ГУАП

КАФЕДРА № 43

ОТЧЁТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| д-р техн. наук, доцент |  |  |  | С. И. Колесникова |
| должность, уч. Степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| ОТЧЁТ О ЛАБОРАТОРНОЙ РАБОТЕ №1 | | | | | |
| ПРИНЦИПЫ ДИНАМИЧЕСКОГО ПРОГРАММИРОВАНИЯ В СИСТЕМНОМ АНАЛИЗЕ | | | | | |
| по дисциплине: СИСТЕМНЫЙ АНАЛИЗ | | | | | |
|  | | | | | |
| РАБОТУ ВЫПОЛНИЛ | | | | | |
| СТУДЕНТ ГР. | 4330М |  |  |  | А.А. Кинько |
|  |  |  | подпись, дата |  | инициалы, фамилия |

Санкт-Петербург 2023

**Цель работы**

Целью работы является освоение метода динамического программирования применительно к анализу практических ситуаций.

**Текст задания**

Согласно варианту №8, для задачи оптимизации прироста продукции:

Планируется распределение начальной суммы млн. р. Между четырьмя предприятиями некоторого объединения. Средства выделяются только в размерах кратных млн. р. Функции прироста продукции от вложенных средств на каждом предприятии заданы таблично (таблица 1). Требуется так распределить вложения между предприятиями, чтобы общий прирост продукции (в млн. р.) был максимальным. Решить задачу на основе функционального уравнения Беллмана.

*Таблица 1. Прирост продукции на предприятиях*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Вкладываемые средства | Функции прироста продукции на предприятии | | | |
|  |  |  |  |
| 400 | 0 | 10 | 15 | 13 | 14 |
| 80 | 13 | 20 | 17 | 16 |
| 160 | 16 | 22 | 21 | 23 |
| 240 | 21 | 25 | 26 | 25 |
| 320 | 25 | 30 | 28 | 27 |
| 400 | 25 | 32 | 30 | 32 |

**Ход решения**

Рассмотрим распределение средств для случая с одним предприятием. Пусть – функция, обозначающая максимальный прирост для вложенных средств и количестве предприятий .

Очевидно, что максимальная прибыль для одной фирмы будет равна приросту этой фирмы:

Теперь рассмотрим два предприятия. Введем значение - вкладываемые средства во вторую фирму. Тогда при общей сумме первая фирма получит . Для получения максимальной прибыли необходимо проверить все , кратные 80 (согласно постановке задачи). Тогда формула принимает вид:

Для трех предприятий формула преображается до:

Несложно заметить, что для расчета суммарного максимального прироста для любого количества фирм формула формализуется в:

Просчитаем вручную все значения функции и внесем их в таблицу.

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

,

*Таблица 2. Максимальная прибыль*

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | Вкладываемые средства | Сумма максимальной прибыли | | | |
|  |  |  |  |
| 400 | 0 | 10 | 25 | 38 | - |
| 80 | 13 | 30 | 43 | - |
| 160 | 16 | 32 | 47 | - |
| 240 | 21 | 36 | 51 | - |
| 320 | 25 | 41 | 56 | - |
| 400 | 25 | 45 | 59 | 74 |

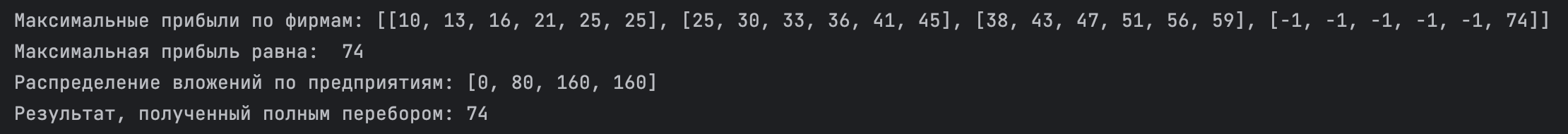
Найдем, какие значения дают наибольшее значение для – , то есть Тогда в четвертую фирму нужно вложить 160 млн. р. Аналогично для , то есть . В третью фирму нужно вложить 160 млн. р. Для , то есть Тогда во вторую фирму нужно вложить 80 млн. р., а в первую не вкладывать ничего.

**Программная реализация.**

Листинг программы (Python 3.6):

profit = [[10, 15, 13, 14],  
 [13, 20, 17, 16],  
 [16, 22, 21, 23],  
 [21, 25, 26, 25],  
 [25, 30, 28, 27],  
 [25, 32, 30, 32]]  
  
profit\_transparent =[[10, 13, 16, 21, 25, 25],  
 [15, 20, 22, 25, 30, 32],  
 [13, 17, 21, 26, 28, 30],  
 [14, 16, 23, 25, 27, 32]]  
  
calculated\_profit = [[-1 for \_ in range(6)] for \_ in range(4)]  
  
max\_num = -1  
def get\_factory\_profit(factory\_num, money, money\_factory\_num, list\_of\_max):  
 if factory\_num == 0:  
 if calculated\_profit[0][money] != -1:  
 return calculated\_profit[0][money]  
 else:  
 calculated\_profit[0][money] = profit[money][0]  
 return calculated\_profit[0][money]  
 else:  
 if calculated\_profit[factory\_num][money] != -1:  
 return calculated\_profit[factory\_num][money]  
 if money < money\_factory\_num:  
 return -1  
 else:  
 list\_of\_max.append(profit[money\_factory\_num][factory\_num] +  
 get\_factory\_profit(factory\_num - 1, money - money\_factory\_num, 0, []))  
 get\_factory\_profit(factory\_num, money, money\_factory\_num + 1, list\_of\_max)  
 max\_profit = max(list\_of\_max)  
 global max\_num  
 max\_num = list\_of\_max.index(max\_profit)  
 if calculated\_profit[factory\_num][money] == -1:  
 calculated\_profit[factory\_num][money] = max\_profit  
 return max\_profit  
  
maximum\_profit = get\_factory\_profit(3, 5, 0, [])  
print("Максимальные прибыли по фирмам:", calculated\_profit)  
print("Максимальная прибыль равна: ",maximum\_profit)  
  
answer = []  
for x1 in range(len(profit)):  
 for x2 in range(len(profit)):  
 for x3 in range(len(profit)):  
 for x4 in range(len(profit)):  
 if profit\_transparent[0][x1] + profit\_transparent[1][x2] + \  
 profit\_transparent[2][x3] + profit\_transparent[3][x4] == maximum\_profit and x1 + x2 + x3 + x4 == len(profit) - 1:  
 answer = [x \* 80 for x in [x1, x2, x3, x4]]  
print("Распределение вложений по предприятиям:", answer)  
  
max\_num = -1  
for x1 in range(len(profit)):  
 for x2 in range(len(profit)):  
 for x3 in range(len(profit)):  
 for x4 in range(len(profit)):  
 if x1 + x2 + x3 + x4 == len(profit) - 1:  
 if profit\_transparent[0][x1] + profit\_transparent[1][x2] + \  
 profit\_transparent[2][x3] + profit\_transparent[3][x4] > max\_num:  
 max\_num = profit\_transparent[0][x1] + profit\_transparent[1][x2] + \  
 profit\_transparent[2][x3] + profit\_transparent[3][x4]  
print("Результат, полученный полным перебором:", max\_num)

**Результат работы программы.**

****

**Выводы.**

В ходе выполнения данной лабораторной работы были получены теоретические сведения о динамическом программировании, а также решена задача распределения средств по предприятиям на основе функционального уравнения Беллмана, в результате чего было получено решение: для получения максимальной прибыли (74 млн. р.) необходимо вложить: в первую фирму – 0 млн. р.; во вторую – 80 млн. р.; в третью – 160 млн. р.; в четвертую – 160 млн. р.