Rayaan Irani and Saaketh Koka

Prof. David Brauchler III

Discrete Computational Structures

27 April 2021

The Traveling Trader

Forex Arbitrage is an NP-hard problem in which one aims to generate a risk-free gain by exploiting the discrepancies between the current market prices of a series of currencies. The most common form of arbitrage that is implemented is Triangle Arbitrage. In this form of arbitrage, the discrepancies between the exchange rates of three currency pairs are traded to produce a small profit. Triangle Arbitrage optimizes for execution speed rather than return on investment on the basis that exchange rates move considerably over short periods of time. We believe that this is not the case and that exchange rates are held generally constant over a period in the magnitude of seconds. As a result of this belief, we created a program that identifies the series of currency exchanges that generates the highest return on investment using recursive backtracking based on an input of a matrix of currency exchange rates taken at the beginning of the program.

The set of all foreign exchange prices represented in a currency matrix can be converted into a weighted directional complete graph that can be stored as an adjacency list. The input of the arbitrage identifier is a Comma Separated Values file, pulled from FactSet, which represents a currency matrix. A currency matrix is a matrix in which the exchange rate between two currencies is represented by the cell with the row of the initial currency and the column of the final currency. In this graph, each currency can be represented by a vertex, and the exchange rate between the currencies is the weight of the edge. The graph generated using this procedure is complete because there is a unique exchange rate between any two currencies in the matrix. Furthermore, the graph is weighted because the exchange rate between any two vertices on the graph varies, and as a result, specific paths have different weights than others. The graph of the currency exchange rates is directional because the exchange rate between any two currencies is neither equivalent to each other nor is the inverse of the rate in the opposite direction. Transaction fees cause this difference in rates levied on the sales of currencies to market makers. The program utilizes an Adjacency List to efficiently store the graph's information in a form that allows for efficient traversal. An Adjacency List is a data structure that for all vertices in a graph records each of the vertices adjacent to a given vertex and the cost to move to every vertex in that list.

The program can identify the sequence of best trades by generating the set of all simple circuits through the graph and generating a cost function based on a product of each of the moves exchange rates. The series of trades that are used in an arbitrage is a closed walk. This is because the currency of the beginning and ending transactions must be the same for the user to realize a profit. While it is true that a closed walk can generate an output, which represents a series of desired trades, for the practical purpose of preventing infinite loops from occurring while traversing the graph, the solution set to the arbitrage question is limited to all simple circuits. This prevention is caused by the simple circuit’s prohibition of repeats of non-terminating vertices. This prohibition limits the number of exchanges that can occur in a trade to the number of currencies in the matrix from an earlier potentially infinite amount. The set of all simple circuits is derived in the program using a recursive backtracking algorithm. This algorithm begins at a starting currency and generates a solution by recursively adding a random adjacent vertex to the solution until the point that either there are no adjacent vertices, or a currency is repeated, or the base currency is reached. If the base currency is reached, the path is recorded and compared against other tracks to determine its return. The return is determined by the product of the exchange rates of each of the currency pairs. In all three cases, the previous transaction is discarded, and another vertex is considered. This process repeats until all possible traversals have been completed. By going through all the possible trades and determining their return, the chain of transactions with the highest return can be found.

Implementing our program with real-world data sets indicates that Closed Circuit Arbitrage can generate strong returns on capital in short periods. When testing the program using a currency matrix derived from the Factset database, the program consistently returns a path that generates a positive profit. The data used as the input for the program indicates real-world conditions because that currency matrix originates from Factset, a financial data provider, provides delayed currency market quotes given to the nearest pip, to which most exchange rates are quoted to. Based on the realistic data input, the positive return seen in all the program's outputs, generally ranging from a gain of 0.2 to 0.4%, supports the program's usability.

Our program is not ready nor is it intended to be used in production in its current state. To get it production ready we would need to obtain access to high-precision, low-latency data, which often costs thousands of dollars per month. We would then need to program an interface to feed the data to our algorithm and another interface to communicate with the brokerage to execute orders.

Works Cited

“Arbitrage.” Edited by Rayaan J Irani, *Wikipedia*, Wikimedia Foundation, 8 Apr. 2021, en.wikipedia.org/wiki/Arbitrage.

“Triangular Arbitrage.” Edited by Rayaan Irani, *Wikipedia*, Wikimedia Foundation, 31 Mar. 2021, en.wikipedia.org/wiki/Triangular\_arbitrage.