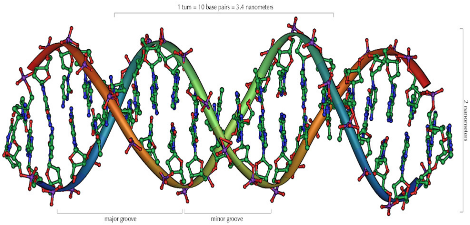
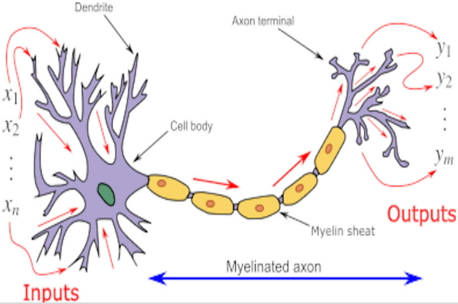
Analysis of Binary Coded Genetic Algorithms

Andreas Geyer-Schulz | 22. April 2025

INFORMATION SERVICE AND ELECTRONIC MARKETS andreas.geyer-schulz@kit.edu





Status:

1.0 First Draft.

Feedback welome.

Learning Goals

You should

- know and understand three approaches to the analysis of schema for binary representations, namely Walsh functions, deceptive problems, and hypercubes, and their implications for coding a representation (binary versus Gray coding)

Content

1. Schema and Representations

2. Deceptive and Hard GA-Problems

3. Hypercubes

Reminder: What Is a Schema?

$$0_1\ 001{*}1\ 0_7\ {*}$$

Matching strings:

All matching strings

Strings	Fitness
00010100	1100
00010101	1101
00011100	1200
00011101	1201
0001*10*	$\bar{f}_s = 1150.5$

In population at time t (State)

Strings	Fitness
00010100	1100
00010101	1101
00010101	1101
00010101	1101
00011101	1201
0001*10*	$\bar{f}_{s,B_t} = 1120.8$

Length: 8

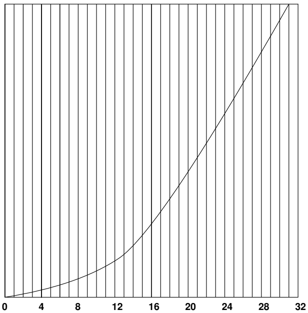
Number of defined positions (order): 6

Defining Length: $6 = 7 - 1$ (last defined position minus first defined position)

A Schema Selects Point Sets over a the Domain of a Function

We visualize in the following how a schema selects point sets over the domain of a function.

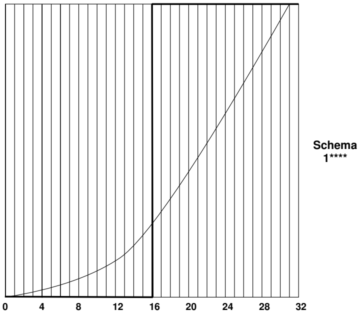
We consider maximizing the function $y = x^2_{\text{decimal}}$ with the domain $x \in \{0, \ldots, 31\}$:



with x coded as a 5-bit binary string: $\mathbf{x} \in \{0, 1\}^5$.

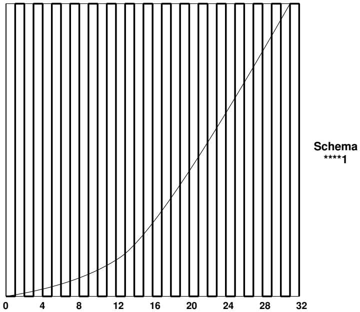
Visualization of a Schema

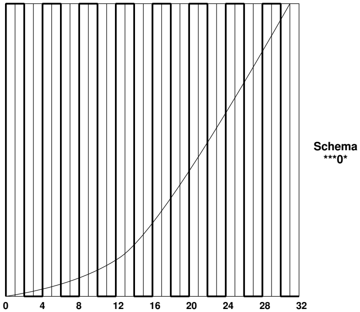
For the visualization we overlay the graph of the function with a rectangular wave form with amplitude values in $\{+1, -1\}$ which assign a $+1$ to all points matching the schema and a -1 to all points not matching the schema.



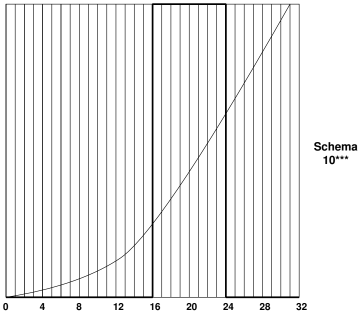
The schema $1 * * * *$ divides the domain in two halves.

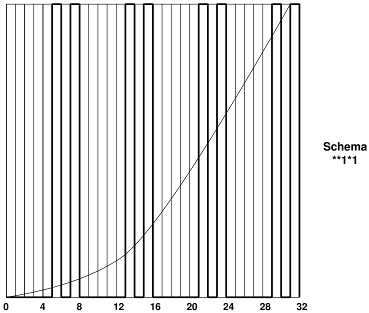
More Divisions in Two Halves





Higher Order Schemata

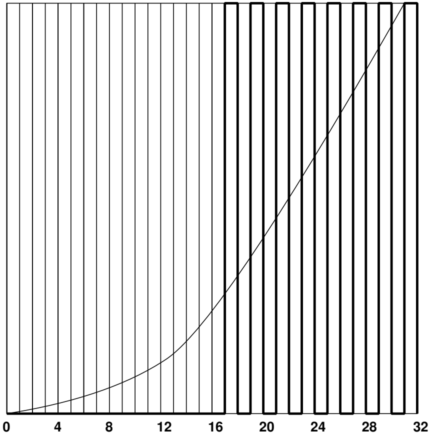




The number of points selected depends on the number of defined positions:

$$\frac{2^k}{2^{dp}}_{k=5,dp=2} = \frac{2^5}{2^2} = \frac{32}{4} = 8$$

Schemata Can Be Combined ...



Combining Schema ****1 and Schema 1** Is Schema 1***1**

A Schema Is a Walsh Function

Definition

A Walsh function has a rectangular wave form with two amplitude values, namely $+1$ and -1. It is defined over a limited time interval T. A Walsh function is defined by two arguments, namely the order number n and the time period t which is usually normalized by $\frac{t}{T}$:

$$WAL(n, t) \tag{1}$$

n denotes the number of zero crossings and t the time period.

For schemata, n is the number zero crossings and $t = \frac{dom(\mathbf{x})}{2^k}$. We use discrete Walsh functions to represent schemata.

Walsh Functions Are Orthogonal Sets I

The value of a discrete Walsh function with $N = 2^k$ terms is given by a continued product:

$$WAL(n_{k-1}, n_{k-2}, \ldots, n_0; t_{k-1}, t_{k-2}, \ldots, t_0) = \prod_{r=0}^{k-1} (-1)^{n_{k-1-r}(t_r + t_{r+1})} \qquad (2)$$

where $n_{k-1}, n_{k-2}, \ldots, n_0$ and $t_{k-1}, t_{k-2}, \ldots, t_0)$ are n and t in their binary representation.

Walsh Functions Are Orthogonal Sets II

The sum of the products of any two discrete Walsh functions is given as the binary summation

$$\sum_{t_{k-1}}^{1} \sum_{t_{k-2}}^{1} \cdots \sum_{t_0=0}^{1} WAL(n_{k-1}, n_{k-2}, \ldots, n_0; t_{k-1}, t_{k-2}, \ldots, t_0) \times WAL(m_{k-1}, m_{k-2}, \ldots, m_0; t_{k-1}, t_{k-2}, \ldots, t_0) \quad (3)$$

By substitution of equation (2) into equation (3) we get:

$$\sum_{t_{k-1}}^{1} \sum_{t_{k-2}}^{1} \cdots \sum_{t_0=0}^{1} \prod_{r=0}^{k-1} (-1)^{(n_{k-1-r}+m_{k-1-r})(t_r+t_{r+1})}$$

$$= \prod_{r=0}^{k-1} \sum_{t_r=0}^{1} (-1)^{(n_{k-1-r}+m_{k-1-r})(t_r+t_{r+1})} \quad (4)$$

$$= \prod_{r=0}^{k-1} \left\{1 + (-1)^{(n_{k-1-r}+m_{k-1-r})}\right\}$$

Walsh Functions Are Orthogonal Sets III

If each $n_t = m_t$ (both are either zero or 1) then equation (4) becomes:

$$\prod_{r=0}^{k-1}(1+1) = 2^k = N$$

If at least one $n_t \neq m_t$ then at least one term of the product in equation (3) becomes zero giving a zero product.
In terms of decimal arguments m, n, and t we get

$$\sum_{t=0}^{N-1} WAL(m,t)WAL(n,t) = \begin{cases} N & \text{for } n = m \\ 0 & \text{for } n \neq m \end{cases} \tag{5}$$

- This implies that Walsh functions form an orthogonal set. By division by N the set of Walsh functions can be normalised to form an **orthonormal system**.
- Orthonormal means orthogonal ($\mathbf{u} \cdot \mathbf{v} = 0$) and and normal (norm (length): $\| u \| = 1$).
- The orthogonal set of Walsh functions set is **ordered**. Several orderings – for different purposes – exist.

The Partition Coefficient Transform I

- We consider a mapping d from the k-bit string into the real numbers (the decoding function of the GA):

$$d : \{0, 1\}^k \xrightarrow{d} R$$

When we want to evaluate a function, e.g. $f(x) = x^2$ with x in binary representation, we have to write $f(d(x))$.

- We denote a schema as $H \in \{0, 1, *\}^k$. We write $H = (h_{k-1}, \ldots, h_i, \ldots, h_0)$.
When we want to evaluate a function with a schema as argument, we write $f(H)$. The value of $f(H)$ is the mean of all $d(x)$ matching the schema:

$$f(H) = \frac{1}{\mid \{d(x) \in H\} \mid} \sum_{\{d(x) \in H\}} f(d(x))$$

The Partition Coefficient Transform II

- We define a function $\alpha : \{0, 1, *\} \xrightarrow{\alpha} \{0, 1\}$ by:

$$\alpha(h_i) = \begin{cases} 0 & h_i = * \\ 1 & \text{otherwise} \end{cases}$$

- We define a **partition number function** $j : \{0, 1, *\}^k \xrightarrow{j} N_0$ for those schemata which share the same fixed positions:

$$j(H) = \sum_{i=0}^{k-1} \alpha(h_i) 2^i$$

Examples: $j(***) = 0$. $j(**0) = 0 + 0 + 1 = 1$ and $j(**1) = 1$ have the same partition number. $j(0*1) = 5$

The Partition Coefficient Transform III

- We define a **partition sign function** $\rho : \{0, 1, *\}^k \xrightarrow{\rho} \{-1, +1\}$ which returns a value of $+1$ if a partition contains an even number of zeros and a value of -1 otherwise:

$$\rho(H) = \prod_{i=0}^{k-1} (-1)^{\beta(h_i)}$$

with $\beta : \{0, 1, *\} \xrightarrow{\beta} \{0, 1\}$ defined as

$$\beta(h_i) = \begin{cases} 1 & h_i = 0 \\ 0 & \text{otherwise} \end{cases}$$

The Partition Coefficient Transform IV

- We can express $f(H)$ as the sum of the **partition coefficients** ϵ_j over **all similarity supersets** H' **which contain** H:

$$f(H) = \sum_{H' \supseteq H} \rho(H')\epsilon_{j(H')}$$

- We give some examples for similarity supersets $H' \supseteq H$:
For $H = 111$: $H' \supseteq H = \{111, 11*, 1*1, *11, 1**, *1*, **1, ***\}$
For $H = ***$: $H' \supseteq H = \{***\}$
For $H = **0$: $H' \supseteq H = \{**0, ***\}$
For $H = 1*0$: $H' \supseteq H = \{1*0, 1**, **0, ***\}$

The Partition Coefficient Transform V

- The partition coefficients ϵ_j for a complete hierarchy (the simularity superset from an individual (a schema without wildcards) to a schema consisting only of wildcards) can be computed by solving the linear system of partition coefficient equations which consists of the partition coefficient equations in the hierarchy. For $f(x) = x^2$ coded as three bit unsigned integer, the equation system is:

$$
\begin{aligned}
f(***) &= \epsilon_0 \\
f(**1) &= \epsilon_0 + \epsilon_1 \\
f(*1*) &= \epsilon_0 + \epsilon_2 \\
f(*11) &= \epsilon_0 + \epsilon_1 + \epsilon_2 + \epsilon_3 \\
f(1**) &= \epsilon_0 + \epsilon_4 \\
f(1*1) &= \epsilon_0 + \epsilon_1 + \epsilon_4 + \epsilon_5 \\
f(11*) &= \epsilon_0 + \epsilon_2 + \epsilon_4 + \epsilon_6 \\
f(111) &= \epsilon_0 + \epsilon_1 + \epsilon_2 + \epsilon_3 + \epsilon_4 + \epsilon_5 + \epsilon_6 + \epsilon_7
\end{aligned}
$$

The Partition Coefficient Transform VI

- To compute all ϵ_j, we need to compute all $f(H)$ in a complete hierarchy (2^k values). For our example:

$j(H)$	H	$f(H)$	ϵ_j
0	***	17.5	17.5
1	**1	21.0	3.5
2	*1*	24.5	7.0
3	*11	29.0	1.0
4	1**	31.5	14.0
5	1*1	37.0	2.0
6	11*	42.5	4.0
7	111	49.0	0.0

  From the ϵ_j, we can compute any schema average we wish:
  $f(**0) = \epsilon_0 - \epsilon_1 = 17.5 - 3.5 = 14.$
  Verify by taking the mean of all matching schemata:
  $(f(000) + f(010) + f(100) + f(110))/4 = (0 + 4 + 16 + 36)/4 = 14.0$

Example Computation of the Partition Coefficient Transform

$f(H) = \sum_{H' \supseteq H} \rho(H') \epsilon_{j(H')}$
Let $H = \{*01\}$. What is $f(H)$?

- The similarity supersets of H are $\{***\}$, $\{**1\}$, $\{*0*\}$, and $\{*01\}$.
- The partition numbers are
  $j(***) = 0$, $j(**1) = 1$, $j(*0*) = 2$, and $j(*01) = 3$.
  Therefore, we need the partition coefficients ϵ_0, ϵ_1, ϵ_2, and ϵ_3.
- The signs of the partition coefficents are
  $rho(***) = 1$, $rho(**1) = 1$, $rho(*0*) = -1$, and $rho(*01) = -1$.
- $f(H) = \epsilon_0 + \epsilon_1 - \epsilon_2 - \epsilon_3 = 17.5 + 3.5 - 7 - 1 = 13$
- **Consistency Check:** By direct computation, we get
  $(f(decimal(001)) + f(decimal(101)))/2$
  $= (f(1) + f(5))/2 = (1 + 25)/2 = 13$

The Meaning of Partition Coefficients

- ϵ_0 is the **average of fitness value of all individuals**.
- We can compare competing schemata:
  $f(**1) = \epsilon_0 + \epsilon_1$ and $f(**0) = \epsilon_0 - \epsilon_1$
  ϵ_1 measures the **influence of switching a single bit** at position 0 of the schema. For the bit at position 1, use ϵ_2. For the bit at position 2, use ϵ_4.
- For higher-order schemata, some ϵ_j model **interaction effects**.

$$f(*11) = \underbrace{\epsilon_0}_{\text{average}} + \underbrace{\epsilon_1}_{\text{influence of}_{**1}} + \underbrace{\epsilon_2}_{\text{influence of}_{*1*}} + \underbrace{\epsilon_3}_{\text{interaction of last 2 bits}}$$

- **Hint:** Similar to the analysis of the contribution of variables and variable combinations for linear regression models.

Deceptive and Hard GA-Problems

A genetic algorithm has problems to find the optimal solution if the building block hypothesis is violated.

- **Goal:** To construct the smallest problem in which the building block hypothesis is violated:
  - Low order building blocks lead to suboptimal longer building blocks, if the building block hypothesis is violated (explained on the next slide).
  - We construct a **minimal deceptive problem**:
    Schemata with 2 defined positions.
- **How does deception affect the convergence of the genetic algorithm to the global optimum?**

Conditions for Deception

We consider 4 schemata with 2 defined positions (order 2):

$$\begin{aligned} f_{00} &= ***0*****0* \\ f_{01} &= ***0*****1* \\ f_{10} &= ***1*****0* \\ f_{11} &= *** \underbrace{1*****1}_{\text{defining length}} * \end{aligned}$$

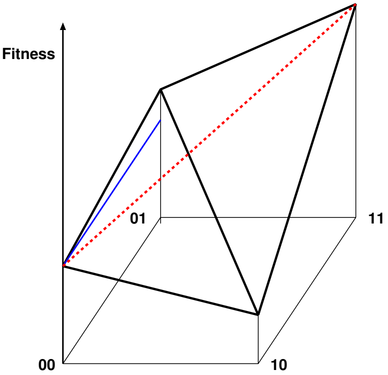
Let f_{11} be the global optimum. This implies the following conditions:
$f_{11} > f_{00}$, $f_{11} > f_{01}$, and $f_{11} > f_{10}$.
Deception (violation of the building block hypothesis) means that one or more of the suboptimal order 1 schemata ($0*$, $*0$) is better than the optimal order 1 schemata ($1*$, $*1$):

$$\begin{aligned} f(0*) &> f(1*) & \quad 1/2(f(00)+f(01)) &> 1/2(f(10)+f(11)) \\ f(*0) &> f(*1) & \quad 1/2(f(00)+f(10)) &> 1/2(f(01)+f(11)) \end{aligned}$$

Analysis of the Deception Conditions

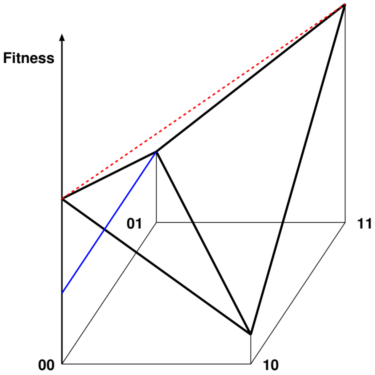
- Both deception conditions can not hold simultaneously, because this violates the assumption that f_{11} is the global optimum:
  Proof: We add the two conditions:
  $f(00) + 1/2f(01) + 1/2f(10) > 1/2f(01) + 1/2f(10) + f(11))$
  By subtracting $1/2f(01) + 1/2f(10)$ from both sides, we get
  $f(00) > f(11)$ which contradicts the assumption.
- We normalize all fitness values: $r = \frac{f(11)}{f(00)}$, $c = \frac{f(01)}{f(00)}$, and $c' = \frac{f(10)}{f(00)}$.
  We rewrite the globality conditions: $r > 1$, $r > c$, and $r > c'$.
  We choose $f(0*) > f(1*)$ and rewrite it in normalized form:
  $1 + c > c' + r$. This implies $1 + c - c' > r$.
- We see that we have two cases depending on $c > 1$ or $c \leq 1$ which lead to 2 types of minimal deceptive problems:
  Type I: $f_{01} > f_{00}$ if $c > 1$.
  Type II: $f_{00} \geq f_{01}$ if $c \leq 1$.

The Minimal Deceptive Problem of Type I



- The blue line shows $f_{01} > f_{00}$.
- The red line helps to to see that f_{01} is not on the plane defined by f_{00}, f_{10}, and f_{11}. This implies that the problem can not be represented as a linear combination $f(x_1, x_2) = b + \sum_{i=1}^{2} a_i x_i$.

The Minimal Deceptive Problem of Type II

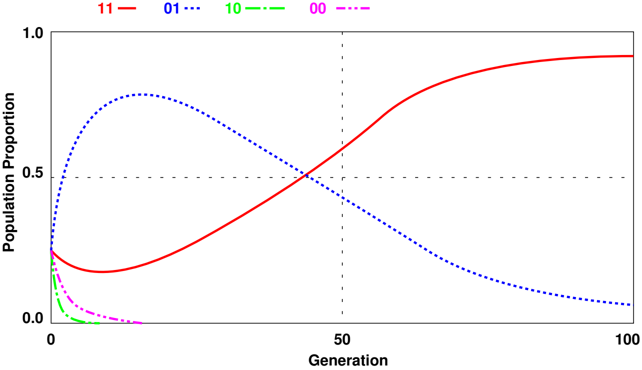


- The blue line shows $f_{00} \geq f_{01}$.
- The red line helps to to see that f_{01} is not on the plane defined by f_{00}, f_{10}, and f_{11}. This implies that the problem can not be represented as a linear combination $f(x_1, x_2) = b + \sum_{i=1}^{2} a_i x_i$.

How Does a Minimal Deceptive Problem of Type I Affect Global Convergence of the GA?

- By specification of the difference equations of the schema theorem (with $p_{mutation} = 0$) for the minimal deceptive problem, we can perform a dynamic analysis and numerical simulations (left as an exercise).
- For the minimal deceptive problem of type I ($f_{01} > f_{00}$) one can prove for any nonzero starting proportions of the four schemata, that $\lim_{t\to\infty} P'_{11} = 1$ where P'_{11} is the proportion of schemata 11 in the population.
- Global convergence is **not affected** for minimal deceptive problems of type I.

Proportions of Schemata for a Minimal Deceptive Problem of Type I



(Sketch. Simulation left as an exercise.)

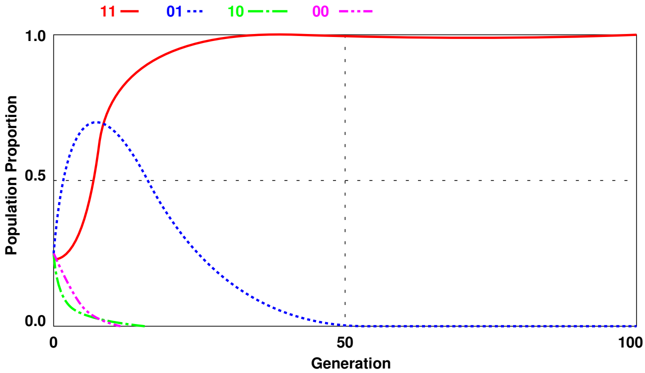
How Does a Minimal Deceptive Problem of Type II Affect Global Convergence of the GA?

For the minimal deceptive problem of type II, we have two cases, depending on initial conditions:

1. **Convergence.** The solution converges to the global optimum despite deception.
2. **Divergence.** The solution diverges from the global optimum.
   If the schema 00 has a too great initial proportion, then the schema 11 may be overwhelmed with convergence to the second best solution.

However, the genetic algorithm converges to the best solution for **most starting conditions**.

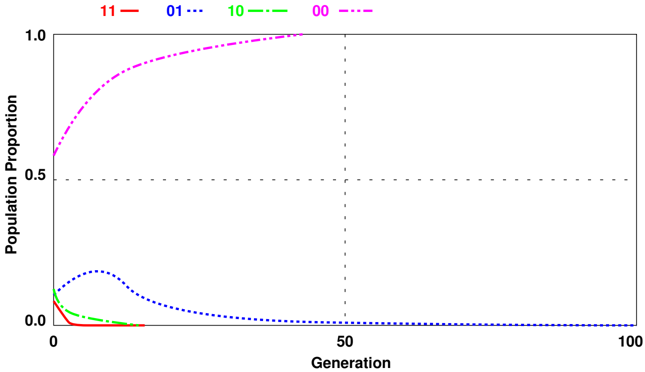
Minimal Deceptive Problems of Type II: Convergence



- **11** is the schema with the highest function value.
- Each schema has a proportion of 25 percent in the starting population.

(Sketch. Simulation left as an exercise.)

Minimal Deceptive Problems of Type II: Divergence



- **11** is the schema with the highest function value.
- The second best schema **00** has a proportion of 60 percent, each of the other schemata a proportion of 10 percent in the starting population.

(Sketch. Simulation left as an exercise.)

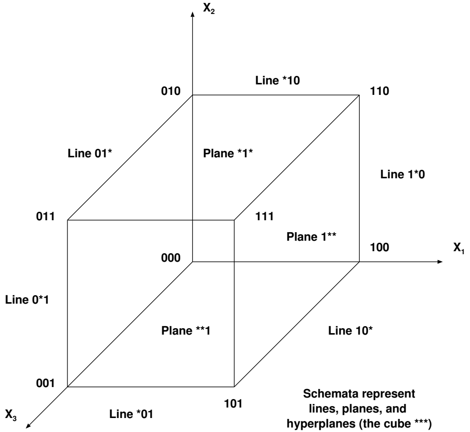
Testing a Problem for Deception with Partition Coefficients

- We use once again $f(x) = x^2$ coded with 3 bit unsigned integer.
- The schema 111 is optimal.
- What are the 1-bit deception conditions?
  $f(**1) < f(**0)$, $f(*1*) < f(*0*)$, and $f(1**) < f(0**)$.
- In terms of ϵ_j this implies:
  $\epsilon_1 < 0$, $\epsilon_2 < 0$, $\epsilon_4 < 0$.
- We know that
  $\epsilon_1 = 3.5$, $\epsilon_2 = 7.0$, $\epsilon_4 = 14.0$.
- This implies that the problem is not deceptive.

Constructing Deceptive Problems with Partition Coefficients

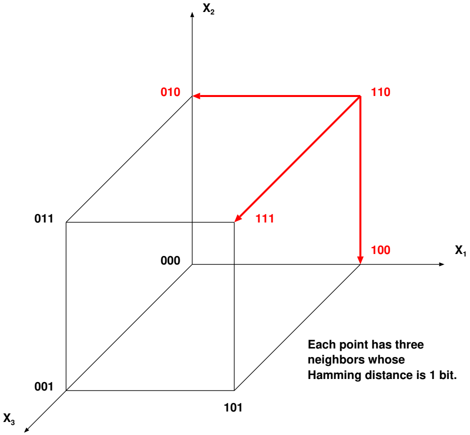
- Express the optimality conditions for f_{1^k} as partition coefficients. For $k = 3$, we get have $f_{111} > f_{000}$, $f_{111} > f_{001}, \ldots$
  Goldberg starts with
  $\epsilon_1 + \epsilon_3 + \epsilon_5 + \epsilon_7 > 0$
  I recommend programming the definitions ...
  Doing this manually is error prone!
- Construct the 1-bit, 2-bit, ... deception conditions in terms of ϵ_j.
  For the 1-bit problem, e.g. $\epsilon_1 < 0$.
  For the 2-bit problem, e.g. $\epsilon_1 + \epsilon_2 < 0$.
- Search for concrete values which fulfill the optimality and at least one of the deception conditions and construct appropriate functions (by linear regression).

Schemeta are Lines, Planes, Cubes, ...



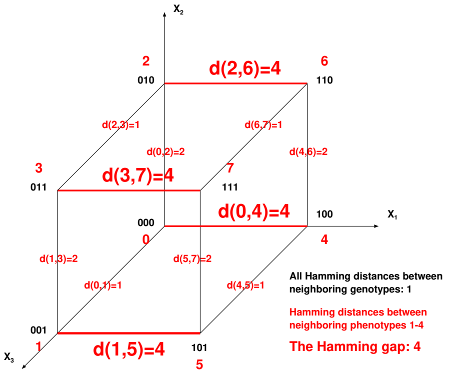
The genetic algorithm samples hyperplanes and the fitness of a hyperplane is the mean fitness of the set of points on the hyperplane.

Local Neighborhoods on the Hypercube



Each point has three
neighbors whose
Hamming distance is 1 bit.

An Illustration of the Hamming Cliff



The phenotypes of the hyperplanes $0**$ and $1**$ have a Hamming distance of 4.

The Hamming cliff is a **problem of binary coding**: A small distance in the local neighborhood of the genotype corresponds to a large distance in the phenotype.

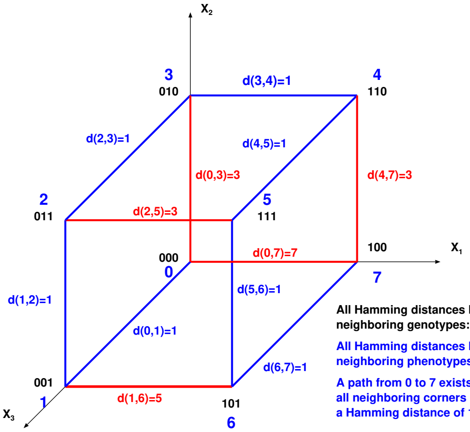
A Desirable Property of a Coding

Definition

Small distances in the space of genotypes should correspond with small distances in the space of phenotypes.

Can we rearrange our coding, so that the Hamming Gap vanishes?

Gray Coding preserves distances



All Hamming distances between neighboring genotypes: 1

All Hamming distances between neighboring phenotypes: 1

A path from 0 to 7 exists on which all neighboring corners have a Hamming distance of 1

Gray coding preserves distances when mapping from genotype to phenotype.

An Algorithm for Gray Coding

Encoding:

1. Code the number as a k-bit binary string.
2. Convert the k-bit binary string as follows:
   We convert the k-bit binary number $\mathbf{x} \in \{0, 1\}^k$ into the corresponding Gray code $\mathbf{y} \in \{0, 1\}^k$ by the mapping gray : $B^k \to B^k$:

$$y_i = \begin{cases} x_i, & \text{if } i = 1, \\ x_{i-1} \bigoplus x_i & \text{otherwise} \end{cases}$$

   where $\bigoplus$ denotes addition modulo 2.

Decoding:

$$x_i = \bigoplus_{j=1}^{i} y_j, \quad \forall i \in \{1, \ldots, k\}$$

Recommended Reading I

- A nice introduction to genetic algorithms is David Goldberg's book *Genetic Algorithms in Search, Optimization and Machine Learning* [Goldberg, 1989]. We have used Goldberg's explanations [Goldberg, 1989, pp. 41-54] as motivating examples as well as appendix E [Goldberg, 1989, pp. 373-378] in this lecture.
- The use of Walsh functions for formal proofs of the schema theorem is due to Bethke (Ph.D. Thesis) and Holland [Holland, 1987]. A complete analysis of the simple genetic algorithm with this technique has been performed by Michael Vose [Vose, 1999].
  The mathematical background on Walsh functions as well as efficient algorithms (down to silicon) can be found in [Beauchamp, 1975].

Recommended Reading II

- Deceptive problems have been introduced by [Goldberg, 1987] and [Bridges and Goldberg, 1987]. For additional examples of the analysis of genetic algorithms with Walsh functions see [Goldberg and Bridges, 1990] and [Bridges and Goldberg, 1991].
- An excellent book on representation issues is Franz Rothlauf's *Representations for Genetic and Evolutionary Algorithms* [Rothlauf, 2006]. You find the Gray coding algorithms on page 101 in [Rothlauf, 2006].

Problem Environments and R-Scripts I

- PartitionCoefficientTransform.R in directory */INNGA05: Contains scripts for coefficient numbers and signs.

References I

- Beauchamp, K. G. (1975).
  Walsh Functions and Their Applications.
  Academic Press, London.
- Bridges, C. L. and Goldberg, D. E. (1987).
  An analysis of reproduction and crossover in a binary-coded genetic algorithm.
  In Grefenstette, J. J., editor, *Genetic Algortihms and their Applications (ICGA'87). Proceedings of the Second International Conference on Genetic Algorithms*, pages 9 – 13. MIT, Lawrence Erlbaum Associates.
- Bridges, C. L. and Goldberg, D. E. (1991).
  The nonuniform walsh-schema transform.
  In Rawlins, G., editor, *Foundations of Genetic Algorithms 1*, pages 13 – 22. Morgan Kaufmann Publishers, San Mateo.

References II

Goldberg, D. (1987).
Computer-aided pipeline operation using genetic algorithms and rule learning. part ii. rule learning control of a pipeline under normal and abnormal conditions.
Engineering with Computers, 3:47 – 58.

Goldberg, D. E. (1989).
Genetic Algorithms in Search, Optimization and Machine Learning.
Addison-Wesley, Reading, 1 edition.

Goldberg, D. E. and Bridges, C. L. (1990).
An analysis of a reordering operator on a ga-hard problem.
Biological Cybernetics, 62(5):397 – 405.

References III

Holland, J. H. (1987).
Genetic algorithms and classifier systems: Foundations and future directions.
In Grefenstette, J. J., editor, *Genetic Algortihms and their Applications (ICGA'87). Proceedings of the Second International Conference on Genetic Algorithms*, pages 82 – 89. MIT, Lawrence Erlbaum Associates.

Rothlauf, F. (2006).
Representations for Genetic and Evolutionary Algorithms.
Springer, Heidelberg.

Vose, M. D. (1999).
The Simple Genetic Algorithm: Foundations and Theory.
Bradford Books, MIT Press, Cambridge.