Multiple Team Formation: A Bio-Inspired Approach

Minor Project Report

Submitted in partial fulfilment of the requirement for the award of the degree

BACHELOR OF ENGINEERING

(Computer Science Engineering)

Submitted to

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA (U.P.)

Submitted by

Jatin Agarwal	Nikita Sharma	Yash Jauhari
20103202	20103208	20103219

Under the supervision of

Dr Ankita Verma

December - 2022

CERTIFICATE OF APPROVAL

The undersigned certify that they have read and recommended to the Department of Computer Science Engineering for acceptance, a project report entitled “Multiple Team Formation Using Generic Algorithm” submitted by

Jatin Agarwal, Nikita Sharma and Yash Jauhari in partial fulfilment for the degree of Bachelor of Engineering in Computer Science Engineering.

……..

(Dr Ankita Verma)

Mentor

ASSISTANT PROFESSOR (SENIOR GRADE)

Department of Computer Science and Information Technology

……..

(Dr Vikas Saxena)

Head of Department

Professor & Head (CSE & IT)

Department of Computer Science and Information Technology

DECLARATION

We, students of "Bachelor of Engineering" in Jaypee Institute of Information Technology", session 2022-23, NOIDA (U.P.) hereby informed you that the work presented in this dissertation entitled "Multiple Team Formation using Genetic Algorithm" is the outcome of our work, is bonafide and correct to the best of our knowledge and this work has been carried out taking care of Engineering Ethics. The work presented does not infringe any patented work and has not been submitted to any other University or anywhere else for the award of any degree or any professional diploma.

Name of Scholars and Enrol. no.:

Jatin Agarwal	**Nikita Sharma**	**Yash Jauhari**
20103202	**20103208**	**20103219**

ACKNOWLEDGEMENT

We take an opportunity to acknowledge and extend our heartfelt gratitude to our guide and the pivot of this enterprise, Dr Ankita Verma, who is most responsible for helping us to complete this work.

She showed us different ways to approach the problem and the need to be persistent to accomplish our goal. Her discernment in the choice of topic, her confidence in us when we doubted ourselves and her admirable guidance are some cogent reasons that make us think over that without her support this thesis would be a chimera.

We are also thankful to Prof Vikas Saxena, Head of Department of Computer Science and Information Technology, for cooperation and support to complete this work. We would also like to express our thanks to Prof Yog Raj Sood, Vice-Chancellor (Chairperson), JIIT NOIDA for providing necessary facilities. We would also convey our thanks to Dr Alka Singhal and Dr Amanpreet Kaur of Department of Computer Science and Information Technology for their continuous support. Thanks are due to all the staff members and lab staff of Department of Computer Science and Information Technology JIIT, NOIDA for providing all help and support.

Name of Scholars and Enrol. no.:

Jatin Agarwal	**Nikita Sharma**	**Yash Jauhari**
20103202	**20103208**	**20103219**

ABSTRACT

In the electronic-world, “multi-functional team” refers to a group of people from various areas of work responsibility, working together and exchanging information through networks. The need for making multi-functional team is that the organisations often require a group of people with good cooperation skills across functional lines and it may happen that the members may not be in the same location. The formation of multi-functional teams has become a key issue in project team management. In theory, there exists no analytical solution to formation of a multi-functional team that deal with uncertain information environment. So, in this paper, to deal with these underlying complexities, a methodology has been developed. The methodology is based on Advanced Genetic Algorithms and Non-Dominated Sorting Algorithms. The Advanced Genetic Algorithm was used to form a team based on a single objective. It is a flawed approach because one can not form a multi-functional team based on a single point objective. Hence, we used a better and optimised approach, i.e., NSGA-II algorithm. In this approach, the members are selected on the basis of all of their skills, i.e., multi- objective approach. The team member is required to be competent in his/her work and also able to share other’s responsibility. In this paper we have compared both the approaches and researched on the better and optimised solution to the problem of “multi-functional team formation”.

TABLE OF CONTENT

Introduction

The paper revolves around the concept of forming a multi-functional team which meets all the requirements of a particular group/project. The problem is an extension of team formation to find a team of experts for multiple projects with particular skill requirement.

In the cited papers that we referred, the common issue that we observed was that the team members are considered to have only one skill and are selected based on their performance in that particular skill only.

To overcome this issue, we have considered that each participant in the selection process has multiple technical skills and have social skills as well. They are selected based on all their skills and their sociometric values with others. Also, he/she can work only for a single project, i.e. only full-time allocation is allowed. Social relationship between individuals is calculated by using sociometric matrix and we proposed a genetic algorithm based method in the lines of G.K. Awal et al [6] who proposed a similar approach for single team formation problem.

There existed one major issue of Cold-Start problem. The problem states that if a person is new in the industry, his/her technical skills may not be very sharpened or may not satisfy industry requirements. Also, he/she may not have sociometric values as a whole. So, our approach deals with the problem of Cold-Start problem and gives a better approach for the same. We have included 'newbies' of the industry in the team formation process as well even when their skill scores are zero, where they can learn from their peers and seniors, and get industry-ready.

Positive social relationship among individuals in a team is very important for the effective execution of a project and good social relationship within teams impacts project performance positively. Various methods have been proposed for solving team formation problem which considers the communication cost between individuals in a team as a measure of the effectiveness of team allocation for a project. Communication cost can be taken as the social distance between individuals, so clearly communication cost has to be reduced to improve the team performance. Social relationship is modelled as being an integer value between -2 and +2, quite similar to the approach introduced by J. L. Morena.

For a start, we have considered a group of teams as a single chromosome. We initialized population of chromosomes which will be used for in the entire research paper. The population is generated randomly out of experts indulged in selection process. On the basis of constraints of the project, the fitness values are generated on the basis of sociometric matrix having sociometric values between -2 to 2, other is skill matrix having the skill ratings which they rate themselves as an individual professional.

This data is generated through a google form, that was circulated well in advance. The inputs are attached further in this paper. Synthetic data was generated as online company data is not available on any site.

Now crossover between two chromosome is performed which is considered as parents and two off-springs are generated which is further included in process. Also, mutation is performed over off-springs.

On the basis of fitness value best chromosome is selected for further generation of chromosome (team). Selection is done independently on the basis of function-1, function-2 namely social values, skill rating(efficiency). Also, independent graphs are generated between generation and their respective fitness values of their social values, skill rating(efficiency), combined values of social values and skill rating(efficiency). All the above-mentioned process was part of Genetic Algorithm.

Now, in order to improve further we applied NSGA-II which is multi-objective optimization technique. The maximization of objective is been performed in which no team can dominate the generated solution teams. The optimization is done on the basis of social values, skill rating(efficiency). The crossover of the two parents is performed which produces two off-springs which is further included in population; sorted on the basis of fitness values. The mutation is performed on offspring in which one of the experts is selected and replaced to form a new chromosome. In this, the pareto front generated has the team which is either not dominated in social value and skill rating(efficiency) or it is best in their independent function (social skill, skill rating(efficiency)).

Objective

In recent years, the growth and popularity of social networks have created a new world of collaboration and communication. Team formation is a new research topic in the area of social network analysis. Consider there is a social network of experts and the goal is to form the best possible team out of them for a given project. The best solution is a team with the maximized communication cost and efficiency value within team members. The objective of this paper is to select team members based on both closeness and efficiency value.

Multiple team formation problem is an interesting problem in the area of social network analysis which asks how experts can be assigned among multiple projects so that the effectiveness of allocation can be maximized. In this paper, firstly, we consider an approach based on Genetic Algorithm and Sociometry and Efficiency matrix for calculating the social relationship between experts in each project. Then we enhanced our approach and used Non-Dominated Sorting Algorithm (NSGA-II) as well as Sociometry and Efficiency matrix. Due to unavailability of data set, experimentation has been performed on synthetic data set generated through a google form, circulated within our peers and results show that the allocation of experts for different projects is effective compared to the algorithm proposed in the literature review.

Experimental results on the google form generated dataset show that our approach completes in lower time compared to the previous methods.

Literature Review

[1] Team formation in social networks based on collective intelligence – an evolutionary approach, by Gaganmeet Kaur Awal & K. K. Bharadwaj

The tremendous growth of the social web has inspired research communities to discover social intelligence, which encompasses a wide spectrum of knowledge characterized by human interaction, communication and collaboration, thereby exploiting collective intelligence (CI) to support the successful existence of social communities on the Web. In this work, they address the team formation problem for generalized tasks where a set of experts is to be discovered from an expertise social network that can collaborate effectively to accomplish a given task. The concept of CI that emerges from these collaborations attempts to maximize the potential of the team of experts, rather than only aggregating individual potentials. Because the team formation problem is NP-hard, a genetic algorithm-based approach is applied to optimize computational collective intelligence in web-based social networks. To capture the essence of CI, a novel quantitative measure Collective Intelligence Index (CII) is proposed that takes two factors into account –the "enhanced expertise score" and the "trust-based collaboration score". This measure relates to the social interactions among experts, reflecting various affiliations that form a network of experts that help to drive creativity by deepening engagements through collaboration and the exchange of ideas and expertise, thereby enriching and enhancing the knowledge base of experts. The presented model also captures the teams' dynamics by considering trust, which is essential to effective interactions between the experts. The computational experiments are performed on a synthetic dataset that bears close resemblance to real-world expertise networks.

Short coming:

The proposed model lacked for experts possessing multiple skills, and also lacked to consider how the availability and personal attributes of experts affect the formation of a team. They failed to incorporate fuzzy trust-reputation measures in addition to trust, distrust and risk to determine the reliability of the team with respect to a given task. Exploiting the interaction patterns for the accurate interpretation of link strength between the experts also needed to be investigated.

[2] A Fast Approach for Multi-Objective Team Formation in Social Networks, by Bahareh Ashenagar, Ali Hamzeh, Negar Foroutan Eghlidi and Ardavan Afshar

In recent years, the growth and popularity of social networks have created a new world of collaboration and communication. Team formation is a new research topic in the area of social network analysis. Consider there is a social network of experts and the goal is to form the best possible team out of them for a given project. The best solution is a team with the minimized communication cost within team members. A social network is modelled as a graph, in which nodes represent experts and an edge between two nodes shows a prior collaboration of the two experts. This paper deals with selecting team members based on both closeness centrality and eigenvector centrality.

Short coming:

Results show that combined cost of a team for this method is more than approximation algorithm, MCC algorithm and MCC-Rare algorithm so they cannot conclude which algorithm is better. A measure that often used to assess the performance of different algorithm is hypervolume quality indicator. It is proposed by Zitzler et al and represents the volume of objective space dominated by solutions. Hypervolume quality indicator represents that each algorithm has larger area between his points and horizontal axes is better.

[3] A Method for Group Formation Using Genetic Algorithm, by Zhamri Che Ani, Azman Yasin, Mohd Zabidin Husin, Zauridah Abdul Hamid

Due to the increasing of complexity in software projects, group work is becoming more important in order to ensure quality software products can be delivered on time. Thus, in universities, group work is seen as a good preparation for students to enter industry because by working in group, it can reduce the individual workload, improve the ability to manage a project and enhance the problem-solving skills. However, due to lack of programming skills especially in Java programming language, most of the students' software project cannot be delivered successfully. To solve this problem, systematic group formation is one of the initial factors that should be considered to ensure that every group consists of quality individuals who are good in programming. This paper presents a method for group formation using genetic algorithm, where the members for each group will be generated based on the students' programming skill.

Short coming:

This research focused on group formation for IT or Computer Science students where programming skill is the most important criteria that have to be considered in order to form a solid group. In order to form balanced groups in a class, the genetic algorithm approach has been chosen in this study. This approach was totally based upon a condition that weak students will learn from strong students in a group and this proposed method was based upon studying of individual performance in a group.

[4] Application of Genetic Algorithms to the Multiple Team Formation Problem, by Jos´e G. M. Esgario, Iago E. da Silva, and Renato A. Krohling

Allocating of people in multiple projects is an important issue considering the efficiency of groups from the point of view of social interaction. In this paper, based on previous works, the Multiple Team Formation Problem (MTFP) based on sociometric techniques is formulated as an optimization problem taking into account the social interaction among team members. To solve the resulting optimization problem a Genetic Algorithm was proposed due to the NP-hard nature of the problem. The social cohesion is an important issue that directly impacts the productivity of the work environment. So, maintaining an appropriate level of cohesion keeps a group together, which will bring positive impacts on the results of a project. The aim of the proposal is to ensure the best possible effectiveness from the point of view of social interaction. In this way, the presented algorithm serves as a decision-making tool for managers to build teams of people in multiple projects.

Short coming:

In this paper, the algorithm was executed for datasets with different levels of complexity and the statistical results of cohesion and time are presented. The performance of the algorithm was compared with the exhaustive method in four out of seven datasets. The comparison with the rest of the datasets was impracticable due to the high computational time required by the exhaustive method for more complex problems. They lacked the use of the matrix of allocation with fractional values which will allow the distribution of the workload of employees in more than one group.

[5] Multiple Team Formation using an Evolutionary Approach, by Vivek Singh Baghel and S. Durga Bhavani

Multiple team formation problem is an interesting problem in the area of social network analysis which asks how experts can be assigned among multiple projects so that the effectiveness of allocation can be maximized. In this paper, the authors have considered an approach based on genetic algorithm and sociometry for calculating the social relationship between experts in each project. Due to unavailability of data set, experimentation has been performed on synthetic data set given by J. H. Guti´errez et al.

Short coming:

The work has proposed a new approach based on genetic algorithm for finding multiple teams for fulfilling the requirements of multiple projects such that the people of selected teams are socially connected with each other or social relationship between each other is positive. Sociometric matrix has been used for calculating the positive relationship between people but the algorithm does not consider new people in the group or the problem of 'cold start'. Further, it does not take into account the fact that even people with negative sociometric values can work together under certain circumstances.

Planning

Data collection/curation

Due to unavailability of dataset we have collected the data. The synthetic data is generated through Google form which was circulated in our batches. That google form contains various questions like Name and some other question in which expert has to rate his/her skill on scale of 0-10. Where,

0 means no experience in that particular skill and

10 means experts had done various projects and has lot of expertise in that particular skill.

Other than technical skills we also asked for various social skills and behaviour like punctuality, social skill, communication etc. which reduce the cold start problem.

As we have included freshers in the team on the basis of social skills and behaviour even if they don't have much technical skills related to industry, so if number of experts in team required in order to fill the requirements of the team according to requirement matrix we have freshers which hare selected on the basis of social skills. Also, the freshers can learn technical skills from their seniors and colleagues and get industry ready in order to fill the requirements of other matrix on the basis of technical as well as social skills

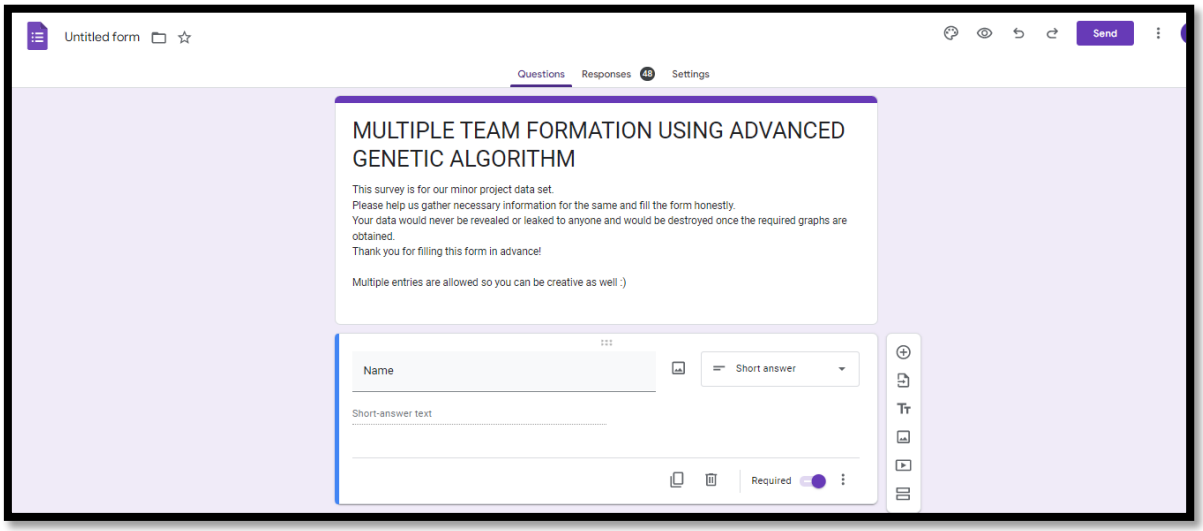


Fig. 1.1

Fig 1.2

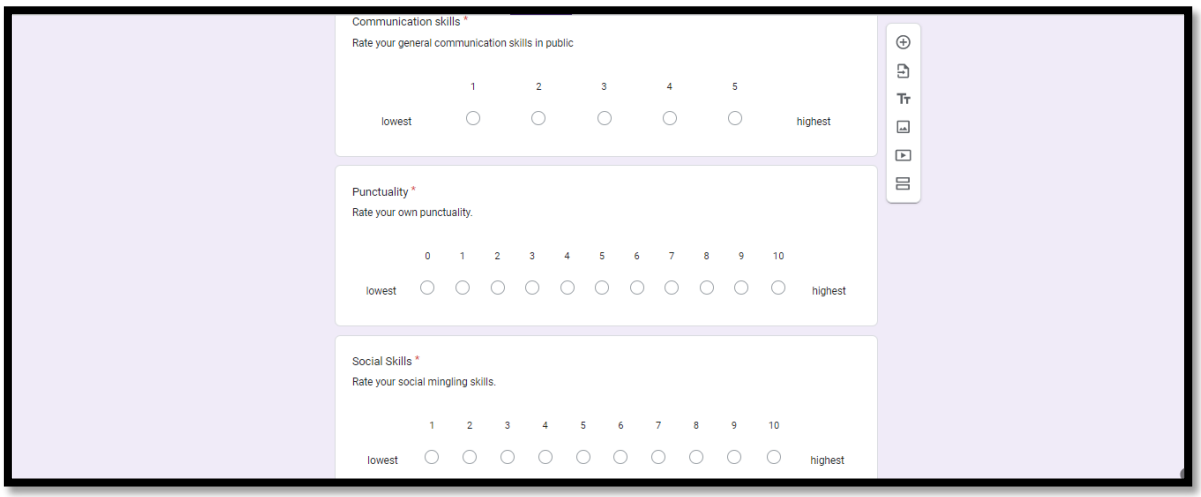


Fig 1.3

Response

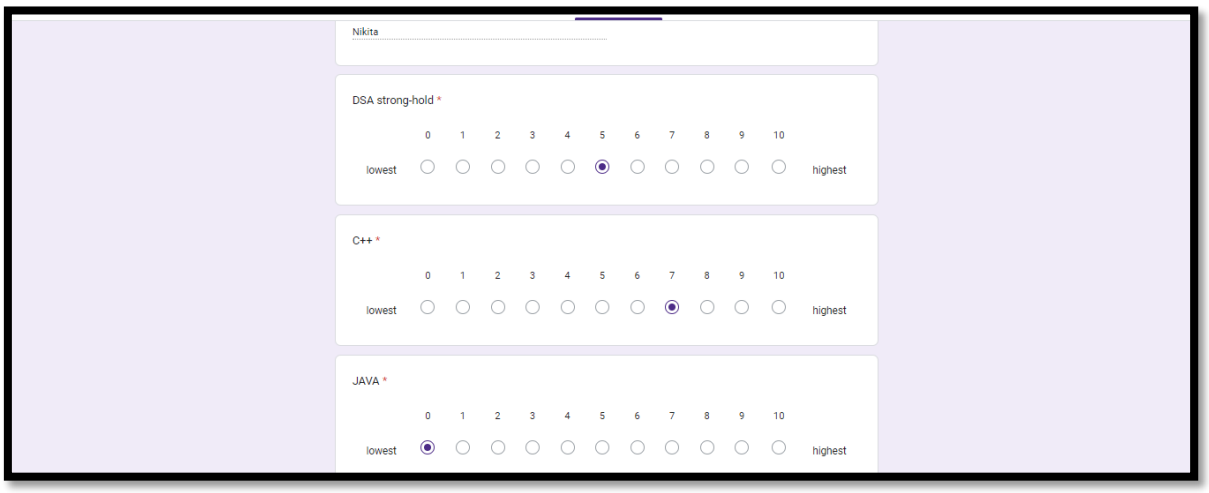


Fig 1.4

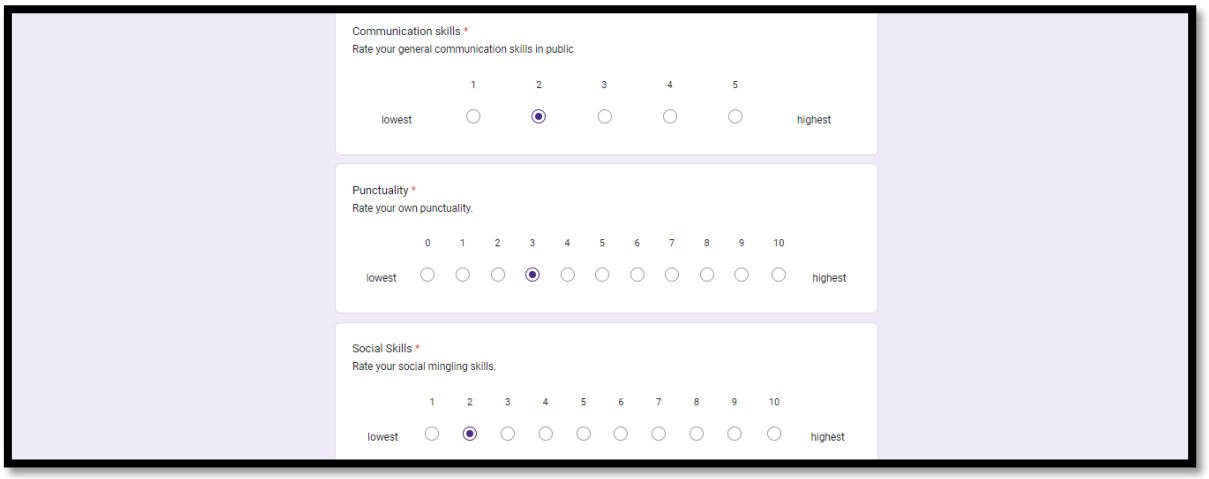


Fig 1.5

Pre-processing

Data generated was saved into excel file where various pre-processing of data is done like multiple entrees of same person was deleted.

Skill matrix

This contain all the information of experts indulged in selection process related to their skills i.e. technical as well as social skills.

Now this data generated in excel file; it is our skill matrix where various experts with their rated skills got stored. This excel file was stored in csv (comma-separated values) extension. This allow the data table to be retrieved easily into various applications and we can manipulate data in columnar format. Also, csv modules provide various classes for reading and writing the data. As we are using lists as data structure in our project which will evaluate

int data type in order to perform arithmetic operations. Since, this data is object type i.e. string so we changed their data type as int. In order to remove the NaN or null values we have selected all the question asked in google form as “required”.

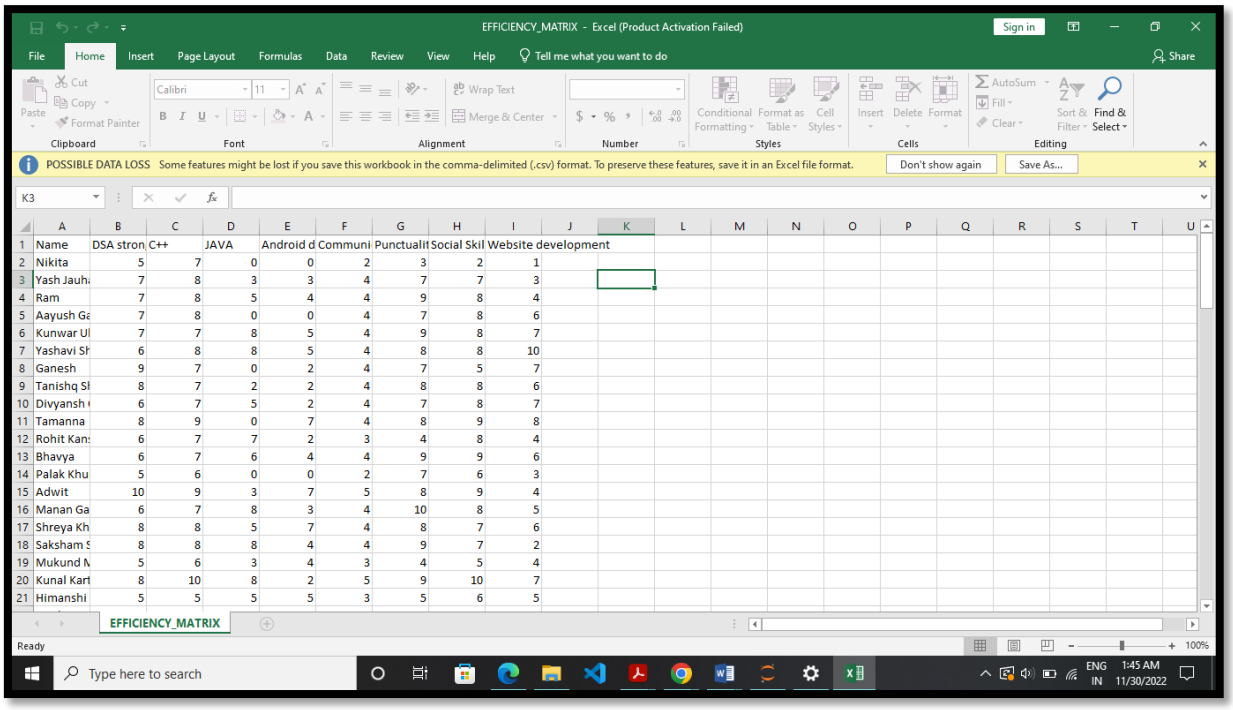


Fig 2.1

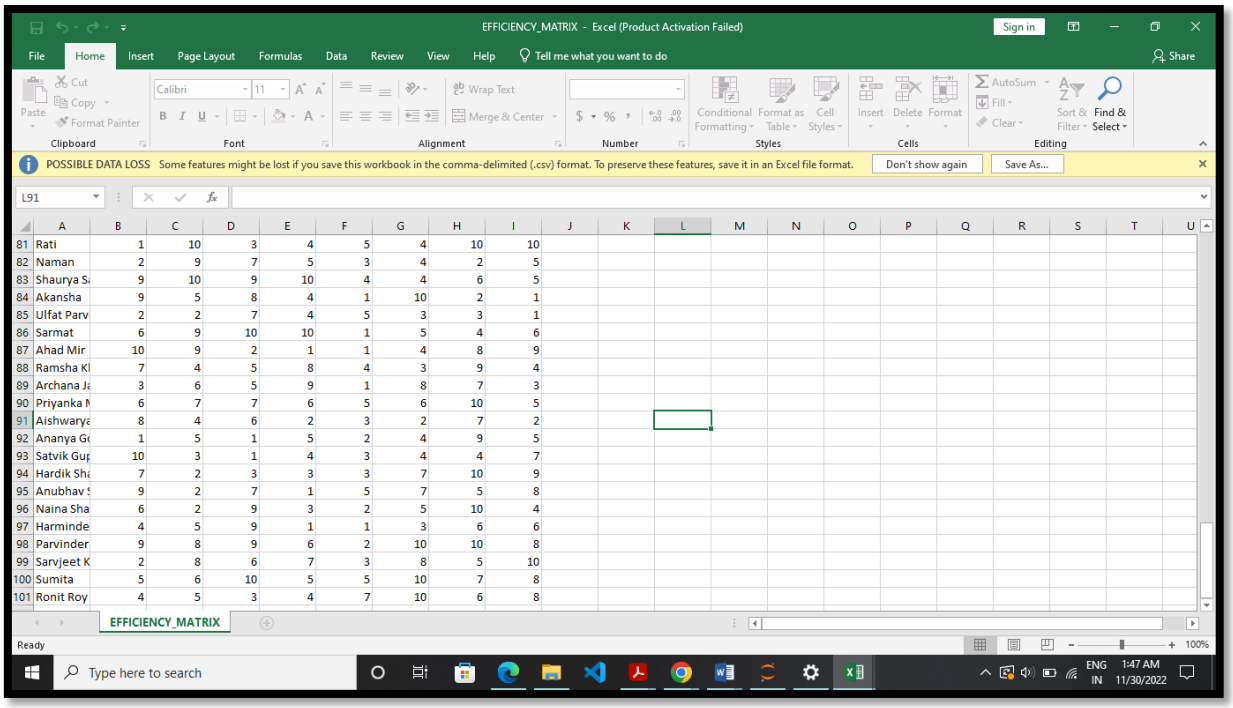


Fig 2.2

Sociometric matrix-

Sociometric matrix shows the relationship of each expert with other experts. It is NxN matrix where N is total number of experts indulged in selection process of team according to requirement matrix.

In this matrix, we can see values are between -2 to +2.

Where;

-2 experts strongly hesitate to work with particular colleague

-1 experts hesitate to work with particular colleague

0 expert is neutral work with particular colleague

+1 expert agree to work with particular colleague

+2 is strongly agree to work with particular colleague

So, this data of sociometric matrix is generated randomly between 100 experts as the data of skill matrix generated was dynamic and the particular person will not fill the value for some person who recently filled google form and this will not create NxN matrix.

So in this , again a cold start problem is solved as the freshers included in team will not be having any conflict with other colleague and vice-versa so we can create a team which will be good in social factors and have good communication in them which will automatically improve team efficiency.

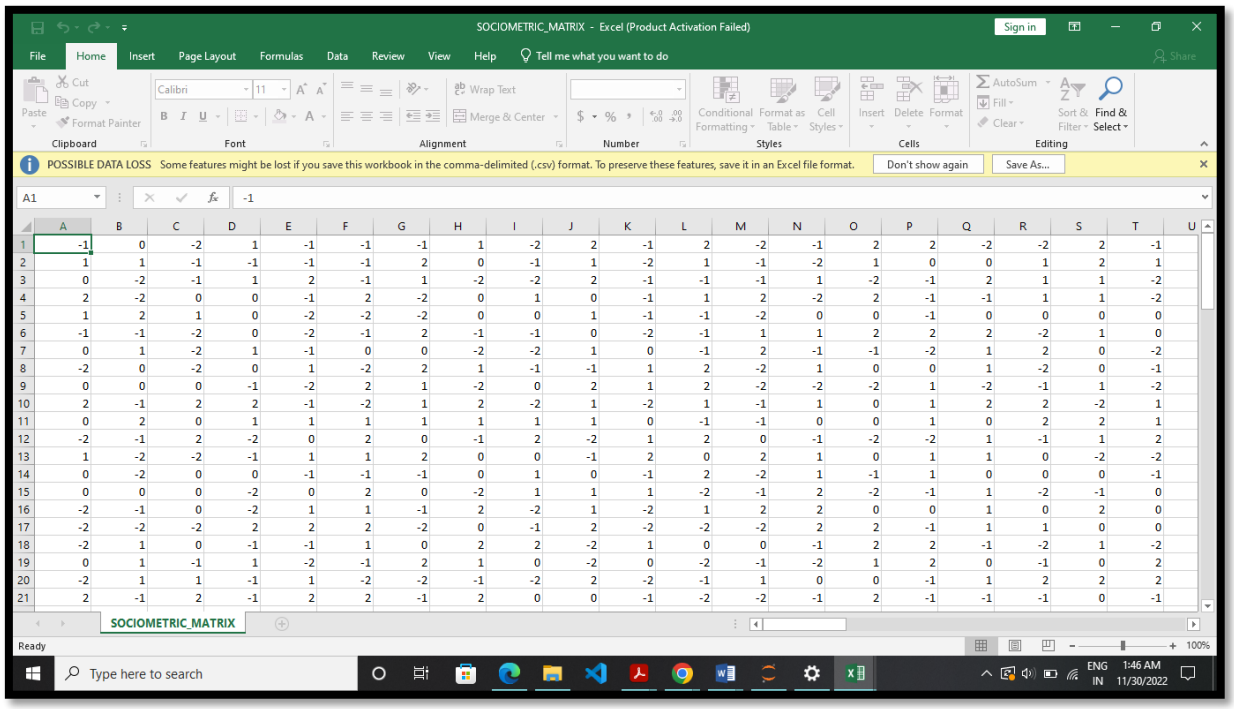


Fig 3.1

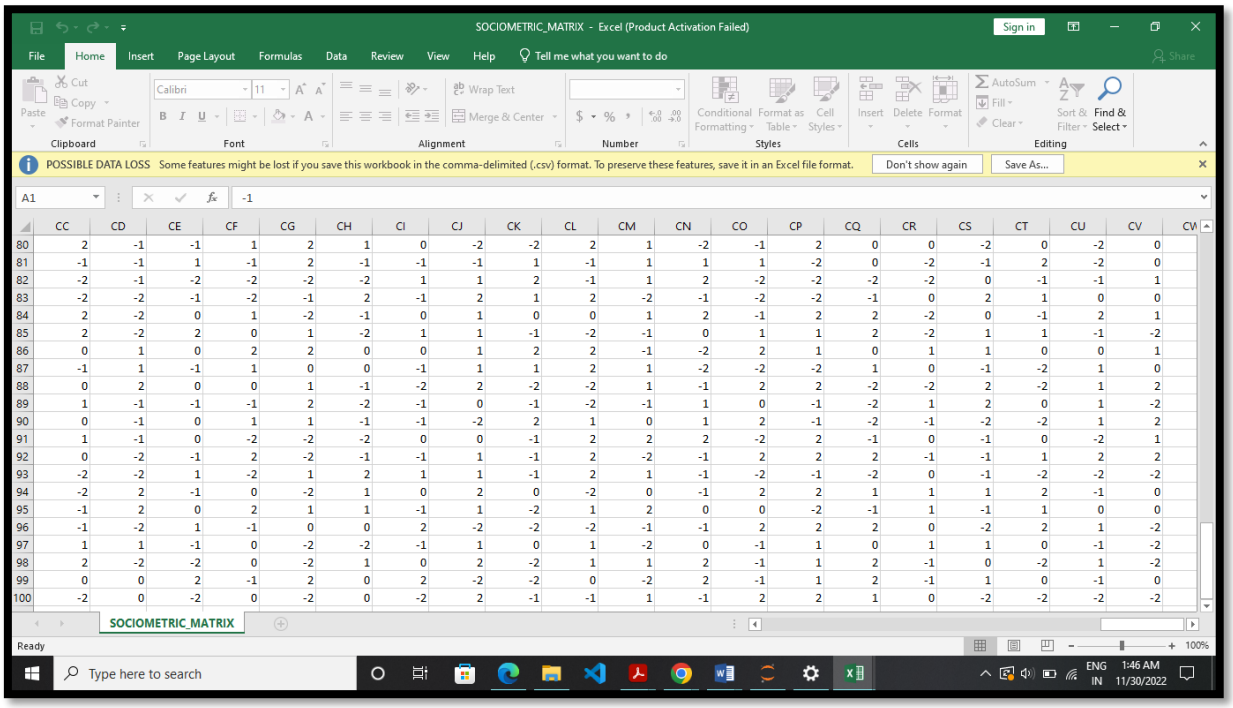


Fig 3.2

Initialization of population in GA

As we have generated our answer through genetic algorithm. So, in that algorithm initial step is generation of initial population.in this we use to randomly generate the team according to requirements generated in requirement matrix. the randomly generated team is selected from the population of experts.

It is the most important step in GA as generation of chromosome in which each gene is an expert/individual such that generated chromosome is a possible solution to problem. Also, the number of experts selected for the particular skill should be equal to the number provided in skill matrix and this should satisfy the requirements provided in matrix (Requirement Matrix)

Now, the generated chromosome or teams' values against the efficiency score and trust score Will be taken as parameter to compare with other generated population through further steps of GA.

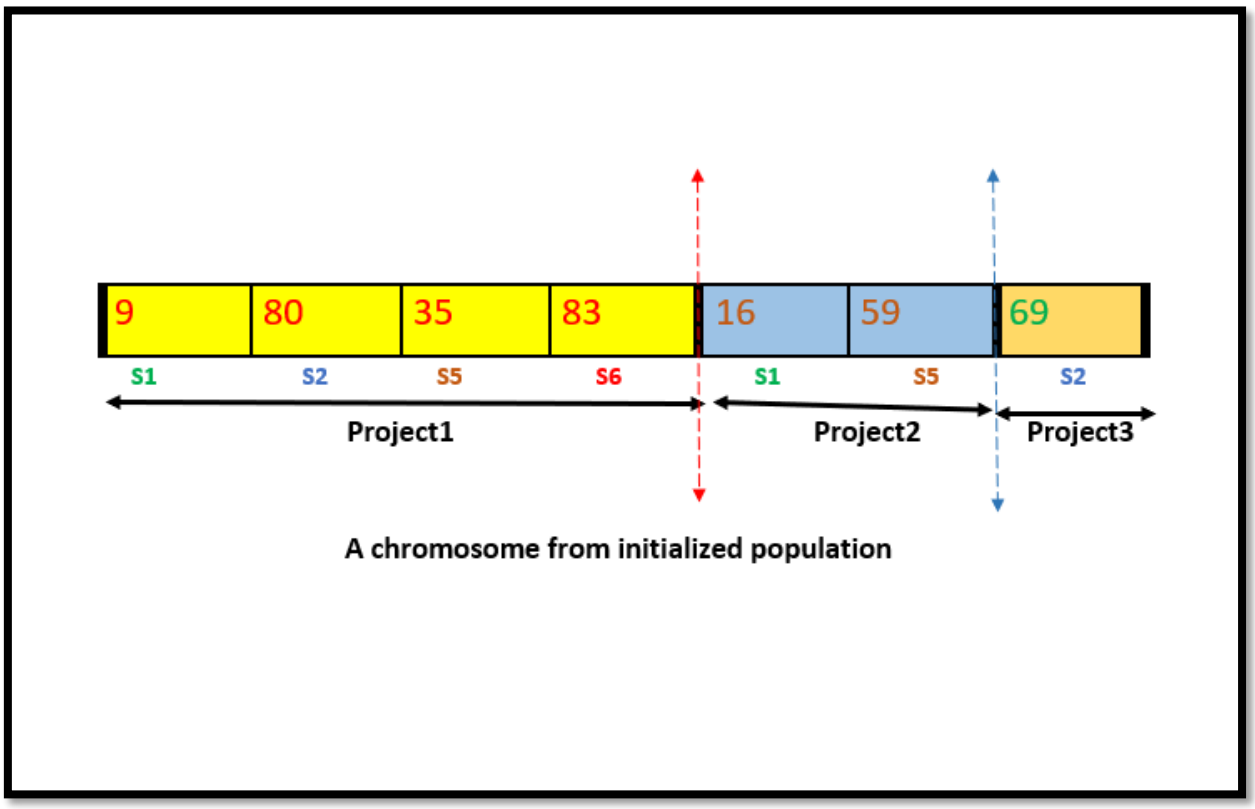


Fig 3.3

Requirement matrix

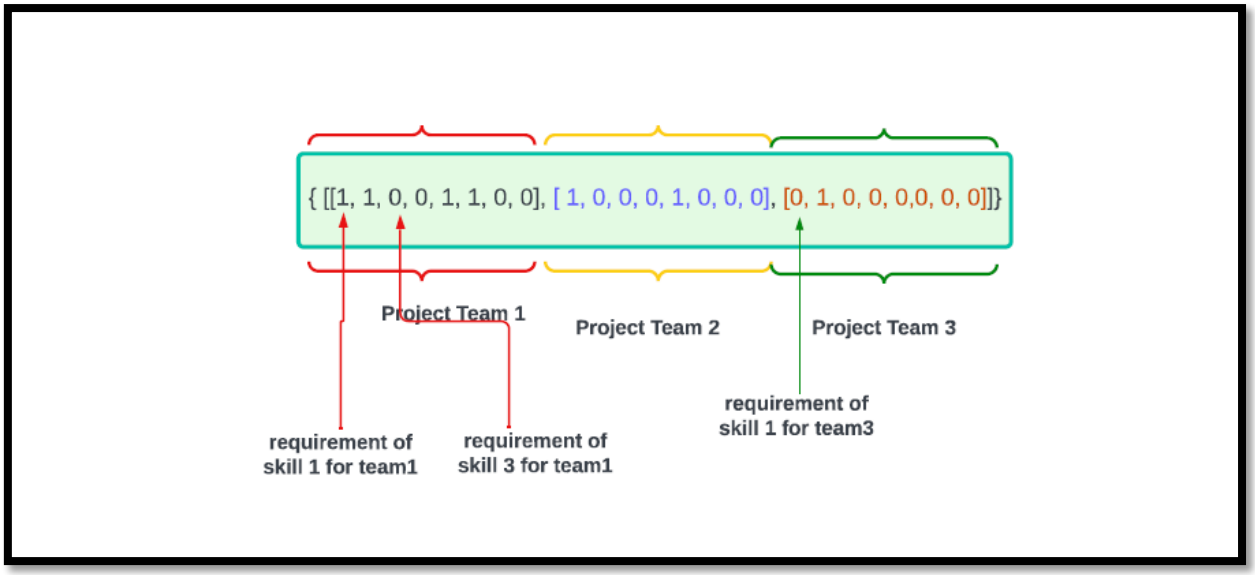


Fig 3.4

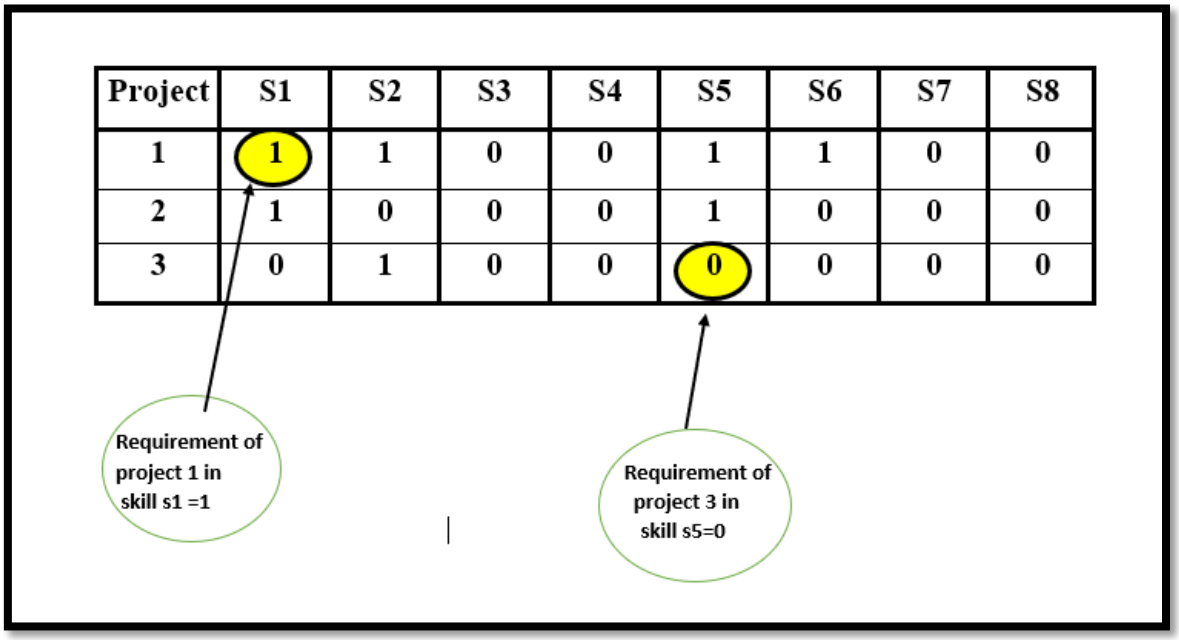


Fig 3.5

Requirement matrix provide the information regarding how many project teams is to be considered at one time in the diagram above three project teams are taken at a time.

Also, the value in this teams shows number of experts required for a particular skill.

However, social skills like communication and punctuality are added on its own to the efficiency score of the team. Also, we can ask for these skills in matrix as requirements in team. This matrix will be used in initialization of population as , the number of experts selected for the particular skill should be equal to the number provided in skill matrix and this should satisfy the requirements provided in matrix (Requirement Matrix).). And, it will be used in generation of chromosomes or teams through the process of mutation and crossover.

In crux, it represents the number of people required having particular skill for a project.

Chromosome- different task and team obtained on the basis of f1,f2

Chromosome is a structure/solution in which each gene is an expert/individual such that generated chromosome is a possible solution to problem. On the basis of fitness value best chromosome is selected for further generation of chromosome (team). Selection is done independently on the basis of function-1, function-2 namely social values, skill rating(efficiency).

We have taken the number of chromosomes as a multiple of 2, so that all the chromosomes can be chosen as parents for crossover. For mutation any of the chromosomes can be used as a parent. Suppose if y number of chromosomes are generated initially, then in the i-th generation y more child chromosomes are generated so out of $2 * y$ chromosomes top y fittest chromosomes will survive for generation (i+1).

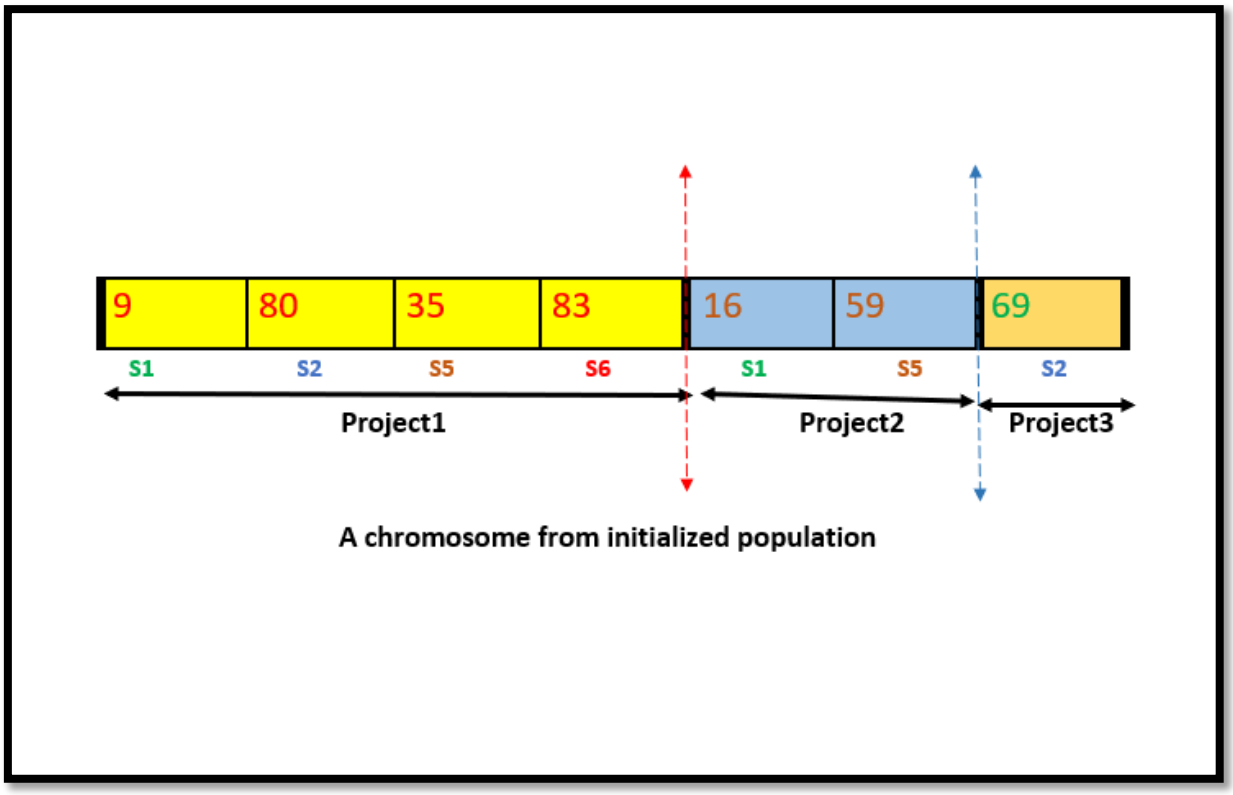


Fig 3.6

Methodology

Genetic Algorithm

The Genetic Algorithms are adaptive heuristic search algorithms that belong to the larger part of evolutionary algorithms. It is inspired by the Charles Darwin's theory of survival of the fittest. This algorithm is used to generate high quality solutions for optimization problem. Each generation consists of a population of chromosomes and each chromosome represent a solution to our problem. Those individuals who are successful (fittest) then mate to create more offspring than others.

Pseudocode

Start
Generate the Initial Population
Calculate the fitness value
Repeat
    Selection
    Crossover
    Mutation
    Calculate the fitness value
Until population has converged
Stop

The multiple team formation problem requires us to find the allocation of people/experts for multiple projects based on different project requirements such that the overall efficiency of the project is maximized. So here we have considered two things to find the efficiency of the whole team. First one the sociometric matrix which tells the relation between different experts containing the values -2, -1, 0, 1 and 2. Where,

-2 experts strongly hesitate to work with particular colleague
-1 experts hesitate to work with particular colleague
0 expert is neutral work with particular colleague
+1 expert agree to work with particular colleague
+2 is strongly agree to work with particular colleague

Then we have the skill matrix which contains the rating (0-10) of each individual in different skills. Requirement matrix tells us about what skills are required in a particular project and how many people should possess those skills. We also have considered those people who have zero or little expertise in a particular skill to be included in the project team.

We have used Genetic algorithm-based approach to solve this problem.

The very first step of G.A. is to generate an initial population. These population consists of 10 chromosomes. Each gene of the chromosome represents an individual/expert such that the chromosome is a feasible solution to our problem statement. It has been taken care that a person is not assigned to multiple teams. These teams are generated on the basis of the requirement matrix.

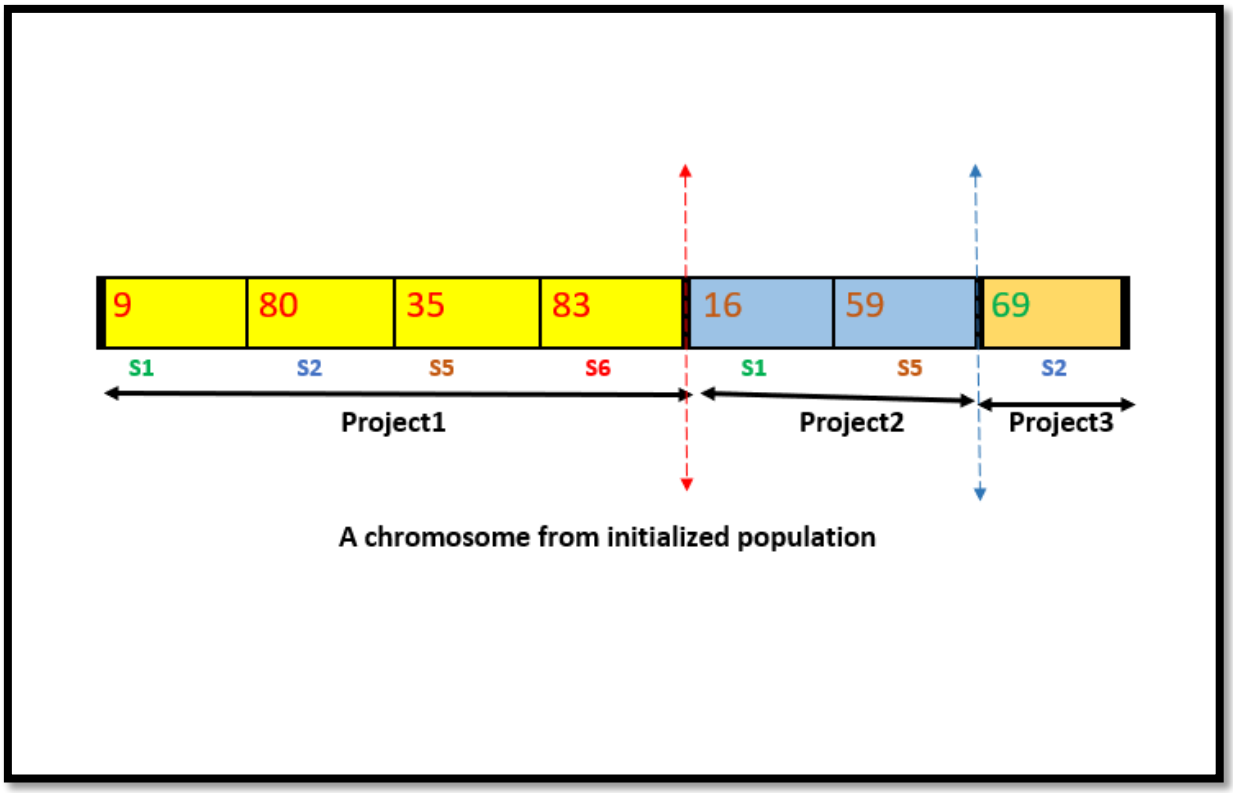


Fig 4.1
Pool of experts (1-100) S1,S2 are skills required for a particular project

We considered two functions to calculate the fitness scores of the chromosome generated. The first fitness function calculates the fitness value by adding the values of sociometric matrix corresponding to the people present in the matrix. The second function calculates the score by adding the ratings of individuals in different skills

GENETIC OPERATORS

1.Crossover- Two-point crossover is used in our proposed model. Two parents are required for generating two new children for the genetic operator. For carrying out the two points crossover first, two points x1 and x2 are selected randomly in such a way that $1 \leq x1 < x2 \leq$ CS. In the next step, all the individuals in sub array from point x1 to x2 of parent 1 and parent 2 are swapped between each other. Refer fig 2a and 2b. in our function two off springs are generated and offspring 'a' is copy of parent2 and offspring 'b' is copy of parent 1 after that the subarray between point x1, x2 in parent 1,2 is swapped in offspring a, b.

In swapping we have to take care of one thing, such that no two same values can occur in one chromosome as an individual cannot work on the two different projects at the same time. Refer to fig 1a and 1b

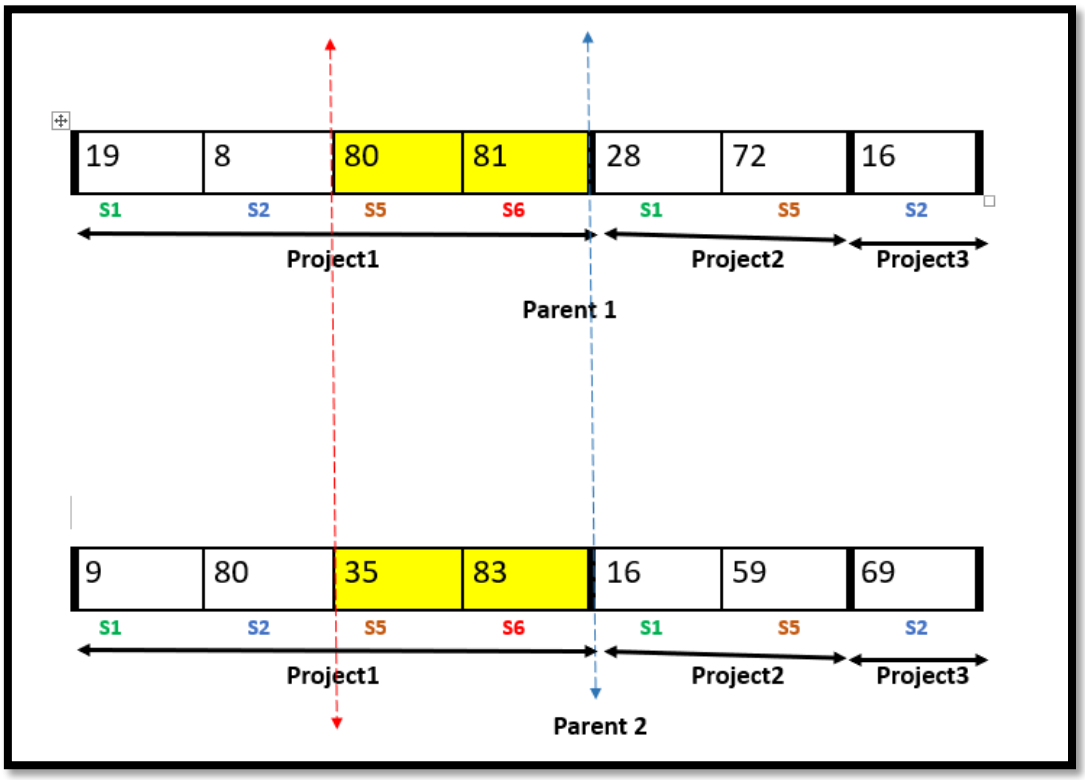


Fig 4.2

Above stated figures are Parent 1 and Parent 2 chosen for the process of crossover

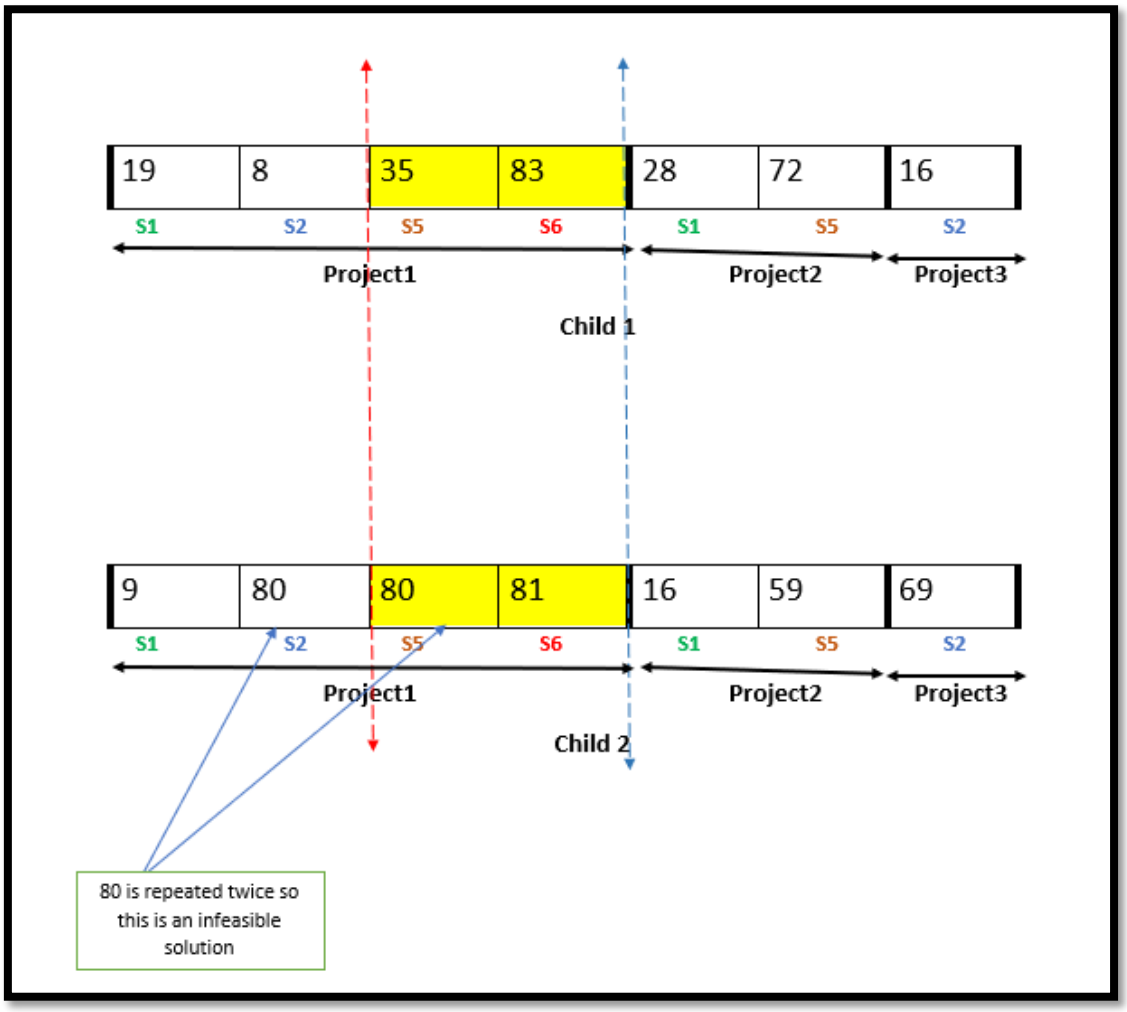


Fig 4.3

Above stated figures are Child 1 and Child 2 generated from the process of crossover

(but the same gene/expert is selected two times thus, the answer is infeasible)

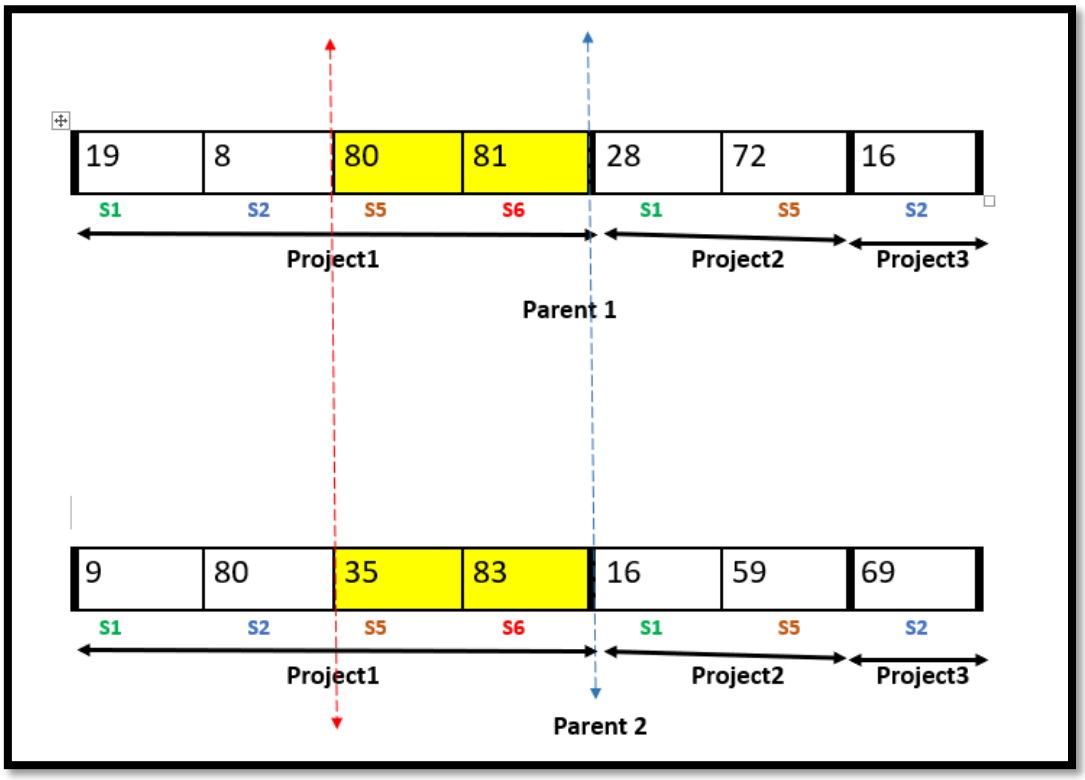


Fig 4.4

Above stated figures are Parent 1 and Parent 2 chosen for the process of crossover

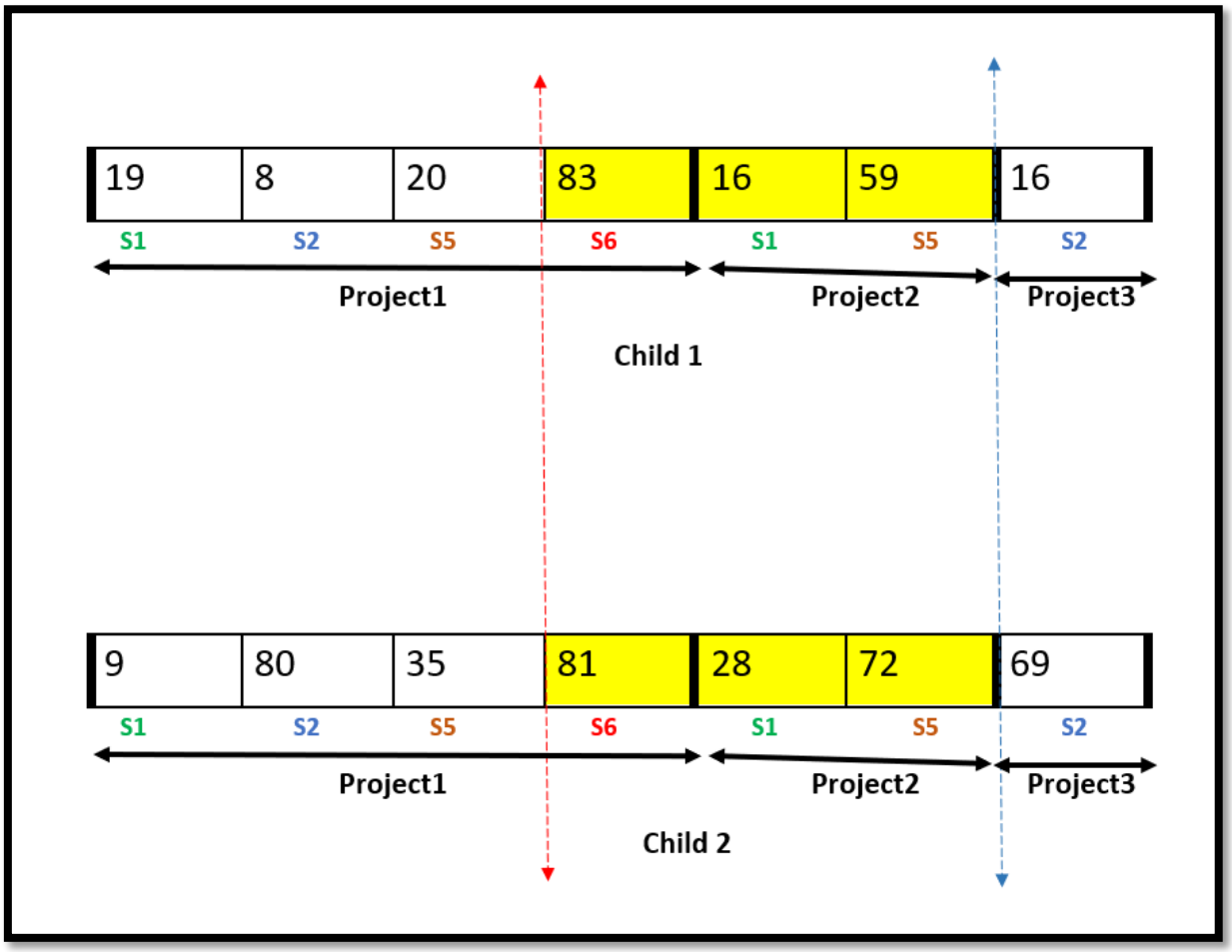


Fig 4.5

Above stated figures are Child 1 and Child 2 generated from the process of crossover

2.Mutation

Mutation is done using random resetting. Since, we have considered all person have all skills. So, in this technique, a point is selected randomly in the chromosome and the value of the individual for that point is reset from the individuals available for that skill and that which has not been taken earlier.if we reset individual 35 with 16,it will generate an infeasible solution as individual 16 gets duplicated in the same chromosome which will violate our condition.

Before Mutation

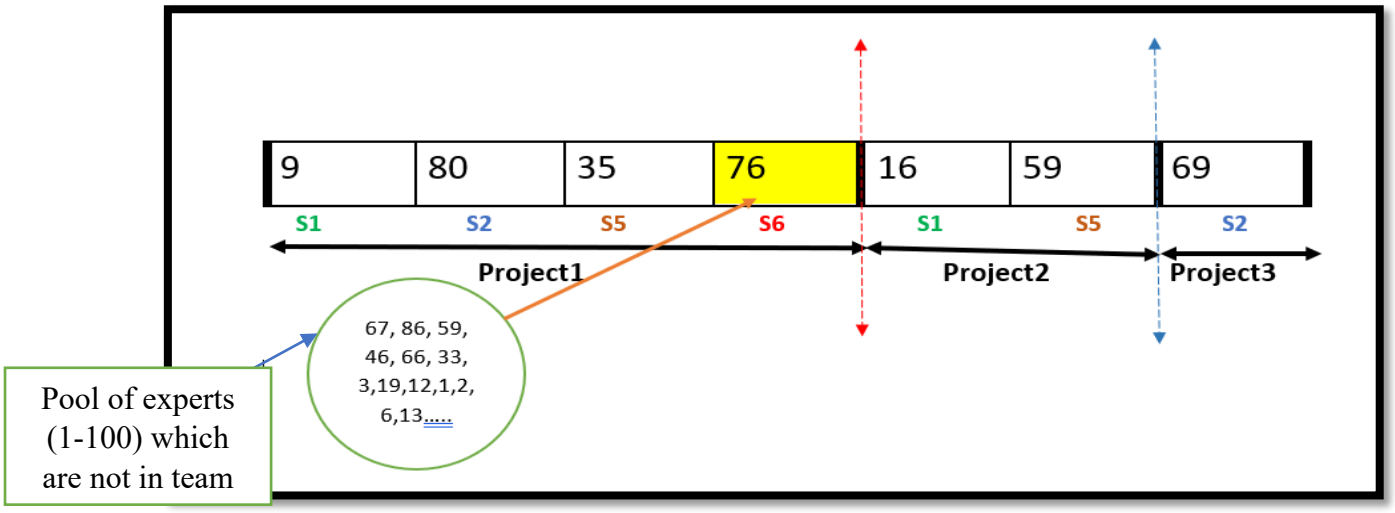


Fig 4.6

Above stated figures are parent 1 chosen for the process of mutation

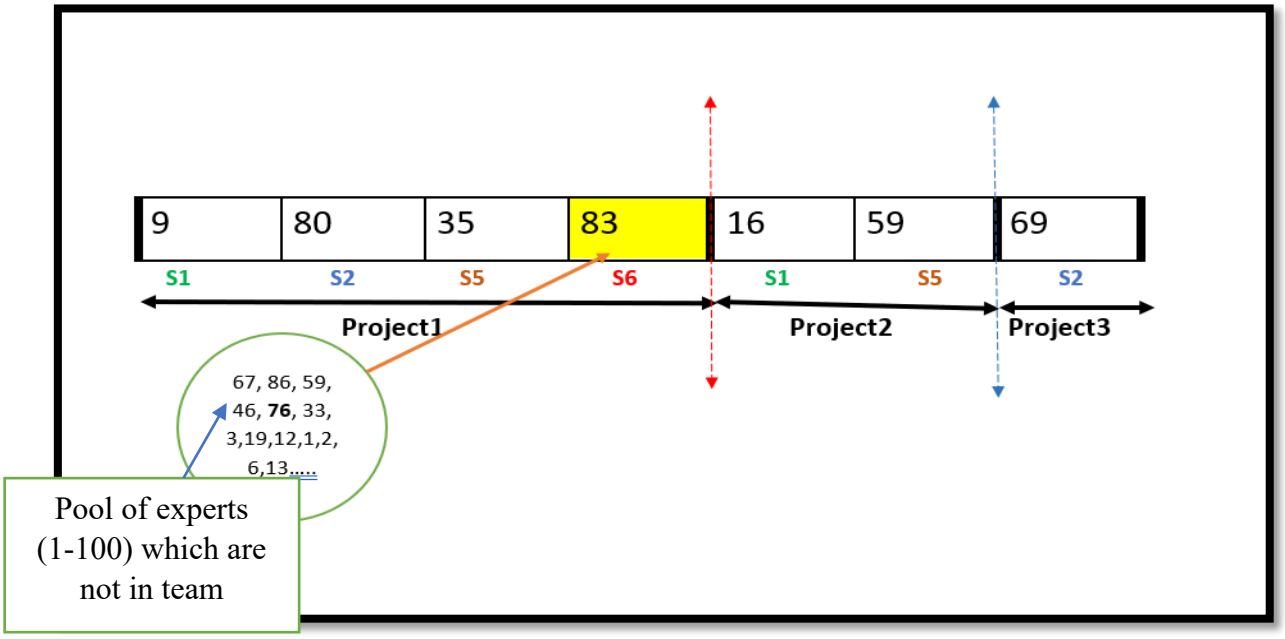


Fig 4.7

Above stated figures are Child 1 generated from the process of mutation

(where the gene/expert swapped is selected from pool of experts which are not in team)

3.Selection-

We have taken the number of chromosomes as a multiple of 2, so that all the chromosomes can be chosen as parents for crossover. For mutation any of the chromosomes can be used as a parent. Suppose if y number of chromosomes are generated initially, then in the i-th generation y more child chromosomes are generated so out of $2 * y$ chromosomes top y fittest chromosomes will survive for generation (i+1). We have used this approach of Tournament Selection in our algorithm

4.Fitness function-

The fitness function simply defined is a function which takes a **candidate solution to the problem as input and produces as output** how "fit" our how "good" the solution is with respect to the problem in consideration.

Calculation of fitness value is done repeatedly in a GA and therefore it should be sufficiently fast. A slow computation of the fitness value can adversely affect a GA and make it exceptionally slow. When the fitness value does not improve after a certain number of iterations, it means that the solution has converged to the best possible solution.

Generic Requirements of a Fitness Function

The following requirements should be satisfied by any fitness function.

- The fitness function should be clearly defined. The reader should be able to clearly understand how the fitness score is calculated.
- The fitness function should be implemented efficiently. If the fitness function becomes the bottleneck of the algorithm, then the overall efficiency of the genetic algorithm will be reduced.
- The fitness function should quantitatively measure how fit a given solution is in solving the problem.
- The fitness function should generate intuitive results. The best/worst candidates should have best/worst score values.

First fitness function (Trust Score)

It is defined as the weighted sum of efficiencies of the projects, efficiency of a project depends on the social relationship among all the individuals of a team. To calculate this we first have to look at the sociometric matrix and find the index corresponding to the two

individuals whose sociometric values needs to be taken into consideration. Maximum effectiveness can be calculated when there is positive relationship among the team members.

Trust Score+=sum of sociomat[individual1][individual 2] of all the experts in the team

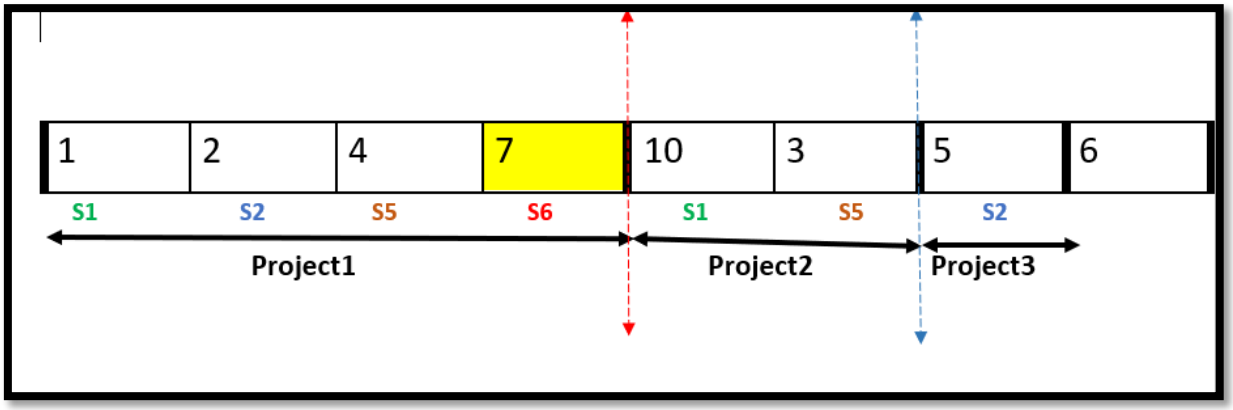


Fig 4.8

	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0	0	-2	1	-1	-1	-1	1	-2	2	-1	2	-2
2	1	0	-1	-1	-1	-1	2	0	-1	1	-2	1	-1
3	0	-2	0	1	2	-1	1	-2	-2	2	-1	-1	-1
4	2	-2	0	0	-1	2	-2	0	1	0	-1	1	2
5	1	2	1	0	0	-2	-2	0	0	1	-1	-1	-2
6	-1	-1	-2	0	-2	0	2	-1	-1	0	-2	-1	1
7	0	1	-2	1	-1	0	0	-2	-2	1	0	-1	2
8	-2	0	-2	0	1	-2	2	0	-1	-1	1	2	-2
9	0	0	0	-1	-2	2	1	-2	0	2	1	2	-2
10	2	-1	2	2	-1	-2	1	2	-2	0	-2	1	-1
11	0	2	0	1	1	1	1	1	1	1	0	-1	-1
12	-2	-1	2	-2	0	2	0	-1	2	-2	1	0	0
13	1	-2	-2	-1	1	1	2	0	0	-1	2	0	0
14	0	-2	0	0	-1	-1	-1	0	1	0	-1	2	-2
15	0	0	0	-2	0	2	0	-2	1	1	1	-2	-1
16	-2	-1	0	-2	1	1	-1	2	-2	1	-2	1	2
17	-2	-2	-2	2	2	2	-2	0	-1	2	-2	-2	-2
18	-2	1	0	-1	-1	1	0	2	2	-2	1	0	0
19	0	1	-1	1	-2	-1	2	1	0	-2	0	-2	-1
20	-2	1	1	-1	1	-2	-2	-1	-2	2	-2	-1	1
21	2	-1	2	-1	2	2	-1	2	0	0	-1	-2	-2
22	-2	-1	1	0	0	-2	-1	2	2	-2	1	-1	-1

Trust Score provided by 1 for

Trust Score provided by 2 for

Fig 4.9

Second fitness function (Skill Score)

It is defined as the weighted sum of efficiencies of the projects, efficiency of a project depends on the skills of all the individuals of a team. To calculate this, we first have to look at the skill matrix and then find the rating of the person in that particular skill. We also have to consider their non-technical skills like communication skills and punctuality as these are also important while making a project team

Skill score+=sum of skill_mat[individual1][required skill]

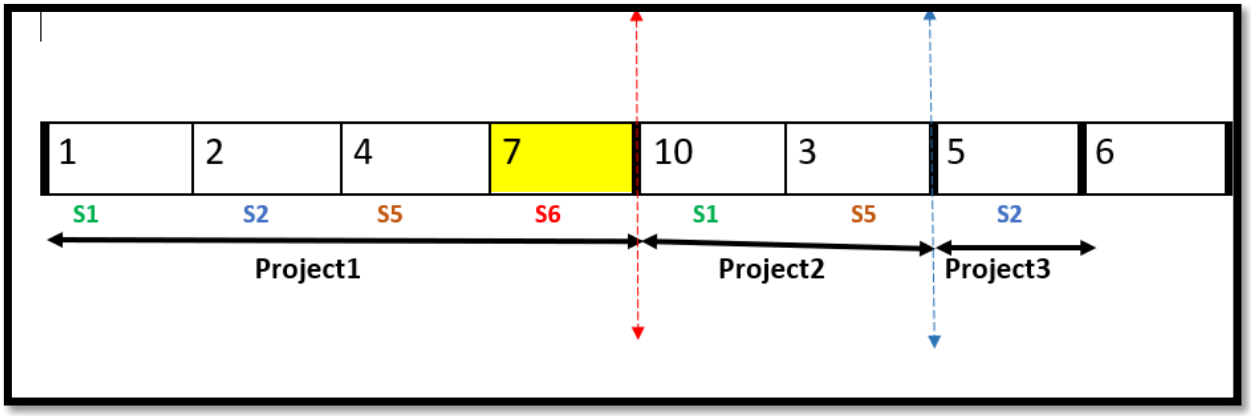


Fig 4.10

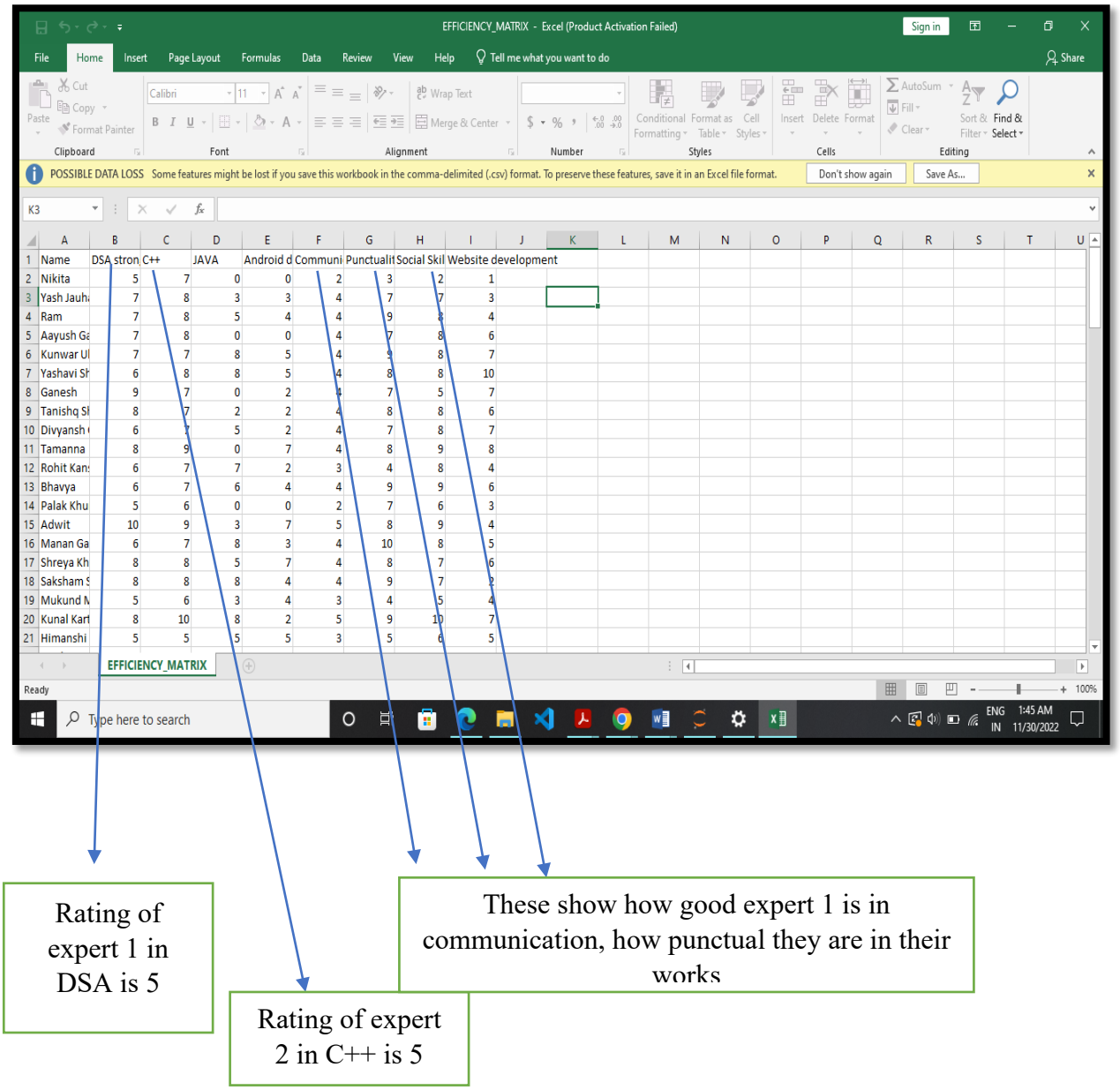

EFFICIENCY_MATRIX - Excel (Product Activation Failed)
POSSIBLE DATA LOSS Some features might be lost if you save this workbook in the comma-delimited (.csv) format. To preserve these features, save it in an Excel file format.
Don't show again
Save As...
EFFICIENCY_MATRIX
Ready
Type here to search
Rating of expert 1 in DSA is 5
Rating of expert 2 in C++ is 5
These show how good expert 1 is in communication, how punctual they are in their works

Fig 4.11

Flowchart of Genetic Algorithm

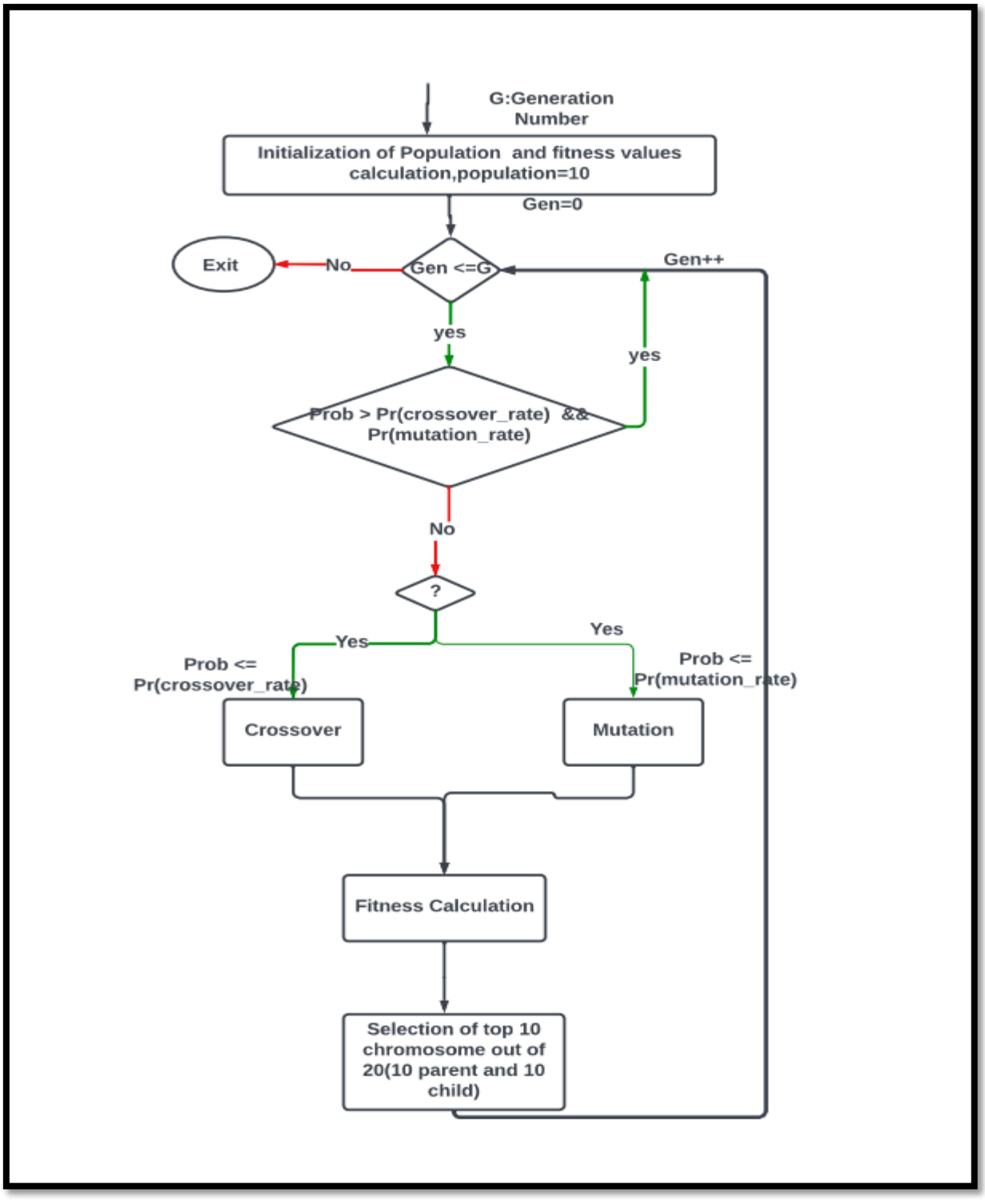


Fig 4.12

NSGA2

The Non – dominated Sorting Genetic Algorithm. It is multi-objective optimization algorithm. The maximization of objective is been performed in which no team can dominate the generated solution teams. The optimization is done on the basis of social values, skill rating(efficiency). The crossover of the two parents is performed which produces two off-springs which is further included in population; sorted on the basis of fitness values. The mutation is performed on offspring in which one of the experts is selected and replaced to form a new chromosome. In this, the pareto front generated has the team which is either not dominated in social value and skill rating(efficiency) or it is best in their independent function (social skill, skill rating(efficiency)).

Pseudocode

Start
Generate the Initial Population
Calculate the fitness value
Repeat
    Selection
    Crossover
    Mutation
    Non-dominated Sorting (generation of pareto front)
    Calculation of Crowding Distance
    Calculate the fitness value
Until population has converged
Stop

The multiple team formation problem requires us to find the allocation of people/experts for multiple projects based on different project requirements such that the overall efficiency of the project is maximized. So here we have considered two things to find the efficiency of the whole team. First one the sociometric matrix which tells the relation between different experts containing the values -2, -1,0,1 and 2.

Where;

-2 experts strongly hesitate to work with particular colleague
-1 experts hesitate to work with particular colleague
0 expert is neutral work with particular colleague
+1 expert agree to work with particular colleague

+2 is strongly agree to work with particular colleague

Then we have the skill matrix which contains the rating (0-10) of each individual in different skills. Requirement matrix tells us about what skills are required in a particular project and how many people should possess those skills. We also have considered those people who have zero or little expertise in a particular skill to be included in the project team.

We have also used Non-dominated sorting Genetic algorithm-based approach to solve this problem.

The very first step of NSGA-II is to generate an initial population. These population consists of 10 chromosomes. Each gene of the chromosome represents an individual/expert such that the chromosome is a possible solution to our problem statement. It has been taken care that a person is not assigned to multiple teams. These teams are generated on the basis of the requirement matrix.

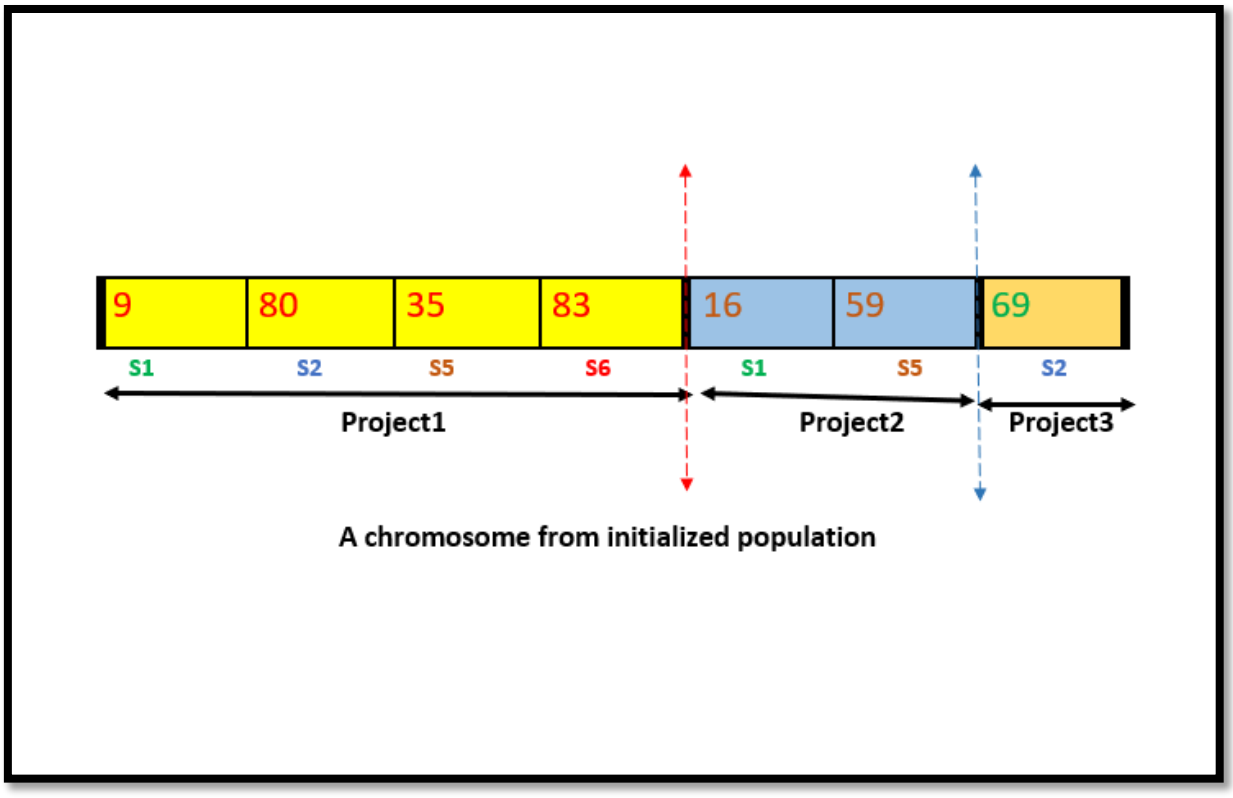


Fig 5.1

S1, S2 are skills required for a particular project

We considered two functions to calculate the fitness scores of the chromosome generated. The first fitness function calculates the fitness value by adding the values of sociometric matrix

corresponding to the people present in the matrix. The second function calculates the score by adding the ratings of individuals in different skills

GENETIC OPERATORS:

1.Crossover- Two-point crossover is used in our proposed model. Two parents are required for generating two new children for the genetic operator. For carrying out the two points crossover first, two points x1 and x2 are selected randomly in such a way that $1 \leq x1 < x2 \leq$ CS. In the next step, all the individuals in sub array from point x1 to x2 of parent 1 and parent 2 are swapped between each other. Refer fig 2a and 2b. in our function two off springs are generated and offspring 'a' is copy of parent2 and offspring 'b' is copy of parent 1 after that the subarray between point x1, x2 in parent 1,2 is swapped in offspring a, b.

In swapping we have to take care of one thing, such that no two same values can occur in one chromosome as an individual cannot work on the two different projects at the same time. Refer to fig 1a and 1b

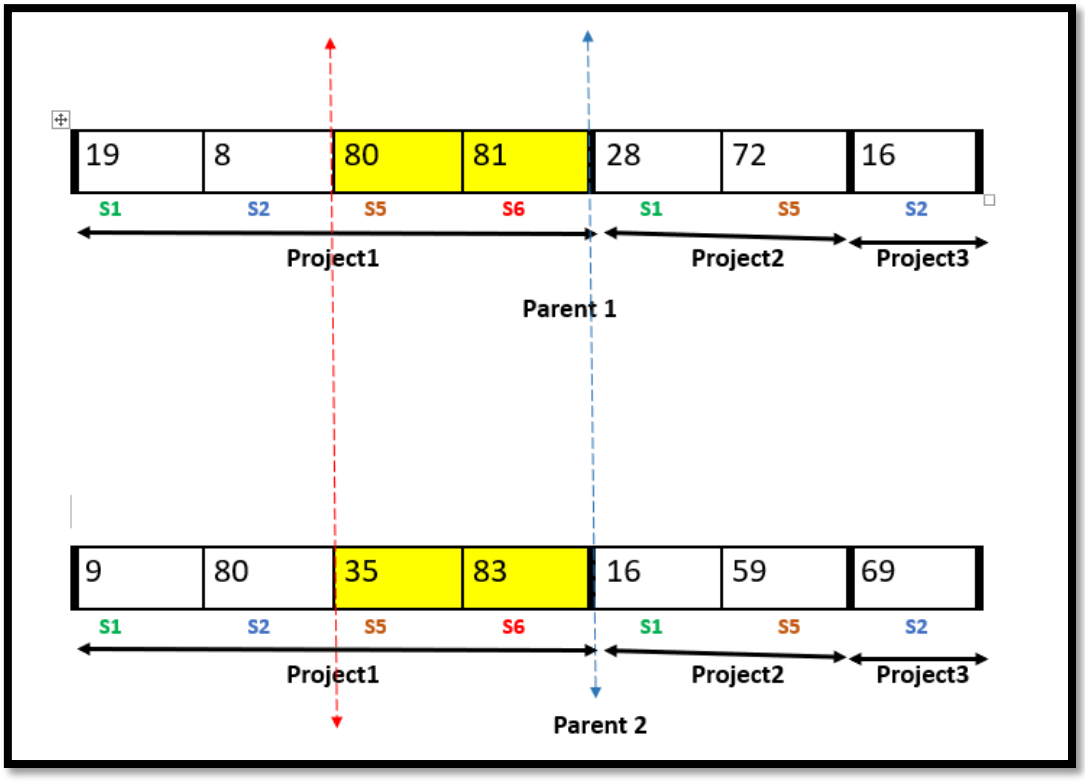


Fig 5.2

Above stated figures are Parent 1 and Parent 2 chosen for the process of crossover

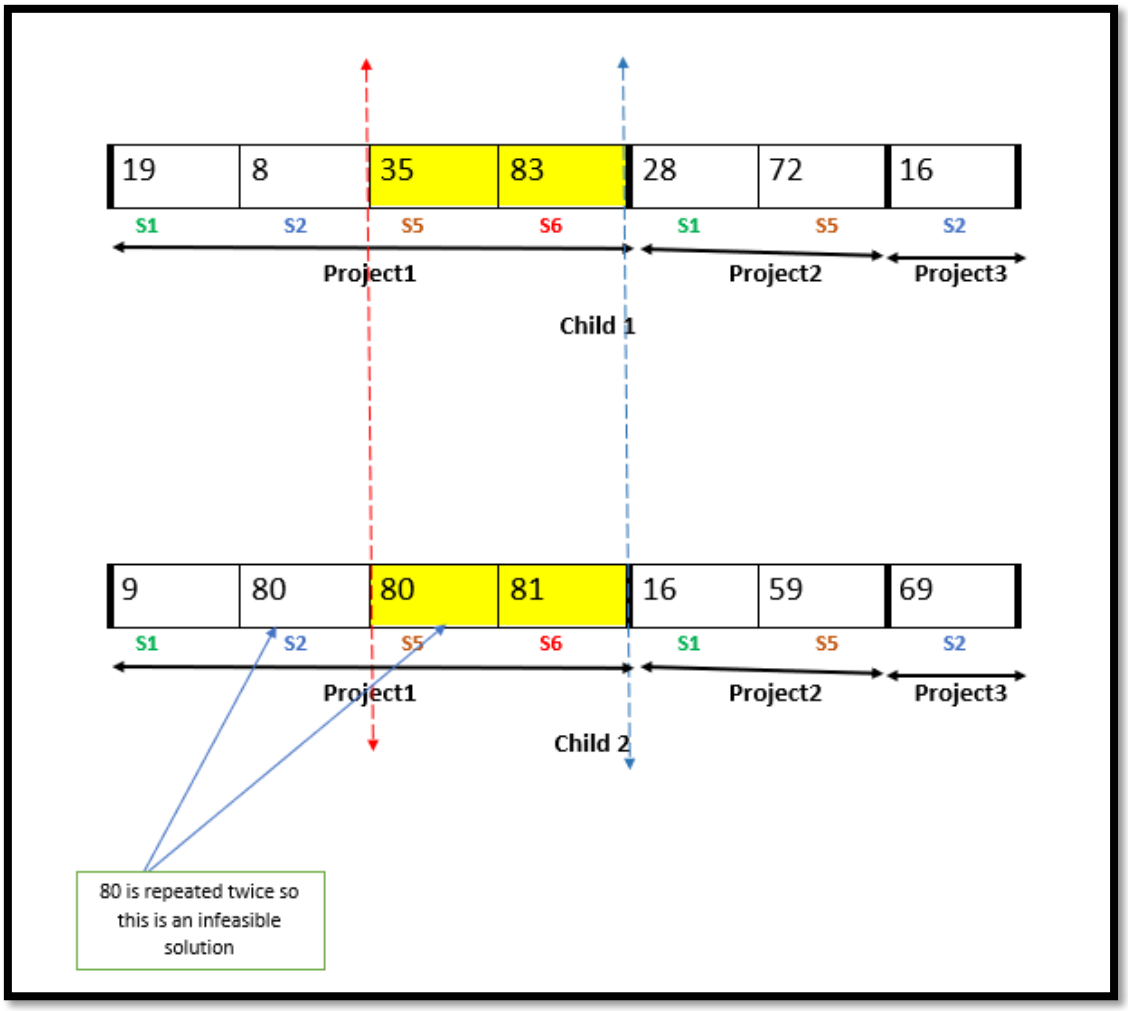


Fig 5.3

Above stated figures are Child 1 and Child 2 generated from the process of crossover

(but the same gene/expert is selected two times thus, the answer is infeasible)

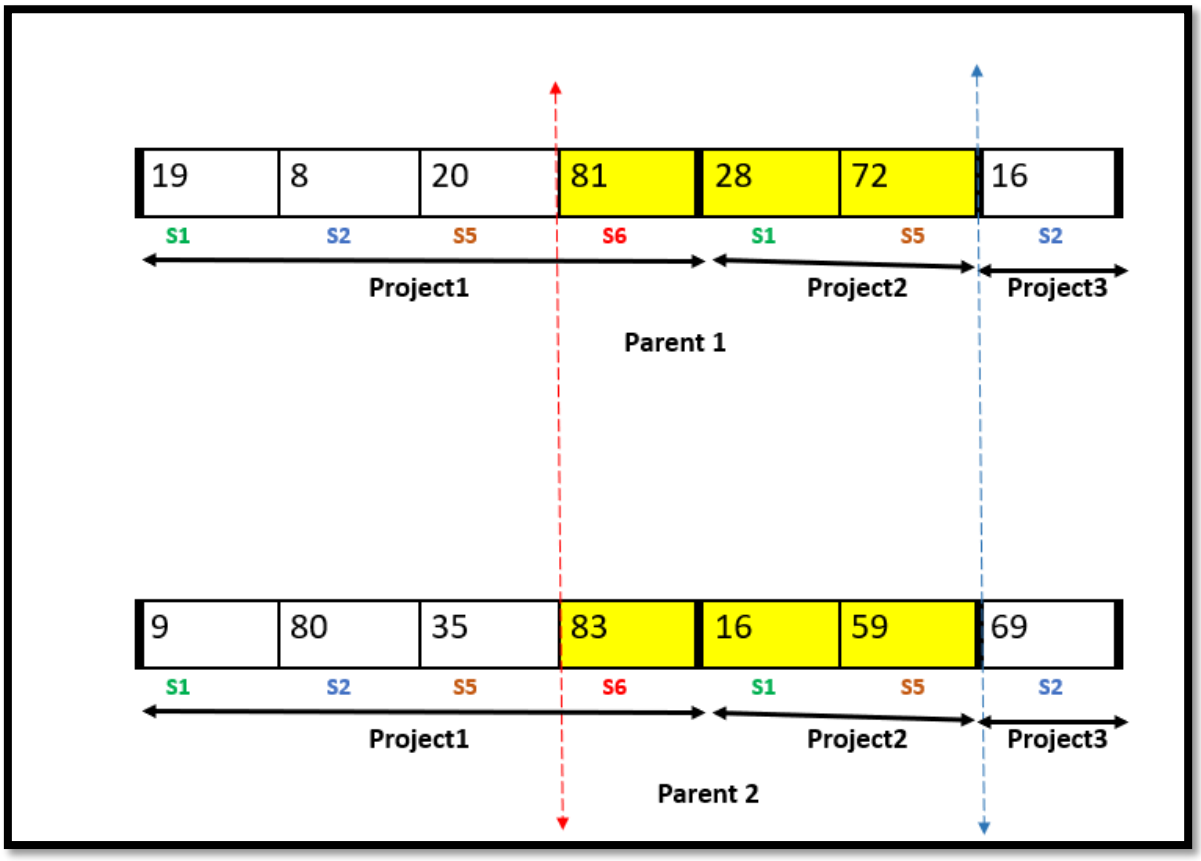


Fig 5.4

Above stated figures are Parent 1 and Parent 2 chosen for the process of crossover

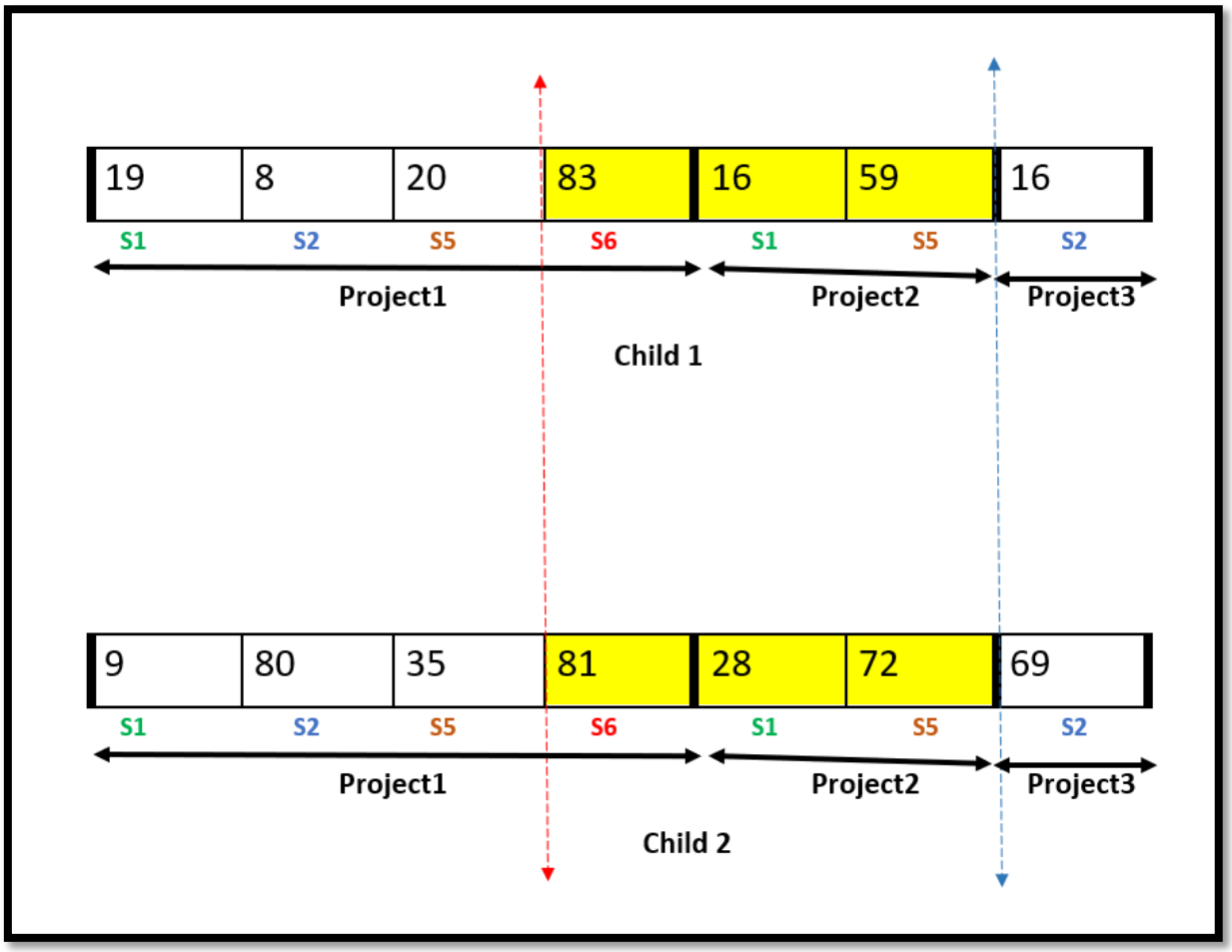


Fig 5.5

Above stated figures are Child 1 and Child 2 generated from the process of crossover

2.Mutation

Mutation is done using random resetting. Since, we have considered all person have all skills. So, in this technique, a point is selected randomly in the chromosome and the value of the individual for that point is exchanged from the individuals available for that skill and one, which had not been taken earlier in the teams. If we reset individual 35 with 16, it will generate an infeasible solution as individual 16 gets duplicated in the same chromosome which will violate our condition.

Before Mutation

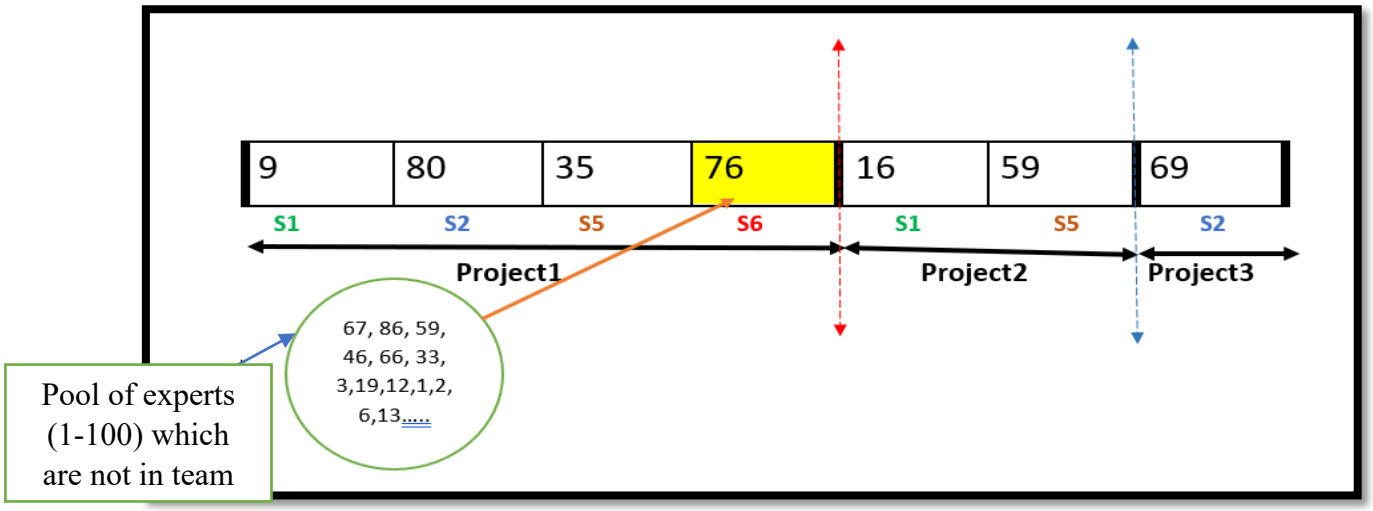


Fig 5.6

Above stated figures are parent 1 chosen for the process of mutation

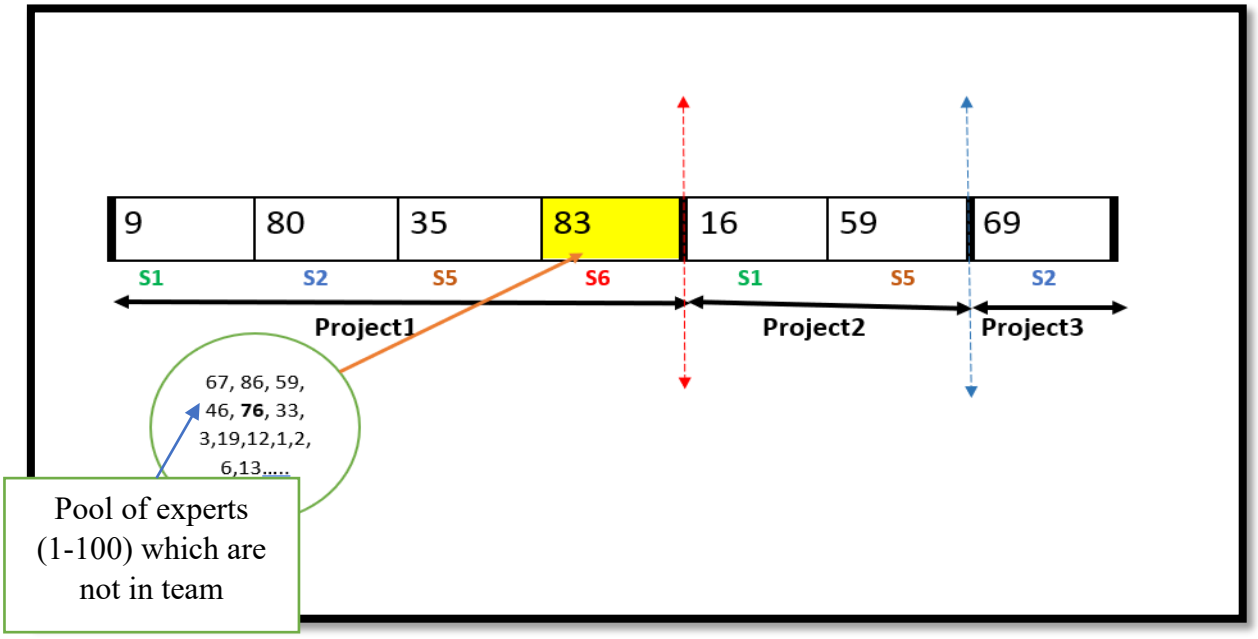


Fig 5.7

Above stated figures are Child 1 generated from the process of mutation

(where the gene/expert swapped is selected from pool of experts which are not in team)

3.Selection-

We have taken the number of chromosomes as a multiple of 2, so that all the chromosomes can be chosen as parents for crossover. For mutation any of the chromosomes can be used as a parent. Suppose if y number of chromosomes are generated initially, then in the i-th generation y more child chromosomes are generated so out of $2 * y$ chromosomes top y fittest chromosomes will survive for generation (i+1). We have used this approach of Tournament Selection in our algorithm

4.Fitness function-

The fitness function simply defined is a function which takes a **candidate solution to the problem as input and produces as output** how "fit" our how "good" the solution is with respect to the problem in consideration.

Calculation of fitness value is done repeatedly in a NSGA-II and therefore it should be sufficiently fast. A slow computation of the fitness value can adversely affect a NSGA-II and make it exceptionally slow. When the fitness value does not improve after a certain number of iterations, it means that the solution has converged to the best possible solution.

Generic Requirements of a Fitness Function

The following requirements should be satisfied by any fitness function.

- The fitness function should be clearly defined. The reader should be able to clearly understand how the fitness score is calculated.
- The fitness function should be implemented efficiently. If the fitness function becomes the bottleneck of the algorithm, then the overall efficiency of the genetic algorithm will be reduced.
- The fitness function should quantitatively measure how fit a given solution is in solving the problem.
- The fitness function should generate intuitive results. The best/worst candidates should have best/worst score values.

First fitness function (Trust Score)

It is defined as the weighted sum of efficiencies of the projects, efficiency of a project depends on the social relationship among all the individuals of a team.To calculate this we first have to look at the sociometric matrix and find the index corresponding to the two individuals whose sociometric values needs to be taken into consideration. Maximum effectiveness can be calculated when there is positive relationship among the team members.

Trust Score+=sum of sociomat[individual1][individual 2] of all the experts in the team

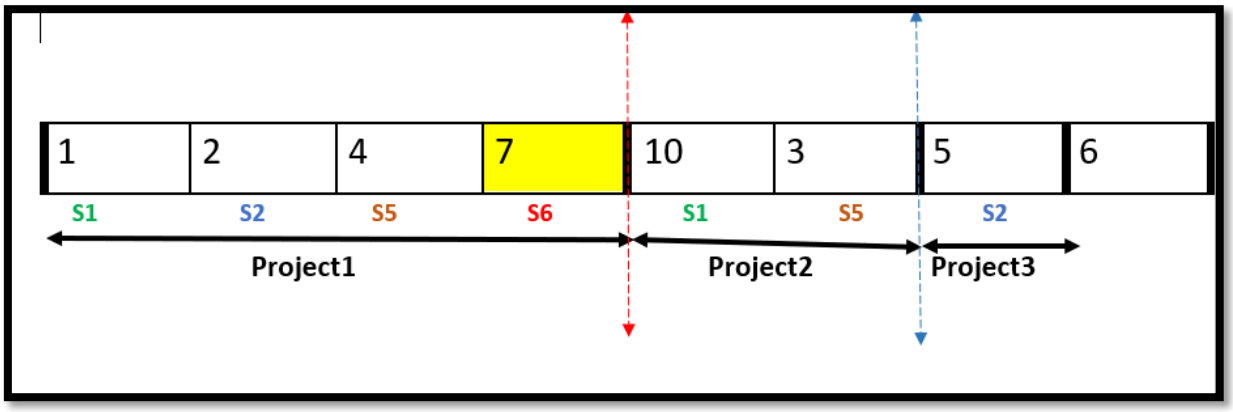
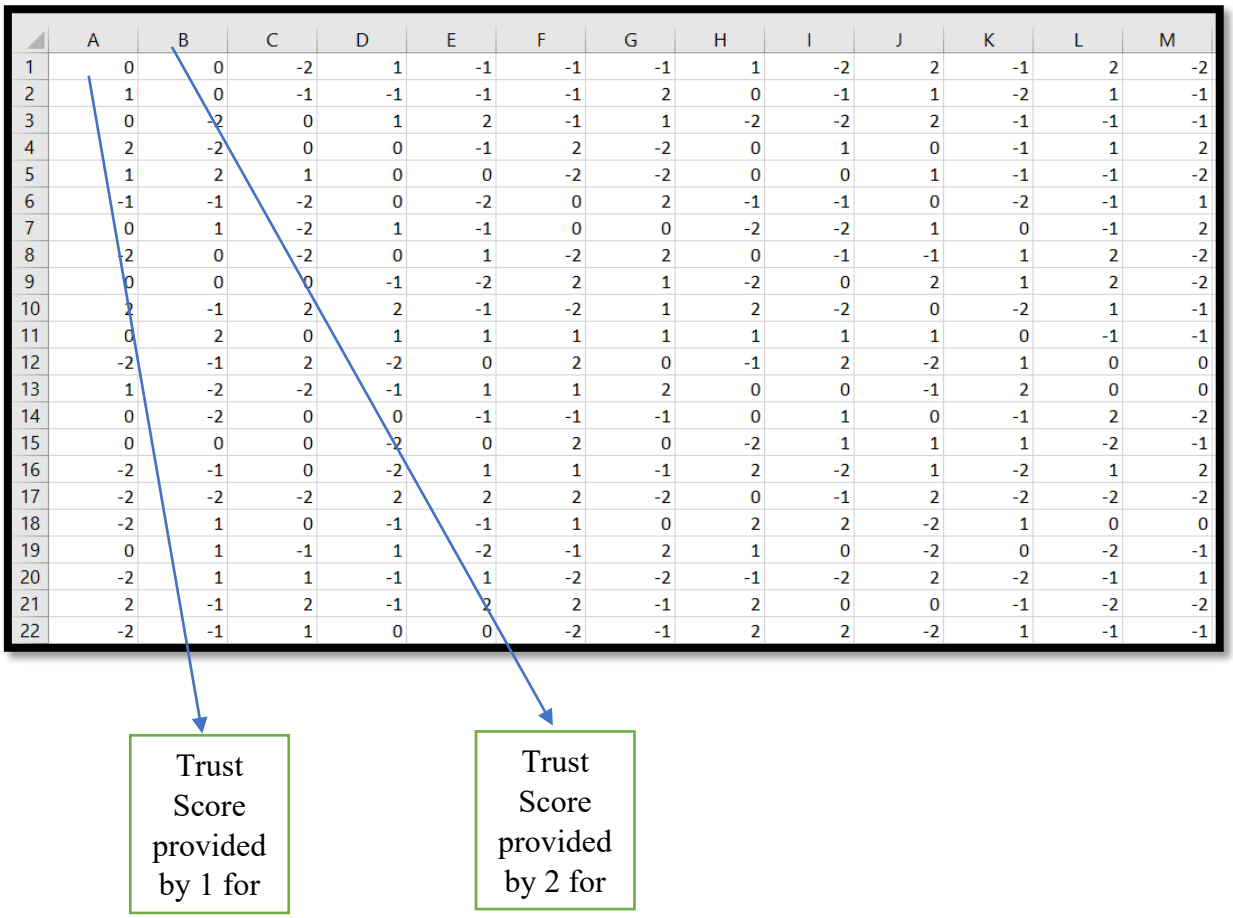


Fig 5.8



	A	B	C	D	E	F	G	H	I	J	K	L	M
1	0	0	-2	1	-1	-1	-1	1	-2	2	-1	2	-2
2	1	0	-1	-1	-1	-1	2	0	-1	1	-2	1	-1
3	0	-2	0	1	2	-1	1	-2	-2	2	-1	-1	-1
4	2	-2	0	0	-1	2	-2	0	1	0	-1	1	2
5	1	2	1	0	0	-2	-2	0	0	1	-1	-1	-2
6	-1	-1	-2	0	-2	0	2	-1	-1	0	-2	-1	1
7	0	1	-2	1	-1	0	0	-2	-2	1	0	-1	2
8	-2	0	-2	0	1	-2	2	0	-1	-1	1	2	-2
9	0	0	0	-1	-2	2	1	-2	0	2	1	2	-2
10	2	-1	2	2	-1	-2	1	2	-2	0	-2	1	-1
11	0	2	0	1	1	1	1	1	1	1	0	-1	-1
12	-2	-1	2	-2	0	2	0	-1	2	-2	1	0	0
13	1	-2	-2	-1	1	1	2	0	0	-1	2	0	0
14	0	-2	0	0	-1	-1	-1	0	1	0	-1	2	-2
15	0	0	0	-2	0	2	0	-2	1	1	1	-2	-1
16	-2	-1	0	-2	1	1	-1	2	-2	1	-2	1	2
17	-2	-2	-2	2	2	2	-2	0	-1	2	-2	-2	-2
18	-2	1	0	-1	-1	1	0	2	2	-2	1	0	0
19	0	1	-1	1	-2	-1	2	1	0	-2	0	-2	-1
20	-2	1	1	-1	1	-2	-2	-1	-2	2	-2	-1	1
21	2	-1	2	-1	2	2	-1	2	0	0	-1	-2	-2
22	-2	-1	1	0	0	-2	-1	2	2	-2	1	-1	-1

Fig 5.9

Second fitness function (Skill Score)

It is defined as the weighted sum of efficiencies of the projects, efficiency of a project depends on the skills of all the individuals of a team. To calculate this, we first have to look at the skill matrix and then find the rating of the person in that particular skill. We also have to consider their non-technical skills like communication skills and punctuality as these are also important while making a project team

Skill score+=sum of skill_mat[individual1] [required skill]

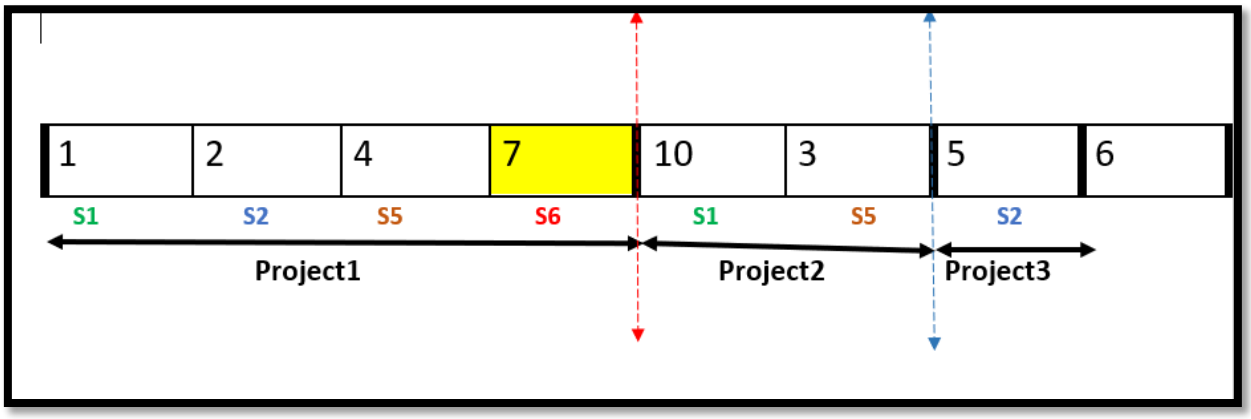
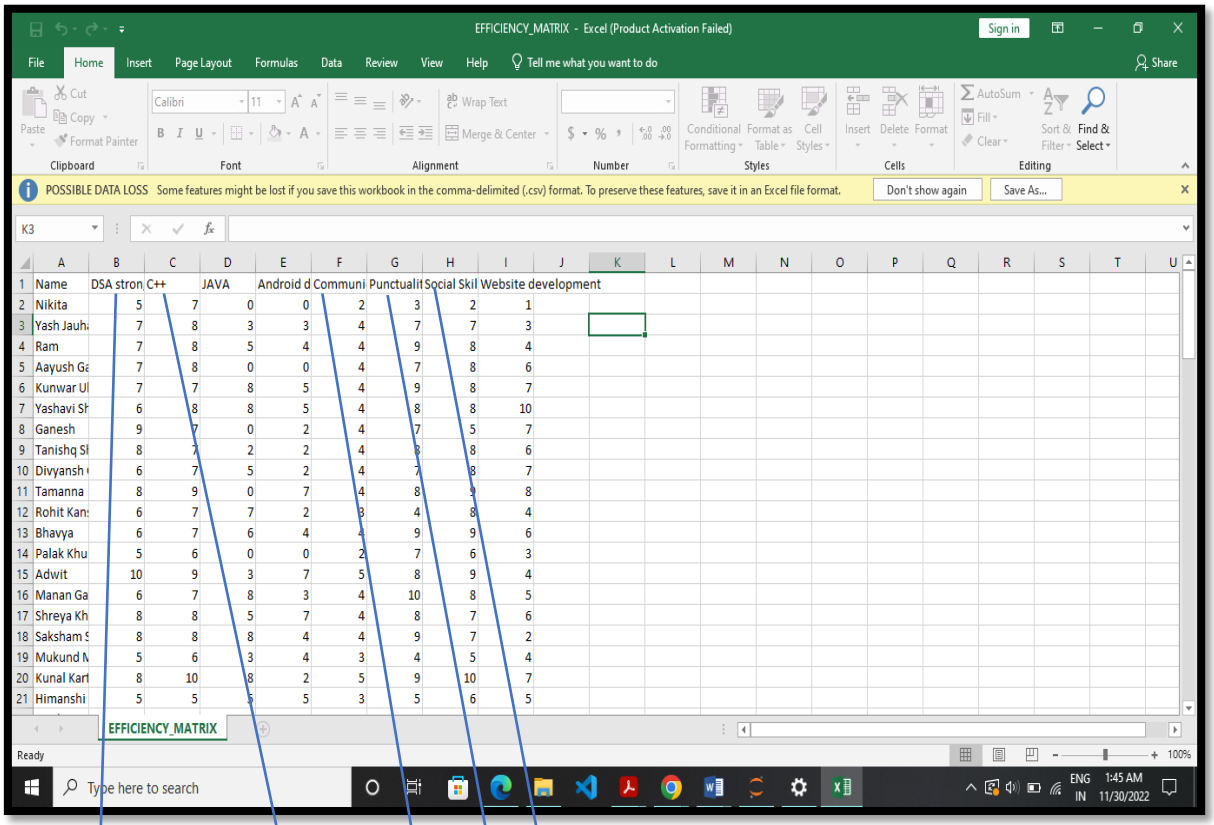


Fig 5.10



Rating of expert 1 in DSA is 5

These show how good expert 1 is in communication, how punctual they are in their works

Rating of expert 2 in C++ is 5

Fig 5.11

Non-dominated sorting

In this our target is to maximize the team ability on the basis of both the function which computes skill rating(efficiency) and social skills (or trust score).i.e. to find out which individual belongs to which front. In this, the pareto front generated has the team which is either not dominated in social value and skill rating(efficiency) or it is best in their independent function (social skill(trust score), skill rating(efficiency)) and likewise 2nd pareto front generated have those experts who are dominated by people in first pareto front. And dominates means values of one is better than the other on the basis of condition

if (values1[p] > values1[q] and values2[p] > values2[q]) or (values1[p] >= values1[q] and values2[p] > values2[q]) or (values1[p] > values1[q] and values2[p] >= values2[q])

i.e. if one expert is greater than other in value1, value2 or both then that expert is dominating the other 1 otherwise it is getting dominated and we increase the count of both dominating or being dominated and when being dominated is zero then the person is best in values or skills and will be introduced in pareto front 1.

Also, if the expert is dominating in one value but get dominated in other value then nothing is to be done with that expert.

Here, value1, value 2 are social skill and efficiency skills.

Crowding Distance:

Now, after taking best fronts to the parent generation of the next population, one of the fronts won't fit probably 100% into the next population which need to have same size as the population before. Now, to keep a good spread in NSGA-II and to avoid local maxima or minima instead of finding the global ones the crowding distance factor decides about which individuals are added to the new population, the individuals with the higher crowding distance are picked first the total crowding distance for each individual is the sum of all distances for every single target indicator. We have two target indicators the front is assorted regarding every single target indicator so one time it is sorted by f1 and in other time sorted by f2 and for every sorting iteration the individual with smallest and bigest value for target indicator get very high distance value and the distance of the other is calculated by following:

$$distance(i) = distance(i) + \frac{o(i+1) - o(i-1)}{o(\max) - 0(\min)}$$

For the point i we take distance smaller and for the next bigger value in the front and build a delta and then it is divided by the delta of the maximum value and the minimum value for target indicator then added to the already calculated distance for the other objectives, the result is that for the given front in this case F3 all individuals obtain a crowding distance value to now choose which individuals are going to the parent population and the ones with the highest crowding distance factors are chosen and passed to the next population.

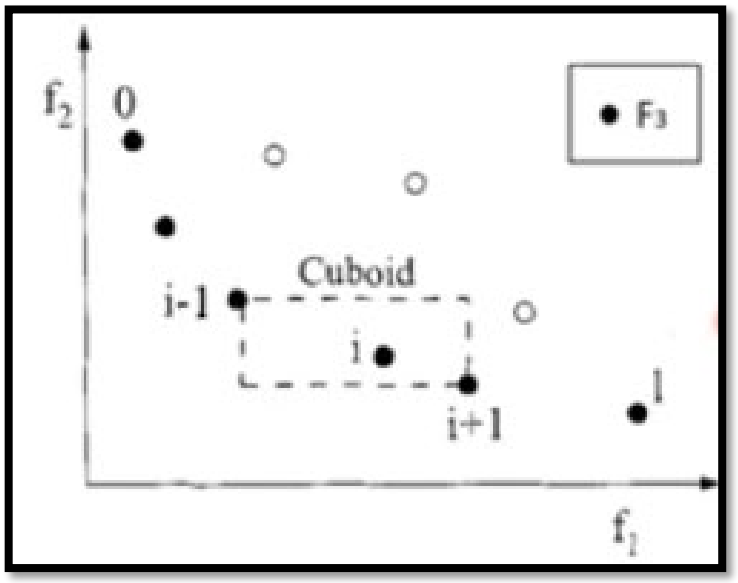


Fig 5.12

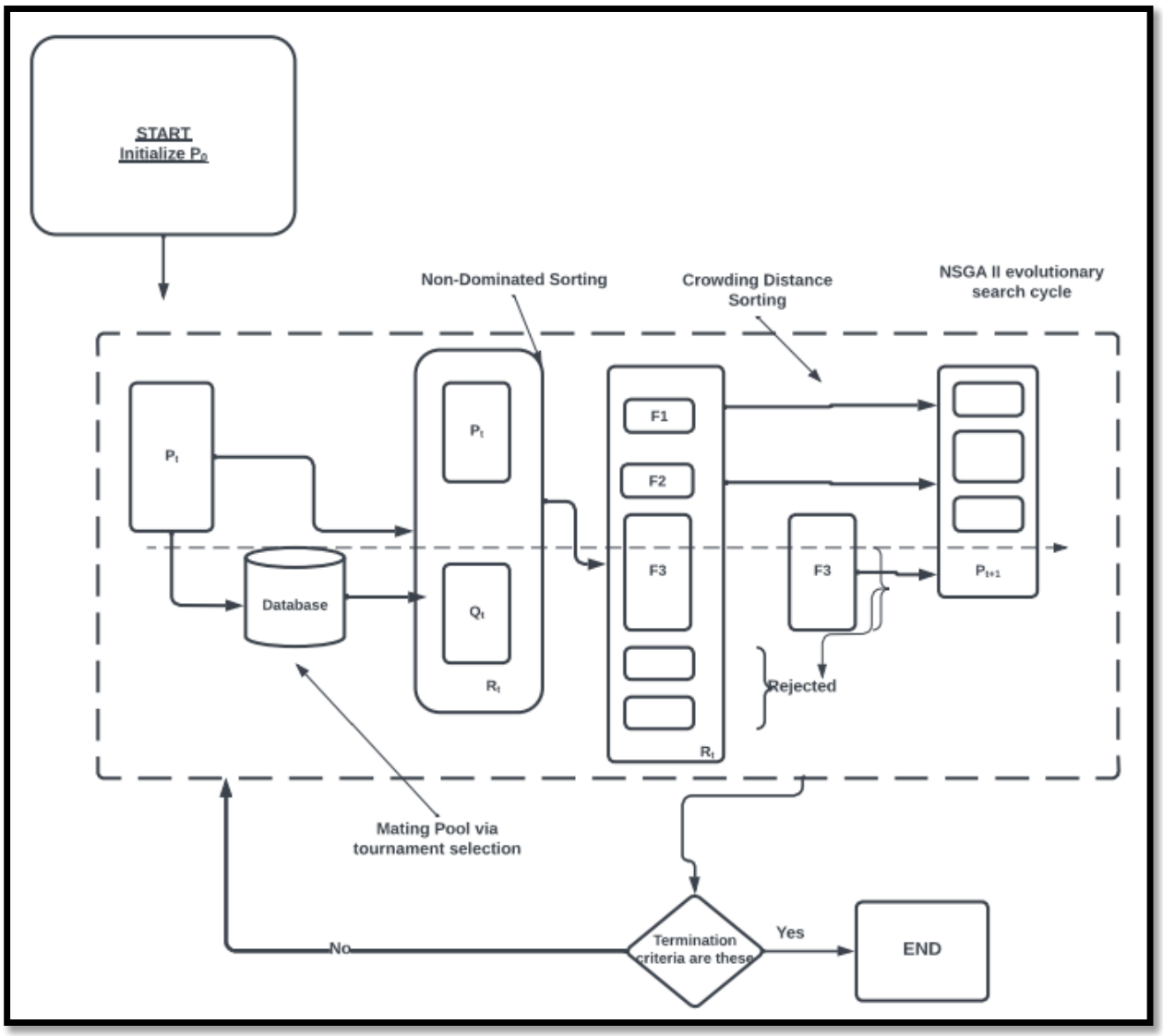

START
Initialize P0
Non-Dominated Sorting
Crowding Distance Sorting
NSGA II evolutionary search cycle
Pt
Database
Pt
Qt
Rt
F1
F2
F3
F3
Rejected
Rt
Pt+1
Mating Pool via tournament selection
Termination criteria are these
Yes
No
END

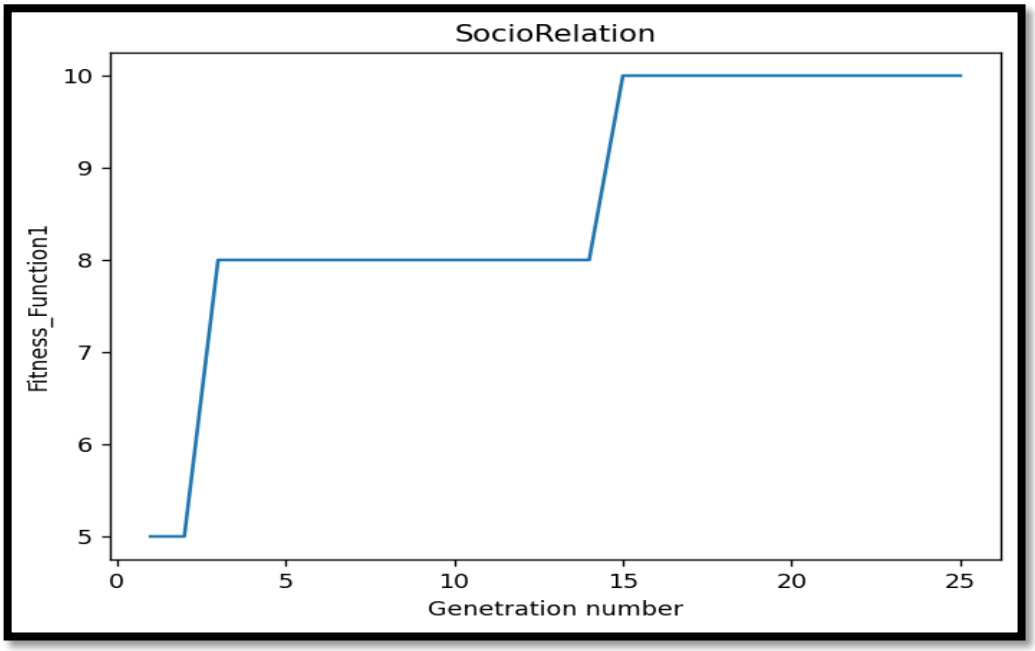
Fig 5.12

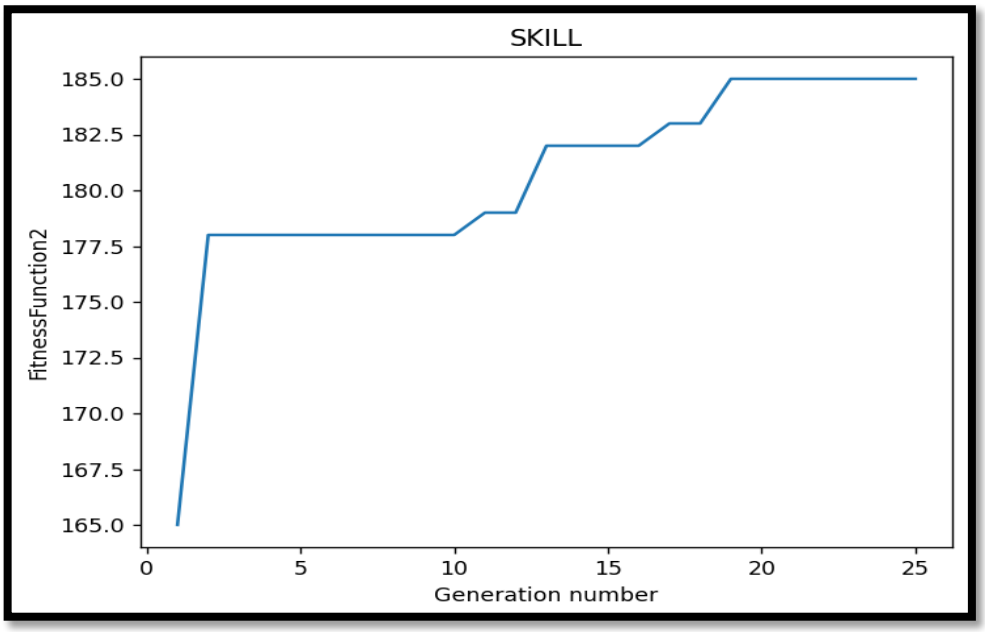
Results

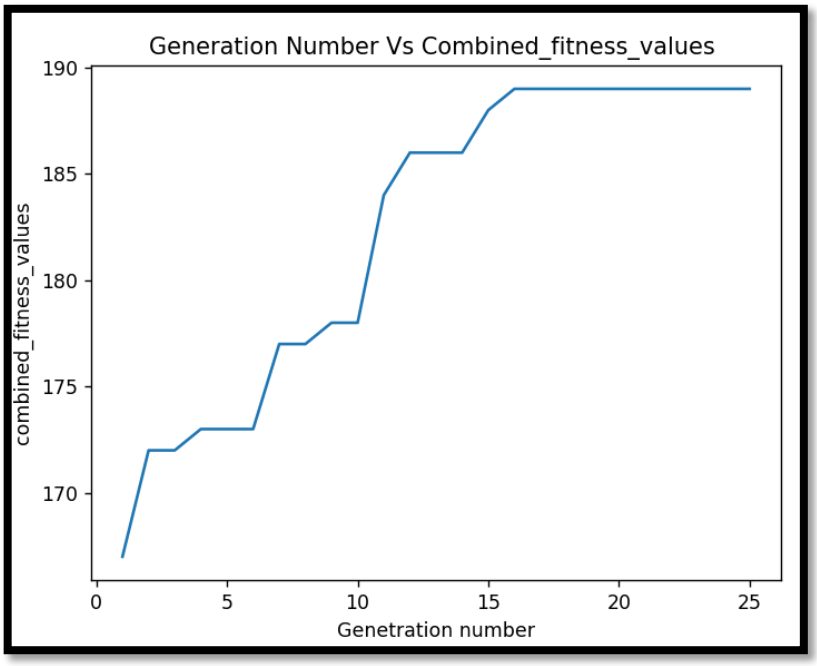
The experiment was performed on different project team size. Results are attached for 3 project teams, 4 project teams and 5 project teams in accordance with Genetic Algorithm, followed by results for 3 project teams with NSGA-II approach. This chapter ends with a look over why our approach is better than using the primitive exhaustive approach which takes 4+ days and still running to work on an expert population of 100 people.

For 3 teams using GA

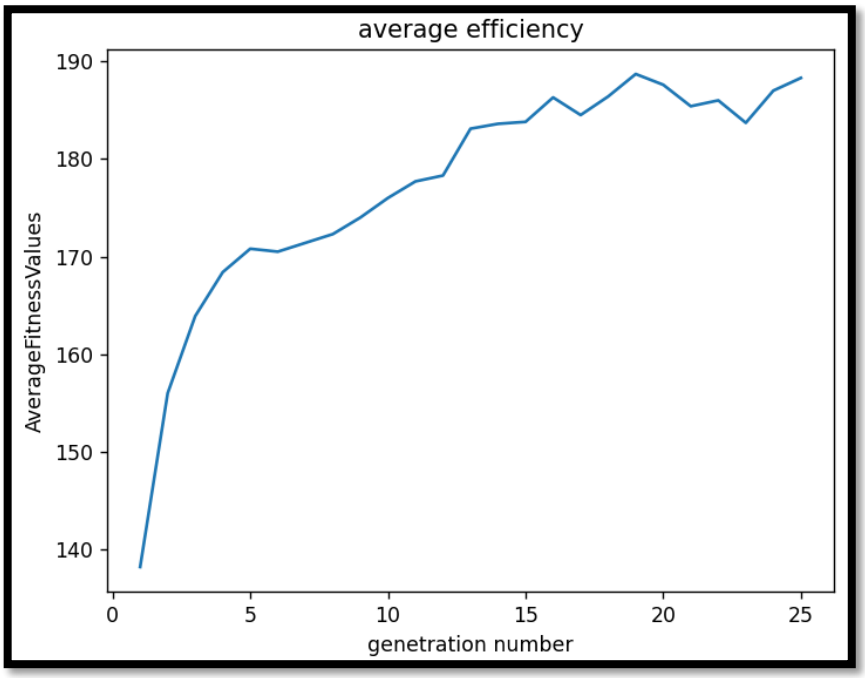
```
requirement_matrix = [[1, 1, 0, 0, 1, 1], [
    1, 0, 0, 0, 1, 0], [0, 1, 1, 0, 0,0]]
```





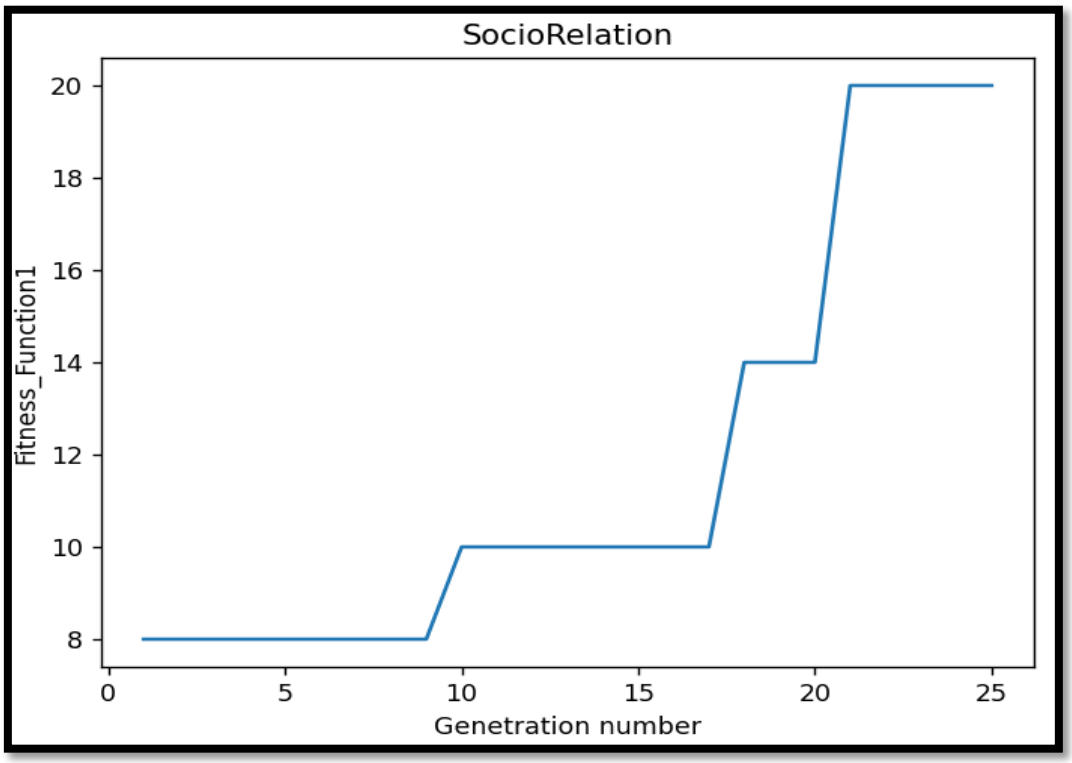


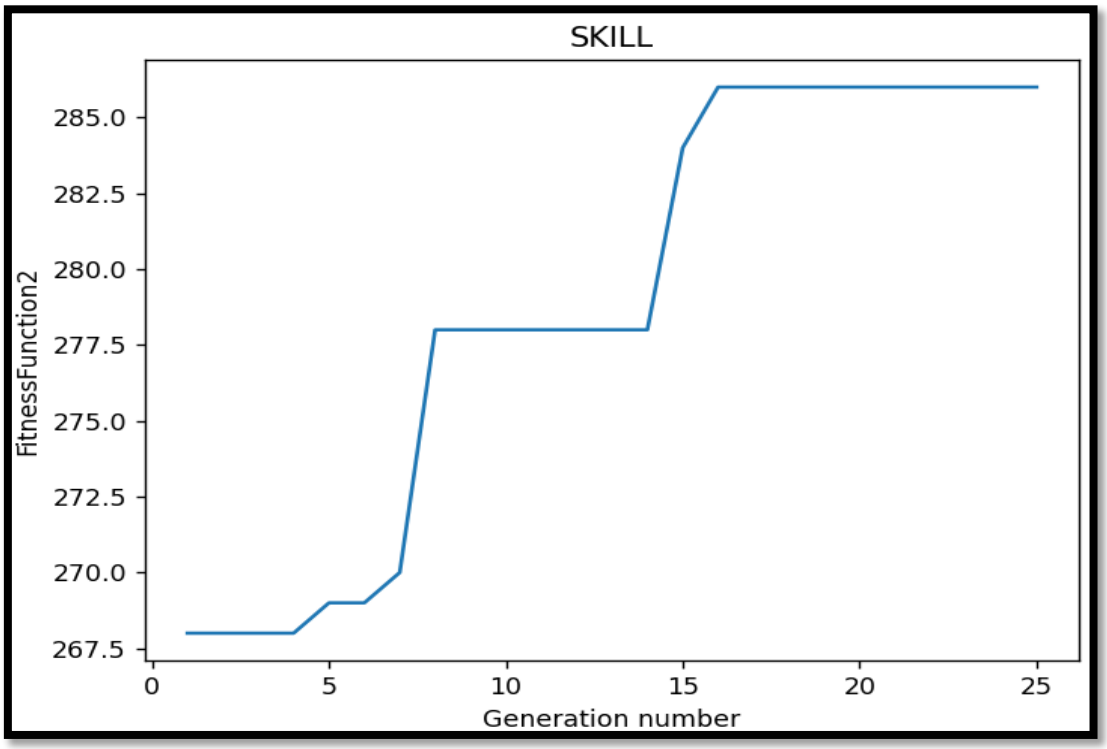

```
Fittest Chromosome [6, 100, 7, 15, 40, 22, 10, 9]
Fitness Value 190
Total Time Taken 0.05710601806640625
```

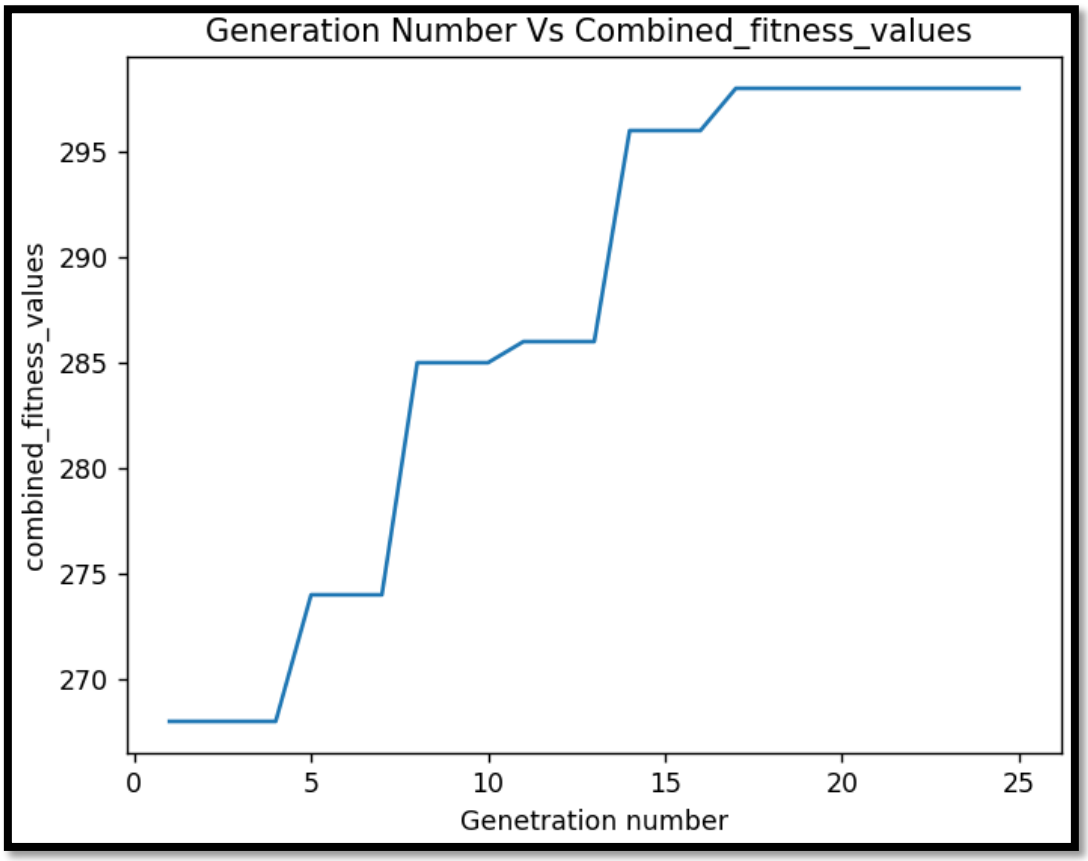

average efficiency
AverageFitnessValues
190
180
170
160
150
140
0
5
10
15
20
25
genetration number

For 5 teams using GA

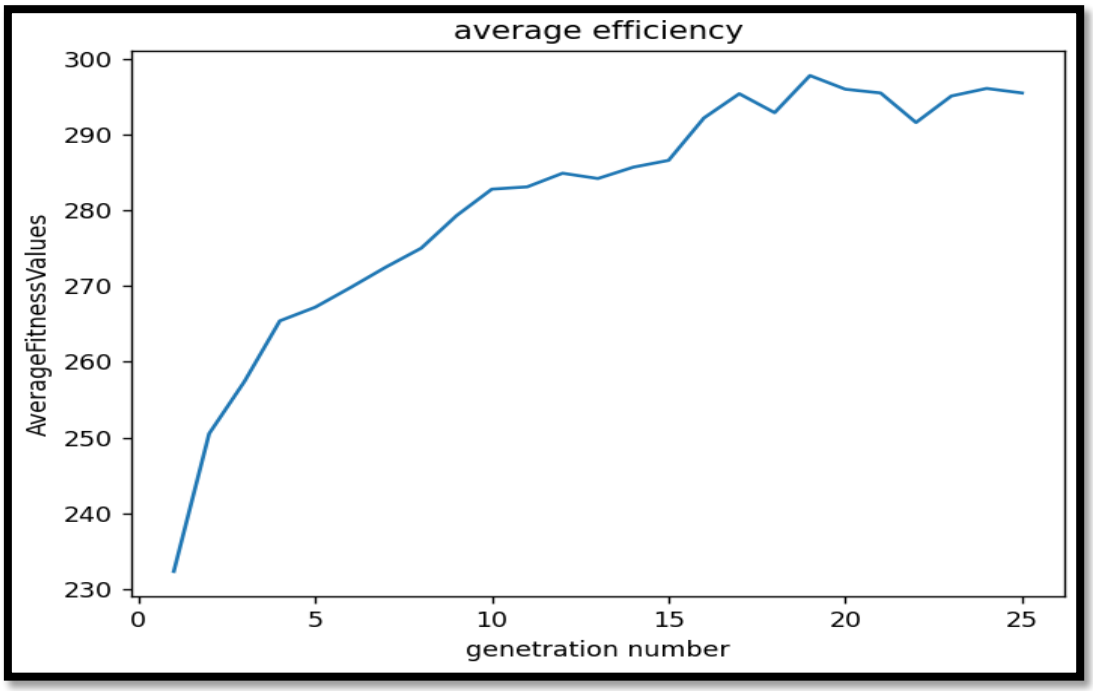
```
requirement_matrix = [[1, 1, 0, 0, 1, 1], [
    1, 0, 0, 0, 1, 0], [0, 1, 1, 0, 0,0],[0,0,1,1,0,0],[1,0,0,1,0,1]]
```





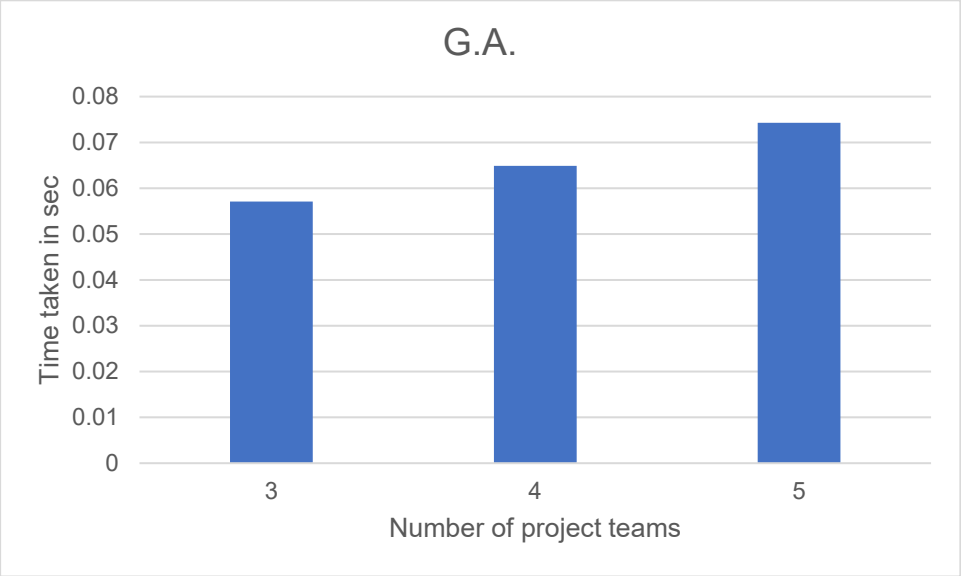



```
Fittest Chromosome [3, 19, 42, 5, 4, 10, 100, 25, 99, 53, 40, 98, 6]
Fitness Value 298
Total Time Taken 0.07433700561523438
```

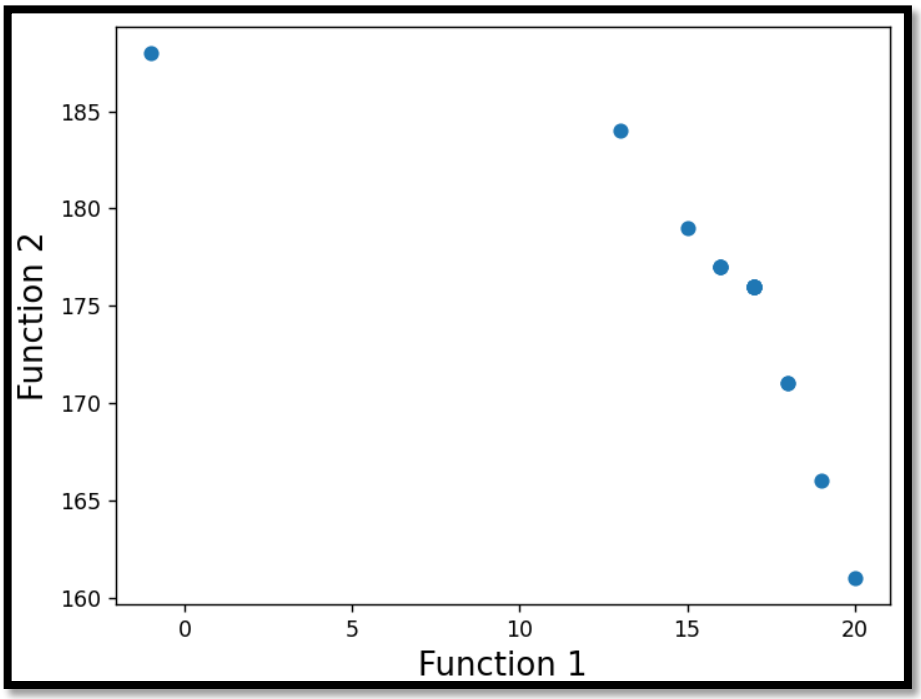


After analysing 3 such results, formulating the end results in a table:

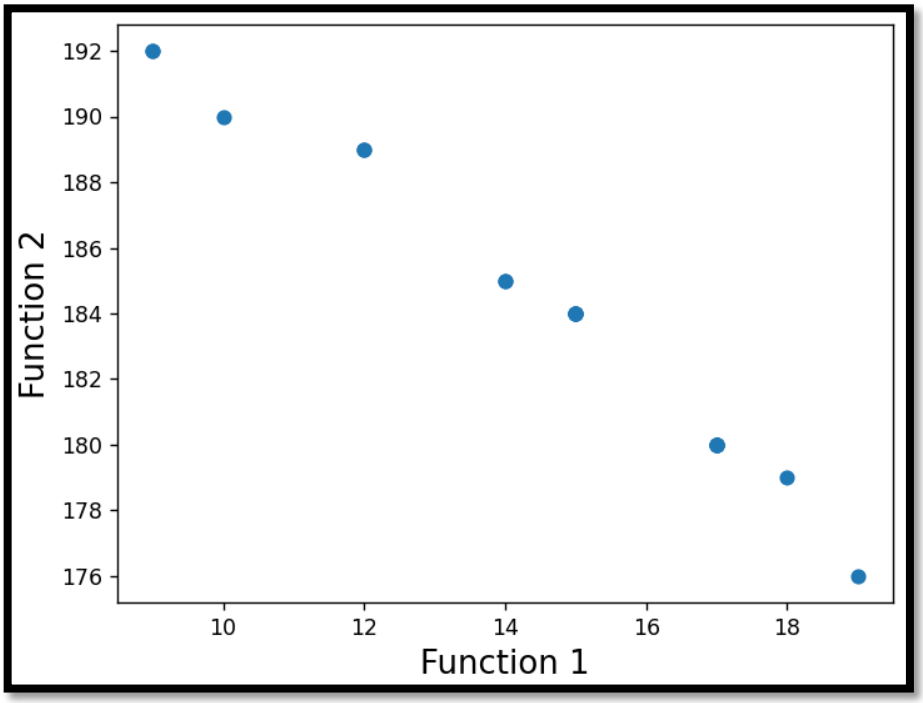
Project Teams	Skills	People Available	Time taken to compute fittest chromosome
3	6	100	0.0571
4	6	100	0.0649
5	6	1000	0.0743



For 3 teams using NSGA-II

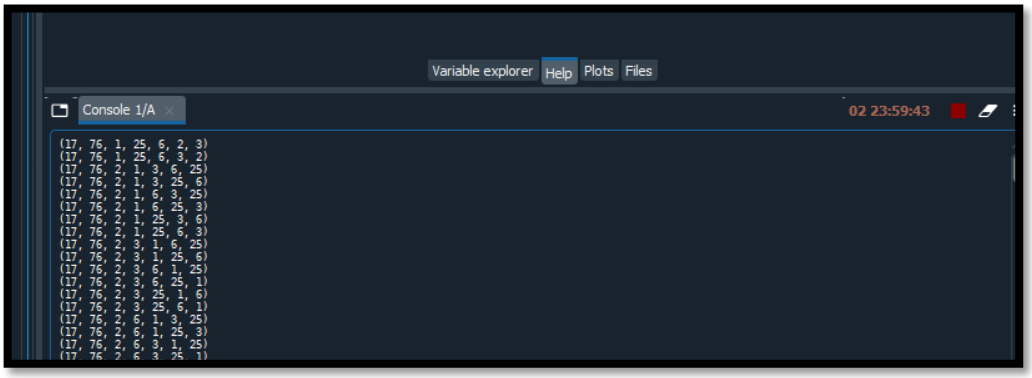


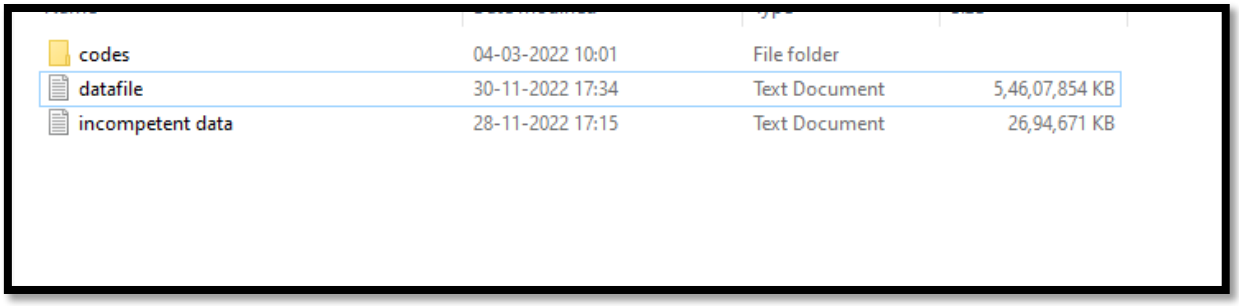

```
The best front for Generation number  920  is
[47, 19, 4, 29, 6, 10, 40] [47, 19, 80, 29, 6, 93, 40] [47, 19, 4, 29, 6, 87, 40] [47, 19, 4, 29, 6,
93, 40] [47, 19, 4, 29, 6, 30, 40] [47, 19, 4, 29, 6, 10, 40] [47, 46, 4, 29, 6, 10, 40] [47, 19, 4,
29, 6, 93, 40] [47, 19, 4, 29, 6, 75, 40] [47, 19, 4, 29, 6, 10, 40] [47, 19, 4, 29, 6, 93, 40] [47,
19, 4, 29, 6, 10, 40] [47, 19, 4, 29, 6, 10, 40] [47, 46, 4, 29, 6, 87, 40] [47, 19, 4, 29, 6, 93, 40]
[47, 19, 4, 29, 6, 10, 40] [10, 19, 60, 29, 6, 22, 40] [47, 19, 4, 29, 6, 10, 40] [47, 19, 4, 29, 6,
10, 40] [47, 19, 4, 29, 6, 10, 40]
```



For 3 teams using Exhaustive Approach

This approach took more than 3 days and yet not came to a visible conclusive end. Hence evident that our proposed method of forming multi-skilled team is far better, feasible and time efficient than other mentioned methods.





Future Scope of this Project

This paper was written in accordance with the constraints that includes one expert can work only in one project at a time, no part time working was accounted for. So, one such scope can be inculcated in this work for a better and efficient team formation.

Another possibility is that the data which was taken for team formation, i.e., experts' skills data, with the help of a google form for now, can be acquired directly via scanning the expert's resumes and CVs or by processing their LinkedIn profiles via a pre-defined algorithm to extract specific skill data.

References

[1] Team Formation by G.K. Awal & K.K. Bharadwaj

[2] Multi-Objective Team Formation by Bahareh, Hamzeh, Eghlidi & Afshar

[3] Group Formation Using Genetic Algorithm by Zhamri, Ani, Yasin, Hussin & Hamid

[4] Application of GA for MTF by Esgario, Silva & Krohling

[5] Multiple Team Formation using Evolutionary Approach by V.S. Baghel & S. Durga Bhavani

[6] https://towardsdatascience.com/introduction-to-genetic-algorithms-including-example-code-e396e98d8bf3

[7] https://www.geeksforgeeks.org/genetic-algorithms/

[8] https://en.wikipedia.org/wiki/Genetic_algorithm

[9] https://oklahomaanalytics.com/data-science-techniques/nsga-ii-explained/