

Министерство образования и науки Российской Федерации

Алтайский государственный технический университет

им. И.И. Ползунова

Отчет

о лабораторной работе №6

по теме: «Ликвидация склада» (задача из финального этапа
международной олимпиады по программированию 1С)

предмет: Математические модели и методы поддержки принятия решений

Выполнил

Студент группы 8ПИЭ-41

Хартов А.Е.

Проверил

Блем А.Г.

Барнаул 2024

Условие:

На предприятии 3 склада. Склад № 3 ликвидируется. Товары с 3 склада нужно перевезти на 1 и (или) 2 склады таким образом, чтобы в результате стоимость товаров (в сумме) на 1 и 2 складах была по возможности одинаковой. При этом перемещать товары между 1 и 2 складами нельзя.

Исходные данные заданы в таблице

№ товара	Цена товара (за шт.,руб.)	Наличие товара на складах		
		Склад 1	Склад 2	Склад 3
1	158	12	18	7
2	235	5	11	8
3	96	18	13	13
4	515	13	20	6
5	485	20	10	21
6	205	12	12	16
7	674	19	15	24
8	615	31	17	9
9	273	6	22	10
10	810	15	14	15
11	426	18	9	18
12	925	22	11	14
13	78	9	17	6
14	324	23	24	19
15	416	16	12	11

Решение методом Монте-Карло:

```
import numpy as np

# Данные из таблицы
data = {
    "item_number": list(range(1, 16)),
    "price_per_unit": [158, 235, 96, 515, 485, 205, 674, 615, 273,
810, 426, 925, 78, 324, 416],
    "stock_1": [12, 5, 18, 13, 20, 12, 19, 31, 6, 15, 18, 22, 9,
23, 16],
    "stock_2": [18, 11, 13, 20, 10, 12, 15, 17, 22, 14, 9, 11, 17,
24, 12],
    "stock_3": [7, 8, 13, 6, 21, 16, 24, 9, 10, 15, 18, 14, 6, 19,
11]
}

# Проверка целостности данных
assert len(data["price_per_unit"]) == len(data["stock_1"]) ==
len(data["stock_2"]) == len(data["stock_3"]), \
    "Данные в словаре должны быть одинаковой длины."

# Стоимости на складах 1 и 2 (без учета склада 3)
cost_1 = sum(np.array(data["price_per_unit"]) *
np.array(data["stock_1"]))
cost_2 = sum(np.array(data["price_per_unit"]) *
np.array(data["stock_2"]))

# Предварительные стоимости товаров на складе 3
prices_stock_3 = np.array(data["price_per_unit"]) *
np.array(data["stock_3"])

# Метод Монте-Карло
def monte_carlo_distribution(data, iterations=100000):
    best_diff = float("inf")
    best_distribution = None

    for _ in range(iterations):
        # Случайное распределение товаров со склада 3
        distribution = np.random.choice([1, 2],
size=len(prices_stock_3))

        # Расчет новых стоимостей складов
        cost_1_temp = cost_1 + sum(prices_stock_3[distribution ==
1])
        cost_2_temp = cost_2 + sum(prices_stock_3[distribution ==
2])

        # Разница в стоимости между складами 1 и 2
        diff = abs(cost_1_temp - cost_2_temp)
        if diff < best_diff:
```

```

        best_diff = diff
        best_distribution = distribution

    return best_distribution, best_diff

# Выполнение метода Монте-Карло
optimal_distribution, minimal_difference =
monte_carlo_distribution(data)

# Результат распределения
distribution_result = {
    "item_number": data["item_number"],
    "to_stock_1": [data["stock_3"][i] if optimal_distribution[i]
== 1 else 0 for i in range(len(data["stock_3"]))],
    "to_stock_2": [data["stock_3"][i] if optimal_distribution[i]
== 2 else 0 for i in range(len(data["stock_3"]))],
}

# Вывод результатов
print("Оптимальное распределение товаров:")
print("Товары на склад 1:", distribution_result["to_stock_1"])
print("Товары на склад 2:", distribution_result["to_stock_2"])
print("Минимальная разница в стоимости:", minimal_difference)

```

Результат:

```

Оптимальное распределение товаров:
Товары на склад 1: [0, 8, 13, 0, 0, 0, 0, 0, 0, 15, 18, 0, 0, 19, 11]
Товары на склад 2: [7, 0, 0, 6, 21, 16, 24, 9, 10, 0, 0, 14, 6, 0, 0]
Минимальная разница в стоимости: 2

```