Probabilistic algorithms

Limitations of the algorithms based on the theory of Mathematical Programming:

- Do not ensure that the global minimum was achieved (unless the problem is convex). Several attempts should be made, changing the initial guess;
- Do not allow to work with discrete variables – information of the derivatives is either useless or not well defined (solution space is disjoint and disconnected); introduction of multiple local minima

Solution ➡ Probabilistic Algorithms

- Greater probability to obtain the global minimum;
- Can handle discrete variables
- Based on observed phenomena in nature
- Random search process guided by probabilistic decisions

Simulated Annealing

Most widely used probabilistic algorithms
- "Simulated Annealing"
- Genetic algorithms

"Simulated Annealing"

Based on a phenomenon from statistical mechanics that is related with balance of a large number of atoms in solids and liquids at a certain temperature solidification of metals and formation of crystals

Rapid cooling ➡ solid state is little stable (atoms adopt positions of local minima in terms of potential energy in the metal lattice). In order to obtain a more stable state of energy (global minimum) ➡ Annealing: the metal is re-heated to high temperatures and cooled slowly (so the atoms can accommodate to find stable locations of global minima potential energy)

Simulated Annealing

Based on the “Metropólis” algorithm

Given T (temperature) ➡ random perturbation of the atoms position ➡ calculate ΔE:

• If $\Delta E<0$ accept the atom’s new configuration;

• If $\Delta E>=0$ decision is based on the following probabilistic function that computes the probability of acceptance:

$$P(\Delta E) = e^{\left(\frac{-\Delta E}{k_B T}\right)}$$

k_B –Boltzmann constant

The decision is obtained by picking a number in a random way ρ in (0,1) and comparing with $P(\Delta E)$:

If $\rho < P(\Delta E)$ ➡ the configuration is accepted

If $\rho > P(\Delta E)$ ➡ the configuration is rejected

Simulated Annealing

Note that:

• If T is high ➡ $P(\Delta E)$ is close to 1;

• If T is near zero $P(\Delta E)$ is very small;

Thus, at each temperature a set of atomic structures is generated by the random perturbation of the position until a “thermal equilibrium” condition is achieved (stable state). The temperature is lowered and the iterations are repeated. The steps are repeated iteratively until the temperature is slowly reduced so to achieve the minimal state of energy.

Analogy with the mathematical problem of optimization

Energy states ➡ Objective function

Atoms configuration ➡ Project variables **x**

T Temperature ➡ parameter to control convergence rate

Only function values are employed

Simulated Annealing

Method's performance

- Temperature T_0;
- Update of T;
- Number of iterations (combinations of project variables) necessary to achieve "thermal equilibrium", before reducing T;

"Cooling Schedule"

If T_0 is low ➡ low probability of reaching the minimum

Choice of T_0:

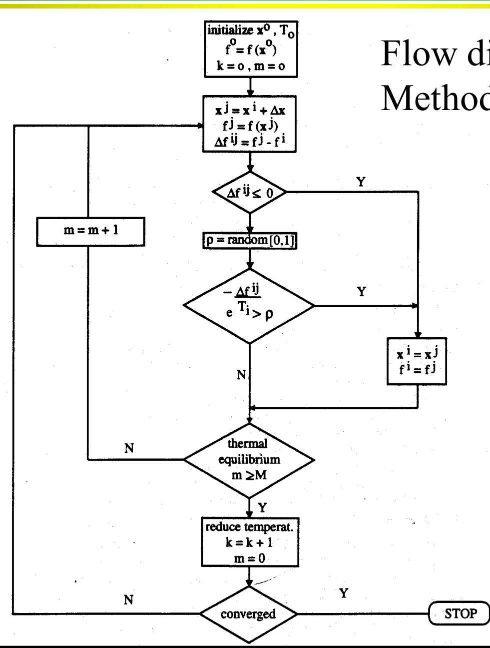
$$P(\Delta E) = 0{,}95 = e^{\left(\frac{-\overline{\Delta f}}{T_0}\right)} \Rightarrow T_0 = \frac{\overline{\Delta f}}{\ln(1/0{,}95)}$$

Temperature update rule:

- $T_{k+1} = \alpha T_k \quad k = 0,1,2,\ldots K \quad 0{,}5 \le \alpha \le 0{,}95$
- Divide $[0, T_0]$ in K steps: $T_k = \dfrac{K-k}{K} T_0 \quad k = 0,1,2,\ldots K$

Simulated Annealing

Flow diagram of the Method



Genetic Algorithm

Obtained from Biology ➡ Darwin's Theory (survival of the fittest). Through generations, useful features to survival are passed on to individuals of the next generations. These features are encoded in the chromosomes. The genetic mechanism of exchange of random information amongst chromosomes of breeding parents are based on the following operations: reproduction, crossover, mutation and inversion of the chromosome code.

Genetic algorithms simulate the mechanisms of natural genetics in optimization problems. Chromosomal code is represented by a "word" (genotype). The operations involve random exchange of number locations in a word. Only function values are employed.

Genetic Algorithm

Representation of the combination of project variables by words with bits that represent the chromosomes. For example:

$$\{x_1, x_2, x_3, x_4\} = \{6,5,3,11\} = \underbrace{0110}_{x_1}\underbrace{101}_{x_2}\underbrace{11}_{x_3}\underbrace{1011}_{x_4}$$

Ideal for discrete or integer variables, in the case of continuous variables ➡ large number of bits for the representation (depends on the desired precision). In this case, the *m* binary digits necessary to represent x_i in an interval delimited by x^L_i and x^U_i with precision x^{incr} is such that:

$$\{x_i^L \leq x_i \leq x_i^U\} \Rightarrow 2^m \geq \left(\left(x_i^U - x_i^L\right)/\,x^{incr} + 1\right)$$

Genetic Algorithm

Work with population of words (or chromosomes) and not with a single point in the design domain ➡ Advantageous parallelized implementation. The result of genetic algorithm is a population of good project variables (“words”).

Example of a sequence of operations in a genetic algorithm:

1. Population size is chosen and the value of each individual is randomly decided (0s and 1s for bits).

2. Reproduction: individuals with a good objective function value are copied to form a new population (Darwin’s Theory), i.e., its probability of being chosen is increased relative to the other individuals of the population. The new population will have multiple copies of the most resilient individuals.

Genetic Algorithm

3. Crossover: individuals of the new population are grouped randomly in pairs for crossover. An integer k is selected between 1 and L-1, where L is the length of the word, and then (e.g., L=9 and k=5):

parent 1: 01101||0111
parent 2: 01001||0001

generates

generated word: 01101||0001
generated word: 2: 01001||0111

Crossover of “a point”. Other possibilities: crossover of “two points” and “multiple points”.

4. Mutation: randomly select an individual and alter its value from 0 to 1 or vice-versa. Avoids uniformity, i.e., the occurrence of many similar individuals within a population, that occurs in the reproduction stage. Otherwise, the chance of finding the best solutions is reduced. The effect in the performance of the algorithm is minor (1 mutation in a 1000 bit operations).