# Landscapes and Other Art Forms.

Darrell Whitley
Computer Science, Colorado State University

# NK-Landscapes and MAX-kSAT

For example: A Random NK Landscape: $n = 10$ and $k = 3$.
The subfunctions:

$$
\begin{array}{llll}
f_0(x_0, x_1, x_6) & f_1(x_1, x_4, x_8) & f_2(x_2, x_3, x_5) & f_3(x_3, x_2, x_6) \\
f_4(x_4, x_2, x_1) & f_5(x_5, x_7, x_4) & f_6(x_6, x_8, x_1) & f_7(x_7, x_3, x_5) \\
 & f_8(x_8, x_7, x_3) & f_9(x_9, x_7, x_8) &
\end{array}
$$

But this could also be a MAXSAT Function,
or an arbitrary Spin Glass problem.

# A General Result over Bit Representations

By Constructive Proof: Every problem with a bit representation and a closed form evaluation function can be expressed as a quadratic (k=2) pseudo-Boolean Optimization problem. (See Boros and Hammer)

$$xy = z \;\; iff \;\; xy - 2xz - 2yz + 3z = 0$$

$$xy \neq z \;\; iff \;\; xy - 2xz - 2yz + 3z > 0$$

Or we can reduce to k=3 instead:

$$f(x_1, x_2, x_3, x_4, x_5, x_6)$$

becomes (depending on the nonlinearity):

$$f1(z_1, z_2, z_3) + f2(z_1, x_1, x_2) + f3(z_2, x_3, x_4) + f4(z_3, x_5, x_6)$$

# GRAY BOX OPTIMIZATION

We can construct "Gray Box" optimization for pseudo-Boolean optimization problems composed of M subfunctions, where each subfunction accepts at most k variables.

Exploit the *general properties of every* Mk Landscape:

$$f(x) = \sum_{i=1}^{m} f_i(x)$$

Which can be expressed as a Walsh Polynomial

$$W(f(x)) = \sum_{i=1}^{m} W(f_i(x))$$

Or can be expressed as a sum of $k$ Elementary Landscapes

$$f(x) = \sum_{i=1}^{k} \varphi^{(k)}(W(f(x)))$$

# BLACK BOX OPTIMIZATION

Don't wear a blind fold when walking about London if you can help it!

# BLACK BOX OPTIMIZATION

The Secret Weapon of Black Box Optimization:

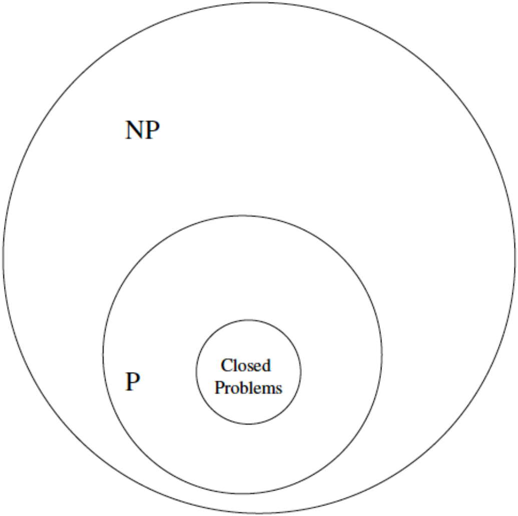
You can only prove polynomial runtime convergence
if the target problem is already in the class P.

This is why most theory (runtime analysis) papers
solve ONEMAX, or Shortest Path, or Minimal Spanning Tree.

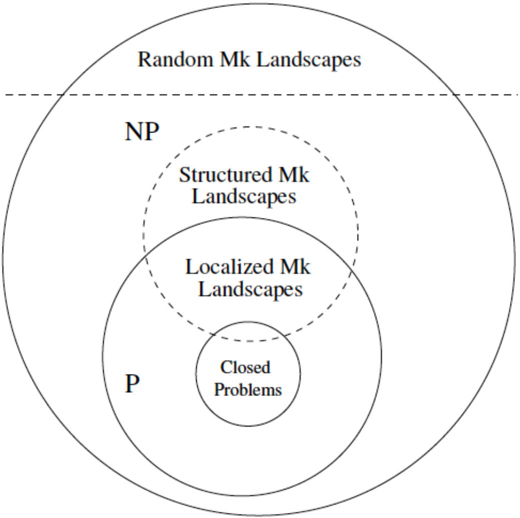
It is difficult to do runtime analysis on problems that are NP-hard.

# BLACK BOX OPTIMIZATION

The Secret of Black Box Optimization:



# GRAY BOX OPTIMIZATION



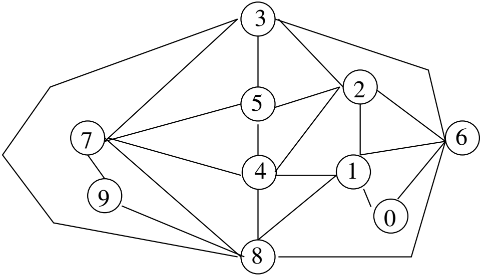
# GRAY BOX OPTIMIZATION

Don't wear a blind fold when walking about London if you can help it!

Keep the methods as general as necessary, but not too general.

Many (most?) combinatorial search methods are not really "Black Box."

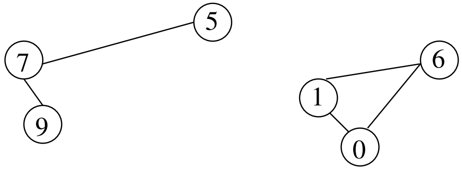
# The Variable Interaction Graph




The Variable Interaction Graph

There is a vertex for each variable in the Variable Interaction Graph (VIG). There must be fewer than $2^k\ M = O(N)$ Walsh coefficients. There is a connection in the VIG between vertex $v_i$ and $v_j$ if there is a non-zero Walsh coefficient indexed by $i$ and $j$, e.g., $w_{i,j}$.

# The Recombination Graph: a reduced VIG

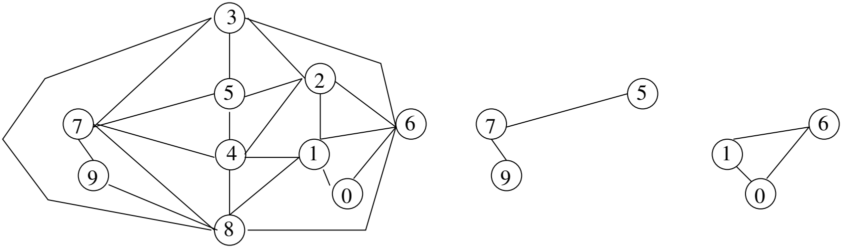



The decomposed Recombination Graph

When recombining the solutions 0000000000 and 1100011101, the vertices and edges associated with shared variables 2, 3, 4, 8 are deleted to yield the **recombination graph**.

If the recombination graph of $f$ contains $q$ connected components, then Partition Crossover returns the best of $2^q$ solutions.

# Decomposed Evaluation




The Variable Interaction Graph

The decomposed Recombination Graph

A new evaluation function can be constructed:

$$g(x) = c + g_1(x_5, x_7, x_9) + g_2(x_0, x_1, x_6)$$

where $g(x)$ can be used to evaluate any solution (parents or offspring) that resides in the subspace **000***0*.

# Partition Crossover and Local Optima

**The Subspace Optimality Theorem:** For any $k$-bounded pseudo-Boolean function $f$, if Parition Crossover is used to recombine two parent solutions that are locally optimal, then the offspring must be a local optima in the hyperplane subspace defined by the bits shared in common by the two parents.

Example: if the parents 0000000000 and 1100011101
are locally optimal, then the best offspring
is locally optimal in the hyperplane subspace **000***0*.

# Partition Crossover and Local Optima

**Corolllary:** The only possible improving move for offspring generated from parents that are locally optimal must flip a bit that the parents shared in common.

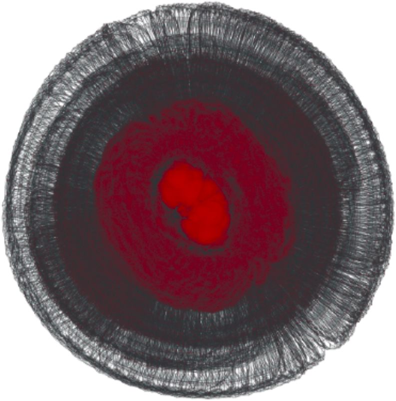
Illustration:

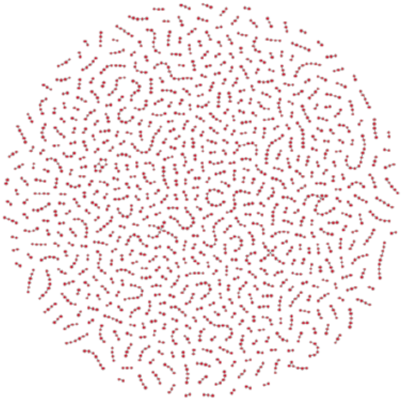
$$g(x) = c + g_1(x_5, x_7, x_9) + g_2(x_0, x_1, x_6)$$

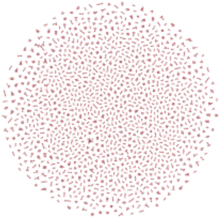
If the parents were locally optimal, every subfunction $g_i$ is locally optimal and offspring cannot be improved by a bit flip.

The only improving moves are on shared bits: **000***0*.

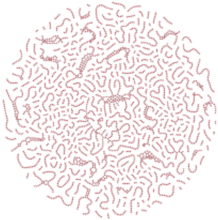
# Decomposed Evaluation for MAXSAT



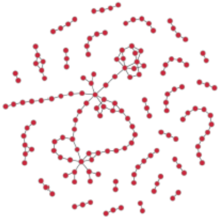




atco_enc3_opt1_13_48



LABS_n088_goal008



SAT_instance_N=49

aaai10-ipc5

# MAXSAT Number of recombining components

| Instance | N | Min | Median | Max |
|---|---|---|---|---|
| aaai10ipc5 | 308,480 | 7 | 20 | 38 |
| AProVE0906 | 37,726 | 11 | 1373 | 1620 |
| atcoenc3opt19353 | 991,419 | 937 | 1020 | 1090 |
| LABSno88goal008 | 182,015 | 231 | 371 | 2084 |
| SATinstanceN111 | 72,001 | 34 | 55 | 1218 |

Imagine:
crossover "scans" $2^{1000}$ local optima and returns the best in O(n) time