Landscapes and Other Art Forms.

Darrell Whitley
Computer Science, Colorado State University

Blind No More = GRAY BOX Optimization

Darrell Whitley
Computer Science, Colorado State University

With Thanks to: Francisco Chicano, Gabriela Ochoa, Andrew Sutton and Renato Tinós

Know your Landscape! And Go Downhill!

Know your Landscape! And Go Downhill!

Know your Landscape!

Know your Landscape!

ROADMAP

MOVE FAST AND BREAK THINGS!

No Free Lunch: Not what you think.

Multi-Funnel Landscapes

Tunneling Between Local Optima in O(1) time: TSP, MAXSAT

Selecting Improving Moves in Constant Time.

High Order Mutation Testing

Elementary Landscapes

NEXT GENERATION GENETIC ALGORITHMS

MOVE FAST AND BREAK THINGS!
GRAY BOX OPTIMIZATION

1. Deterministic Recombination in $O(n)$ time.

   Tunneling Between Local Optima in O(1) time.

   TSP and MAXSAT

2. Deterministic Move Selection in $O(1)$

   Select Improving Moves in Constant Time.

No Need for Random Mutation. Deterministic Recombination.

No Free Lunch

Variations on No Free Lunch

For ANY measure of algorithm performance:

The aggregate behavior of ALL possible search algorithms is equivalent when compared over any two discrete functions.

The aggregate behavior of any two search algorithms is equivalent when compared over all possible discrete functions.

At each distinct "iteration" of search
the aggregate behavior of all possible search algorithms is IDENTICAL at each and every iteration.

No Free Lunch

Variations on No Free Lunch

Consider any algorithm A_i applied to function f_j.
On(A_i, f_j) outputs the order in which A_i visits the elements in the codomain of f_j. For every pair of algorithms A_k and A_i and for any function f_j, there exist a function f_l such that

$$On(A_i, f_j) \equiv On(A_k, f_l)$$

Consider a "BestFirst" local search with restarts.
Consider a "WorstFirst" local search with restarts.

For every j there exists an l such that

$$On(BestFirst, f_j) \equiv On(WorstFirst, f_l)$$

No Free Lunch

ENUMERATION is a search algorithm.

Thus, No Free Lunch implies that on average,
no search algorithm is better than enumeration.

NFL IGNORES RESAMPLING

An algorithm is modeled as a permutation
representing the order in which new points are tested.

Behavior is defined in terms of the evaluation function output
which defines the co-domain of the function.

No Free Lunch

Assume that one is given a fixed set of co-domain values.
Set of Functions = Set of Permutations.

BEHAVIORS	FUNCTIONS
A1: 1 2 3	F1: A B C
A2: 1 3 2	F2: A C B
A3: 2 1 3	F3: B A C
A4: 2 3 1	F4: B C A
A5: 3 1 2	F5: C A B
A6: 3 2 1	F6: C B A

No Free Lunch

Assume $(A > B)\&(B > C)$.
Take 2 steps, return the maximum found.

	F1	F2	F3	F4	F5	F6
A1	A	A	A	B	A	B
A2	A	A	B	A	B	A
A3	A	A	A	B	A	B
A4	B	B	A	A	A	A
A5	A	A	B	A	B	A
A6	B	B	A	A	A	A

No Free Lunch

Theorem:
NFL holds for a set of functions IFF
the set of functions form a permutation set.

The "Permutation Set" is the closure of a set
of functions with respect to a permutation operator.
(Schmacher, Vose and Whitley–GECCO 2001).

F1:	0	0	1	2	F7:	0	2	0	1
F2:	0	1	0	2	F8:	0	2	1	0
F3:	1	0	0	2	F9:	1	2	0	0
F4:	0	0	2	1	F10:	2	0	0	1
F5:	0	1	2	0	F11:	2	0	1	0
F6:	1	0	2	0	F12:	2	1	0	0

No Free Lunch

Theorem:
NFL holds for a set of functions IFF
the set of functions form a permutation set.

CAREFUL! IS THIS TRUE?

Hmmmmmmm.

One direction of the IFF was always very clear.

No Free Lunch

Theorem:
NFL holds for a set of functions IFF
the set of functions form a permutation set.

CAREFUL! IS THIS TRUE?
We (Whitley, Vose) were "sloppy" about our assumptions.

TRUE: if you DO NOT know what algorithms will be compared.

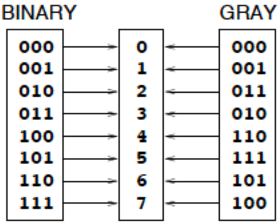
FALSE: if you know in advance that a fixed number (e.g., 2) of specific predefined algorithms will be compared.

Take Home Message

Beware of theoreticians :-)

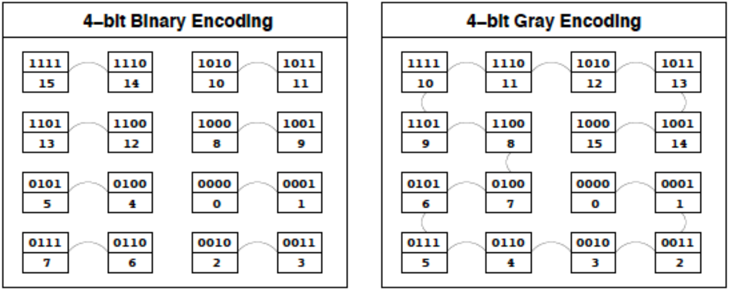
With a small shift in assumptions,
half of what you know (There is a $proof$ for that!)
is suddenly very wrong.

No Free Lunch

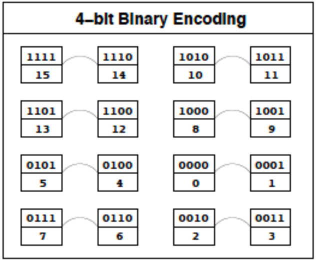


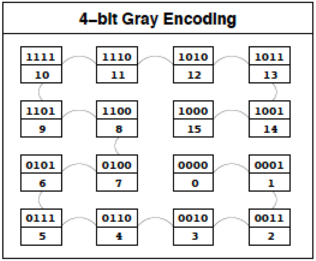
	Gray Matrix	Degray Matrix
3-bits	$\begin{vmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{vmatrix}$	$\begin{vmatrix} 1 & 1 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \end{vmatrix}$
5-bits	$\begin{vmatrix} 1 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}$	$\begin{vmatrix} 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 \end{vmatrix}$

No Free Lunch



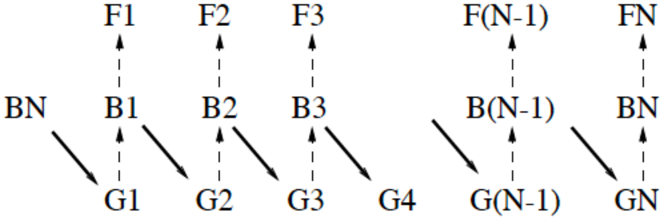
No Free Lunch





If the function f(integer) is unimodal,
then f(gray) is unimodal, but f(binary) is not!

No Free Lunch



No Free Lunch

R1:	000	001	010	011	100	101	110	111
R2:	000	001	011	010	110	111	101	100
R3:	000	001	010	011	101	100	111	110
R4:	000	001	011	010	111	110	100	101
R5:	000	001	010	011	100	101	110	111

FOCUSED No Free Lunch

We can use the following indices into f_b to create a subset of 4 functions.

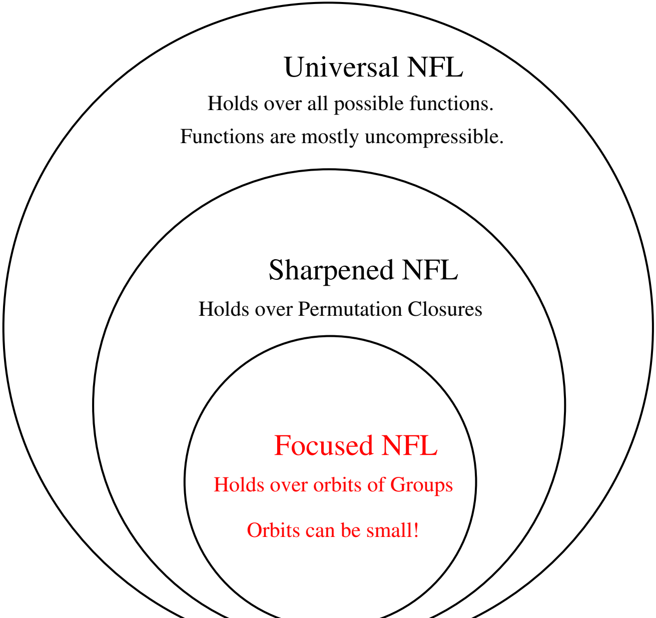
P1:	0	1	2	3	4	5	6	7
P2:	0	1	3	2	6	7	5	4
P3:	0	1	2	3	5	4	7	6
P4:	0	1	3	2	7	6	4	5

Technically, a *group* has been defined that is closed with respect to the application of a graying matrix M to the bit representation b.

This group has two cycles of length $1, 2$ and 4,
which can be denoted by: (0)(1)(2 3)(4 6 5 7)

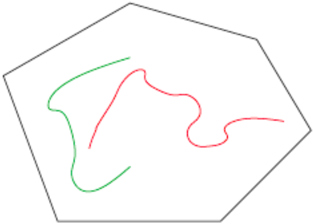
Since 2 and 1 are factors of 4, the orbit is 4.

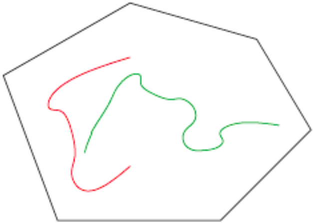
FOCUSED No Free Lunch



Searching for a limited number of steps.

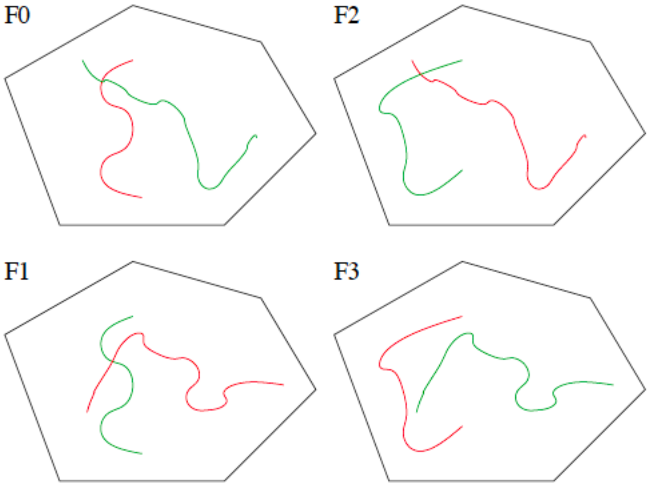
Benignly Interacting Algorithms



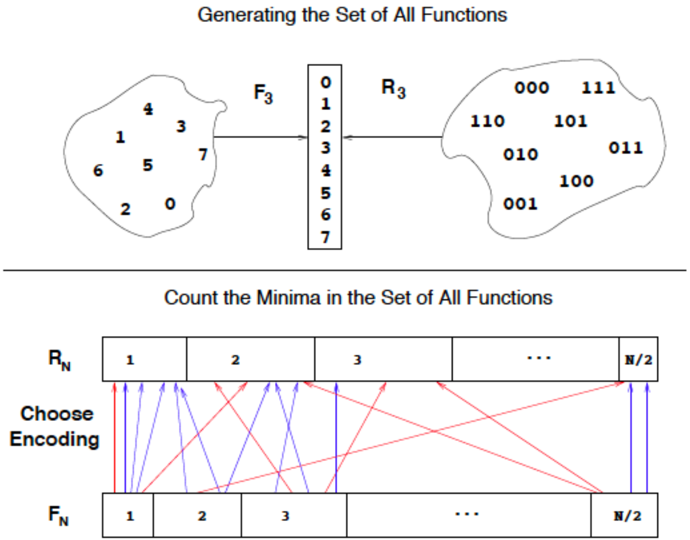


Searching for a limited number of steps.

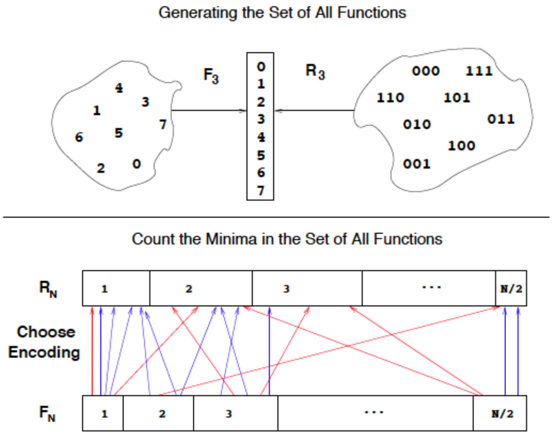
Constructing Focused Sets of Functions



FOCUSED No Free Lunch

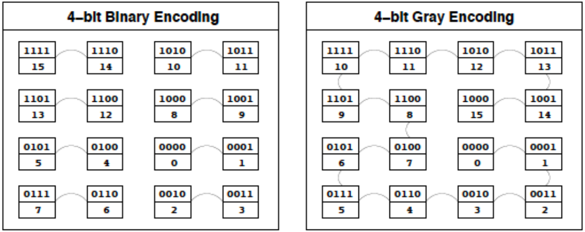


FOCUSED No Free Lunch



(What sneaky assumption did I just introduce?)

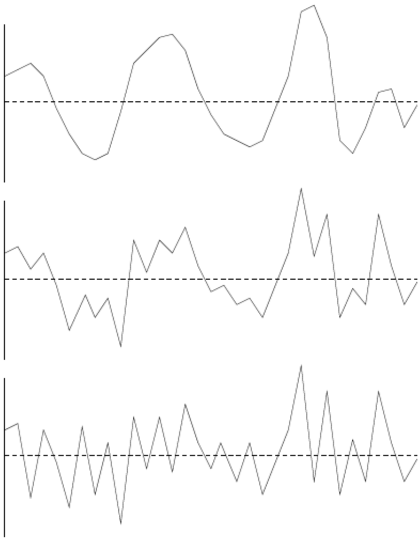
FOCUSED No Free Lunch



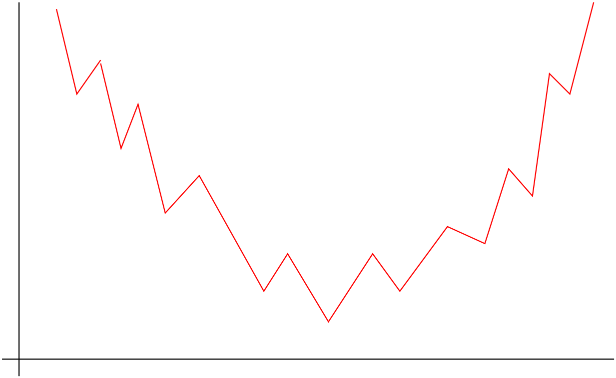
If the function f(integer) is unimodal,
then f(gray) is unimodal, but f(binary) is not!

(What sneaky assumption did I just introduce?)

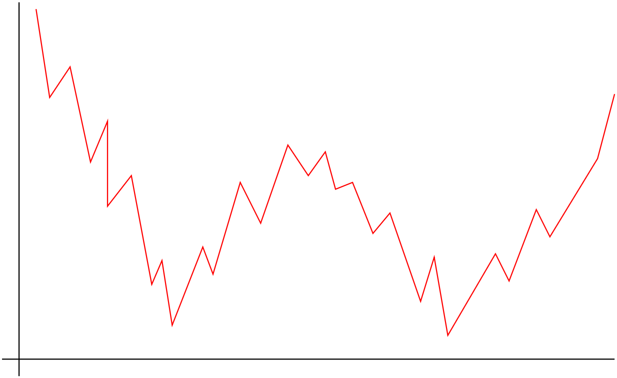
The Landscape Matters!



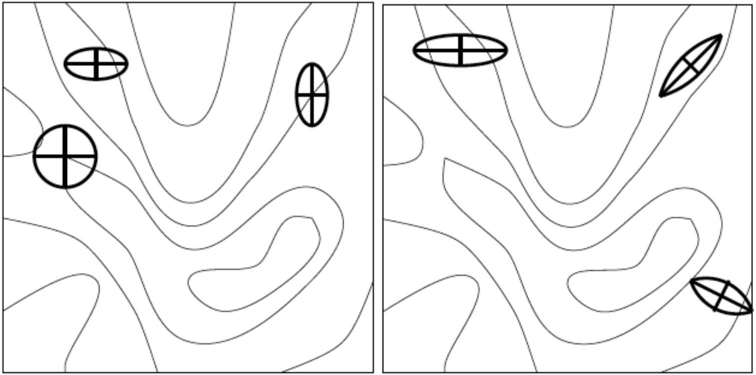
A Rough Bowl: One Funnel



Two Bowls: Two Funnels



Evolution Strategies: adaptive mutations



Simple Mutations　　Correlated Mutation via Rotation

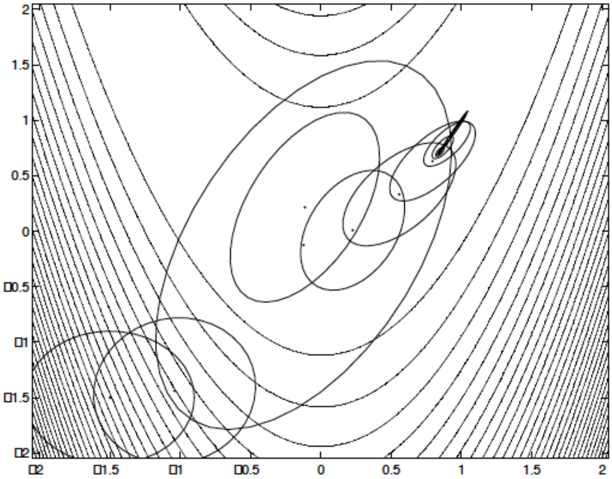
CMA and PCA "Rotations"

CMA and Principal Components

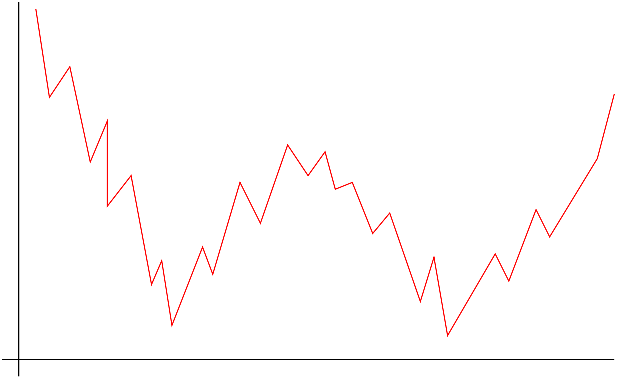
Given a data set of sample points, we want to perform an eigenvalue/eigenvector decomposition. The eigenvectors are represented by a rotation matrix $\mathbf{R}$. Let Λ be the diagonal eigenvalue matrix. Let $\mathbf{X}$ represent a matrix of data vectors. Using PCA we find $\mathbf{R}$ and Λ such that

$$\mathbf{R} \cdot \mathbf{X}\mathbf{X}^T = \Lambda \mathbf{R}$$

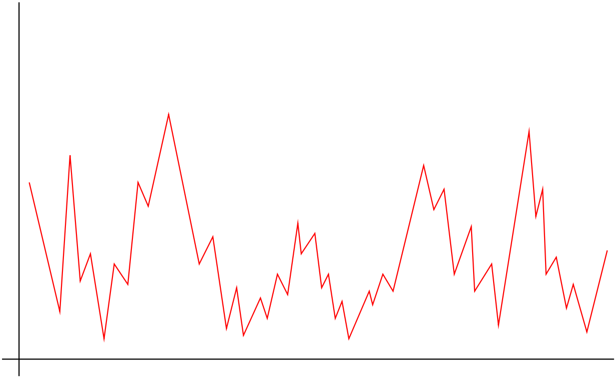
CMA running on Rosenblatt's Function



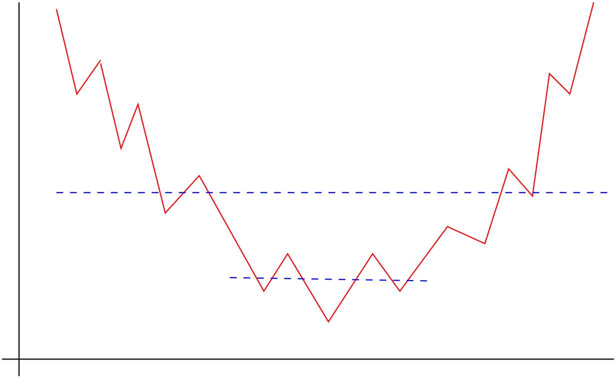
Breaking CMA: Two Funnels



Multi-Funnel



In some cases, you can measure “Dispersion”



In some cases, you can measure "Dispersion"