Team Control Number

# 10751

Problem Chosen

# B

# 2020
# HiMCM
# Summary Sheet


Summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary
summary summary summary summary summary summary summary summary summary summary summary summary
summary summary summary.

# Contents

# 1 Introduction

It is a global issue that the decrease in biodiversity due to the extinction of endangered plants is a serious problem for the balance of the eco To avoid this, people need to spend money on protecting the plants and animals.

However, in some places like Florida, greatly needing biodiversity conservation, people do not have enough money to protect all of the imperiled species there. They need to find out a proper scheme of funding and protecting to make full use of the funds to get as much benefit as possible.

Different projects of conservation have different cost, benefit, and required time, as shown in the table of threatened plants data (abbreviated as TPD in the following parts of the paper). It is to be determined which projects are selected in the optimal plan according to these factors. After choosing the projects, when to start the projects should also be determined to balance the funds spent over time as possible. To make it clear what is going to be done, the problem is restated mathematically in Section 2. The model and the method to derive the results are to be explained in Section 5.

We will write a non-technical memo in Section 8 to give our proposal according to our model.

# 2 Restatement of the problem

In Florida, many plant species are facing the challenge of extinction. Various plant conservation projects with different budgets and timelines are now being conducted to resolve this problem. However, one major issue is that funds mainly from the annual fundraising are not sufficient to support all projects. Therefore, it is essential to make a plan for the managers who monitor the whole plan to help them precisely decide which project should be included and when and how to allocate their fund. What's notable is that the final goal of this plan is not to simply save as many as plant species as possible. Instead, it should generate greatest benefit meanwhile taking account of different factors, like feasibility of success and the uniqueness of this species. The time spanning to finish the whole plan is also a factor that needs to be seriously considered because as more time being taken to protect one species, the likelihood of another plant dying out shall decrease, finally leading to less benefit. Another factor is the funds spent on these projects. Obviously, a project with median benefit but high expense doesn't fit. The most suitable plan we need to find out should include all variables mentioned and give the best result.

The mathematical description of the problem is given in Section 5. To be specific, the problem is to reach the goal described in Section 5.8.

# 3 List of symbols

To make our model concise and straight forward, a list of symbols are defined as following, as shown in Table 1. The meaning of the symbols are to be explained in detail in the rest of this article. In the table, the symbols in lower case are variables w.r.t. a project, and the symbols in upper case are variables w.r.t. the plan. The difference between a project and the plan is described in Section 5.1.

Note that in this article, all numberings (like $x$ here numbering the projects, and $n$ later numbering the years) start with 0 instead of 1 to meet the convention in computer science.

The symbol $\sum_j^n$ means to sum for $j = 0, 1, \ldots, n-1$. Similar notation is used for other operators like $\prod$, Var, and so on.

If $X$ is a variable related to the plan, $X^*$ denotes the actual value of the variable in the final optimal plan.

# 4 General assumptions

**Assumption 1.** *The funds $F$ are provided off at one time.*

We cannot consider the situation that different funds are raised in different years, which can help us have constraints seperately on different years. It is also constrainted that funds cannot overdraft from other years.

**Assumption 2.** *Conservation cannot be paused.*

Therefore, we regard the effective benefit in conservation as a whole.

**Assumption 3.** *Different conservations can be done simultaneously and do not affect each other.*

In the process of planning, even if some projects fail, it will not affect the subsequent plan. Based on the independence of projects, both their effective benefits and costs can be added directly, rather than coupled together in some nonlinear way. It won't change the cost of the rest of the project due to my execution on any project.

[1]The unit osu is invented to represent the unit of benefit.

Table 1: List of symbols

| Symbol | Domain | Unit | Meaning |
|---|---|---|---|
| $x$ | $\mathbb{N}$ | | The index of projects |
| $D$ | $\mathbb{N}$ | year | Duration of the plan |
| $d_x$ | $\mathbb{N}$ | year | Duration of project $x$ |
| $n$ | $\mathbb{N}$ | year | Time (index of years) |
| $Z$ | $\mathscr{P}(\mathbb{N})$ | | The set of chosen projects |
| $F$ | $[0, +\infty)$ | dollar | Total funds |
| $b_x$ | $[0, +\infty)$ | osu[1] | Benefit of project $x$ |
| $u_x$ | $[0, 1]$ | | Taxonomic uniqueness of project $x$ |
| $s_x$ | $[0, 1]$ | | Feasibility of success of project $x$ |
| $c_{x,n}$ | $[0, +\infty)$ | dollar | Cost of project $x$ in the $n$th year |
| $c_x$ | $[0, +\infty)$ | dollar/year | Total cost of project $x$ |
| $C_n$ | $[0, +\infty)$ | dollar/year | Cost of the plan in the $n$th year |
| $C$ | $[0, +\infty)$ | dollar | Cost of the plan |
| $\beta_x$ | $[0, +\infty)$ | osu | Effective benefit of project $x$ |
| $B$ | $[0, +\infty)$ | osu | Total effective benefit of the plan |
| $T$ | $[0, +\infty)$ | osu/year | Time efficiency of the plan |
| $\xi$ | $\mathbb{N}$ | | The longest project in the plan |
| $a_x$ | $\mathbb{N}$ | year | When project $x$ starts |

**Assumption 4.** *The conservation plan should not change when it is ongoing.*

**Assumption 5.** *The maximization of time efficiency is prior to the balancing of funds spent over time.*

# 5 The model

## 5.1 Projects and the plan

The word **project** is used to represent a conservation of a certain species of plant. According to the TPD, there are some **raw data** attributed to a project $x$, which are the **unique id**, the **taxonomic uniqueness** $u_x$, the **feasibility of success** $s_x$, the **cost** $c_{x,n}$ of the $n$th year. These concepts are explained in Section 5.2.

The word **plan** is used to represent how the projects are going to be executed, including which projects are executed (the set $Z$) and when they should start (the sequence $\{a_x\}$). Deciding the which projects to be executed is to maximize the time efficiency (described in Section 5.5). Deciding when the projects start is to minimize the variance of yearly cost (described in Section 5.6).

## 5.2 Raw attributes of projects

To make the concepts more clear and make it easy to use them later, although the meanings of the raw data are described in the TPD, their meanings should be re-described in a mathematical way.

**Definition 1** (benefit)**.** *The* ***benefit*** *of project $x$ is a real number $b_x \in [0, +\infty)$.*

The benefit is an important indicator indicating how imperiled the species is and how easy the project can be done. However, $b_x$ cannot be directly used to measure how much can we benefit from finishing the project. Another derived concept called **effective benefit** should be used instead, as explained in Section 5.3.

**Definition 2** (taxonomic uniqueness)**.** *The* ***taxonomic uniqueness*** *of project $x$ is a real number $u_x \in [0, 1]$.*

The taxonomic uniqueness is a measure of how the species is unique from other species. It is stipulated that $u_x = 0$ if there exists a species that is exactly the same as the species conserved in project $x$, and $u_x = 1$ if the species conserved in project $x$ does not have any similarities with any other species.

**Definition 3** (feasibility of success)**.** *The* ***feasibility of success*** *of project $x$ is a real number $s_x \in [0, 1]$.*

The meaning of $s_x$ is the probability of the success of project $x$.

**Definition 4** (yearly cost)**.** *The* ***yearly cost*** *of project $x$ is a real number $c_{x,n} \in [0, +\infty)$.*

The meaning of $c_{x,n}$ is cost of project $x$ in the $n$th year.

### 5.3 Effective benefit

Although $b_x$ is defined for a project, hardly can a project lead to so much benefit finally. A new concept should be defined to describe how much can people actually benefit from finishing the project.

**Definition 5** (effective benefit)**.** *The* ***effective benefit*** *of a project is a real number* $\beta_x \in [0, +\infty)$ *defined as*

$$\beta_x := b_x u_x s_x. \tag{1}$$

Here is the explanation of why the effective benefit should be defined as Equation 1.

First, consider the effect of $u_x$. If $u_x = 0$, which means there is a species the same as the conserved species, the conservation is useless because even if the conserved species die out, there are other species taking the place. If $u_x = 1$, which means there are no similar species to the conserved species, the conservation is meaningful and can exert full benefit. From these two cases, it is reasonable to multiply $b_x$ with $u_x$, which means $u_x$ represents the portion of $b_x$ that can be exerted if the project is successfully finished.

Then, consider the effect of $s_x$. It is a probability, so the effect should be considered statistically. Imagine there are many parallel universes, in each of which project $x$ is carried out under the same condition. According to the law of large numbers, the portion of successful executions is the probability $s_x$, exerting the benefit, while the rest does not make any benefit. Then, the mean of all the benefits is $s_x \cdot b_x + \left(1 - s_x\right) \cdot 0 = b_x s_x$.

Combining the two effects above, the formula for effective benefit (Equation 1) can be derived.

From now on, when measuring how much can a project benefit people, $\beta_x$ is used instead of $b_x$.

By summing up the effective benefit for all projects of the plan, the total effective benefit can be defined.

**Definition 6** (total effective benefit)**.** *The* ***total effective benefit*** *of the project is a real number* $\boldsymbol{B} \in [0, +\infty)$ *defined as*

$$\boldsymbol{B} := \sum_{x \in Z} b_x. \tag{2}$$

The $\boldsymbol{B}$ is an important indicator of how much a plan is expected to benefit if the plan is executed.

### 5.4 Duration

The **duration** of a project is the time for which it lasts. It can be defined according to the yearly costs of the project.

**Definition 7** (duration of project)**.** *The* ***duration of project*** $x$ *is a real number* $d_x \in \mathbb{N}$ *defined as*

$$d_x := \min \left\{ m \in \mathbb{N} \middle| \forall n \geq m : c_{x,n} = 0 \right\} \tag{3}$$

According to the TPD, a project must be able to end, which means such $d_x$ must exist.

The duration of the plan should be defined as the time interval between the time when the first project of the plan starts and the time when the last project of the plan ends. However, due to Assumption 5, the duration of the plan should unnecessarily be defined in a so straight-forward way. A better definition is given in Equation 6 in Section 5.5.

### 5.5 Time efficiency

**Definition 8** (time efficiency)**.** *The* ***time efficiency*** *of the plan is a real number* $\boldsymbol{T} \in [0, +\infty)$ *defined as*

$$\boldsymbol{T} := \frac{\boldsymbol{B}}{\boldsymbol{D}}. \tag{4}$$

The time efficiency of the plan is the average benefit gain during a unit of time. It should be maximized to make the plan as efficient as possible, whose reason can be found in Section 5.8.

According to Assumption 5, $\boldsymbol{T}$ should be maximized before we think about how $\boldsymbol{a}_x$ can be adjusted to balance the funds spent over time. To maximize the time efficiency, the duration of the plan is the same as the duration of the project with the largest duration ($\xi$) in the plan, where $\xi$ is the project with largest duration among $\boldsymbol{Z}$

$$\xi := \arg\max_{x \in Z} d_x. \tag{5}$$

Then, when $\boldsymbol{a}_x$ are adjusted to minimize variance of yearly costs, the start time of projects with shorter duration than $\boldsymbol{d}_\xi$ will never be changed to make them start before $\xi$ starts or end after $\xi$ ends because otherwise $\boldsymbol{D}$ increases and thus makes $\boldsymbol{T}$ smaller. Therefore, rather than the straight-forward definition shown in Section 5.4, there is a better definition.

**Definition 9** (duration of the plan)**.** *The* ***duration of the plan*** *is a real number* $D \in \mathbb{N}$ *defined as*

$$D := \max_{x \in Z} d_x. \tag{6}$$

It is obvious that if $C$ stays the same, the larger $T$ is, the better the plan is. In fact, maximizing $T$ is of the most priority. It may be surprising that the most prior goal is not maximizing $\frac{T}{C}$, which is more intuitive. We do not try to maximize $\frac{T}{C}$ because if we did that, it would lead to a trivial result: the plan only contains one project with the minimum $\frac{\beta_x}{d_x c_x}$, which is not what we want to see.

*Proof.* Without loss of generality, assume that there are only 2 plant candidates for the plan, and then the mathematical induction can be used to generalize the conclusion to any number of plant candidates.

Without loss of generality, assume that $d_0 \le d_1$, which means $D = d_1 \ge d_0$.

Case 1: $\frac{\beta_0}{c_0} \le \frac{\beta_1}{c_1}$.

In this case,

$$\frac{\beta_0}{c_0} \le \frac{\beta_0 + \beta_1}{c_0 + c_1} \le \frac{\beta_1}{c_1}.$$

Thus, if the plan consists of both the 2 projects,

$$\frac{T}{C} = \frac{\beta_0 + \beta_1}{c_0 + c_1} \cdot \frac{1}{d_1} \le \frac{\beta_1}{c_1 d_1}.$$

Therefore, the plan consisting of only project 1 is better than that of both projects.

Case 2: $\frac{\beta_0}{c_0} > \frac{\beta_1}{c_1}$.

Similarly to case 1, in this case,

$$\frac{\beta_0}{c_0} > \frac{\beta_0 + \beta_1}{c_0 + c_1} > \frac{\beta_1}{c_1}.$$

Thus, if the plan consists of both the 2 projects,

$$\frac{T}{C} = \frac{\beta_0 + \beta_1}{c_0 + c_1} \cdot \frac{1}{d_1} < \frac{\beta_0}{c_0 d_1} \le \frac{\beta_0}{c_0 d_0}.$$

Therefore, the plan consisting of only project 0 is better than that of both projects.

From the 2 cases above, it can be derived that there always exists a plan with single project having the largest $\frac{T}{C}$ compared to any other plan. □

## 5.6 Variance of yearly costs

First, yearly costs need defining. The yearly cost in the $n$th year of the plan is the sum of the costs of the projects in the $n$th year. It should take into account the different start time of the projects.

**Definition 10** (start time)**.** *The* ***start time*** *of project $x$ is a real number* $a_x \in \mathbb{N}$.

**Definition 11** (yearly costs)**.** *The* ***yearly cost*** *of the plan in the $n$th year is a real number* $C_n \in [0, +\infty)$ *defined as*

$$C_n := \sum_{x \in Z} c_{x, n - a_x}. \tag{7}$$

To measure how balanced the spent funds are distributed through time, the variance of yearly costs $\mathrm{Var}_n^D C_n$ is defined. In Section 5.8, it is going to be minimized to balance the funds spent over time.

It can be proved that minimizing $\mathrm{Var}_n^D C_n$ is the same as minimizing $\sum_n^D C_n^2$.

*Proof.* In this proof, symbol "$\sim$" denotes that two expressions have the same monotonicity.

$$\begin{aligned}
\operatorname*{Var}_n^D C_n &= \frac{1}{D} \sum_n^D \left( C_n - \frac{C}{D} \right)^2 \\
&\sim \sum_n^D \left( C_n^2 - \frac{2C}{D} C_n + \frac{C^2}{D^2} \right) \\
&= \sum_n^D C_n^2 - \frac{2C}{D} \sum_n^D C_n + \sum_n^D \frac{C^2}{D^2} \\
&= \sum_n^D C_n^2 - \frac{C^2}{D} \\
&\sim \sum_n^D C_n^2.
\end{aligned}$$

Thus, minimizing $\operatorname{Var}_n^D C_n$ is the same as minimizing $\sum_n^D C_n^2$. □

## 5.7 Constraints to the plan

First, consider the constraint due to the limitation of funds. To describe the constraint, the total funds and the cost of the plan needs defining.

**Definition 12** (total funds)**.** *The **total funds** is a real number $F \in [0, +\infty)$.*

**Definition 13** (cost of project)**.** *The **cost of project** is a real number $c_x \in [0, +\infty)$ defined as*

$$c_x := \sum_n^\infty c_{x,n}. \tag{8}$$

The sum $c_x$ must converge because all terms following the $d_x$th term are 0.

**Definition 14** (cost of the plan)**.** *The **cost of the plan** is a real number $C \in [0, +\infty)$ defined as*

$$C := \sum_{x \in Z} c_x. \tag{9}$$

Due to Assumption 1, the constraint should be

$$C \le F. \tag{10}$$

There are also constraints resulting from the restrictions for $a_x$ due to the optimization of $T$. According to the discussions in Section 5.5, any projects in the plan should not start before the longest project start or end after the longest project end. Let the start time of the longest project in the plan be the 0th year, and then we have constraints for the start time.

$$\forall x \in Z : a_x \in \left[0, D - d_x\right). \tag{11}$$

## 5.8 The goal of the model

The goal of the model is to

1. Decide proper $Z$ to make the plan have the maximum $T$,
2. Decide the minimum $F$ to satisfy step 1 under the constraint Equation 10,
3. Decide the $a_x$ for each chosen project in step 1 to minimize $\operatorname{Var}_n^D C_n$ under the constraint Equation 11.

In this way, the derived plan can meet the requirements described in Section 2.

To reach the goal, we need to develop some approaches based on the model, which are described in Section 6.

# 6 The approach

## 6.1 Find the maximum effective benefit that can be loaded into the package with given funds

We turn the question into a question with $n$ items and $a$ backpack with capacity $F$. The cost of item $x$ is $C[x]$ and the value is $\beta[i]$. Find out which items are loaded into the backpack to maximize the total value. We used the recursive algorithm. Here is our procedure

Firstly, Convert the data in TPD into an array.

Secondly, calculate the value of each item under unit weight $K[x] = \frac{\beta[x]}{C[x]}$, and sort them from large to small according to their value. If the they are same, sort them according to their weight.

Thirdly, we set a sequence $ok[x]$ to store whether the corresponding items are loaded or not, 1 means yes, 0 means no.(the initial values are all 0).

Define the function $v = fn(F, POS)$, input: $F$, that is, the current maximum weight that can be carried, and $POS$ refers to the number of positions in the array to calculate.
The algorithm of recursive function is shown as follows:
1. Judge whether the weight of the object at $POS$ is bigger than in $F$.
2. If True, give up the item in the position, and the function returns $fn(F, POS + 1)$. If it is the last one, return 0 directly.
3. If Fulse, continue with the following process.
4. If put in,
$v_1 = C[pos] + fn(F - C[pos], pos + 1)$
If you don't put it in,
$v_2 = fn(F, pos + 1)$
If $v_1 >= v_2$, execute put: Mark $ok[POS]$ as 1, and the function returns $v_1$
If $v_1 < v_2$, the function returns $v_2$.

## 6.2 Minimize the variance

## 6.3 The maximum time efficiency of the program with a given funds

On the basis of the first program, remove the knapsack candidates from the project with the largest time, run again and repeat. Then we can find the plan with the maximum value of the $\frac{\beta_x}{t}$

## 6.4 Minimize the funds when the time efficiency maximum

On the basis of the Chinese version, the funds are reduced to when the $min\beta_x$ can no longer be included in the plan.

## 6.5 Statement

### 6.5.1 Why not use the classical DP method to solve the knapsack problem?

Hence, we figured out that we can use the recursive and the Cache to have a smaller time complexity.

### 6.5.2 Why do we need to round up money?

The data in the database is float numbers, which cannot be the key of the array so that we cannot use the normal knapsack algorithm(dp) to calculate the best knapsack. Hence, the knapsack problem with a float size cannot be solved.

# 7 Pros and cons

Our model have some pros and cons in solving the problem described in 1. The pros are

1. Pro 1.
2. Pro 2.
3. Pro 3.

The cons are

1. Con 1.
2. Con 2.
3. Con 3.

# 8 The memo

Table 2 shows the schedule of the optimal plan. The $n$th column denotes the $n$th year. The last row sums the costs of the projects, representing the fundraising schedule.

The total cost of the plan is $11010118.27.

Table 2: Schedule of executing the projects

<table>
<tr><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-502</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-436</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-536</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-183</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-480</td></tr>
<tr><td></td><td colspan="4">1-Flowering Plants-135</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-481</td></tr>
<tr><td colspan="5">1-Flowering Plants-176</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-475</td></tr>
<tr><td colspan="5">1-Flowering Plants-546</td></tr>
<tr><td colspan="5">1-Flowering Plants-558</td></tr>
<tr><td colspan="5">1-Flowering Plants-553</td></tr>
<tr><td colspan="5">1-Flowering Plants-442</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-537</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-548</td></tr>
<tr><td colspan="5">1-Flowering Plants-426</td></tr>
<tr><td colspan="5">1-Flowering Plants-452</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-455</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-133</td></tr>
<tr><td colspan="5">1-Flowering Plants-168</td></tr>
<tr><td colspan="5">1-Flowering Plants-476</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-137</td></tr>
<tr><td colspan="5">1-Flowering Plants-485</td></tr>
<tr><td colspan="5">1-Flowering Plants-528</td></tr>
<tr><td colspan="5">1-Flowering Plants-520</td></tr>
<tr><td colspan="5">1-Flowering Plants-514</td></tr>
<tr><td colspan="5">1-Flowering Plants-517</td></tr>
<tr><td></td><td></td><td colspan="3">1-Flowering Plants-529</td></tr>
<tr><td colspan="5">1-Flowering Plants-557</td></tr>
<tr><td colspan="5">1-Flowering Plants-179</td></tr>
<tr><td colspan="5">1-Flowering Plants-530</td></tr>
<tr><td colspan="5">1-Flowering Plants-440</td></tr>
<tr><td colspan="5">1-Flowering Plants-513</td></tr>
<tr><td colspan="5">1-Flowering Plants-524</td></tr>
<tr><td colspan="5">1-Flowering Plants-508</td></tr>
<tr><td colspan="5">1-Lichens-567</td></tr>
<tr><td>$2600492.21</td><td>$2555863.21</td><td>$3024886.35</td><td>$1488407.44</td><td>$1340469.06</td></tr>
</table>

# A The program used to decide $Z$

```
require_relative 'plant'

BEGIN_TIME = Time.now
SORTED_TPD = TPD.sort

def puts_plants_set taken
  puts TPD.sum('') { taken.include?(_1) ? '|' : ' ' }
end

def max_effective_benefit_recursive remaining_funds, pos
  input = [remaining_funds, pos]
  return @cache[input] if @cache[input]
  return @cache[input] = [0, 0] if pos >= @plants.size || (plant = @plants[pos]).rounded_cost >
   ↪ remaining_funds
  benefit_if_not_take, set_if_not_take = max_effective_benefit_recursive remaining_funds, pos +
   ↪ 1
  benefit_if_take, set_if_take = max_effective_benefit_recursive remaining_funds -
   ↪ plant.rounded_cost, pos + 1
  benefit_if_take += plant.integerized_effective_benefit
  set_if_take = set_if_take.add plant
  @cache[input] = benefit_if_take > benefit_if_not_take ? [benefit_if_take, set_if_take] :
   ↪ [benefit_if_not_take, set_if_not_take]
end

def max_effective_benefit_under funds
  max_effective_benefit_recursive funds, 0
end

def max_time_efficiency_under funds
  @plants = SORTED_TPD.clone
  max_time_efficiency, max_time_efficiency_taken = 0r, []
  loop do
    @cache = {}
    max_effective_benefit, taken = max_effective_benefit_under funds
    plant_to_be_removed, index_to_be_removed = @plants.each_with_index.max_by { |plant, i|
     ↪ taken.include?(plant) ? plant.duration : 0 }
    puts_plants_set taken
    break unless plant_to_be_removed && taken.include?(plant_to_be_removed)
    new_time_efficiency = max_effective_benefit.quo plant_to_be_removed.duration
    max_time_efficiency, max_time_efficiency_taken = new_time_efficiency, taken if
     ↪ new_time_efficiency > max_time_efficiency
    @plants.delete_at index_to_be_removed
  end
  [max_time_efficiency, max_time_efficiency_taken]
end

def max_efficiency_under
  funds = TPD.sum &:rounded_cost
  max_efficiency, max_efficiency_taken, max_efficiency_funds = 0r, 0, funds
  loop do
    puts "Funds: #{funds}"
    max_time_efficiency, taken = max_time_efficiency_under funds
    break if taken.empty?
    cost = taken.sum &:rounded_cost
    new_efficiency = max_time_efficiency / cost
    max_efficiency, max_efficiency_taken, max_efficiency_funds = new_efficiency, taken, cost if
     ↪ new_efficiency > max_efficiency
    funds = cost - 1
```

```
  end
  [max_efficiency, max_efficiency_taken, max_efficiency_funds]
end

(0..510000).step 3000 do |funds|
  IO.write 'output.txt',
           "#{funds}: #{max_time_efficiency_under(funds).join ' '}\n",
           mode: 'a'
end

puts "Time passed: #{Time.now - BEGIN_TIME}"
```

The `plant.rb` file defines the data structure of a plant and imports the constant `TPD`.

# B The program used to decide $\{a_x\}$

```
require_relative 'plant'

BEGIN_TIME = Time.now

def Array.product array, *arrays
  array.to_a.to_enum :product, *arrays.map(&:to_a)
end

TAKEN = 14796153683959
TAKEN_ARRAY = TAKEN.to_a
DURATION = TAKEN.max_by(&:duration).duration

FACTOR = 100.0 / TAKEN.reduce(1) { _1 * (DURATION - _2.duration + 1) }

result = Array.product(*TAKEN.map { 0..DURATION - _1.duration }).min_by.with_index do |starts,
 ↪ i|
  print "\r#{(i * FACTOR).round}%"
  DURATION.times.sum do |year|
    TAKEN_ARRAY.zip(starts).sum do |plant, start|
      year >= start ? plant.costs[year - start] || 0 : 0
    end ** 2
  end
end

puts "\r100%"
p result
puts "Time passed: #{Time.now - BEGIN_TIME}s"
```

# C The program used to plot the graph of max $T$ related to $F$

```
from scanf import scanf
import matplotlib.pyplot as plt
import numpy as np

fig, ax = plt.subplots()
funds_list = []
max_time_efficiency_list = []
with open('output.txt') as f:
  for line in f:
    funds, numerator, denominator, _ = scanf("%d: %d/%d %d", line)
    funds_list.append(funds * 1e2)
    max_time_efficiency_list.append(numerator / denominator * 1e-6)
ax.ticklabel_format(style='sci', scilimits=(-1, 1), useMathText=True)
```

```
ax.plot(funds_list, max_time_efficiency_list)
ax.plot(11100000, 8190088e-6/5, '.')
plt.show()
```

The program reads the result of the program in Appendix A.

# D Table of max $T$ related to $F$ for $F < F^*$

If $F$ is given, under the constraints given by Equation 10, $T$ can be maximized by properly deciding $Z$. The max $T$ related to $F$ for $F < F^*$ is given in Table 3. For $F \geq F^*$, max $T$ is the same as $T^*$. Data in Table 3 are calculated by the program in Appendix A.

Table 3: Table of max $T$ related to $F$ before $T$ is maximized

| $F$ | max $T$ |
|---|---|
| $0.00 | 0.00 |
| $300000.00 | 435567.00 |
| $600000.00 | 622541.67 |
| $900000.00 | 727932.67 |
| $1200000.00 | 810146.67 |
| $1500000.00 | 878511.67 |
| $1800000.00 | 909344.67 |
| $2100000.00 | 937911.67 |
| $2400000.00 | 937911.67 |
| $2700000.00 | 937911.67 |
| $3000000.00 | 957915.40 |
| $3300000.00 | 1010095.00 |
| $3600000.00 | 1058935.00 |
| $3900000.00 | 1111114.60 |
| $4200000.00 | 1160443.00 |
| $4500000.00 | 1203976.60 |
| $4800000.00 | 1253305.00 |
| $5100000.00 | 1291070.20 |
| $5400000.00 | 1328921.20 |
| $5700000.00 | 1373425.00 |
| $6000000.00 | 1403508.00 |
| $6300000.00 | 1430026.80 |
| $6600000.00 | 1456288.20 |
| $6900000.00 | 1473947.80 |
| $7200000.00 | 1497505.20 |
| $7500000.00 | 1522926.40 |
| $7800000.00 | 1533631.60 |
| $8100000.00 | 1556091.40 |
| $8400000.00 | 1571313.80 |
| $8700000.00 | 1590420.80 |
| $9000000.00 | 1603152.20 |
| $9300000.00 | 1622259.20 |
| $9600000.00 | 1622259.20 |
| $9900000.00 | 1622259.20 |
| $10200000.00 | 1628825.20 |
| $10500000.00 | 1631451.60 |
| $10800000.00 | 1631451.60 |
| $11100000.00 | 1638017.60 |

## References

[1] Ref 1.