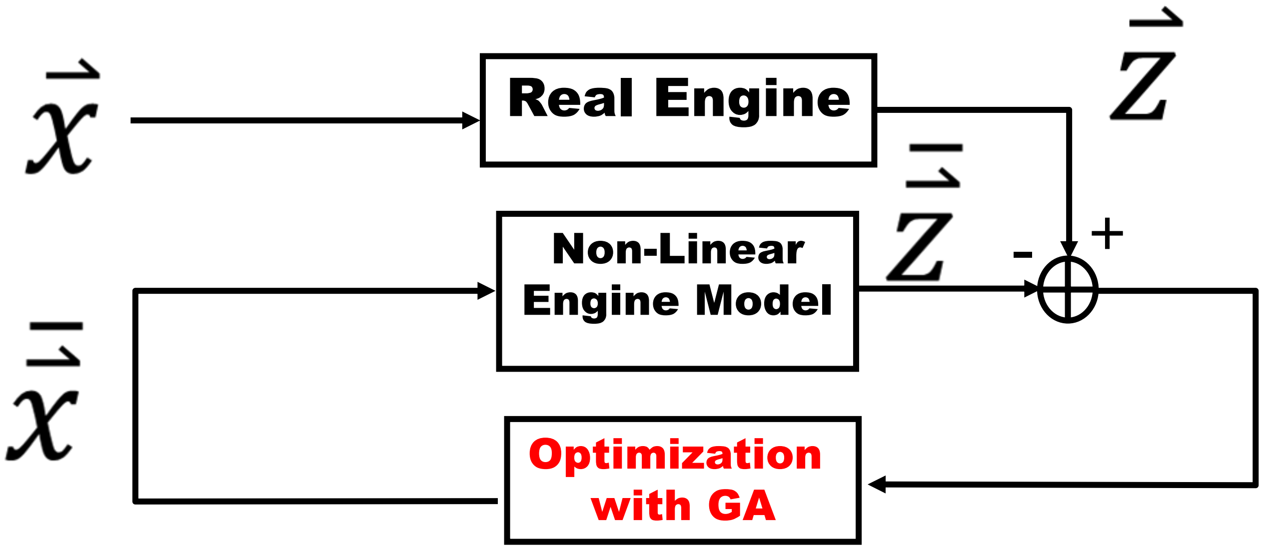
Genetic Algorithms and Gas Turbine Diagnostics

Optimization for Diagnostics

Gas turbine diagnostics as a searching or optimisation problem



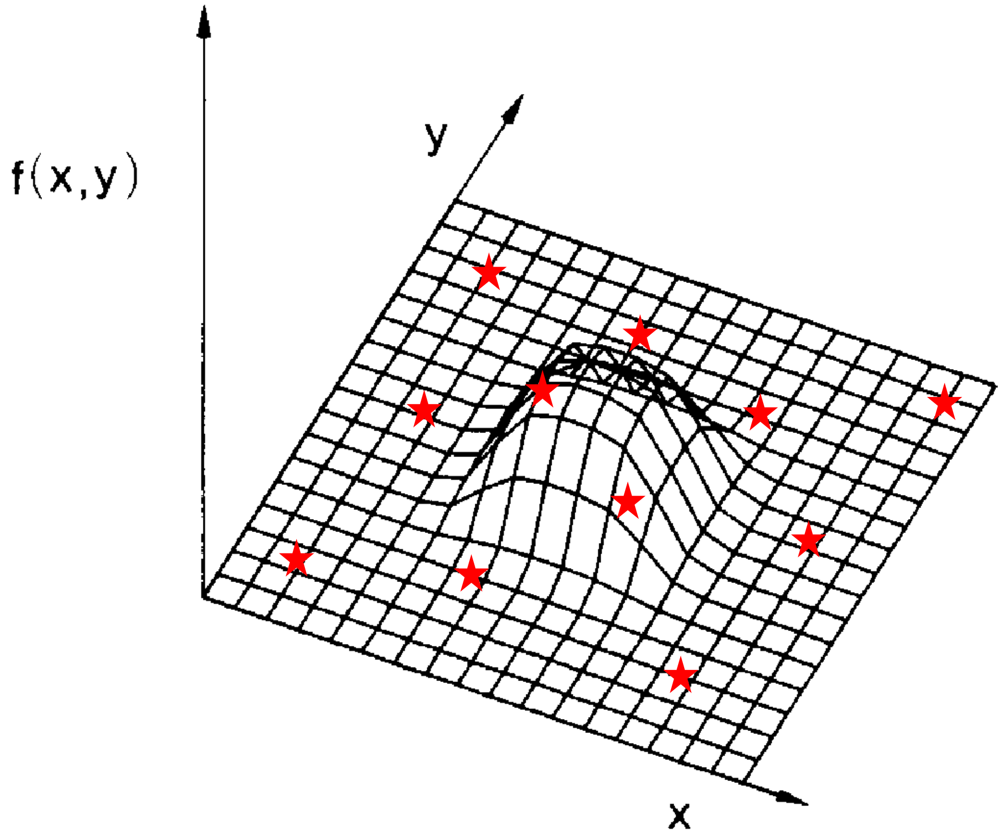
$$\vec{e} = \vec{z} - \bar{\vec{z}}$$

$$\text{Objective Function} = \sum_i \Phi\left(\| z_i - h\left(x_i\right) \|\right)$$

Optimization

- Random search

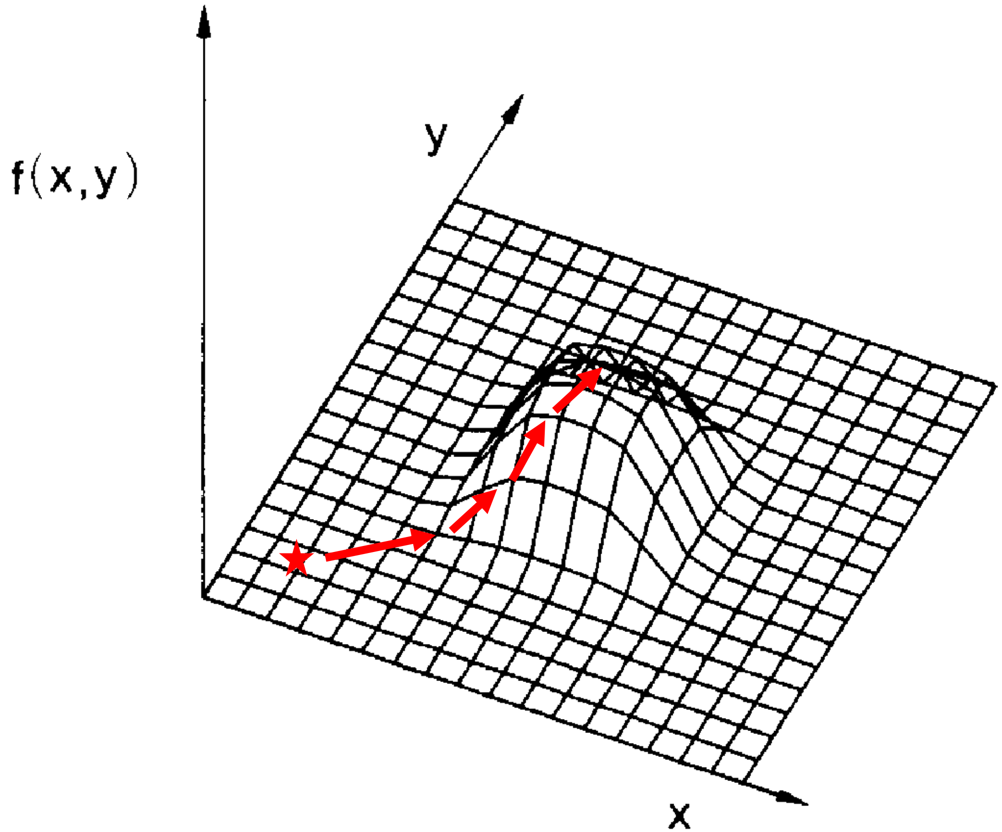
- Conventional optimization – Hill Climbing



Single peak, continuous function

Optimization

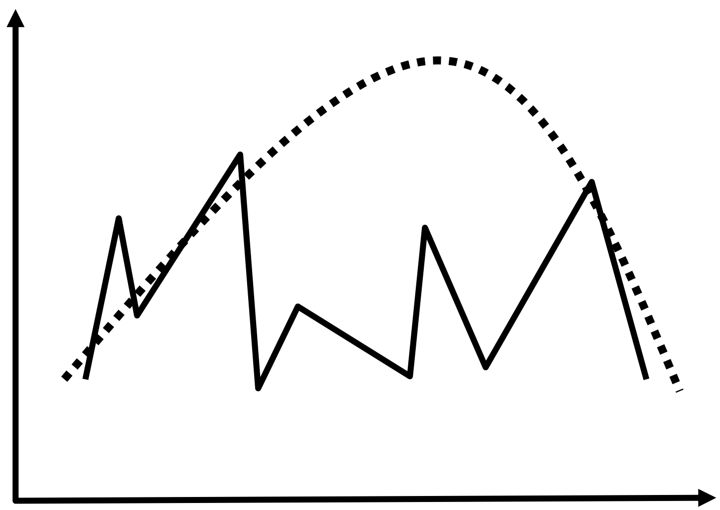
- Random search
- Conventional optimization – Hill Climbing



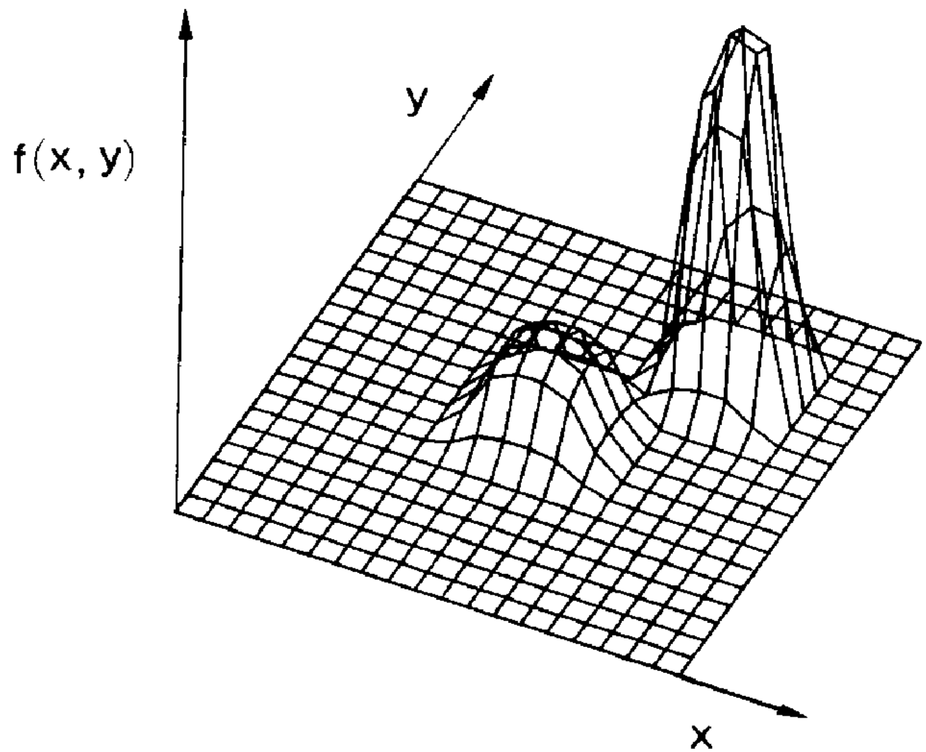
Single peak, continuous function

Optimization

Why choose Genetic Algorithm?



Discontinuous function

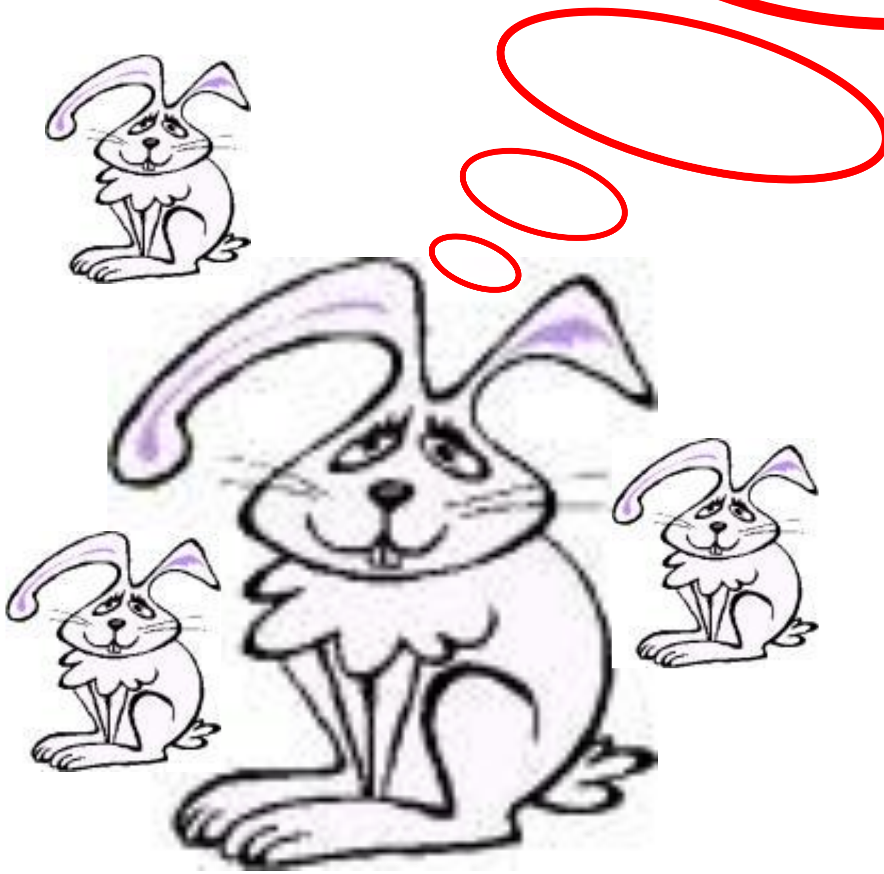


Multiple peak function

Genetic Algorithms

- ***A Genetic Algorithm (GA) is a search heuristic that mimics the process of natural selection.***
- ***A GA generates solutions to optimization problems using techniques inspired by natural evolution, such as inheritance, mutation, selection and crossover.***
- ***A GA requires a balance between exploiting the best solution and exploring the search space***

Natural Evolution

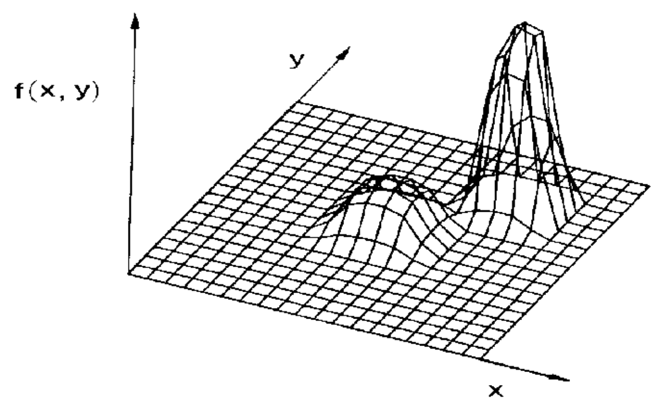


GA & Natural Evolution

A genetic algorithm follows a step-by-step procedure that closely matches the story of the rabbits.

- **rabbit population**
- **individual rabbit**
- **genes**
- **faster & smarter**
- **breeding rabbits**
- **become faster and smarter**

- **String population**
- **Individuals: strings/chromosomes**
- **String elements: units(genes)**
- **Evaluation with “fitness”**
- **Produce next generation with GA operators**
- **Searching for the best**



Five elements, five questions:

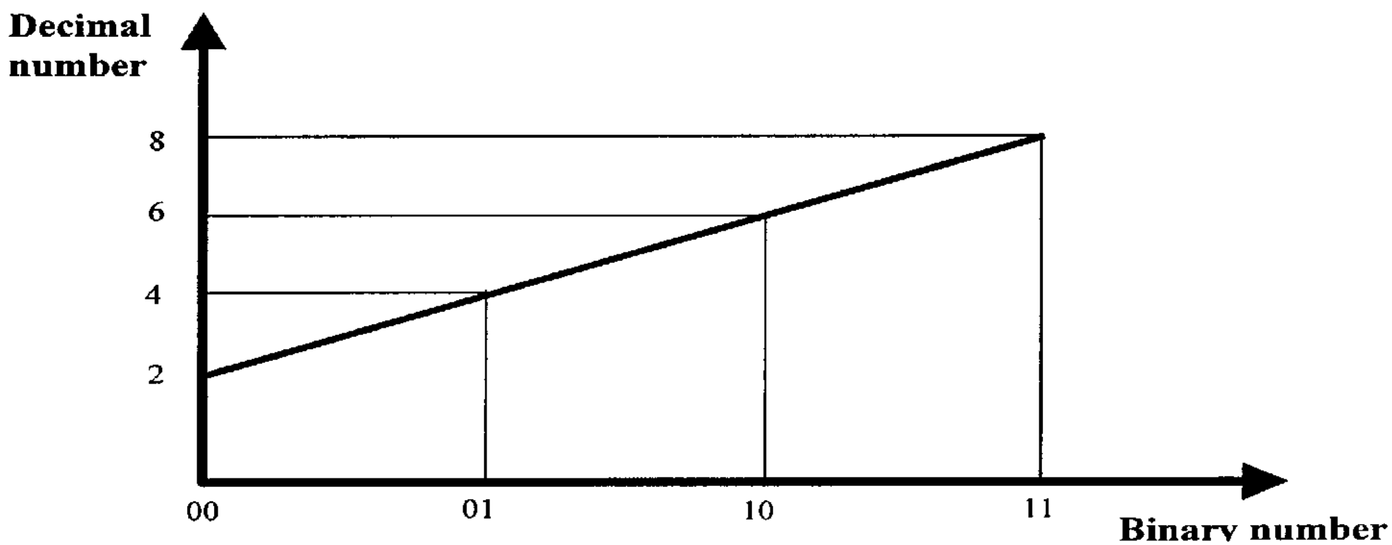
1. How to represent GA strings?
2. How to create an initial population?
3. How to evaluate the quality of GA strings?
4. How to generate new strings to improve the GA population?
5. How to choose the values of GA parameters

GA – Key elements

Representation of strings:

-- binary or real coded:

Real number (2,8) → Binary number (00,11)



GA - key elements

$$z = f(x, y)$$

Search space: x: [0, 10] ; y: [0, 10]

Individuals / Strings

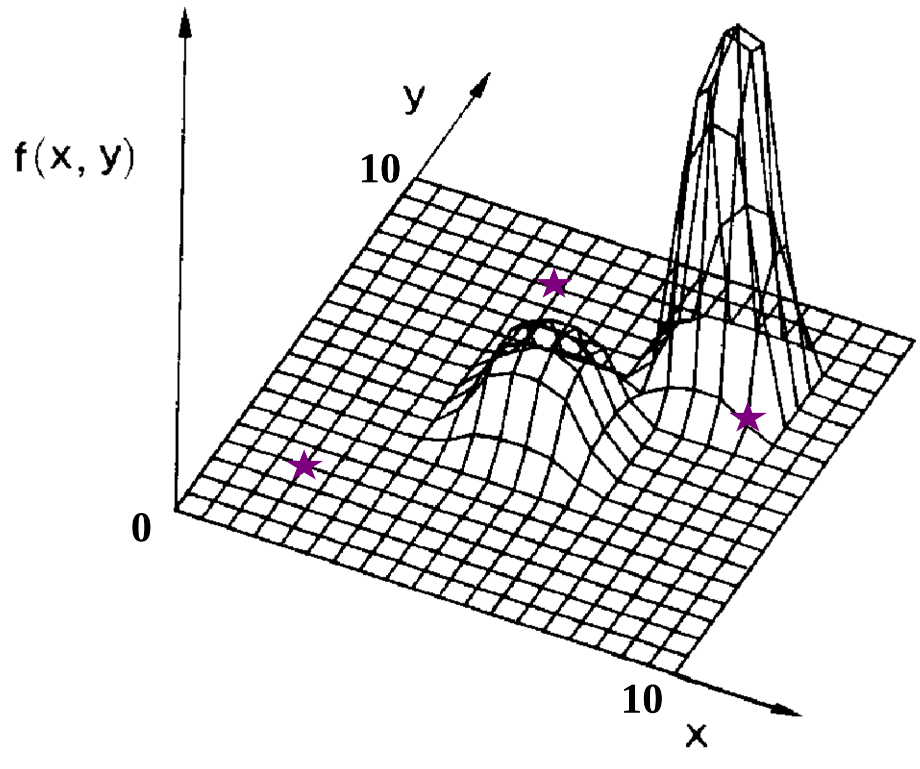
(1, 1)

(4, 9)

(9, 6)

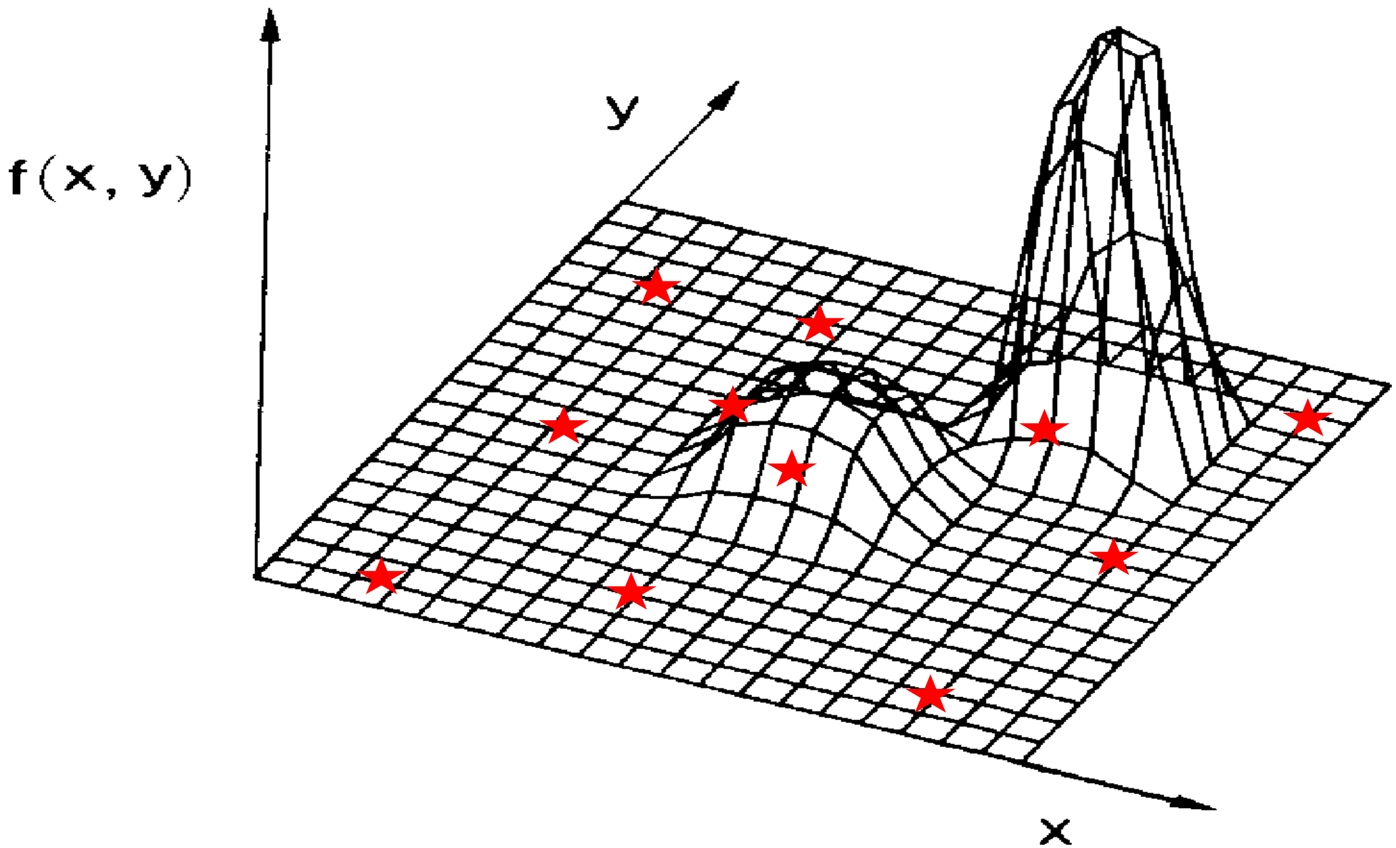
⋮

Population: number of strings



GA – Key elements

Initial population -- random selection



GA – Key elements

Evaluation:

A **fitness function** is used to assess if the solution is "good" or "bad"

The fitness function is related to the **objective function**:

- always positive
- high value for better fitness

For example,

- to maximize an objective function

$$Fitness = ObjeciveFunction$$

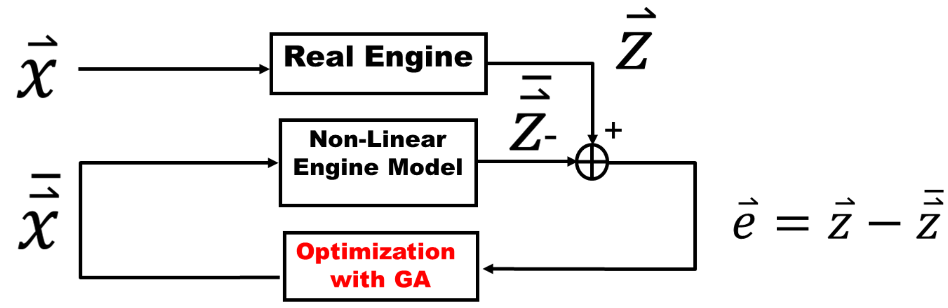
- to minimize an objective function

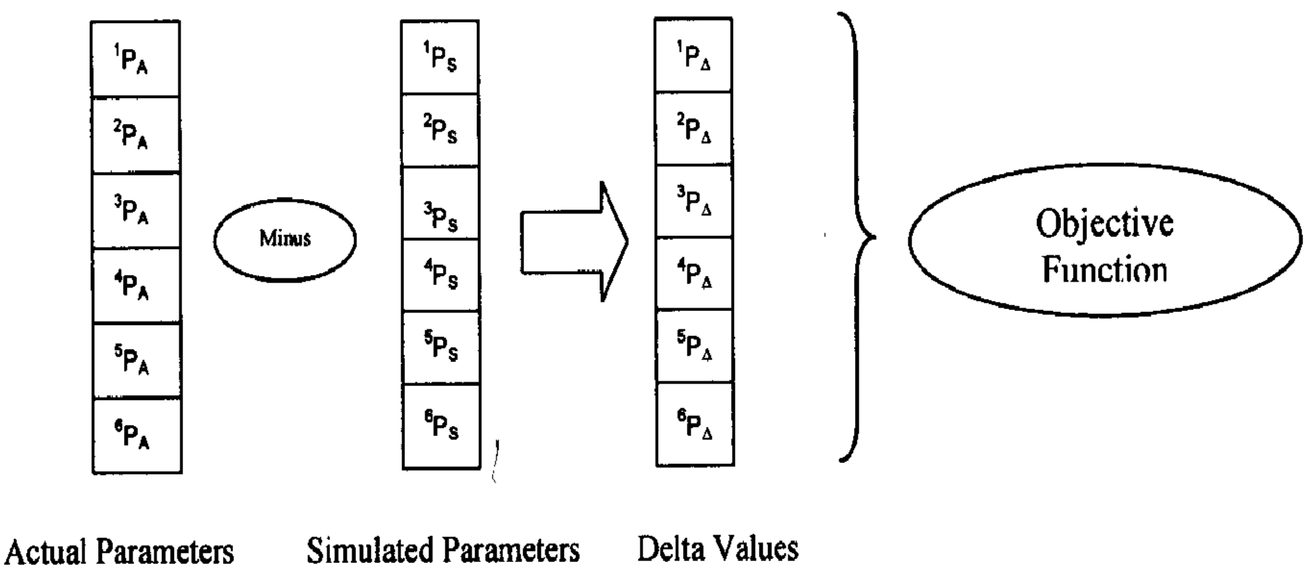
$$Fitness = \frac{1}{1 + ObjectiveFunction}$$

GA – Key elements

Objective Functions:

$$J(x) = \sum_{i} \frac{\left| z_j - h_j(\vec{x}) \right|}{\left| z_{ref} \cdot \sigma_j \right|}$$

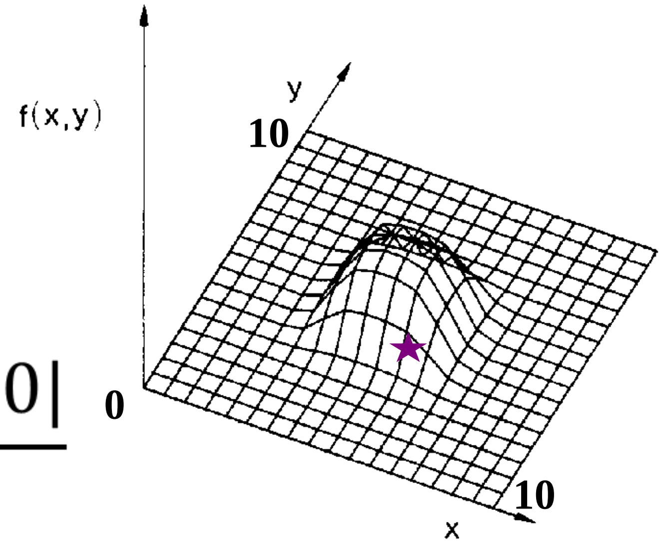




GA – Key elements

$$z = f(x, y)$$

$z = 100 - (x-5)^{**}2 - (y-5)^{**}2$



To search a solution with $z_{target} = 100$

Objective Function (errors) $= \dfrac{|z - 100|}{100}$

Fitness $= \dfrac{1}{1 + OF}$

Example:

String: (6, 4)

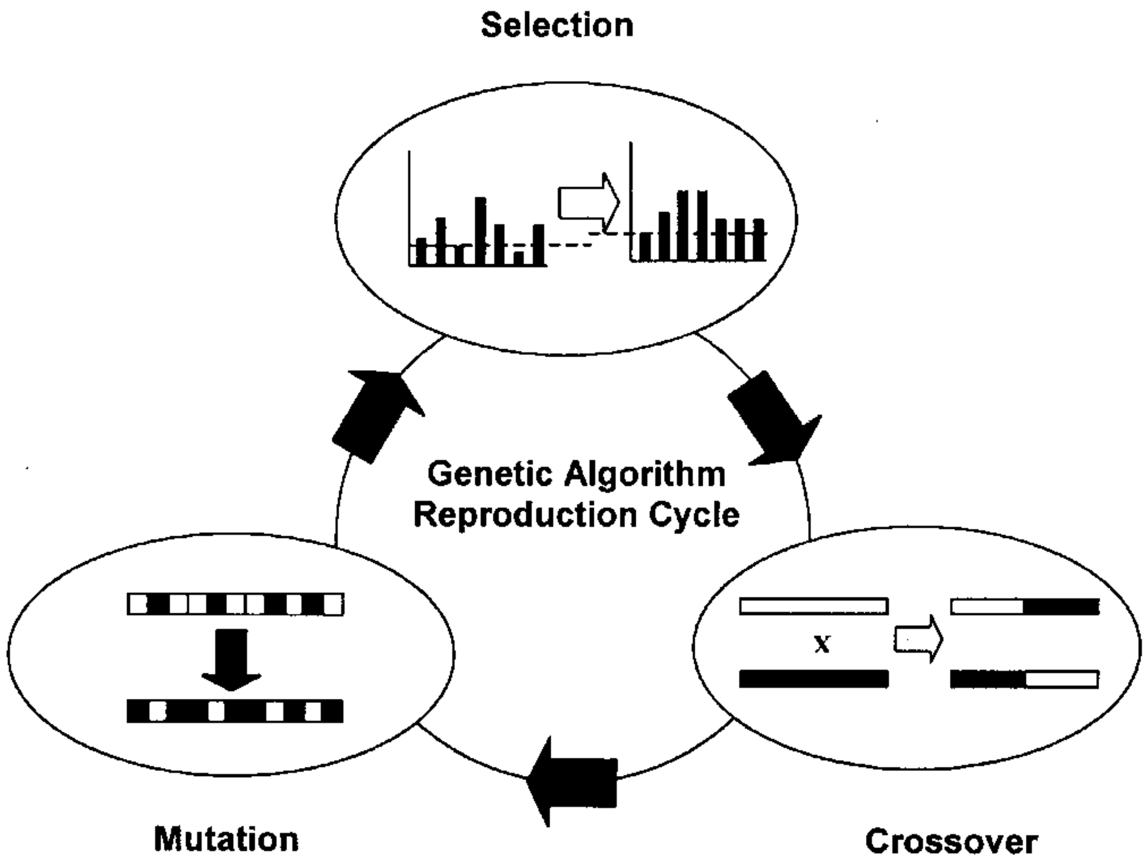
$z = 100-(6-5)^{**}2 - (4-5)^{**}2 = 98$

Objective Function $= |98 - 100| / 100 = 0.02$

Fitness $= 1 / (1 + 0.02) = 0.98$

GA – Key elements

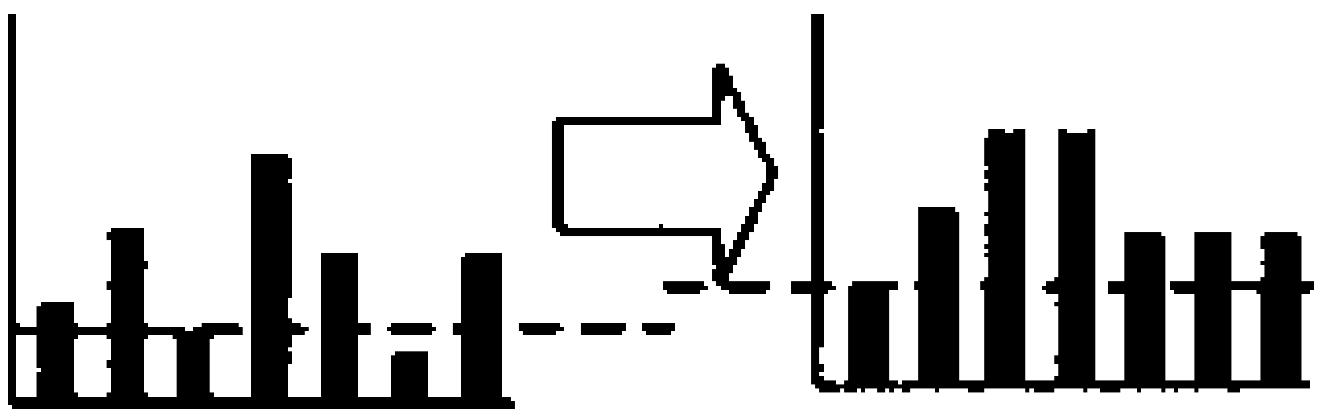
4. Three searching operations in GA:



GA – Key elements

Selection: -- individual strings are copied for the next generation according to the values of their fitness

- Set a threshold
- Exclude strings with fitness lower than the threshold



GA – Key elements

Crossover: -- Randomly create new strings from current population

	Binary	Real
• Parent 1:	XXX | **XXX**	**Num1**
• Parent 2:	YYY | **YYY**	**Num2**
• Child 1:	XXX | **YYY**	**Num1+Ran1*(Num2-Num1)**
• Child 2:	YYY | **XXX**	**Num1+Ran2*(Num2-Num1)**

GA – Key elements

Mutation: **-- Randomly change the value of a string position in order to avoid losing some potential useful kind of strings**

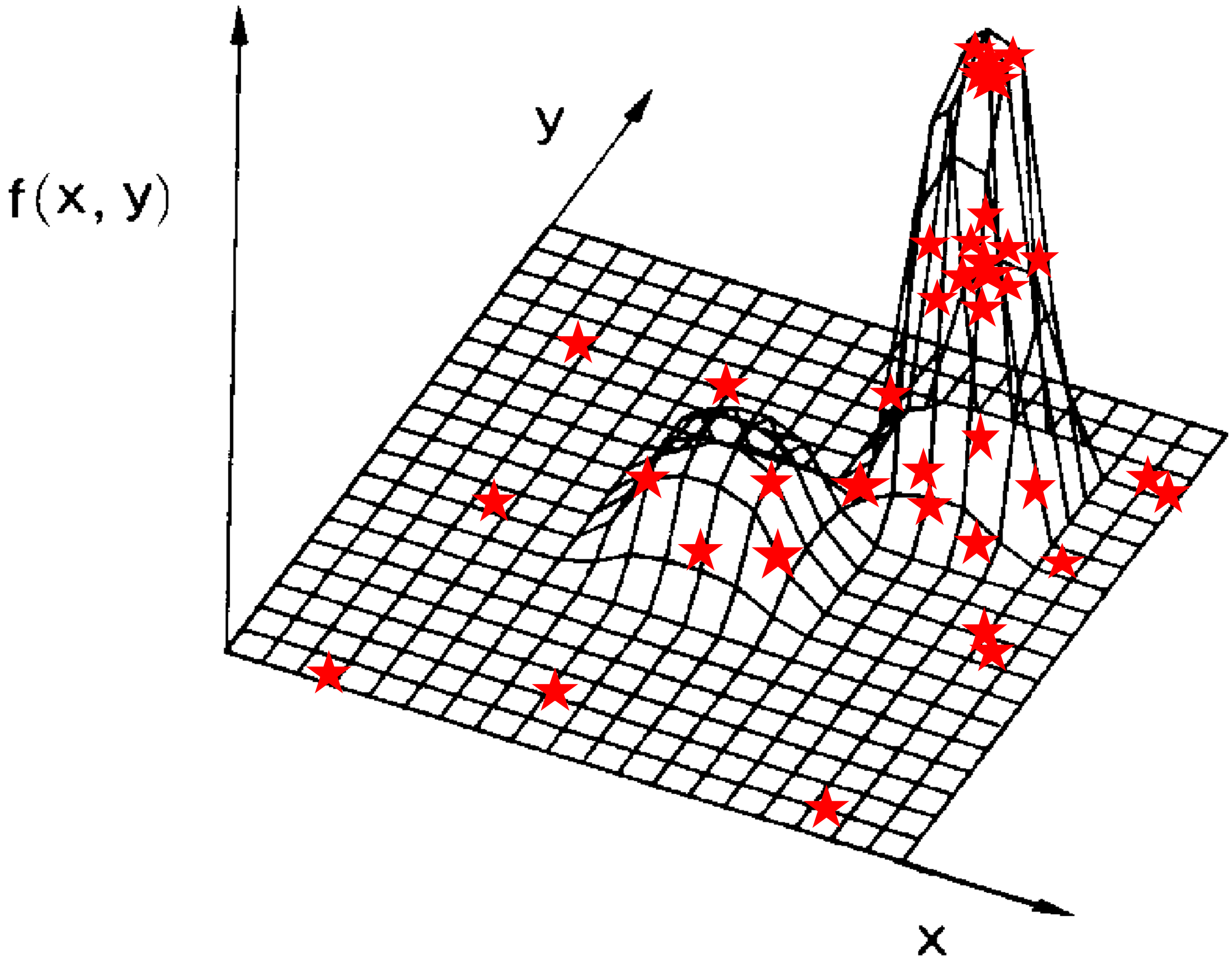
	Binary	Real
• Parent:	10**0**01**0**1**0**0	(Num_u, Num_l)
• Child:	10**1**01**1**1**1**0	Num_l+Ran*(Num_u – Num_l)

GA – Key elements

5. Selection of GA parameters:

- **Population size** (such as 50 strings/individuals)
- **Probability of crossover** (such as 60%)
- **Probability of mutation** (such as 40%)
- **Search space** (such as [-0.5, 0.5])

GA – Key elements



Genetic Algorithms

Procedures for Genetic Algorithms:

1. **Choose GA parameters** (population size, probabilities of GA operations, etc.)
2. **Genetic representation of strings/individuals** (real coded or binary)
3. **Create initial population of solutions** (random generation)
4. **Evaluation of strings/individuals -> Fitness**
5. **Selection** (based on “fitness”)
6. **Production of next generation with GA operations** (crossover/mutation)
7. **Check convergence criteria. If not converged, go to step 4.**

Genetic Algorithms

Convergence criteria:

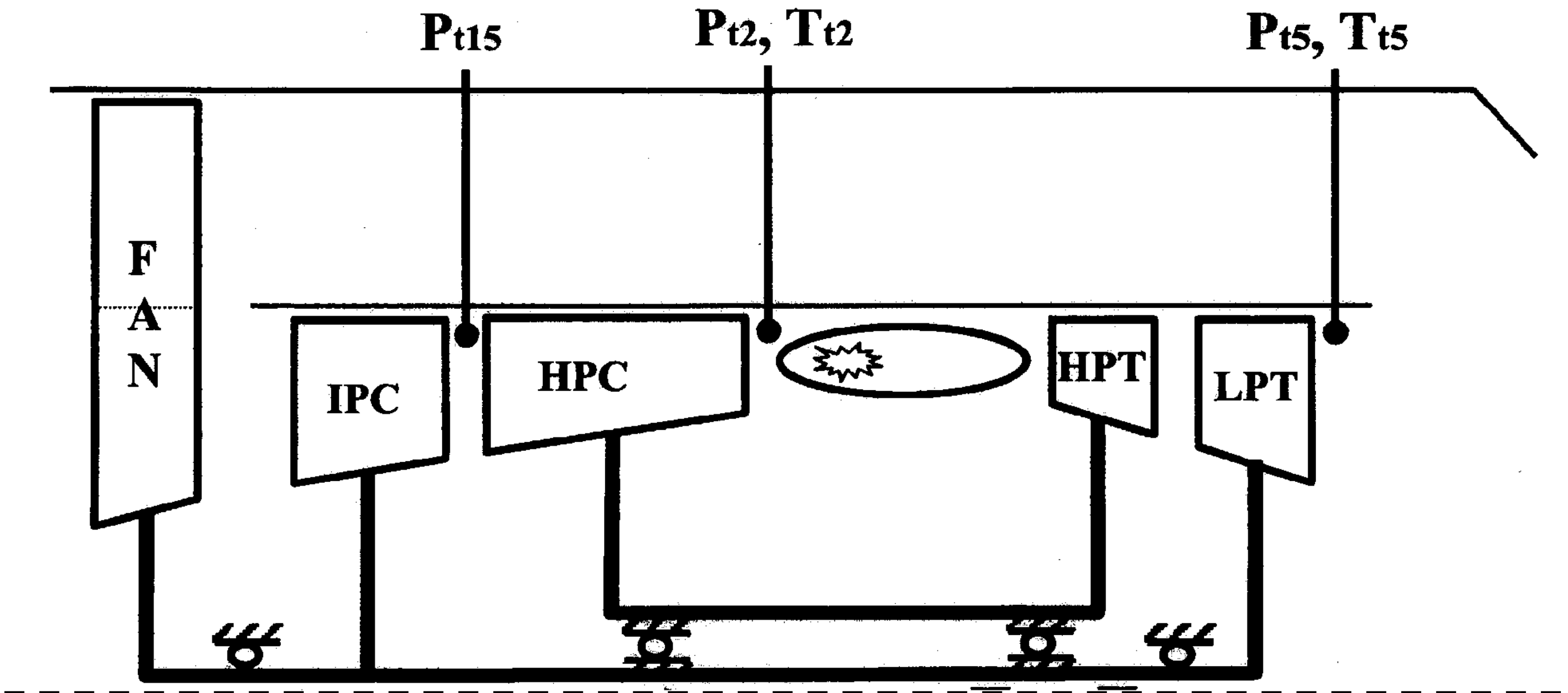
- **Fitness > set threshold for the fitness**
- **Number of Generations > maximum number of generations**

Genetic Algorithms

Distinctive features :

- **No derivatives**
- **Constraints** (penalty functions)
- **Global search -- to avoid getting stuck at a local minimum**
- **Different from random search**
- **Different from conventional optimisation**
- **Probabilistic rather than deterministic for string generation**
- **Balance between exploiting the best solution and exploring the search space**

Example of GA application



A two-shaft turbofan engine

Example of GA application

Engine instrumentation set

Parameter	Unit	Meanings
N_{HP}	%	Relative HP shaft speed
N_{LP}	%	Relative LP shaft speed
m_f	kg/s	Fuel flow rate
P_{t2}	kPa	HP compressor exit total pressure
T_{t2}	K	HP compressor exit total temperature
P_{t5}	kPa	LP turbine exit pressure
T_{t5}	K	LP turbine exit temperature

Example of GA application

Objection Function:

$$OF = \sum_{i=1}^{N} \left| \frac{Z_{degraded,i} - Z_{model,i}}{Z_{degraded.i}} \right|$$

GA Fitness:

$$FITNESS = \frac{1}{1 + OF}$$

Example of GA application

GA Parameter Selection

- **Population size: 100**
- **Probability of crossover: 60%**
- **Probability of mutation: 40%**
- **Single component faults – fault cases**
- **Search space: [-5.0 or 0.0, 5.0]**

Example of GA application

Assumed fault cases

Fault Class	Faulty Component	Component Parameters		Implanted Faults
CFC1	IP compressor	η_{IPC}	Thermal efficiency	
		Γ_{IPC}	Flow capacity	
CFC2	HP compressor	η_{HPC}	Thermal efficiency	-2%
		Γ_{HPC}	Flow capacity	-4%
CFC3	Combustor	η_b	Combustion efficiency	
CFC4	HP turbine	η_{HPT}	Thermal efficiency	
		Γ_{HPT}	Flow capacity	
CFC5	LP turbine	η_{LPT}	Thermal efficiency	
		Γ_{LPT}	Flow capacity	
CFC6	HP shaft bearings	C_{HPB}	Friction coefficient	
CFC7	LP shaft bearings	C_{LPB}	Friction coefficient	

Example of GA application

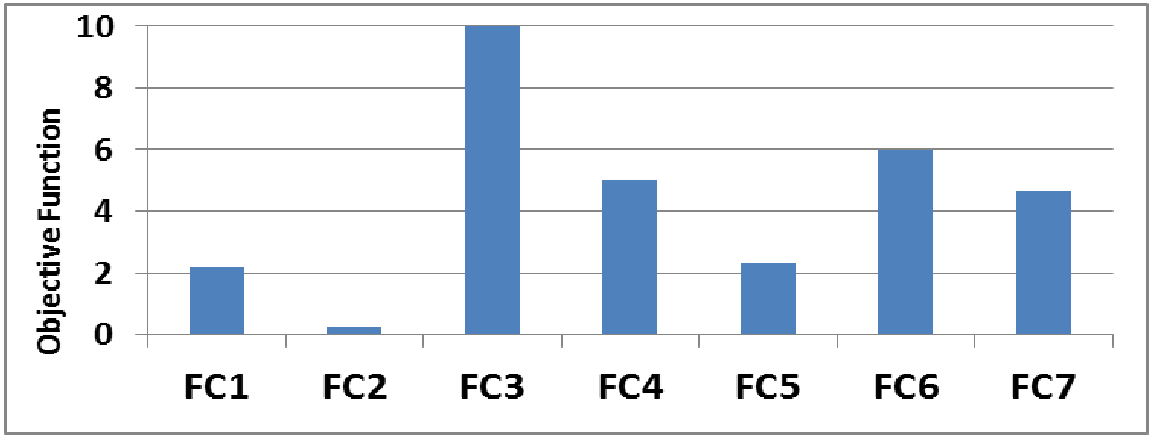
Diagnostic results with GA

Component Fault Cases (CFC)	Component parameters & implanted fault	Search space	Search results		
			Predicted degradation	Objective function	Fitness
CFC1	ΔηIPC ΔΓIPC	[-5% - 0%] [-5% - 0%]		2.174	0.315
CFC2	ΔηHPC -2% ΔΓHPC -4%	[-5% - 0%] [-5% - 0%]	-1.94% -3.86%	0.242	0.805
CFC3	Δηb	[-5% - 0%]		68.1	0.01
CFC4	ΔηHPT ΔΓHPT	[-5% - 0%] [-5% - +5%]		5.016	0.17
CFC5	ΔηLPT ΔΓLPT	[-5% - 0%] [-5% - +5%]		2.331	0.30
CFC6	CHPB	[0 – 1.08]		6.010	0.14
CFC7	CLPB	[0 – 1.20]		4.673	0.17

Example of GA application

Diagnostic results with GA

Objective Function:



Fitness: