Foundation of GAs

Live and Die: The fundamental Theorem

Genes and Alleles

Alphabet V+ = {0, 1, *}

H: a schema taken from V+

The order of schema H:

o(H)= number of fixed positions present in the template

The defining length of a schema H:

δ(H)= distance between the first and last specific string position

Live and Die: The fundamental Theorem

m(H,t) = number of particular schema H at time t in the population A(t)

During reproduction a string A_i is selected with probability

$$p_i = \frac{f_i}{\sum f_j}$$

Live and Die: The fundamental Theorem

Effect of Reproduction:

$$m(H,t+1) = m(H,t)\frac{f(H)}{f}$$

where f(H) is the average fitness of schema H

Above-average schemata grow and below-average schemata die off

$$m(H,t+1) = m(H,t)\frac{(f+cf)}{f} = (1+c)m(H,t)$$

$$m(H,t) = m(H,0)(1+c)^t$$

Live and Die: The fundamental Theorem

Effect of Crossover:

one-point crossover

random selection of a mate

random selection of a crossover site

p_c : crossover rate

p_s : crossover survival probability

$$p_s \geq 1 - p_c \frac{\delta(H)}{l-1}$$

Effect of Reproduction + Crossover

$$m(H,t+1) \geq m(H,t)\frac{f(H)}{f}\left[1 - p_c \frac{\delta(H)}{l-1}\right]$$

Live and Die: The fundamental Theorem

Assumption: Crossover within the defining length of the schema is always disruptive.

Not true!

Consider a schema 11*****.

If a string such as 1110101 were recombined between the first two bits with a string such as 1000000 or 0100000, no disruption occurs in schema 11*****.

Also, if 1000000 and 0100000 were recombined exactly between the first and second bit a new offspring becomes a member of 11*****.

Live and Die: The fundamental Theorem

$$m(H,t+1) = m(H,t)\frac{f(H)}{f}(1 - p_c losses) + p_c gains$$

$$m(H,t+1) \geq m(H,t)\frac{f(H)}{f}\left[1 - p_c \frac{\delta(H)}{l-1}\left(1 - \frac{m(H,t)}{n}\right)\right]$$

Assumption: Selection of the first parent is fitness based and the second parent is chosen randomly.

When both parents are chosen based on fitness, the form becomes

$$m(H,t+1) \geq m(H,t)\frac{f(H)}{f}\left[1 - p_c \frac{\delta(H)}{l-1}\left(1 - \frac{m(H,t)}{n}\frac{f(H)}{f}\right)\right]$$

Live and Die: The fundamental Theorem

Schemata with both above-average observed performance and short defining length are going to be sampled at exponentially increasing rate.

Effect of Reproduction + Crossover + Mutation:

p_m : mutation rate

$$m(H,t+1) \geq m(H,t)\frac{f(H)}{f}\left[1-p_c\frac{\delta(H)}{l-1}\left(1-\frac{m(H,t)}{n}\frac{f(H)}{f}\right)\right](1-p_m)^{o(H)}$$

Schema Theorem

Short,

low-order,

above-average schemata (building blocks)

receive exponentially increasing trials in subsequent generations.

Schema Processing at Work

TABLE 2.1 GA Processing of Schemata—Hand Calculations

String Processing

String No.	Initial Population (Randomly Generated)	x Value (Unsigned Integer)	$f(x)$ x^2	pselect_i $\frac{f_i}{\Sigma f}$	Expected count $\frac{f_i}{\bar{f}}$	Actual Count from Roulette Wheel
1	0 1 1 0 1	13	169	0.14	0.58	1
2	1 1 0 0 0	24	576	0.49	1.97	2
3	0 1 0 0 0	8	64	0.06	0.22	0
4	1 0 0 1 1	19	361	0.31	1.23	1
Sum			1170	1.00	4.00	4.0
Average			293	0.25	1.00	1.0
Max			576	0.49	1.97	2.0

Schema Processing

		Before Reproduction: String Representatives	Before Reproduction: Schema Average Fitness $f(H)$
H_1	1 * * * *	2,4	469
H_2	* 1 0 * *	2,3	320
H_3	1 * * * 0	2	576

TABLE 2.1 (*Continued*)

String Processing

Mating Pool after Reproduction (Cross Site Shown)	Mate (Randomly Selected)	Crossover Site (Randomly Selected)	New Population	x Value	$f(x)$ x^2
0 1 1 0 | 1	2	4	0 1 1 0 0	12	144
1 1 0 0 | 0	1	4	1 1 0 0 1	25	625
1 1 | 0 0 0	4	2	1 1 0 1 1	27	729
1 0 | 0 1 1	3	2	1 0 0 0 0	16	256
Sum					1754
Average					439
Max					729

Schema Processing

After Reproduction			After All Operators		
Expected Count	Actual Count	String Representatives	Expected Count	Actual Count	String Representatives
3.20	3	2,3,4	3.20	3	2,3,4
2.18	2	2,3	1.64	2	2,3
1.97	2	2,3	0.0	1	4

Focus on Schemas

Examples:

	P(t)	f(x)	P(t+1)	f(x)
x1:	011010	1	011010	1
x2:	100111	0	011000	1
x3:	110010	0	000110	3
x4:	011000	1	000110	3
x5:	000110	3	000110	3
x6:	000111	1	000111	1
x7:	110110	0	101001	2
x8:	101001	2	101001	2

Selection Rule: The number of children is proportional to a chromosome's relative performance.

What is the effect on the patterns in the population?

Implicit Parallelism

Theorem: The number of representatives from any *schema* S increases in proportion to the observed relative performance of S.

Let S = 0#####

Let N(S,t) be number of elements of S at t.

Then f(S,t) = (1+1+3+1)/4 = 1.5

So, N(S,t+1) = 1.5 * N(S,t)

A large number of schema are processed without explicit computation of utilities.

The Two-Armed and K-Armed Bandit Problem

Why should exponentially increasing samples be given to the observed best building block?

Two-armed bandit problem

Tradeoff between exploitation and exploration

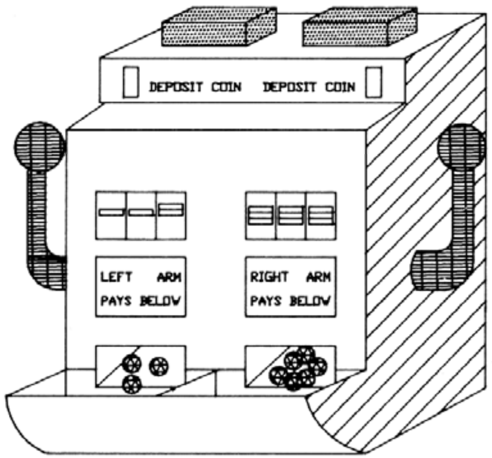

FIGURE 2.1 The two-armed bandit problem poses a dilemma: how do we search for the right answer (exploration) at the same time we use that information (exploitation)?

The Two-Armed and K-Armed Bandit Problem

n experimentations to each arm for total of N trials,

q(n): probability that the worst arm is observed the best after n trials on each arm

Expected loss: $L(N,n) = |\mu_1 - \mu_2| \cdot [(N-n)q(n) + n(1-q(n))]$

The optimal experiment size n^*

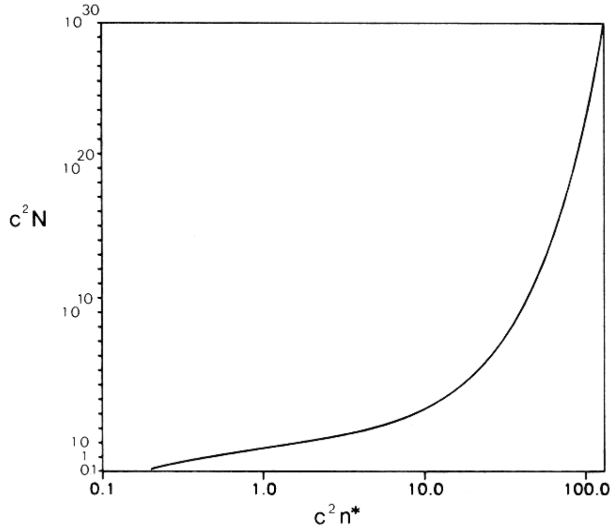

FIGURE 2.2 The modified total number of trials (c^2N) grows at a greater than exponential function of the modified optimal experiment size (c^2n^*) in the one-shot, decision-theory approach to the two-armed bandit problem.

The Two-Armed and K-Armed Bandit Problem

To allocate trials optimally (minimal expected loss), we should give slightly more than exponentially increasing trials to the observed best arm.

We need to allocate exponentially increasing numbers to the observed best schemata.

Building blocks receive exponentially increasing trials in future generations.

The Two-Armed and K-Armed Bandit Problem

GA can be thought of the composition of many k-armed bandits.

A set of eight schemata that competes at three positions in the strings of length seven is eight-armed bandit problem.

With three positions fixed over a string of length seven, there are 35 of the eight-armed bandit problems.

```
* 0 0 * 0 * *
* 0 0 * 1 * *
* 0 1 * 0 * *
* 0 1 * 1 * *
* 1 0 * 0 * *
* 1 0 * 1 * *
* 1 1 * 0 * *
* 1 1 * 1 * *
```

How many schemata are processed usefully?

The number of schemata processed in a string population with length *l* and size *n* is somewhere between 2^l and $n2^l$.

Not all of these schemata are processed with high probability because crossover destroys those with relatively long defining lengths.

What is the lower bound on those schemata that are processed in a useful manner - those that are sampled at the desirable exponentially increasing rate?

The number of schemata is proportional to n^3.

How many schemata are processed usefully?

The number of schemata is proportional to n^3.

- Consider schemata with defining length $l_s < \varepsilon(l-1)+1$.
- Total number schemata of length l_s or less in a particular string: $2^{ls-1}(l-l_s+1)$
- The number of such schemata in the whole population:
  $n2^{ls-1}(l-l_s+1)$
- Pick a population size $n=2^{ls/2}$: one or fewer of all schemata is of order $l_s/2$ or more
- If we count only one half of the schemata that have higher order than $l_s/2$,

$$n_s \geq n(l - l_s + 1)2^{l_s - 2} = (l - l_s + 1)\frac{n^3}{4}$$

How many schemata are processed usefully?

Despite the disruption of long, high-order schemata by crossover and mutation, GAs inherently process a large quantity of schemata while processing a relatively small quantity of strings.

The Building Block Hypothesis

- Implicit Parallelism + Crossover Effect
- Short and low-order and highly fit schemata are sampled, recombined, and resampled to form strings of potentially higher fitness.
- It is claimed that building blocks combine to form better strings. It seems reasonable, but do we have any evidence?
- Walsh-schemata transform: Bethke (1981) and Holland (1987)
  - Given a particular function and coding, building block combine to form optima or near optima.

The Building Block Hypothesis

The five-bit coding example for regularity implied in building block processing

- H_1 = 1**** and H_2 =0**** (See Figure 2.3)
- H_1 = ****1 and H_2 =****0 (See Figure 2.4)
- H_1 =10*** and H_2 =11*** (See Figure 2.6)

The periodicity permits the Walsh function analysis and the analysis determine the expected static performance of GA.

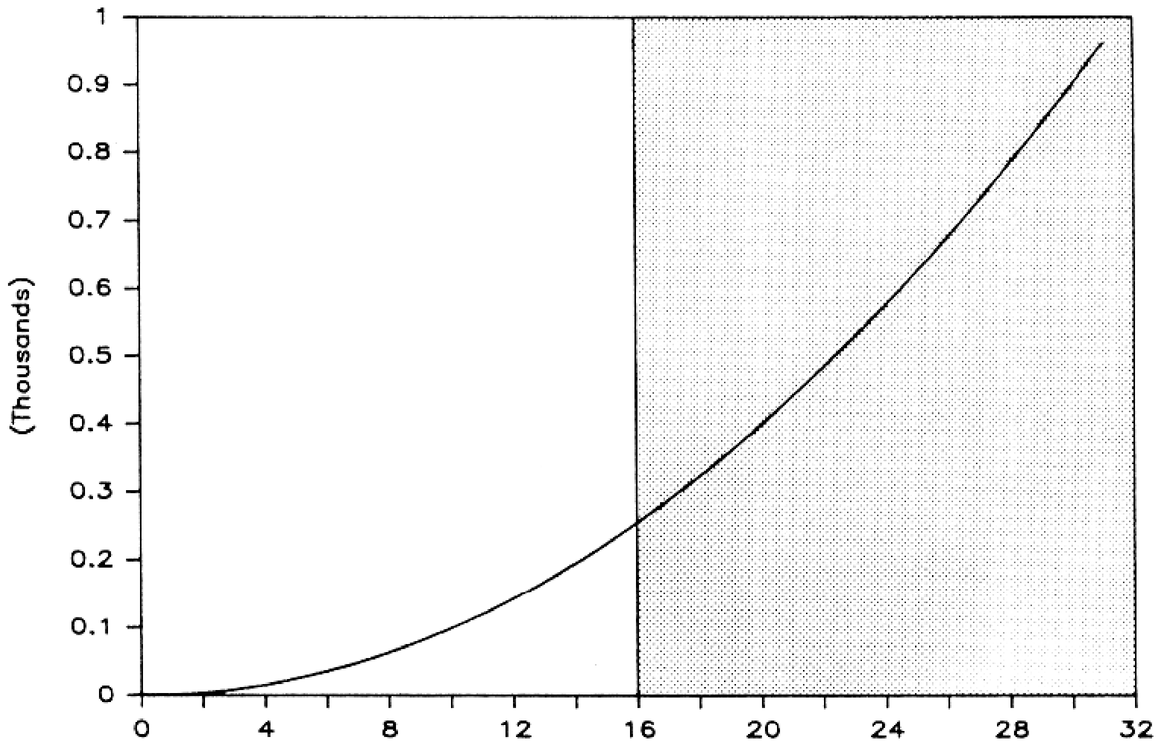

FIGURE 2.3 Sketch of schema 1**** overlaying the function $f(x) = x^2$.

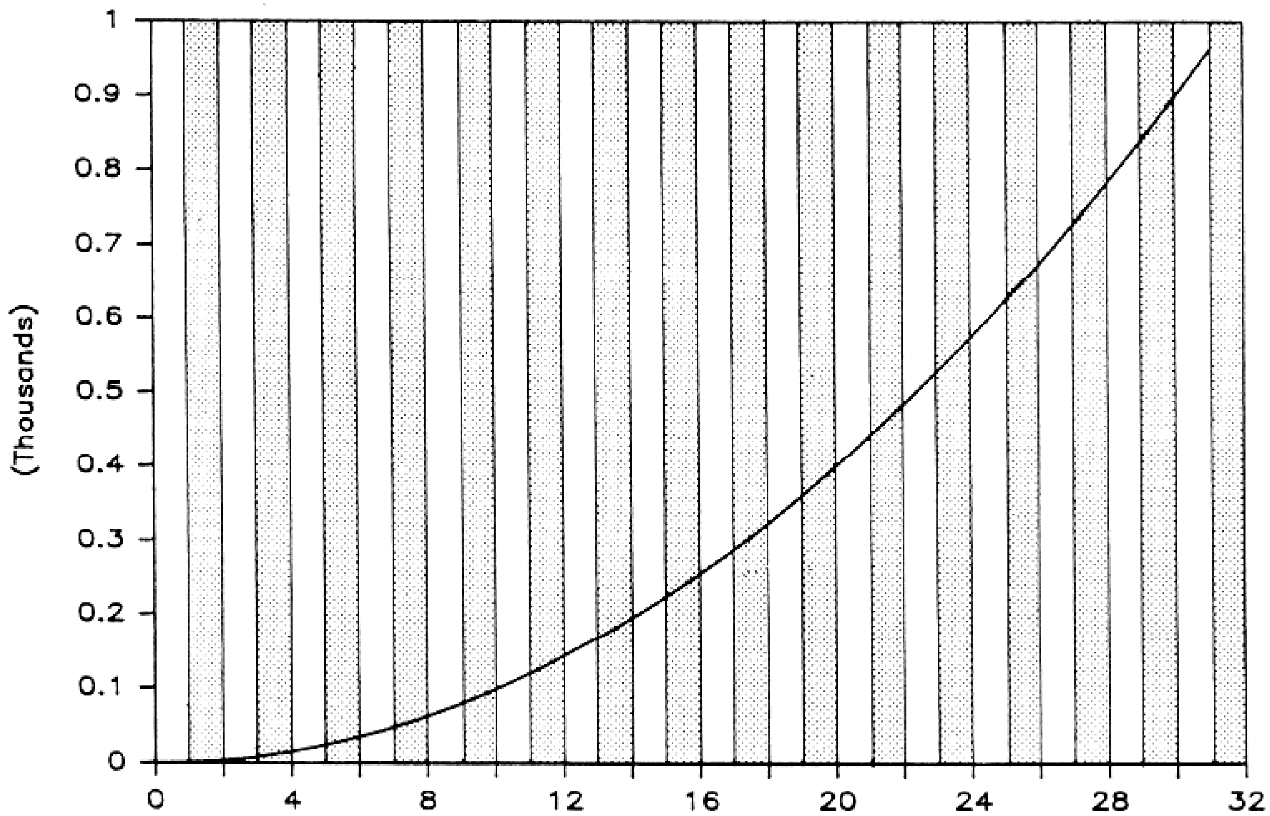

FIGURE 2.4 Sketch of schema ****1.

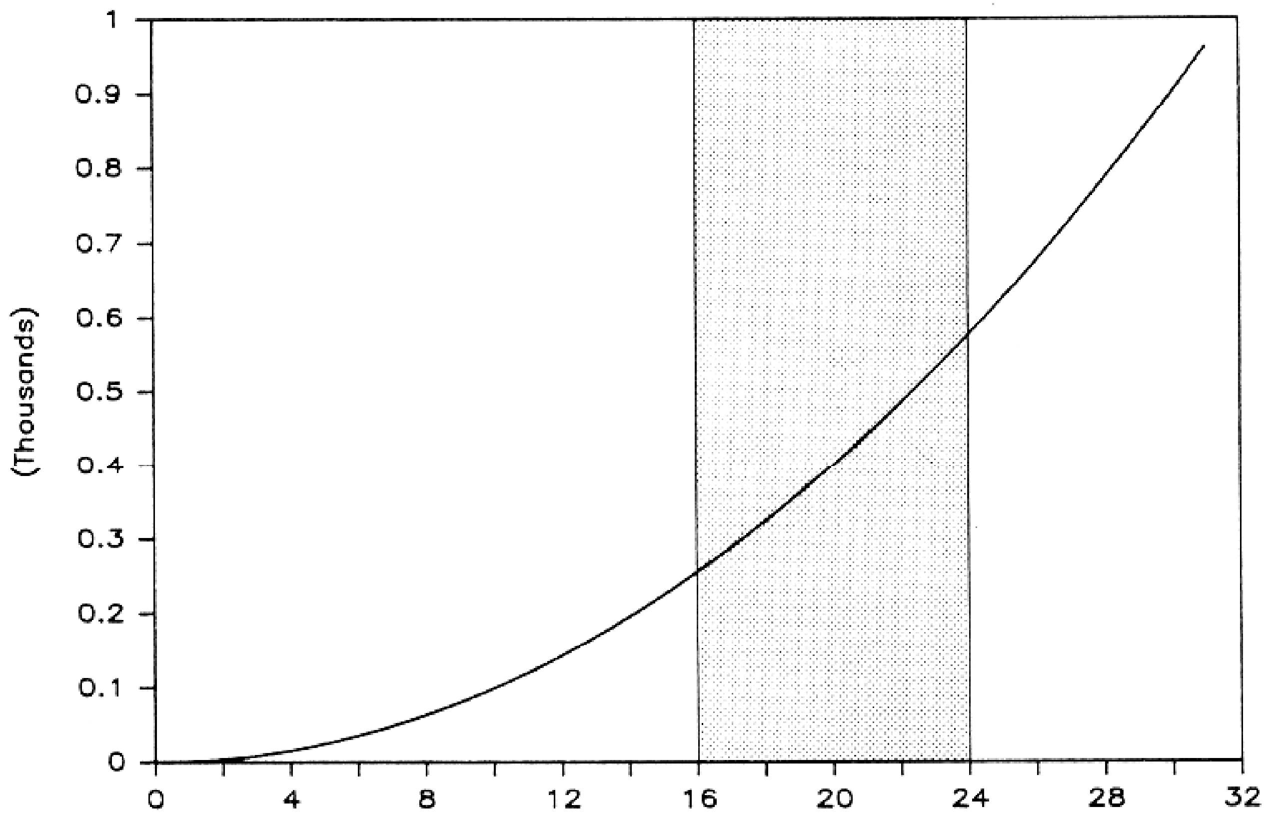

FIGURE 2.6 Sketch of schema 10***.

The Building Block Hypothesis

Generalization of the result to arbitrary codings and functions has proved difficult.

A number of test cases that are provably misleading for the simple three-operator GA: GA-deceptive problems

Simple GA depends upon the recombination of building blocks to seek the best points.

Royal Road Functions by Mitchel, Holland and Forrest.

The Minimal Deceptive Problem

- What makes a problem difficult for a simple GA?
- The simplest problem that causes a GA to diverge from the global optimum.
- The problem that violates the building block hypothesis: the short, low-ordered building blocks lead to incorrect longer, higher order building blocks.
- Despite the effort to fool a simple GA, it is surprising that the GA-deceptive problem is not usually GA-hard (does not usually diverge from the global optimum).

The Minimal Deceptive Problem

Problem Construction: Deceptive two-bit problem

Global condition

$f_{11} > f_{00}$, $f_{11} > f_{01}$, $f_{11} > f_{10}$

Deceptive condition

$f(0*) > f(1*)$ or $f(*0) > f(*1)$

*	*	*	0	*	*	*	*	*	0	*	f_{00}
*	*	*	0	*	*	*	*	*	1	*	f_{01}
*	*	*	1	*	*	*	*	*	0	*	f_{10}
*	*	*	1	*	*	*	*	*	1	*	f_{11}
			|←			$\delta(H)$			→|		

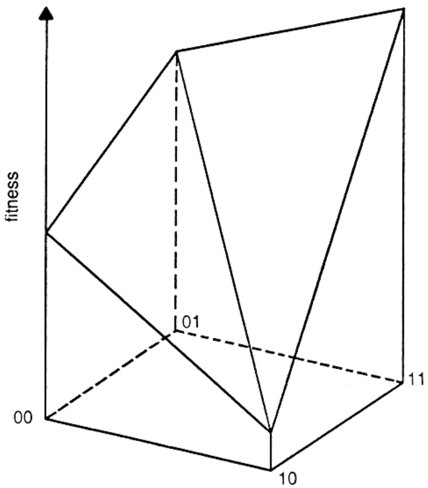

FIGURE 2.8 Sketch of Type I, minimal deceptive problem (MDP) $f_{01} > f_{00}$.

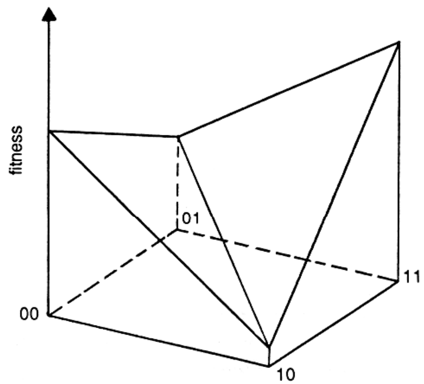

FIGURE 2.9 Sketch of Type II, minimal deceptive problem (MDP) $f_{00} > f_{01}$.

The Minimal Deceptive Problem

Epistasis: Nonlinearity (dominant/recessive gene) among the four points in each case.

cannot be expressed as $f(x_1,x_2)=\Sigma a_i x_i + b$

Deceptive three-bit problem (Appendix E)

Schema Analysis of the Two-bit Problem

- In the two-bit problem, a schema is not lost even if a crossover occurs between the schema's outermost defining bits.
- See Table 2.2.
- Computation of the expected proportion P of each of the four competing schemata.
- MDP result
  - The Type I MDP is not GA-hard (See Figure 2.10).
  - The Type II MDP converge to the best solution for most starting conditions (See Figures 2.11, 2.12).

TABLE 2.2 Crossover Yield Table in Two-Bit Problem

X	00	01	10	11
00	S	S	S	01 10
01	S	S	00 11	S
10	S	00 11	S	S
11	01 10	S	S	S

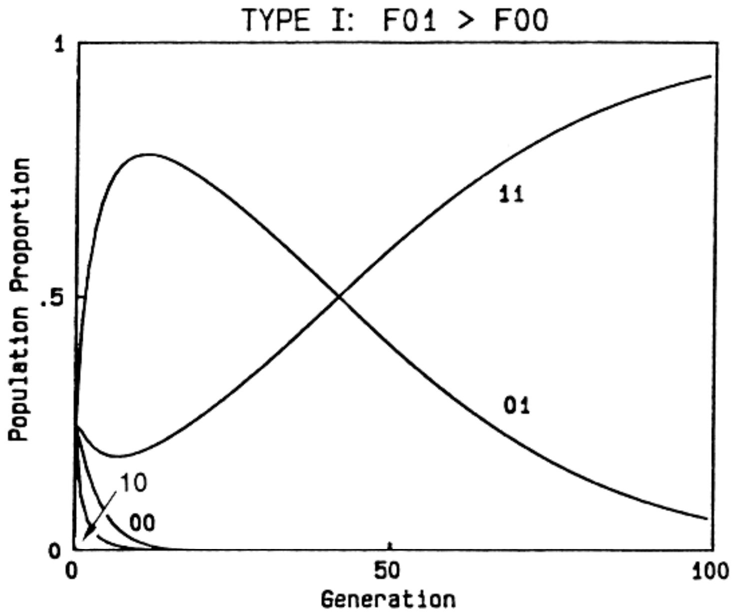

FIGURE 2.10 Numerical solution of a Type I, minimal deceptive problem (MDP): $r = 1.1, c = 1.05, c' = 0.0$.

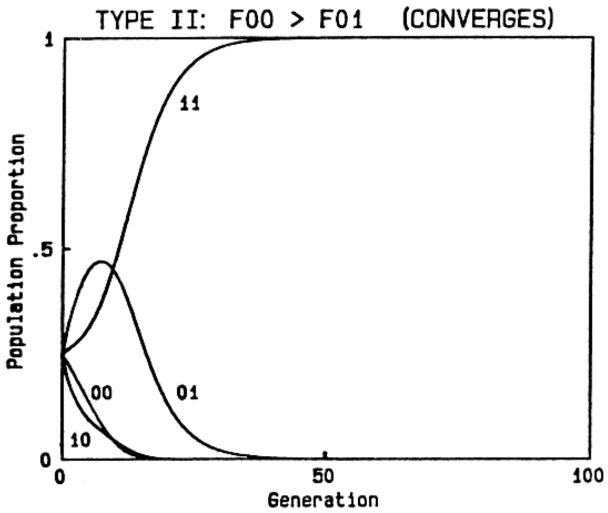

FIGURE 2.11 Numerical solution of a Type II, minimal deceptive problem that converges: $r = 1.1$, $c = 0.9$, $c' = 0.5$ with equal initial proportions.

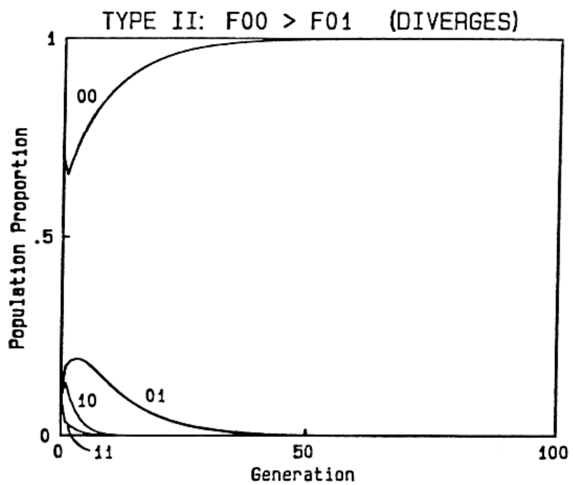

FIGURE 2.12 Numerical solution of a Type II, minimal deceptive problem that diverges: $r = 1.1$, $c = 0.9$, $c' = 0.5$ with unequal initial proportions.

Schemata as Hyperplanes

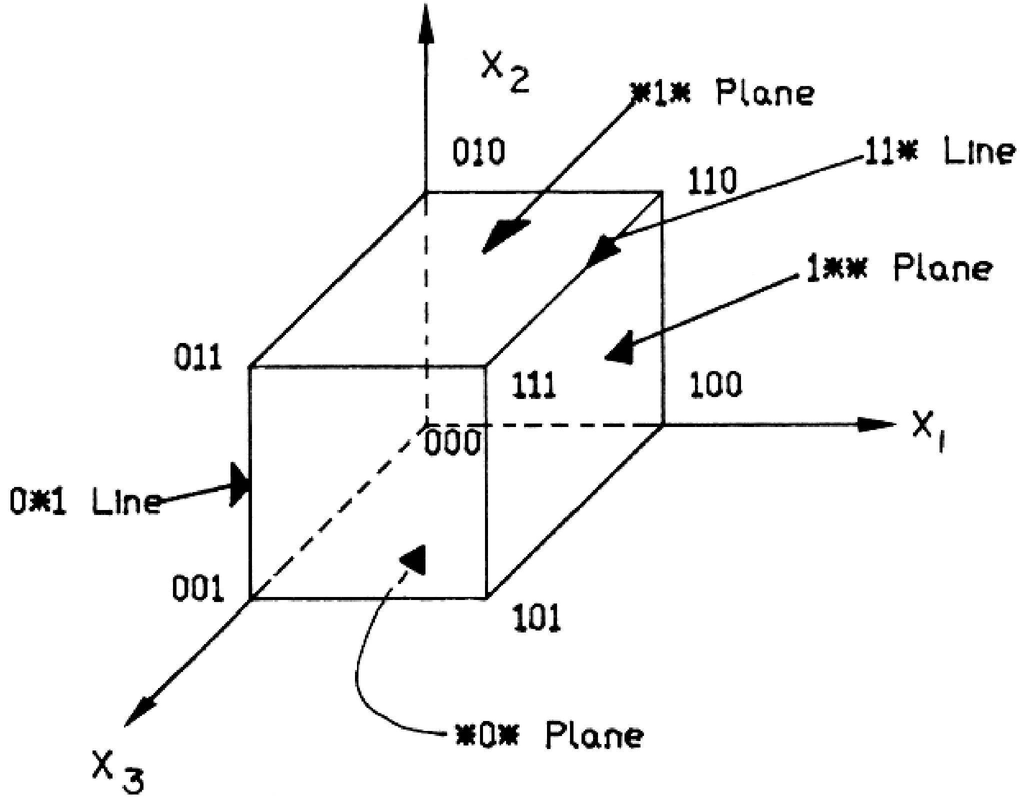

FIGURE 2.13 Visualization of schemata as hyperplanes in three-dimensional space.