Genetic Algorithm

Whatever can be translated into a bitstring (text, music, images, …) can be exposed to a genetic algorithm, which then performs a search for an optimal solution. The procedure mimics Darwin's 'survival of the fittest' idea. One starts with a ***diverse*** initial population of guesses and checks for the fitness of an arbitrarily chosen guess (e.g., 0010111). This fitness could e.g., be computed as the number of differences to the optimal bitstring the so-called Hamilton difference (H=4). Then comes a **test** which eliminates all candidates with a fitness lower than a threshold S. This is done for all candidates of the original population. Then a **crossover** of all surviving candidates is performed, which means that strings are cut into two parts and parts from different candidates are stick together to form a new candidate. This mimics the break-up of cells into haploid cells and sexual reproduction by forming a new cell, the nucleus of human babies. Programming this procedure shows that even with repeated crossover an optimal solution might not be approached – the optimisation process can get stuck (only a self-similar set of candidates remains in the pool, e.g., 'feudal nobility'). To get out of this impasse **mutation** can occur, which means that (one or more) bits in some candidate strings are arbitrarily switched. For human reproduction the mutation probability of this event has been estimated to be extremely low, around 10^{-13}. Since the evolution of our species took a very long time, it can also be assumed that this (non-zero!) mutation rate is also the result of evolutionary selection. The termination criterion can be the optimal solution – or the solutions close enough to it. As long as they are not met, the procedure is repeated.

Evidently the genetic algorithm is a search algorithm and as such it is used in a variety of technical applications. Its core consists of **a variety producing procedure** (e.g., first an initial variety and during the repeated loops a crossover mechanism) and second **a test** (e.g., being close enough to a *predefined* optimal solution). As such the genetic algorithm is the central formal metaphor of **Artificial Intelligence** (compare Herbert Simon, 1969, The Sciences of the Artificial). In the form presented (with a predefined optimal bitstring) today's AL lingo usually calls it '**supervised learning**'. To describe the many existing statistical methods of discovering clusters in data without a predefined goal bitstring AL jargon uses the term '**non-supervised learning**'. Note that the well-known **Principal Component Analysis** (invented by Karl Pearson, 1901) arrives a most probable *linear dynamic model* of a dataset only due to its predetermined goal to *only look for linear relationships*. In this case the assumption of linearity replaces the immediate assumption of an optimal solution! It thus remains in the area of supervised learning and at the same shows that the distinction between supervised and not supervised is not useful. It makes more sense to dig deeper into the process of variety production by mimicking pattern recognition and interpretation taking place in the human brain (early example John von Neumann).

Genetic Algorithm