**Assignment 1**

A white text on a black background

Description automatically generated

**Paper Code**:

**Name**: Sehun Bang

**Student** **ID**: 20111406

**Table of Contents**

1. **Introduction.**
2. **Expectation**
   1. **The Currency table.**
   2. **Negative logarithm**
   3. **Bellman Ford-Algorithm expectation**
3. **Programming Design**
   1. **Coding Design**
   2. **Results**
4. **Conclusion**
   1. **Comparison**
   2. **Conclusion**
5. **Introduction**

This report aims to see if we can benefit from the changing exchange rates between the US Dollar, NZ Dollar, and AU Dollar. We use simple math tricks called negative logarithms and a method named dynamic programming to turn these rates into a kind of map (or graph). By finding the shortest paths on this map, we can figure out the best ways to exchange currencies and potentially make a profit. We hope this report makes it clearer where the good opportunities might be.

1. **Expectations**

**2.1 The Currency table.**

**This is the Currency exchanged rate I have researched.**

|  |  |
| --- | --- |
| **Vertex List** | |
| **A** | **USD** |
| **B** | **NZD** |
| **C** | **AUD** |

|  |  |  |  |
| --- | --- | --- | --- |
| **From/To** | **A** | **B** | **C** |
| **A** | 1 | 1.68 | 1.57 |
| **B** | 0.60 | 1 | 0.94 |
| **C** | 0.64 | 1.07 | 1 |

**Benefit per cycle**

1. **Cycle 1 (USD -> NZD -> AUD -> USD)**

r (AB)+ r (BC) + r (CA) = 1.68 \* 0.94 \* 0.64= 1.010688

You should earn about one cent per cycle.

1. **Cycle 2 (USD -> AUD - > NZD -> USD)**

r (AC) + r (CB) + r (CA) = 1.57 \* 1.07 \* 0.60= 1.00794

You should earn 0.7 cent per cycle.

**2.2 Negative logarithm**

|  |  |
| --- | --- |
| **Vertex List** | |
| **A** | **USD** |
| **B** | **NZD** |
| **C** | **AUD** |

1. **Cycle 1 (USD -> NZD -> AUD -> USD)** *// A-> B -> C -> A*

|  |  |  |  |
| --- | --- | --- | --- |
| **From/To** | **A** | **B** | **C** |
| **A** | 1 | -0.225 | -0.194 |
| **B** | 0.225 | 1 | 0.027 |
| **C** | 0.194 | -0.027 | 1 |

w (AB) = -log (r (AB)) which is about **-0.225**

w (BC) = -log (r (BC)) which is about **0.027**

w (CA) = -log (r (CA)) which is about **0.194**

w (AB) + w (BC) + w (CA) = -log (r (AB) \*r (BC) \*r (CA)) about **-0.0046** which is lower than 1.

Therefore, the cycle has a negative weight.

1. **Cycle (USD -> AUD - > NZD -> USD)**

|  |  |  |  |
| --- | --- | --- | --- |
| **From/To** | **A** | **B** | **C** |
| **A** | 1 | -0.222 | -0.196 |
| **B** | 0.222 | 1 | 0.029 |
| **C** | 0.196 | -0.029 | 1 |

w (AC) = -log (r (AC)) which is about **-0.196**

w (CB) = -log (r (CB)) which is about **-0.029**

w (BA) = -log (r (BA)) which is about **0.222**

w (AC) + w (CB) + w (BA) = -log (r (AC) \* r (CB) \* r(BA))

which is about **-0.0034** which is lower than **1** as well.

Therefore, the cycle has a negative weight as well.

* 1. **Bellman Ford-Algorithm expectation**

**Using the previous negative logarithm tables.**

**d(u) is Cycle 1 (USD -> NZD -> AUD -> USD)**

**d’(u) is Cycle 2 (USD -> AUD -> NZD -> USD)**

|  |  |
| --- | --- |
| **Vertex List** | |
| **0** | **USD** |
| **1** | **NZD** |
| **2** | **AUD** |

**Iteration 1**

|  |  |  |  |
| --- | --- | --- | --- |
| **u** | **0** | **1** | **2** |
| **d(u)** | 0 | -0.225 | -0.194 |
| **d’(u)** | 0 | -0.222 | -0.196 |

**Iteration 2**

|  |  |  |  |
| --- | --- | --- | --- |
| **u** | **0** | **1** | **2** |
| **d(u)** | 0 | -0.225 | -0.198 |
| **d’(u)** | 0 | -0.225 | -0.196 |

**Our expectation of distance for Cycle 1 is.**

USA 0, NZD -0.225, and AUD -0.198

**Our expectation of distance for Cycle 2 is.**

USA 0, NZD -0.225, and AUD -0.196

1. **Coding**

**3.1 Coding Design**

**3.1.1 class Main**

**A screenshot of a computer code

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer code

Description automatically generated**

**3.1.2 Class Bellman Ford**

In the Bellman-Ford class the integer became double because of the type of data we have collected within the negative Logarithm.

**A computer screen shot of a code

Description automatically generated**

**A white screen with black text

Description automatically generated**

****

**3.2 Result**

The output of the program was:

A screenshot of a computer code

Description automatically generated

According to the computing shortest distance calculation.

|  |  |
| --- | --- |
| **Vertex List** | |
| **0** | **USD** |
| **1** | **NZD** |
| **2** | **AUD** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **0** | **1** | **2** |
| **Cycle 1** | 0 | -0.225 | -0.198 |
| **Cycle 2** | 0 | -0.225 | -0.196 |

1. **Conclusion**
   1. **Comparison**

|  |  |
| --- | --- |
| **Vertex List** | |
| **0** | **USD** |
| **1** | **NZD** |
| **2** | **AUD** |

|  |  |  |  |
| --- | --- | --- | --- |
|  | **0** | **1** | **2** |
| **Cycle 1** | 0 | -0.225 | -0.198 |
| **Cycle 2** | 0 | -0.225 | -0.196 |

The expected results for both the cycle and the computation output showed the same shortest distance from USD to other units.

**Let's reconsider the given values:**

Cycle 1: 0 -0.225 -0.198 = -0.423

Cycle 2: 0 -0.225 -0.196 = -0.421

**Cycle 1 has a smaller distance value. (0.002 smaller).**

* 1. **Conclusion**

In conclusion, both cycles of the exchange rate present opportunities for gains. However, Cycle 1 appears to be more efficient than Cycle 2. This is evident as the sum of the shortest distances for Cycle 1 (-0.423) is slightly smaller than that of Cycle 2 (-0.421), making Cycle 1 superior by a difference of 0.002.