Study On Implementation Of Genetic Algorithm In Security System

By

Nurmohammad Khan – 2015-1-50-029

Mahadi Hasan Pavel – 2015-1-50-024

Under the supervision

of

Dr. Mohammed Arifuzzaman

Assistant Professor

Department of Electronics and Communications Engineering

East West University

A thesis is submitted in partial fulfillment of the requirements for the degree of Bachelor of Science in Electronics and communication Engineering.

Department of

Electronics and Communication Engineering

East West University, Dhaka-1212

December 2018

Spring, 2019

DECLARATION

This is certified that the project is done by us under the course "Thesis (ICE 498)". The thesis of **Implementation of Genetic Algorithm in Security System** has not been submitted elsewhere for the requirement of any degree or any other purpose except for publication.

Signature-

Nurmohammad Khan

2015-1-50-029

Mahadi Hasan Pavel

2015-1-50-024

ACCEPTANCE

This thesis paper is submitted to the **Department of Electronics and Communications Engineering, East West University** id submitted in partial fulfillment of the requirements for the degree of B.Sc in ICE under complete supervision of the undersigned.

Supervisor

Dr. Mohammed Arifuzzaman

Assistant Professor

Department of Electronics and Communications Engineering

East West University

Chairperson

Dr. Mohammed Moseeur Rahman

Assistant Professor

Department of Electronics and Communications Engineering

East West University

ACKNOWLEDGEMENT

First, we would like to thank almighty Allah for giving us the strength, proper knowledge wisdom and understanding for completing our thesis work.

Dr. Mohammed Arifuzzaman, Assistant Professor, Department of Electronics and Communications Engineering, East West University whom we regard as our mentor and supervisor, we thank him for the expertise and intelligence he has displayed while supervising this thesis.

We cannot forget our sincere honorable faculty of ECE in the academic interactions and company they have accorded to us especially **Dr. Mohammed Moseeur Rahman,** Chairperson, Department of Electronics and Communications Engineering, East West University.

ABSTRACT

In our thesis paper, we work on implementation of genetic algorithm in security system. In our growing world security is a big issue. We have lot of information and some of this information must have to keep safe. So we use different kind of encryption technique but none is 100% safe. By using brut force we can break any encryption technique or any password but it needs so many time in worst case. Human kind don't have so many time. So we try to develop something better with genetic algorithm.

Genetic algorithm is a powerful tool for solving any big problem. It is a metahuristic search solving technique for big problem or equations. So here we use genetic algorithm for break any security system or how it can be.

TABLE OF CONTENTS

CHAPTER 1

1.1 Security System in Computer:

On the off chance that you need a PC to be superbly secure, you could fill it with cement and dump it in the sea. This would shield any data on the PC from unseemly use. Tragically, the PC would be totally unusable, so you most likely would prefer not to do that! Since you need to both utilize your PC and guard it, you should rehearse great PC security. PC security enables you to utilize the PC while protecting it from dangers.

PC security can be characterized as controls that are set up to give classification, trustworthiness, and accessibility for all parts of PC frameworks. These parts incorporate information, programming, equipment, and firmware. This is a mind boggling definition. We should represent the definition by demonstrating to you a typical day for Samantha, a security director simply contracted for a little organization. The organization doesn't have any PC security yet, so she knows to begin with the very nuts and bolts.

1.2 Why it is important?

Think your business is too little to even think about attracting dangers? Enormous misstep. Digital lawbreakers don't by and large target people or organizations - they target vulnerabilities. A business of two is as inclined to assault as a huge company if a weakness is recognized.

A key point is that digital assaults are computerized. They always test for shortcomings 24x7. Also, similar to the cyborg in the Eliminator motion picture, they completely won't stop.

CHAPTER 2

Security Cracking

2.1 Popular security breaking techniques:

Dictionary attack:

The lexicon assault utilizes a basic document containing words that can be found in a word reference, henceforth its somewhat direct name. As it were, this assault utilizes precisely the sort of words that numerous individuals use as their secret phrase. Shrewdly gathering words, for example, "letmein" or "super administratorguy" won't keep your secret phrase from being split along these lines – well, not for in excess of a couple of additional seconds.

Brute force attack:

Like the word reference assault, the animal power assault accompanies a special reward for the programmer. Rather than basically utilizing words, an animal power assault gives them a chance to identify non-lexicon words by working through all conceivable alpha-numeric blends from aaa1 to zzz10.

It's not speedy, gave your secret phrase is over a bunch of characters long, however it will reveal your secret phrase in the long run. Animal power assaults can be abbreviated by tossing extra registering pull, as far as both handling power – including saddling the intensity of your video card GPU – and machine numbers, for example, utilizing appropriated figuring models like online bitcoin diggers.

Phishing:

There's a simple method to hack: approach the client for his or her secret word. A phishing email drives the clueless peruser to a faked sign in page related with whatever administration it is the programmer needs to get to, mentioning the client to put right some horrible issue with their security. That page at that point skims their secret phrase and the programmer can go use it for their own motivation.

Why try heading off to the inconvenience of splitting the secret phrase when the client will joyfully give it you in any case?

Social engineering:

Social building takes the entire "ask the client" idea outside of the inbox that phishing will in general stay with and into this present reality.

A most loved of the social designer is to call an office acting like an IT security tech fellow and just request the system get to secret phrase. You'd be astounded at how frequently this functions. Some even have the fundamental gonads to wear a suit and name identification before strolling into a business to ask the assistant a similar inquiry up close and personal.

Malware:

A keylogger, or screen scrubber, can be introduced by malware which records all that you type or takes screen captures amid a login procedure, and after that advances a duplicate of this document to programmer focal.

Some malware will search for the presence of an internet browser customer secret word record and duplicate this which, except if appropriately scrambled, will contain effectively available spared passwords from the client's perusing history.

CHAPTER 3

Introduction of Genetic Algorithm

3.1 What is genetic algorithm?

In a software engineering and activities explore, a hereditary calculation of genetic algorithm is a meta heuristic motivated by the procedure of regular choice that has a place with the bigger class of developmental calculations.

In other word, we can say GA is a technique for understanding both obliged and unconstrained enhancement issues that depends on normal determination, the procedure that drives natural development.

The GA over and over alters a populace of individual arrangements. Each progression, the hereditary calculation chooses people aimlessly from the present populace to be guardians and utilizations them to create the youngsters for the people to come.

3.2 Basic Steps of genetic algorithm

If we want to know about genetic algorithm we should know about some basic things:

1. Initial population
2. Fitness function
3. Selection
4. Crossover
5. Mutation

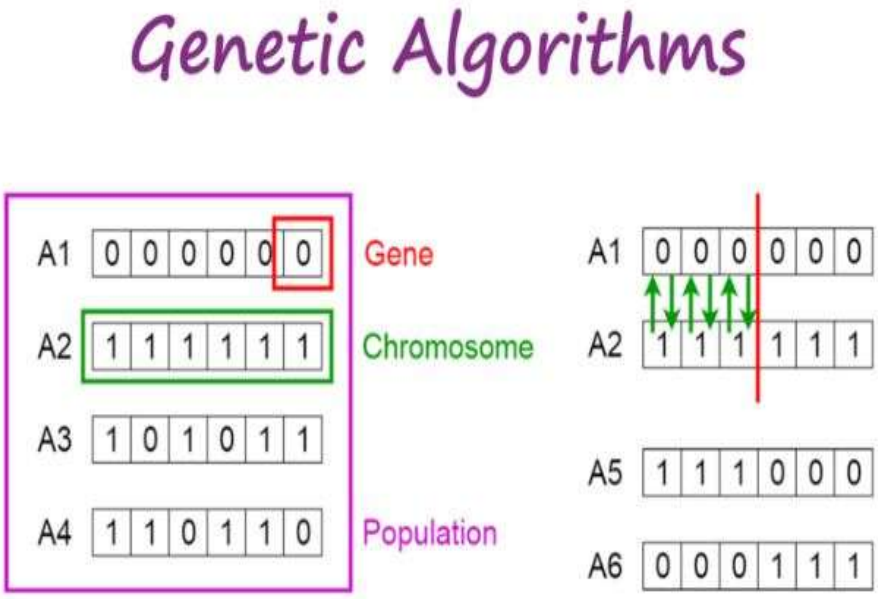


Figure 3.1:Basic steps of genetic algorithm

Initial Population:

Initial Population is the initial phase in the Hereditary Calculation Procedure. Populace is a subset of arrangements in the present generation. The procedure starts with a lot of people which is known as a Populace. Every individual is an answer for the issue you need to tackle. An individual is portrayed by a lot of parameters known as Qualities. Qualities are joined into a string to shape a Chromosome. When managing hereditary calculations, the assorted variety of the populace ought to be kept up else it may prompt a condition known as Untimely Intermingling. In a hereditary calculation, the arrangement of qualities of an individual is spoken to utilizing a string, as far as a letters in order. Typically, twofold qualities are utilized. We state that we encode the qualities in a chromosome.

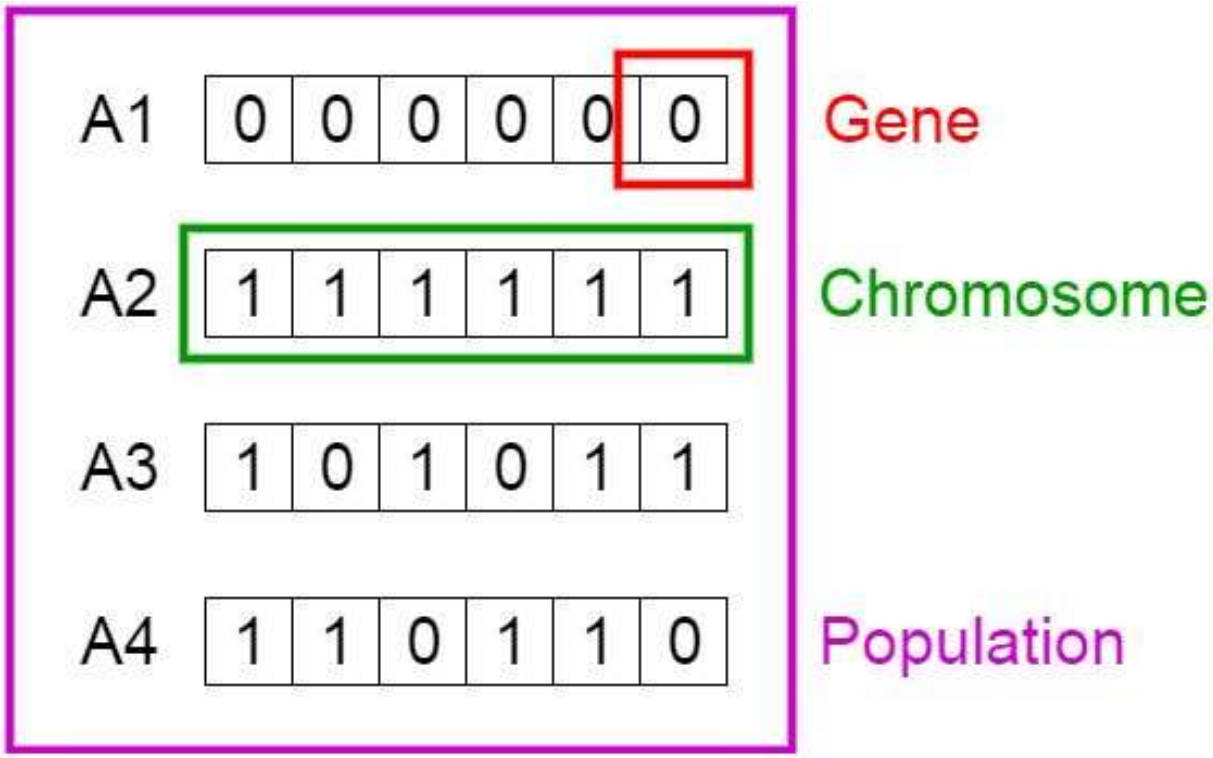


Figure 3.2:Initialise population

Fitness Function:

The wellness work decides how fit an individual is (the capacity of a person to rival different individual). It gives a wellness score to every person. The likelihood that an individual will be chosen for propagation depends on its wellness score. Fitness Capacity assesses how close a given arrangement is to the ideal arrangement of the ideal issue. It decides how fit an answer is.

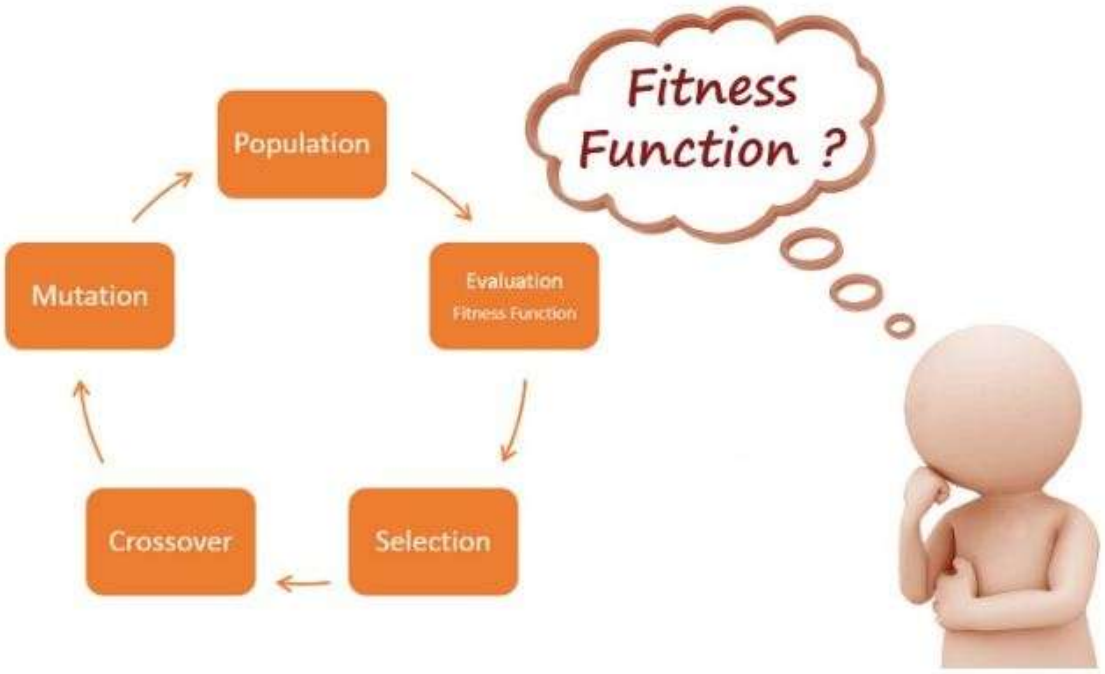


Figure 3.3:Fitness Function

In hereditary calculations, every arrangement is commonly spoken to as a string of double numbers, known as a chromosome. We need to test these arrangements and think of the best arrangement of answers for take care of a given issue. Every arrangement, in this way, should be granted a score, to show how close it came to meeting the general detail of the ideal arrangement. This score is created by applying the wellness capacity to the test, or results acquired from the tried arrangement.

A fitness function have some characteristics:

1. Have to be fast to compute.
2. It should quantitatively gauge how fit a given arrangement is or how fit people can be delivered from the given arrangement.

Sometimes, computing the wellness work straightforwardly probably won't be conceivable because of the inalienable complexities of the current issue. In such cases, we do wellness estimation to suit our necessities. The accompanying picture demonstrates the wellness figuring for an answer of the 0/1 Rucksack. It is a straightforward wellness work which just entireties the benefit estimations of the things being picked (which have a 1), filtering the components from left to directly till the backpack is full.

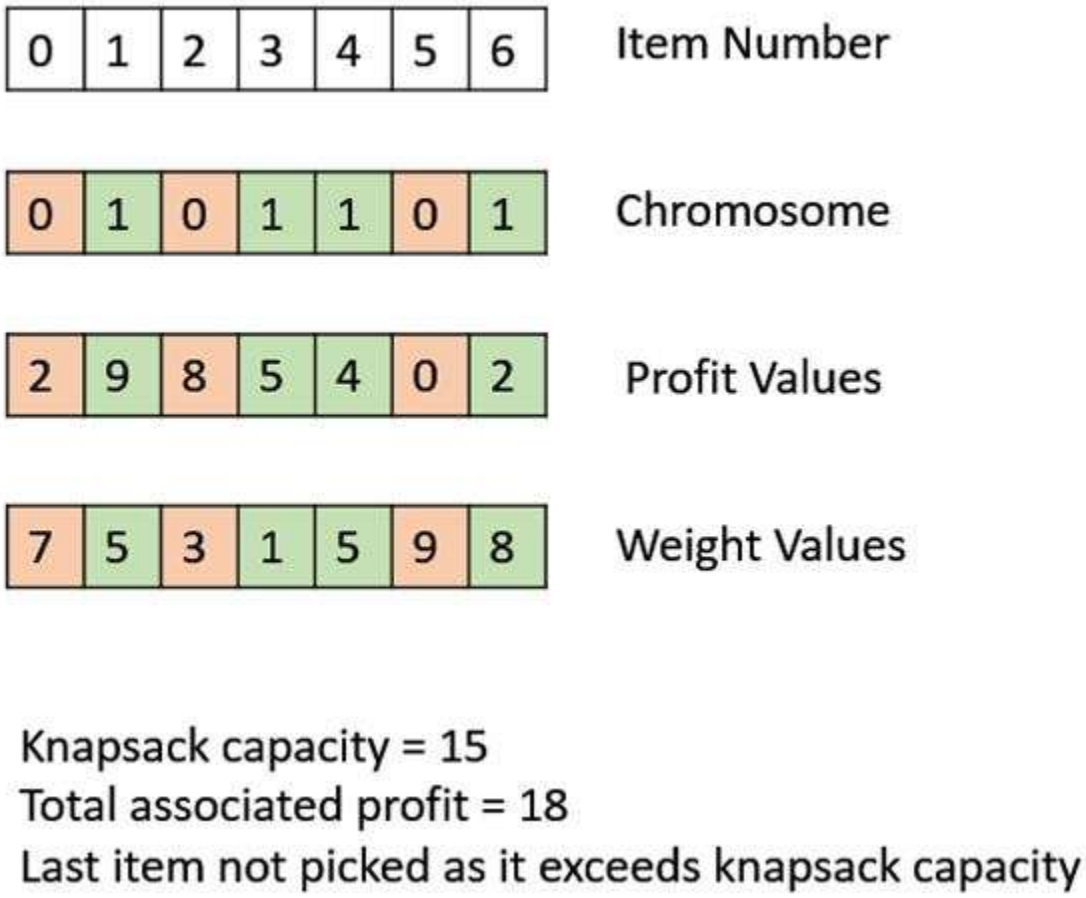


Figure 3.4:Fitness Function

Selection:

Selection is the way toward choosing guardians which mate and recombine to make off-springs for the people to come. Parent choice is significant to the intermingling rate of the GA as great guardians drive people to a superior and fitter arrangements. Choice is the phase of a hereditary

calculation in which singular genomes are looked over a populace for later rearing (utilizing the hybrid administrator).

Notwithstanding, care ought to be taken to avert one incredibly fit arrangement from assuming control over the whole populace in a couple of ages, as this prompts the arrangements being near each other in the arrangement space in this manner prompting lost decent variety. Keeping up great decent variety in the populace is very significant for the accomplishment of a genetic algoithm. This taking up of the whole populace by one incredibly fit arrangement is known as untimely combination and is an unwanted condition in a genetic algorithm.

Some Selection techniques:

1. Roulette wheel selection:

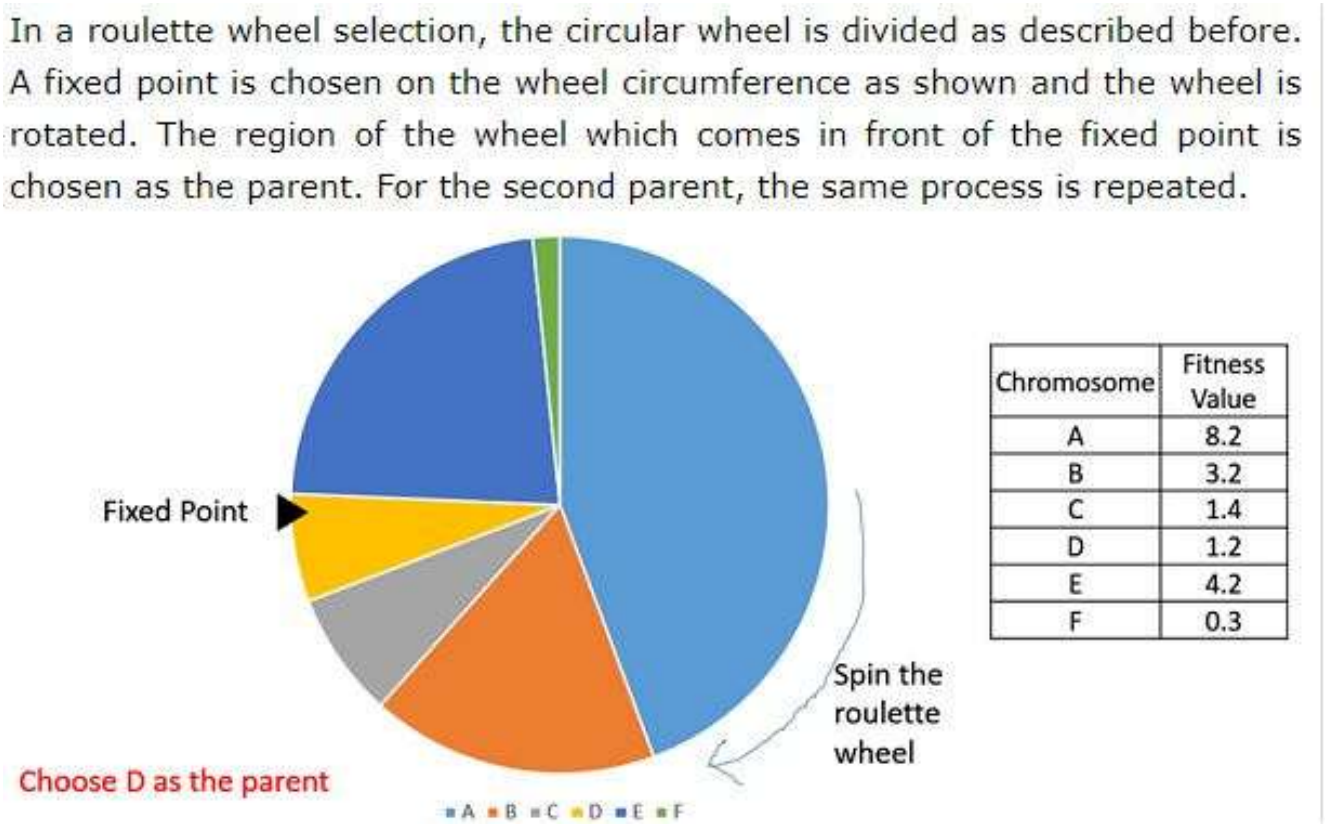


Figure 3.5:Selection in GA

2. Tournament Selection:

In K-Way tournament selection, we select K individuals from the population at random and select the best out of these to become a parent. The same process is repeated for selecting the next parent. Tournament Selection is also extremely popular in literature as it can even work with negative fitness values.

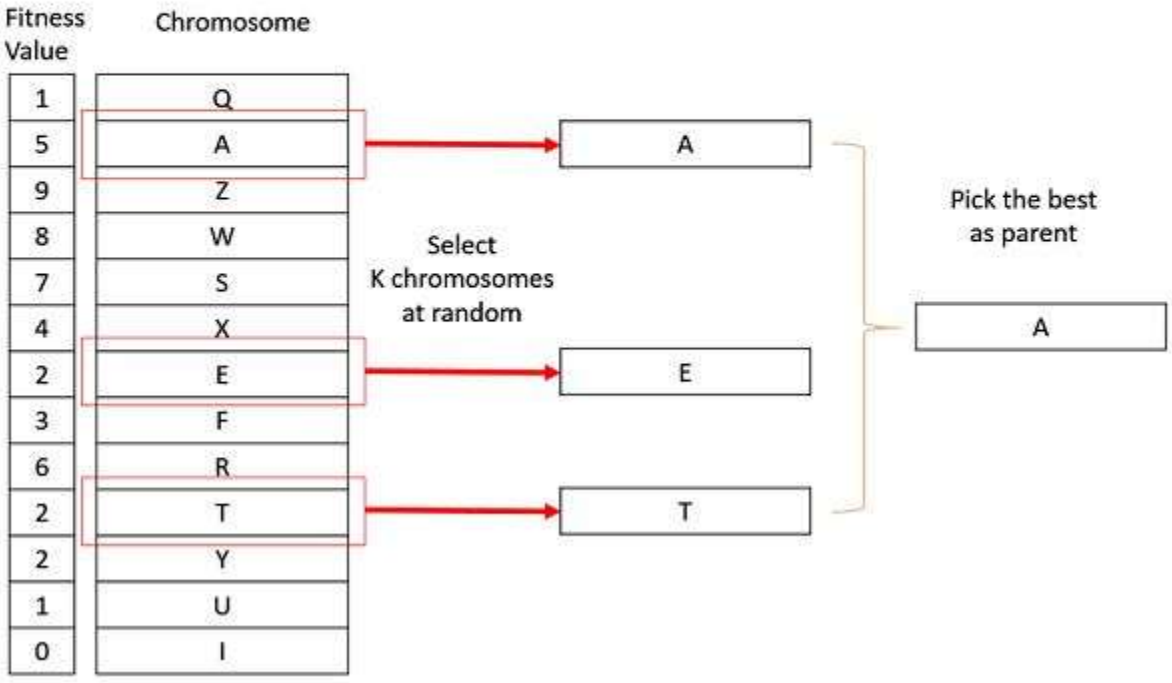


Figure 3.6:Tournament Selection

Crossover:

In genetic algorithm and transformative calculation, hybrid, likewise called recombination, is a hereditary administrator used to join the hereditary data of two guardians to produce new posterity. It is one approach to stochastically produce new arrangements from a current populace, and comparable to the hybrid that occurs amid sexual propagation in science. Arrangements can likewise be produced by cloning a current arrangement, which is undifferentiated from abiogenetic proliferation. Recently created arrangements are normally transformed before being added to the populace. The hybrid administrator is practically equivalent to proliferation and organic hybrid. In this more than one parent is chosen and at least one off-springs are created utilizing the hereditary material of the guardians. Hybrid is typically connected in a GA with a high likelihood.

Some techniques of crossover:

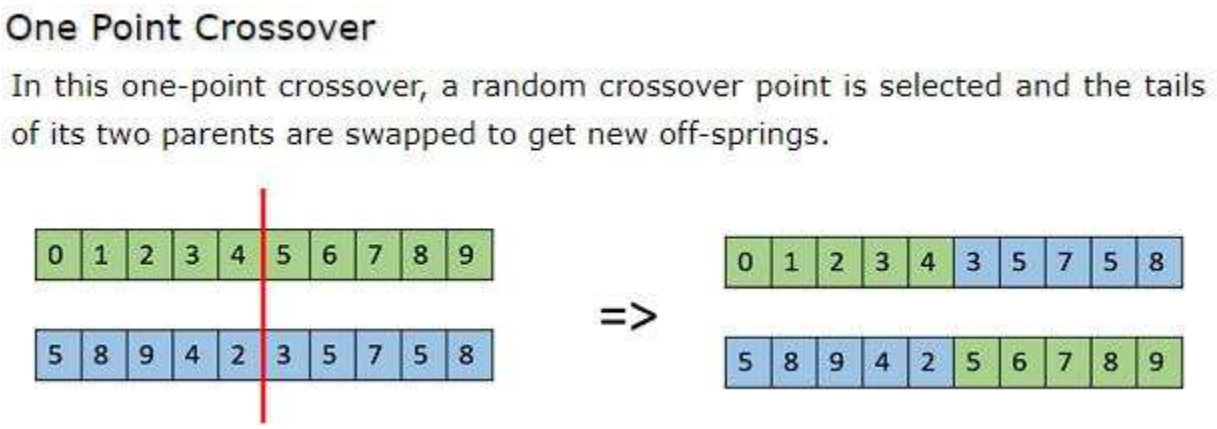
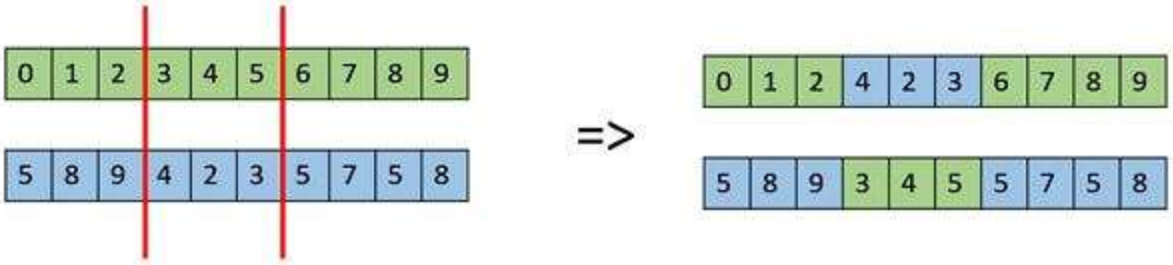


Figure 3.7:Crossover in GA

Multi Point Crossover

Multi point crossover is a generalization of the one-point crossover wherein alternating segments are swapped to get new off-springs.



Uniform Crossover

In a uniform crossover, we don't divide the chromosome into segments, rather we treat each gene separately. In this, we essentially flip a coin for each chromosome to decide whether or not it'll be included in the off-spring. We can also bias the coin to one parent, to have more genetic material in the child from that parent.

Diverse calculations in transformative calculation may utilize distinctive information structures to store hereditary data, and each hereditary portrayal can be recombined with various hybrid administrators. Run of the mill information structures that can be recombined with hybrid are bit exhibits, vectors of genuine numbers, or trees.

Mutation:

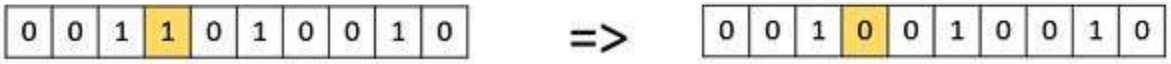
Transformation is a hereditary administrator used to keep up hereditary assorted variety from one age of a populace of hereditary calculation chromosomes to the following. It is similar to organic transformation. Change modifies at least one quality qualities in a chromosome from its underlying state. In transformation, the arrangement may change totally from the past arrangement. Thus GA can go to a superior arrangement by utilizing transformation. Change happens amid development as per a client determinable transformation likelihood. Change is the piece of the GA which is identified with the "investigation" of the pursuit space. It has been seen that change is basic to the union of the GA while hybrid isn't.

In straightforward terms, transformation might be characterized as a little irregular change in the chromosome, to get another arrangement. It is utilized to keep up and present decent variety in the hereditary populace and is typically connected with a low likelihood – pm. On the off chance that the likelihood is high, the GA gets decreased to an arbitrary pursuit.

Some of mutation techniques:

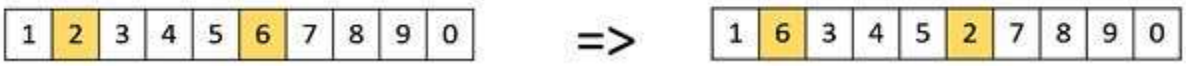
Bit Flip Mutation

In this bit flip mutation, we select one or more random bits and flip them. This is used for binary encoded GAs.



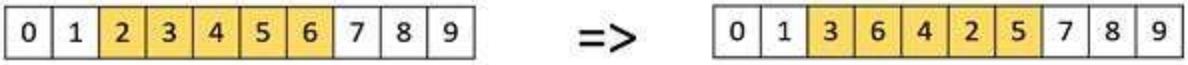
Swap Mutation

In swap mutation, we select two positions on the chromosome at random, and interchange the values. This is common in permutation based encodings.



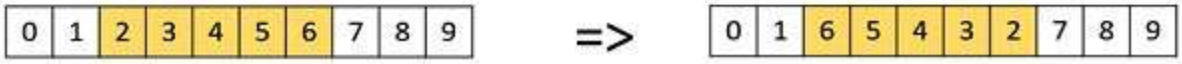
Scramble Mutation

Scramble mutation is also popular with permutation representations. In this, from the entire chromosome, a subset of genes is chosen and their values are scrambled or shuffled randomly.



Inversion Mutation

In inversion mutation, we select a subset of genes like in scramble mutation, but instead of shuffling the subset, we merely invert the entire string in the subset.



CHAPTER 4
Importance of genetic Algorithm

4.1 Why do we use genetic algorithm:

Let's discuss about a problem known as "Infinite monkey theorem" Assume that has 50 keys, and the word to be composed is "banana". In the event that the keys are squeezed haphazardly and freely, it implies that each key has an equivalent possibility of being squeezed. At that point, the shot that the principal letter composed is 'b' is 1/50, and the possibility that the second letter composed is 'an' is additionally 1/50, etc. Consequently, the shot of the initial six letters spelling "banana" is:

(1/50) × (1/50) × (1/50) × (1/50) × (1/50) × (1/50) = (1/50)6 = 1/15 625 000, i.e., short of what one out of 15 billion. Yet at the same time not zero, thus a result is as yet conceivable.

The monkey will type the word 'banana' 1 out of 15,625,000,000 occasions. Presently given us a chance to assume the monkey hits a key for every second the measure of time being taken for this occasion to happen in the most pessimistic scenario is 495 years approx.

Now in the event that is recreate a computer program that the above issue and complete a Brute Force look for "banana", the measure of calculation and time included will be huge.

But, in this event it need to type the equivalent, it will take me under 6 seconds to do it. Why? Since I know letters, and I know the word banana and its spelling.

So, would it be able to utilize Evolution Theory and improve that computer program? Indeed, and this is on account of the idea of Genetic Algorithms.

From that "Infinite monkey theorem" maybe we get some idea why do we need genetic algorithm. Therefor genetic algorithm have lots of advantages. It can discover fit arrangements in a less time. (fit arrangements are arrangements which are great as indicated by the characterized heuristic) The irregular change certifications to some degree that we see a wide scope of arrangements. Coding them is actually simple contrasted with different calculations which does likewise work.

4.2 There some advantages of genetic algorithm:

1. Genetic Algorithm is easy to understand;
2. It search from the population point, not from a single point;
3. It use payoff or objective information, not derivatives;
4. It supports multi objective optimization;
5. It gives a good result in noisy environments;
6. It works for both types of problems discrete and continuous;
7. It can operates in various representation and ext.

Scientists use Genetic Algorithm since they rush to execute and appear to work alright by and by. In the event that you just arrangement on upgrading a solitary issue a solitary time, at that point Genetic Algorithm is sensible. On the off chance that you have various issues, for example, settling as new data enables you to improve your reproduction, at that point Genetic Algorithm is likely not the best decision. On the off chance that you know something about the issue, at that point use it. On the off chance that you don't, at that point think about a Bayesian Global Optimization technique or a Derivative Free Method. It may take more work to get your recreation to interface pleasantly to the solver, however the long haul results will be justified, despite all the trouble.

Hereditary Algorithms have certain characteristics that makes it most mainstream among the diverse GA. Genetic Algorithm utilizes both hybrid and transformation administrators which makes its populace progressively assorted and in this way increasingly invulnerable to be caught in a neighborhood optima. In principle the decent variety additionally encourages the calculation to be quicker in coming to the worldwide optima since it will enable the calculation to investigate the arrangement space quicker. This does not imply that decent variety is constantly invited, close to the start of the hunt we need high assorted variety in the populace and along these lines hybrid rate ought to be high to investigate the arrangement area while at the finish of the inquiry the assorted variety ought to be kept negligible in light of the fact that at this stage the populace contains collected data about the ideal arrangement that was amassed over the ages and high assorted variety will cause the loss of this gathered data. Close to the finish of the ages the populace ought to be focused on a few nearby ideal (ideally the worldwide optima is one of them) and consequently we should abuse this circumstance by applying change to seek in the area.

For purpose of fruition, Hereditary calculations are reprimanded on the bases that transformative procedure is moderate in nature and consequently we can't rely upon it to take care of genuine issues. This thinking does not mull over the distinction in time scale among regular and counterfeit frameworks

In our growing world, day by day we are depend on AI. Without AI we can’t think a little. In AI simulation we need genetic algorithm. Genetic algorithm work by mimicking the rationale of Darwinian determination, where just the best are chosen for replication. Over numerous ages, regular populaces advance as indicated by the standards of characteristic choice and expressed by Charles Darwin in The Origin of Species. Just the most fit components in a populace are probably going to endure and create posterity, in this way transmitting their natural heredity to new ages. Utilized when you can code characteristics that you think may add to a particular, non-evolving issue. The accentuation is on having the capacity to code these characteristics (at times you comprehend what they are) and that the issue is to a substantial degree perpetual (generally developments don't unite). Hereditary calculations can address convoluted issues with numerous factors and countless results by recreating the transformative procedure of "survival of the fittest" to achieve a characterized objective. They work by producing numerous irregular responses to an issue, disposing of the most noticeably bad and cross-pollinating better answers. Rehashing this end and recovery process step by step improves the nature of the responses to an ideal or close ideal condition. Genetic algorithm are well at proficiently looking through a substantial state-space of arrangements, and combining on at least one great arrangements, however not really the 'best' arrangement.

Figuring terms, a hereditary calculation actualizes the model of calculation by having varieties of bits or characters (twofold string) to speak to the chromosomes. Each string speaks to a potential arrangement. The hereditary calculation at that point controls the most encouraging chromosomes hunting down improved arrangements.

A run of the mill utilization of this innovation is endeavor to decide the best course for conveying item from indicate A point B, when the conditions may intermittently change. These conditions could be identified with climate, street conditions, traffic stream, surge hour, and so forth. Ordinarily as well as could be expected be the quickest, briefest, most picturesque, most financially savvy, or a blend thereof.

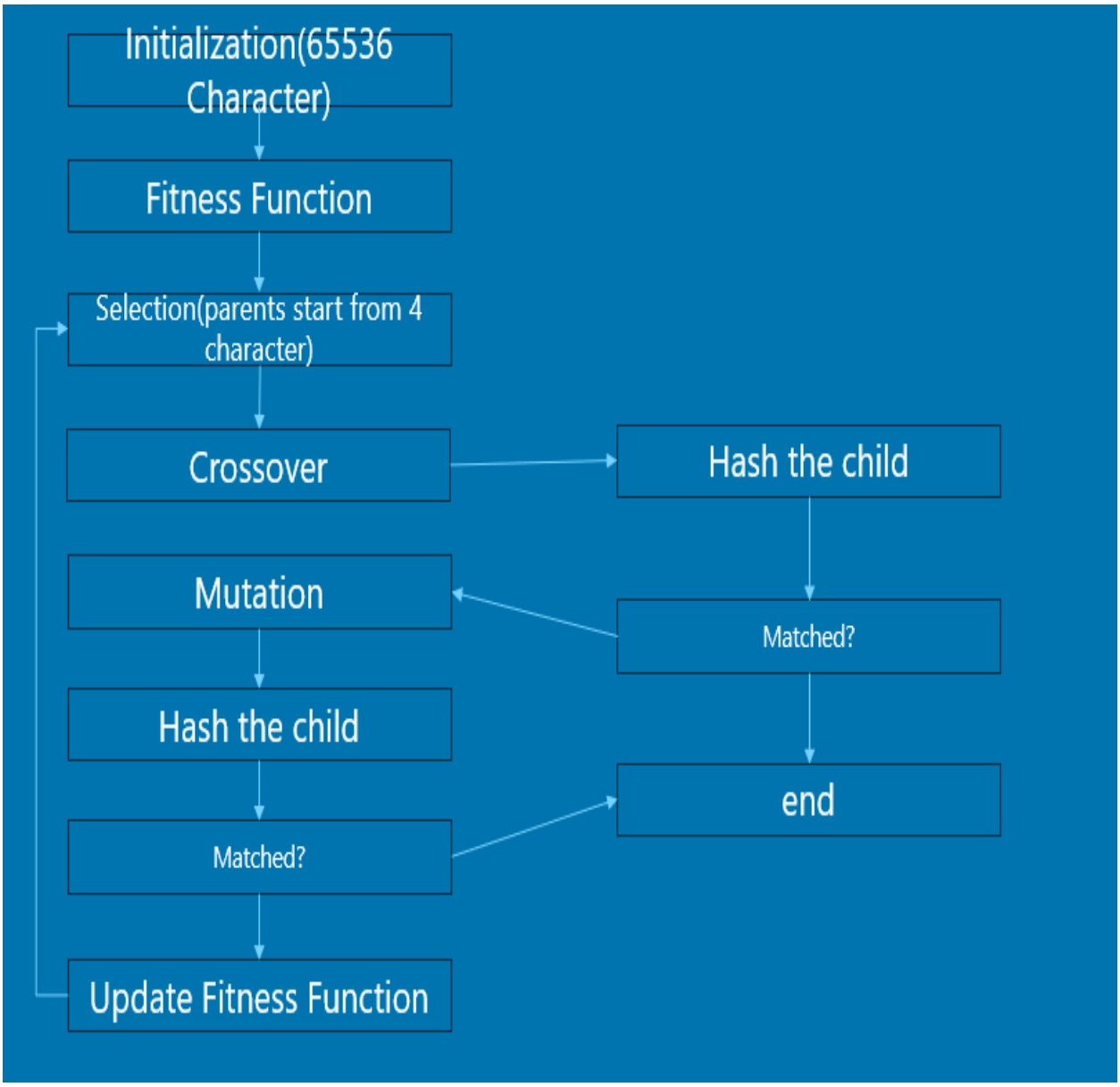
Some Important Application field On Genetic Algorithm:

- Robotics;
- Financial Planning;
- Travelling Salesman Problem and sequence scheduling;
- code Breaking;

CHAPTER 5

Algorithm and Code

5.1 Algorithm:



5.2 Code in C:

```
#include <stdio.h>
#include <stdlib.h>

#define NUMBER_ORGANISMS 100
#define NUMBER_GENES     20
#define ALLELES          4
#define MUTATION_RATE    0.001

#define MAXIMUM_FITNESS NUMBER_GENES
#define FALSE 0
#define TRUE  1


// global variables
char **currentGeneration, **nextGeneration;
char *modelOrganism;
int *organismsFitnesses;
int totalOfFitnesses;


// function declarations
void AllocateMemory(void);
int DoOneRun(void);
  void InitializeOrganisms(void);
  int EvaluateOrganisms(void);
  void ProduceNextGeneration(void);
    int SelectOneOrganism(void);

// functions
int main(){
  int finalGeneration;
  AllocateMemory();
  finalGeneration = DoOneRun();
  printf("The final generation was: %d\n",
         finalGeneration);
}


void AllocateMemory(void){
  int organism;

  currentGeneration =
    (char**)malloc(sizeof(char*) * NUMBER_ORGANISMS);
  nextGeneration =
    (char**)malloc(sizeof(char*) * NUMBER_ORGANISMS);
  modelOrganism =
    (char*)malloc(sizeof(char) * NUMBER_GENES);
  organismsFitnesses =
    (int*)malloc(sizeof(int) * NUMBER_ORGANISMS);

  for(organism=0; organism<NUMBER_ORGANISMS; ++organism){
    currentGeneration[organism] =
      (char*)malloc(sizeof(char) * NUMBER_GENES);
    nextGeneration[organism] =
```

```
      (char*)malloc(sizeof(char) * NUMBER_GENES);
  }
}


int DoOneRun(void){
  int generations = 1;
  int perfectGeneration = FALSE;

  InitializeOrganisms();

  while(TRUE){
    perfectGeneration = EvaluateOrganisms();
    if( perfectGeneration==TRUE ) return generations;
    ProduceNextGeneration();
    ++generations;
  }
}


void InitializeOrganisms(void){
  int organism;
  int gene;

  // initialize the normal organisms
  for(organism=0; organism<NUMBER_ORGANISMS; ++organism){
    for(gene=0; gene<NUMBER_GENES; ++gene){
      currentGeneration[organism][gene] = rand()%ALLELES;
    }
  }

  // initialize the model organism
  for(gene=0; gene<NUMBER_GENES; ++gene){
    modelOrganism[gene] = rand()%ALLELES;
  }
}
int EvaluateOrganisms(void){
  int organism;
  int gene;
  int currentOrganismsFitnessTally;

  totalOfFitnesses = 0;

  for(organism=0; organism<NUMBER_ORGANISMS; ++organism){
    currentOrganismsFitnessTally = 0;

    // tally up the current organism's fitness
    for(gene=0; gene<NUMBER_GENES; ++gene){
      if( currentGeneration[organism][gene]
          == modelOrganism[gene] ){
        ++currentOrganismsFitnessTally;
      }
    }

    // save the tally in the fitnesses data structure
    // and add its fitness to the generation's total
    organismsFitnesses[organism] =
```

```
      currentOrganismsFitnessTally;
    totalOfFitnesses += currentOrganismsFitnessTally;

    // check if we have a perfect generation
    if( currentOrganismsFitnessTally == MAXIMUM_FITNESS ){
      return TRUE;
    }
  }
  return FALSE;
}
void ProduceNextGeneration(void){
  int organism;
  int gene;
  int parentOne;
  int parentTwo;
  int crossoverPoint;
  int mutateThisGene;

  // fill the nextGeneration data structure with the
  // children
  for(organism=0; organism<NUMBER_ORGANISMS; ++organism){
    parentOne = SelectOneOrganism();
    parentTwo = SelectOneOrganism();
    crossoverPoint  =  rand() % NUMBER_GENES;

    for(gene=0; gene<NUMBER_GENES; ++gene){

      // copy over a single gene
      mutateThisGene = rand() % (int)(1.0 / MUTATION_RATE);
      if(mutateThisGene == 0){

        // we decided to make this gene a mutation
        nextGeneration[organism][gene] = rand() % ALLELES;
      } else {
        // we decided to copy this gene from a parent
        if (gene < crossoverPoint){
          nextGeneration[organism][gene] =
            currentGeneration[parentOne][gene];
        } else {
          nextGeneration[organism][gene] =
            currentGeneration[parentTwo][gene];
        }
      }
    }
  }

  // copy the children in nextGeneration into
  // currentGeneration
  for(organism=0; organism<NUMBER_ORGANISMS; ++organism){
    for(gene=0; gene<NUMBER_GENES; ++gene){
      currentGeneration[organism][gene] =
        nextGeneration[organism][gene];
    }
  }
}
```

```
int SelectOneOrganism(void){
  int organism;
  int runningTotal;
  int randomSelectPoint;

  runningTotal = 0;
  randomSelectPoint = rand() % (totalOfFitnesses + 1);

  for(organism=0; organism<NUMBER_ORGANISMS; ++organism){
    runningTotal += organismsFitnesses[organism];
    if(runningTotal >= randomSelectPoint) return organism;
  }
}
```

CHAPTER 6
Limitations

6.1 Limitations of Genetic Algorithm:

Everything has to be some limitations so genetic algorithm has also some limitations in spite of the fact that Genetic calculations has turned out to be a quick and incredible critical thinking approach, a few impediments are discovered installed in it. Some limitation are shown below:

- First, thought in making a hereditary calculation is characterizing a portrayal for the issue. The language used to determine hopeful arrangements must be powerful; i.e., it must almost certainly endure irregular changes with the end goal that deadly mistakes don't result.

- Most noteworthy impediment of hereditary calculations is the coding of the wellness work with the goal that a higher wellness can be achieved and better answers for the current issue are delivered. A wrong decision of the wellness capacity may prompt basic issues, for example, unfit to discover the answer for an issue or far more terrible, restoring a wrong answer for the issue.

- Alongside settling on a decent decision of wellness work, different parameters of a Genetic Algorithm like populace size, transformation and hybrid rate should likewise be picked with consideration. A little populace size won't give the Genetic Algorithm enough arrangement space to deliver precise outcomes. A high recurrence of hereditary change or poor choice plan will bring about upsetting the gainful pattern and the populace may enter blunder disaster, changing unreasonably quick for determination to regularly realize assembly.

- It isn't fitting to utilize Genetic calculations for investigative issues. In spite of the fact that Genetic calculations can discover exact answers for these sort of issues, conventional diagnostic techniques can locate similar arrangements in less time with couple of computational advances.

- Untimely intermingling is another issue GA researchers need to consider when arrangements of Generic calculation are created. It might happen that an individual can be substantially more fit than any of its rivals. So this individual may replicate a lot progressively new people lessening the assorted variety of the new populace and driving the calculation to unite on the nearby ideal that speaks to that specific individual as opposed to looking through the wellness scene altogether enough to locate the worldwide ideal. This sort of wastefulness is for the most part found in little issues where even possibility varieties in multiplication rate may make one genotype become prevailing over others.

6.2 Moore's law:

Moore's law is the observation that the number of transistors in a dense integrated circuit doubles about every two years.

Curve of Moore's law:

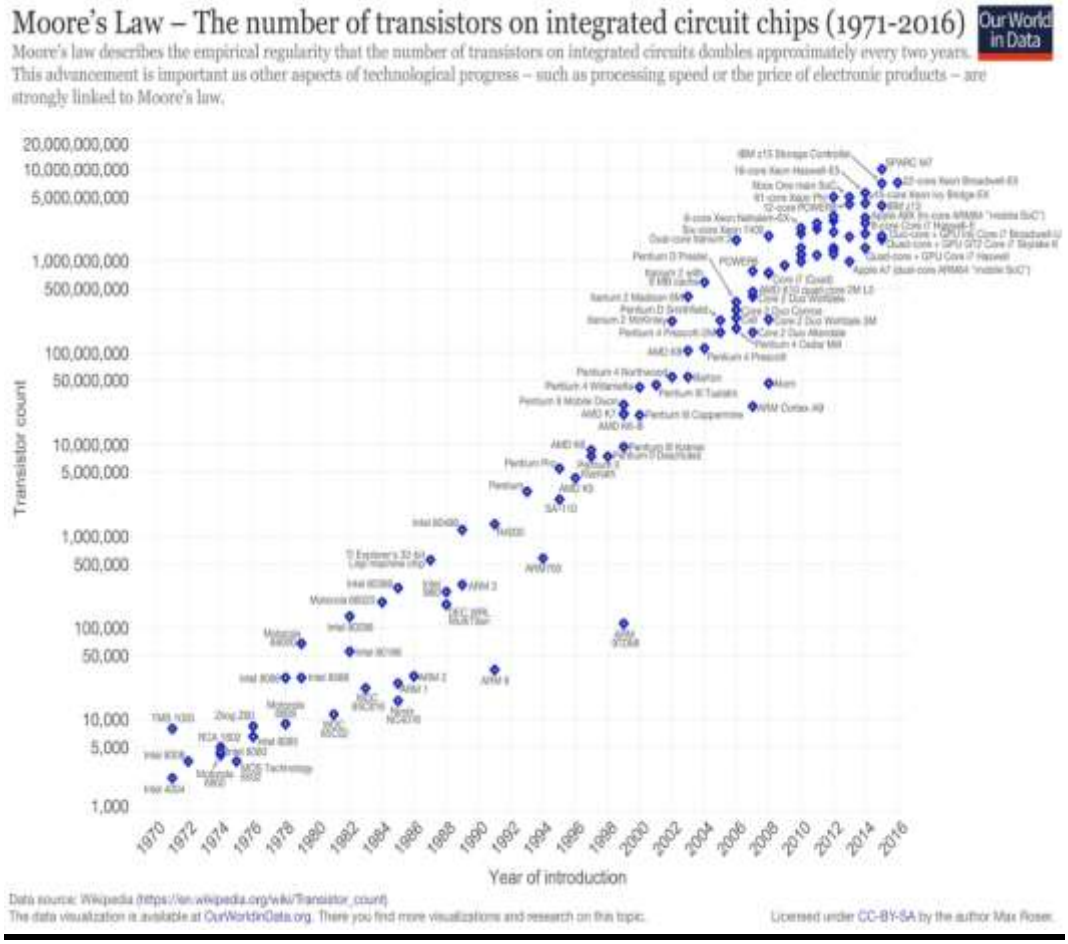


Figure 6.1:Curve of moore's Law

CHAPTER 7

Introduction Of Quantum Computer

7.1 What is quantum computer:

Quantum computer is the utilization of quantum-mechanical marvels, for example, superposition and snare to perform calculation. A quantum PC is utilized to perform such calculation, which can be executed hypothetically or physically.

An established PC has a memory comprised of bits, where each piece is spoken to by either a one or a zero. A quantum PC, then again, keeps up a succession of qubits, which can speak to a one, a zero, or any quantum superposition of those two qubit states;[8]:13– 16 a couple of qubits can be in any quantum superposition of 4 states,[8]:16 and three qubits in any superposition of 8 states. As a rule, a quantum PC with {\displaystyle n} n qubits can be in any superposition of up to {\displaystyle 2^{n}} 2^{n} distinctive states.[8]:17 (This thinks about to a typical PC that must be in one of these {\displaystyle 2^{n}} 2^{n} states at any one time).

A quantum PC works on its qubits utilizing quantum entryways and estimation (which additionally adjusts the watched state). A calculation is made out of a fixed arrangement of quantum rationale entryways and an issue is encoded by setting the underlying estimations of the qubits, like how an established PC works. The estimation as a rule closes with an estimation, crumbling the arrangement of qubits into one of the {\displaystyle 2^{n}} 2^{n} eigenstates, where each qubit is zero or one, breaking down into an established state. The result can, in this manner, be at most {\displaystyle n} n traditional bits of data. On the off chance that the calculation did not finish with an estimation, the outcome is an imperceptibly quantum state. (Such imperceptibly states might be sent to different PCs as a feature of circulated quantum calculations.)

Quantum calculations are regularly probabilistic, in that they furnish the right arrangement just with a specific known probability.[9] Note that the term non-deterministic registering must not be utilized all things considered to mean probabilistic (figuring) on the grounds that the term non-deterministic has an alternate importance in software engineering.

A quantum PC with a given number of qubits is in a general sense not the same as an established PC made out of a similar number of traditional bits. For instance, speaking to the condition of a n-qubit framework on an established PC requires the capacity of 2n complex coefficients, while to portray the condition of a traditional n-bit framework it is adequate to give the estimations of the n bits, that is, just n numbers. In spite of the fact that this reality may appear to show that qubits can hold exponentially more data than their traditional partners, care must be taken not to disregard the way that the qubits are just in a probabilistic superposition of the majority of their states. This implies when the last condition of the qubits is estimated, they might be found in one of the conceivable setups they were in before the estimation. It is commonly off base to think about an arrangement of qubits as being in one specific state before the estimation. The qubits are in a superposition of states before any estimation is made, which straightforwardly influences the conceivable results of the calculation.

CONCLUSION

There is a proverb that “No System Is Secure”. I strongly believe with this kind of statement. All the system have faith on strong, unique hardest combination of symbols called password. Password cracking is difficult now a days but not impossible. Brute force suggest to make all the combination of symbols and then check. But is this so easy? Only from 50 symbols it may take(worst case) 452 years to find the word banana. In present computer system we use Unicode as encoding. Which means our computer can detect 65536 different symbols. In other word we can select our password from 65536 symbols. Now calculate how many configurations you may get. wait, Don’t try. Because it will approaches nearly infinity. So in this modern security system brute force won’t work actually. So what can we do? Genetic Algorithm has been introduced to overcome this problem which is inspired from the famous revolutionary theory of Charles Darwin. It may take only 6 seconds to find the word banana from the 50 symbols. But our job has not been done yet. If we apply Genetic Algorithm to find a password from 65536 symbol it may take(worst case) million billion billion years for regular computer. What can we do? Moore’s law tells us that transistors become double in every two years. But unfortunately it also won’t help us because we can’t wait for billion years. What is the ultimate solution? The only option we have that is switching transistor based computer system to quantum computer system. In quantum computing system computer works through Qbit. Qbit can not only be set to 0 or 1 but also be set to 1 and 0 which is called superposition. In this type of computing system finding a password from 65536 symbols is a matter of few seconds.

REFERENCES

1. https://www.safewise.com/home-security-faq/how-do-security-systems-work/ **[Online Accessed 15 Oct 2018]**

2. https://www.britannica.com/technology/computer-security**[Online Accessed 27 Oct 2018]**

3. https://en.wikipedia.org/wiki/Computer_security**[Online Accessed 5 Nov 2018]**

4. https://www.securitymagazine.com/articles/89255-five-typical-cyber-attack-techniques-used-against-business-travelers**[Online Accessed 14 Nov 2018]**

5. https://www.rapid7.com/fundamentals/types-of-attacks/**[Online Accessed 25 Nov 2018]**

6. http://www.crossdomainsolutions.com/cyber-security/tools-techniques/**[Online Accessed 3 Dec 2018]**

7. https://en.wikipedia.org/wiki/Cyberattack**[Online Accessed 10 Dec 2018]**

8. https://blog.netwrix.com/2018/05/15/top-10-most-common-types-of-cyber-attacks/**[Online Accessed 15 Dec 2018]**

9. https://en.wikipedia.org/wiki/Genetic_algorithm**[Online Accessed 17 Dec 2018]**

10. https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3**[Online Accessed 21 Dec 2018]**

11. https://www.mathworks.com/help/gads/what-is-the-genetic-algorithm.html**[Online Accessed 23 Dec 2018]**

12. https://www.geeksforgeeks.org/genetic-algorithms/**[Online Accessed 25 Dec 2018]**

13. https://www.tutorialspoint.com/genetic_algorithms/**[Online Accessed 30 Dec 2018]**

14. https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_introduction.htm **[Online Accessed 2 Jan 2019]**

15. https://www.kdnuggets.com/2018/03/introduction-optimization-with-genetic-algorithm.html **[Online Accessed 5 Jan 2019]**

16. https://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol1/hmw/article1.html **[Online Accessed 11 Jan 2019]**

17. https://www.sciencedirect.com/topics/engineering/genetic-algorithm **[Online Accessed 17 Jan 2019]**

18. https://en.wikipedia.org/wiki/Infinite_monkey_theorem **[Online Accessed 20 Jan 2019]**

19. https://whatis.techtarget.com/definition/Infinite-Monkey-Theorem **[Online Accessed 24 Jan 2019]**

20. https://www.techopedia.com/definition/20002/infinite-monkey-theorem **[Online Accessed 20 Feb 2019]**

21. https://en.wikipedia.org/wiki/Quantum_computing **[Online Accessed 27 Feb 2019]**

22. https://www.research.ibm.com/ibm-q/learn/what-is-quantum-computing/ **[Online Accessed 2 March 2019]**

23. https://www.technologyreview.com/s/612844/what-is-quantum-computing/ **[Online Accessed 5 March 2019]**

24. https://www.dwavesys.com/quantum-computing **[Online Accessed 20 March 2019]**

25. https://www.microsoft.com/en-us/quantum/what-is-quantum-computing **[Online Accessed 2 April 2019]**

26. https://venturebeat.com/2019/04/21/quantum-computing-is-a-marathon-not-a-sprint/ **[Online Accessed 3 April 2019]**

27. https://whatis.techtarget.com/definition/quantum-computing **[Online Accessed 4 April 2019]**

28. http://theconversation.com/in-the-future-everyone-might-use-quantum-computers-112063 **[Online Accessed 4 April 2019]**

29. https://www.brainz.org/15-real-world-applications-genetic-algorithms/ **[Online Accessed 5 April 2019]**

30. https://www.tutorialspoint.com/genetic_algorithms/genetic_algorithms_application_areas.htm **[Online Accessed 6 April 2019]**