SOFT COMPUTING

SEARCH STRATEGIES

Dr.B.K.Tripathy

SYLLABUS

- Genetic algorithms
- Hybrid systems

GENETIC ALGORITHMS

GENETIC ALGORITHMS

- The **origin of the theory of evolution** is the book
- **ON THE ORIGIN OF SPECIES**
- written by **Charles Darwin**

- The **central concept** lies in the process called
- **SURVIVAL OF THE FITTEST**
- **This principle is used by nature in its selection**

GENETIC ALGORITHMS CONTD…

- All living creatures are **descendants of older species**
- Any **variation occurs** is due to **natural selection**
- some individuals have **greater chance of reproduction**
- Due to some **heritable differences**
- And are said to have **higher “fitness”**
- **Fitness measures the success of an organism**

REQUIREMENTS FOR EVOLUTION

- **Evolution:** The theory that **all living things have changed in response to environmental conditions** by the natural selection of **randomly occurring mutations**

- **Mutation:** The hypothetical occurrence of **new forms arising through change in gene construction of the nucleus** and **differing sufficiently from the parent forms** to constitute new varieties

- **FOUR REQUIREMENTS** for evolutions to take place
- **1. FITNESS 2.VARIATION 3.REPRODUCTION 4.HEREDITY**

KEY CONCEPTS ELABORATELY

- **FITNESS:** It measures the **ability of an individual to survive and reproduce**
- The **difference between the values of one organism to another** is the higher number of offspring of one with the higher fitness value

- **VARIATION:** It is **essential** for evolution to take place
- **This is the series of changes occurring in species**
- **Types of Variation (2):**
- **Inherited variation**
- **Environmental variation**

KEY CONCEPTS CONTD…

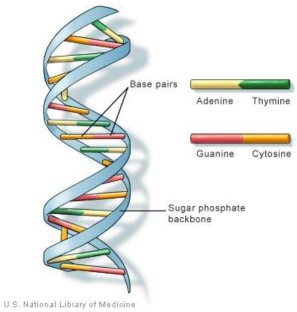
- **INHERITED VARIATION:** It is a **genetically inherited character**
- Off springs **inherit half of their characters** from **each of their parents**.
- **EXAMPLE:** Colour of hair

- **ENVIRONMENTAL VARIATION:**
- **There are two factors**
- **Climate**
- **Culture**
- These factors affect organisms

KEY CONCEPTS CONTD…

- **Reproduction:** This is the **process** in which **offsprings are generated from their parents**
- **TWO** types: 1. Asexual, 2.Sexual
- **Asexual reproduction:** It **involves only one organism**
- **Sexual reproduction:**
- It requires **two different organisms** for its occurrence
- It is the **most common method of reproduction**
- **HEREDITY:** This is a process in which offsprings get some characteristics of parents
- It **helps in the variation in the population** and hence may cause the **development of new species**

GENETIC MECHANISM



- Genetic mechanism behind evolution : Put forth by **Gregory Mendel** in 20th century
- Each **cell** contains **chromosomes**

In biology, a gene is a sequence of DNA or RNA that codes for a molecule that has a function

- Are **strands (Twisted together) of DNA (Deoxyribonucleic acid)**
- Consist of **genes (Smallest hereditary unit)**
- **Genes** are inherited form **parents**
- This **(inheritance)** happens during reproduction and the **process is called crossover**
- The **set of all chromosomes** is called **genome**

In humans, genes vary in size from a few hundred DNA bases to more than 2 million bases.

GENETIC MECHANISM CONTD…

- **Each gene encodes a different trait** (**distinguishing characteristic**)
- There is a **different variety of gene** called the **allele**
- **For Example:**
- There are **different alleles** for **brown hair** and **dark hair**

- The **flow of gene** in a population or
- **Transfer of alleles or genes** from **other populations or due to mutation**,
- which is a **change in the structure of a gene** that is usually the consequence of an **error in the transcription of DNA**

EVOLUTIONARY COMPUTATION

- The field of evolutionary computation is considered **a STRONG TOOL TO DEAL WITH COMPLEX PROBLEM SOLVING**

- It is often required of **a computer program to easily adapt to an evolving environment**
  - To be **innovative** and **original**
  - To **perform tasks** that are **simply too difficult** to be **programmed manually**, especially in the **field of artificial intelligence**

- **The theory of evolution has been an inspiration to develop these concepts**
- because **IT IS A WAY OF FINDING OPTIMAL SOLUTIONS**

EVOLUTIONARY COMPUTATION CONTD…

- In biological terms the **solutions** are found in the **set of all possible genetic sequences**
- Are the **individuals with the highest "fitness"**
- Also, as the **human organism develops complex mechanisms to solve problems that would threaten his survival,**
- **computer programs need to develop innovative ways to get an optimal solution**

EVOLUTIONARY COMPUTATION CONTD…

- The constant changes in the **computational world** are well represented by **the ever-changing natural world**

- **Another parallelism** is
- The **ability to evaluate different possibilities at the same time**, as the different species that are simultaneously tested on their fitness.

GENETIC ALGORITHMS

- Developed during **late 1950s and early 1960s**
- By some **evolutionary biologists**
- As a result of their trials to develop algorithms to model aspects of natural evolution through programming in computers
- Evolutionary computing inherent
- **Biologists did not realise then that their strategy would be applicable to artificial problems**
- **Evolution- inspired algorithms** were **independently developed by**
- **G.E.P. Box, G.J. Friedman, W.W. Bledsoe and H.J. Bremermann**

GENETIC ALGORITHMS

- These algorithms were applied by them for **function optimization** and **machine learning**
- This was around 1962 and **was not pursued further immediately**
- The **next step** in the development was by **I. Rechenberg in 1965**, when he **introduced** a technique called **EVOLUTION STRATEGY**
- It had **very little similarity to genetic algorithms**
- **Most of the concepts** used in **genetic algorithms** like **population, crossover and mutation** of parents to produce offspring **were not there**

GENETIC ALGORITHMS

- The **technique of evolutionary programming** was introduced by **three scientists** in **1966**
- **L.J. Fogel, A.J. Owens and M.J. Walsh**

- It is the next important step in this direction
- The **solutions** were represented by **finite state machines**
- It **follows the evolution strategy of Rechenberg**
- The process was:
- To randomly mutate one of these simulated machines and keeping the better of the two

GENETIC ALGORITHMS CONTD…

- Both the areas of **evolution strategy** and the **evolutionary programming** technique are still being **pursued for research**

- The **most important aspect** which was **missed in both** these approaches **was recognition of the importance of crossover**

- **John Holland** was the **first to propose** the **concept of crossover** and other related **recombination operators**

- **THESE CONCEPTS FORM THE BASIS OF GENETIC ALGORITHM**

GENETIC ALGORITHMS CONTD…

- These **ideas** he put forth while **working on adaptive systems** in as early as **1962**
- Real breakthrough in **1975**

- He published a **book** titled:
- **ADAPTATION IN NATURAL AND ARTIFICIAL SYSTEMS**

- The contents are **based upon his own work** and by **some of his colleagues**

GENETIC ALGORITHMS CONTD…

- This book encompasses the **idea of adaptive digital systems** and the **three ingredients of genetic algorithm**

- **Crossover**
- **Mutation**
- **Selection**

- Defined and used
- **Thus the problem solving strategy using biological evolution was simulated**

GENETIC ALGORITHMS CONTD…

- The **theoretical soundness of genetic algorithm** could be established through the **introduction of the notion of schemata.**
- Further, the **dissertation of K. D. Jong** carried the ideas forward by **showing that GA can perform well on variations like over a wide range of test functions, and noisy and discontinuous searches**
- Several published books; like **Davis (1991)**, **Goldberg (1989)**, **Holland (1975)**, **Michalewicz (1992)** and **Deb (1995)** contain most of the GA studies available till the corresponding time.

GENETIC ALGORITHMS CONTD…

- **Several conference proceedings** update the knowledge and contributions at regular intervals
- Many efforts have been made to **apply GA to structural engineering problems**
- This includes works of **Goldberg and Samtani (1986), Jenkins (1991)**, **Deb (1991)** and **Rajeev and Krishnamurthy (1992)**
- But the applications of GA are **not limited to structural engineering only**
- **Several applications** have been obtained in the fields of **biology, computer science, image processing and pattern recognition, physical science, social sciences and neural networks**

GENETIC ALGORITHMS CONTD…

- At first, these applications were mainly theoretical
- As research continued to proliferate, genetic algorithms **migrated into the commercial sector**
- Their **rise fueled by the exponential growth of computing power and the development of the Internet**
- Today, **evolutionary computation is a thriving field**

GENETIC ALGORITHMS CONTD…

- Genetic algorithms are "**solving problems of everyday interest"**

- in diverse areas as
- **stock market prediction and portfolio planning**
- **aerospace engineering**
- **microchip design**
- **biochemistry and molecular biology**
- **scheduling at airports and assembly lines**

GENETIC ALGORITHMS CONTD…

- The power of evolution has **touched virtually any field one cares to name**
- **Shaping the world around us invisibly in countless ways**
- **New uses continue to be discovered as research is ongoing**

- And at the heart of it all lies nothing more than **Charles Darwin's simple, powerful insight:**
- That the **random chance of variation**, **coupled with the law of selection**, is a **problem-solving technique of immense power** and **nearly unlimited application**

GENETIC ALGORITHMS CONTD…

- The **language used** to describe the elements of a genetic algorithm **is directly derived from biology**, from which they draw inspiration
- The **set of all possible solutions to the problem** is called the **search space, or population**
- **Each feasible solution** is an **individual**

- Each feasible solution is characterised by **two** factors:
- A **chromosome**, which is typically **encoded as a bit string**
- A **value of fitness**

GENETIC ALGORITHMS CONTD…

- **Short blocks of different bits of a chromosome** that encode a different parameter of the solution are the **genes,** which can have different values, or **alleles**: in a bit string an allele is 0 or 1
- The **position of a gene** in a chromosome is the **locus**
- The **collection of all the genes** is the **genotype**

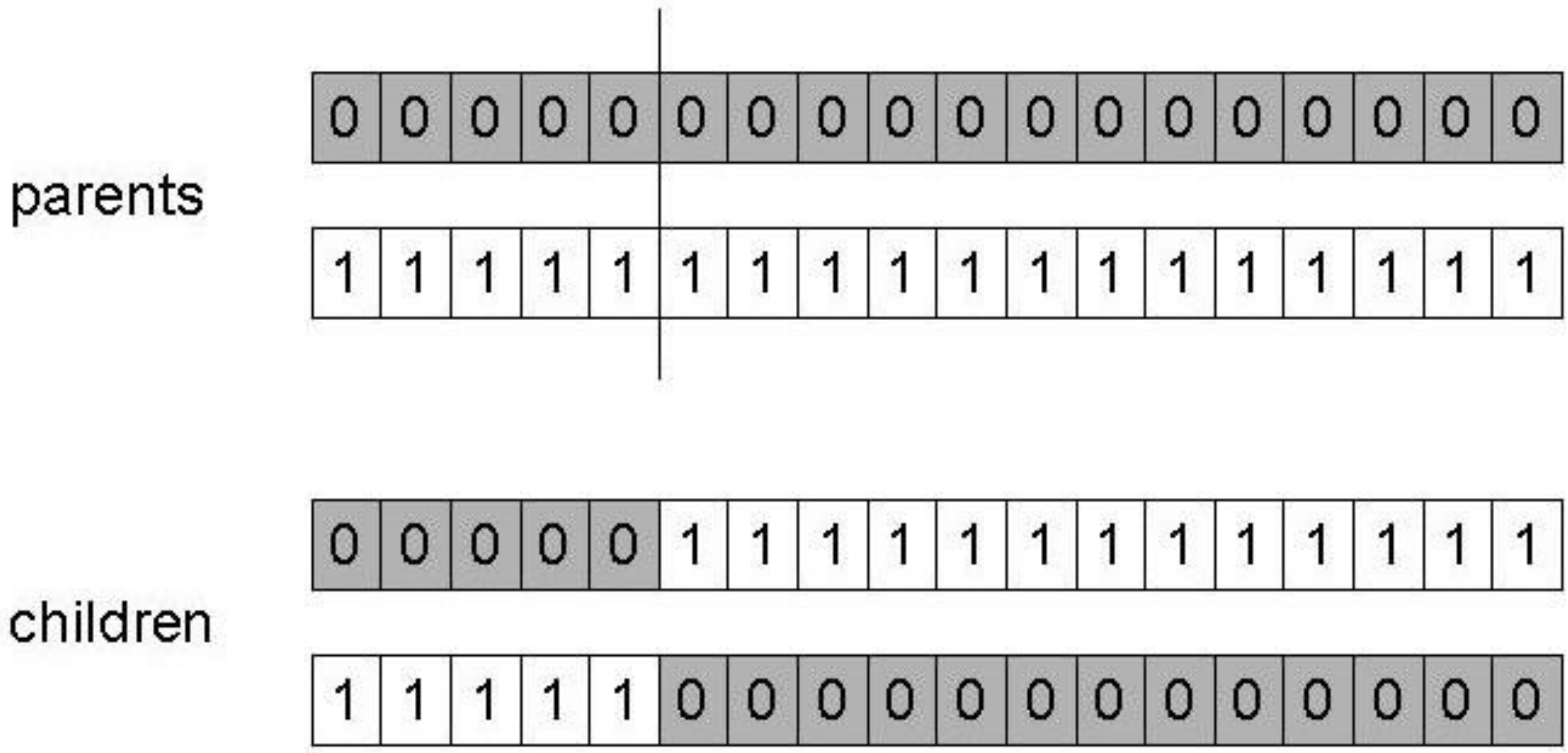
GENETIC ALGORITHMS CONTD…

- In a genetic algorithm
- Some individuals are **chosen as the parents**
- Selection is based on **their fitness**
- **Individuals with higher fitness** are more likely to **be selected**
- The parents are then allowed to produce offspring
- By applying **reproductive operators**
- Reproductive operators are
- **Crossover**
- **Mutation**
- **Crossover operators** recombine the chromosomes of the parents
- The **most common type** is called **single point crossover**

GENETIC ALGORITHMS CONTD…

- A **random point** in which **to cut the two genes** is selected
- The **first part of the first parent** is then **combined with** the **second part of the second parent** to create a child
- Similarly, the **second part of the first parent** is **combined with** the **first part of the second parent** to produce the **second child**

SINGLE POINT CROSS OVER



TWO POINT CROSS OVER

- Here two random sites are chosen
- The contents bracketed by these sites are exchanged between two mated parents
- If the cross-site 1 is six and cross-site 2 is twelve, the strings between six and twelve are exchanged

Parents

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

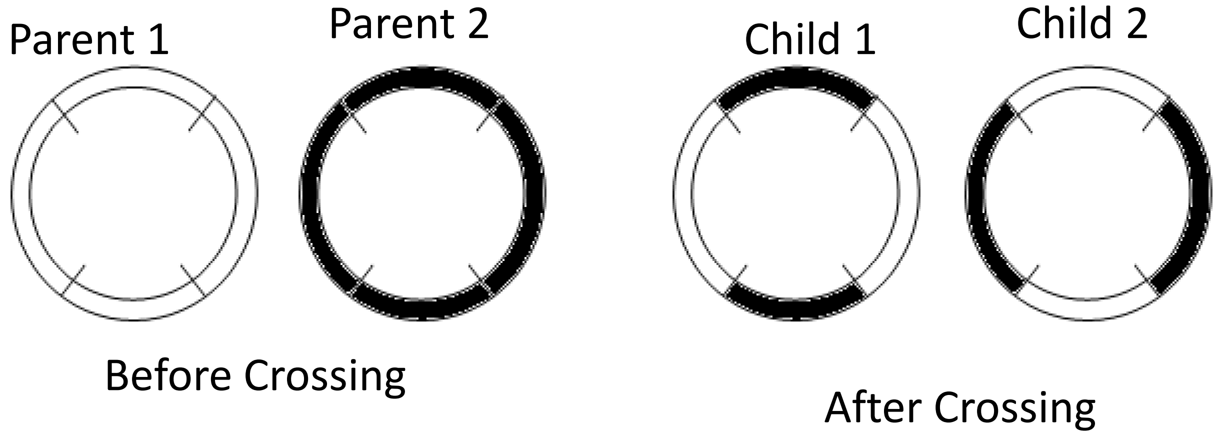
Children

0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0

1	1	1	1	1	0	0	0	0	0	0	0	1	1	1	1	1	1

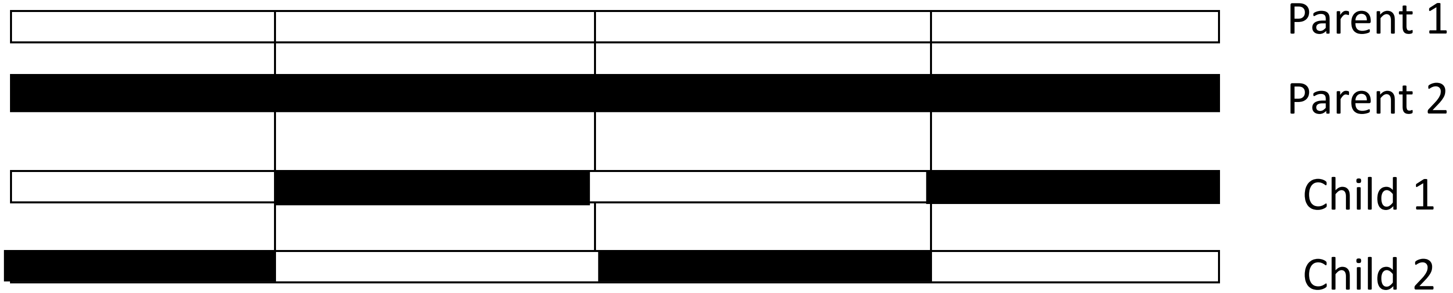
MULTI POINT CROSS OVER

- In a multi-point cross over, again there are two cases.
- **Even number of cross-sites**
- Odd number of cross-sites
- In case of even numbered cross-sites, the string is treated as a ring with no beginning or end
- The cross-sites are selected around the circle uniformly at random



MULTI POINT CROSS OVER CONTD…

- **Odd number of Cross sites**



UNIFORM CROSS OVER

- An extreme of multi-point cross over
- **Without Mask:**
- Each bit from either parent is selected with a probability of 0.5 and then interchanged
- It is radically different from one-point cross over
- **With Mask:**
- Sometimes gene in the offspring is created by copying the corresponding gene from one or the other parent chosen according to a randomly generated **cross over mask**

UNIFORM MASK CONTD…

- Mask comprises of 1s and 0s
- **Mask bit = 1: gene is copied from the first parent**
- **Mask bit = 0: gene is copied from second parent**
- **To produce the second offspring: The parents are exchanged**
- A NEW CROSS OVER MASK IS GENERATED FOR EACH PAIR OF PARENTS
- **Offspring** therefore **contains a mixture of genes from each parent**
- The **number of effective crossing points is not fixed** but **averages to L/2** (where L is the chromosome length)

UNIFORM CROSSOVER CONTD…

Parent 1	1	0	0	1	1	0	0	0	1	0	1	0	1	1	0	0	1	1
Parent 2	0	0	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1
Child 1	0	0	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1
Child 2	1	0	0	1	1	0	0	0	1	0	1	0	1	1	0	0	1	1

Uniform Crossover without mask

With Mask **1 0 0 1 0 1 1 1 0 0 1 0 1 0 0 1 0 0**

Parent 1	1	0	0	1	1	0	0	0	1	0	1	0	1	1	0	0	1	1
Parent 2	0	0	0	1	0	0	1	1	0	0	1	1	0	0	1	0	0	1
Child 1	1	0	0	1	0	0	0	0	0	0	1	1	1	0	1	0	0	1
Child 2	0	0	0	1	0	0	0	0	0	0	1	1	1	0	1	0	0	1

TWO IMPORTANT CONCEPTS

- **Chromosome:** A microscopic, **thread like bundle** of **Deoxyribo Neucleic Acid** (DNA) molecules which **collectively carry the hereditary material** in **subunits called genes**

- **Mutation:** **The process by which new forms arise**

  By a changing or being changed the hypothetical occurrence of new forms

  Arising through change in gene construction of the nucleus and

  Differing sufficiently from the parent forms to constitute new varieties

GENETIC ALGORITHMS CONTD…

- **Crossover does not always occur**
- Its **probability** is just **between 60% and 70%**

- **Mutation is rarer**
- It occurs with a **probability between 0.01% and 0.02%**
- **When it happens**, **one bit changes from 0 to 1 or vice versa**
- **Mutation assures** that the **population doesn’t become too static** and therefore incapable of reproduction

GENETIC ALGORITHMS CONTD…

- The **off springs form a new generation** and the **process starts from the beginning**
- Again, **fitness is evaluated** for **each individual of this new population**
- The **selected ones reproduce** until a **certain number of new generations have been produced** and analyzed
- The **problem is then satisfied**

GENETIC ALGORITHMS CONTD…

- Genetic algorithms are
- **Good** at taking **larger, potentially huge search spaces and**
- **navigating them** looking for **optimal combinations** of things and
- **solutions which we might not find in a life time**
- Genetic algorithms are **very different** from most of the **traditional optimization methods**

GENETIC ALGORITHMS CONTD…

- Genetic algorithms need **design space** to be covered into **genetic space**
- So, genetic algorithms **work with a coding of variables**
- The **advantage** of working with a **coding of variable space** is that
- **Coding discretizes the search space** even though the function may be continuous

GENETIC ALGORITHMS

- A more **striking difference** between genetic algorithms and most of the traditional optimization methods is that
- **GA uses a population of points at one time** in contrast to the **single point approach by traditional optimization methods**
- This means that GA **processes a number of designs at the same time**
- To **improve the search direction** in traditional optimization methods, **transition rules are used** and they are **deterministic** in nature but **GA uses randomized operators**
- **Random operators** improve the search space in an **adaptive manner**

GENETIC ALGORITHMS

- Three **most important aspects of GA** are
  - **Definition of objective function**
  - **Definition and implementation of genetic representation**
  - **Definition and implementation of genetic operators**

- Once these three have been defined, the GA should work fairly well beyond doubt
- We can, **by different variations**, **improve the performance**, **find multiple optima** (species if they exist) or **parallelize the algorithms**

REPRODUCTION IN GA

- Is usually the **first operator** applied on population
- **Chromosomes are selected** from the population to be parents to cross over and produce offspring
- According to Darwin's evolution theory the **best ones should survive and create new offspring**
- That is why **reproduction operator** is sometimes known as the **SELECTION OPERATOR**
- There are a number of **reproduction operators** in GA
- The essential idea is that the **above average strings are picked** from the current population and
- Their **multiple copies are inserted in the mating pool in a probabilistic manner**

SELECTIONS IN GA

- The **various methods** of selecting chromosomes for parents to cross over are

- **Roulette-wheel** selection
- **Boltzmann** selection
- **Tournament** selection
- **Rank** selection
- **Steady-state** selection

ROULETTE-WHEEL SELECTION

- The commonly used reproduction operator is the proportionate reproductive operator where **a string is selected from the mating pool** with a **probability proportional to the fitness**
- Thus, i^{th} string in the population is selected with a probability proportional to F_i where F_i is the fitness value for that string
- Since the population size is usually kept fixed in a simple GA, the **sum of the probabilities** of **each string being selected** for the mating pool **must be one**
- The probability of the i^{th} selected string is $p_i = \dfrac{F_i}{\sum_{j=1}^{n} F_j}$

ROULETTE-WHEEL SELECTION CONTD…

- One way to implement this selection scheme is to imagine a **Roulette-wheel** with its **circumference for each string** marked **proportionate to string's fitness**

- The **fitness of the population** is calculated as **Roulette-wheel is spun 'n' times** (usually n = 8), each time selecting an instance of the string **chosen by the Roulette-wheel pointer.**

- Since the **circumference of the wheel** is marked according to a **string's fitness**, the Roulette-wheel mechanism is expected to make $F_i / \overline{F}$ copies of the i^{th} string of the mating pool.

A ROULETTE-WHEEL

ROULETTE-WHEEL SELECTION CONTD…

- The average fitness $$\overline{F} = \frac{1}{n}\left(\sum_{j=1}^{n} F_j\right)$$

  This Roulette-wheel selection scheme can be simulated easily.
- Using the fitness value F_i of all strings, the probability of selecting a string being copied, can be calculated by adding the individual probabilities from the top of the list.
- The Roulette-wheel concept can be simulated by realizing that the string in the population represents the cumulative probability.
- Thus the first string represents the cumulative values from 0 .
- Hence cumulative probability of any string lies between 0 – 1.

ROULETTE-WHEEL SELECTION CONTD…

- **In order to choose n strings, n random numbers between zero and one are created at random.**
- Thus, the string that represents the chosen random number in the cumulative probability range (calculated from fitness value) for the string, is copied to the matting pool.
- This way, the string with a higher fitness value will represent a larger range in the cumulative probability values and therefore, has a higher probability of being copied into the mating pool.
- **On the other hand, a string with a smaller fitness value represents a smaller range in cumulative probability values and has a smaller probability of being copied into the mating pool.**

BOLTZMANN SELECTION

- **Simulated annealing** is a method of **functional minimization or maximization.**
- This method **simulates the process of slow cooling of molten metal** to achieve the **minimum function value** in a minimization problem.
- The cooling phenomenon is **simulated by controlling a temperature like parameter** introduced with the concept of **Boltzmann probability distribution**
- A system in **thermal equilibrium** at a temperature T has its **energy distributed probabilistically** according to

$$P(E) = \exp(-E / kT)$$

where ‘k’ is Boltzmann constant

GENETIC ALGORITHMS

- This expression suggests that a system at a
- **High temperature has almost uniform probability of being at any energy state**, but at a
- **Low temperature it has a small probability of being at a high energy state**.

- Therefore, by controlling the temperature T and assuming search process follows Boltzmann probability distribution, the convergence of the algorithm is controlled.

- **THIS IS BEYOND OUR SCOPE AND DETAILS CAN BE FOUND IN THE BOOK BY DEB IN 1995.**

TOURNAMENT SELECTION

- GA **uses a strategy to select the individuals** from population and insert them into a mating pool.
- Individuals from the mating pool are used to generate new offspring, which are the basis for the next generation.
- As the individuals in the mating pool are the ones whose genes will be inherited by the next generation, **it is desirable that the mating pool consists of good individuals.**
- **A selection strategy in GA is simply a process that favours the selection of better individuals in the population for the mating pool.**
- There are **two important issues in the evolution process of genetic search, population diversity** and **selective pressure**, as given by **Whitley in 1989**.

GENETIC ALGORITHMS

- **Population diversity** means that the genes from the already discovered good individuals are exploited while promising the new areas of the search space continue to be explored.
- **Selective pressure** is the degree to which the better individuals are favoured.
- The higher the selective pressure the more, the better individuals are favoured.
- **THE SELECTIVE PRESSURE DRIVES GA TO IMPROVE POPULATION FITNESS OVER SUCCEEDING GENERATIONS.**
-

GENETIC ALGORITHMS

- The **convergence rate of GA is largely determined by the selective pressure and population diversity**.
- In general, **higher selective pressure results in higher convergence rates.**
- However, **if the selective pressure is too high, there is an increased chance of GA prematurely converging to local optimal solution** because the population diversity of the search space to be exploited is lost.

RANK SELECTION

- The Roulette-wheel **will have problem when the fitness values differ very much**
- For example, if the best chromosome fitness is 90%, **its circumference occupies 90% of Roulette-wheel**, and then other chromosomes will have very few chances to be selected.
- **Rank selection first ranks the population and taken every chromosome, receives fitness from the ranking**.
- The worst will have fitness 1, the next 2, ..., and the best will have fitness N (***N* is the number of chromosomes in the population**).
- The Roulette-wheel selection is applied to the modified wheel

STEADY- STATE SELECTION

- **This is not a particular method of selecting the parents**
- The main idea of the selection is that
- **BIGGER PART OF CHROMOSOME SHOULD SURVIVE TO THE NEXT GENERATION**

- Here, GA works in the following way
- In every generation are selected, a few (**good individuals with high fitness for maximization problem**) chromosomes, for creating new off springs
- Then, some (**bad with low fitness**) chromosomes are removed and new offspring is placed in that place

- **THE REST OF THE POPULATION SURVIVES IN A NEW GENERATION**

CROSSOVER AND MUTATION PROPERTIES

- After the reproduction phase is over, the population is enriched with better individuals
- Reproduction makes clones of good strings, but does not create new ones
- **Cross over operator is applied to the mating pool with a hope that it would create a better string**
- The aim of the cross over operator is to search the parameter space
- In addition, search is to be made in a way that the information stored in the present string is maximally preserved because these parent strings are instances of good strings selected during reproduction

CROSSOVER AND MUTATION PROPERTIES

- **Cross over is a recombination operator**, which proceeds in three steps
  - The **reproduction operator** selects at random a pair of individual strings for mating
  - A **cross site** is selected at random along the string length
  - The position values are swapped between two strings following the cross site
- Cross over rate is the probability of cross over and it is denoted by

$$P_C = \frac{number\ of\ pairs\ to\ be\ crossed}{population\ size}$$

- This probability is normally fixed as 0.5 to 1 for population size of 30 to 200.

CROSS OVER RATE

- Accordingly the number of pairs to be crossed is decided
- We have seen that **with random cross-sites**, the **children strings produced may not have a combination of good substrings** from parent strings **depending on** whether or not the crossing site falls in the appropriate place
- But **we do not worry about this too much** because if good strings are created by cross over, **there will be more copies of them in the next mating pool generated by the reproduction operator**
- But **if good strings are not created by cross over**, they **will not survive too long**, because **reproduction will select against those strings in subsequent generations**

CROSS OVER RATE

- It is clear from this discussion that the **effect of cross over may either be detrimental or beneficial**
- Thus, in order to preserve some of good strings that are already in the mating pool, **not all strings in the mating pool are used in cross over**
- When a cross over probability of PC is used only 100 percent strings in the population are used in the cross over operation and 100(1- PC) percentage of the population remains as it is in the current population
- Even though the best 100(1- PC)% of the current population can be copied deterministically to the new population, this is usually preferred at random
- **A CROSS OVER OPERATION IS MAINLY RESPONSIBLE FOR THE SEARCH OF NEW STRINGS**

THE BASIC GENETIC ALGORITHM

- **Step 1:** Start with a randomly generated population
- **Step 2:** Calculate the fitness of each chromosome in the population
- **Step 3:** Repeat the following steps until n offsprings have been created:

  3.1 Select a pair of parent chromosomes from the current population

  3.2 With probability p_c crossover the pair at a randomly chosen point to form two offsprings

  3.3 Mutate the two offsprings at each locus with probability p_m
- **Step 4:** Replace the current population with the new population
- **Step 5:** Go to Step 2

FITNESS FUNCTION

- As pointed earlier, **GA mimics the Darwinian theory** of survival of the fittest and principle of nature to make a search process.

- Therefore, **GAs are usually suitable for solving maximization problems**

- **Minimization problems** are usually **transformed into maximization problems** by some **suitable transformation**
- In general, fitness function F(X) is the first **derived from the objective function** and used in successive genetic operations.

FITNESS FUNCTION

- Certain genetic operations require that **fitness function** be **non-negative**, although certain operators **do not have this requirement**
- Let $f(X)$ be the objective function
- Consider the following transformations:
- $F(X) = f(X)$ for maximization problem
- $F(X) = 1/f(X)$ for minimization problem, if $f(X) \neq 0$
- $F(X) = 1/(1+f(X))$, if $f(X) = 0$
- A number of such transformations are possible
- The fitness function value of the string is known as **string's fitness**

APPLICATIONS OF GENETIC ALGORITHMS

- Genetic algorithms can be applied in a variety of different situations in **science, engineering**, but also in **economics and social sciences.**

- **Combinatorial optimization problems:** an example of this is the **Travelling Salesman Problem**
- The aim, given a list of cities and their distances, is to find the **shortest route to visit each city once and then return to the starting point**
- The benefits of the use of genetic algorithms are linked to the **mutation** and **crossover operators**
- The former allows a **parallel search**, while the latter allows **individuals to share information**

APPLICATIONS OF GENETIC ALGORITHMS

- **Economics:** genetic algorithms are applied in
- Game theory (**During the last 16 years 13 peoples working in game theory have received Nobel prize in Economics**)
- A field of economics that studies strategic decision making
- The cooperation and conflicts between rational agents making decisions

- They are also applied in the game of prisoner’s dilemma, a game that illustrates the possibility that two rational beings might decide not to cooperate, even if it seems their best interest to do so

PRISONER’S DILEMMA

- The **Prisoner's Dilemma** is a classic problem in **game theory.**
- It has the paradoxical outcome that members of a group will consciously steer towards a sub-optimal outcome in certain scenarios
- The game is usually phrased in terms of two suspects, both of whom have been arrested for a major crime, who are offered a bargain.
- If both stay silent, each of them can still be convicted of a minor crime and sentenced to 6 months in prison.

PRISONER'S DILEMMA

- If one of them confesses and implicates the other, this provides evidence of a major crime.

- The confessor is rewarded by being let off of all crimes, and the other suspect will serve ten years in prison.

- If both confess, they will both serve two years in a plea for the major crime

- It is obvious that the best outcome (the **Pareto optimum**) for the group would be **if both prisoners cooperated and stayed silent: Six months for both prisoners**.

PRISONER’S DILEMMA

- However, in the "**default**" setting of the Prisoner's dilemma,
- **we assume that the prisoners are not given the chance to work out such a strategy and that they are interested in their own wellbeing first**.

- **Prisoner A will now analyse his options:**
- If Prisoner B chooses "don't confess", Prisoner A's best choice will be "confess": A gets out of prison immediately.
- If Prisoner B chooses "confess", Prisoner A's best choice will be "confess", too: 2 years is better than 10 years.

PRISONER’S DILEMMA

- Using this reasoning, both prisoners will choose "confess" as providing the **best outcome for themselves** in all circumstances, even though it is not best result *for the group*.

- The strategy "confess" is a strictly **dominant strategy**: The choice of the Prisoner B does not change the way Prisoner A will act. The "confess/confess" scenario is also the only **Nash equilibrium** in this problem.

- When a situation like the prisoner's dilemma, where the Pareto efficient outcome is not the actual outcome, occurs in a market economy, that can be an example of a **market failure**

APPLICATIONS IN ECONOMICS

- Genetic algorithms can be used to explore the possibility of developing cooperative behaviour
- Economics uses the following features of genetic algorithms: the ability to represent individuals as part of a population whose agents differ in strategy, the processing of information in parallel and the selection of better-performing strategies

APPLICATIONS OF GENETIC ALGORITHMS

- **Scheduling:** GAs are often applied in order to solve production scheduling problems because they operate in a set of various solutions in parallel

- The population of solutions consists in different and often conflicting answers

- The **goal of the genetic algorithm** is to **find the best way to assign the time** to the activities **to maximize the overall performance**

APPLICATIONS OF GENETIC ALGORITHMS

- **Robotics:** genetic algorithms are applied in this field **to obtain good navigating systems**

- The **navigation system** must be able to **direct the robot in a constricted environment** and get it to **its destination**, **without it crashing into objects and getting lost**

- The **genetic algorithm** scans the intermediary points **of every candidate path** to find the **most direct move** to the **destination**; **when an obstacle is found, it stops** and **goes back to the last free path section analysed.**

APPLICATIONS OF GENETIC ALGORITHMS

- It is proven that **genetic algorithms are able to perform very well in real life and complex situations**, because they are **adaptive to the environment** to which they are applied, **just like real life organisms** **adapt to the natural environment** in which they find themselves.
- The study of GAs is **also used** to model
- **Ecosystems**
- **Immune systems**
- **Cognitive systems**
- **social systems**

APPLICATIONS OF GENETIC ALGORITHMS

- **Industrial design** by parameterization
- **Network design** by construction

- **Routing** – finding the shortest path to deliver the packets, perform dynamic and anticipatory routing in telecommunications, easy of switching and optimize the cell tower placement with best coverage of signals

- **Time series prediction** – learning the strategies based on the history of data
- **Database mining** – classification, prediction, association
- **Control systems**
- **Artificial life systems**

APPLICATIONS OF GENETIC ALGORITHMS

- **Chemistry :** molecular conformation
- **Automotive design** – GA is used to **design the four wheeler vehicles** that are **efficient** in **shape**, **fuel consumption** and **safety**

- The modelling done by GA provides **more options for the designers in less cost**

- **Engineering design** – optimize the **structural and operational designs** in **machineries**, **factories**, buildings for **optimization of heat exchangers**, robot gripping arms, **satellite booms**, building trusses, turbines etc., It is also **used to identify the fault**

APPLICATIONS OF GENETIC ALGORITHMS

- **Evolvable** - design of **circuits in VLSI design**, **embedded systems**, **reconfiguring circuits** which are stochastic to evolve **new configuration from the old model**

- **Routing** – selecting the **most efficient and cost effective routing** for **shipment**, **air traffic**, **trip plan packet** by travel agency, **improving the productivity** etc.

- **Gaming** – GA are programmed to **find the most successful strategies** from the previous game histories. It learns from its experience in gaming that uses game theory
- **Security** – encryption for sensitive data and breaking those codes. Using GA one build or break the complex code for security

APPLICATIONS OF GENETIC ALGORITHMS

- **Molecular design** – used in applied chemistry and medicine for understanding of molecular structure, protein functions, predicting the binding between the protein molecules used by pharmaceutical industry to invent medicine for particular diseases

- **Gene Expression** - analysis of gene expression in micro array data helps in classifying the genes or to identify the genetic cause for the disease which moved treating patients based on personalized medicine

APPLICATIONS OF GENETIC ALGORITHMS

- **Finance Management** – market trend prediction, study of economic meltdown, assisting in investments by understanding the strategies using GA

- **Marketing** – GA helps the merchandisers in production and marketing consultants in planning and advertisement to sell the products

HYBRID SYSTEMS

WHY HYBRID SYSTEMS?

- Every system has some positive features and some negative features
- So, when we can combine two or more basic systems by taking most of their positive features then we are supposed to get a hybrid system which outperforms the individual systems
- An Example:
- When rough set was introduced in 1982, it was supposed by many to be a competitive model to fuzzy sets
- In a paper entitled “Fuzzy sets vs Rough sets” Pawlak tried to provide a comparison of the two models of uncertainty and show that rough set is a better model than fuzzy sets

EXAMPLE CONTD…

- However, it was shown by Dubois and Prade in 1990 in their paper, ”Rough Fuzzy sets and Fuzzy rough Sets”, International Journal of General Systems that Fuzzy set and rough sets are not competing models.
- These are complementary models
- They combined these two models to propose Fuzzy rough sets and Rough fuzzy sets
- It has been observed that in real life applications these two hybrid models are better than the individual models as is seen in the development of rough fuzzy c-means clustering algorithms

CHARACTERISTICS OF SOFT COMPUTING

- Main characteristics of Soft Computing (SC) is its intrinsic capability to create hybrid systems that are based on a (loose or tight) integration of constituent technologies
- This integration provides complementary reasoning and searching methods that allow us to combine domain knowledge and empirical data to develop flexible computing tools and solve complex problems
- There is interaction of knowledge and data in SC
- To tune knowledge derived models we first translate domain knowledge into an initial structure and parameters
- Then use global or local data search to tune the parameters

ADVANTAGE OF SOFT COMPUTING HYBRID SYSTEMS

- To control or limit search by using prior knowledge we first use global or local search to derive the models (structure + parameters) we embed knowledge in operators to improve global search
- Then we translate domain knowledge into a controller to manage the solution convergence and quality of the search algorithm