united nations educational, scientific and cultural organization

international atomic energy agency

the
abdus salam
international centre for theoretical physics

SMR/1220-31

Workshop on

Nuclear Reaction Data and Nuclear Reactors: Physics, Design and Safety

13 March - 14 April 2000

Miramare - Trieste, Italy

Genetic Algorithms:

Theory and Applications in the Safety Domain

M. Marseguerra
Polytechnic of Milan
Italy

GENETIC ALGORITHMS: THEORY AND APPLICATIONS IN THE SAFETY DOMAIN

M. MARSEGUERRA AND E. ZIO

Dept. of Nuclear Engineering, Polytechnic of Milan, Via Ponzio 34/3

20133 Milan – ITALY

E-mail: marzio.marseguerra@polimi.it

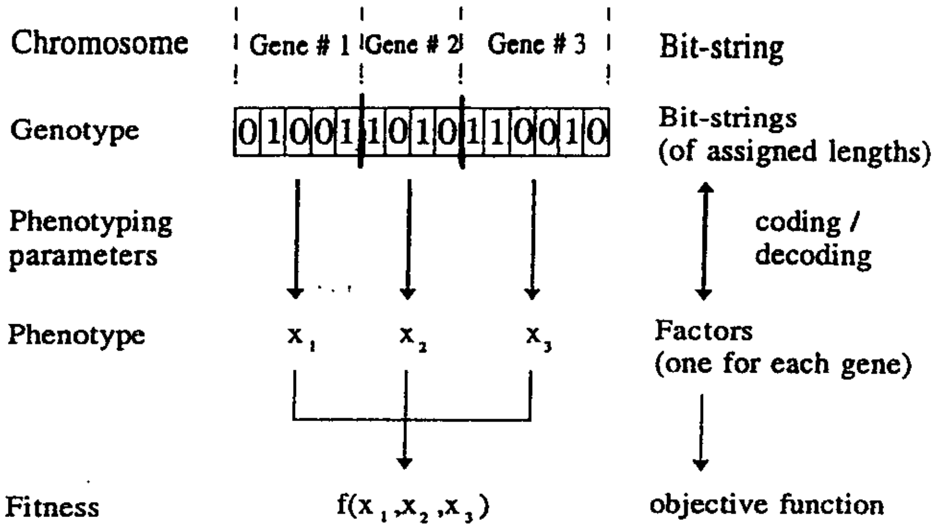
GENETIC ALGORITHMS (GA)

What are they?

Numerical search tools inspired by the rules of the natural selection.

Purpose:

Find the maximum (minimum) of a given real objective function of $n \geq 1$ real variables and subject to various linear or non linear constraints.

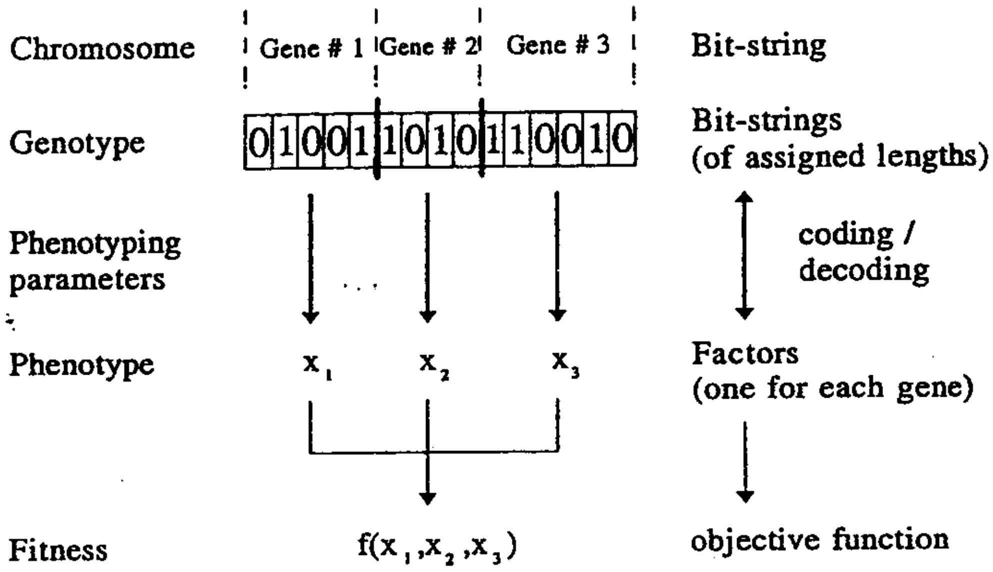


TERMINOLOGY (borrowed from biology)

The GAs operate on a population of N chromosomes.

- chromosome: string of binary digits, partitioned in n substrings called genes, one for each variable of the objective function.
- gene: substring of a chromosome. To the i–th substring ($i = 1, 2, ..., n$) the user assigns n_i digits.
- control factor: real number resulting from decoding the digits of a gene. Each chromosome has as many genes (or control factors) as variables in the objective function.
- fitness of a chromosome: value of the objective function in correspondence of the control factors of the chromosome.

EACH CHROMOSOME GIVES RISE TO A TRIAL SOLUTION TO THE PROBLEM



- Consider an objective function $f(x_1, x_2, ..., x_n)$, of $n \geq 1$ variables and assume that the range of each variable is assigned, viz. $x_i \in (a_i, b_i)$.
- A population is a collection of N individuals (N is assigned by the user).
- Each individual is a chromosome constituted by n genes, the i–th of them made up by n_i bits. The relation between $x \in (a_i, b_i)$ and its binary counterpart β is

$$x = a_i + \beta \, \frac{b_i - a_i}{2^{n_i}}$$

a_i, b_i, n_i are called the phenotyping parameters of the gene.

Modus operandi of a GA search

1. Generate a population of N chromosomes by random sampling all the bits of the N individuals.

2. Manipulate the chromosomes (the strings) of the old population (the parents) to get a new population (the children) hopefully characterized by an increased mean fitness.

3. Repeat step 2 until some convergence criterion is reached.

STEP #1. GENERATE THE INITIAL POPULATION

In our experience we have found very important to distinguish between the following cases:

- **Random sampling.**
  If the various variables of the objective function have independent ranges, then the bits of each chromosomes may be generated at random.

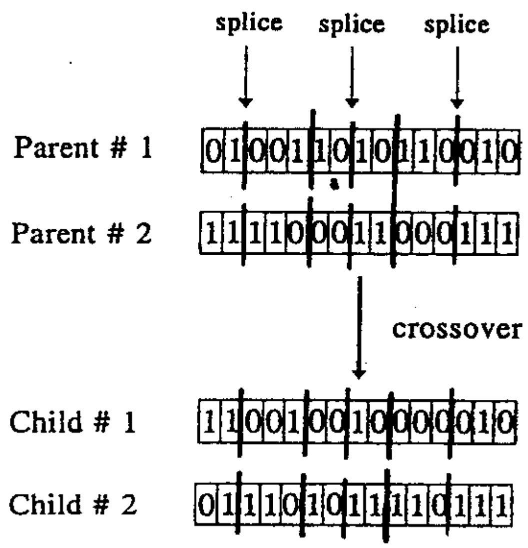
- **Conditional sampling.**
  If the range of a variable is conditioned on the value of other variables, a possible recipe is:
  - sample x_1 at random within its nominal range (a_1, b_1);
  - determine the reduced range of x_2, i.e. the range of x_2 conditioned on the value selected for x_1;
  - sample x_2 at random within its reduced range;
  - repeat sequentially the above two steps, so that x_i is sampled at random within its reduced range, as conditioned on the values selected for $x_1, x_2, ..., x_{i-1}$.

Of course, this procedure heavily rests on the (arbitrary) ordering of the variables. This ordering must be wisely selected on physical grounds.

STEP #2. MANIPULATE THE CHROMOSOMES (FROM PARENTS TO CHILDREN)

2.1 Generate a temporary new population by means of:
Standard Roulette Selection (cumulative sum of fitnesses)
Hybrid Roulette Selection
Random Selection
Fit – Fit
Fit – Weak

2.2 Mate two parents and generate two children
One–site crossover
Two–sites crossover



2.3 Replacement of two among the four individuals 2–parents + 2–children by means of:
Children live, parents die
Fittest individuals
Weakest individuals
Random replacement

2.4 Mutation: among all the bits of the population a random number of them is switched with an assigned mutation probability ($\sim 10^{-3}$)

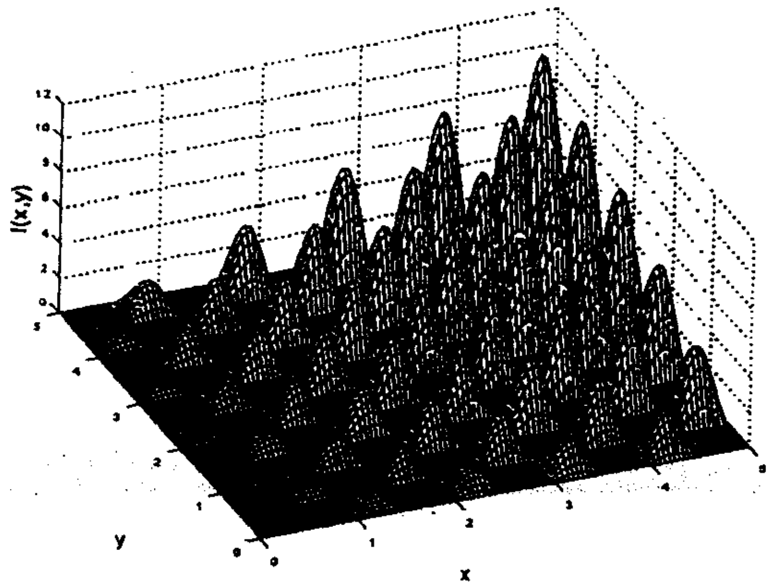
STEP #3. CONVERGENCE CRITERIA

They are usually based on the relative increase of a fitness feature between two successive generation.
The fitness feature may be

- mean fitness
- median fitness
- best fitness

The calculation is also stopped when the assigned number of population generations is reached.

Example : Search for the maxima of a multimodal function



$$z = f(x,y) = \begin{cases} \sqrt{x}\cdot \sin(2\pi x)\cdot\sqrt{y}\cdot \sin(2\pi y) & \text{if } \sin(2\pi x)\cdot \sin(2\pi y) > 0 \\ 0 & \text{otherwise} \end{cases}$$

Number of local maxima = 50

2^{nd} higher maximum / global maximum = 0.889

Standard GA procedure

In correspondence of different sets of parameters

↓

different maxima

(Even though the most frequent is the global maximum)

Possible explanation

Poor genetic diversity

↓

Inducement of species and niches

INDUCEMENT OF SPECIES AND NICHES

- **Biology:**

  Species: class of individuals sharing common features.

  Niche: a set of functions performed by the individuals of a species.

  Environment: the collection of the external conditions, including the interactions with the other species.

  In our species the individuals are partitioned in groups (places, activities, ...). This partitioning turned out to be of utmost advantage for the survival and development of the species.

- **GENETIC ALGORITHMS**

  The population is divided in sub–populations with scarce mutual interactions.

  i) Isolation by distance

  Each subpopulation lives in an island

  The various islands evolve almost separately

  Emigrants: selection of emigrants
  Replacement in the new island

  ii) Spatial Mating

  Definition of a deme

  - individuals disposed on a planar grid: square demes
  - individuals disposed on a circular wheel: linear demes
  - some cases with stochastic deme boundaries

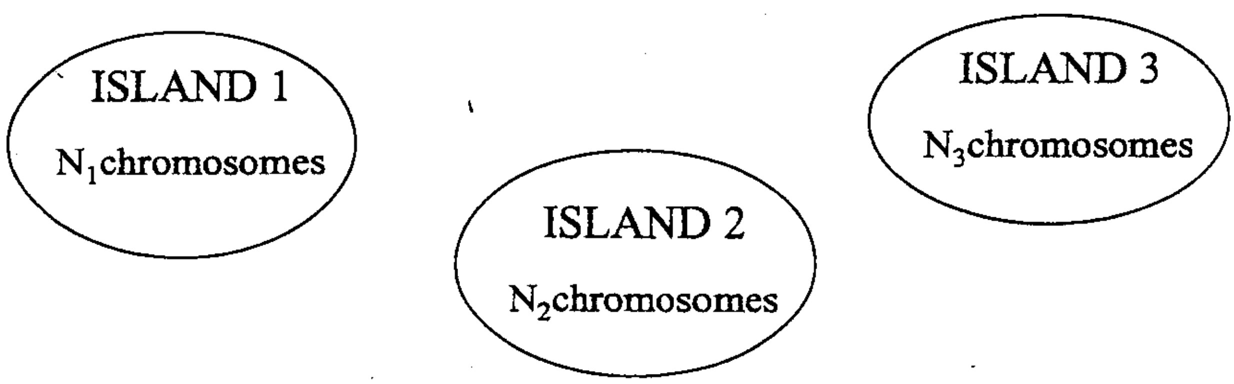
NICHES CREATION

Objective : to favour genetic diversity so as to increase the probability of finding a 'good' maximum

Method 1: GENETIC ALGORITHMS WITH ISLANDS

DISTANCE INDUCED ISOLATION :

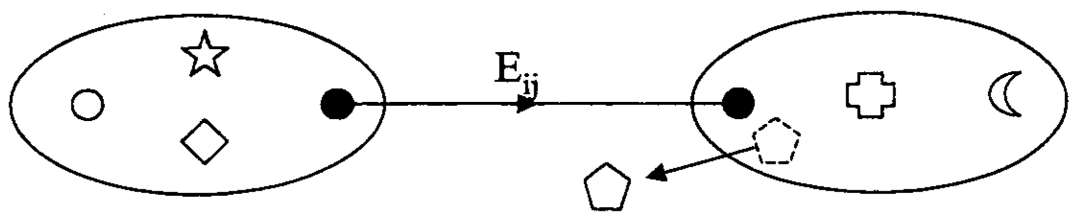
the population is subdivided into islands whose interactions are limited → each island can converge on a different maximum of the objective function



- Rule 1 : selection and substitution within each island
(no interaction)

- Rule 2 : migration from one island *i* to another *j* with probability E_{ij} through clonation

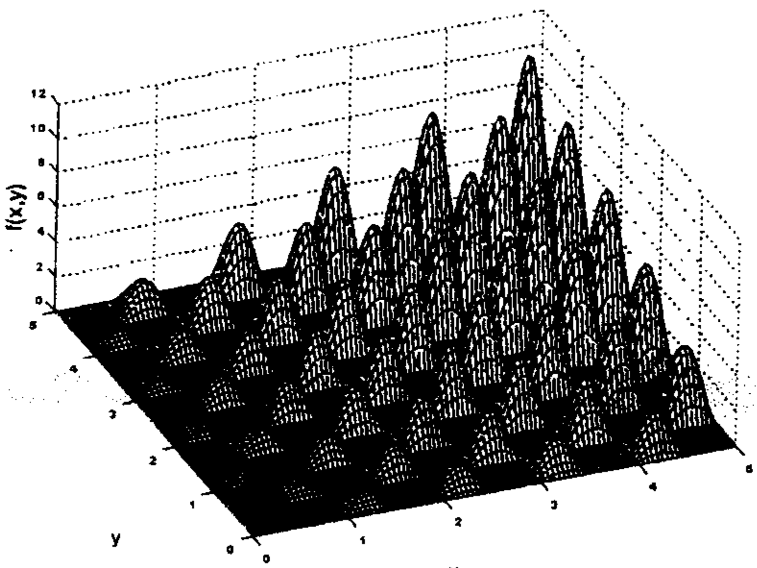
The migrating individual is cloned so that the original remains in island *i* where as the copy travels to island *j*, where at the same time another individual is eliminated so ast to keep constant the population N_j



ISLAND i : $N_i = 4$ ISLAND j: $N_j=3$

Application :

search for the maximum of a multimodal function



$$z = f(x,y) = \begin{cases} \sqrt{x} \cdot sin(2\pi x) \sqrt{y} \cdot sin(2\pi y) \text{ se } sin(2\pi x) \cdot sin(2\pi y) > 0 \\ 0 \text{ otherwise} \end{cases}$$

Method : GENETIC ALGORITHM WITH ISLANDS

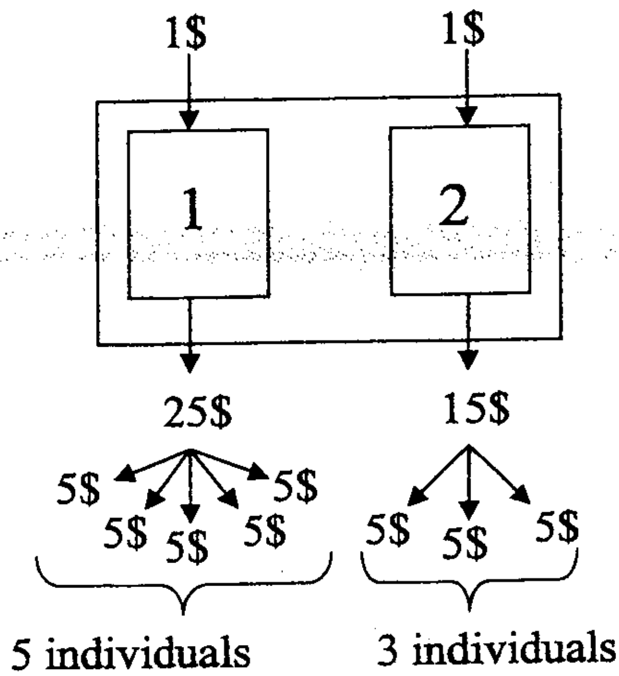
5 islands with populations of 10, 20, 30, 50, 80 individuals

migration matrix $$E = \begin{bmatrix} - & 0.05 & 0.05 & 0.05 & 0.1 \\ 0.05 & - & 0.05 & 0.1 & 0.15 \\ 0.05 & 0.1 & - & 0.1 & 0.15 \\ 0.05 & 0.1 & 0.15 & - & 0.2 \\ 0 & 0 & 0 & 0.05 & - \end{bmatrix}$$

Result : the genetic algorithm has found the largest error with a relative error $< 2 \cdot 10^{-5}$ on the coordinates x e y

Method 3 : GENETIC ALGORITHM WITH PARTINIONING

Double slot-machine :



pro capite gain: (= fitness)	increasing with the total gain (fitness value) decreasing with the number of individuals which share it

Possible implementation of the sharing's algorithm

$$f'(i) = \frac{f(i)}{\sum_{j=1}^{n} r(d_{ij})}$$

where

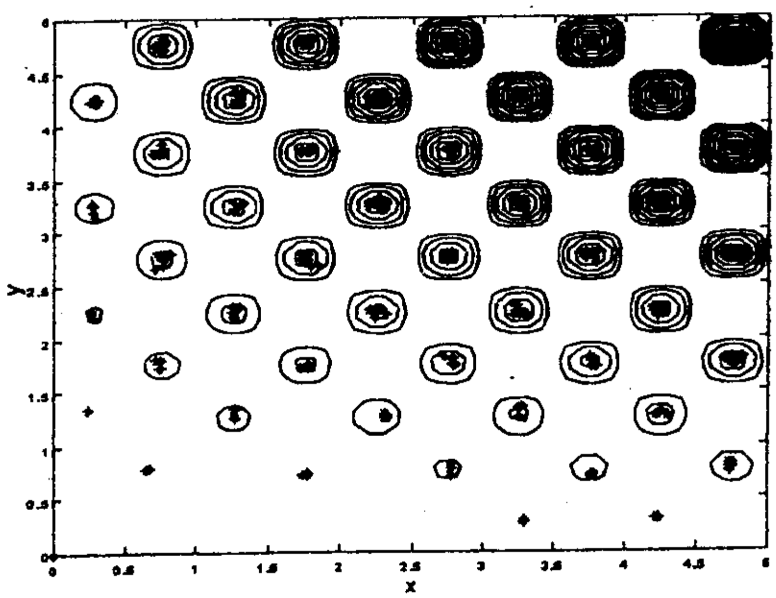
d_{ij} = Euclidean distance between individuals i and j

$$r(d) = 1 - \frac{d}{\alpha} \qquad for \quad d \le \alpha$$
$$= 0 \qquad otherwise$$

Search for the maxima of a multimodal function

$\sqrt{y}$

$$z = f(x,y) = \begin{cases} \sqrt{x} \cdot sin(2\pi x) \cdot sin(2\pi y) \; if \; sin(2\pi x) \cdot sin(2\pi y) > 0 \\ 0 \; otherwise \end{cases}$$



Result:

Population: 500 individuals

α = 0.5

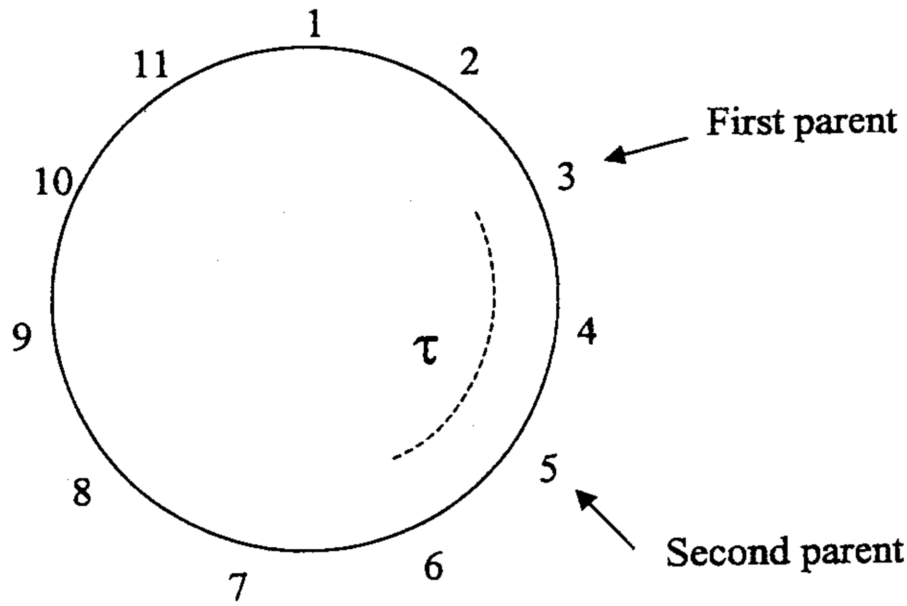
N. of detected peaks = 47 (over 50)

Method 2 : GENETIC ALGORITHM WITH SPATIAL MATING

Standard genetic algorithm : selection of both parents from the whole population

Genetic algorithm with spatial mating: selection of one parent from the whole population and of the second from the 'neighbourhood' of the first one

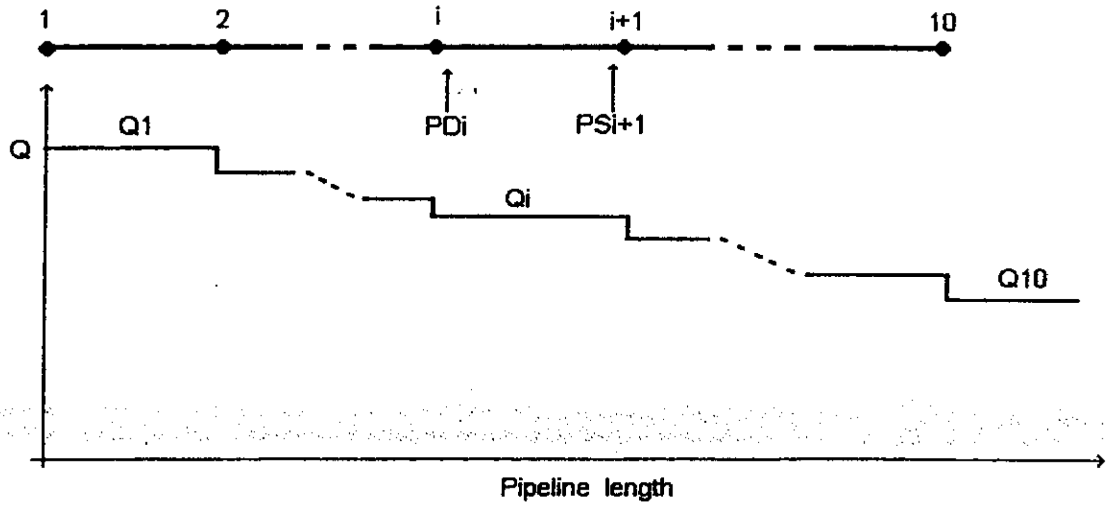
Example: monodimensional circular geometry



Procedure :

1) selection of first parent within the whole population
2) selection of second parent shifting, say, k steps to the right (or left) where k is sampled from $e^{-k/\tau}$ (τ = average distance between the two parents)

NATURAL GAS PIPELINE SYSTEM



<u>Input</u> : (11 values)

$PS_1 \equiv$ suction pressure to compressor 1

$Q_1 \equiv$ flow rates , i = 1, ..., 10

<u>Model</u> : (70 known parameters)

$$PD_i^2 - PS_{i+1}^2 = K_i Q_i^2 \text{ , } i = 1, \ldots, 10 \qquad [1]$$

$$HP_i = Q_i \left[A_i \left(\frac{PD_i}{PS_i} \right)^{R_i} - B_i \right] \text{ , } i = 1, \ldots, 10 \qquad [2]$$

K_i , A_i , B_i , R_i = given constants

Constraints : $(P_{i-1})_{min} \le PS_i \le (P_{i-1})_{max}$ [3]

$(P_i)_{min} \le PD_i \le (P_i)_{max}$ [4]

$1 \le \frac{PD_i}{PS_i} \le (S_i)_{max}$ [5]

$(S_i)_{max}$, $(P_i)_{min}$, $(P_i)_{max}$ = given constants

<u>Target</u> :

Estimate of 20 pressure values , PD_i , (i = 1, ... , 10) and PS_i , (i = 2, ... ,11) which minimize the total horsepower $HP = \sum_{i=1}^{10} HP_i$

2

*1 psi = 0.069 bar

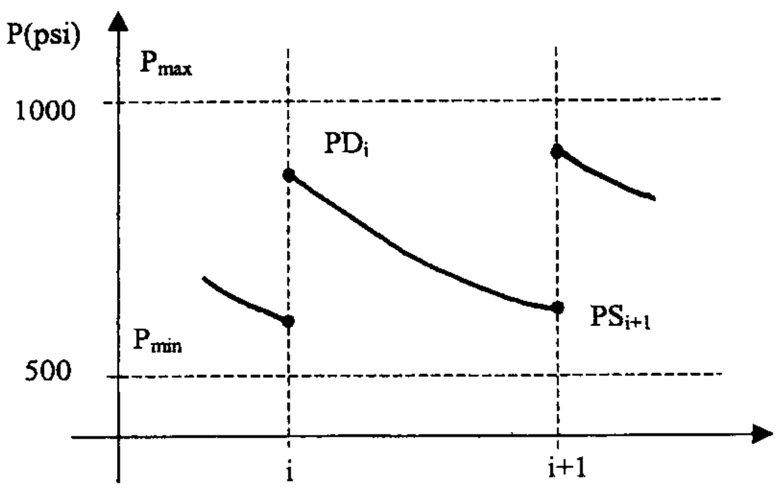
† 1 mmcfd = 10^6 ft^3/day = 0.3277 m^3/s

500 psi = 34.5 bar

i	K_i [psi^2·mmcfd^{-2}]	R_i	A_i [hp·mmcfd^{-1}]	B_i [hp·mmcfd^{-1}]	$P_{i,min}$ [psi]	$P_{i,max}$ [psi]	$S_{i,max}$
1	0.800	0.217	215.8	213.9	500	1000	1.6
2	0.922	0.217	215.8	213.9	500	1000	1.6
3	1.870	0.217	215.8	213.9	500	1000	1.5
4	0.894	0.217	215.8	213.9	500	1000	1.3
5	0.917	0.217	215.8	213.9	500	900	1.6
6	0.989	0.217	323.7	320.8	500	1000	1.6
7	0.964	0.217	215.8	213.9	500	900	1.75
8	1.030	0.217	215.8	213.9	500	1000	1.5
9	1.950	0.217	215.8	213.9	500	1000	1.6
10	1.040	0.217	215.8	213.9	500	1000	1.6

Parameters of the gas pipeline

Q_0 600.0 mmcfd = 196.6 m^3/s



In the piece of pipeline between compressors i and $i+1$, the gas pressure is constrained to remain within ($P_{i,min}$, $P_{i,max}$).

$$P_{i,\min} \le PD_i \le P_{i,\max}$$

$$P_{i,\min} \le PS_{i+1} \le P_{i,\max}$$

Solution # 1 [Wong and Larson, "Optimization of Natural-Gas Pipeline Systems Via Dynamic Programming", IEEE Transactions on Automatic Control]

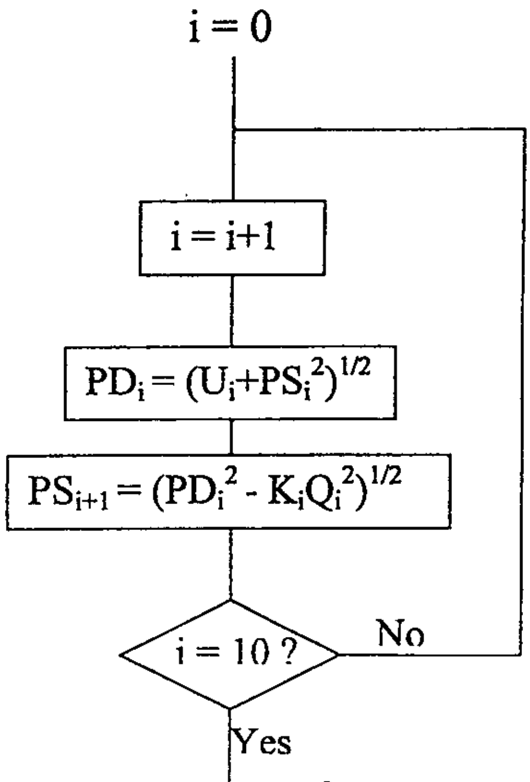
Dynamic programming by using an iterative functional equation in terms of a minimization cost function

The resulting pressures will be shown in the last figure

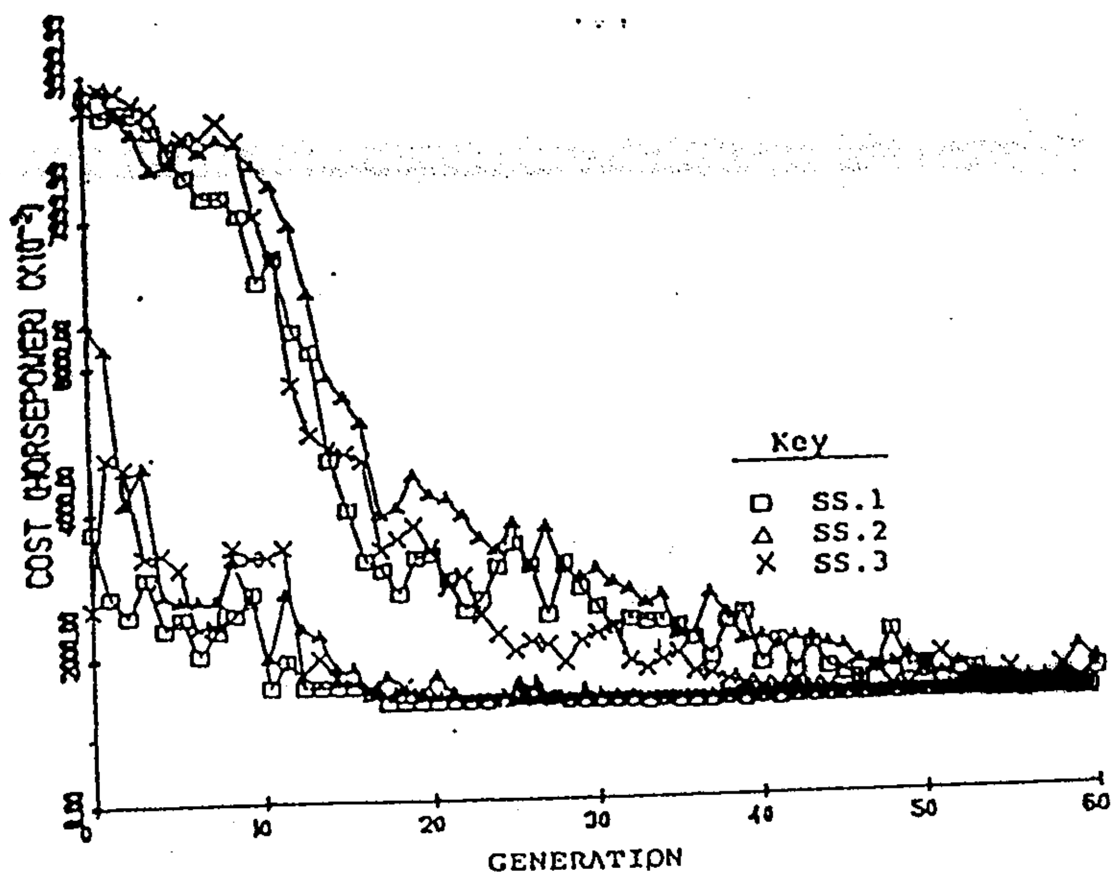
Solution # 2 [Goldberg, "Genetic Algorithms in Search, Optimization and Machine Learning" pagg.125-132, Addison-Wesley Publishing Company]

- The control factors are $U_i = PD_i^2 - PS_i^2$
- The constraints are replaced by a penalty function

Given PS_1 and U_i, i=1, ..., 10 :



4


COST (HORSEPOWER) (X10^-3)
0
10
20
30
40
50
60
GENERATION
Key
SS.1
SS.2
SS.3

5

Solution # 3

1st attempt :

The repetition of the Goldberg's approach was unsuccessful :

- pressures out of ranges
- complex pressures

2nd attempt :

Throughout the population initialisation and the breeding procedure, chromosomes are accepted only if the control factors (the 10 genes) satisfy the independent constraints :

- Case with 4 compressors : good results in ~1^h CPU-time
- Case with 10 compressors : computation interrupted after ~5^h CPU-time

3rd *attempt* :

- The control factors have been changed from the Goldberg's $U_i = PD_i^2 - PS_i^2$ to $U_i = PS_{i+1}$

- The initial population was created according to the "*conditional sampling*" method here proposed. By so doing the side of the 10-D hypercube has been reduced from 500 psi to 110 psi

- Case with 10 compressors : results slightly better than Goldberg's in ~30" CPU-time

Creation of the population by "*conditional sampling*" :

Consider the i-th pipe-section between compressors i and i+1. The value PS_i is known either from the calculation of the preceding pipe-section or assigned as the initial condition for i=1.

The conditioned range for PS_{i+1} is evaluated as follows :

i) From constraint [3] it follows that

$$(P_i)_{min}^2 \leq PS_{i+1}^2 \leq (P_i)_{max}^2$$

ii) Moreover, from equation [1] it follows that

$$PS_{i+1}^2 = PD_i^2 - K_iQ_i^2$$

so that PS_{i+1}^2 must also belong to the range :

$$(PD_i)_{min}^2 - K_iQ_i^2 \leq PS_{i+1}^2 \leq (PD_i)_{max}^2 - K_iQ_i^2$$

where the range of PD_i^2 is established by constraints [4] and [5] :

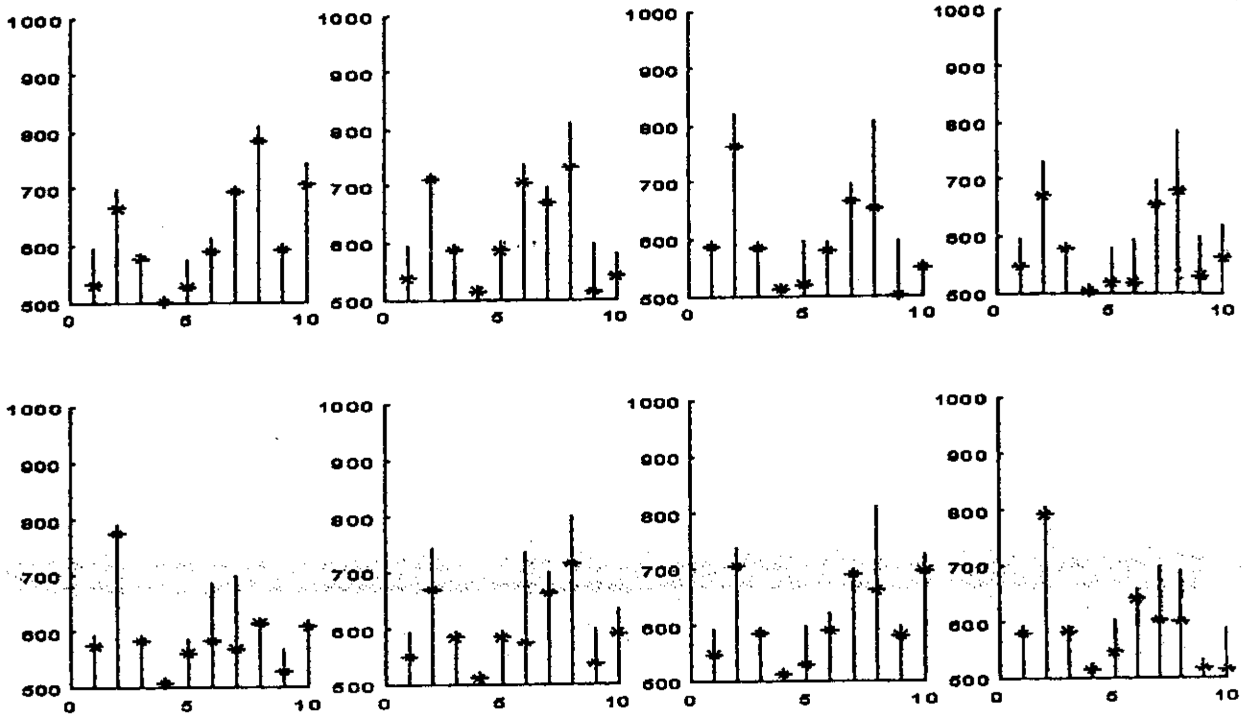
$$\max \{(P_i)_{min}^2, PS_i^2\} \leq PD_i^2 \leq \min \{(P_i)_{max}^2, (S_1)_{max}^2\, PS_i^2\}$$

iii) Finally, the range for PS_{i+1} is

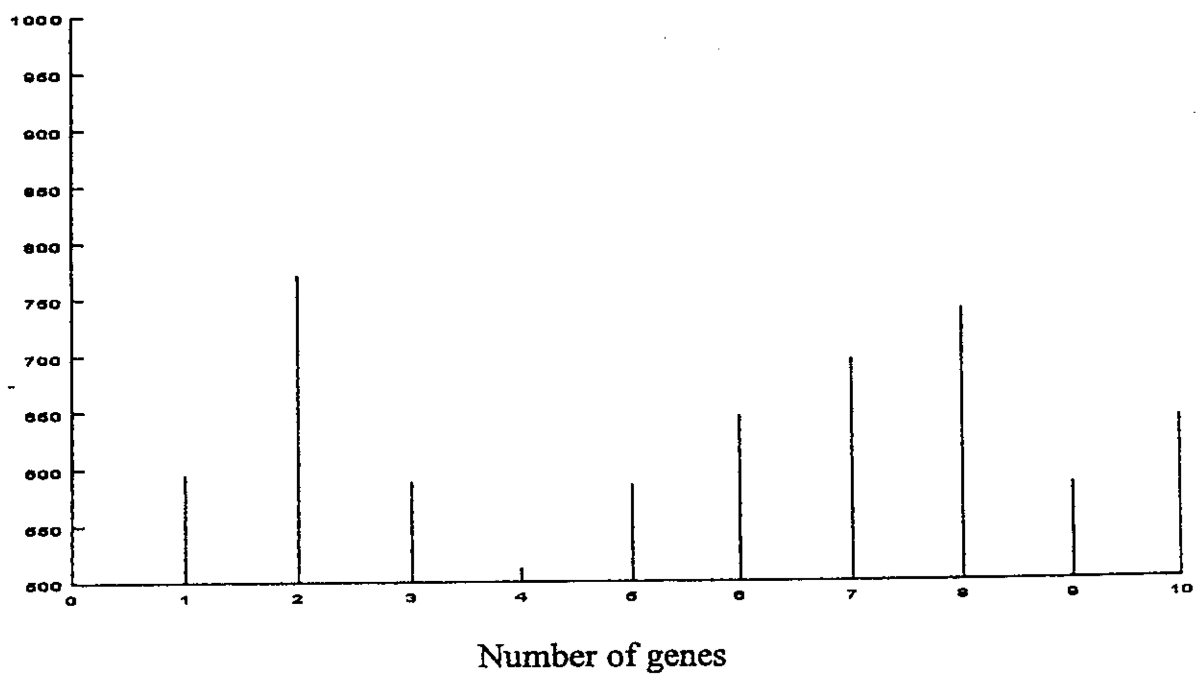
$$\max \{(P_i)_{min}^2, (PD_i)_{min}^2 - K_iQ_i^2\} \leq PS_{i+1}^2 \leq \min \{(P_i)_{max}^2, (PD_i)_{max}^2 - K_iQ_i^2\}$$

Once the range for PS_{i+1} is established, a value PS_{i+1} is uniformly sampled.

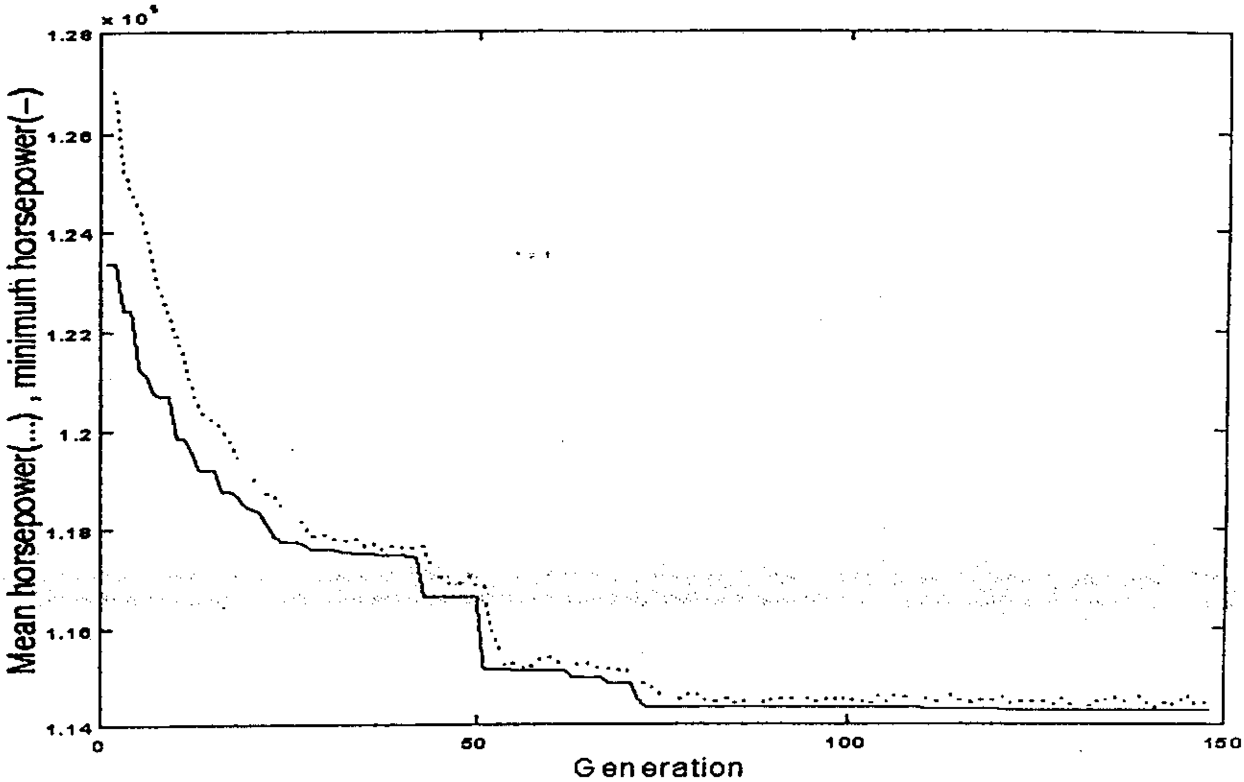
Then it is possible to proceed to the next pipe-section .

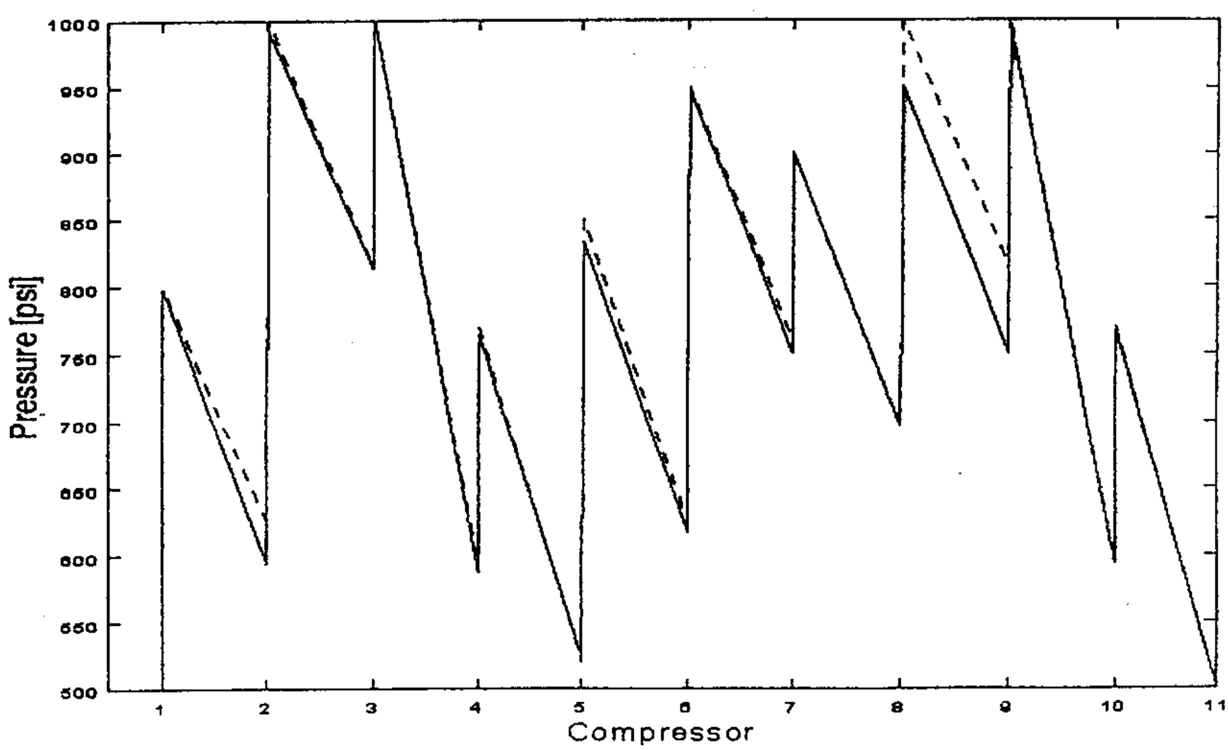


Eight examples of chromosome initialisation by the "*conditional sampling*" method



Mean ranges of control factors during initialisation by the "*conditional sampling*" method





--- Wong & Larson

___ Present work

System Design Optimization By Genetic Algorithms

INTRODUCTION

When designing a system, several choices must be made concerning the type of components to be used and their assembly configuration.

The choice is driven by the interaction of reliability/availability objectives and economic needs.

Standard approaches to determine optimal solutions of design problems often encounter difficulties when including realistic cost and reliability issues.

PROBLEM STATEMENT

PLANT DESIGN

SAFE OPERATION :

- RELIABILITY
- AVAILABILITY
- ACCIDENT RISK
  → CONSEQUENCES

↔

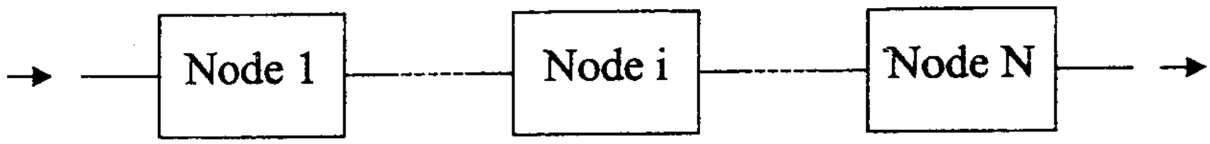
ECONOMICS:

- PROFIT
- PURCHASE COSTS
- REPAIR COSTS
- NO–SERVICE FINE
- DAMAGE PAYBACK

ISSUES :

- COMPONENTS CHOICE
- CONFIGURATION CHOICE
- AGING
- REPAIR STRATEGIES

PLANT = N nodes in series logic



Node k = ensemble of n_k components in series, parallel and/or standby

PLANT CONFIGURATION = vector of the states (functioning, failed, standby) of the components

CUT SET = system configuration of failure

The plant is potentially risky: some cut sets are '<u>accidents</u>' with damaging consequences (eg. to the environment).

OBJECTIVE FUNCTION

In order to guide the selection, the designer defines an objective function which accounts for all the relevant aspects of plant operation.

Here we consider as objective function the net profit drawn from the plant during the mission time T_M.

$$G = P - (C_A + C_R + C_D + C_{ACC})$$

$$P = P_t \cdot \int_0^{T_M} \frac{A(t)dt}{(1+i)^t}$$ **= plant profit**

$$C_A = \sum_{j=1}^{N_C} C_j$$ **= components cost**

$$C_R = \sum_{j=1}^{N_C} C_{Rj} \cdot \int_0^{T_M} \frac{I_{Rj}(t)dt}{(1+i)^t}$$ **= repair cost**

$$C_D = C_U \cdot \int_0^{T_M} \frac{[1 - A(t)]dt}{(1+i)^t}$$ **= non-service penalty**

$$C_{ACC} = \sum_{k=1}^{N_{ACC}} I_{ACC,k} \cdot \frac{C_{ACC,k}}{(1+i)^{t_{ACC,k}}}$$ **= reimbursement for damages from an accident**

THE GENETIC ALGORITHM OPTIMIZATION APPROACH

- Population of chromosomes (bit-strings) ⇒ possible solutions.
- Evolution: parents selection, crossover, replacement, mutation.

In this work

- Available alternative node configurations are numbered.
- System configuration is identified by a sequence of integers.
- Chromosome = system configuration = single gene containing all the indexes of the node configurations.

- **<u>Parents selection</u> = standard roulette rule (selecting the parents in proportion to their values of fitness)**

- **<u>Crossover</u> = inserting at random a separator in the homologous genes of the selected parents**

- **<u>Replacement</u> = keeping the fittest two, and eliminating the remaining among the two parents and two children**

- **<u>Mutation</u> is performed with probability 10^{-3}.**

NUMERICAL APPLICATION

- System with 3 nodes.
- 4 alternatives for each node

N	C 1	C 2	C 3	C 4
A	1/1	1/2	1/3	2/3
B	1/1	1/1 +1sb	1/1 +2sb	1/1 +3sb
C	1/1	1/1 +1sb	1/1 +2sb	1/1 +3sb

i/j = configuration i-out-of-j G
i/j + k sb= configuration i-out-of-j G with k additional standby components

- 64 possible system configurations.
- Each node requires 2 bits
- System config. = one 6-bit gene
- Population = 30 chromosomes

Simplifying assumptions are made:

i) **Node A: all components equal;**

ii) **All standby's are cold;**

iii) **No repair is allowed;**

iv) **One damaging accident: node C.**

⇒ analytic evaluation of G

Component i	Failure rate λ_i [10^{-3} y^{-1}]	Purchase cost C_i [10^3 $]
a	2.6	0.7
b1	5.3	0.3
b2	3.6	0.3
b3	4.7	0.7
b4	2.6	0.7
c1	8.1	4.0
c2	5.3	6.0
c3	7.0	2.0
c4	4.2	8.0

Profit per unit time P_t	$[10^3\ \$ \cdot y^{-1}]$	0.94
Downtime penalty per unit time C_U	$[10^3\ \$ \cdot y^{-1}]$	3.00
Accident reimbursement cost C_{ACC}	$[10^3\ \$]$	420
Mission time [y]		30

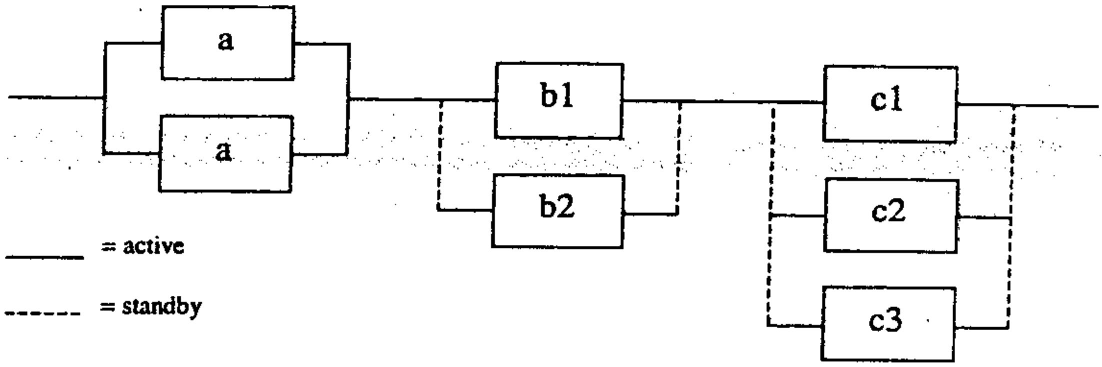
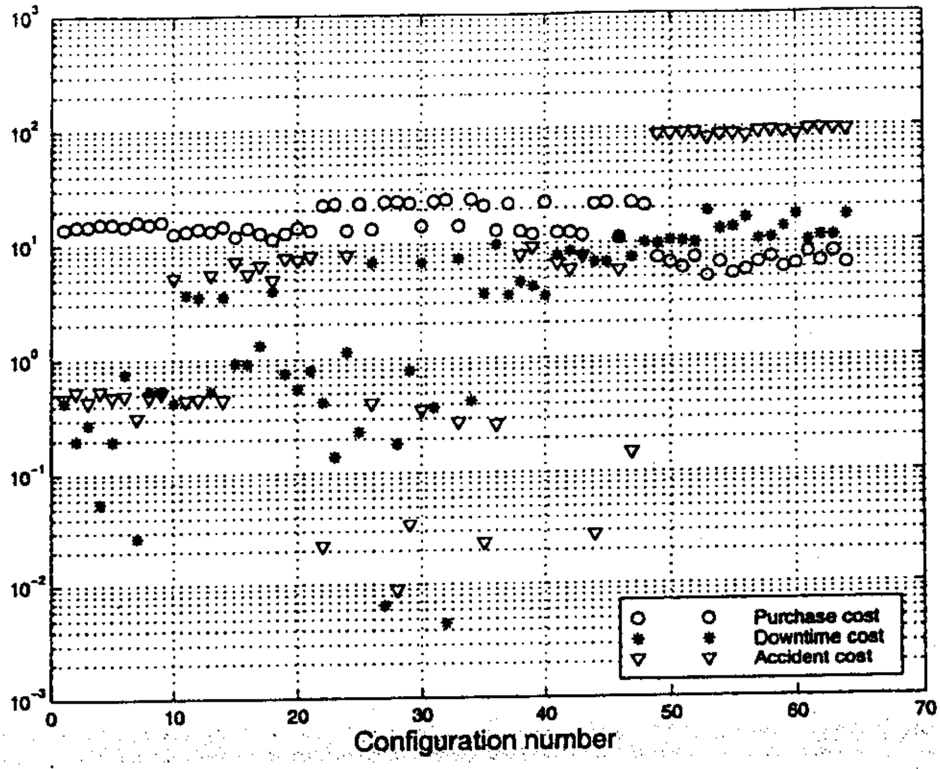


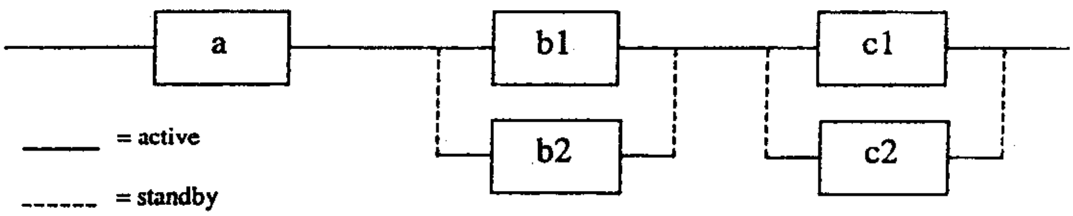
Figure 1: Sketch of the optimal configuration



- The system purchase cost is rather insensitive to the configuration
- Downtime costs are lower for the first best alternatives
- The last (worst) configurations are strongly penalized by high accident costs as indeed they correspond to having node C with only one single component and no redundancy.
- Analytic validation: same optimal configuration

REPAIRABLE COMPONENTS

Component i	Repair rate μ_i [10^{-1} y^{-1}]	Repair cost C_{Ri} [10^3 \$·y^{-1}]
a	1.0	2.5
b1	3.0	1.5
b2	1.0	0.5
b3	3.0	4.0
b4	1.0	2.5
c1	5.0	21.0
c2	3.0	29.0
c3	5.0	12.0
c4	3.0	48.5



Repairs reduce the number of redundant components in the optimal configuration.

OPTIMIZATION OF A SHALE OIL PLANT

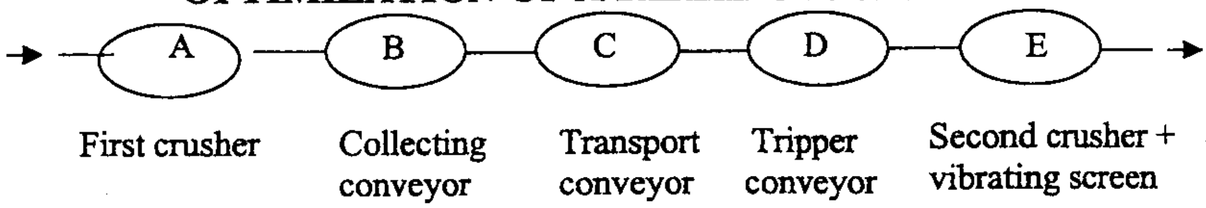


Figure 5: Sketch of the shale oil plant

Node	Number of alternative configurations	Type of components	Operational logic
A	7	a	3-out-of-3 G 3-out-of-4 G 3-out-of-5 G 3-out-of-6 G 3-out-of-7 G 3-out-of-8 G 3-out-of-9 G
B	16	b1, b2, b3	2-out-of-2 G 2-out-of-3 G
C	14	c1, c2	1-out-of-1 G 1-out-of-1 G + 1 standby 1-out-of-1 G + 2 standby
D	14	d1, d2	1-out-of-1 G 1-out-of-1 G + 1 standby 1-out-of-1 G + 2 standby
E	7	e	3-out-of-3 G 3-out-of-4 G 3-out-of-5 G 3-out-of-6 G 3-out-of-7 G 3-out-of-8 G 3-out-of-9 G

Table 5: Potential node configurations

System configurations 153,664

Component i	Failure rate λ_i $[y^{-1}]$	Repair rate μ_i $[y^{-1}]$	Purchase cost C_i $[10^6 \$]$	Repair cost C_{Ri} $[10^6 \$ \cdot y^{-1}]$
a	$1.5 \cdot 10^{-3}$	$4.0 \cdot 10^{-2}$	3.0	0.55
b1	$2.0 \cdot 10^{-4}$	$8.0 \cdot 10^{-3}$	5.0	10.0
b2	$2.0 \cdot 10^{-3}$	$8.0 \cdot 10^{-2}$	3.0	6.2
b3	$2.0 \cdot 10^{-2}$	$8.0 \cdot 10^{-1}$	1.0	2.1
c1	$1.0 \cdot 10^{-4}$	$8.0 \cdot 10^{-3}$	10.0	41.0
c2	$1.0 \cdot 10^{-3}$	$8.0 \cdot 10^{-2}$	5.0	20.0
d1	$1.0 \cdot 10^{-4}$	$8.0 \cdot 10^{-3}$	7.0	28.0
d2	$1.0 \cdot 10^{-3}$	$8.0 \cdot 10^{-2}$	3.0	12.0
e	$1.7 \cdot 10^{-3}$	$4.0 \cdot 10^{-2}$	5.0	0.85

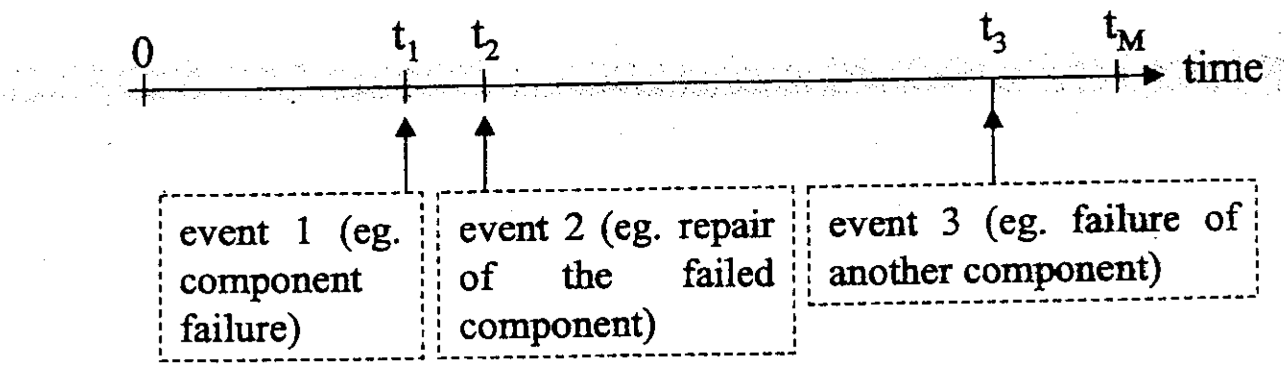
Table 6 : Component data

Profit per unit time P_t	$[10^6 \$ \cdot y^{-1}]$	20.0
Downtime penalty per unit time C_U	$[10^6 \$ \cdot y^{-1}]$	200.0
Accident 1 (node A) reimbursement cost $C_{ACC,1}$	$[10^6 \$]$	70.0
Accident 2 (node E) reimbursement cost $C_{ACC,2}$	$[10^6 \$]$	50.0
Interest rate i		3%
Mission time T_M	[y]	50

Table 7 : System data

MONTECARLO METHOD

Objective : simulate the system evolution from the initial time ($t_0=0$) to the mission time (t_M)



Simplifying assumption : failure and repair times are distributed exponentially

$$\begin{cases} \lambda_i = \underline{\text{failure}} \text{ rate of the i-th component} \\ \mu_i = \underline{\text{repair}} \text{ rate of the i-th component} \end{cases}$$

$\Lambda e^{-\Lambda(t-t_k)}\, dt$ = probability of the next event in (t, t+dt) given that the previous one has occurred in t, where Λ is the Σ rates out of the configuration at time t_k

THE SYSTEM MODEL

1) imperfect repair with probability P: $\begin{cases} \lambda_i \to \lambda_i \cdot \pi_{\lambda_i} \\ \mu_i \to \mu_i / \pi_{\mu_i} \end{cases}$
2) different kinds of repair intervention
3) number of repair teams fixed for each kind
4) component repair priority:
   higher priorities are repaired first
5) an 'accident' cut set is an absorbing state (the system cannot be repaired)

The Monte Carlo generates a large number of system life histories and in the end it estimates the following mean values:

- T_S = plant service time
- T_{NS} = plant out-of-service time
- $T_{REP}(i)$, $i=1,\ldots,N_C$ = total repair time of component i
- $P_{ACC}(i)$, $i=1,\ldots,N_{ACC}$ = frequency of 'accident' cut set i
- $A(t)$ = plant instantaneous availability
  (probability that the system is UP at time t)

COUPLING GENETIC ALGORITHMS AND MONTE CARLO :

A) General Scheme

Each chromosome of the population encodes a proposed configuration design.

The Monte Carlo code estimates the objective function G in correspondence of each chromosome.

⇓

The evolution of the genetic algorithm, and embedded Monte Carlo, leads to the configuration with G_{MAX}

QUESTION :

- how can we be sure that the solution found is optimal?

ANSWER :

- No algorithm can guarantee that the maximum found, of a non linear, multivariate, constrained function, is the global one.

- One can, however, demand that some criteria are satisfied. In our case:
  - validation of the procedure on simple cases of known solution
  - validation of the procedure on more complex cases for which the solution can be 'guessed' a priori(on physical ground)
  - stability of the optimal configuration for different sequences of pseudorandom numbers

B) Detailed procedure

During its evolution, the genetic algorithm considers thousands of proposed system configurations: it is not possible to run a full Monte Carlo simulation for each of them, to obtain statistically significant estimates.

⇓

- During its evolution, the genetic algorithm re-examines the same configuration several times
- Each time a configuration is proposed, a short Monte Carlo (say 100 trials) is run: the estimate of G thereby obtained is scarcely statistically significant
- The best configurations are re-proposed by the genetic algorithm over and over (~10^4 times)
- For these 'good' configurations, the Monte Carlo estimates are repeated over and over, thus building statistically significance

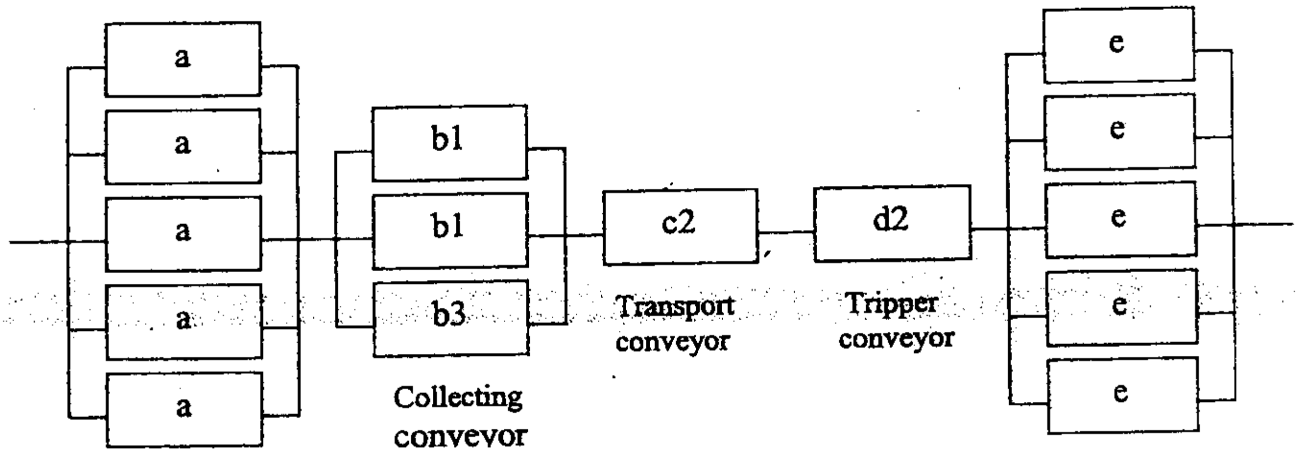


Figure 6: Sketch of the optimal configuration for the shale oil plant

Configuration index in decreasing order of optimality	Total net profit at T_M [10^6 $]
1	471.57 ± 0.08
2	470.20 ± 0.05
3	469.39 ± 0.07

Table 8: Monte Carlo results with 10^6 trials for the three best system configurations