UTM

UNIVERSITI TEKNOLOGI MALAYSIA

COMPUTATIONAL INTELLIGENCE

SCSJ3563

SECTION 03

2021

GROUP PROJECT

DR HAZA NUZLY BIN ABDULL HAMED

NAME	MATRIC NUMBER
PUTERA MUHAMMAD SYABIL BIN SARIANTO	A18CS0235
PHARVEISH A/L SELVAM	A18CS0233

SECTION A

Declaration Page

1. We understand what plagiarism is and we are aware of the University Teknologi Malaysia's policy in this regard.
2. We declare that the work hereby submitted is our own original work.
3. We have not used work previously produced by another student or any other person within section and other section to hand in as our own.
4. We have not allowed, and will not allow, anyone to copy our work with the intention of passing it off as his or her own work.

SECTION B

Problem statement

Having a customized PC puts a lot of advantages on us because it is built based on our requirements. It also can increase the performance of the PC as it allows us to pick desired components that focus on performance and we can get it for a cheap price as it opens options for us. However, when comparing custom PCs with built-in PCs, people tend to buy built-in computers. The main reason this happens when buying a PC is because they are unfamiliar or have no knowledge with computer components. The terminology of PC components and the use of it could be overwhelming for some people especially to those who are elders or non primary users of computers.

Thus, this project aims to make a recommendation program that is capable of selecting PC components to build a custom PC based on the user budget and PC performance. The hardware of PC components that will be used to build a PC in this project is processor, motherboard, memory, graphic card, hard disk, power supply and also a monitor.

Algorithm Selection Justification

To select the best computer within a specified budget, which is RM3500, a genetic algorithm(GA) is the best algorithm to do so. The reason this algorithm is chosen is because many possible combinations of hardware can be chosen to build the best PC within the specified budget. So each hardware type will represent a gene and these genes will be a chromosome(computer).

Other than that, the problem in this project is focusing on optimization problems as we want to minimise our spending of resources and want a good combination of hardware for PCs. Hence, in our GA, the chromosomes will undergo genetic algorithms such as crossover and mutation for 200 generations in order to determine the best computer for the given budget.

Genetic representation is used to represent the solution in an array that exists in a phenotype space. In this GA implementation , we will be using integer representation. Integers are whole numbers with no negative values.

Chromosome is a set of parameters which contain the proposed solution by genetic algorithm to solve the problem. The set of solutions is called population.

Chromosome\Gene	Monitor	CPU	Mainboard	Graphic Card	Power Supply	RAM	Storage
1	8	6	5	6	1	3	7
2							
…							

In this problem, we will use this table as the chromosomal representation. We have 7 hardware items to choose from, meaning that there are 7 genes for each chromosome. Each gene will have 10 items, meaning 10 models of hardware for each type.

GA Development Steps

Data collection:

The data used in this project is based on a catalogue of a customized computer shop, C-ZONE SDN BHD located in IOI Putrajaya. The price used is the standard retail price that usually consumers had when buying a custom PC. And our budget for buying a PC is RM3500 as this project is aiming for a low-end performance PC. Hence RM3500 is a reasonable budget and a good benchmark for a low-end PC.

Variable:

There are 6 variables for each hardware component that were used in this system. The variables are CATEGORY, ID, BRAND, PRICE, PERFORMANCE, and DESCRIPTION.

Initialize Procedure:

Initialization is a first step in the GA process. Usually initialization is done at random. It needs to ensure even spread and mixture of the possible allele values. In this problem, we also use random initialization for each gene in a chromosome. Each gene we randomly generate an integer.

For the variables, we set it as:

1. Budget = RM3500.00
2. Crossover Probability = 0.9
3. Mutation Probability = 0.1
4. Population Size = 30
5. Generation = 200

Fitness Function:

Fitness function is an evaluation function that will evaluate how good the solution given for optimizing the parts price with the budget given. The higher the fitness, the better the solution provided by the GA.

This is the fitness function,f(x) in this GA :

$$if(Total\ Price\ <\ Budget)$$

$$f(x)\ = \frac{Total\ Price}{Budget} \times\ 100$$

$$if(Total\ Price\ >\ Budget)$$

$$f(x)\ = \frac{Budget}{Total\ Price} \times\ 100$$

Since this GA chooses the total price that is as close as the budget given, it means that the total price can be either higher or lower than the Budget.

Example :

Monitor : SAMSUNG 21.5 TN MONITOR 1920x1081 (RM309)

CPU : AMD APU A6 9500 2/2 3.5GHz (RM 170)

Mainboard : GALAX B450M (RM249)

Graphic Card : GIGABYTE RX 570 GAMING 4G (RM629)

Power supply : DX/AX BRONZE 500w (RM195)

RAM : VALUE 8GB (RM169)

Storage : PATRIOT SCORCH 128GB (RM149)

TotalPrice = 309+170+249+629+195+169+149

=1870

Since TotalPrice<Budget

Use

$$f(x) = \frac{(Total\ Price)}{Budget} \times 100$$

$$f(x) = \frac{(1870)}{3500} \times 100$$

$$f(x) = 53.4286$$

Therefore, fitness for chromosome 1 is 53.4286.

Parent Selection: Tournament selection

Parent selection is a process to select a parent with mate and recombine to create the new generation. This step is also important since a good parent will make the algorithms become better and fitter. In this problem, we used tournament selection to select parents. We select the individuals from the population at random and choose the higher fitness to become the parent. This process is repeated for selecting the next parent.

Crossover: 1-point crossover

Crossover is a process where it will crossover the parents and create children. In this problem, we used 1-point crossover. The GA algorithm will choose a random point on two parents and shift it. It will create children by shifting the tails.

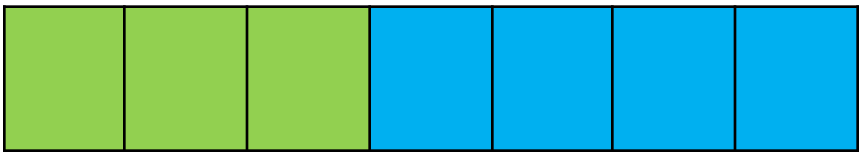
Example:

Parent 1:

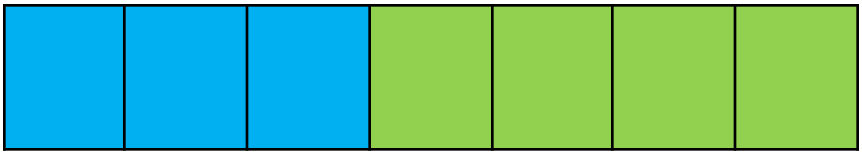
Parent 2:

After 1-point Crossover occur at random point

Children 1:



Children 2:



In our project, we set the crossover probability as high as possible to ensure that the crossover will occur in GA. For mutation probability, we set it as low as possible since we can.

Mutation: Random resetting mutation

Mutation is a genetic operator used to maintain genetic diversity from one generation of the population of the GA chromosomes to the next generation. Mutation alters one or more gene values in a chromosome from its initial state. In this problem, we use random resetting mutation. We select one or more genes and replace their value with another random value.

Example:

Before mutation

8	6	5	4	1	3	7

After Random Resetting mutation

8	7	5	4	1	4	7

Mutation alters one or more gene values in a chromosome from its initial state. Hence GA can come to a better solution by using mutation. Mutation occurs during evolution according to a user-definable mutation probability. This probability should be set low. If it is set too high, the search will turn into a primitive random search.

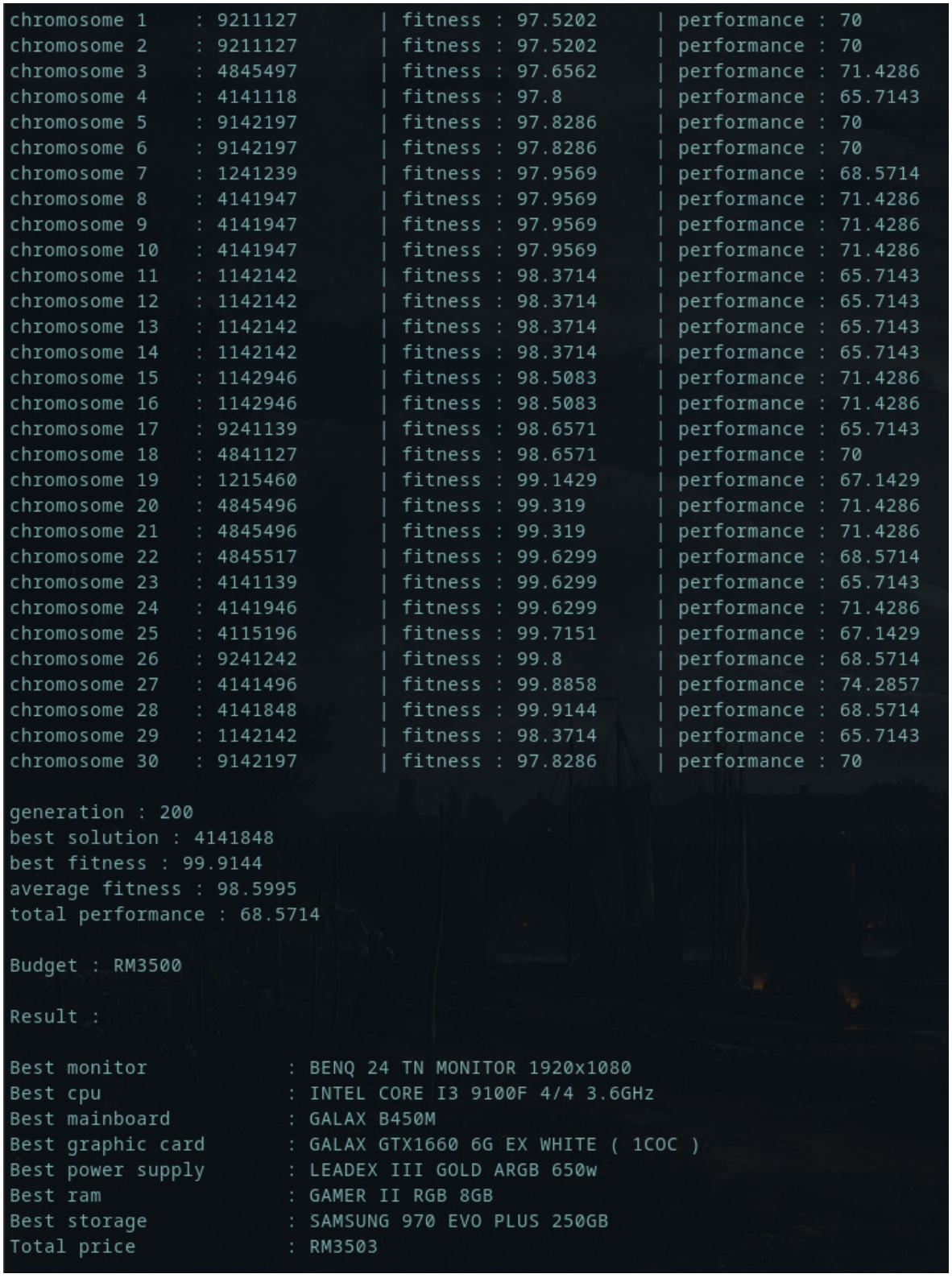
Survival Selection: Fitness based selection

Survivor selection is a replacement process based on the factors that are already set based on the problems. But sometimes survivor selection does a combination to replace it. In this project, we use fitness based selection. The worst chromosomes will be removed and only two chromosomes will be chosen as the next generation.

Termination Condition:

Termination condition is important in determining when the GA will end. The termination condition is usually applied when the solution is close to the optimal or at the end of the run. In this project the termination condition is **when the generation is equal to maximum which is 200 or if found chromosome with 100 fitness.**The GA will keep running until it meets one of the termination conditions.

Interface PrintScreen



The above figure shows the final generation of our GA algorithm. It is the final output of our developed program. It shows the final generation of the chromosomes, which has undergone crossover and mutation. It will also display the best solution, best fitness, average fitness of all chromosomes and the total performance of the best chromosome And then it shows the planned budget, the details of the best hardware combination of the computer(chromosome with best fitness) and the total price of that hardware combination.

SECTION C:

Procedure:

1. The budget is set as RM3500

2. Set the variable as constant value which as Crossover Probability is set as 0.9, Mutation Probability is set as 0.1, Population Size is set as 30 and Generation is 200

3. The algorithm will start to generate based on the input. All the genetic process will occur. (Parent selection, crossover, mutation and survival selection.) For each gene, it will choose the best item from 10 items in the list by checking it performance and price that suitable with the budget given.

Pseudo code:

```
BEGIN

    INITIALISE population with random candidate solutions ;

    EVALUATE each candidate ;

    REPEAT UNTIL ( generation ==200 || fitness==100) DO
```

1 SELECT parents ;

2 CROSSOVER pairs of the parents ;

3 MUTATE the resulting offspring ;

4 EVALUATE new candidates ;

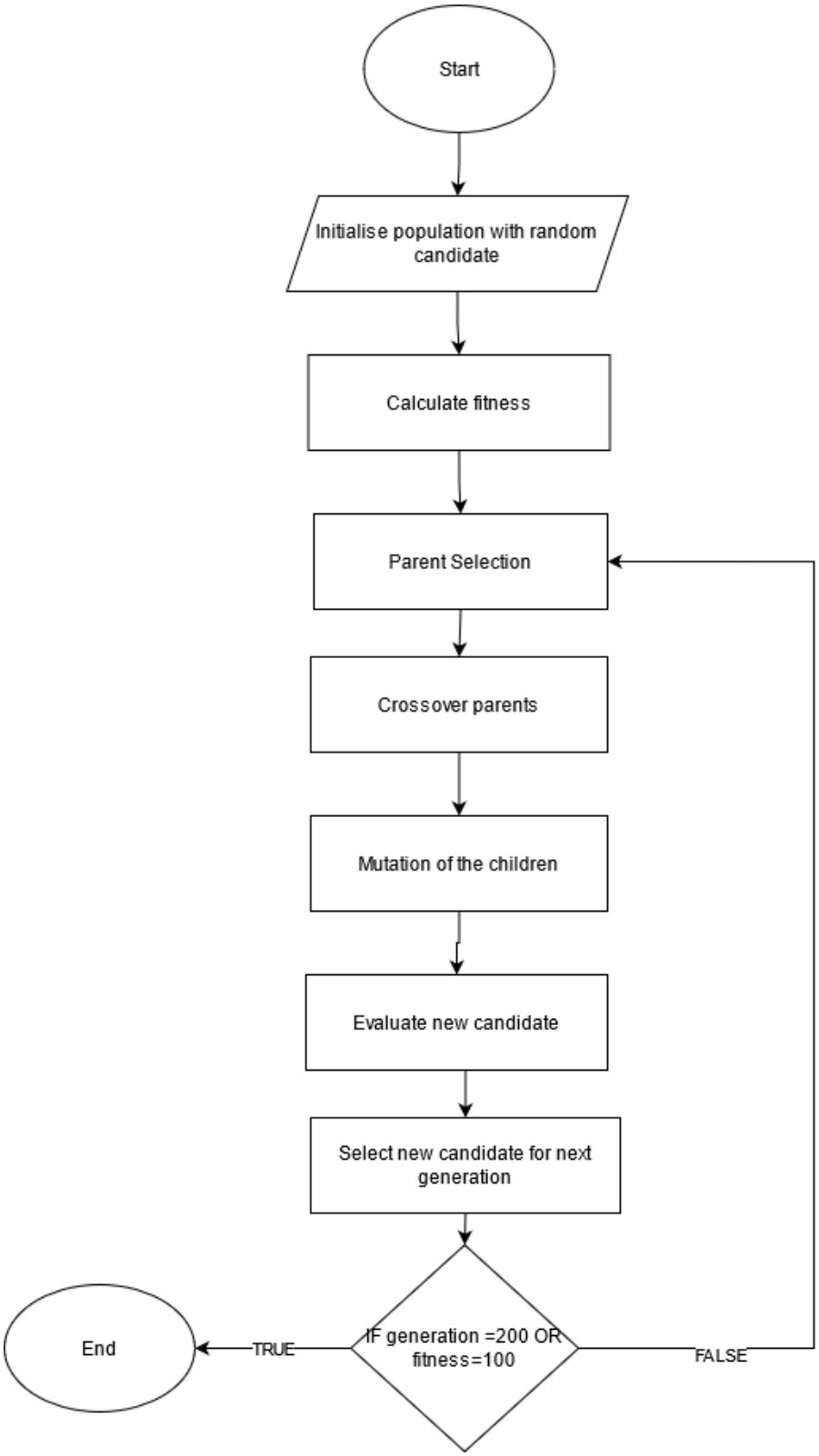
5 SELECT individuals for the next generation ;

OD

END

4.It ends once the termination condition is achieved. After that the most suitable suggestion for each hardware to build the best PC according to the budget will be shown along with the total price of those items.

Flow chart:


Start
Initialise population with random candidate
Calculate fitness
Parent Selection
Crossover parents
Mutation of the children
Evaluate new candidate
Select new candidate for next generation
IF generation =200 OR fitness=100
TRUE
End
FALSE