

Московский государственный университет имени М.В. Ломоносова
Факультет вычислительной математики и кибернетики
Кафедра математических методов прогнозирования



Курс «Алгоритмика»

Первая лабораторная работа

Задача об Арбитраже

Работу выполнил
студент **Сюй Минчуань**

Содержание

1	Введение в задачу об Арбитраже	3
1.1	Постановка задачи	3
1.2	Описание данных	3
2	Метод решения, теоретический анализ и программная реализация алгоритма	4
2.1	Предварительный анализ задачи	4
2.2	Описание метода решения	4
2.3	Анализ алгоритма	5
2.4	Описание программной реализации	5
2.5	Результат поставленной задачи	5

1 Введение в задачу об Арбитраже

1.1 Постановка задачи

Арбитражем называется использование разницы в курсах обмена валют для получения прибыли. Например, в течение короткого периода времени за 1 доллар США можно купить 0.75 фунта стерлингов, за 1 фунт стерлингов – 2 австралийских доллара, а за 1 австралийский доллар – 0.7 доллара США. То есть осуществление такой сделки для доллара США принесёт прибыль, равную $0.75 \times 2 \times 0.7 = 1.05$ или 5%.

Имеется n валют c_1, \dots, c_n и таблица курсов валют R размером $n \times n$, в которой указывается, что за одну единицу валюты c_i можно купить $R[i, j]$ единиц валюты c_j .

В этой задаче требуется:

1. разработать алгоритм для определения максимального значения $R[c_i, c_{i_1}] \cdot R[c_{i_1}, c_{i_2}] \cdot \dots \cdot R[c_{i_{k-1}}, c_{i_k}] \cdot R[c_{i_k}, c_i]$.
2. Обосновать корректность алгоритма и оценить его эффективность.
3. Реализовать алгоритм в виде программы для исходных данных, и выполнить эксперименты.

1.2 Описание данных

Представлена таблица исходных данных.

Валюта	USD	EUR	GBP	JPY	CHF	CAD	AUD	RUB
USD	1	0,8651	0,7340	111,77	0,9290	12,538	13,692	720,144
EUR	11,559	1	0,8483	129,19	10,740	14,492	15,826	832,316
GBP	13,626	11,788	1	152,29	12,657	17,083	18,656	98,124
JPY	0,0089	0,0077	0,00657	1	0,0083	0,01122	0,01225	0,6443
CHF	10,765	0,9312	0,7901	120,29	1	13,495	14,739	77,52
CAD	0,7976	0,6901	0,5854	89,15	0,7411	1	10,921	57,43
AUD	0,7304	0,6319	0,5360	81,64	0,6784	0,9157	1	52,60
RUB	0,01389	0,01201	0,01019	15,520	0,01290	0,01741	0,01901	1

Таблица 1: Таблица обменных курсов валют R размером $n \times n$: за одну единицу валюты c_i можно купить $R[i, j]$ единиц валюты c_j

2 Метод решения, теоретический анализ и программная реализация алгоритма

2.1 Предварительный анализ задачи

Представленная задача состоит в том, что нам нужно найти путь(цикл), в котором произведение значений обменных курсов максимальна для каждой валюты. Задачу можно преобразовывать на задачу нахождения минимальной путь для каждой точки таким образом:

- Предположим, что дан полный связанный граф $G = (V, E)$, где V - множество всех валют, а E - множество всех возможных сделок с обменными курсами.
- Учитывая, что хотим найти максимум произведения, мы можем все данные прологарифмировать, затем умножить -1 .

Значения исходных данных варьируются в диапазоне $[0, 1000]$, при логарифмирования и умножении на -1 значения меняются в $[-\log 100, +\infty]$. Это значит, что обычный алгоритм Дейкстры не применим, так как в графе содержатся ребра, у которых веса меньше нуля. Но нас спасает алгоритм Беллмана-Форда, который сможет обрабатывать отрицательные ребра.

Алгоритм Беллмана-Форда состоит в том, что мы выполняем операцию "релаксации" для каждого ребер $|V| - 1$ раз, после чего получим минимальные пути от какой-то начальной точки до всех остальных. Если же после всех $|V| - 1$ обходов можно ещё сделать релаксацию, то это значит что в графе существуют отрицательные циклы.

Наша задача как раз состоит в нахождении такого минимального отрицательного пути. Если же отрицательного пути не существует, то и арбитраж бессмысленный.

2.2 Описание метода решения

В общем, метод решения нашей задачи выглядит следующим образом:

1. Прологарифмировать исходную матрицу смежности и затем взять знак минуса.
2. Выполнить алгоритм Беллмана-Форда для каждой вершины графа

3. Если алгоритм Беллмана-Форда выдаст результат о существовании отрицательных циклов, то надо найти все отрицательные циклы и среди из них минимальный по величине длины пути.
4. Запомнить и восстановить путь на исходной матрице смежности и соответствующее произведение значений на сделки.

2.3 Анализ алгоритма

Корректность алгоритма Беллмана-Форда уже обсуждена в https://en.wikipedia.org/wiki/Bellman%E2%80%93Ford_algorithm. Обычная реализация Беллмана-Форда требует $\mathcal{O}(|V||E|)$. Если мы запускаем алгоритм для каждой начальной вершины, то общий затрат времени будет $\mathcal{O}(|V|^2|E|)$

2.4 Описание программной реализации

Программная реализация на Python представлена в `ipynb` файле. Для запуска программы необходимо установлен Jupyter Notebook.

2.5 Результат поставленной задачи

Как можно увидеть в картине, наш алгоритм выдал хороший результат: для каждой валюты нашелся путь, по которому будет зарабатывается прибыль (то есть произведение обменных курсов больше единицы). Для почти всех валют надо зарабатывать на пути через все валюты. Можно отметить, что для 5-ой валюты (CAD) путь получилось так: $CAD \rightarrow AUD \rightarrow RUB \rightarrow CAD$, то есть не проходил через все валюты, а только 3 из них.

from 0 : [0, 2, 3, 5, 1, 4, 6, 7, 0]
1.0009966430467783

from 1 : [1, 3, 5, 6, 7, 0, 2, 4, 1]
1.0005572875046422

from 2 : [2, 3, 5, 6, 7, 0, 1, 4, 2]
1.0008460874930094

from 3 : [3, 5, 6, 7, 0, 1, 4, 2, 3]
1.0008460874930094

from 4 : [4, 6, 7, 0, 2, 3, 5, 1, 4]
1.0009966430467783

from 5 : [5, 6, 7, 5, 6, 7, 6, 7, 5]
1.0001420928824434

from 6 : [6, 7, 0, 2, 3, 5, 4, 1, 6]
1.000781423451686

from 7 : [7, 0, 2, 3, 5, 1, 4, 6, 7]
1.0009966430467785