Optimisation

Optimisation techniques

Optimisation techniques

- Calculus-Based

Optimisation techniques

- Calculus-Based
  - Indirect

Optimisation techniques

- Calculus-Based
  - Indirect
  - Direct

Optimisation techniques

- Calculus-Based
  - Indirect
  - Direct
- Enumerative

Optimisation techniques

- Calculus-Based
  - Indirect
  - Direct
- Enumerative
- Stochastic

Optimisation techniques

- Calculus-Based
  - Indirect
  - Direct
- Enumerative
- Stochastic
  - Stochastic hill-climbing

Optimisation techniques

- Calculus-Based
  - Indirect
  - Direct
- Enumerative
- Stochastic
  - Stochastic hill-climbing
  - Simulated annealing

Optimisation techniques

- Calculus-Based
  - Indirect
  - Direct
- Enumerative
- Stochastic
  - Stochastic hill-climbing
  - Simulated annealing
  - Genetic algorithms

Optimisation techniques

- Calculus-Based
  - Indirect
  - Direct
- Enumerative
- Stochastic
  - Stochastic hill-climbing
  - Simulated annealing
  - Genetic algorithms

D.E. Goldberg, Genetic Algorithms in Search, Optimization & Machine Learning, Addison Wesley, 1989

Evolution (biology)

Britannica Enciclopædya

> “theory in biology postulating that the various types of plants, animals, and other living things on Earth have their origin in other preexisting types and that the distinguishable differences are due to modifications in successive generations.”

http://www.britannica.com

Evolution (optimisation)

- Evolution as an optimisation scheme
- The different kinds of species evolve to the optimal adaptation to the surrounding environment. Thus, evolution is an 'algorithm' that searchs for the best solution creating a set of individuals (a generation), it decides which individuals are the best ones, and, by means of crossover, keeps the good genetic characteristics for the next generation – that will be closer to the optimal solution – and removes the individuals with worst genetic content

Biology $\leftrightarrow$ GA

- Environment $\leftrightarrow$ Objective function
- Individual $\leftrightarrow$ Set of parameters
- Generation $\leftrightarrow$ Set of solutions

How a GA works

How a GA works (Kangaroos)

How a GA works (Kangaroos)

1. Highest peak in Himalaya?

How a GA works (Kangaroos)

1. Highest peak in Himalaya?
2. Parachute a lot of kangaroos at random places

How a GA works (Kangaroos)

1. Highest peak in Himalaya?
2. Parachute a lot of kangaroos at random places
3. Kangaroos wander and create more kangaroos

How a GA works (Kangaroos)

1. Highest peak in Himalaya?
2. Parachute a lot of kangaroos at random places
3. Kangaroos wander and create more kangaroos
4. Each generation, remove low-altitude kangaroos

How a GA works (Kangaroos)

1. Highest peak in Himalaya?
2. Parachute a lot of kangaroos at random places
3. Kangaroos wander and create more kangaroos
4. Each generation, remove low-altitude kangaroos
5. At the very end, there are only kangaroos on Mt. Everest and (perhaps) on Mt. K2

Particularities of GAs

Particularities of GAs

- GAs do not work directly with the parameters. They work with a codification (an encoding scheme) of the parameters (the chromosome)

Particularities of GAs

- GAs do not work directly with the parameters. They work with a codification (an encoding scheme) of the parameters (the chromosome)
- Whereas most methods employ a single solution which evolves to reach the local optimum, GAs work on a population of many possible solutions simultaneously

Particularities of GAs

- GAs do not work directly with the parameters. They work with a codification (an encoding scheme) of the parameters (the chromosome)
- Whereas most methods employ a single solution which evolves to reach the local optimum, GAs work on a population of many possible solutions simultaneously
- GAs only need the objective function to determine how fit an individual is. Neither derivatives nor other auxiliary knowledge are required

Particularities of GAs

- GAs do not work directly with the parameters. They work with a codification (an encoding scheme) of the parameters (the chromosome)
- Whereas most methods employ a single solution which evolves to reach the local optimum, GAs work on a population of many possible solutions simultaneously
- GAs only need the objective function to determine how fit an individual is. Neither derivatives nor other auxiliary knowledge are required
- GAs use probabilistic rules to evolve (randomness does not mean directionless!)

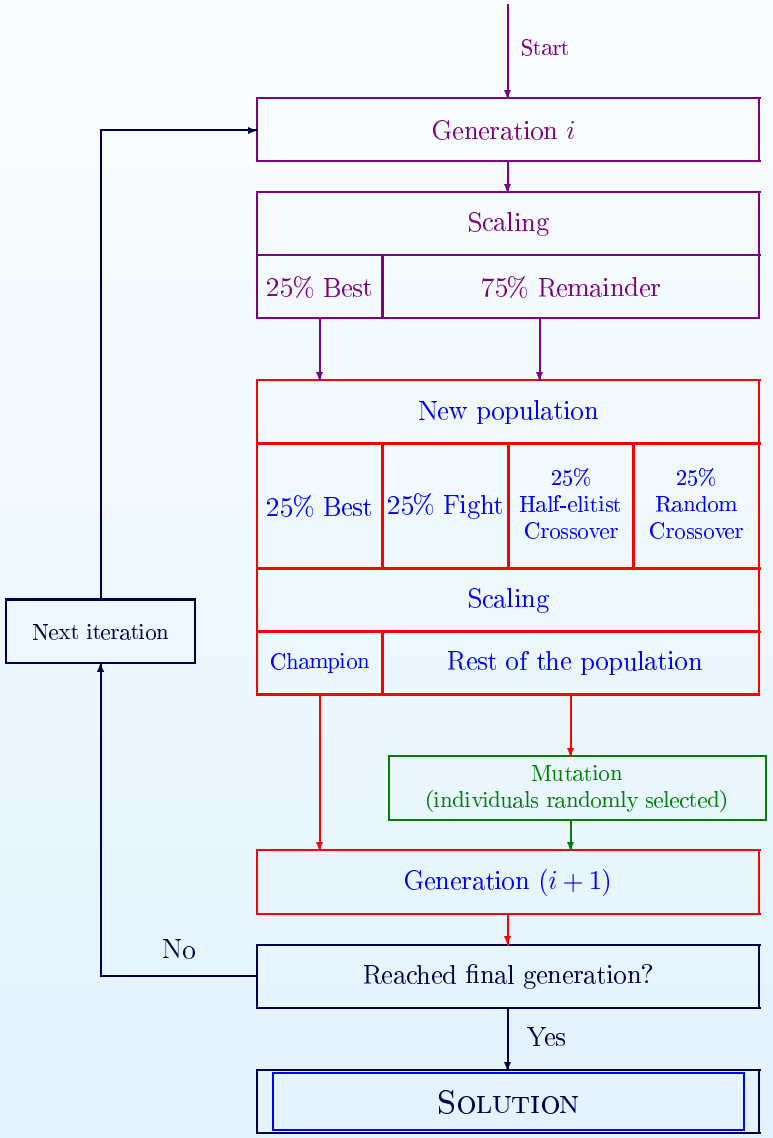
Examples of encoding

- Range $[0, 1]$
  - Binary representation
  - Discretise the range in $2^n - 1$ parts
  - n is the length of the chromosome
  - each 1 or 0 is a gene
- Travel Salesman Problem (TSP)
  - Each city one number $[1, 2, 3, \dots]$
  - The position in the gene determines the travelling order $1 - 3 - 25 - 78 - \dots$

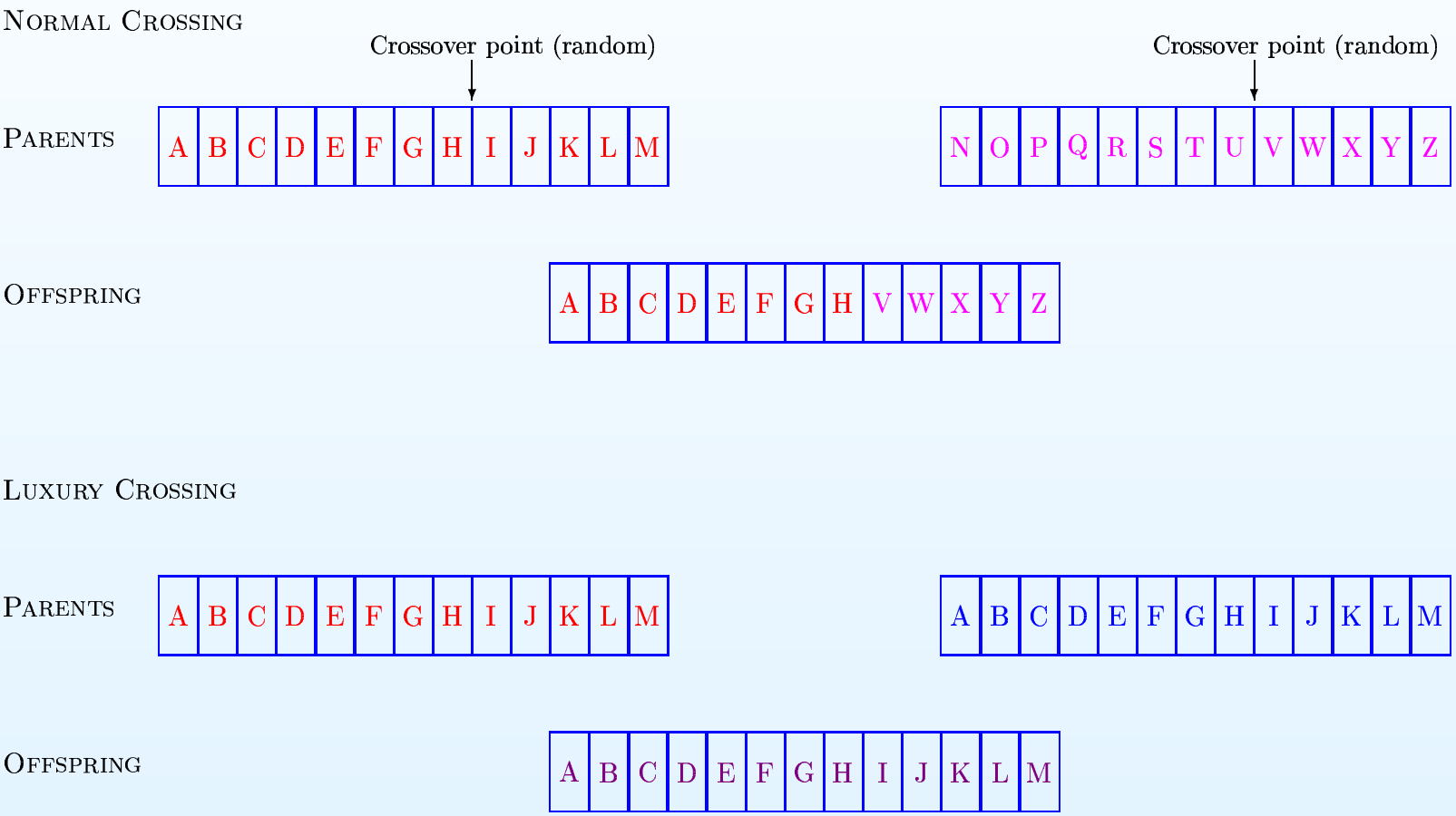
Objective function

$$\chi^2 = \sum_j \frac{\left(\mathcal{M}_j^{\text{exp}} - \mathcal{M}_j^{\text{th}}\right)^2}{\left(\Delta\mathcal{M}_j^{\text{exp}}\right)^2}$$

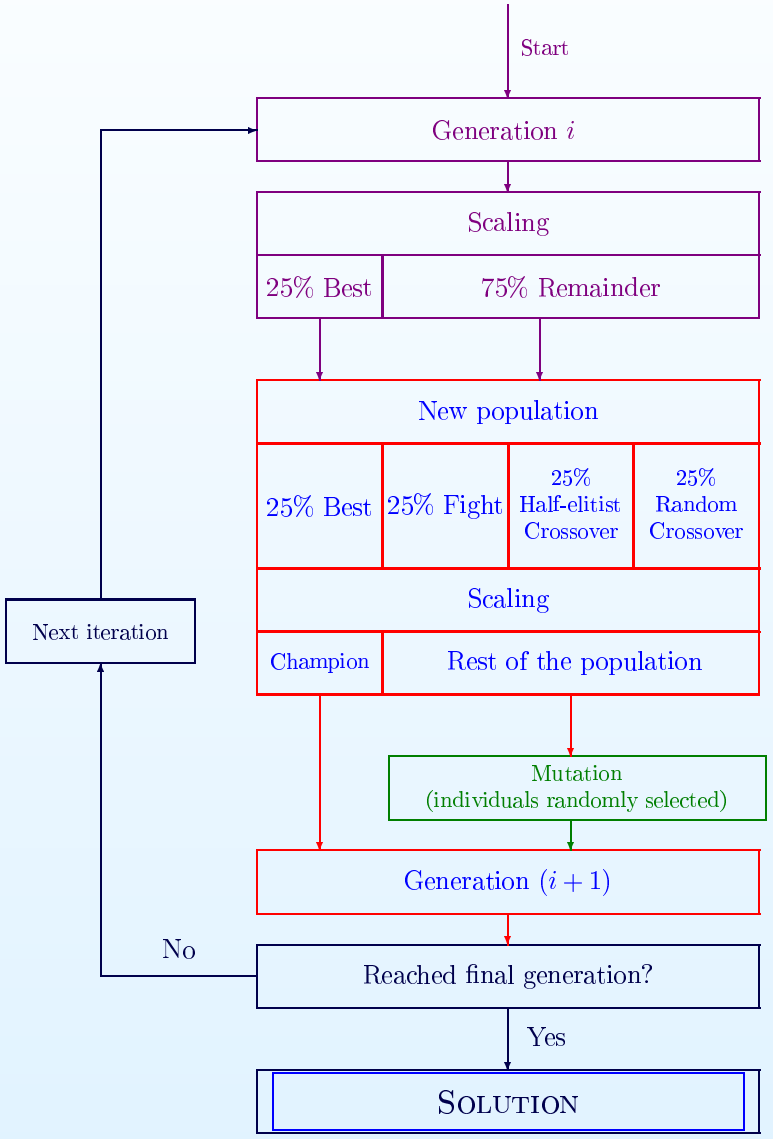
Example of a genetic algorithm



Crossover

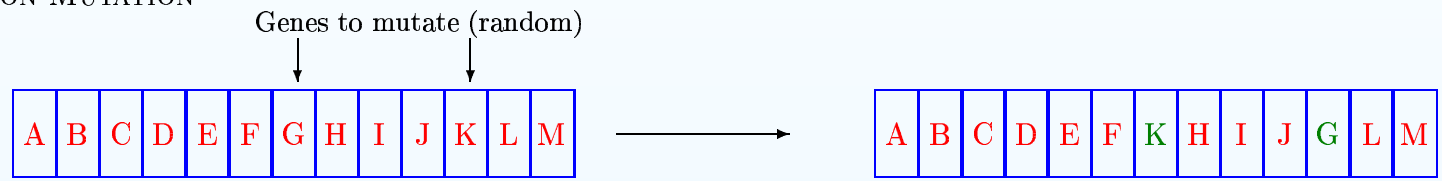

Normal Crossing
Crossover point (random)
Crossover point (random)
Parents
A B C D E F G H I J K L M
N O P Q R S T U V W X Y Z
Offspring
A B C D E F G H V W X Y Z
Luxury Crossing
Parents
A B C D E F G H I J K L M
A B C D E F G H I J K L M
Offspring
A B C D E F G H I J K L M

Example of a genetic algorithm

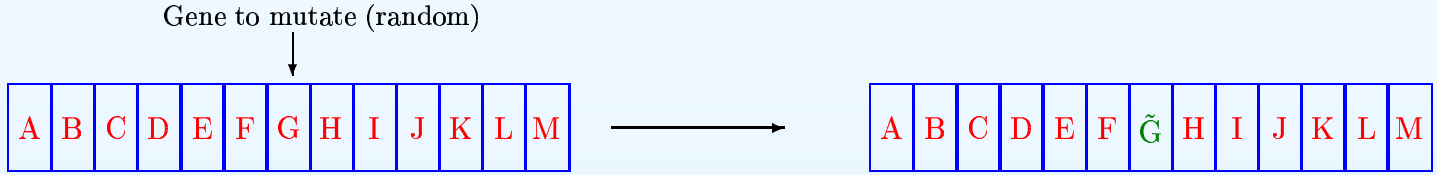


Mutation

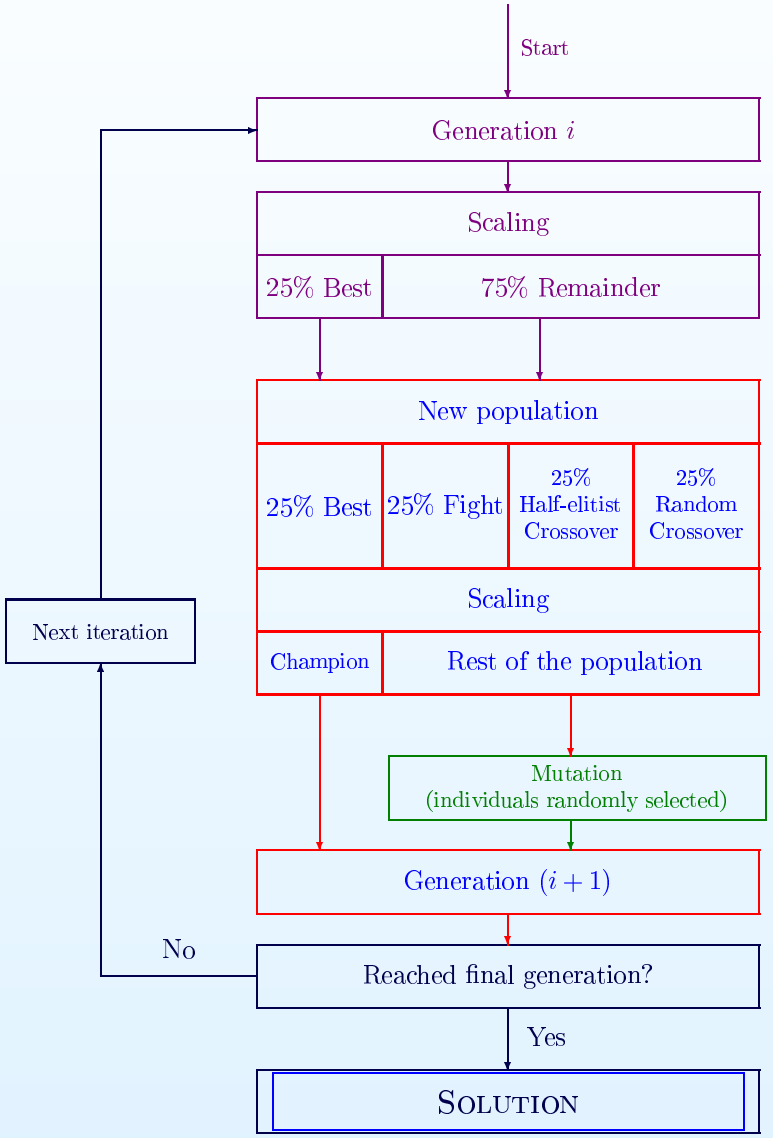
PERMUTATION MUTATION



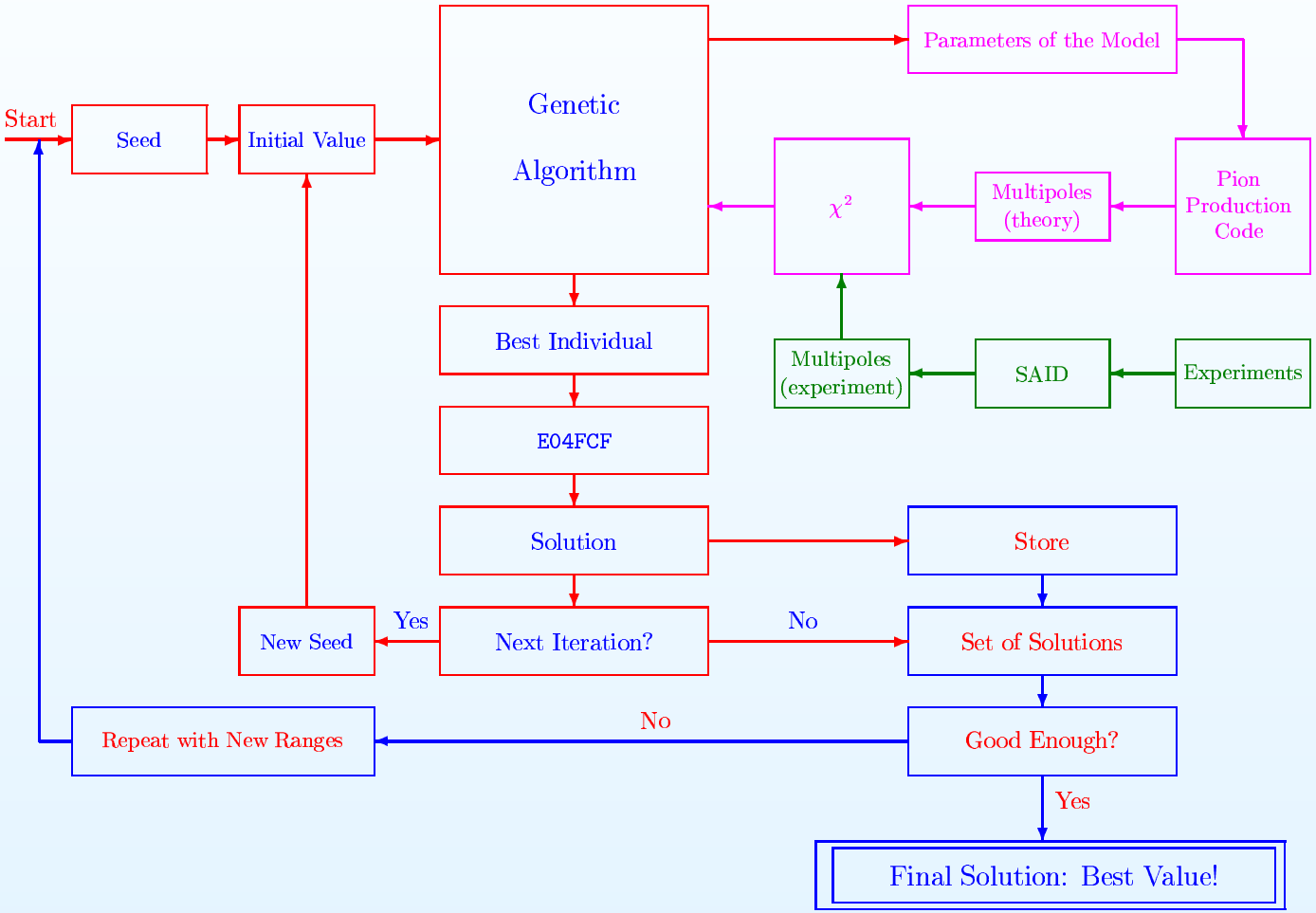
SHIFT MUTATION



Example of a genetic algorithm



Minimisation scheme



Minimisation

The variation range of each parameter is linearly mapped into the $[0,1]$ interval for the genetic algorithm and into $(-\infty,+\infty)$ for the E04FCF routine

$$x_j = \arcsin\left[\frac{\lambda_j - \lambda_j^{max}}{\lambda_j^{max} - \lambda_j^{min} - 1}\right]$$

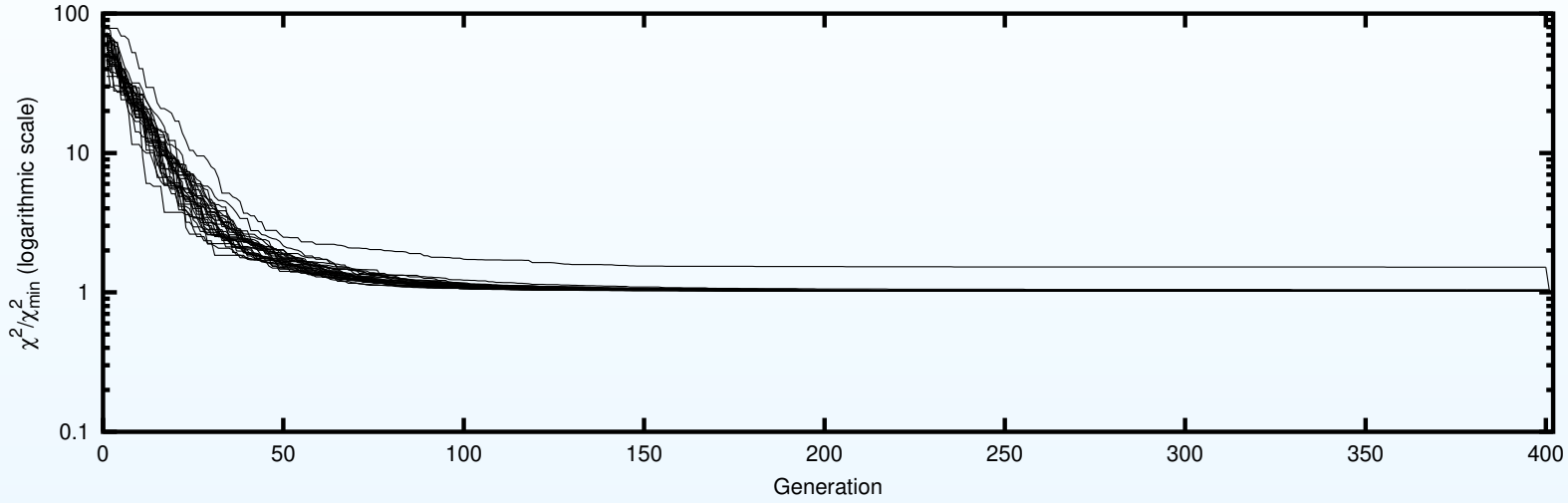
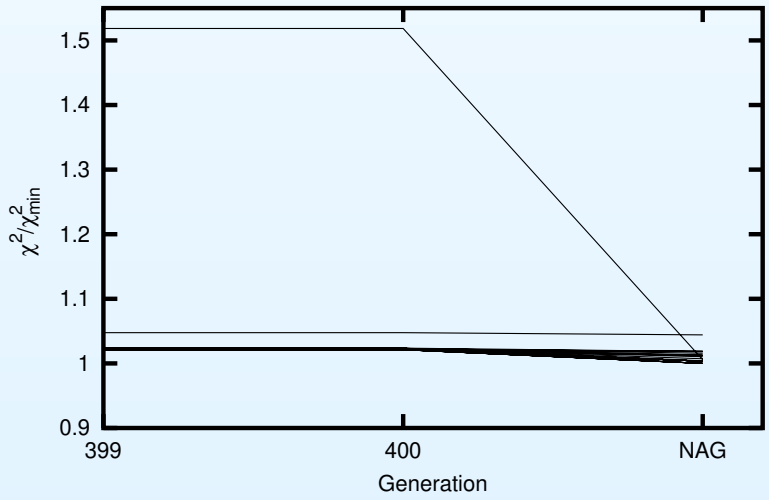
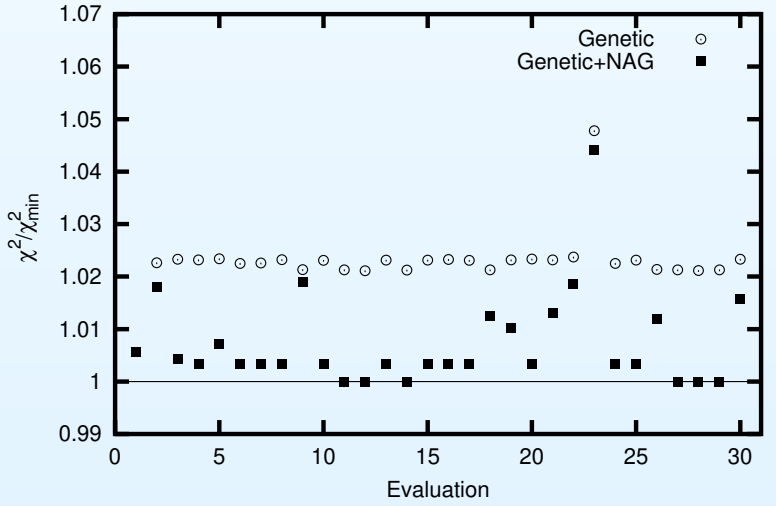
where λ_j is the model parameter, x_j is the mapping of λ_j into $(-\infty,+\infty)$, λ_j^{max} is the highest value of the variation range, and λ_j^{min} the lowest value. Before we run the E04FCF routine we normalize the χ^2 to unity which the minimum search

E04FCF

Finds an unconstrained minimum of a sum of squares

$$\text{Minimise } F(x_1, \ldots, x_n) = \sum_{j=1}^{m} |f_j(x_1, \ldots, x_n)|^2$$

of m nonlinear functions in n variables ($m \geq n$). This algorithm does not require to know derivatives.

From a starting point $x_1^{(1)}, \ldots, x_n^{(1)}$ the routine applies Newton method in order to reach the minimum. The Newton method uses a finite-difference approximation to the Hessian matrix to define the search direction

Evolution of the minimisation