CHAPTER 3

FUNDAMENTAL CONCEPTS OF NON-TRADITIONAL OPTIMIZATION TECHNIQUES

3.1 INTRODUCTION

The purpose of this chapter is to discuss the fundamental concepts of various non-traditional optimization techniques such as Simulated Annealing, Genetic Algorithm, Tabu Search and other techniques including Particle Swarm Optimization, Cellular Automata, Guided Local Search, Iterated Local Search, Greedy Randomized Adaptive Search procedures and Artificial Neural Network.

3.2 SIMULATED ANNEALING (SA)

Simulated Annealing (Cerny 1985 and Kirkpatrick et al 1983) is an optimization technique that belongs to a class of stochastic (i.e. random) search techniques. SA mimics the process of annealing. The desired result is the crystalline structure, compared to fast untempered cooling which results in a brittle defective structure.

When the solid has reached its melting point, a large amount of energy is present within the material. As the temperature is reduced, the energy within the material decreases. Based on this physical metaphor of cooling a molten substance, SA algorithm was developed to solve a real world problem. SA can be defined in four steps namely Initial Solution, Randomly Tweak Solution, Acceptance Criteria and Reduce Temperature.
3.2.1 Initial Solution

In most of the optimization techniques, the initial solutions are generated in a stochastic manner (i.e. random manner). In SA, these stochastic solutions are loaded into a list called “Current Solution”. This initial procedure gives a base from which to search for a more optimal solution to the problem.

3.2.2 Assess Solution

Assessing the solution consists of decoding the current solution and then performing whatever action is necessary to evaluate it against the given problem. The encoded solution simply consists of a set of variables. These variables would be decoded from the current solution and the energy of the solution assessed based upon how well it solved the given problem.

3.2.3 Randomly Tweak Solution

Tweaking the solution begins by copying the current solution into a list called “Working Solution”. Then the working solution is randomly modified depending upon the encoding of variables. Once the working solution has been tweaked, the solution defined in the previous step is assessed. This random trial is based upon the Metropolis Monte Carlo Simulation algorithm (Metropolis 1953).

3.2.4 Acceptance Criteria

At this juncture, the algorithm possesses two solutions. The first one called the original solution or the current solution and the second one is called the tweaked version (or) working solution. Each has an associated energy, which is the strength of the solution (lower the energy, the better the
solution). The working solution is then compared to the current solution. If the working solution has less energy than the current solution, then the algorithm copies the working solution to the current solution and moves on to temperature reduction. However if the working solution has greater energy than the current solution, then the acceptance criterion is applied to evaluate the probability of acceptance of the solution and this is done using the following equation based on the law of thermodynamics.

\[ P(\delta E) = \exp\left(-\frac{\delta E}{T}\right) \]  

(3.1)

At higher temperature, SA permits acceptance of poor quality solutions in order to evaluate more of the available solutions. As the temperature decreases, the search allowable also decreases until equilibrium is reached when the temperature reaches 0°C.

### 3.2.5 Reduce Temperature

After some number of iterations through the algorithm, temperature is reduced by a small amount. The temperature reduction can be calculated using a geometric equation given below.

\[ T_{i+1} = \alpha \cdot T_i \]  

(3.2)

where \(\alpha\) is a constant less than one. Apart from Equation (3.2), many cooling strategies are available including linear and non-linear functions.

### 3.2.6 Repeat

A number of iterations can be performed at a single temperature. The temperature can be reduced after a set of desired iterations and the above
process continues until the temperature reaches zero. The algorithm for SA is given in Figure 3.1.

![Simulated Annealing Flowchart](image)

**Figure 3.1 Simulated Annealing**

3.3 GENETIC ALGORITHM (GA)

The Genetic Algorithm, introduced by John Holland (1962), is a search algorithm that works over a population of encoded design variables to solve a given problem. GA mimics the phenomenon of natural evolution, species search for increasingly beneficial adaptations survival within their
complex environments. The search takes place in the species’ chromosomes where changes, and their effects, are graded by the survival and reproduction.

The Genetic Algorithm, instead of trying to optimize a single solution as in the case of SA, works with a population of candidate solutions that are encoded as chromosomes. The chromosome consists of some set of genes that represent the independent variables for the problem at hand. GA generally consists of four different phases namely: Initiation, Evaluation, Selection and Recombination and is shown in Figure 3.2.

![Figure 3.2 Genetic Algorithm](image-url)
3.3.1 Initiation

At the beginning of iterations, GA generates some set of random solutions to form the initial population. Alternatively, this can be done by seeding the population with known fit chromosomes.

3.3.2 Evaluation

The Evaluation phase simply provides a way to rate how each chromosome solves the problem under consideration. It involves decoding of the chromosomes into the variable space of the problem and then checking the result of the problem using these parameters. Then the fitness is computed from the result.

3.3.3 Selection

Selection is quite possibly the most important and most misunderstood part of GA. In this phase, chromosomes are selected for proliferation to future populations based on their fitness. This selection process is a double-edged sword. If selection involves only the highest-fit chromosomes, the solution space becomes very limited due to lack of diversity. If selection is performed randomly, there is no guarantee that future generations will increase the quality of solution.

The result of the selection process is a set of chromosomes that will take part in recombination. Large varieties of selection algorithms exist. Roulette-wheel is one of the most widely used selection algorithm in GA. Other selection algorithms include tournament selection, rank-based selection, progeny testing, individual selection, family selection, within-family selection and combined selection. Roulette-wheel selection performs selection from the
population based upon the fitness of the chromosome. The higher-fit chromosome is more likely to be chosen for propagation to the next generation. In other words, the probability of selection is proportional to the fitness of the chromosome. A sample Roulette-wheel selection is shown in Figure 3.3.

<table>
<thead>
<tr>
<th>CH1</th>
<th>1 (7%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>CH2</td>
<td>5 (38%)</td>
</tr>
<tr>
<td>CH3</td>
<td>4 (32%)</td>
</tr>
<tr>
<td>CH4</td>
<td>1 (7%)</td>
</tr>
<tr>
<td>CH5</td>
<td>2 (16%)</td>
</tr>
</tbody>
</table>

![Roulette-wheel selection procedure](image)

**Figure 3.3** Roulette-wheel selection procedure

In Figure 3.3, CH1…CH5 represent the chromosomes in a particular generation along with their fitness value. Since the chromosome (CH2) has higher fitness value, it is copied twice for future generation. Similarly CH3 and CH5 are copied according to the fitness value. Chromosomes CH1 and CH4 are not selected at all, and they disappear from the subsequent population.

### 3.3.4 Recombination

In recombination, pairs of chromosomes are recombined, possibly modified and then placed back into the population as the next generation. The original sets of chromosomes are generally called as “parents” and the resulting modified chromosomes are the “children or offspring”. One or more genetic operators are applied on the chromosome with some probability.
Crossover and Mutation are widely used genetic operators which are analogous to natural genetics.

3.3.4.1 Crossover

The crossover operator takes two chromosomes, separates them at a random site in both chromosomes and then swaps the tails of the two chromosomes, resulting in two new chromosomes which are referred as offspring. If the random site is selected at a single point, then it is called single-point crossover and on the other hand if more than one random site were chosen, then it is called multi-point crossover.

This crossover operator does not introduce any new chromosome into the population, but it simply inter-mixes the existing population of chromosomes to create new chromosomes. This allows the GA, to search for new candidate solutions to solve the problem at hand. The single or two point crossover are the most widely used operator.

3.3.4.2 Mutation

The mutation operator introduces a random change into a gene in the chromosome. The mutation operator provides the ability to introduce new material into the population. Since chromosomes simply intermix with existing chromosomes, mutation provides the chance to expand the solution space.

3.3.5 Shortcomings of GA

GA has been extensively applied in the field of structural optimization to solve a wide variety of problems such as Optimal Stacking

### 3.3.5.1 Premature Convergence

The problem of premature convergence is because of lack of diversity within the population of chromosomes. When a majority of the chromosomes in the population is similar, the selection process has less material to work with and the rate of fitness increase slows and consequently increases the computation time. Premature convergence can be detected by comparing the average fitness of the population with the maximum fitness. If they are fairly close to one another, then premature convergence has occurred.

One important reason for the premature convergence is a population that is too small for the problem. Secondly, the selection algorithm also plays a major role leading to premature convergence.

### 3.3.5.2 Epistasis

Epistasis is defined as the interdependence among the variables (i.e. genes) encoded in the chromosome. If no gene is related to another in the
chromosome, epistasis is small. If genes are dependent on one another, then the epistasis is high and can create difficulties for the recombination algorithms.

3.4 TABU SEARCH (TS)

Tabu Search (Glover 1989, 1990 and Glover and Laguna 1997) relies on the systematic memory to guide the search process. Tabu Search uses a local search, that at every step, makes the best possible move from ‘s’ to a neighbor solution ‘s1’ even if the new solution is worse than the current one. In the latter case, the move that affects the objective function is also chosen to prevent local search from immediately returning to a previously visited solution. More generally to avoid cycling, Tabu Search explicitly memorizes recently visited solutions and forbids moving back to them. TS forbids reversing the effect of recently applied moves by declaring tabu those solution attributes that change in the local search. The tabu status of solution attributes is then maintained for a number of ‘tt’ iterations; the parameter ‘tt’ is called tabu tenure or the tabu list length. Unfortunately, this may forbid moves towards attractive, unvisited solutions. To avoid such an undesirable effect, an aspiration criterion is used to override the tabu status of certain moves. Most commonly, the aspiration criterion drops the tabu status of moves leading to a better solution than the best solution visited so far.

The use of a short term memory in the search process is probably the most widely applied feature of Tabu Search. TS algorithms that only rely on the use of short term memory are called simple tabu search. To increase the efficiency of simple TS, long term strategies can be used to intensify or diversify the search.
3.5 OTHER TECHNIQUES

3.5.1 Particle Swarm Optimization

Particle swarm optimization (Eberhart and Kennedy 1995) is a population based stochastic optimization technique, inspired by social behavior of bird flocking or fish schooling. Particle Swarm Optimization (PSO) shares many similarities with evolutionary computation techniques such as Genetic Algorithms. The system is initialized with a population of random solutions and searches for optima by updating generations. However, unlike Genetic Algorithm, PSO has no evolution operators such as crossover and mutation. In PSO, the potential solutions, called particles, fly through the problem space by following the current optimum particles. In PSO, each single solution is a "bird" in the search space, technically stated as "particle". All the particles have fitness values which are evaluated by the fitness function to be optimized, and have velocities which direct the flying of the particles. The particles fly through the problem space by following the current optimum particles. PSO is initialized with a group of random particles (solutions) and then searches for optima by updating generations. In every iteration, each particle is updated by following two "best" values. The first one is the best solution (fitness) it has achieved so far. (The fitness value is also stored.) This value is called pbest. Another "best" value that is tracked by the particle swarm optimizer is the best value, obtained so far by any particle in the population. This best value is a global best and called gbest. When a particle takes part in the population as its topological neighbor, the best value is a local best and is called ‘lbest’. After computing the two best values (pbest and gbest), the particle velocity and the particle position are calculated for further iterations. The above process is then repeated till the termination criterion is attained.
3.5.2 Cellular Automata

Cellular automata (Wolfram 1994) are simple mathematical idealizations of natural systems. They consist of a lattice of discrete identical sites, each site taking on a finite set of, say, integer values. The values of the sites evolve in discrete time steps according to deterministic rules that specify the value of each site in terms of the values of neighboring sites. Cellular automata may thus be considered as discrete idealizations of the partial differential equations often used to describe natural systems. Their discrete nature also allows an important analogy with digital computers. Cellular automata may be viewed as parallel-processing computers of simple construction.

3.5.3 Guided Local Search

Guided Local Search (Voudouris and Tsang 1995) is a metaheuristic technique which modifies the given objective function during the search process with a new augmented cost function. The augmented cost function encapsulates the original objective function plus additional penalty terms associated with each solution.

3.5.4 Iterated Local Search

Iterated Local Search (Lourenço et al 2002, Martín et al 1991) is a simple and powerful metaheuristic technique. In this technique, local search is performed, starting from an initial solution. Once the local search is stuck, the locally optimal solution is perturbed by a move in a neighborhood different from the one used by local search. This perturbed solution is the new starting solution for the local search that takes further in the algorithm. Finally, an acceptance criterion is decided as to which of the two locally optimal
solutions could be selected as a starting point for the next perturbation step. The main motivation for Iterated Local Search is to build a randomized walk in a search space of the local optima with respect to some local search algorithm.

3.5.5 Greedy Randomized Adaptive Search

Greedy Randomized Adaptive Search Procedure (Feo and Resende 1989, 1995) is an iterative procedure which consists of two phases, namely, a construction phase and a local search phase. In the construction phase a solution is constructed from scratch, adding one solution component at a time. At each step of the construction heuristic, the solution components are ranked according to some greedy function and a number of the best-ranked components are included in a restricted candidate list. Then, one of the components of the restricted candidate list is chosen randomly, according to a uniform distribution. Once a full candidate solution is constructed, this solution is improved by a local search phase.

3.5.6 Artificial Neural Network

Artificial Neural Networks (ANN, Rosenblatt 1958) are very simple implementations of local behavior observed within our own brains. The brain is composed of neurons, which are the individual processing elements. Neurons are interconnected by axons that end at the neuron in a synapse. The synapse is responsible for relaying a signal to the neuron.

ANNs attempt to mimic the basic operation of the brain. Information is passed between the neurons, and based upon the structure and synapse weights, a network behavior is provided.
3.6 SUMMARY

In this chapter, the fundamental concepts of some of the non-traditional optimization techniques have been discussed along with their advantages and disadvantages. The formulations of optimization problem of the proposed research are furnished in the next chapter.