A New Research on Numerical Techniques and Apply it in Design of Genetical Algorithm

Mohammad Kiani and Mehdi Kiani

Department of Mechanical Engineering, Jawaharl Nehru Technological University Hydarabad, India.

ABSTRACT

The genetic algorithm is a model of machine learning which derive its behavior from a metaphor of the processes of evolution in nature. Heating and cooling the material affects both the temperature and the thermodynamic free energy. While the same amount of cooling brings the same amount of decrease in temperature it will bring a bigger or smaller decrease in the thermodynamic free energy depending on the rate that it occurs, with a slower rate producing a bigger decrease. An empirical research into solvent the cession model using Genetic Algorithm and Simulated bake is offering. Various parameters efficacious the algorithms are studied and their penetration on convergence to the final optimum solution is shown.

INTRODUCTION

The genetic algorithm is a model of machine learning which derive its behavior from a metaphor of the processes of evolution in nature (Chu, 2009). This is done by the creation within a machine of a population or individuals represented by chromosomes, in essence a set of character strings that are analogous to the base-4 chromosomes that we see in our own dna (De Vicente, Juan, 2003). The individuals in the population then go through a process of evolution. The evolution usually starts from a population of randomly generated individuals, and is an iterative process, with the population in each iteration called a generation. In each generation, the fitness of every individual in the population is evaluated; the fitness is usually the value of the objective function in the optimization problem being solved. The more fit individuals are stochastically selected from the current population, and each individual's genome is modified (recombined and possibly randomly mutated) to form a new generation. The new generation of candidate solutions is then used in the next iteration of the algorithm. Commonly, the algorithm terminates when either a maximum number of generations has been produced, or a satisfactory fitness level has been reached for the population. Genetic algorithms are simple to implement, but their behavior is difficult to understand. In particular it is difficult to understand why these algorithms frequently succeed at generating solutions of high fitness when applied to practical problems. The building block hypothesis (bbh) consists of:

- A description of a heuristic that performs adaptation by identifying and recombining "building blocks", i.e. low order, low defining-length schemata with above average fitness.
- A hypothesis that a genetic algorithm performs adaptation by implicitly and efficiently implementing this heuristic (Goldberg, David, 2002; Koza, John, 1992).

There are limitations of the use of a genetic algorithm compared to alternative optimization algorithms:

1. Repeated fitness function evaluation for complex problems is often the most prohibitive and limiting segment of artificial evolutionary algorithms. Finding the optimal solution to complex high-dimensional, multimodal problems often requires very expensive fitness function evaluations. In real world problems such as structural optimization problems, a single function evaluation may require several hours to several days of complete simulation. Typical optimization methods cannot deal with such types of problem. In this case, it may be necessary to forgo an exact evaluation and use an approximated fitness that is computationally efficient. It is apparent that amalgamation of approximate models may be one of the most promising approaches to convincingly use GA to solve complex real life problems (Holland, John, 1992).
2. Genetic algorithms do not scale well with complexity. That is, where the number of elements which are exposed to mutation is large there is often an exponential increase in search space size. This makes it extremely
difficult to use the technique on problems such as designing an engine, a house or plane. In order to make such problems tractable to evolutionary search, they must be broken down into the simplest representation possible. Hence we typically see evolutionary algorithms encoding designs for fan blades instead of engines, building shapes instead of detailed construction plans, airfoils instead of whole aircraft designs. The second problem of complexity is the issue of how to protect parts that have evolved to represent good solutions from further destructive mutation, particularly when their fitness assessment requires them to combine well with other parts.

3. The "better" solution is only in comparison to other solutions. As a result, the stop criterion is not clear in every problem.

The name and inspiration come from annealing in metallurgy, a technique involving heating and controlled cooling of a material to increase the size of its crystals and reduce their defects. Both are attributes of the material that depend on its thermodynamic free energy. Heating and cooling the material affects both the temperature and the thermodynamic free energy. While the same amount of cooling brings the same amount of decrease in temperature it will bring a bigger or smaller decrease in the thermodynamic free energy depending on the rate that it occurs, with a slower rate producing a bigger decrease. This notion of slow cooling is implemented in the Simulated Annealing algorithm as a slow decrease in the probability of accepting worse solutions as it explores the solution space.

![Genetic Algorithm](image)

**Fig. 1:** Genetic algorithm.

**Acceptance Function:**

Let's take a look at how the algorithm decides which solutions to accept so we can better understand how it's able to avoid these local optimums. First we check if the neighbor solution is better than our current solution. If it is, we accept it unconditionally. If however, the neighbor solution isn't better we need to consider a couple of factors. Firstly, how much worse the neighbor solution is; and secondly, how high the current 'temperature' of our system is. At high temperatures the system is more likely accept solutions that are worse.

Accepting worse solutions is a fundamental property of metaheuristics because it allows for a more extensive search for the optimal solution. The method was independently described by Scott Kirkpatrick, C. Daniel Gelatt and Mario P. Vecchi in 1983, and by Vlado Černý in 1985. The method is an adaptation of the Metropolis–Hastings algorithm, a Monte Carlo method to generate sample states of a thermodynamic system, invented by M.N. Rosenbluth and published in a paper by N. Metropolis et al. (1953).
Fig. 2: Block diagram of LPFGs design algorithm compounded with Simulated Annealing Algorithm and Steepest Descent Algorithm.

Fig. 3: Quantum annealing.

The Hungarian mathematician D.Köling equable an indispensable theorem for the expansion of the “Hungarian procedure” to solve this pattern. The difficulty can also be formulated as an whole number - programming model and dissolve by techniques such as “Branch-and- limit technique”. Reference (Cha, Sung-
Hyuk, 2009) states that the Hungarian algorithm for resolving the cession pattern is more efficient than branch line-and-bound algorithms. This paper attempts to unravel the same pattern using two non-traditional techniques: Genetic Algorithm and Simulated bake (Vose, Michael, 1999). It is basically an empirical research into the different parameters affecting these three algorithms and adapting them to difficulty. These three approaches are discussed one by one.

**Algorithm Approach:**

Optimization algorithms based on the mechanics of normal genetics. Were prime forecasted by bob Holland and were subsequently expand by different researches. Each possible solution is encoded in the form of a field and a crowd of strings is created which is further processed by three operators: breeding, intersecting, and Mutation. Breeding is a process in which individual field are copied according to their fitness subordinate (Here the fitness subordinate is taken to be the total cost subordinate). Intersecting is the process of interchange the content of two field at some point(s) with a eventuality. Finish, Mutation is the process of flipping the value at a specific place in a field with a very low eventuality. A more encyclopedic remedy of GA can be found in (De Vicente, Juan, 2003; Fraser, Alex S., 1957). Now, for adapting GA to our difficulty, it is intransitive that we expand an encoding plot. Consider the case when N=4 and let us assume that automaton M1 is assigned to man m1, automaton M2 to man m2, and automaton M3 to man m3 as shown: After encoding of the field, the crowd choice for intersecting is done by “Binary tournament choice” method. Here s=2 field are randomly chosen and compared, the best one being selected for parenthood.

This is frequent M times where M is the size of the crowd. Reference (Fraser, Alex S., 1957) also cites a procedure for begetting the parent field which are then ready for intersecting. Here simple intersecting will not work; instead we choose the procedure of Partially Matched intersecting (PMX) which was initially expansion for tackling the “Traveling Salesman difficulty” (De Vicente, Juan, 2003). The notion of PMX can be perception by considering an instance: Now the section between the selected intersecting points is swapped and the remains of the values are changed according to the above rule (this means 1 in the section outside the two intersecting points is replaced by 4 and 2 in the section outside the two intersecting points is superseded by 5). Intersecting, we have a family of parent crowd and children crowd out of which we are to select the crowd for next repeat. Here we have a choice of change the crowd size at each repeat. We must memorize the variety in crowd or else it may lead to unripe convergence to a solution which may not be optimal. One procedure of selecting the crowd may be to arrange the entire crowd in ascensional order of their objective subordinate value (the field that decodes to lowest total cost of cession will have the highest objective subordinate value) and choose a predetermined number of individual field from each batch i.e., from those that are above medium, from those around the medium, and from those below the medium. This threshold can be set by using the notion of mean and standard deflection applied to the crowd.

For example, if we presume the field values to be normally distributed with mean value μ and standard deflection σ, we divide the crowd into four batches: those having values above μ + 3*σ, those having values between μ and μ - 3*σ, and those having values between μ - 3*σ and μ - 3*σ, and those having values lesser than μ - 3*σ. In this way the variety in crowd is maintained.

Another aspect is that the field with the best objective subordinate value at each repeat is stored in a separate array and subsequently compared with the best field of the crowd at next repeat. In this way, the best field cannot defect.

Also note that we are not using spur but a slight variant of it (Inversion) by choosing two accidental spots in a field and interchange the corresponding values at that situation. Inversion is permissible only when the sum of the costs at these situation before interchange is greater than the sum of costs communication with these situation after interchange.

Two cases were accomplishment: one in which Inversion was used and another in which Inversion was not used. In both the cases, the response converged to the final optimized value. On an average, there was not much difference in the number of repeat required to reach the final value in both the cases.

**Simulated Annealing Method:**

The simulation bake method the process of slow chilled of molten metal to arrive the minimum subordinate value in a minimization difficulty. It is a point-by-point procedure. The algorithm begins with an primary point and a high temperature T. A second point is taken at accidental in the adjacency of the primary point and the difference in the subordinate values (ΔE) at these two points is calculation. The second spot is selection according to the Metropolis algorithm which states that if the second spot has a little subordinate value, the spot is accepted; otherwise the spot is accepted with a eventuality exp. (-ΔE / T). This completes one repeat of the simulated annealing method. In the next descendant, another point is created at accidental in the neighborhood of the stream point and the Metropolis algorithm is used to accept or disprove the point. In order to simulate the thermic balance at every heat, a number of points (m) is usually tested at a specific temperature before decrease the heat. When the response converges to a specific value, we store the corresponding field in a apart array.
Then we proceed with our search again in the area of field created by the second procedure. Comparison with the field which was outset stored in an apart array.

![Graph](image)

**Fig. 4:** Global minimum of both inversion.

The minimum of these two (the one with lesser subordinate value) is selected as the final response. The consequential parameters affecting simulated bake are the number of repeat \( (m) \) at each step and the cooling program. The total number of repeat is commensurate to \( m \) as well as the rate of variation of temperature. The cooling program is based on Newton’s law of cooling. This model of cooling can be compared to the evacuation of an outset charged capacitor in a RC circuit as they both follow exponential rottenness law. For all feasible purposes, it is presumption that the capacitor is fully disembarkation at \( t=3*RC \). Hence, in our program we also ran our schedule from \( T_{\text{max}}=600 \) to \( T_{\text{min}} \) around \( 600*\exp(-3) \), keeping the number of repeat \( m \) fixed \((=30)\).

\( T_{\text{max}} \) is commonly computed by calculation the medium of computing values at several points. The schedule was run on a standard desktop with processor Intel Pentium 2, 1.20 GHz and the test case considered was the one given the difficulty was also coded in AMPL with MINOS 3.5 as the solver and it took 0.02125s on the standard desktop mentioned before. While resolving the difficulty using Genetic Algorithm, the average time taken was 0.02s while the time taken for resolving it using simulated bake was 0.05s (The time was noted on a standard desktop with processor Intel Pentium 2, 1.20 GHz).

**Conclusions:**

The genetic algorithm is a model of machine learning which derive its behavior from a metaphor of the processes of evolution in nature. This is done by the creation within a machine of a population or individuals represented by chromosomes, in essence a set of character strings that are analogous to the base-4 chromosomes that we see in our own dna. An empirical research into resolvent the cession model using Genetic Algorithm and Simulated bake is offering. Various parameters efficacious the algorithms are studied and their penetration on convergence to the final optimum solution is shown.
Fig. 5: Various schedules for constant (m).

Fig. 6: Various schedules for varying (m).

REFERENCES


