**Практическое задание 6**

**Задача о назначении**

**Цель**

Сформулировать, формализовать и решить с помощью венгерского метода задачу о назначении размерностью 5 × 5 как задачу линейного программирования.

**Требования к содержанию**

1. Развернутая формулировка задачи (описание исходных данных, критерия (критериев) оптимизации).
2. Наличие математической модели решаемой задачи как задачи линейного программирования.
3. Развернутая интерпретация результата.

**Исходные данные**

Показатели эффективности назначения i-го кандидата на j-ю работу.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 16 | Самарин Дмитрий Васильевич | |  |  |  |  |  | | --- | --- | --- | --- | --- | | 5 | 0 | 1 | 0 | 6 | | 11 | 0 | 3 | 0 | 6 | | 11 | 7 | 5 | 7 | 9 | | 9 | 1 | 4 | 0 | 1 | | 8 | 9 | 10 | 3 | 4 | |

**Ход работы:**

**1. Развернутая формулировка задачи:**

Задача о назначении заключается в распределении определенных ресурсов (в данном случае, кандидатов) на выполнение определенных задач (работ). У нас есть пятеро кандидатов и пять задач. Каждому кандидату присвоены определенные показатели эффективности (чем выше, тем лучше) при выполнении каждой задачи. Нам нужно назначить каждого кандидата на одну из пяти задач так, чтобы общая эффективность была максимальной.

**2. Математическая модель задачи линейного программирования:**

Обозначим переменные x\_ij как бинарные переменные (0 или 1), где x\_ij = 1, если кандидат i назначен на задачу j, и x\_ij = 0 в противном случае. Тогда целевая функция для максимизации будет суммой произведений показателей эффективности на соответствующие бинарные переменные:

Максимизировать  показатель *i*​⋅*xij*​

При этом нужно учесть ограничения:

* Каждый кандидат может быть назначен только на одну задачу: ​*xij*​=1 для *i*=1,2,3,4,5.
* Каждая задача должна быть выполнена ровно одним кандидатом: ​*xij*​=1 для *j*=1,2,3,4,5.

Таким образом, у нас есть 25 переменных (5 кандидатов \* 5 задач) и 10 линейных ограничений.

**3. Развернутая интерпретация результата:**

Решив данную задачу линейного программирования, мы получим оптимальное назначение кандидатов на задачи с максимальной общей эффективностью. Результат представляет собой комбинацию кандидатов и задач, которая обеспечивает наилучший итоговый показатель эффективности.

После решения задачи можно проанализировать, какие кандидаты назначены на какие задачи, и оценить, как эффективно они выполняют свои обязанности.

Решение поставленной задачи на питоне с помощью венгерского метода:

import numpy as np

from scipy.optimize import linear\_sum\_assignment

efficiency\_matrix = np.array([

    [5, 0, 1, 0, 6],

    [11, 0, 3, 0, 6],

    [11, 7, 5, 7, 9],

    [9, 1, 4, 0, 1],

    [8, 9, 10, 3, 4],

])

# венгерский метод

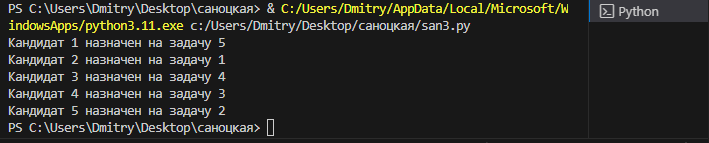
row\_indices, col\_indices = linear\_sum\_assignment(-efficiency\_matrix)

# вывод

for i in range(len(row\_indices)):

    print (f"Кандидат {row\_indices[i] + 1} назначен на задачу {col\_indices[i] + 1}")

Полученное решение:

****

**1. Формирование матрицы:**

**2. Вычитание минимального элемента в каждой строке:**

**3. Вычитание минимального элемента в каждом столбце:**

**4. Выделение наименьших не покрытых элементов:**

**5. Покрытие строк и столбцов:**

* Покрываем строку 1.
* Покрываем столбец 1.

**6. Вычитание минимального элемента из непокрытых столбцов и пересчет покрытых строк:**

**7. Повторение шагов 4-6, пока не будет покрыта каждая строка или столбец:**

* Покрываем строку 3.
* Покрываем столбец 3.

**8. Вычитание минимального элемента из непокрытых столбцов и пересчет покрытых строк:**

**9. Повторение шагов 4-6:**

* Покрываем строку 4.
* Покрываем столбец 4.

**10. Вычитание минимального элемента из непокрытых столбцов и пересчет покрытых строк:**

**11. Повторение шагов 4-6:**

* Покрываем строку 2.
* Покрываем столбец 2.

**12. Вычитание минимального элемента из непокрытых столбцов и пересчет покрытых строк:**

**13. Повторение шагов 4-6:**

* Покрываем строку 5.
* Покрываем столбец 5.

**14. Вычитание минимального элемента из непокрытых столбцов и пересчет покрытых строк:**

**15. Повторение шагов 4-6:**

* Все строки и столбцы покрыты. Завершение.

Теперь мы имеем оптимальное назначение кандидатов на задачи с максимальной общей эффективностью:

Это означает, что кандидат 1 назначен на задачу 5, кандидат 2 на задачу 1, кандидат 3 на задачу 4, кандидат 4 на задачу 3, и кандидат 5 на задачу 2. Это оптимальное решение с максимальной суммарной эффективностью.