal{F}^{max}$

PROOF:
Both $\mathbb{F}^{max}$ and $\mathcal{F}^{max}$ neighborhoods have minima at alternating points: only one endpoint can be an optimum. Let f represent a function in $\mathbb{F}^{max}$. When the endpoint of f is an optimum in $\mathbb{F}$, it must also be an optimum in $\mathcal{F}$, since its internal neighborhood does not change and it has no external neighbor. Thus $f \in \mathbb{F}^{max}$ implies $f \in \mathcal{F}^{max}$. Let f be a function in $\mathcal{F}^{max}$. An endpoint that is an optimum in $\mathcal{F}$ will fail to be an optimum in $\mathbb{F}$ if that optimum is greater than the opposite endpoint, which is a saddle. Thus $\mathbb{F}^{max}$ is a proper subset of $\mathcal{F}^{max}$. QED

It will first be shown that over all functions in $\mathcal{F}^{max}$ the set of Binary representations induce fewer minima than the set of all "reflected" Gray code representations. We will then specialize the result for $\mathbb{F}^{max}$. First, we need to establish some foundational concepts and principles.

First, note that we can divide $\mathcal{F}^{max}$ and $\mathbb{F}^{max}$ into *odd* and *even* functions. Since every other point is an optima, an *even* function has minima at the even numbered integers; an *odd* function has minima at the odd numbered integers. There is a one-to-one mapping between the *odd* and *even* functions. If Π is a permutation corresponding to an *even* functions, then $R(\Pi)$ is a unique corresponding *odd* function.

The Subset Lemma, $\mathbb{F}^{max} \subset \mathcal{F}^{max}$ still holds for the set of *even* functions in F. Thus, without loss of generality, proofs will be done only for the set of *even* functions, but apply equally to *odd* functions.

3.1 EXPECTED FREQUENCY

Let i be a value in the range (output) of the functions that make up $\mathcal{F}$ for L bit functions, where the range is 0 to $2^L - 1$. Define $T = 2^L - 1$.

Let $f_o(i,p)$ be the frequency with which the value i occurs as a local optimum at position p over all functions in $\mathcal{F}^{max}$. Let $f_s(i,p)$ be the frequency with which the value i occurs as a saddle at position p over all functions in $\mathcal{F}^{max}$. Let $\mathcal{E}(f(p))$ represent the expected value of the function (permutation) at position p.

LEMMA 2. The Position-Based Frequency Lemma for $\mathcal{F}^{max}$ and $\mathbb{F}^{max}$

The following frequency relationship hold for both $\mathcal{F}^{max}$ and $\mathbb{F}^{max}$, although the actual frequencies vary.
$\forall$ *i in the range of* $\mathcal{F}$, $1 < i < (T-1)$ *and* $\forall$ *p*

Property 1.
$$f_s(i,p) < f_s(i+1,p) \quad and \quad f_o(i,p) > f_o(i+1,p)$$

Property 2.
$$f_o(i,p) = f_s(T-i, T-p)$$

Property 3.
$\forall$ *j where* $j < T : \sum_{k=o}^{j} f_s(k, T-p) < \sum_{k=o}^{j} fr_o(k,p)$

Property 4.
$\forall p, q$ *if p is an optimum and q is a saddle*
$$\mathcal{E}(f(p)) < \mathcal{E}(f(q))$$

PROOF.

Property 1.

Case a For any position p which is an optimum and not an endpoint; consider i from smallest to largest.

$i = 1$ and $i = 2$ are always optimal in all functions in $\mathcal{F}^{max}$

In general, consider any value $i > 2$. There are $\binom{i}{2}$ ways to buffer i and make it saddle point with $(2^L - 3)$ valves remaining over which to construct all possible functions in $\mathcal{F}^{max}$ that have i as a saddle.

Consider $i+1$. There are $\binom{i+1}{2}$ ways to make $i+1$ a saddle point with $(2^L - 3)$ valves remaining over which to construct all possible functions in $\mathcal{F}^{max}$ that have k as a saddle.

Furthermore the $\binom{i}{2}$ ways to buffer i when constructed from values less than i are a proper subset of the $\binom{i+1}{2}$ ways available to buffer $i+1$ constructed from all values less than $i+1$. The ways of ordering remaining values is the same with respect to i or $i+1$ since they are adjacent values; in other words, i and $i+1$ have the same relative rank with respect to all other points.

Case b When considering the ways that points can appear as saddles the case is exactly symmetric to the situation in case a.

Case c When p is an endpoint position of the permutation the value 1 is never a saddle, but value 2 can be a saddle when buffered by 1.

Consider any value i. There are i ways to buffer i and make it a saddle point with $(2^L - 2)$ valves remaining over which to construct all possible functions in $\mathcal{F}^{max}$ that have i as a saddle.

Consider any value $i+1$. There are $i+1$ ways to buffer $i+1$ and make it a saddle point with $(2^L - 2)$ valves remaining over which to construct all possible functions in $\mathcal{F}^{max}$ that have i as a saddle.

Other than these changes in counting ways to build buffers, the situations are exactly the same as those that hold for points that are not endpoints.

Thus, the cases for optima and saddles are symmetric with identical probabilities and it follows that:

$$f_s(i,p) < f_s(i+1,p) \text{ and } f_o(i,p) > f_o(i+1,p)$$

Property 2.

Consider position p and $T-p$. If p is an optimum at the p^{th} position in the permutation when counting left to right starting at 0, then $T-p$ is a saddle at the p^{th} position in the permutation when counting right to left starting at T. However, if we switch the problem from minimization to maximization, then $T-p$ is an optimum at the p^{th} position in the permutation when counting right to left starting at T, and p is a saddle at the p^{th} position in the permutation when counting left to right starting at 0. Note that counting positions from left to right or right to left does not change the number of ways we can construct different functions in $\mathcal{F}^{max}$ after having placed an element i at position at p or element $T-i$ at position $T-p$: we can reverse the permutation, fill all open positions, then reverse the permutation again to its original order. Thus it follows that

$$\forall i, \forall p : f_o(i,p) = f_s(T-i, T-p)$$

Property 3.

We know $f_o(i,p) = f_s(T-i, T-p)$ regardless of whether p and $(T-p)$ are endpoints or not. It follows that forall p

$$\sum_{k=0}^{T} f_s(k, T-p) = \sum_{k=0}^{T} f_o(k,p)$$

However, because the frequencies in $f_s(i, T-p)$ monotonically increase with increasing i and the frequencies in $f_o(i,p)$ monotonically decrease with increasing i, it follows that as long as we sum over all j such that $j < T$ then

$$\forall\ j \text{ where } j < T : \sum_{k=o}^{j} f_s(k, T-p) < \sum_{k=o}^{j} fr_o(k,p).$$

Property 4.

$$f_s(i,q) < f_s(i+1,q) \Longrightarrow \mathcal{E}(f(q)) > \frac{1}{N}\sum_{i=0}^{2^L-1} i$$

$$f_o(i,p) > f_o(i+1,p) \Longrightarrow \mathcal{E}(f(p)) < \frac{1}{N}\sum_{i=0}^{2^L-1} i$$

QED

4 RECURSIVE CONSTRUCTION OF PERMUTATIONS

It is useful to look at functions as permutations, and to construct the set of functions/permutations recursively. Permutations over 2^{L+1} elements will be decomposed into sets of permutations over subsets of 2^L elements.

Divide the set of 2^{L+1} values into two equal sets. There are $\binom{2^{N+1}}{2^N}$ ways to do this. Construct and concatenate all possible permutations over each set. There are $(2^N!)(2^N!)$ ways to do this. The set of all permutations over any subset of 2^N unique values is isomorphic with the set of all functions in $\mathcal{F}_N$. Repeating this operation over all of the possible $\binom{2^{N+1}}{2^N}$ decompositions results in all possible functions in $\mathcal{F}_{N+1}$ since

$$2^{N+1}! = \binom{2^{N+1}}{2^N}(2^N!)(2^N!) = \frac{2^{N+1}!}{2^N!2^N!}2^N!2^N!$$

We now prove Theorem 4; the proof is given in two parts: 4a and 4b.

THEOREM 4a: *For all functions in $\mathcal{F}^{max}$ with bit representations of length L, the set of Reflected Gray code representations over all functions in $\mathcal{F}^{max}$ induce more total minima than the set of all Binary coded representations.*

PROOF

The proof is inductive and constructive.

By observation the set of all functions when $L = 2$ and $L = 3$ is such that $C(B, \mathcal{F}^{max}) < C(G, \mathcal{F}^{max})$.

We assume the theorem is true for N and show that the $N+1$ case must be true. Denote functions in $\mathcal{F}^{max}$ with $L = N$ bit encodings by $\mathcal{F}_N^{max}$. When $L = N+1$ functions in the set $\mathcal{F}_{N+1}^{max}$ can be constructed from functions in the set $\mathcal{F}_N^{max}$ in the following fashion.

Divide all possible values in the range of $\mathcal{F}_{N+1}$ into two equal sized sets. Label the two sets A' and B'; note that these are sets of values and not sets of functions. There are $\binom{2^{N+1}}{2^N}$ ways to split the set of values in $\mathcal{F}$ when $L = N+1$. We next look at all possible functions over sets A' and B'. This is also just all possible permutation over sets A' and B' and each set of permutations (i.e., functions) is isomorphic with $\mathcal{F}_N$ since we can sort the components of A' and B' and index these components from 0 to $2^N - 1$.

We then construct all possible functions over the sets A' and B' and then select only those functions that are also members of $\mathcal{F}_N^{max}$. We will denote the filtered sets by A and B. Note that A and B are sets of functions composed of elements drawn from A' and B' respectively, and both A and B are isomorphic with $\mathcal{F}_N^{max}$.

All functions which are members of $\mathcal{F}_{N+1}^{max}$ can be constructed by concatenating the corresponding sets of functions A and B. (both of which are isomorphic with $\mathcal{F}_N^{max}$). This construction works due to the fact that concatenation of two functions from $\mathcal{F}_N$ can destroy optima, but never create optima. Only the end points that are concatenated are affected. If both endpoints are optima, one must collapse. If both are saddles, the number of optima in $\mathcal{F}$ is not changed by concatenation. If one is an optimum and the other a saddle, the saddle cannot become an optimum, but the optimum may or may not collapse. There are no other cases.

The proof now proceeds in two parts.

PART 1: Construction of Set $\mathcal{X}$

When we consider all possible ways of splitting $\mathcal{F}_{N+1}$ into A' and B' then concatenate all possible sets A and B we obtain a set $\mathcal{X}$ such that

$$\pi \in \mathcal{F}_{N+1}^{max} \Longrightarrow \pi \in \mathcal{X}.$$

Both A and B are isomorphic with $\mathcal{F}_N^{max}$. *By the inductive hypothesis, the set of connections over A induces fewer minima in Binary than Gray.* The same is true of connections over B. Thus, we need only consider the new connections that are recreated when A and B are concatenated to create functions in $\mathcal{F}_{N+1}$.

If it is true for each particular sets of value A' and B' (and particular set of functions A and B) then it is true over all functions in $\mathcal{F}^{max}_{N+1}$.

When concatenating functions in A with functions in B, we can only concatenate functions with optima at even numbered indices with functions in B with even number indices. This does not effect the outcome of the following counting argument due to the symmetric of $\mathcal{F}^{max}_{N}$: when $\mathcal{F}^{max}_{N}$ is divide into odd and even functions, this create 2 equal and symmetric subsets. Only functions with optima in even positions are considered; the situation for functions with optima in odd-numbered positions and the counting arguments are identical.

Consider recursively constructing a function over 2^{L+1} elements from two permutations over 2^L elements. Under a Binary representation, the values from functions in A with even numbered indices are connected to values of functions in B with even numbered indices. This is because the binary mapping of the concatenated functions from A and B are identical, except when concatenated, the binary representations of the function from A are prefixed with a 0 and those of the function from B are prefixed by a 1.

Thus, under a Binary representation, optima in A are connected to optima in B and saddles in A are connected to saddles in B.

Any "reflected" Gray code, on the other hand, *reverses* the order of set of bit strings mapping the function in B. Then 0 is appended to the bit strings mapping onto the function from A and a 1 is appended to the *reversed* set of bit strings mapping to the function in B.

Thus, under any "reflected" Gray representation, optima in A are connected to saddles in B and saddles in A are connected to optima in B.

More precisely, an optimum at position p in A is connected to point p in functions in B using standard Binary representation; however, it is connected to the saddle point $T-p$ under any reflected Gray representation.

We next show that, in expectation, connecting optima to optima (as occurs under a Binary representation) causes more optima to collapse than connecting optima to saddles (as occurs under reflected Gray encodings).

Select and fix a specific function from A and iterate over all possible functions in B. Next consider any optimum in A at position p. Let x be the value of the optimum in A. We know that $x < T$, since T cannot be an optimum.

Let $\mathcal{E}(f(p))$ represent the expected value of the function (permutation) at position p. Therefore, iterating over all functions $f \in B$ we know from the Frequency theorem that

$$\sum_{j=o}^{x} f_s(j, T-p) < \sum_{j=o}^{x} fr_o(j,p).$$

and since smaller values appear more often as optima at p than as saddles at $T-p$, then

$$\mathcal{E}(f(p)) < \mathcal{E}(f(T-p))$$

therefore, concatenation between the permutation from A and all the permutations of B causes more optima to collapse in Binary than Gray.

PART 2: Filtering Set X for Members of $\mathcal{F}^{max}_{N+1}$

One additional issue must be addressed. Not all concatenations of a specific function in A with all functions in B results in a function in $\mathcal{F}^{max}_{N+1}$. Since we are working with even numbered optima, the relevant endpoint from A will be a saddle and the relevant endpoint from B will be an optimum. (Again, the case is symmetric for odd-numbered functions.) Thus, when the endpoint saddle in the final position of A is less than the endpoint "optimum" in the initial position of B, the optimum in B collapses and the resulting function is not a member of $\mathcal{F}^{max}_{N+1}$. Another way of viewing this is, for a specific function $f \in A$, we only concatenated with members of B where the first element in B is less than the last element of A. When the endpoint saddle in A is one of the 2 largest values in the set of 2^{N+1} values, *none* of the functions in B are discarded. In this case the desired result holds. When the final element of A is the value 2, the value 1 must also be in A (adjacent to 2) and *all* of the members of B in this case are discarded. In both cases, the result holds. But these are extreme cases.

We wish to show that when we filter out the members of $\mathcal{F}^{max}_{N+1}$ from the constructed set in the previous section, that the expected value of the optima connected under the binary representation is less than the expected value of the saddles connected under any reflected gray representation. Also, since we are filtering the set that was constructed, the inductive argument used in the construction cannot be used at this point.

Consider any functions/permutation in $\mathcal{F}^{max}_{N+1}$. Next consider the following set of points.

Optima	Saddles
$a = p$	$w = p + 1$
$b = 2^{T-1} + p$	$x = 2^{T-1} - p + 1$
$c = 2^{T-1} - p - 1$	$y = 2^{T-1} - p$
$d = 2^{T} - p - 1$	$z = 2^{T} - p$

Note that these points have a closed-neighborhood relationship, which will be referred to as the *Closed-Binary-Gray-neighborhood.* Points (a, b) are neighbors under Binary; points (c, d) are also neighbors under Binary. The pairs (a,w) (b,x) (c,y) and (d,z) are all neighbors under Binary, Gray and in F. But under any reflected Gray code, the following pairs of points are neighbors: (a,z) (d,w) (c,x) (b,y). Thus, each of the four optima in the closed-neighborhood has a Binary and Gray neighbor in this set of points.

We already know the set $\mathcal{X}$ has more optima under Gray code than Binary. For all functions in $\mathcal{X}$, if the pair (a,z) and the pair (b,y) are not end points in the functions that made up A and B, then for any specific function in $\mathcal{F}^{max}_{N+1}$, it is true that:

$$f(a) < f(w), f(b) < f(x), f(c) < f(y), f(d) < f(z)$$

$$\{f(a) + f(d)\} < \{f(w) + f(z)\}$$
and
$$\{f(b) + f(c)\} < \{f(x) + f(y)\}$$

Denote these the *closed-neighborhood* constraints. Since the functions are being constructed recursively, the closed-neighborhood constraints also hold for any arbitrary optima and some appropriate value of $l \geq 3$, where l replaces T when computing the positions of the closed-neighborhood. When $l = 2$, the rightmost two bits are $a' = 00, w' = 01, b' = 10, x' = 11$, and if a, b, w, x are the corresponding strings, then $f(a) < f(w)$ and $f(b) < f(x)$. A closed-neighborhood still exists, since (b,w) and (a,x) are neighbors under Gray code. When $l = 2$, there can only be exactly 1 optimum in Binary, but 1 or 2 can exist in Gray. When $l = 1$, the rightmost bit is $a' = 0$ and $w' = 1$, and the corresponding string pair (a,w) are neighbors under both Gray and Binary – and there is no difference between Binary and Gray. Thus, a closed-Binary-Gray-neighborhood exists for every optima, and all closed-neighborhoods favor a Gray code.

Assume none of the points in the closed-neighborhood are endpoints.

Under the Binary encoding, *only two* of the four points that are optima in F, i.e., (a, b, c, d), can remain local optima due to collapse within this set of points. This collapse always occurs. (It can also be proven that the number of optima in Binary must be less than 2^{L-2} for functions in $\mathcal{F}^{max}$ and $\mathbb{F}^{max}$ and thus over half of the optima in Binary must always collapse [4]).

However, under a Gray code, if $f(a) < f(z)$ and $f(b) < f(y)$ and $f(c) < f(x)$ and $f(d) < f(w)$ then there is no collapse of optima under Gray code through this set of points.

Note that it is possible that

$$\{f(a) < f(w) < f(d) < f(z)\} \text{ or } \{f(d) < f(z) < f(a) < f(w)\}$$

which allows at most one optimum to collapse in Gray code; the connecting "or" in this case is an exclusive-or. It is also possible that

$$\{f(c) < f(y) < f(b) < f(x)\} \text{ or } \{f(b) < f(x) < f(c) < f(y)\}$$

which allows at most one additional optimum to collapse in Gray code. But at most 2 optima can collapse under Gray code via this set of points. By construction there are many cases were no optima collapse under a Gray code (e.g. all optima are less than all saddles.) And cases exist were only one may collapse. Thus, in expectation, more optima must collapse under Binary than Gray code.

Of course, this is already known to be true by construction for the set $\mathcal{X}$. However, as $\mathcal{X}$ is filtered, we also know there are sets of points that violate the usual constraints over the set of closed-neighbors: a, b, c, d, w, x, y z. These are the endpoints. We further filter from $\mathcal{X}$ those cases where

$$a = 0, \; c = T - 1, \; w = 1, \; y = T \quad \text{in function A and}$$

$$b = 0, \; d = T - 1, \; z = T, \; x = 1 \quad \text{in function B}$$

where $f(y) < f(b)$, thus causing b to collapse as a minimum and not be a member of $\mathcal{F}^{max}_{N+1}$. We thus filter out the *internal* endpoints (e.g., the last element of A, the first element of B) that violate the normal constraints associated with the closed-neighborhood structure. Note that

$$\{f(a) + f(d)\} < \{f(w) + f(z)\}$$
and
$$\{f(b) + f(c)\} < \{f(x) + f(y)\}$$

not only remains true for the filtered set, but is now true for a larger percentage of the set $\mathcal{F}^{max}_{N+1}$ compared to $\mathcal{X}$. The only potential violations are now at *external endpoints* at the beginning and end of the concatenated strings. But we also know from the frequency lemma,

For $q = w, x, y, z$ and For $p = a, b, c, d$
$$f_o(i, p) > f_o(i + 1, p) \Longrightarrow \mathcal{E}(f(p)) < \frac{1}{N} \sum_{i=0}^{2^L - 1} i$$
$$f_s(i, q) < f_s(i + 1, q) \Longrightarrow \mathcal{E}(f(q)) > \frac{1}{N} \sum_{i=0}^{2^L - 1} i$$

and that this holds even for endpoints. Thus, for any optimum at position p, that optimum is connected to

another optimum under a Binary encoding and to a saddle in q under Gray; the only exception is when that optimum is connected to the same saddle in both Binary and Gray. Therefore, in expectation the optimum collapses more often under a Binary encoding than any reflected Gray encoding.
QED.

THEOREM 4b: *For all functions in $\mathbb{F}^{max}$ with bit representations of length L, the set of reflected Gray code representations over all functions in $\mathbb{F}^{max}$ induce more total minima than the set of all Binary coded representations.*

PROOF: To show this is true for $\mathbb{F}^{max}$ we need only filter the set of strings $\mathcal{F}^{max}_{N+1}$ a second time. This time all cases of external endpoints that violate the closed neigh-
borhood property are removed. Thus for all points,

$$\{f(a)+f(d)\} < \{f(w)+f(z)\}$$
and
$$\{f(b)+f(c)\} < \{f(x)+f(y)\}$$

We also know

For $q \in \{w,x,y,z\}$ and for $p \in \{a,b,c,d\}$

$$f_o(i,p) > f_o(i+1,p) \Longrightarrow \mathcal{E}(f(p)) < \frac{1}{N}\sum_{i=0}^{2^L-1} i$$
$$f_s(i,q) < f_s(i+1,q) \Longrightarrow \mathcal{E}(f(q)) > \frac{1}{N}\sum_{i=0}^{2^L-1} i$$

Thus, for every string and every neighborhood the closed-neighborhood constraints hold, and the expected value of every optimum is less than the expect value of any saddle. Therefore, in expectation optima collapse more often under a Binary encoding than any reflected Gray encoding.
QED.

In hindsight, the proofs might be simplified, but the current versions provide very detailed information that support the following, much stronger, conjecture.

Conjecture: *For every individual string in $\mathbb{F}^{max}$ the Gray code for that individual string induces more optima than the corresponding Binary encoding.*

The empirical data proves this is true for 3 bit strings. The fact that the closed-neighborhood constraint must hold for every neighborhood of every string suggest that Gray must induce more minima than Binary, even at the level of the individual string. It is also possible to show that this is true for functions over all strings of length 4. This is done by iteratively constructing the closed-neighborhoods one at a time and propagating the resulting constraints to show that for every string, every collapse of any remaining optima in Gray space forces the collapse of at least one or more remaining optima in Binary space.

5 CONCLUSIONS

This paper proves that any Reflected Gray code preserves more optima than a Binary representation over the set of "worst case" functions that have the maximum possible number of minima. These functions have 2^{L-1} optima and a local search operator can never make more than 1 move before encountering a local optimum. These functions are extremely unlikely to be encountered in practice.

Due to the No Free Lunch result, since Gray codes induces more optima than Binary over functions with 2^{L-1} optima, Gray codes induce fewer optima than Binary over all remaining functions–providing Gray code with a Free Lunch over a clear and pragmatically defined subset of all possible functions.

6 ACKNOWLEDGEMENTS

This work was supported by NSF grant IRI-9503366 and AFOSR grant F49620-97-1-0271. The US Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright notation thereon.

References

[1] R. Caruana and J. Schaffer. Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms. In *Proc. of the 5th Int'l. Conf. on Machine Learning*. Morgan Kaufmann, 1988.

[2] Keith E. Mathias and L. Darrell Whitley. Transforming the Search Space with Gray Coding. In J. D. Schaffer, editor, *IEEE Int'l. Conf. on Evolutionary Computation*, pages 513–518. IEEE Service Center, 1994.

[3] N.J. Radcliffe and P.D. Surry. Fundamental limitations on search algorithms: Evolutionary computing in perspective. volume 1000. Springer-Verlag, 1995.

[4] Darrell Whitley and Soraya Rana. Representation, search and genetic algorithms. In *Proceedings of the Fourteenth National Conference on Artificial Intelligence*, 1997.

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