2020

23rd Annual High School Mathematical Contest in Modeling (HiMCM) Summary Sheet
(Please make this the first page of your electronic Solution Paper.)

Team Control Number: 10878
Problem Chosen: B

Please paste or type a summary of your results on this page. Do not include the name of your school, advisor, or team members on this or any page.
Papers must be submitted as an Adobe PDF electronic file, and typed in English, with a readable font of at least 12-point type. Papers must be within the 25-page limit.

Many social communities have poured amounts of resources and funds to support an increasing number of endangered plant conservation project. Yet there is a huge chance that some projects are over tended whereas others are in sever poverty of funds. To better help endowment organizations--FRPCE, in particular--fund biodiversity conservation programs for endangered and threatened species, we practiced a series of methods to 1) determine the best arrangement of the specified 48 conservation projects' starts and 2) the minimum funds to be raised.

For the first task, a number of variables were determined and then formed into two sub-objectives, the profits of each program and equilibrium. Then, we were able to apply genetic algorithm (GA) to this question, using the two objectives as parameters. Through running codes of GA we wrote in MATLAB, we found one particular plan that was fairly close to global optimal output. This plan gave us corresponding costs per year, economic value generated by each program after it was completed, and the beginning year of each project out of 48. Thus, we finished our first mission.

The second one – to figure out the minimum funds – is essentially gained based on the result from the precious task. That is to say, now that we already had the costs in each year, and we had assumed the funds to be raised changed linearly, what we needed to do is to search for the best-fit line with which the funds needed every year could accordingly follow. To do so, Lingo was adapted as the tool to achieve the optimization. It turned out that the best-fit line is horizontal, namely, a constant. Hence, the funds needed to be raised is all the same.

Dear FRPCE Board,

Thank you for giving us the opportunity to assist you to efficiently raise money. We have come up with great results from our model.

As per your requirements, we have carried our preliminary analysis on the optimal arrangement for each plant conservation project to be started. Specifically, we first list a number of characteristic of plants, among which we choose some factors that we are most intrigued and deem as most important. We analyze selected ones in details. Then, we constructed an algorithm designed to search for the best fundraising plan based on the following factors:

- The economic value brought by each program after it is finished
- The variance in costs for every year

Our result indicated a very great consistency between these two factors. That is, our plan maximize the benefits generated by each project, and at the same time minimize the variance, as can be seen in our model result part. We are confident to say that we have successfully completed the first task.

We solved the second task on the basis of the result of the first problem. Precisely, we optimized on the costs for each year to attain the minimum funds to be raised. By assuming you will change the amount of funds every year linearly, we realized our optimization through lingo. The result showed that the minimum fund you can raise every year is actually a constant.

In the end, we sincerely hope you adopt our plan, as it would save a lot of budget and more efficiently raise money.

Best,
Team 10878

Content

1.Introduction

In modern society, most people have the awareness of a mammal or bird that has become extinct in recent centuries, but few could name an extinct plant. In fact, since the 1750s, at least 571 species of plants have gone extinct in the wild, according to a global survey recently published in Nature Ecology & Evolution; more than eight plant species have disappeared worldwide every three years, on average, since 1900. This pace of extinction is as much as 500 times plants' natural or background extinction rate. If the range is narrowed within the United States, no region reflects the importance of plant species more than South Florida. The South Florida Ecosystem supports the only subtropical ecological communities in the continental United States: about 60 percent of the native plant species in the south of Lake Okeechobee originated from the tropics. However, a dramatic population increase and economic expansion in South Florida has been accompanied by extensive land-use alteration. In the past 50 years, more than 3,237,485 ha (8 million acres) of forest and wetland habitats have been cleared in Florida to accommodate the expanding human population, causing many plants to become endangered or even extinct. Therefore, it is of such importance to take measures to protect these endangered plant species from now on.

We are asked to assist The Florida Rare Plant Conservation Endowment (FRPCE), which aims to provide long-term funding to support conservation-related projects for Florida, to 1)recommend the priority of recovery projects as to when they should be funded, 2)find minimum funds to be raised, and 3)writing a non-technical memo for FRPCE board to explain our results and make suggestions. There are three subgoals in the first task: we have to identify some objective functions contributing to the funds required to be raised, come up with a "best" plan using these objectives, listing plants' general characteristics, and state those we will apply in our model.

2.Assumptions and justifications

Assumption 1: Individual plant protection projects will not generate benefits until they are completed, and from then the benefits would be produced annually.

Justification: We consider the economic value of a project as a process. There must be a period of time when there is no profit, instead, merely costs required to make the project run. After, there will be benefits. This makes sense because only after a plant is successfully saved, can it generate profits through attracting tourists to visit this plant, which is also why the profits would generate annually.

Assumption 2: A 25-years limit exists to bound all conservation projects.

Justification: Since we assumed the benefits brought by a program will start creating after it is completed, if we do not limit the time, the benefits would go on indefinitely. Thus, it is of great need to set a boundary to the maximum span of a project. According to the attachment sheet, it would be best to set 25 years as the maximum duration, because it only contains data within 25 years.

Assumption 3: All projects will not suffer halfway failure.

Justification: In reality, plant protection projects may fail in the middle because of many factors such as technical problems and natural disasters. As those factors are not important to our model, we consider that the success rate given in the attachment is the probability of success after the completion of the project, excluding the probability of failure halfway.

Assumption 4: If a project no longer produces cost, it means that the project ends

Justification: If a plant protection project does not produce cost from year (j +1) to year 25, it means the project has been finished and suggests the organization will not invest money and give consideration to improve it. Therefore, we assume that the project ends in year j.

Assumption 5: The cost of each project is not influenced by inflation.

Justification: Since the entire plant-protection plan lasts 25 years long. From an economic point of view, monetary inflation is a certainty. However, both the currency and the proposed amount of funds raised would be affected by inflation, equivalent to the fact that it would have no change in their value. So in the model, we don't take inflation into account.

Assumption 6: Funds for every year follows a linear pattern to change.
Justification: For the sake of reality and practicality, there is by no means a sudden huge drop or rise in the number of funds raised across years, which can lead to concerns among donators. A linear pattern, however, allows the funds to change steadily.

Assumption 7: All projects are carried out under normal conditions and environment

Justification: All conditions and requirements contained in the project will operate under normal conditions, excluding any unforeseen contingencies that may result in a sudden increase or decrease in funding requirements.

3.Model of fundraising schedule

3.1 Goals of Model

Our model seeks to address the following concerns:

- Construct the best fundraising plan, under related objectives we use, for each plant conservation program, 48 in total,.
- Determine the minimum funds to be raised.

3.2 Determination of objectives and variables

In order to make the plan reliable, and practical, we divide "best" into two parts: profits of the plan and the equilibrium of funds; but before going deeper, we have to first list some necessary variables in order to represent all the factors we are going to discuss.

3.2.1 List of variables

In order to formulate the mathematical model of making financial plans for plant conservation programs, we first list necessary variables defined below:

(1) x_{ij} is the decision variable. If i^{th} program starts in the j^{th} year, x_{ij} is 1. Otherwise, it is 0.

(2) d_i represents the time period required to finish i^{th} program in years.

(3) C_{ik} represents the cost of i^{th} program in the k^{th} year after its start

(4) q_j represents the need for funds in the j^{th} year.

(5) $\bar{q}$ represents the mean value of the total need for funds.

(6) y_{1i} represents the expected relative conservation value of funding i^{th} species over another.

(7) y_{2i} represents the uniqueness of the species conserved by i^{th} program.

(8) y_{3i} represents the chance of success for i^{th} program.

(9) t_i represents the time range available for i^{th} program to create benefits.

(10) $J \in \{1,2 \dots ,25\}$ represents time range.

(11) $I \in \{1,2 \dots ,48\}$ represents programs range.
all of which would be explained specifically later.

3.2.2 Objectives

When selecting objectives, we first listed a number of characteristics of endangered plant species: narrow geographical distribution, dependence on the special environment, poor diffusion ability, a small number of individuals, high economic value, poor reproduction ability, etc. Amid them, we choose factors which we are interested most to sum them up into two major aspects: how beneficial the plan is (profits) and how stable the financial demand across years is (equilibrium), which are discussed in detail separately.

(i) Profits

Factors influencing profits

Several factors are influencing profits, three of them are chosen in our consideration: we use "benefit" (y_{1i}) to reflect the economic value of the endangered species, and "uniqueness" (y_{2i})to represent the number of individuals. What's more, each program will only create benefits from the year after it succeeds. This indicates when calculating a program's benefit, we need to consider whether or not this program will be successful beforehand. Therefore, we introduce the third factor--"the feasibility of success" (y_{3i}).

Relationship between three factors and profits

Since the value of benefits (y_{1i}) is a measure of relative conservation value--given by attachment B--the profit then is directly proportional to y_{1i}. As for uniqueness" (y_{2i}), when it is higher, the benefit of the program will also get higher. The reason is that the impact of protecting them would be stronger to the ecosystem, as otherwise no any other species would make up for its absence. As a result, these two factors all affect the priority of the program positively and their relationship with profits (P) of the program should be expressed as:

$$P \propto y_{1i} y_{2i}$$

We can get the expected value of the possible benefit of i^{th} program by multiplying y_{1i} by y_{3i}. Besides, since we have assumed that after finishing the program, it will lead to benefit annually, we have to first calculate the remaining years for the program to generate economic value. Hence, we introduce a new variable t_i to store how many years available for i^{th} program to create benefit after it is finished. According to attachment B and our assumption, there is a 25-year boundary for us to finish all the programs, so only the benefit created in these 25 years will be considered.

Decision variable x_{ij}

Yet, since i^{th} program can start in any year as long as it can be finished in these 25 years, there should be a limit condition that $t_i \geq 0$ and we create a new variable $x_{ij}, i \in I, j \in J$, to determine whether i^{th} program starts in j^{th} year. x_{ij} is a decision variable, which is 1 if i^{th} program starts in j^{th} year, 0, otherwise.

Objective function for profits

Since the starting year for i^{th} program should also be included into the duration of i^{th} program, represented by d_i, the time creating benefits of i^{th} program can be expressed as:

$$t_i = 25 - (j + d_i - 1), i \in I, j \in J$$

Because every program will start at one of 25 years , when we are counting the total benefit of all the programs in these 25 years, we can iterate every j and see the value of x_{ij} to see if i^{th} program starts at j^{th} year. if so, we can then determine the remaining years for this program to generate benefits, as shown below:

$$\sum_{j \in J} t_i x_{ij} \ , \forall i \in I, \forall j \in J$$

If $x_{ij} = 1$, then this formula gives the benefits-generating years of i^{th} program, otherwise the whole formula would give 0.

Finally, we need to synthesis all the factors that can influence the profits into one single equation. Our first objective function Z_1—maximizing the profits of the whole plan-- can be seen below:

$$maxZ_1 = \sum_{i \in I} \sum_{j \in J} t_i x_{ij} y_{1i} y_{2i} y_{3i}$$

(ii) Equilibrium

In order to make the fundraising demand of our program more reliable and balanced, we select equilibrium of our financial plan, represented as E.

Annual cost q_k

First of all, we create a new variable q_k to represent the demand for costs of k^{th} year. Since each program will merely create financial demand when it is running, when calculating the financial demand for k^{th} year caused by i^{th} program, we need to determine whether k^{th} year is in the period between i^{th} program's start and its end. The i^{th} program continues for $(d_i - 1)$ years after it starts at j^{th} year, continuously creating financial demands during the lasting years. Therefore, as long as the starting year for i^{th} program is no more than $(d_i - 1)$ years before k^{th} year, i^{th} program will generate financial demand in k^{th} year. The decision variable can be expressed as $x_{i(k-g+1)}, g \in \{1,2 \dots, d_i\}$. Also, $(k - g)$ should not be less than 0.

Secondly, we can get another variable c_{im} from the attachment, which represents the cost for i^{th} program in the m^{th} year since its start. Therefore, when we need to calculate the costs for i^{th} program in k^{th} year, the expression will be:

$$cost = x_{i(k-g+1)} c_{ig} \ , g \in \{1,2 \dots, d_i\}$$

If $x_{i(k-g+1)} = 1$, it means that i^{th} program starts in year $(k - g + 1)$. In other words, this is the g^{th} year since its start, so there exists costs. Otherwise, the value of cost would be 0, indicating the program has been completed and thus there is not costs generated.

When we are able to express the cost for a single program in one year, we can apply the sum operator to the present equation, obtaining the total cost of each year:

$$q_k = \sum_{i=1}^{48} \sum_{g=1}^{d_i} x_{i(k-g+1)} C_{ig} \, , g, k, i \in N, g \leq k$$

Object function for equilibrium

After getting the expression of financial demand in a single year, we need to find a way to illustrate the variance of data, which is our second objective function(Z_2) and we want to minimize. According to the definition of variance, we get:

$$minZ_2 = \sum_{k \in J} (q_k - \bar{q})^2$$

where $\frac{1}{25-1}$ is missing here because it is a constant, which does not influence the result of optimization. The smaller the variance among data, the more balanced the financial demand is. Therefore, when we need a more reliable plan for fundraising, our object function should be to minimize the variance among data.

Standardization of data

Yet, since the data is huge, in which each $n_k, k \in J$ is higher than 10,000 dollars, the square will lead to a dramatic huge data if we put the raw data into the calculation directly. Hence, standardization of data is needed before calculating the variance. We employ the equation $x - x_{min} / x_{max} - x_{min}$ to normalize the data in order to map the result to the interval [0,1].

3.2.3 Constraints

There are some constraints for the two objectives. With decision variable x_{ij}, we can create a 0-1 matrix with a size of 25*48, denoting when each program starts. Since each program should start at least and only once, x_{ij} must meet the following conditions:

$$\sum_{j \in J} x_{ij} = 1, i \in I$$

and thus, for 48 programs altogether, it follows:

$$\sum_{i \in I} \sum_{j \in J} x_{ij} = 48$$

By these limits, it can be guaranteed that every program is taken while none is repeated.

3.2.4 Model as a whole

Finally, we can build our complete optimization model from precious discussion.

Objectives:

$$maxZ_1 = \sum_{i\in I}\sum_{j\in J} t_i x_{ij} y_{1i} y_{2i} y_{3i}$$

$$minZ_2 = \sum_{k\in J}(q_k - \bar{q})^2$$

Constraints:

$$\sum_{j\in J} x_{ij} = 1, i \in I$$

$$\sum_{i\in I}\sum_{j\in J} x_{ij} = 48$$

$$t_i = 25 - (j + d_i - 1), i \in I, j \in J$$

$$q_k = \sum_{i=1}^{48}\sum_{g=1}^{d_i} x_{i(k-g+1)} C_{ig}\ , g, k, i \in N, g \leq k$$

3.3 Genetic algorithms

3.3.1 The strategy of genetic algorithm

To solve our model, we adopt the genetic algorithm(GA). GA is a search heuristic inspired by Charles Darwin's theory of natural evolution. This algorithm reflects the process of natural selection. Its essence can be summarized into three parts: 1) heredity: there must be a process in place by which children receive the properties of their parents; 2) variation: there must be a variety of traits present in the population or a means with which to introduce variation; 3) selection: there must be a mechanism by which some members of a population have the opportunity to be parents and pass down their genetic information and some do not. This is typically referred to as "survival of the fittest".

3.3.2 Basic steps

The flowchart of the genetic algorithm for searching feasible layouts of slots each plant conservation project sits in for 25 years is shown in the figure below. The flowchart can be categorized into five processes: (1) chromosome encoding, (2) population size initialization,

(3) population fitness evaluation, (4) termination judgment, and (5) evolutionary operations of the genetic algorithm, all of which will be discussed in more detail later.

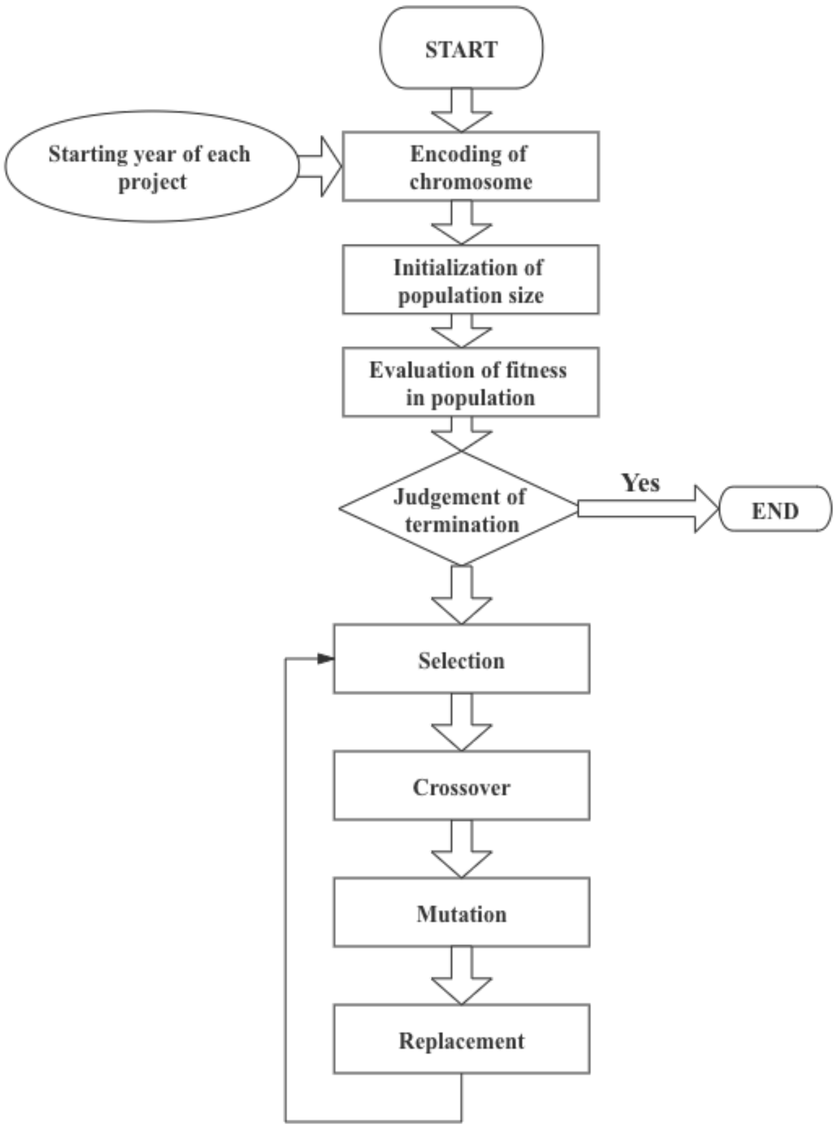

Figure 1. Flowchart of GA in our model

3.3.3 objective function

Before encoding for chromosomes, we first needed to set the objective function, otherwise called termination judgment, to evaluate each plan.

As mentioned before, currently there were two objectives taking into our consideration: the benefits (Z_1)and the equilibrium(Z_2). To optimize our plan more efficiently, we formed these two objectives into a single one, which we called Z_{total}. The relationship between these three objectives could be expressed as follows:

$$Z_{total} = \frac{Z_1}{Z_2}$$

As we discussed earlier, we desired to maximize the benefits and minimize the variation. Therefore, the optimal value of Z_{total} would be as large as possible.

3.3.4 Initialization of population and fitness evaluation

After setting up the objective function, we started encoding genes(data) for every chromosome. The chromosome, in our case, represented the starting year of 48 plant conservation projects with a length of 48, and thus the genes of a chromosome represented an array of starting years, so each chromosome (x) could be expressed as:

$$x = (t1, t2, t3, \ldots, t48)$$

where t_i represented the starting year of the ith project. The table below listed four examples of chromosomes after encoding the process, which would also be used for later discussion.

Table 1. Examples of chromosomes

	project 1	project 2	project 3	...	project 47	project 48
x_1	20	9	7	...	1	3
x_2	1	1	3	...	14	18
x_3	4	16	18	...	2	3
x_4	23	32	12	...	1	5

It was clear to see that for chromosome x1, it contained 20,9,7..3, indicating the first project started in the twentieth year, the second project started in the ninth year. the third project started in the seventh year and so on. Hence, each individual symbols a fund-raising plan for 48 projects.

Then, with the knowledge of encoding, a random population consisting of 200 individuals (possible candidate solutions) was generated at the initialization stage of the GA. Note here "random" meant randomly encoded chromosomes. We determined our fitness value for each individual to be equal to the value of its own objective function. Namely,

$$fitness(x_i) = Z_{total}(x_i)$$

For the convenience of later discussion, we adopted f as the abbreviation of the fitness value for each individual.

3.3.5 Evolutionary operations

There were essentially three key steps: *selection, crossover, and mutation*, which we would discuss in detail separately.

Selection: this step was achieved by roulette wheel selection and Monte Carlo methods. Roulette wheel selection stated that the probability of one individual being selected was determined by accumulative probability (q). Accumulative probability was equal to the sum of all individuals' possibilities of being selected before itself. The reason we didn't directly use the probability of one individual to be selected was that this could let the algorithm be trapped in local optima. Therefore, we needed to compute the possibility for each individual to be selected $p(x_i)$:

$$p(x_i) = \frac{f(x_i)}{\sum_{i=1}^{N} f(x_i)}$$

where $f(x_i)$ denoted the fitness value for individual x_i. This indicated that p equaled the ratio of an individual's fitness value to the sum of the fitness value of the population. Then, q could be expressed as:

$$q(x_i) = \sum_{t=1}^{i} p(x_t)$$

After computing every individual's q, we adopt Monte Carlo methods to perform selection on the population: Firstly, an array m(size) was created, where size denoted the size of the population. Each value in the array was a random number ranging from 0 to 1. This array was used to give a random probability and then to be compared with accumulative probability to determine whether or not an individual was selected. The specific mechanism could be illustrated by the example below:

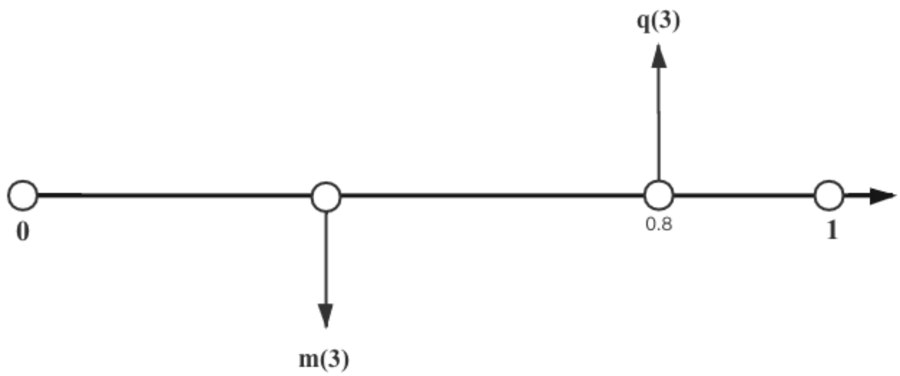

Figure 2. Illustration of Monte Carlo methods

In this example, the third individual was being examined, and its accumulative value q(3) = 0.8. The random array gave the corresponding value m(3). We could think of m(3) here as a pointer directing to the fact that this individual was being selected. If the value of accumulative value included the value of this pointer, it meant this individual was officially selected. In the example, since m(3) < q(3), m(3) dropped within the range of q(3), indicating that the third individual was selected.

In essence, if $q(x_i)$ was larger than the element m(i), then individual x_i was selected to go through the next step--crossover--and move currently examined element in the matrix to the next one. Otherwise, move currently examined individual to the next one.The table below showed examples of selection operation.

Table 2. The process of selection operation in GA

	project 1	...	project 48	$p(x_i)$	$q(x_i)$	times of being selected(4 in total)
x_1	20	...	1	0.4	0.4	2

x_2	1	...	14	0.2	0.6	1
x_3	4	...	2	0.3	0.9	1
x_4	23	...	1	0.1	1	0

As we could see in the table above, even though the accumulative probability for x_4 ($q(x_4)$) was 1, it was never selected. This meant that whether or not this individual was selected was proportional to the probability of one being selected but not completely.

This whole process was iterated until the maximum iteration time was reached. After that, the population set was updated as $S_{new\ population}$.

Crossover: the new generation was produced by implementing the one-point crossover between two parents. The steps involved in the crossover operator could be described as follows:

(1) Two individuals were selected according to the selection operator. In our case, starting from i=1, we selected individual x_i to be paired with individual x_{i+1}.

(2) Generated a random crossover point in the two parents.

(3) Exchanged genetic materials, in this case, the starting year of each project, after the crossover point.

Noted that we set our probability of crossover to 0.95.

The table below showcased this process by using four chromosomes mentioned previously, showing the examples of crossover operation in GA:

Table 3. Chromosomes after selection operation

x_1	20	9	7	...	1	3
x_2	1	1	3	...	14	18
x_3	4	16	18	...	2	3
x_4	23	32	12	...	1	5

crossover
======>

Table 4. Chromosomes before selection operation

x_1	20	9	18	...	2	3
x_2	1	1	12	...	1	5
x_3	4	16	7	...	1	3
x_4	23	32	3	...	14	18

where chromosome x_1 was paired with x_3, and x_2 with x_4. The crossover point was 3, so paired chromosomes would start exchanging their data of the starting year from the third project.

Just as the process of selection, we repeated crossover over and over again. Then we anew updated the population $S_{new\ population}$.

Mutation: to perfectly simulate natural selection, mutation was bound to be considered. we set the probability of mutation to 0.001. This meant in the process of traversing the

population, for each individual, if the generated random number was smaller than the probability for mutation, it was selected and undergoes random alteration of its genes. In our case, we mutated genes by changing the starting year of a certain project to be randomly generated numbers which were integers from 1 to 25. The figure below showed an example of a mutation operation.

Table 5. Examples of mutation operation in GA

	Before mutation						After mutation					
x_1	20	9	7	...	1	3	7	9	7	...	1	3
x_2	1	1	3	...	14	18	1	1	2	...	14	18
x_3	4	16	18	...	2	3	4	16	18	...	3	3
x_4	23	8	12	...	1	5	23	10	12	...	1	5

Since the probability of mutation was set to 0.001, it was a very rare thing that many gene materials in a chromosome was changed at the same time. In the table above, among four chromosomes, there was only one data, namely the starting year of one project, of each chromosome changed. However, any part of an individual's genes could be mutated. Again, we updated the information about the population to $S_{new\ population}$, each time a mutation was done.

After completing these three essential steps, we returned to selection and kept doing this until our maximum iteration times were reached.

3.3.6 Experiments

GA had been implemented using MATLAB. We conducted a number of experiments. Each time we added certain iteration times to the original ones. Then, we compared the results generated from all experiments and found the best. 2,000 was the initial iteration times, adding constant was 1,500, and the final iteration time was 8,000. Two figures in the next page showed the results from the experiment of 2,000-times iteration and of 8,000-times iterations.

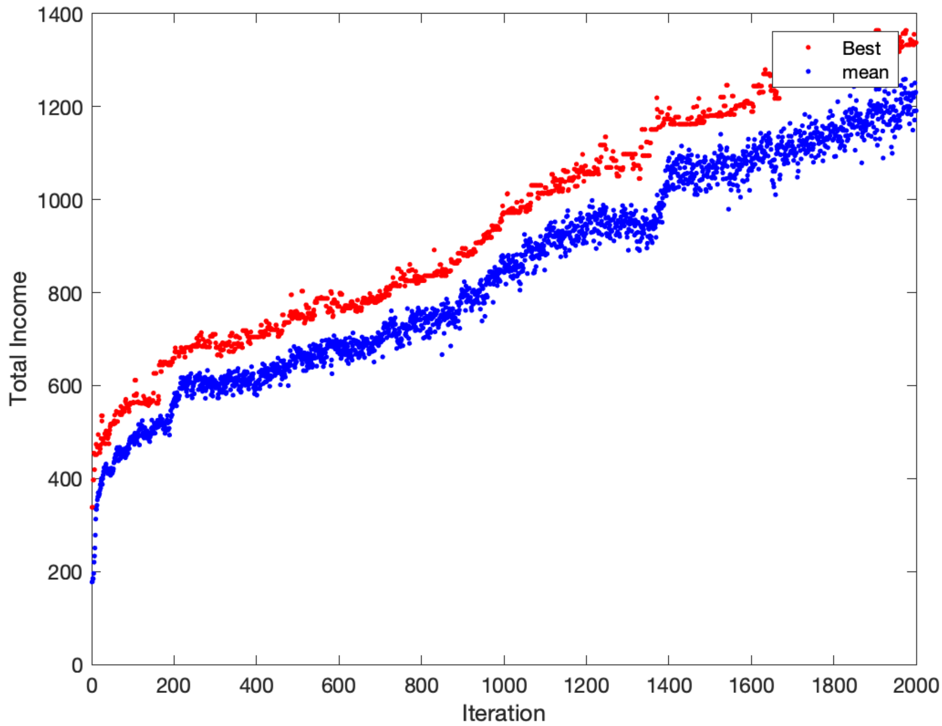

Figure 3. Outcome after 2000 iterations

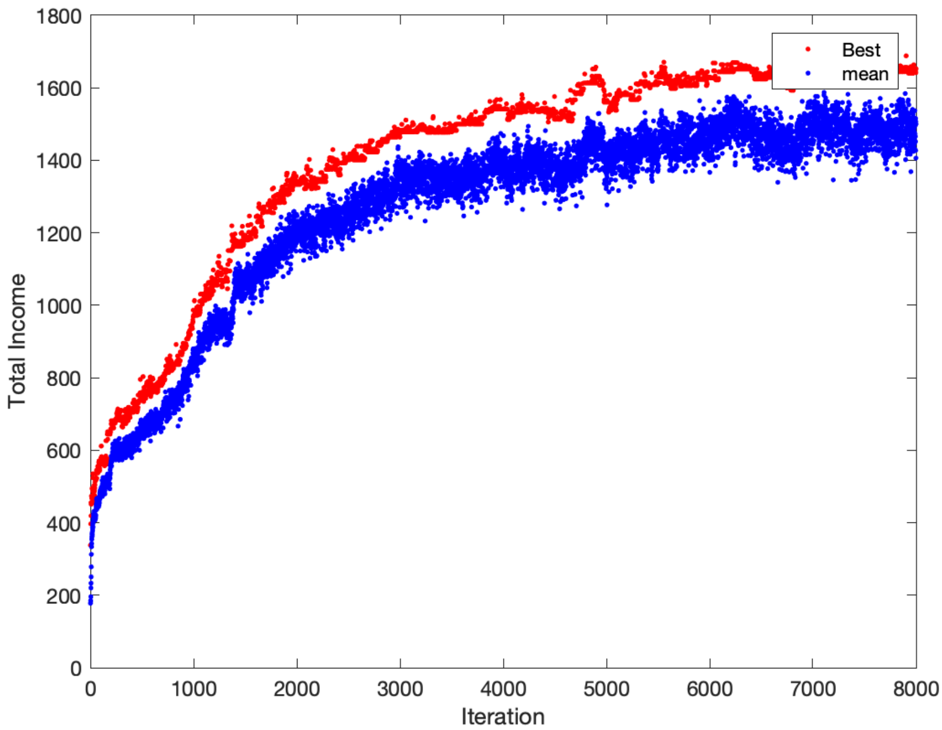

Figure 4. Outcome after 8000 iterations

where x denoted the iteration time and y denotes the objective value(fitness value). The red dots were the optimal objective value for every iteration, representing the best individual; while the blue ones were the mean objective value for every iteration, representing the mean individual.

It was clear to see that when iteration time was 2,000, the optimal objective value was still in an increasing trend. In contrast, when it reached 8,000, the results of optimization began to be steady, indicating the current optimal value was fairly close to the global optimal one.

Thus, we could safely conclude that when iteration times reached 8,000, its corresponding starting-year matrix was able to give us the best plan for the project's startups. The figure below showed the costs in each year given by our algorithm when iteration time was 8,000, which would be used in the fundraising model.

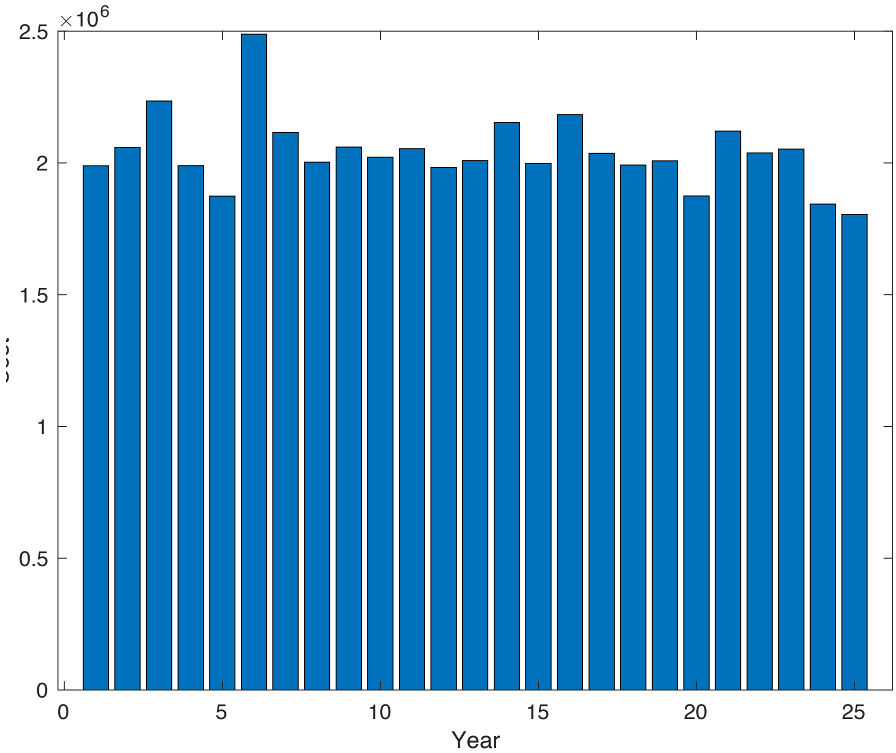

Figure 5. Costs of all programs in each year

It could be shown that the variation in costs across 25 years was relatively small, which means the goal of minimizing equilibrium was achieved very successfully. We stored each year's cost into an array called c with a size of 25, hence c(25) which would be used in later discussion.

4. Model of minimum funds

According to 3.3.6, we had had cost array noted as c(25). The next thing was to calculate the actual, minimum funds required for every year. Since we assumed that how funds changed across years followed a certain linear patter, we set up a linear function Y,

$$Y = m * c + m$$

where m and n were to be determined, and c was the cost array. Therefore, our objective here was to seek for a line Y that was fitting into the cost polyline(line connected by the value in cost array) best, as illustrated below.

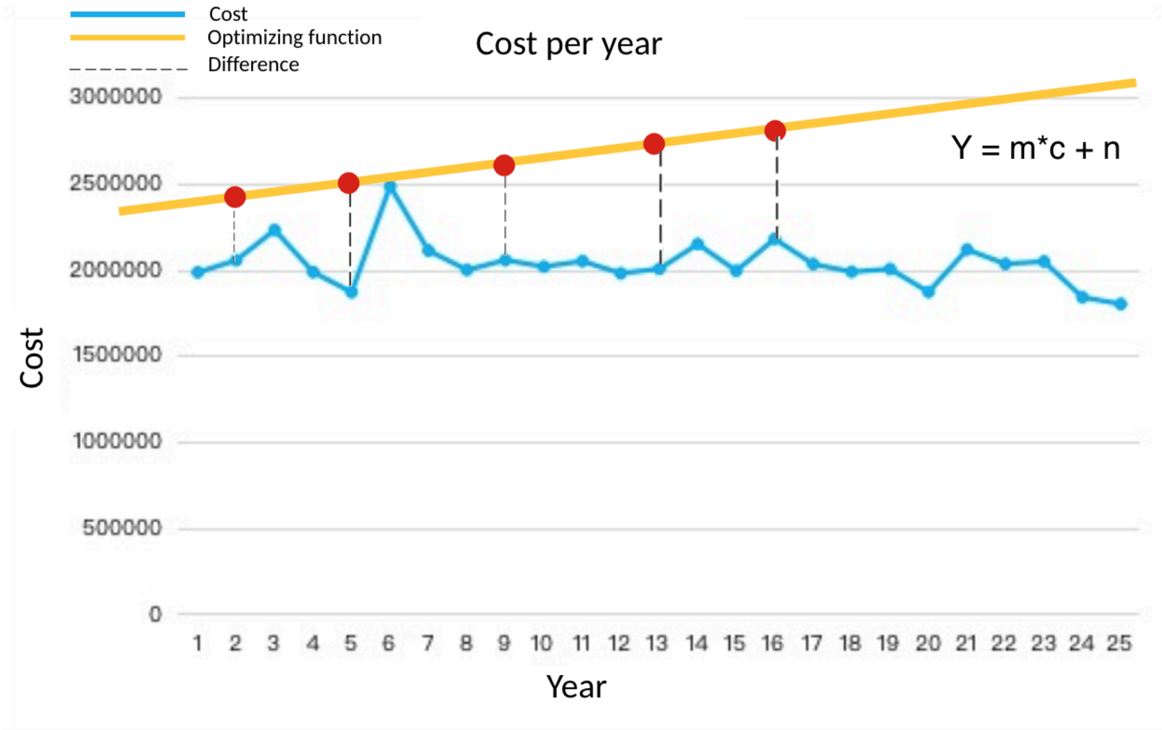

Figure 6. Finding the best-fit line

where the blue line denotes the cost curve; the yellow one denotes the fitting line Y to be optimized, also the actual funds raised in each year; dotted line denoted the distance between the value of the function to be optimized and the value of the cost curve given an independent variable which, in this case, is a certain year.

The goal of this fitting line was to minimize the funds required and at the same time trying to make them more evenly spread out, which meant to minimize the difference line, so our objective function for this line (Z_3) was as follows:

$$minZ_3 = \sum_{t \in T} [mt + n - c(t)]^2$$

where T is time set, and $T = \{1,2,3,...,25\}$, denoting the current year being examined.

Since we should at least raise funds that were numerically equal to costs, he constraint for this line was, thus, that the sum of actual funds from year 1 to year t is greater than or equal to the sum of the costs from year 1 to year t, which can be expressed as:

$$\sum_{t \in T} mt + n \geq \sum_{t \in T} c(t)$$

Therefore, the model as a whole to determine the best-fit line is as the following：

Objective:

$$minZ_3 = \sum_{t \in T} [mt + n - c(t)]^2$$

Constraint:

$$\sum_{t \in T} mt + n \geq \sum_{t \in T} c(t)$$

Putting them into Lingo, we were able to get the value of m and n. It turned out that $m = 0.00000849267$. It was so small to affect the final result that we could safely approximated it to 0. Therefore, the optimized function Y is a horizontal line, and its value is exactly n, which is solved to be 2107286. This means that the amount of funds needed to be raised every year is all the same as 2,107,285 dollars.

5. Model Results

5.1 Best plan for 48 projects to start up

For the best plan for projects' initialization, the genetic algorithm has been iterated 8,000 times to finally find an output that is fairly close to the global optimum. The corresponding decision variable x_{ij} then gives us which year a project starts, shown in the figure below.

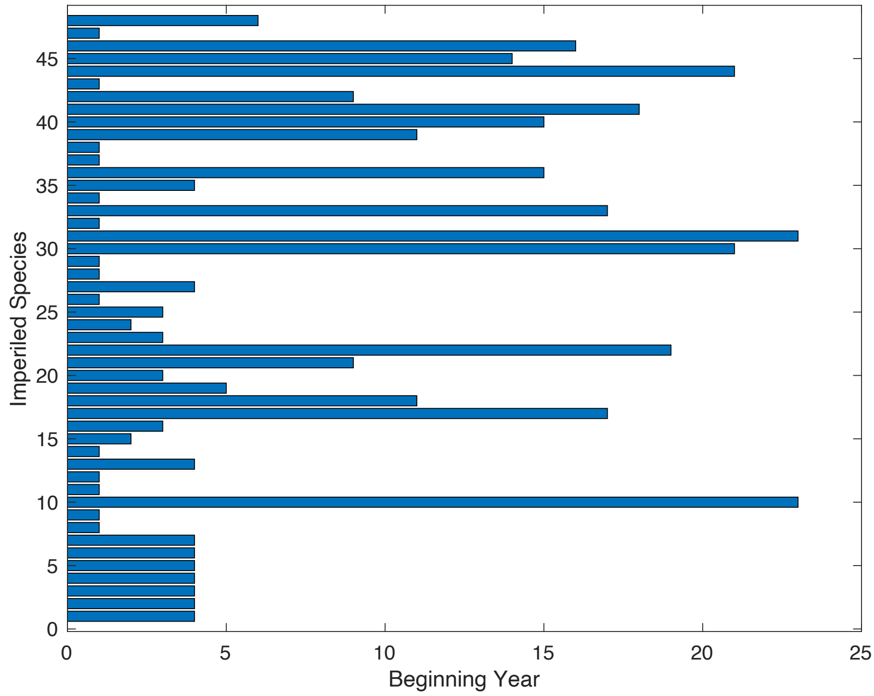

Figure 7. Starting year of each project

From the figure above, we can find that most of projects start before the 15^{th} year. The latest project is the 10^{th} one, beginning in 22^{nd} year, and then the complete schedule for 48 conservation programs is screenshotted from Excel, as shown below:

	Year1	Year2	Year3	Year4	Year5	Year6	Year7	Year8	Year9	Year10	Year11	Year12	Year13	Year14	Year15	Year16	Year17	Year18	Year19	Year20	Year21	Year22	Year23	Year24	Year25
Project1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project2	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project3	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project4	0	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Project5	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project6	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project7	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project8	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project9	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Project11	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project12	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project13	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project14	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project15	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project16	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
Project18	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
Project19	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
Project20	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project21	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project23	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project24	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project25	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project26	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project27	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project28	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project29	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Project31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
Project32	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1
Project34	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0
Project35	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1
Project37	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project38	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project39	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0
Project40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0
Project41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0
Project42	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Project43	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
Project44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1
Project45	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1		
Project46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1
Project47	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0
Project48	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Figure 8. The complete schedule for 48 projects in 25 years

5.2 Minimum funds to be raised

By optimizing the function of actual funds, We found that the best-fit line is a horizontal one with the value 2,107,285, which is the minimal fund FRPCE can raise. Hence, every year will raise exactly $2,107,285. The figure below represents this.

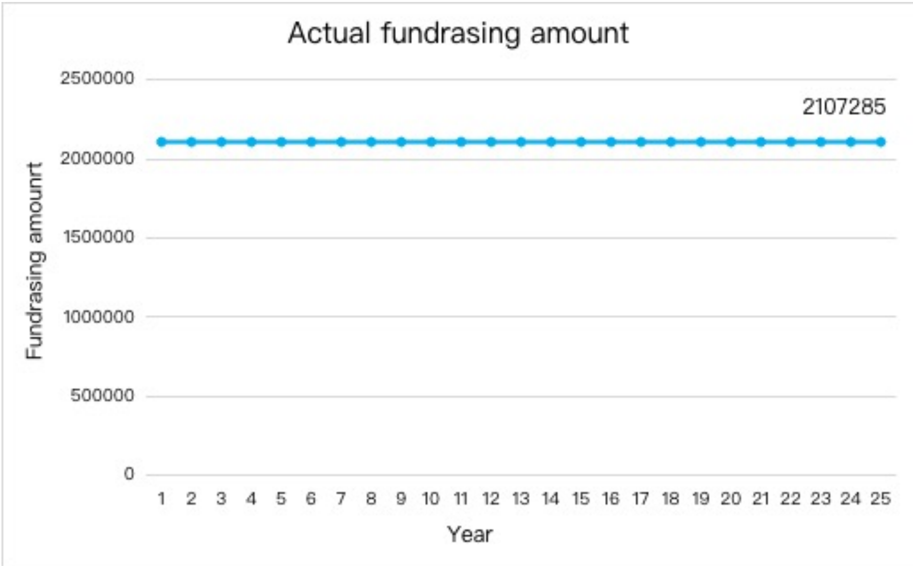

Figure 9. Minimum funds raised every year

5.3 Profits

Even though we are not asked to calculate the profits every year, it is one of our objectives in the modeling part. The result can be shown below.

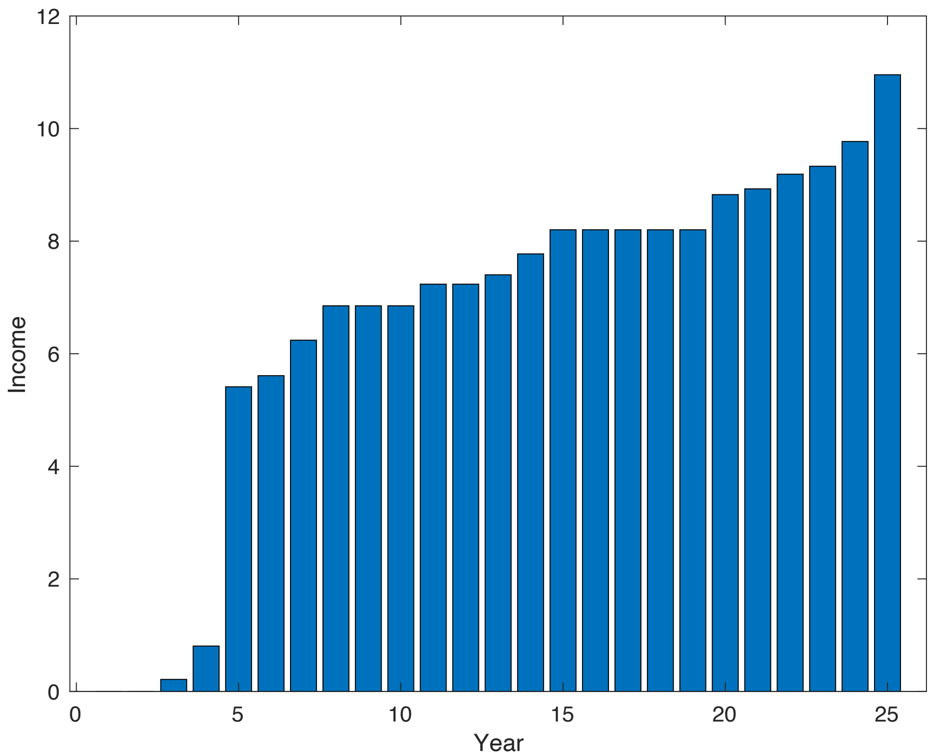

Figure 10. Profits in each year

Noted the figure is produced from data that has been normalized, so the value of y only ranges from 0 to 12. It is clear that before year 3, there is no benefit or income at all. In fact, it is reasonable considering that the fewest duration among 48 projects is exactly three years. Since we have already assumed that the benefits will be generated only after the program has been finished, it would be nothing strange that the incomes brought by these endangered plant conservation projects are supposed to create after year 3 at best. As time goes by, the incomes are continuously growing until year 25 is reached, which is also the year when the benefits are greatest. This makes sense since more and more projects have been done and thus begin generating profits as years move on.

6.Strengths and Weaknesses

6.1 Strengths

1.To form our model, we take two main factors into account, which are the profits brought by each program after it is complete and the equilibrium of funds raised across the years. This allows us to give a very comprehensive perspective for the FRECP board as to how to raise funds better.

2.In the first model to give the best fundraising plan, we do not directly optimize the decision variable x_{ij}, which could have become zero-one programming, and thus an uncreative approach realized by Lingo. Rather, we first think of the goal to be optimized in other way. That is, we create a matrix that is used to store the starting year of each program, which becomes our goal to be optimized. Thus, it becomes integer programming, realized by genetic algorithm, which largely enhances the efficiency of solving this problem.

3.Our model is able to enable the solution very close to the global optima. Concerning that the large dimensions and database our problem has, by applying the genetic algorithm, we can efficiently and largely explore the search space with the help of evolutionary operations such as selection, crossover, and mutation. This makes its population more diverse and thus more immune to be trapped in local optima. In our case, it is nearly impossible to list all possible plans for the fundraising schedule; yet, GA largely and effectively explores the searching space and as the iteration reaches 8,000 times, it has been fairly close to the global optima.

6.2 Weaknesses

1.We do not consider the interest rate. We assumed for a given program, its cost would be the same just as stated in attachment B. If taking interest rates into account, the original cost for a program in a given year will change according to that year's interest rate. Therefore, our model could have been more comprehensive.

7.Conclusion

Overall, we selected two main factors--the benefits produced by programs in a year and the equilibrium of funds needed to be raised--and used two function to denote them. Then, we formed these two functions into a single one that became our objective to be optimized. We desire its value to be as large as possible so that it could provide a better plan. we then designed an implementation of GA which aimed at optimizing the objective function and finding global optima by efficiently exploring as large searching space as possible. Afterwards, the algorithm gave us an output pretty close to global optima, as the growing trend of objective value began steady. Thus, we could adopt the corresponding fundraising plan at this time. Though not required, we gave the costs and profits brought by programs in each year, if FRPCE board decided to use our plan. After finishing the first task, we started minimizing funds by optimizing by optimizing the costs in each year which was given by the algorithm's output. Based on our assumption that the change in funds across 25 years followed a linear patter, we applied lingo to find the best-fit line for the funds to be raised every year, which turned out to be a constant.

Compared to other methods, the plan we suggest for FRPCE based our model was fairly comprehensive and useful, considering how fit GA is for our model and its own effectiveness. We believe that FRPCE would more efficiently raise funds if they adopt our plan.

The end of our solution