The United Nations
University

SMR/748 - 9

ICTP-INFN-UNU-MICROPROCESSOR LABORATORY
THIRD COURSE ON BASIC VLSI DESIGN TECHNIQUES
21 November - 16 December 1994

GENETIC ALGORITHMS: AN OVERVIEW

Silvio TURRINI
Digital Equipment Corporation
Western Research Lab
Palo Alto, CA 94301
U.S.A.

These are preliminary lecture notes, intended only for distribution to participants.

Genetic Algorithms: an Overview

Silvio Turrini

**Digital Equipment Corporation
Western Research Lab
Palo Alto, California**

digital Western Research Laboratory

1

What are Genetic Algorithms ?

Search procedures based on natural selection and genetics (Darwinian approach).

How many variations ?

Evolution Strategies

On line adaptation using one basic operator (mutation).

[German school '60 - Rechenberg & Schwefel].

Genetic Algorithms (pure genetic inspiration)

Fixed population of fixed length binary strings (k-ary , floating point, permutation like, Lisp codes) and three basic operators : selection, crossover and mutation.

[American school '60 - Holland, De Jong, Goldberg].

2

Genetic or Evolutionary Programming (Lisp programming environment)

Instead of string of bits, S-expressions are used as "genes" of the chromosomes (Lisp programs).

Paradigm :Population of programs, through the three basic operators, generate better and better programs.

['90 - Koza, Stanford University]

Hybrid Genetic Algorithm (Hybridization of pure genetic algorithms with other algorithm)

Genetic operators and the problem representation are taylored for a "specialized" algorithm.

[American school '80 - Davis and others]

and.....many, many others.

How are Gas different from other methods ?

1. GAs work with a coding of parameter set, not the parameters themselves
2. GAs search from a population of points, not a single point.
3. GAs use payoff (objective function) information, not derivative or other auxiliary knowledge.
4. GAs use probabilistic transition rules, not deterministic ones.

What are they used for ?

1. Search and Optimization.
2. Decision making.
3. Classification.
4. Genetic learning (Classifiers)

Learning the Lingo

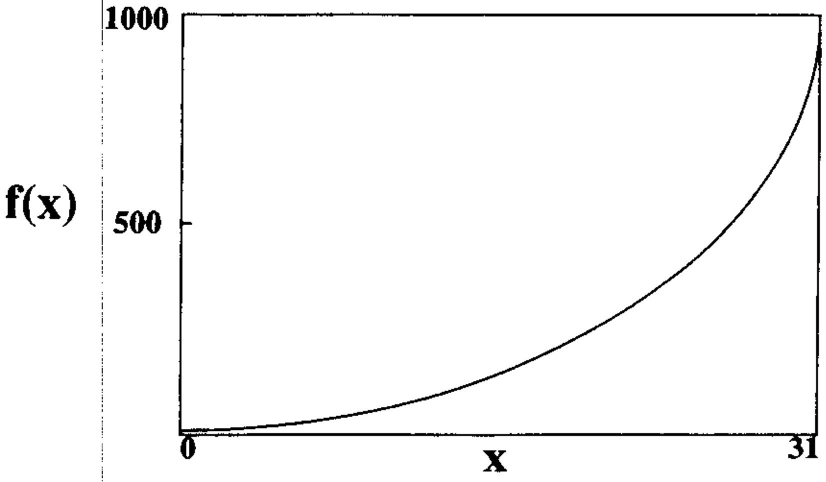
Comparison Natural and GA Terminology

chromosome	string
gene	feature, detector
allele	feature value
locus	string position
genotype	structure
phenotype	decode structure
epistasis	nonlinearity
breeding techniques	selection mechanisms
introns	non functional patches

Ideas and analogies might either come from nature or the researcher.

A simple optimization problem

Maximize the function : $f(x) = x^2$ on the integer interval [0, 31].



Initial population (4 strings) :

1)	0	1	1	0	1	3)	0	1	0	0	0
2)	1	1	0	0	0	4)	1	0	0	1	1

Let's apply the basic three operators :

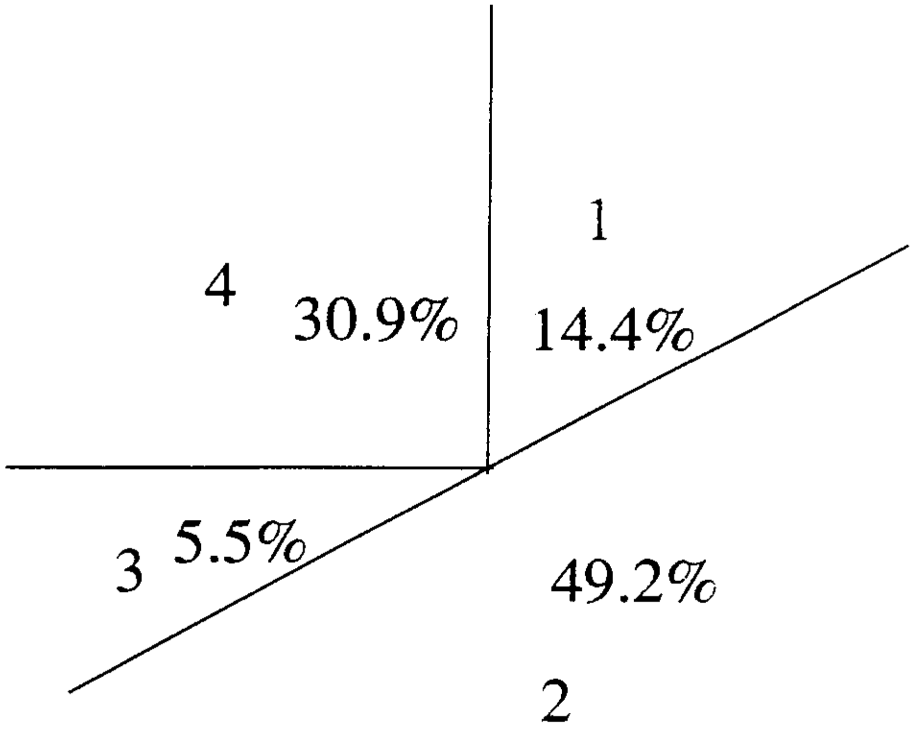
1. Reproduction
2. Crossover
3. Mutation

6

Reproduction

Selection scheme : Roulette wheel selection

Rule : Roulette slots sized in proportion to the correspondent string fitness.



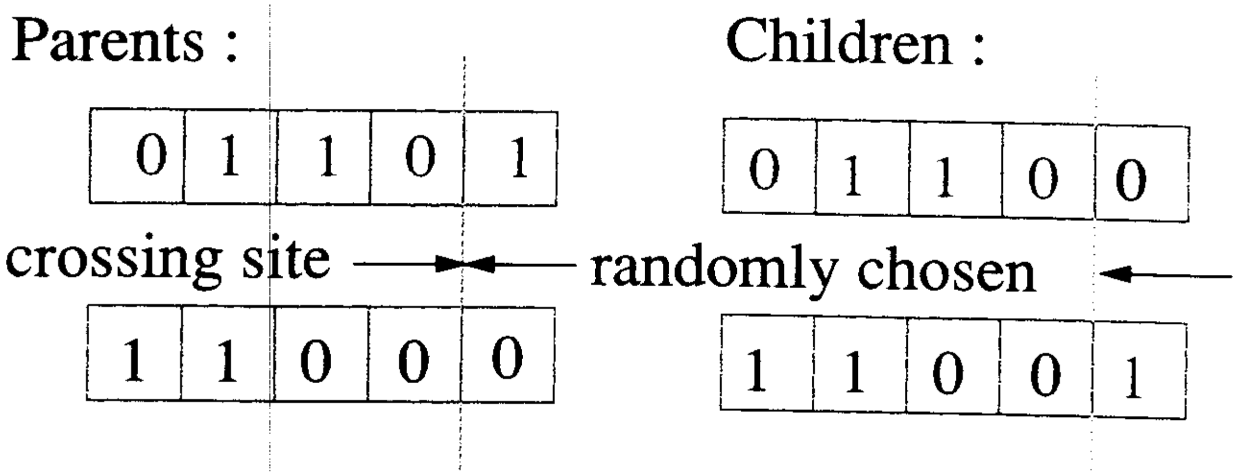
7

Table with Strings and Fitness Values

Optimization of function f(x)

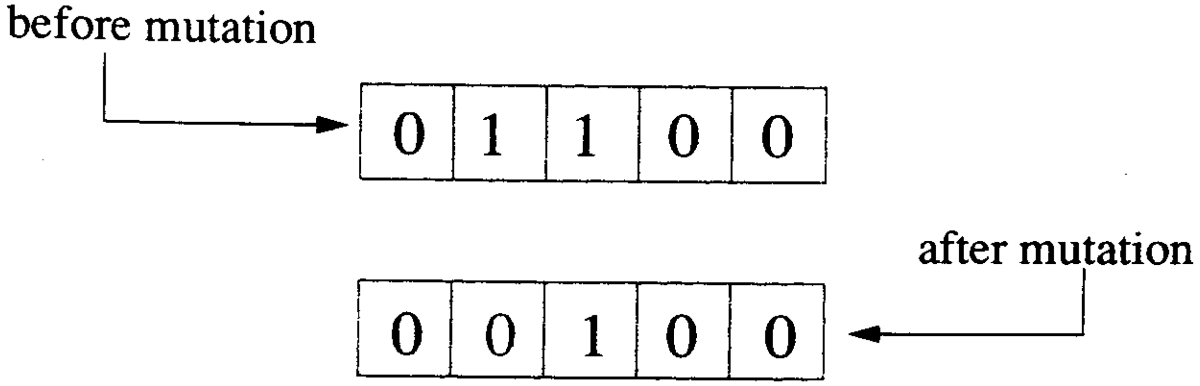
Chrom. #	String	Fitness	% of Total
1	01101	169	14.4
2	11000	576	49.2
3	01000	64	5.5
4	10011	361	30.9
Total		**1170**	**100.0**

Crossover (one point crossover)



Mutation

Mutation simply flips with low probability some bits when copying during crossover.



before mutation

0	1	1	0	0

after mutation

0	0	1	0	0

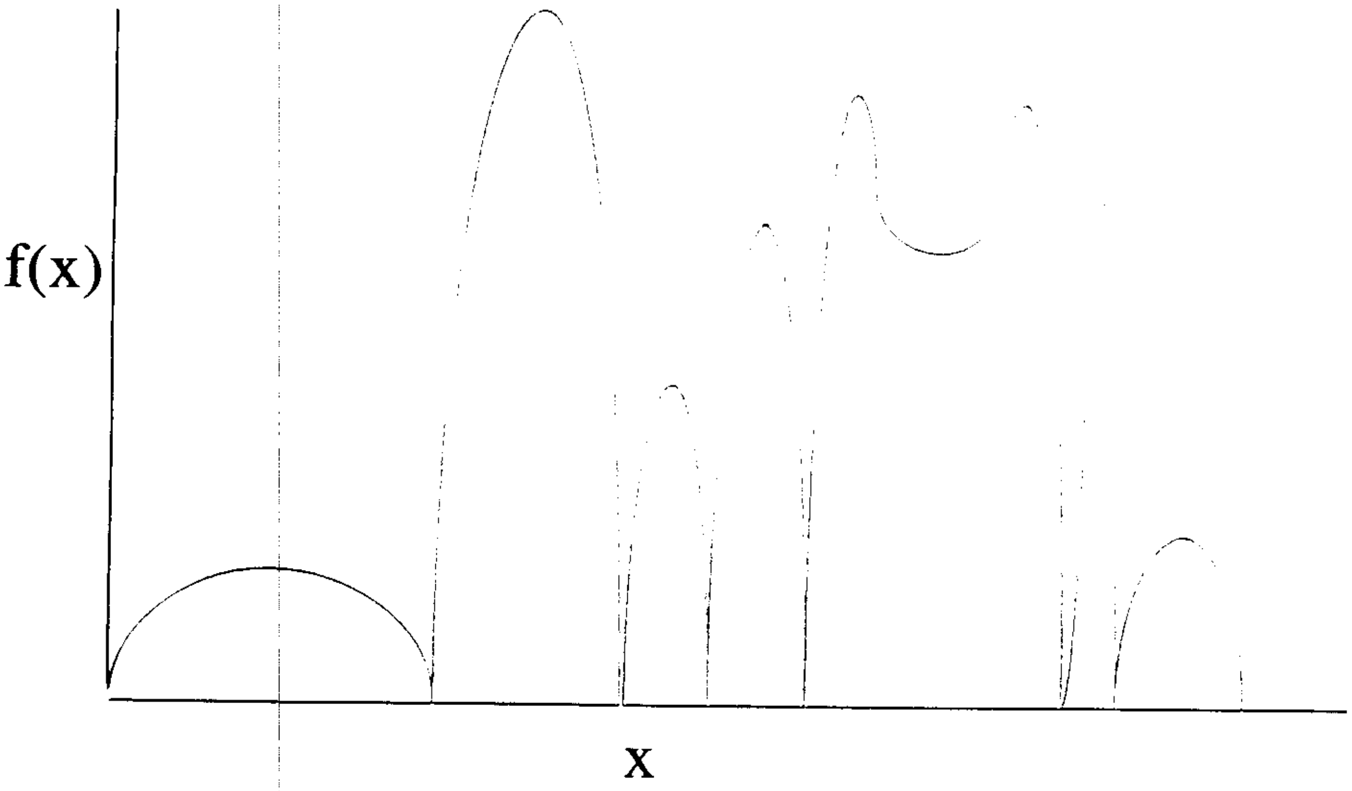
Starting with a random population of strings,after few generations the population converges to the optimal value.

GAs wouldn't be so interesting if "real" functions were so easy.

GAs can find global optima in difficult (non derivable, for instance) functions with billions of local optima.

Optimization of "difficult" functions

How about objective functions like this ?



In 1975 De Jong studied the problem of optimization using GAs. He came up with new performance estimators and a set of optimal parameter values for the various operators.

The standard De Jong's test suite

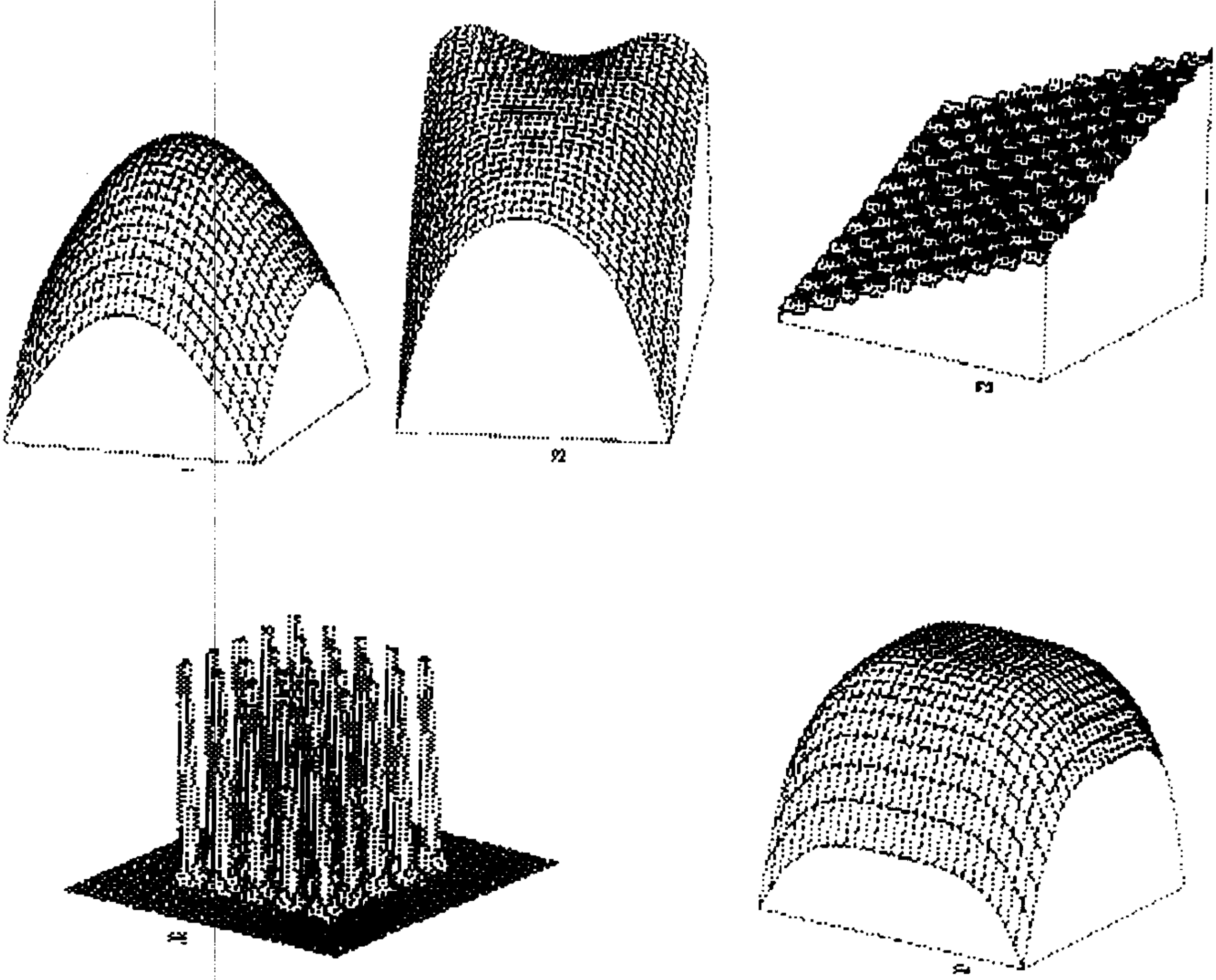
Test environment of five problems in function minimization with the following characteristics:

1. Continuous/discontinuous
2. Convex/nonconvex
3. Unimodal/multimodal
4. Quadratic/nonquadratic
5. Low-dimensionality/high-dimensionality
6. Deterministic/stochastic

Different selection schemes, crossover and mutation operators with optimal probabilities have been tested and optimized over a a large number of runs and population sizes.

Later,GAs have been used as parameter optimizers of other GAs (meta-genetic algorithms)

De Jong's test suite



The Fundamental Theorem

Assumptions :

1. Strings to be constructed over a binary **alphabet** V = {0, 1} and binary decoding is assumed.

2. A schema is a template defined over the **alphabet** {0, 1, *} describing a pattern of bit strings in the search space $\{0, 1\}^l$ where l = length of the string and * specifies a "don't care" allele.

Example of strings and schemata

String	Representation
A	100111
B	010011
schema 1	**0*11
schema 2	**0***
schema 3	****11

(13)

Properties and Definitions

1. There are 3^l schemata or "**similarities**" **over a** binary string of length l.
2. Order of a schema H, o(H) = number of fixed positions in the schema
3. Defining length of H, $\delta(H)$ = distance between the first and last specific string position.
4. Notation m = m(H, t) means : at time t, there are m copies of schema H in the population.
5. During reproduction a string get selected and is copied according to its fitness with probability :

$$pi = f_i/\Sigma f_j$$

Ex :

schema = 0 1 1 * 1 * *

$o(H) = 4,\ \delta(H) = 5 - 1 = 4$

(14)

After picking a nonoverlapping population of size n with replacement, we expect to have m(H, t+1) representative of schema H at time t+1 as given by the equation :

$$m(H, t+1) = m(H, t) \times n \times f(H) / \sum f_i$$

where $f(H)$ is the average fitness of the strings representing scheme H at time t.

But the average fitness of the entire population can be written as : $\bar{f} = \sum f_i / n$ therefore :

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

Above-average schemata grow, below-average schemata die off.

Suppose a schema H remains above average an amount of time $c \times \bar{f}$ with c a constant.

The schema difference equation can be rewritten as :

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

Starting at t = 0 and assuming a stationary value of c we get the equation :

$$m(H,t) = m(H,0) \times (1+c)^{t}$$

or a geomeric progression, the discrete analog of an exponential form.

How crossover affects schemata ?

S_1 = 0 1 1 | 1 0 0 0

H_1= * 1 * | * * * 0

H_2= * * * | 1 0 * *

crossover site (randomly chosen)

In the previous example crossover destroys schema H1 while schema H2 will survive.

The longer the schema the more likely is its disruption by crossover.

Survival probability under simple crossover :

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

If we introduce crossover and mutation with the respective probabilities (p_c and p_m) and similar consideration, for every scheme H we get :

$$m(H, t+1) \le m(H, t) \times \frac{f(H)}{\bar{f}} \left[1 - \left(p_c \cdot \frac{\delta(H)}{l-1} \right) - o(H) \times p_m \right]$$

Conclusion

Short, low-order, above-average schemata receive exponentially increasing trials in subsequent generations.

Implicit parallelism

If we consider a population of n binary strings of length l ,a lower bound on the number of schemata processed at each generation is $O(n^3)$

The Building Block Hypothesis

Highly fit schemata of low defining length are sampled and recombined to form better and better strings if :

1. They are consistently emphasized by selection.
2. They are respected and exchanged by the genetic operators.

The building block hypothesis suggests an analogy with Fourier Analysis, in which an arbitrary curve (periodic) can be approximated by the sum of sines and cosines of progressively higher frequencies (Walsh function analysis with GAs).

The Minimal Deceptive Problem

Suppose we have :

0*******0* ---> f_{00}

0*******1* ---> f_{01}

1*******0* ---> f_{10}

1*******1* ---> f_{11}

<--- $\delta(H)$ --->

Let's assume f_{11} is the global optimum:

$(f_{11} > f_{00})$ $(f_{11} > f_{01})$ $(f_{11} > f_{10})$

Element of deception :

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

These "low-order" schemata lead the GAs away from "high-order" schemata. Today deception is one of the major areasof research in GAs.

digital Western Research Laboratory

Is it really that simple ?

Simple answer : NO

For selection there are three main categories of schemes: stabilizing , directional, disruptive and all of them can have scaling.

(A few) alternative selection schemes for the first category:

1. Deterministic sampling
2. Remainder stochastic sampling with/without replacement
3. Stochastic sampling with/without replacement
4. Stochastic tournament
5. Competitive selection
6. Bipartite competition
7. Tournament fitness selection

Advanced Operators :

1. Dominance
2. Diploidy
3. Abeyance
4. Segregation
5. Translocation
6. Multiple chromosome structures.
7. Duplication and deletion
8. Sexual determination and differentiation

Why do we need so many different strategies?

One simple reason is that we have finite (and usually small) populations and we want a fast answer.

Premature convergence and diversity

GAs with small populations and conventional reproduction schemes suffer from premature convergence (all members are the same).

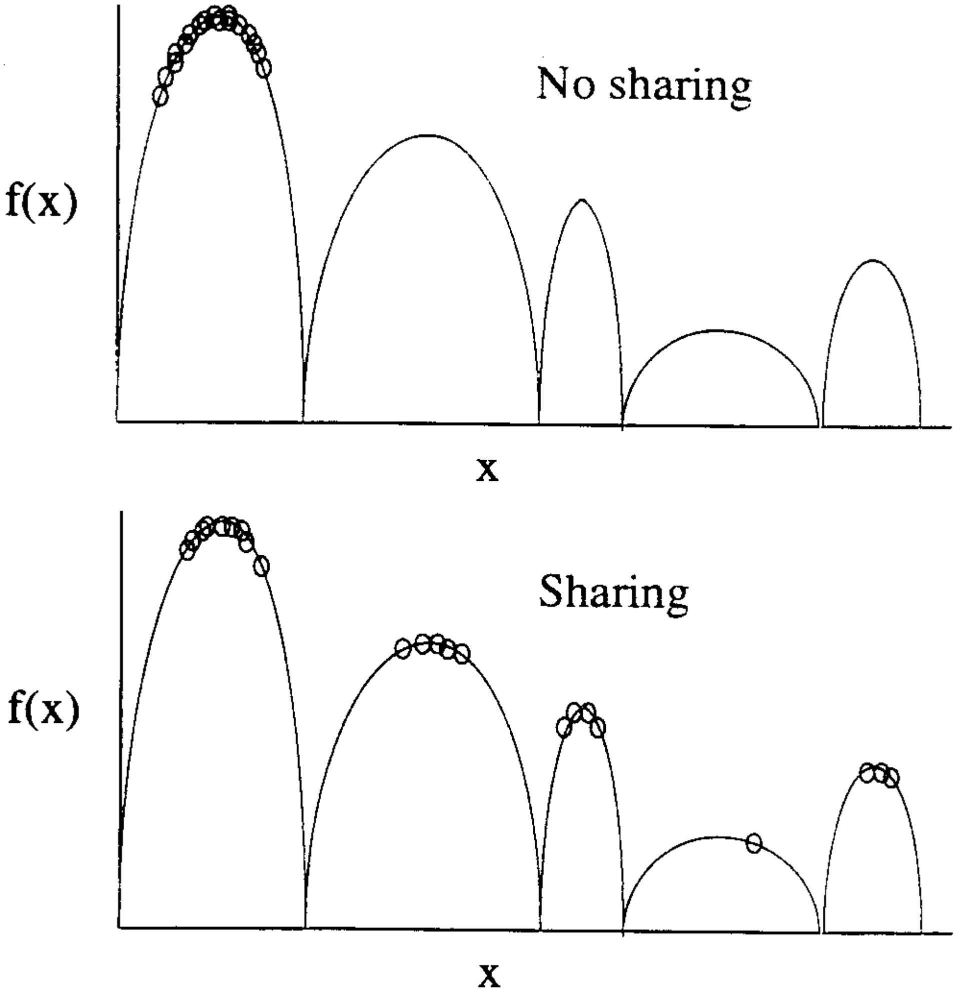
The basic idea is to let fitness increase, but also keep diversity in the population.

Reproduction schemes :

1. Generational replacement.
2. Steady-state reproduction.
3. Elitist schemes.
4. CHC (no mutation is used, but after the population converged, it gets "restarted".
5. Restriction on mating and sharing.

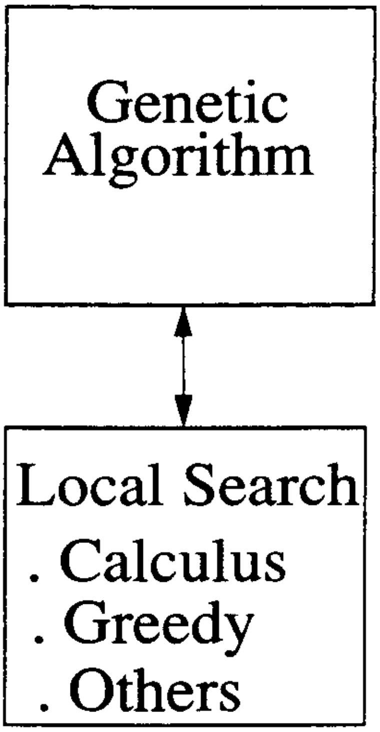
One solution : Niche and Speciation

In sharing, fitness is scaled depending on the neighbourhood and how close individuals are between each other.



Hybrid schemes

Why don't we separate global (genetic) from local (gradient-like, hillclimbers, etc.) optimization ?



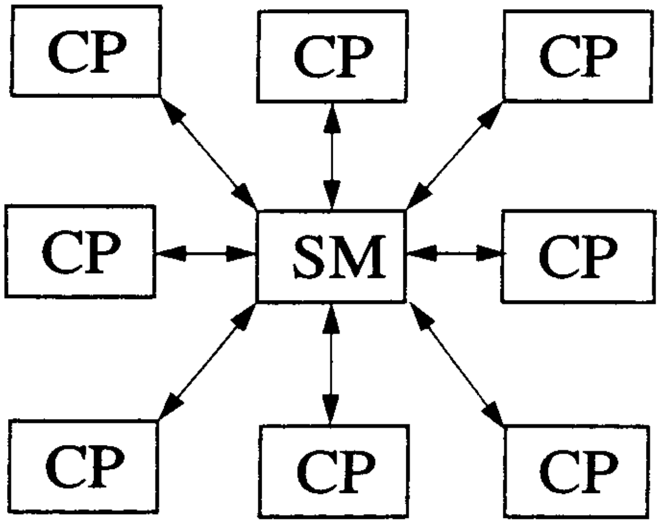
Genetic Algorithm

Local Search
. Calculus
. Greedy
. Others

Parallel Genetic Algorithms

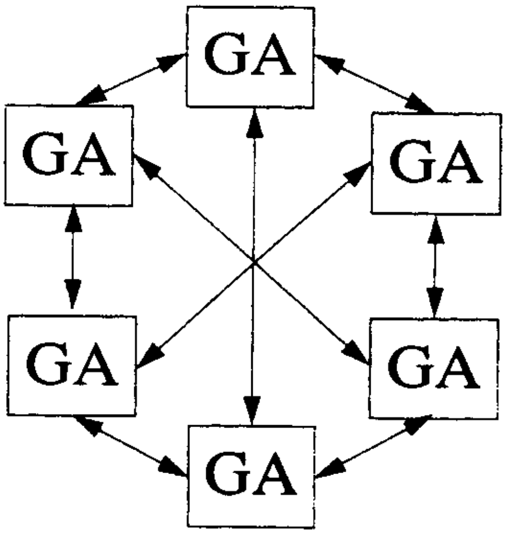
Grefenstette (1981) examined :

1. Syncronous master-slave
2. Semisyncronous master-slave
2. Distributed asyncronous concurrent
3. Network



CP = concurrent process

SM = shared memory



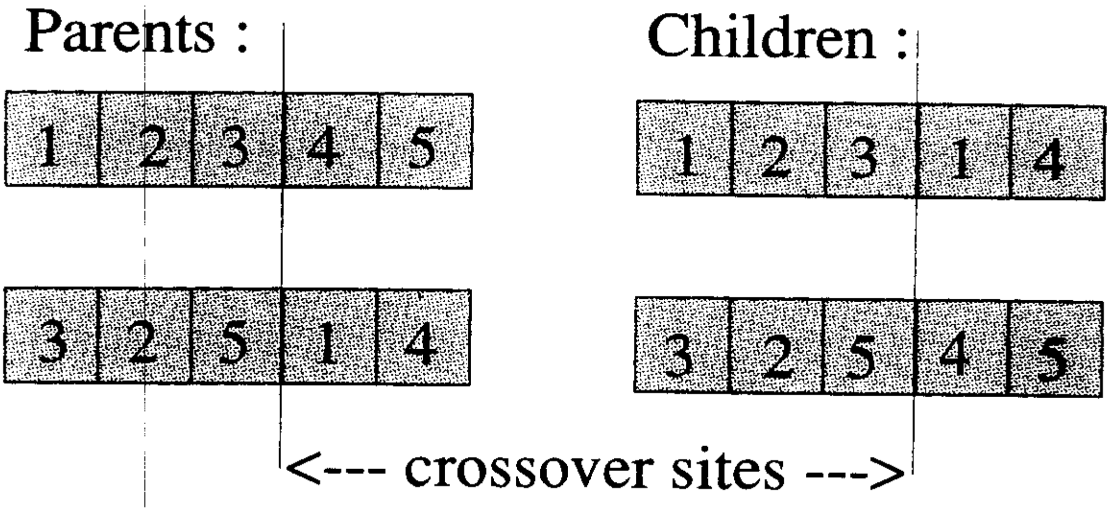
Network GA

Reordering operators

What about problems which involve finding an optimal order for a sequence of N items ?

Instead of a Cartesian space we have to deal with a Permutation space (problem of representation) and crossover doesn't work any more.

Example :



Parents :

1 2 3 4 5

3 2 5 1 4

Children :

1 2 3 1 4

3 2 5 4 5

<--- crossover sites --->

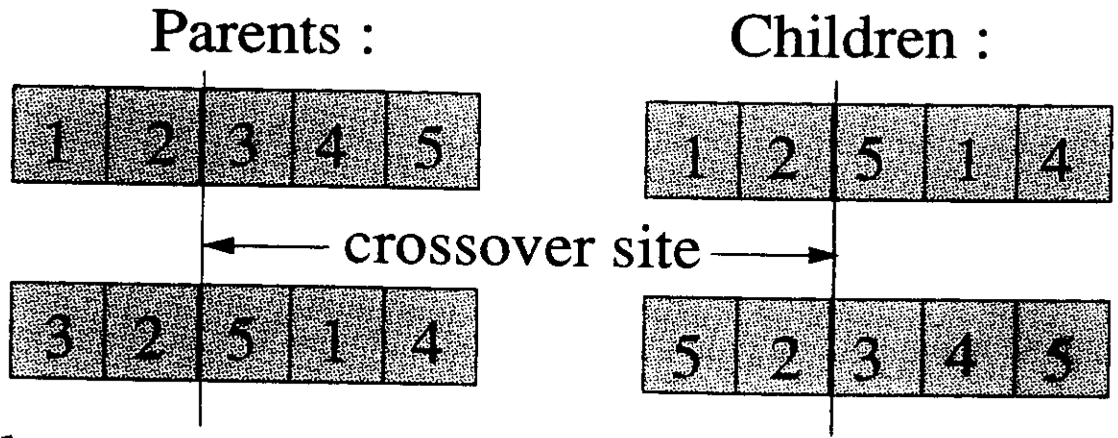
generates unacceptable permutations.Two basic solutions : 1) new operators, 2) penalty function

Reordering Operators

Crossover :

1. Partial Matched Crossover (PMX)
2. Order Crossover (OX)
3. Cycle Crossover (CX)
4. Uniform Order Crossover (UOX)
5. Inversion and, many, many others....

Let's see one of them, PMX :



Parents :

1 2 | 3 4 5

3 2 | 5 1 4

Children :

1 2 | 5 1 4

5 2 | 3 4 5

← crossover site →

Exchange (swap) : 3 <--> 5, 4 <--> 1, 5 <--> 4

The penalty function method cannot directly be applied to the space of permutations.

From Stirling approximation :

$$n! \sim \sqrt{2\Pi n} \times \left(\frac{n}{e}\right)^{n}$$

so the probability of a “good”permutation if we have n independent variables is :

$$P_{p} = \frac{n!}{n^{n}} \sim \frac{\sqrt{2\Pi n}}{e^{n}}$$

For the interesting cases (n > 1000), we will never get a single good permutation! There are other representations, but decoding is computationally very expensive.