A Discussion of Linear Programming and its Application to Currency Arbitrage Detection

Rachel Smith
University of Redlands

Follow this and additional works at: https://inspire.redlands.edu/cas_honors
Part of the Finance Commons, and the Finance and Financial Management Commons

Recommended Citation
A Discussion of Linear Programming and its Application to Currency Arbitrage Detection

Rachel Smith

Abstract
This paper explores the concept of currency arbitrage detection using basic Linear Programming methods. A thorough introduction to currency exchange is given. The Arbitrage Theorem is proven, and the key related concept of pricing via arbitrage is illustrated with an extended example. Next, the basics of Linear Programming are explained theoretically and shown through example. This information is then used to solve for arbitrage opportunities between five major currencies at one point in time, yielding four distinct solution sets. Results are discussed and relativized in the discussion section, where the reader can also find a brief summary of some related literature in the field.

1 Introduction
This paper seeks to explore the field of Linear Programming and apply its theoretical concepts to a real-world problem. More specifically, Linear Programming methods will be used to detect opportunities for arbitrage between five major currencies in the foreign exchange market. In the first section, an overview of basic terminology regarding currency exchange is provided. In the second section, the concept of arbitrage is defined, an extended example of pricing via arbitrage is provided, and the arbitrage theorem is proven. An introduction to the basics of Linear Programming is given; those concepts reviewed are subsequently used to formulate a Linear Program with currency exchange data for the top five traded currencies. The program is solved using a computerized solver and results are presented and discussed. Therefore, discussion section also contains mention of several important papers relevant to the topic of currency arbitrage detection. While models for currency arbitrage detection do exist, it is not the aim of this paper to create a model for future use, but rather to explore the theoretical basis of Linear Programming and apply this knowledge to currency arbitrage detection for one set of exchange rates.

1.1 What is Currency Exchange?
Currency is a vitally important part of daily life. It is well known that many different currencies exist in the world, and yet it is often not well understood how exchange rates
are determined, or what the process of foreign currency exchange involves. A solid understanding of currency and its exchange is necessary to move forward productively in a discussion of currency arbitrage detection; a basic explanation is given below.

A currency is the unit in which prices in any economy are listed. The value of a currency, in which prices for staples such as food, shelter, and clothing are listed, measures the general cost of living in a country or political region.[1] The US Dollar, Euro, and Pound are all examples of major world currencies. In order to understand the true value of a currency and therefore the basic price of any economy, the currency is usually compared to other currencies. The rate that results from this comparison is called the exchange rate. To restate this, the exchange rate or foreign exchange rate between two currencies is the rate at which one currency may be exchanged for the other.[1][2] There are two types of exchange rates: a spot rate and a forward rate. The spot rate is the exchange rates for currencies intended to be immediately purchased, or for immediate settlement. The forward rate is an exchange rate for currencies that will be exchanged on a date in the future. Actual trades between currencies take place on the foreign exchange market, or forex.[3] It is a characteristic of the foreign exchange market that it is extremely liquid, that is, there are a large number of sellers and buyers of currencies readily available at any given time, making it very easy to trade. Major traders in the forex market are known as market makers, and smaller traders are often called dealers or brokers.[2]

The comparative nature of an exchange rate implies that currencies can only be listed on the market in pairs, i.e. that the exchange rate only exists when one currency is being compared to another. A listing of two currencies on the forex market is called a currency pair. For example, the currency pair of US Dollars (USD) and Euros (EUR) appears as such: EUR/USD; out of convention, this particular currency pair is nearly always displayed in this order. However, other currency pairs are listed according to the direction of purchase. That is, the first currency listed is called the base currency, the second the quote currency; the exchange rate displays one unit of the base currency, expressed in terms of the quote currency. Buying the EUR/USD currency pair means buying Euros while selling US Dollars and selling this currency pair means selling Euros while buying US Dollars. [2]

There are two prices listed on the forex market along with each currency pair: the bid price and the ask price. The bid price is the price that a dealer is willing to pay for the currency pair, i.e. what the seller will receive if selling the currency pair on the market. The ask price is the price at which the dealer is willing to sell the currency pair, i.e. the price that the buyer must pay for the currency pair. Viewing an example can be helpful in understanding the nuances of the exchange market. Let the rate for EUR/USD be 1.30970/1.30984. Then 1.30970 is the bid price and means that one can sell 1 Euro for 1.30970 US Dollars. Similarly, 1.30984 is the ask price and means that one can buy 1 Euro for 1.30984 USD. Note that the ask price is always higher than the bid price; the discrepancy between the bid and ask price is called the bid-ask spread or spread. The spread is measured in a unit called a pip. In the case of stronger

---

1This was the listed rate on March 7, 2013 at 8:38 PM, according to oanda.com
currencies, such as the Euro or US Dollar, one pip is equal to .0001 difference in price. So in the example listed above, the spread is 1.4 pips because the total difference in ask price and bid price is .00014 Dollars. The spread represents the cost of trading with a market maker on the foreign exchange market. [2] Often, when currency exchange rates are quoted outside of the forex market, i.e. in a newspaper or other media outlet, only one rate is displayed. This rate is the mid-market rate and lies between the bid and ask rates; the mid-market rate is calculated as the median average of the bid and ask rates quoted by all the buyers and sellers on the market. [2]

What exactly determines the value of a currency is contested. Many economists believe that long-term rates are determined by the fiscal and monetary policy of a nation, as well as that nation’s economic health. [3] In the short-term, and in the most literal sense, exchange rates are determined by the supply and demand for currencies on the forex. Countries whose exchange rates are determined solely by the foreign exchange market, where there is no regular government intervention to determine rates, are said to have free or floating currency. Most nations in the world today follow some form of this model. In other cases, currencies are fixed if they are not allowed to fluctuate according to market demands. [3] In this situation, the currency may be pegged to the price of gold or another currency’s value, which the government ensures by buying or selling their currency on the market to hold their currency’s value constant.

The most famous example of a country with this policy is China, who has traditionally pegged their currency to the USD. [3] During the post-WWII era, the member countries of the Bretton Woods System had de facto fixed exchange rates, since they pegged their currencies to gold or the USD, which itself was backed by gold. [2] During this economic era, changes in exchange rates were allowed only in extreme circumstances and with the approval of the International Monetary Fund, an institution that oversaw the Bretton Woods system. This system collapsed in the early 1970s and gave way to floating currencies. [4] As indicated above, most developed or developing countries today have floating currencies, with China being a notable exception. The ever-changing nature of currency exchange rates, now determined by market forces, means that there are some opportunities for investors to profit from discrepancies or asymmetries that may exist in the foreign exchange market. While some investors profit from speculation, others search for currency arbitrage opportunities, which present a more calculated and less risky way to generate earnings in the market.

2 Currency Arbitrage Detection

The term Currency Arbitrage can actually be used in reference to several phenomenon. A successful attempt to exchange currencies in such a manner that one ends up with more units of the base currency than one began with is referred to as a currency arbitrage. [5] Explained more explicitly: say an investor begins with 1 USD, and exchanges currencies in a circular fashion and in a certain pattern. In any situation where the investor ends up with a sum greater than 1 USD, (s)he has completed currency arbitrage. The gains from such a scheme may be very small on a unit-scale —
think fractions of a cent — but can add up to significant figures if exchanging large amounts at once, as in the case of major traders or corporations repatriating their funds. This particular type of currency arbitrage is referred to by some as space or geographic arbitrage, because gains of this type were initially achieved by finding discrepancies in quoted rates in different geographical markets.\cite{6} Another example of currency arbitrage would be to take advantage in disparities between forward and spot exchange rates, as well as quoted interest rates within two or more currency markets. This model, referred to as covered interest arbitrage, or time arbitrage enables the investor to borrow from the bank with the most favorable combination of interest rates and forward exchange rates. \cite{7,6} As is discussed in the following section, arbitrage can be understood in a number of ways, meaning that both of the phenomena mentioned above are classified as arbitrage, though one occurs in one moment and the other over a period of time.\footnote{\textsuperscript{2}For a number of definitions and interpretations of the term arbitrage, refer to Christofides, Hewins and Salkin} This paper explores the former type of currency arbitrage, i.e. in which an investor exchanges currency at one point in time to generate a profit. The aim of this section of the paper is to discuss the ways in which to detect opportunities for currency arbitrage, given exchange rates between multiple currencies.

### 2.1 What is Arbitrage?

First, it is important to understand what exactly arbitrage means. A definition given by Corneujols and Tütüncü in Optimization Methods in Finance\cite{8} states that

**Definition 2.1.** An arbitrage is a trading strategy

1. that has a positive initial cash flow and has no risk of a loss later (type A), or
2. that requires no initial cash input, has no risk of a loss, and a positive probability of making profits in the future (type B).

That is to say, in a situation where an investor can guarantee that (s)he will profit from his/her investments, there is arbitrage. Note that traditional finance texts understand arbitrage to mean certain profit earned from simultaneously buying and selling securities; some texts explicitly state that this occurs over a period of time, while others do not.\cite{7,9,8} Referring again to currency, this definition makes it easier to understand why both examples given in the previous section are called arbitrage. Exchanging currencies in a certain pattern to earn a profit may not seem to fit the traditional definition of arbitrage given here, but consider it this way: when exchanging currency, one is simultaneously buying and selling currencies. Recall that currency is bought or sold in a pair; the nature of a currency pair implies that a trader is actually completing two transactions at once — a purchase and a sale. For currency arbitrage to work in the manner discussed above, the investor would need to submit purchase and sale orders in a bundle all at the same time. In reality, the investor would not be making one trade
followed by another, but would be submitting orders for trades all at once so that the
exchange pattern was executed in one moment. Put this way, it becomes much easier
to understand this as arbitrage, as it is the simultaneous buying and selling of several
securities, in this case currencies, in order to earn a sure profit. The second example
above is a much more classic example of arbitrage, that is, simultaneously exchanging
currencies and investing in countries with more favorable interest rates over a period
of time.[7]

The concept of arbitrage is perhaps best understood through an example. Here
an example that Ross uses in his text is closely followed, explaining arbitrage in the
context of two possible investments, and then going on to explain the Law of One
Price, which arises as a consequence of the observations made in the example.

2.1.1 Understanding Arbitrage: Option Pricing as an Example of
Arbitrage

Suppose an investor is deciding whether or not to purchase an option. By purchasing
an option, the investor is either entitled to purchase or to sell a stock at a fixed price
at some time in the future. An option that entitles an investor to purchase a stock at
a fixed price is called a call option, whereas an option that entitles an investor to sell
at a fixed price is called a put option. The two concepts are very similar, but are useful
in situations with opposite outcomes; call options are useful when the price of a stock
is rising, and put options when the price of the stock is falling. During the following
example, the word “option” will be used, but will refer exclusively to a call option.[9]

Let the nominal interest rate be \( r \) and the present price of the stock in question be
$50 per share. Suppose it is known that at the end of one time period, the stock will
have either halved or doubled, i.e. that the price will be either $25 or $100 per share.
Further, suppose that at \( t = 0 \), the investor may buy an option to purchase one share
of stock for $75 at time \( t = 1 \). Let \( C \) denote the cost of this option for one share, and
\( C_y \) the cost of options for \( y \) shares of the stock. Additionally, the investor can purchase
\( x \) shares of the stock at time \( t = 0 \) at a cost of 50x (current market price), where each
share will either be worth $25 or $100 at time \( t = 1 \) (future market price). Note that
\( x \) and \( y \) can take on both positive and negative values, where positive values denote
a purchase and negative values denote a sale. The goal in this exercise is to find the
appropriate price for the option, \( C \). As will be seen, the appropriate value for \( C \) is the
one that does not result in positive present value gain, i.e. does not allow arbitrage.
Table 1 below illustrates the situation outlined above.

<table>
<thead>
<tr>
<th>Value of Investment</th>
<th>( t = 0 )</th>
<th>( t = 1 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x )</td>
<td>50</td>
<td>25</td>
</tr>
<tr>
<td>( y )</td>
<td>( C )</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1: Possible values of investments \( x \) and \( y \) at \( t = 0 \) and \( t = 1 \)

At time \( t = 0 \) the investor purchases \( x \) shares of stock and \( y \) options at a cost of
50x + \( C_y \). If the cost given by this formula is a positive number, i.e. if it results in
a cash outflow, then the money to finance the purchase should be withdrawn from a
d bank and repaid at \( t = 1 \) with interest at rate \( r \). If the cost is a negative number and
therefore represents a cash inflow, then that sum should be deposited in a bank and
withdrawn at time \( t = 0 \) with interest at rate \( r \). The total value of the investment at
time \( t = 1 \) will depend on the price \( p \) of the stock at \( t = 1 \) and is given by

\[
\text{value} = \begin{cases} 
100x + 25y & p = 100 \\
25x & p = 25 
\end{cases}
\]

These formulas are derived by observing that if \( p = 100 \) at \( t = 1 \), then the \( x \) shares of
stock will be worth \( 100x \), and the \( y \) options will be worth \( (100 - 75)y = 25y \). In the
second case, however, when \( p = 50 \), the \( x \) shares of stock will be worth \( 25x \), and the
\( y \) units of options will be worthless. An arbitrage scheme would aim that the holdings
would be the same at \( t = 1 \), regardless of the price \( p \) at \( t = 1 \). Therefore, choose \( y \)
such that

\[
100x + 25y = 25x \\
\implies y = -3x
\]

Note that since \( y \) and \( x \) have opposite sign, when \( x \) units of stock are purchased at
\( t = 0 \), \( 3x \) units of stock options are sold at \( t = 0 \), and vice-versa. So with the equation
\( y = -3x \), the value of the investment at \( t = 1 \) will always be \( 25x \). As a result, it follows
that after paying off the loan (in the case \( 50x + Cy > 0 \)) or withdrawing the deposited
money from the bank (in the case \( 50x + Cy < 0 \) at \( t = 1 \), the investor will have gained
the amount below:

\[
\text{gain} = 25x - (50x + Cy)(1 + r) \\
= 25x - (50x - 3xC)(1 + r) \\
= x[25 - (50 - 3C)(1 + r)] \\
= (1 + r)x[3C - 100 + 50(1 + r)^{-1}] 
\]

As can be seen from this equation, the only way the gain will be equal to zero is in
that case that \( 3C = 100 + 50(1 + r)^{-1} \). In any other case, there exists an arbitrage.
Said differently, an investor can guarantee a positive gain by letting \( x > 0 \) when
\( 3C > 100 + 50(1 + r)^{-1} \) or letting \( x < 0 \) when \( 3C < 100 + 50(1 + r)^{-1} \). Therefore the
price of \( C \) to ensure that no arbitrage exists should be

\[
C = \frac{100 + 50(1 + r)^{-1}}{3} \quad (1)
\]

Referring to the definition of arbitrage given by Corneujols and Tütüncü, which is
listed above, the example just discussed can be classified as a type A arbitrage, be-
cause there is an initial positive cash flow and no risk of loss later. That is, when the
pricing of the option allows for arbitrage, the investor will always receive cash at \( t = 0 \)
from the combined purchase and sale of securities, and there is no risk of losing funds
2 CURRENCY ARBITRAGE DETECTION

by \( t = 1 \). This example is useful in understanding the Law of One Price, which can often be applied to see existence of arbitrage between two investments.

The Law of One Price: Consider two investments, the first of which costs the fixed amount \( C_1 \) and the second the fixed amount \( C_2 \). If the (present value) payoff from the first investment is always identical to that of the second investment, then either \( C_1 = C_2 \) or there is an arbitrage.\(^9\)

Another way to understand the law of one price, as given above, is the generalized law of one price, which views the scenario using inequalities, rather than equalities. This can be useful in situations in which inequalities are a recurring feature of investing scenarios.

The Generalized Law of One Price: Consider two investments, the first of which costs the fixed amount \( C_1 \) and the second the fixed amount \( C_2 \). If \( C_1 < C_2 \) and the (present value) payoff from the first investment is always at least as large as that from the second investment, then there is an arbitrage.\(^9\)

The proof of this is rather intuitive and goes as follows:

Let \( C_1 \) and \( C_2 \) represent the fixed costs of two investments. Let \( P_1 \) be the present-value payoff from \( C_1 \) and \( P_2 \) be the present-value payoff from \( C_2 \). If \( C_1 < C_2 \) and \( P_1 \geq P_2 \) it can be shown there is an arbitrage:

Since \( C_1 < C_2 \Rightarrow P_2 - P_1 = c \), where \( c > 0 \) and represents the difference in price between the two investments. Additionally, since \( P_1 \geq P_2 \Rightarrow P_1 - P_2 = p \), where \( p \geq 0 \) and represents the difference in payoff between the two.

Then an arbitrage can be obtained if at time \( t = 0 \), the investor sells one unit of \( C_2 \) and buys one unit of \( C_1 \), so that (s)he is left with a remainder of \( c \) dollars. At time \( t = 1 \) the investor will have earned \( P_1 \), which is greater than or equal to \( P_2 \), meaning that the discrepancy in his/her earnings and the earnings of an investor who purchased \( C_2 \) will be \( p \), which is greater than or equal to zero. Note that (s)he will not have lost money relative to the other investor. In fact, the total relative gain at the end of the investment will be \( p + c \), which is known to be greater than zero because \( c > 0 \).

This type of investment scheme would work in any situation where two investments have equal present value payoff but inequal price. The phenomenon described in this example can be succinctly generalized to a situation with many different investment opportunities and outcomes. This is presented in the following section.

2.1.2 The Arbitrage Theorem

Now consider a situation in which an experiment has a set of \( m \) possible outcomes \( i = (1, 2, \ldots, m) \), and \( n \) wagers \( j = (1, 2, \ldots, n) \) concerning these outcomes are available. If an amount \( x \) is bet on wager \( j \), then \( r_{ij}x \) is received if the outcome of the experiment is \( i \). That is, \( r_{ij} \) represents the return on the \( j \)th wager for the \( i \)th outcome. A betting strategy is a vector \( x = (x_1, x_2, \ldots, x_n) \) with dimension \( n \), understood to mean that \( x_1 \) is a bet on wager 1, \( x_2 \) is a bet on wager 2, \ldots, and \( x_n \) is a bet on wager \( n \). If the
outcome of the experiment is $i$, then the return from $x$ is [9]:

$$\text{Return from } x = \sum_{j=1}^{n} r_{ij} x_j$$

With this definition of the return from $x$, The Arbitrage Theorem is presented below.

**Theorem 2.1.** The Arbitrage Theorem: states that exactly one of the following is true: Either

(a) there is a probability vector $p = (p_1, p_2, \ldots, p_m)$ for which

$$\sum_{i=1}^{m} r_{ij} p_i = 0 \text{ for all } j = 1, 2, \ldots, n.$$ 

or else

(b) there is a betting strategy $x = (x_1, x_2, \ldots, x_n)$ for which

$$\sum_{j=1}^{n} r_{ij} x_j > 0 \text{ for all } i = 1, 2, \ldots, m.$$ 

This proof relies on principles of Linear Programming and can best be understood after becoming familiar with the Duality Theorem of Linear Programming. So before this theorem can be proven, a discussion of Linear Programming and Optimization is required.

### 2.2 Linear Programming

The field of Linear Programming (LP) was developed during and immediately after World War II. The problems to which the Linear Programming tool were applied ranged from planning crop rotation, to routing of ships between harbors, and to the assessment of the flow of commodities between industries of the economy. These problems, the methods of solving them, and their solutions were presented in the summer of 1949 at the University of Chicago at a conference held by the Cowles Commission for Research in Economics. It was at this conference that the diverse problems were unified to provide a mathematical framework and a computational method, called the simplex algorithm, to formulate such problems explicitly and solve them efficiently.\cite{10}

The general problem of linear programming and the Simplex method for solving is often attributed to George B. Dantzig, Marshall Wood, and their associates of the U.S. Department of the Air Force.\cite{11}\cite{12}\cite{13} This development coincided with the development of electronic digital computers, which became increasingly necessary tools to solve problems in which hand computation would not have been realistic.\cite{10}

Linear Programming problems can be understood as optimization programming problems subject to linear constraints. An optimization problem is a problem that seeks to maximize or minimize a function, which in turn consists of a number of variables.

\[3\text{For a detailed account of the origins of linear programming and its historical influences, refer to Dantzig, Chapter 2.}\]
(or functions), with each of the variables or functions subject to certain constraints. Programming problems deal with determining optimal allocations of limited resources to meet given objectives. Explained differently, programming problems are used to deal with situations in which a number of resources are to be combined to yield one or more products, with certain restrictions on some or all of the following: the total amount of each resource available, the quantity of each product made, and the quality of each product.\[11\]

With a little more background, it becomes easier to understand what a Linear Programming problem truly is. First, we define a few important terms. A linear function is a function of the form:

\[ f(x_1, x_2, \ldots, x_n) = c_1 x_1 + c_2 x_2 + \cdots + c_n x_n = \sum_{j=1}^{n} c_j x_j, \]

where \( c_1, c_2, \ldots, c_n \) are real numbers and \( x_1, x_2, \ldots, x_n \) are real variables.

For a linear function \( f \) and a real number \( b \), the equation

\[ f(x_1, x_2, \ldots, x_n) = b \]

is called a linear equation, and the inequalities

\[ f(x_1, x_2, \ldots, x_n) \geq b \]
\[ f(x_1, x_2, \ldots, x_n) \leq b \]

are called linear inequalities. In the field of Linear Programming, linear equations and linear inequalities are referred to as linear constraints.\[12\] With these functions defined, the definition of a Linear Programming problem can now be understood.

Definition 2.2. A Linear Programming problem is a problem that optimizes a linear function subject to a finite number of linear constraints, typically applied to situations that deal with resource allocation seeking a particular outcome.\[10][12\]

The linear function that is to be maximized or minimized in an LP problem is called the objective function of that problem. Values of \( x_1, x_2, \ldots, x_n \) that satisfy all the constraints of an LP problems constitute a feasible solution of that problem. A feasible solution that optimizes the objective function (maximizes or minimizes, depending on the goals of the problem) is called an optimal solution and the corresponding value of the objective function is the optimal value of the LP problem. LP problems that have no feasible solutions at all are called infeasible, and LP problems that have infinite feasible solutions but no optimal solution are referred to as unbounded. As a consequence of these definitions, every LP problem can be classified according to these three categories: either it is feasible with an optimal solution, it is infeasible, or it is unbounded.\[12\]

It is easier to understand the terminology just explained through an example, which closely follows Chvátal's text. The following example is called the diet problem and is a widely used to illustrate the basic form of a Linear Programming problem.

Say that Jane wonders how much she must spend on food in order to get all the energy, protein, and calcium that she needs every day. She aims for a total of 2,000 kcal of energy, 55 g of protein, and 800 mg of calcium daily. She chooses three foods to build her diet from and compiles the nutritional data for these foods in Table 2.


Theoretically, her goals could easily be achieved by consuming 13 servings of milk a day, for a total daily cost of only $1.17. But she may tire of drinking milk, or realize that consuming 13 servings of milk in a day may cause health issues of its own. So Jane decides to impose limits on the number of servings of each food she is willing to eat per day.

Oatmeal at most 5 servings per day
Chicken at most 4 servings per day
Milk at most 8 servings per day

Oatmeal is denoted as $x_1$, Chicken as $x_2$, and Milk as $x_3$ so that limits can be rewritten as such:

\[
0 \leq x_1 \leq 5 \\
0 \leq x_2 \leq 4 \\
0 \leq x_3 \leq 8
\]

Daily requirements for daily calorie, protein, and calcium intakes are then rewritten as a set of inequalities with the variables just specified:

\[
110x_1 + 205x_2 + 160x_3 \geq 2000 \\
4x_1 + 32x_2 + 8x_3 \geq 55 \\
2x_1 + 12x_2 + 285x_3 \geq 800
\]

Here the first equation represents calorie intake, the second protein, and the third calcium. The coefficients on the left-hand side of the equation correspond to the nutritional values of each food, as specified above. For example, the value $110x_1$ refers to the fact that each serving of Oatmeal ($x_1$) contains 110 calories. The right hand sides of the equations correspond to the daily goals for calorie, protein, and calcium intake, respectively. So the value 2000 in the first equation refers to the desired minimum

<table>
<thead>
<tr>
<th>Food</th>
<th>Serving Size</th>
<th>Energy (kcal)</th>
<th>Protein (g)</th>
<th>Calcium (mg)</th>
<th>Price per Serving (cents)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oatmeal</td>
<td>28g</td>
<td>110</td>
<td>4</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>Chicken</td>
<td>100g</td>
<td>205</td>
<td>32</td>
<td>12</td>
<td>24</td>
</tr>
<tr>
<td>Milk</td>
<td>237cc</td>
<td>160</td>
<td>8</td>
<td>285</td>
<td>9</td>
</tr>
</tbody>
</table>

Table 2: Nutritional Value and Cost of Select Foods
calorie intake. Incorporating the price per serving (in cents) from Table 2, the total cost of whatever menu Jane ends up with will be:

\[ 3x_1 + 24x_2 + 9x_3 \]

So Jane's diet problem is rewritten as a linear program by seeking to:

\[
\begin{align*}
\text{minimize} & \quad 3x_1 + 24x_2 + 9x_3 \\
\text{subject to} & \quad 110x_1 + 205x_2 + 160x_3 \geq 2000 \\
& \quad 4x_1 + 32x_2 + 8x_3 \geq 55 \\
& \quad 2x_1 + 12x_2 + 285x_3 \geq 800 \\
& \quad x_1 \leq 5 \\
& \quad x_2 \leq 4 \\
& \quad x_3 \leq 8 \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]

In this example, the objective function can be clearly identified as the cost expression \(3x_1 + 24x_2 + 9x_3\), and the linear constraints as being the inequalities which the cost expression is minimized subject to. While this is a useful example to understand the formulation of an LP problem, it is important to note that general LP solving methods are often most easily applied to LP problems that are in standard form. These problems seek to maximize an objective function subject to linear constraints that are "\(\leq\)" equations, with the restriction that all variables \(x_1, \ldots, x_n \geq 0\). A more formalized explanation of standard form is given below.

Suppose that, for given constants \(c_j, b_i\), and \(a_{ij}\) with \(i = 1, \ldots, m\) and \(j = 1, \ldots, n\), we want to choose values of \(x_1, \ldots, x_n\) that will

\[
\text{maximize } \sum_{j=1}^{n} c_jx_j
\]

subject to

\[
\sum_{j=1}^{n} a_{ij}x_j \leq b_i \text{ for } i = 1, \ldots, m.
\]  \(\text{(3)}\)

\(x_j \geq 0 \text{ for all } j = 1, \ldots, n.\)

This LP problem is presented in standard form. It should be observed that the terminology regarding standard form equations is not unified in the field of Linear Programming. Some authors use the term canonical form rather than standard form, or define standard form to be a minimization problem, subject to "\(\geq\)" or "\(=\)" constraints. For example, Dantzig defines canonical form as a minimization problem subject to linear equality constraints, while Strum, Ross, and Chvátal use the term standard...
form and define it as the maximization problem shown above. This paper uses the term standard form in the same manner as Strum, Ross, and Chvátal.

Equation (3) will also be referred to as a primal linear program. Every primal linear program has what is called a dual problem. How the dual program of any given primal is defined depends on a few key aspects of the primal. Chvátal offers a simple and effective explanation of finding the dual of any primal program, which is followed below:

For this example it is best to view the LP problem as such:

$$\text{maximize } \sum_{j=1}^{n} c_j x_j$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \quad (i \in I) \quad (3.1)$$
$$\sum_{j=1}^{n} a_{ij} x_j = b_i \quad (i \in E)$$
$$x_j \geq 0 \quad (j \in R)$$

where the set $i = 1, \ldots, m$ is split into two disjoint subsets $I$ and $E$, which correspond to inequalities and equalities, respectively. The set $j = 1, \ldots, n$ has a subset $R$, which includes the variables $x_j$ that are specified as having non-negative value. These variables are called restricted. Note that variables may, in fact, have upper or lower bounds, but unless they are specified as non-negative, they are not called restricted for this purpose. Those variables $x_j$ that are not restricted, i.e. $x_j$ such that $j \notin R$ are called free variables, and the set of all $j \notin R$ will be called $F$. Note that standard form as defined in equation (3) can easily be achieved in equation (3.1) if $E = \emptyset$ and $F = \emptyset$. [12]

Then the dual problem of equation (3.1) is written as such:

$$\text{minimize } \sum_{i=1}^{m} b_i y_i$$
subject to

\[ \sum_{i=1}^{m} a_{ji} y_i \geq c_j \quad (j \in R) \]  \hspace{1cm} (3.2)

\[ \sum_{i=1}^{m} a_{ji} y_i = c_j \quad (j \in F) \]

\[ y_i \geq 0 \quad (i \in I) \]

where all free variables in equation (3.2) are \( y_i \) such that \( i \notin I \), or better, for \( i \in E \). Chvátal includes a very helpful table in his text, which is reproduced below:

<table>
<thead>
<tr>
<th>In the primal</th>
<th>In the dual</th>
<th>Denoted by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inequality constraints</td>
<td>Restricted variables</td>
<td>( i \in I )</td>
</tr>
<tr>
<td>Equation constraints</td>
<td>Free variables</td>
<td>( i \in E )</td>
</tr>
<tr>
<td>Restricted variables</td>
<td>Inequality constraints</td>
<td>( j \in R )</td>
</tr>
<tr>
<td>Free variables</td>
<td>Equation constraints</td>
<td>( j \in F )</td>
</tr>
</tbody>
</table>

Table 3: Primal-Dual Correspondence

Using the results from the table above, it can be determined that the dual problem of equation (3) is to choose values \( y_1, y_2, \ldots, y_m \) that will

\[
\text{minimize} \quad \sum_{i=1}^{m} b_i y_i
\]

subject to

\[ \sum_{i=1}^{m} a_{ji} y_i \leq c_j \quad (4) \]

\[ y_i \geq 0 \text{ for } i = 1, \ldots, m \]

Recall again, that equation (4) is formatted as such because in equation (3), \( x_j \geq 0 \) for all \( j = 1, \ldots, n \), meaning there are no free variables, which implies only inequality constraints will exist in the dual. Similarly, the fact that \( i \in I \) for all \( i = 1, \ldots, m \) in equation (3) implies that \( y_i \geq 0 \) for all \( i = 1, \ldots, m \) in equation (4). [12]

This explanation of the relationship between primals and duals is difficult to conceptualize without a concrete example. Refer again to Jane's diet problem, written in
the LP formulation below:

\[
\begin{align*}
\text{minimize} & \quad 3x_1 + 24x_2 + 9x_3 \\
\text{subject to} & \quad 110x_1 + 205x_2 + 160x_3 \geq 2000 \\
& \quad 4x_1 + 32x_2 + 8x_3 \geq 55 \\
& \quad 2x_1 + 12x_2 + 285x_3 \geq 800 \\
& \quad x_1 \leq 5 \\
& \quad x_2 \leq 4 \\
& \quad x_3 \leq 8 \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]

In this problem, it can be seen that \( j = 1, 2, 3 \) and that \( i = 1, 2, 3, 4, 5, 6 \), since there are three variables and six equations. Note that all of the equations above are inequalities, but that they are not all of the same type. That is, note the presence of "\( \leq \)" and "\( \geq \)" inequalities in the above LP. The "\( \leq \)" present a particular problem because they do not conform to a standard format as seen in either the primal or the dual problem given by equations (3.1) or (3.2). To remedy this, simply change the inequalities into equalities by introducing \textit{slack variables}. This is a process that is explained in further detail in section 2.4 (The Simplex Method) of this paper. A brief description will suffice here: notice that \( x_1 \leq 5 \) has the same meaning as \( x_1 + x_4 = 5 \), where the variable \( x_4 \) represents the difference between \( x_1 \) and 5. It is possible to address all three inequalities in the above LP by introducing slack variables \( x_4, x_5, \) and \( x_6 \), so that the above "\( \leq \)" inequalities can be rewritten as:

\[
\begin{align*}
& x_1 + x_4 = 5 \\
& x_2 + x_5 = 4 \\
& x_3 + x_6 = 8
\end{align*}
\]

This changes \( j, \) and now means that \( j = 1, 2, 3, 4, 5, 6, \) like \( i \). The entire LP now appears as follows:

\[
\begin{align*}
\text{minimize} & \quad 3x_1 + 24x_2 + 9x_3 + 0x_4 + 0x_5 + 0x_6 \\
\text{subject to} & \quad 110x_1 + 205x_2 + 160x_3 \geq 2000 \\
& \quad 4x_1 + 32x_2 + 8x_3 \geq 55 \\
& \quad 2x_1 + 12x_2 + 285x_3 \geq 800 \\
& \quad x_1 + x_4 = 5 \\
& \quad x_2 + x_5 = 4 \\
& \quad x_3 + x_6 = 8 \\
& \quad x_1, x_2, x_3 \geq 0
\end{align*}
\]

Now it can be seen that the number of inequalities is 3, therefore the \( i \in I \) for \( i = 1, 2, 3, \) In the same way, \( i \in E \) for \( i = 4, 5, 6, \) Note that \( x_j \geq 0 \) for all \( j = 1, 2, 3, 4, 5, 6, \) so
\( j \in R \) for all \( j = 1, 2, 3, 4, 5, 6 \). Therefore, the dual of the program can now be written as such:

\[
\begin{align*}
\text{maximize} & \quad 2000y_1 + 55y_2 + 800y_3 + 5y_4 + 4y_5 + 8y_6 \\
\text{subject to} & \quad 110y_1 + 4y_2 + 2y_3 + y_4 + 0y_5 + 0y_6 \leq 3 \\
& \quad 205y_1 + 32y_2 + 12y_3 + 0y_4 + 4y_5 + 0y_6 \leq 24 \\
& \quad 160y_1 + 8y_2 + 285y_3 + 0y_4 + 0y_5 + y_6 \leq 9 \\
& \quad y_4 \leq 0 \\
& \quad y_5 \leq 0 \\
& \quad y_6 \leq 0 \\
& \quad y_4 \geq 0 \text{ for } i = 1, 2, 3
\end{align*}
\]

Stopping to consider the meaning of this program makes it clear that this is the correct dual program formation. This LP seeks to maximize the number of calories, subject to linear cost restraints.

With a firm understanding of the concept of a primal and dual linear program, it is now possible to move forward. An important theorem for understanding Linear Programming problems is stated without proof. This theorem was conceived in its original form Gale, Kuhn, and Tucker in 1951, and has been taken here from Ross's text. The proof of this theorem relies on the relationship between the primal and dual, most notably on the relationship between their respective optimal solutions. The proof also uses solving techniques that are important in the Simplex Method, described in section 2.4 of this paper.[12]

**Theorem 2.2. (Duality Theorem of Linear Programming)** If a primal and its dual linear program are both feasible, then they both have optimal solutions and the maximal value of the primal is equal to the minimal value of the dual. If either problem is infeasible, then the other does not have an optimal solution.[9]

Using this theorem, it is now possible to prove the Arbitrage Theorem.

### 2.2.1 Proving the Arbitrage Theorem

Recall, for any vector \( x = (x_1, x_2, \ldots, x_n) \) representing \( j \) wagers on \( i \) possible outcomes, the expected return from outcome \( i \) is:

\[
\text{Return from } x = \sum_{j=1}^{n} r_{ij} x_j \tag{2}
\]

Using this information, it is possible to understand the statement of The Arbitrage Theorem:

\[\text{For the proof of the Duality Theorem please refer to Chvátal, Chapter 5 or Dantzig, Section 6-3}\]
Theorem 2.3. The Arbitrage Theorem: states that exactly one of the following is true: Either
(a) there is a probability vector \( p = (p_1, p_2, \ldots, p_m) \) for which
\[
\sum_{i=1}^{m} r_{ij} p_i = 0 \text{ for all } j = 1, 2, \ldots, n.
\]
or else
(b) there is a betting strategy \( x = (x_1, x_2, \ldots, x_n) \) for which
\[
\sum_{j=1}^{n} r_{ij} x_j > 0 \text{ for all } i = 1, 2, \ldots, m.
\]

During the proof, Ross’s text is followed.

Proof: Let \( x_{n+1} \) be the amount that the investor is sure to win. Consider the LP problem to maximize \( x_{n+1} \). Recall that \( r_{ij} \) is defined as the expected return of wager \( j \) from the outcome \( i \). Therefore, if the investor uses the betting strategy \( (x_1, x_2, \ldots, x_n) \), then (s)he will receive \( \sum_{j=1}^{n} r_{ij} x_j \) if the outcome is \( i \), where \( i = 1, \ldots, m \) and represents the set of all the possible outcomes \( \{1, \ldots, m\} \). Therefore, (s)he will want to choose a betting strategy \( (x_1, x_2, \ldots, x_n) \) and \( x_{n+1} \) in order to maximize \( x_{n+1} \)

subject to
\[
\sum_{j=1}^{n} r_{ij} x_j \geq x_{n+1} \text{ for } i = 1, \ldots, m. \tag{5}
\]

Here a few things from the Section 2.1.2 (The Arbitrage Theorem) of this paper must be recalled. Namely, that the left half of the linear constraining function in (5) is the Return from \( x \) given in Equation (2). Additionally, the Return from \( x \) is set greater than or equal to \( x_{n+1} \) in the constraining function because the goal is to achieve a return on wagers, regardless of outcome, that is larger than the amount that is sure to be won \( (x_{n+1}) \) through arbitrage. A last important observation as that \( x_j \) can be either positive or negative for all values of \( j = 1, \ldots, n \). Note that a positive value of \( x_j \) for some \( j = 1, \ldots, n \) would denote a purchase of a security, while a negative value of \( x_j \) for some \( j = 1, \ldots, n \) would denote a sale of a security.

Moving forward, let \( a_{ij} = -r_{ij} \) for \( i = 1, \ldots, m \) and \( a_{(i)(n+1)} = 1 \), so that (5) may be rewritten in standard form. Note that \( a_{ij} \) is equal to \( -r_{ij} \) because standard form calls for the linear constraint function to be less than or equal to something. Also choose \( a_{(i)(n+1)} = 1 \). The expression \( a_{(i)(n+1)} \) denotes the coefficient value for each of the \( i = 1, \ldots, m \) outcomes for the variable \( x_{n+1} \). Since this variable represents the amount gained through arbitrage, it is desirable that it always has a coefficient value of 1. Put differently, it is undesirable to allow any coefficient value other than 1 for \( x_{n+1} \) in all \( i \) cases, because \( x_{n+1} \) is not part of the investor’s betting strategy, rather a result from it. That is, he or she cannot make a bet on the amount of arbitrage to be gained directly, therefore the coefficient value for \( x_{n+1} \) cannot be flexible as it is for
the rest of the $x_n$. Additionally, the amount gained through arbitrage should remain the same for all $i$ outcomes, by the definition of arbitrage, meaning it is necessary to have the same coefficient value for $x_{n+1}$ in all outcomes. Both of these objectives are neatly achieved by choosing $a_{i(n+1)} = 1$. The reason 1 is chosen as the coefficient, rather than some other value such as .5 or 2 is because $2x_{n+1}$ can be expressed more simply by just setting $x_{n+1}$ equal to this value. For example, say $x_{n+1} = 5$. Then setting $a_{i(n+1)} = 2$ is like saying that $2x_{n+1}$, or $2(5) = 10$ is gained through arbitrage in any case. However, one could simply let $x_{n+1} = 10$ and avoid this circumstance. In the end, this allows for cleaner, simpler, and more elegant equations. Using this information, (5) is now rewritten as follows:

$$\text{maximize } x_{n+1}$$

subject to

$$\sum_{j=1}^{n} a_{ij} x_j + x_{n+1} \leq 0 \text{ for } i = 1, \ldots, m.$$  \hspace{1cm} (6)

It is helpful for some to visualize this in matrix form. Observe that since $i = 1, \ldots, m$ and $j = 1, \ldots, n$, the summation above of $a_{ij}$ creates a coefficient matrix $A$ that is $m \times (n + 1)$, while the vector $x$ has $n + 1$ entries and solution vector $b$ has $m$ entries.

$$\begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_n \\
    x_{n+1}
\end{bmatrix}
\begin{bmatrix}
    a_{11} & a_{12} & \cdots & a_{1n} & 1 \\
    a_{21} & a_{22} & \cdots & a_{2n} & 1 \\
    \vdots & \vdots & & \vdots & \vdots \\
    a_{m1} & a_{m2} & \cdots & a_{mn} & 1
\end{bmatrix}
\begin{bmatrix}
    x_1 \\
    x_2 \\
    \vdots \\
    x_n \\
    x_{n+1}
\end{bmatrix}
\leq
\begin{bmatrix}
    0 \\
    0 \\
    \vdots \\
    0 \\
    0
\end{bmatrix}$$

Next the dual program of (6) is found. A simplified method of doing so is provided by Hadley: for any standard-form LP problem:

$\text{Ax} \leq a$ for vector $x$ with only non-negative entries, seeking to maximize $c = b'x$

Its dual can be written as:

$\text{A'y} \geq b$ for vector $y$ with only non-negative entries, seeking to minimize $d = a'y$.\[11\]

Hadley's general rule is useful for understanding the relationship between primal and dual equations. Note, however, that the problem (6) above has unrestricted variables.
Referring again to Table 3, this implies that free variables in the primal mandate equality constraints in the dual. Following this logic, to find the dual program of (6), take the transposes of the appropriate matrices above, find the fitting linear constraints, and choose \( y_1, \ldots, y_m \) such that (7) below is the equivalent dual of (6):

\[
\begin{align*}
\text{minimize} & \quad 0 \\
\text{subject to} & \\
\sum_{i=1}^{m} a_{ji} y_i &= 0 \quad \text{for } j = 1, \ldots, n. \\
\sum_{i=1}^{m} a_{(n+1)i} y_i &= 1 \\
y_i &\geq 0 \quad \text{for } i = 1, \ldots, m.
\end{align*}
\]

Expressed in matrix form, the relation between (6) and (7) becomes more apparent:

\[
\begin{bmatrix}
0 \\
y_1 \\
y_2 \\
\vdots \\
y_{m-1} \\
y_m
\end{bmatrix}
\begin{bmatrix}
a_{11} & a_{21} & \cdots & a_{m1} \\
a_{12} & a_{22} & \cdots & a_{m2} \\
\vdots & \vdots & \ddots & \vdots \\
a_{1n} & a_{2n} & \cdots & a_{mn} \\
1 & 1 & \cdots & 1
\end{bmatrix}
\begin{bmatrix}
y_1 \\
y_2 \\
\vdots \\
y_{m-1} \\
y_m
\end{bmatrix}
= \begin{bmatrix}
0 \\
0 \\
\vdots \\
0 \\
1
\end{bmatrix}
\]

Again, the matrix format is helpful for visualizing the problem and understanding its meaning. When looking at the matrix equation, it becomes clear that the dual is feasible if and only if \( y \) is a vector for which the expected return is 0 for all possible outcomes \( i = 1, \ldots, m \). The last row of the matrix equation (7) indicates that the sum of all \( y_i \) is equal to 1. Because of this, and the fact that \( y_i \geq 0 \forall i = 1, \ldots, m \), it can be observed that \( y \) is a probability vector; each probability must be non-negative and the sum of the probabilities all possible outcomes must be 1. With this context, the last row of the matrix equation (7) makes sense, and is consistent with the definition of the probability vector \( p \), given in The Arbitrage Theorem.

So suppose that the dual is feasible. Then by the Duality Theorem, the primal is also feasible because there exists at least one solution of \( x \) that satisfies the linear constraints
specified. That is, at least the trivial solution \( x_j = 0 \) for all \( j = 1, \ldots, n, n+1 \) conforms to the linear constraints of both the primal and the dual. Then, again by the Duality Theorem of Linear Programming, the optimal solution (minimum) of the dual is equal to the maximum of the primal. The minimum value of the dual in any case is equal to 0 because the coefficients in the objective function are all equal to 0. That is, for any probability vector \( y \) for which the dual is feasible, the numerical value of the dual will be 0, meaning that the optimal value of the dual is 0. This fact implies that the maximum optimal value of the primal is also zero, meaning that \( x_{n+1} \), or the amount the investor is sure to win, is 0. This means there is no opportunity for arbitrage.

Alternately, suppose the dual is not feasible. Then by the Duality Theorem, the primal has no optimal solution. This implies that the maximum value of \( x_{n+1} \neq 0 \), which implies the possibility of a positive payoff from a betting strategy, regardless of outcome. That is, if the maximum value of the primal is not zero, this implies there is a value for \( x_{n+1} \) that is greater than zero. This is true because if the only feasible values of \( x_{n+1} \) in the primal were less than zero, then zero would be chosen as the maximum feasible value of the function. It is key to remember that a value of \( x_{n+1} \) can always be achieved by setting all \( x_n \) equal to zero, that is, by not making any bets at all. This implies that if zero is not the maximum, then the maximum value of \( x_{n+1} \) must be greater than zero.

Since the dual is feasible only when there is a \( y \) for which the expected return is 0 for each possible outcome \( i = 1, \ldots, m \), this implies that if such a \( y \) does not exist, then there exists at least one betting strategy \( x \) for which \( x_{n+1} > 0 \), i.e. there exists at least one opportunity for arbitrage.

\[ \therefore \text{The arbitrage theorem has been proven.} \]

### 2.3 Formulating a Currency Arbitrage Linear Program

With knowledge of the basic concepts and terminology of Linear Programming, it becomes possible to apply those concepts to the currency arbitrage problem. Recall that the goal of this work is to generate US Dollars by exchanging currencies in a circular pathway. To clarify, the investor hoping to find an arbitrage, or arbitrageur, would begin with a certain amount of US dollars and execute currency trades in such a fashion as to end up with more USD than (s)he began with. This process will become clearer through calculations.

Using the following table of exchange rates, a Linear Programming problem is formulated by following the steps for LP formation, outlined by Dantzig.\(^5\)

Using this exchange rate data, Dantzig's steps are now examined and a linear program is formed.[10]

**Step 1: Define the Activity Set**

The first step involves decomposing the system being studied into its most elementary

---

\(^5\)The rates provided in the table are mid-market exchange rates and were obtained at 5 PM on February 14, 2013 from Oanda.com, an online currency exchange platform.
functions. These will be called activities. A unit for each activity must also be chosen so that its quantity can be measured. In the currency arbitrage detection problem, or currency problem for short, each exchange is an elementary function. That is, an exchange USD → EUR is one activity, while EUR → JPY is another activity. These activities can be measured in the quantity of the base currency that is exchanged. So the EUR → JPY activity is measured in terms of how many Euros are exchanged for Yen. As the number of currencies being examined increases, this type of notation becomes cumbersome. Therefore, each currency is given a number that represents it:

<table>
<thead>
<tr>
<th>Currency</th>
<th>Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>USD</td>
<td>0</td>
</tr>
<tr>
<td>EUR</td>
<td>1</td>
</tr>
<tr>
<td>JPY</td>
<td>2</td>
</tr>
<tr>
<td>GBP</td>
<td>3</td>
</tr>
<tr>
<td>CHF</td>
<td>4</td>
</tr>
</tbody>
</table>

The order in which the numbers were assigned based on the currencies' respective rankings in exchange volume. This indicates that USD is the most highly traded currency in the forex market, with EUR in second place, JPY in third place, and so on.[15]

After assigning numerical values to each currency, it becomes much simpler to define each activity. Let \( x_{ij} \) represent an exchange from currency \( i \) into currency \( j \). Then \( x_{01} \) denotes the exchange from USD to EUR, \( x_{10} \) from EUR to USD, and so on.

**Step 2: Define the Item Set**

The second step in forming an LP problem involves determining the classes of objects that are consumed or produced by the activities, called items, and choosing their units of measure. In the currency problem, the items produced or consumed by each currency exchange are simply the currencies themselves. Concretely, an exchange USD → EUR (\( x_{01} \)) consumes USD and produces EUR. Similarly, the units of measure are simply the amount of each currency possessed.
Step 3: Determine the Input-Output Coefficients
In this step of the LP problem formulation, the quantity of each item consumed or produced by each activity at unit level is determined. This step is easy to complete in the currency problem, because the exchange rates between various currencies, or cross rates, take on the coefficient values. Referring again to the USD→EUR ($x_{01}$) exchange and the rates displayed in Table 4, the arbitrageur will obtain .7431 Euros for each US Dollar exchanged. Thus, if the arbitrageur began with 2 USD, an equation describing the amount of USD the arbitrageur owns after exchanging 1 USD for EUR would be: 1USD+.7431EUR.

Step 4: Determine the Exogenous Flows
The exogenous flows of the LP problem are the net inputs or outputs of the items, derived from performing activities within the system. In the currency problem, the arbitrageur begins with USD and will end up with USD, though the amounts with which he or she begins and ends the exchange will not be equivalent (assuming an opportunity for arbitrage exists).

Let $x_0$ represent the initial amount of USD, or the total USD that flows into the system, and $y$ represent the total amount of USD after all exchanges, or the amount that flows out of the system. If arbitrage is successful, then $y \geq x_0$. It is helpful to think of $y$ as being made up of two components: $x_0$ and whatever profit is generated through the system, called $p$. So $y = x_0 + p$. While it is the goal of the arbitrageur to maximize $p$, it is easier in computations to record the total inflow and outflow from the system, and to calculate any earnings from arbitrage afterwards. Note that mathematically, maximizing $y$ will have the same effect as maximizing $p$, since $y = x_0 + p$ and once a value for $x_0$ has been determined, it remains constant. Therefore, the total USD inflow to the system is $x_0$ and the outflow is $y$.

The inputs and outputs of the other currencies must be considered and assigned values as well. The arbitrageur in the currency problem begins with no other currencies and does not wish to end his or her exchanges with any other currencies. This indicates that the total input for EUR, JPY, GBP, and CHF will be 0. Similarly, the total output for each of these currencies will also be 0. Note that inputs and outputs of 0 do not imply the amount of each currency held must always be 0. Indeed, the entire currency problem would be rather pointless if no other currencies could be held in the course of an exchange cycle. An exogenous flow value of 0 only implies that the arbitrageur neither begins nor ends his or her currency exchanges with holdings of any of these currencies.

Step 5: Determine the Material Balance Equations
The last step is to use the variables assigned to the activities and the coefficients determined above to create material balance equations for each item. These equations should ensure that the flows into that item determined by each activity and its coefficient equal its outflows. In the currency problem, exchanges between currencies should preserve the value of money. For this to be true, the total amount of each currency available should equal the total amount of that currency that has been distributed.
Using USD as the example item, a material balance equation can be written as such:

Total amount of USD available = total USD converted from other currencies [16]  
The left-hand side of the equation, the total amount of USD available, can be broken down into the following components: initial dollar input and dollars converted from other currencies. The right-hand side of the equation, total amount of USD converted from other currencies, can be broken down as such: the final dollar holdings and dollars distributed to other currencies. [16] Incorporating these components, the equation is rewritten:

Initial dollar input + dollars converted from other currencies = final dollar holdings + dollars distributed to other currencies. Substituting in the activity variables, exogenous flow variables, and coefficients derived in the previous steps, a true equation can be written as such:

\[ x_0 + (1.3459x_{10} + 0.010713x_{20} + 1.562x_{30} + 1.09x_{40}) = y + x_{01} + x_{02} + x_{03} + x_{04} \]

Rewriting this so that \( x_0 \) is isolated on the right-hand side of the equation yields the following:

\[ y + x_{01} + x_{02} + x_{03} + x_{04} - (1.3459x_{10} + 0.010713x_{20} + 1.562x_{30} + 1.09x_{40}) = x_0 \]  

Note again that the coefficients for \( x_{ij} \) for \( i = 1, \ldots, 4 \) used in (8) are the exchange rates from the listed base currencies to the dollar. The coefficients for all exchanges \( x_{ij} \) for \( j = 1, \ldots, 4 \) in are 1 because any exchange originating in dollars consumes only one dollar at unit-level. The other equations in the system are formulated using the same method as (8), but with the appropriate exchange rates. These are displayed below:

\[ x_{10} + x_{12} + x_{13} + x_{14} - (0.7431x_{01} + 0.00796x_{21} + 1.1607x_{31} + 0.81x_{41}) = 0 \]  
\[ x_{20} + x_{21} + x_{23} + x_{24} - (93.359x_{02} + 125.642x_{12} + 145.82x_{32} + 101.762x_{42}) = 0 \]  
\[ x_{30} + x_{31} + x_{32} + x_{34} - (0.6403x_{03} + 0.8617x_{13} + 0.006859x_{23} + 0.6979x_{43}) = 0 \]  
\[ x_{40} + x_{41} + x_{42} + x_{43} - (0.9176x_{04} + 1.2349x_{14} + 0.00983x_{24} + 1.4332x_{34}) = 0 \]  

The set of equations (8)-(12) are the material balance equations for the currency linear program. Note that the variables \( x_{ij} \geq 0 \), since there cannot be a negative amount of currency converted.

**The Currency Problem Linear Program**

Using the sets of equations obtained through the 5 steps of Linear Program formation, the LP can be written for the currency problem:

\[ \text{Max } z = y \]  

(13)
subject to

\[
\begin{align*}
y + x_{01} + x_{02} + x_{03} + x_{04} - (1.3459x_{10} + 0.010713x_{20} + 1.562x_{30} + 1.09x_{40}) & = x_0 \\
x_{10} + x_{12} + x_{13} + x_{14} - (0.7431x_{01} + 0.007962x_{21} + 1.1607x_{31} + 0.81x_{41}) & = 0 \\
x_{20} + x_{21} + x_{22} + x_{24} - (93.359x_{02} + 125.642x_{12} + 145.82x_{32} + 101.762x_{42}) & = 0 \\
x_{30} + x_{31} + x_{32} + x_{34} - (0.6403x_{03} + 0.8617x_{13} + 0.006859x_{23} + 0.6979x_{43}) & = 0 \\
x_{40} + x_{41} + x_{42} + x_{43} - (0.9176x_{04} + 1.2349x_{14} + 0.00983x_{24} + 1.4332x_{34}) & = 0
\end{align*}
\]

for \( x_{ij} \geq 0, \ i = 0, \ldots, 4 \) and \( j = 0, \ldots, 4 \)

The coefficient values above are put into matrix form. The Matrix A, below, represents the coefficients on the left-hand side of equations (8)-(12). The vector \( \mathbf{b} \) represents the right hand side of the equations (8)-(12).

\[
\mathbf{A} = \begin{bmatrix}
1 & 0.1 \cdot 0.7431 & 0 & 0 & 0 & \cdots \\
0 & 0 & -0.8359 & 0 & 0 & \cdots \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots \\
0 & 0 & 0 & 0 & 0 & \cdots \\
\end{bmatrix} \quad \mathbf{y} = \begin{bmatrix}
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\end{bmatrix} \quad \mathbf{z}_0 = \begin{bmatrix}
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\cdots \\
\end{bmatrix}
\]

A few important caveats to this model exist. The first is that mid-market exchange rates were used, which ignore the bid-ask spread. Recall that the bid-ask spread represents the cost to trade with a market maker. Additionally, most dealers on the forex charge a small fee or a commission for transactions made, on top of the bid-ask spread. This could significantly alter the amount of profit generated or favor optimal solutions with minimal number of trades, so as to minimize transaction fees. The Currency LP formulated above does not account for transaction cost, as the primary focus of this paper is not to actually generate profit or to develop a sustainable and fast-working method for detecting arbitrage using real-time rates. For the purposes of this document, the above LP will suffice for an investigation into the field Linear Programming, and a general investigation of arbitrage opportunities in currency markets.

The linear program formulated above is solved using the Microsoft Excel LP solver. Many more complex LP solvers are available in the free web, as well as part of mathematical software packages such as Matlab, Mathematica, R, and so on. While these calculations can be carried out by a wide variety of computer programs, they all generally use the same solving method: The Simplex Method. This method was initially developed by Dantzig, and has since been revised and modified, but the basic principles of solving remain the same. The process is outlined in the next section.

### 2.4 The Simplex Method

The Simplex Method is an iterative solving process that relies on principles of Linear Algebra to find solution sets for systems of linear equations subject to the program's objective. The process begins with a linear programming problem that is presented in
standard form. Recall the standard form LP, equation (3) from section 2.2:

\[
\text{maximize } \sum_{j=1}^{n} c_j x_j
\]

subject to

\[
\sum_{j=1}^{n} a_{ij} x_j \leq b_i \text{ for } i = 1, \ldots, m.
\]

\[
x_j \geq 0 \text{ for all } j = 1, \ldots, n.
\]

The first step to solving this equation is the introduction of slack variables to transform the set of linear inequality constraints into linear equality constraints, which was briefly mentioned in section 2.2.[14] Consider a basic inequality:

\[
a_1 x_1 + a_2 x_2 + \cdots + a_n x_n \leq b
\]

This can be re-written in the following form:

\[
x_{n+1} = b - (a_1 x_1 + a_2 x_2 + \cdots + a_n x_n), \text{ where } x_{n+1} \text{ represents the difference between the } a_j x_j \text{ for } j = 1, \ldots, n \text{ and } b.
\]

Re-arranging this equation once more, it appears as follows:

\[
a_1 x_1 + a_2 x_2 + \cdots + a_n x_n + x_{n+1} = b
\]

Here it becomes apparent that the initial linear inequality has been converted into an equality by the introduction of the slack variable \(x_{n+1} \).[14] Applying this logic to the entire standard form linear program above, a new system of equations emerges:

\[
\text{maximize } c_1 x_1 + c_2 x_2 + \cdots + c_n x_n + 0 x_{n+1} + \cdots + 0 x_{n+m}
\]

subject to

\[
\begin{align*}
    a_{11} x_1 + a_{12} x_2 + \cdots + a_{1n} x_n + x_{n+1} &= b_1 \\
    a_{21} x_1 + a_{22} x_2 + \cdots + a_{2n} x_n + x_{n+2} &= b_2 \\
    a_{i1} x_1 + a_{i2} x_2 + \cdots + a_{in} x_n + x_{n+i} &= b_i \\
    a_{m1} x_1 + a_{m2} x_2 + \cdots + a_{mn} x_n + x_{n+m} &= b_m
\end{align*}
\]

Where \(x_1, \ldots, x_n\) are called the structural variables and \(x_{n+1}, \ldots, x_{n+m}\) are the slack variables. In the above set of equations, \(a_{ij}\) are the structural coefficients, which correspond to the structural variables, and the \(b_i\) are the upper bounds, which set the maximum values for each of the \(m\) equations.[14] Note in the objective function, all the slack variables all have coefficients of 0. This is because the values of \(x_{n+1}, \ldots, x_{n+m}\) are irrelevant to the solution of the problem, since the variables are meaningless. Thus,
when finding feasible solutions for the problems, only the values of the basic variables are considered as relevant to the original problem. However, having the equations in this form makes finding feasible solutions much easier, therefore it is a useful tool in solving LP problems.[14]

Once an equation is in this form, the simplex algorithm can proceed. Note that the above set of equations will contain m equations with n + m unknowns. So it's possible to set a number of these variables equal to zero and attempt to solve for the remaining variables. Every combination of variables chosen to be set to zero represents a potential corner or a fundamental system of equations. When the system of equations found at a potential corner has a unique solution, it is called a corner or a basic solution. The corners of a system represent the so-called edges of the region of feasible solutions, graphically speaking. At a corner, the variables whose value are zero are called corner variables or nonbasic variables, while the nonzero variables are the basic variables. Each corner can be expressed by rewriting the objective function and basic variables in terms of the nonbasic variables. This expression is called a dictionary and is unique to that corner. Any corner at which each basic variable is non-negative is called a feasible corner. At a feasible corner, the nonbasic and basic variables form a basic feasible solution.[14] With this terminology and basic knowledge, the simplex algorithm can be summarized in three steps, as given by Strum in his text:

Step 1: Adjust the dictionary
Here the term adjusting refers to solving equations of basic feasible solutions for the nonbasic variables so that the objective function and basic variables are expressed in terms of the nonbasic variables. Once this has been done, the system has been adjusted.

Step 2: Examine the objective function as expressed in the current dictionary
Let \( s_1, \ldots, s_n \) be the nonbasic variables and the objective function be:

\[
P = P_0 + k_1 s_1 + k_2 s_2 + \cdots + k_n s_n
\]

Examine this equation and ascertain which of the following applies:

1. If \( k_i < 0 \) for all \( i = 1, \ldots, n \) then the algorithm is terminated.
2. If no \( k_i \) is strictly positive, but some \( k_i = 0 \), the current corner is an optimal corner, though there may be alternative optimal solutions.
3. If some of the \( k_i \)'s are positive, then increase any \( s_i \) whose coefficient \( k_i \) is largest.

If there does not exist a limit on how much we can increase \( s_i \), then there are unbounded feasible solutions. If this is not the case, increase the \( s_i \) to the maximum extent allowed by the constraints, through which some other variables will necessarily be reduced to 0, and a new feasible corner will be reached.

Step 3: Repeat Step 1
The algorithm described in the first two steps will converge in almost all cases.\(^6\)

\(^6\)The cases in which this algorithm does not converge are degenerate or nonconvergent cases. For more information, refer to Strum.
This basic explanation of the simplex method is sufficient to understand the theoretical approach used by computer solving systems. Essentially, the most extreme solutions of the system are tested until an optimal solution is found. This is done through the process of finding feasible corners, which are the extremes of the feasible region, and testing them to find the optimal corner for maximizing or minimizing the objective function. More detailed explanations and examples of the simplex method are provided by Strum, Dantzig, and Chvátal.

2.5 The Currency Linear Program: Solutions and Interpretations

Recall the currency linear program formed in section 2.3:

$$\text{Max } z = y$$

subject to

$$y + x_{01} + x_{02} + x_{03} + x_{04} - (1.3459x_{10} + 0.010713x_{20} + 1.562x_{30} + 1.09x_{40}) = x_0$$

$$x_{10} + x_{12} + x_{13} + x_{14} - (0.7431x_{01} + 0.007966x_{21} + 1.1607x_{31} + 0.81x_{41}) = 0$$

$$x_{20} + x_{21} + x_{23} + x_{24} - (93.359x_{02} + 125.642x_{12} + 145.82x_{32} + 101.762x_{42}) = 0$$

$$x_{30} + x_{31} + x_{32} + x_{34} - (0.6403x_{03} + 0.8617x_{13} + 0.006859x_{23} + 0.6979x_{43}) = 0$$

$$x_{40} + x_{41} + x_{42} + x_{43} - (0.9176x_{04} + 1.2349x_{14} + 0.00983x_{24} + 1.4332x_{34}) = 0$$

for $$x_{ij} \geq 0, i = 0, \ldots, 4$$ and $$j = 0, \ldots, 4$$

Note that this linear program would be unbounded if there were no upper bound constraints placed on the $$x_{ij}$$. This is because if there is any amount of profit to be gained through an exchange cycle, one could theoretically trade in a cyclical pattern so that the profit generated through arbitrage approached infinity. Therefore, either bounds are placed on the number of cycles that can be made, or else the amount of any one currency that can be traded at a time. It is most realistic to limit the amount of any one currency that can be traded at a time, as this is something that occurs in practice. A daily trading limit is the amount of any given currency that can be traded on the forex market until the quoted exchange rate will change. These limits are set by market makers such as large banks or treasuries, who hold large currency reserves.[15]

In order to explore the affect the limits have on the outcome of the problem, the LP (13) has been solved six times, with different daily trading limits assigned to each currency. The units of the LP are millions, and in each case, the total input was selected as $$x_0 = 10$$, implying a total initial input of 10 million USD. This amount was chosen because it is a round figure larger than any one single constraint, and because it seems feasible that a single wealthy individual might have this sum of money to exchange. The LP was solved using Microsoft Excel Linear Program solver because it
is a relatively small data set and could easily be processed by this solving software. As was previously indicated, most solvers use some method of the Simplex Method; Microsoft Excel Solver uses the Revised Simplex Method, which is a faster and more efficient version of the Simplex Method detailed in Section 2.4 of this paper.\(^7\) It was found that the six sets of constraints produced only four distinct solution sets. These solutions and their interpretations follow below:

**Solution 1**

The model was initially solved with the following constraints:

\[
\begin{align*}
    x_{0j} &\leq 5 \text{ for } j = 1, 2, 3, 4 \\
    x_{1j} &\leq 5 \text{ for } j = 0, 2, 3, 4 \\
    x_{2j} &\leq 100 \text{ for } j = 0, 1, 3, 4 \\
    x_{3j} &\leq 5 \text{ for } j = 0, 1, 2, 4 \\
    x_{4j} &\leq 5 \text{ for } j = 0, 1, 2, 3 
\end{align*}
\]

Recall that the LP is measured in millions. So the constraint \(x_{0j} \leq 5 \text{ for } j = 1, 2, 3, 4\) implies that a maximum of 5 million USD may be traded for any of the other currencies in the model at any one time. The constraint \(x_{2j} \leq 100 \text{ for } j = 0, 1, 3, 4\) implies a maximum trading limit of 100 million Yen to any other currency in the model, and so on. With these constraints, the solution vector \(\mathbf{x}\) was as follows:

\[
\mathbf{x} = \begin{bmatrix}
10.00652362 \\
0.00000000 \\
3.244274381 \\
0.00000000 \\
3.84008550 \\
4.72891064 \\
0.982985089 \\
0.00000000 \\
0.00000000 \\
0.00000000 \\
0.00000000
\end{bmatrix}
\]

Interpreting the values given in the vector as values assigned to variables gives this solution meaning:

---

\(^7\)Other proprietary LP solving packages can be found in MATLAB and Mathematica. Open-source and free solvers include glpk, a GNU LP kit, and R-project
Indeed, the above matrix equation makes the solution set much easier to understand. It can be seen that \( y = 10.00652562 \). This implies that the total outflow from the system is $10,006,525.62. Recall that the total input, \( x_0 \) was only $10 million. Recall once more that \( y = x_0 + p \) where \( p \) is the profit generated through arbitrage; thus, \( p = 10006525.62 - 10000000 = 6525.62 \), so the total profit is $6,525.62.

The exchange pattern implied by the variable assignments in the solution vector above is shown in Figure 1: Referring again to the solution vector above, it can be observed that several of the exchanges are at the maximum trade limits. This implies that the relative complexity of the solution provided above, i.e. the number of circular trade patterns observed in Figure 1, is due to the limits imposed on the linear program. This, in turn, implies that perhaps increasing the trade limits would lead to a simpler solution that would involve fewer trades overall. However, the next two solution sets of the program, which have higher trade limits, resulted in solutions with the same number or more trades overall. These will be discussed below.

Solution 2
Next, it is considered what happens when all the trading limits are raised. Each of the stronger currencies’ limits were increased moderately, while the JPY limit was increased rather substantially. This set of solutions includes two cases, in which two
different sets of trade maximums produced the same overall trading pattern, but with different values assigned to each nonzero variable.

Case 1 Observe the Currency LP with the following restraints:

\[
\begin{align*}
\text{x}_{0j} & \leq 7 \text{ for } j = 1, 2, 3, 4 \\
\text{x}_{1j} & \leq 6 \text{ for } j = 0, 2, 3, 4 \\
\text{x}_{2j} & \leq 300 \text{ for } j = 0, 1, 3, 4 \\
\text{x}_{3j} & \leq 6 \text{ for } j = 0, 1, 2, 4 \\
\text{x}_{4j} & \leq 7 \text{ for } j = 0, 1, 2, 3 \\
\end{align*}
\]

This LP produced the following results:

\[
\begin{bmatrix}
\text{y} \\
\text{z}_0 \\
\text{z}_2 \\
\text{z}_3 \\
\text{z}_4 \\
\text{z}_6 \\
\text{z}_8 \\
\text{z}_{10} \\
\text{z}_{20} \\
\text{z}_{30} \\
\text{z}_{40} \\
\text{z}_{42} \\
\text{z}_{44}
\end{bmatrix}
= 
\begin{bmatrix}
10.00904006 \\
0.4910259947 \\
300.00 \\
0.00 \\
4.91959147 \\
0.00 \\
0.00 \\
0.00 \\
0.00 \\
0.00 \\
0.00 \\
0.00 \\
0.00 \\
0.00
\end{bmatrix}
\]

This set of constraints and solutions will be referred to in other sections as Solution 2.1. Observe that in this solution set, \( y = 10.00904006 \). Let the profit for Case 1 be denoted as \( p_1 \) implying a profit \( p_1 = 9040.06 \).

Observe Case 2 below, in which the trade limits have been raised yet again:

\[
\begin{align*}
\text{x}_{0j} & \leq 8 \text{ for } j = 1, 2, 3, 4 \\
\text{x}_{1j} & \leq 7 \text{ for } j = 0, 2, 3, 4 \\
\text{x}_{2j} & \leq 700 \text{ for } j = 0, 1, 3, 4 \\
\text{x}_{3j} & \leq 6 \text{ for } j = 0, 1, 2, 4 \\
\text{x}_{4j} & \leq 9 \text{ for } j = 0, 1, 2, 3 \\
\end{align*}
\]

With the solution:

\[
\begin{bmatrix}
\text{y} \\
\text{z}_0 \\
\text{z}_2 \\
\text{z}_3 \\
\text{z}_4 \\
\text{z}_6 \\
\text{z}_8 \\
\text{z}_{10} \\
\text{z}_{20} \\
\text{z}_{30} \\
\text{z}_{40} \\
\text{z}_{42} \\
\text{z}_{44}
\end{bmatrix}
= 
\begin{bmatrix}
10.0197453 \\
8.00 \\
8.00 \\
8.00 \\
700.00 \\
6.3051425 \\
6.199 \\
7.00 \\
0.00 \\
0.00 \\
0.00 \\
0.00 \\
0.00 \\
0.00
\end{bmatrix}
\]
This set of constraints and solutions will be referred to in other sections as Solution 2.2. Note that the solution vector for Case 2 implies the same general exchange pattern as Case 1. However, Case 2 has a profit $p_2 = 10974.53$. Since both of these cases follow the same currency exchange pattern, i.e. the same variables in each case are assigned a value of zero, it appears the primary difference between the two is the values assigned to each nonzero variable, including $y$. The discrepancy in the $y$ variables in Case 1 and Case 2 implies a different in profit between the two cases. The overall difference in profit between Case 1 and Case 2 is $p_2 - p_1 = 10974.53 - 9040.06 = 1934.47$. Note that both of these profit values are also higher than that generated by Solution 1. It appears, then, that increasing trade limits increases the opportunity for profit-generating exchange. This will be further explained in subsequent solution sets.

Some important observations can be made about both cases detailed above. The most important observation is that both of these cases produce the same general currency exchange paths. The pattern of exchanges implied by the variable assignments in Cases 1 and 2 is seen in Figure 2. Note that Figure 2 differs from Figure 1 only in that an exchange path from GBP to JPY has been added. Otherwise, the exchange patterns in Solutions 1 and 2 are exactly the same. This begs the question as to why this exchange path has been added to the optimal solution set. Both the constraints on GBP exchanges and JPY have been increased in both cases of Solution 2; however (as will be seen in Table 5), neither Case 1 or Case 2 maximizes $x_{32}$ (GBP→JPY). Notably, $x_{20}$ (JPY→USD) is maximized in both cases - which was also true of Solution 1. This fact leads to the conclusion that the increase in the JPY trading limit demanded that more funds be transferred to Yen to enable the maximum value of $x_{20}$ possible within the constraints of the LP. This conclusion implies that the JPY→USD trade is particularly profitable and should be taken in all solution sets. It will be seen later that this is true.

The second important observation is that an increase in the trade limits leads to an increase in profit from arbitrage. The results outlined in Table 5 lead to the question of which particular exchanges in trade maximums actually caused the increase in profits. This question leads to the third important consideration: which of the variables in each case is actually designated to
be at the trade maximum. Table 6 lists the nonzero variables for Solution 2. A “Yes” indicates that the given variable was at its trade maximum in the solution, while a “No” indicates that it was not at the trade maximum.

<table>
<thead>
<tr>
<th>Variable</th>
<th>At Maximum Trade Limit?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Case 1</td>
</tr>
<tr>
<td>$x_{01}$</td>
<td>Yes</td>
</tr>
<tr>
<td>$x_{03}$</td>
<td>No</td>
</tr>
<tr>
<td>$x_{04}$</td>
<td>Yes</td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>Yes</td>
</tr>
<tr>
<td>$x_{20}$</td>
<td>Yes</td>
</tr>
<tr>
<td>$x_{40}$</td>
<td>Yes</td>
</tr>
<tr>
<td>$x_{13}$</td>
<td>No</td>
</tr>
<tr>
<td>$x_{14}$</td>
<td>Yes</td>
</tr>
<tr>
<td>$x_{31}$</td>
<td>Yes</td>
</tr>
<tr>
<td>$x_{41}$</td>
<td>Yes</td>
</tr>
<tr>
<td>$x_{32}$</td>
<td>No</td>
</tr>
<tr>
<td>$x_{42}$</td>
<td>No</td>
</tr>
<tr>
<td>$x_{34}$</td>
<td>Yes</td>
</tr>
<tr>
<td>$x_{43}$</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Table 6: Variables at Maximum Trade Limits for Cases of Solution 2

There are only two variables for which Cases 1 and 2 are not in alignment, that is, there are only two occurrences where a variable is at its trade maximum in one case and not the other. These have been emphasized in Table 6 above, and occur at $x_{03}$ and $x_{40}$. These variables represent the exchanges USD→GBP and CHF→USD, respectively. The variables for which both cases are in alignment seem to indicate profitable trades, which should be maximized regardless. In order to better understand which trades are truly profitable, other solutions with different trade limits are explored.

Solution 3
It can be observed that in the previous solution set, the variable $x_{20}$ (JPY→USD) was always at maximum. Therefore a new solution set was explored in which all of the limits except JPY are the same as Solution 2.2. Based on the observations about the profitability of the JPY→USD exchange path, one would expect an increase in funds
allocated to JPY, and a subsequent increase in profits. The limits for this solution set are shown below.

\[ x_{0j} \leq 8 \text{ for } j = 1, 2, 3, 4 \]
\[ x_{1j} \leq 7 \text{ for } j = 0, 2, 3, 4 \]
\[ x_{2j} \leq 2000 \text{ for } j = 0, 1, 3, 4 \]
\[ x_{3j} \leq 6 \text{ for } j = 0, 1, 2, 4 \]
\[ x_{4j} \leq 9 \text{ for } j = 0, 1, 2, 3 \]

With the solution:

\[
\begin{bmatrix}
    0.01208259 \\
    8 \\
    8 \\
    8 \\
    2.5220295 \\
    2000 \\
    0 \\
    6.5843 \\
    7 \\
    5.8437306 \\
    7 \\
    0 \\
    5.1543 \\
    9 \\
    9 \\
    9 \\
    9 \\
    0 \\
    0 \\
    0 \\
    0 \\
    0 \\
    0
\end{bmatrix}
\]

This solution set leads to several important observations. The first is that the currency exchange path is different than the preceding two solution sets. This indicates that increasing the JPY exchange limit does indeed have a tangible effect on the currency exchanges taken in the optimal LP. Additionally, increasing only the JPY exchange limit, while holding the others constant, increases the overall profit earned from the exchanges — as was expected. For this solution, \( y = 10.01208259 \), indicating a profit \( p = 12082.59 \), which represents a $1,108.06 increase from Solution 2.2.

Figure 3 below shows the exchange path taken in this model: Some notable differences from Figures 1 and 2 include the addition of nonzero values for the variables \( x_{2,0} \) and \( x_{1,2} \), as well as the elimination of variable \( x_{4,3} \). This denotes the addition of exchanges USD→JPY and EUR→JPY, and the elimination of the exchange CHF→GBP.

Figure 3: Currency Exchange Pattern for Solution 3
supports the observation above that the increase in the trading limit for JPY leads the model to reallocate funds to JPY, since two exchanges to the yen were added in this more profitable solution set.

Solution 4
The previous solution sets point to the general fact that trading limits constrain the amount of profit that can be achieved through currency trading. Based on observations of previous models, increasing trading limits increases the output of the currency exchange. This leads to the question of what happens when limits are completely removed.

As was previously stated, when there are no trading limits on the model, the feasible region of the Currency LP is unbounded, meaning that the amount of output generated by the system grows infinitely. Additionally, imposing only one bound on outgoing USD has the same result: the possible profit generated through arbitrage is unbounded. This is because there are profitable exchanges between the other unbounded currencies that can generate profit forever. The bounds for Cases 1 and 2 of this solution set were obtained by beginning with all 5 currencies bounded and removing as many bounds as possible, while still producing a feasible solution. It was found that trading limits imposed on only USD and EUR consistently generated a feasible and optimal solution.

Case 1 of this solution set has the following bounds:

\[ x_{0j} \leq 7 \text{ for } j = 1, 2, 3, 4 \]
\[ x_{1j} \leq 6 \text{ for } j = 0, 2, 3, 4 \]

For the sake of consistency, upper trading limits for USD and EUR were chosen from existing solution sets. The constraints chosen here correspond to those in Solution 2.1. The limits in Case 1 led to the following solution:

\[
\begin{bmatrix}
 y \\
 z_{01} \\
 z_{02} \\
 z_{03} \\
 z_{04} \\
 z_{10} \\
 z_{20} \\
 z_{21} \\
 z_{24} \\
 z_{22} \\
 z_{30} \\
 z_{31} \\
 z_{34} \\
 z_{43} \\
 z_{44}
\end{bmatrix}
= \begin{bmatrix}
 10.01088083 \\
 7 \\
 7 \\
 0 \\
 2614.6262637 \\
 0 \\
 6 \\
 6 \\
 5.88709901 \\
 0 \\
 0 \\
 10.27122505 \\
 3.79523099
\end{bmatrix}
\]

The first observation about this solution set is that there are far fewer nonzero variables. Additionally, certain variables, most notably \( z_{20} \) and \( z_{42} \) have values far higher than any previous trading limit would have allowed. This highlights the fact that the \( z_{42} \), or CHF→JPY, and \( z_{20} \), or JPY→USD, trading paths are particularly profitable.
second and equally notable observation is that while the total output of this solution is higher than Solution 2.1, which has corresponding limits on USD and EUR, it is lower than Solutions 2.2 and 3, which have higher trading maximums on USD and EUR. In fact, the total profit for Solution 4, Case 1 (Solution 4.1) is $10,880.83, whereas the profit for Solution 2.2 and Solution 3 is $10,974.53 and $12,082.59, respectively. This observation, coupled with the fact that all active exchange paths in Solution 4.1 flowing from USD and EUR to other currencies are at the respective trade maximums, indicates that the constraints on USD and EUR — not just the constraints on JPY — also have a tangible effect on the profitability of the exchange.

In order to examine the effect of the USD and EUR trade maximums, a second case was tested, in which the daily trading limits on USD and EUR were set equal to those in Solutions 2.2 and Solution 3. The limits of Case 2 are shown below:

\[x_{0j} \leq 8 \text{ for } j = 1, 2, 3, 4\]
\[x_{1j} \leq 7 \text{ for } j = 0, 2, 3, 4\]

With the solution:

\[
\begin{bmatrix}
  y \\
  x_{01} \\
  x_{02} \\
  x_{03} \\
  x_{04} \\
  x_{10} \\
  x_{20} \\
  x_{30} \\
  x_{40} \\
  x_{12} \\
  x_{13} \\
  x_{14} \\
  x_{21} \\
  x_{31} \\
  x_{41} \\
  x_{23} \\
  x_{32} \\
  x_{42} \\
  x_{43} \\
  x_{44}
\end{bmatrix}
= 
\begin{bmatrix}
  10.01248885 \\
  8 \\
  8 \\
  8 \\
  8 \\
  2988.190875 \\
  0 \\
  0 \\
  0 \\
  0 \\
  0 \\
  0 \\
  6.93995003 \\
  0 \\
  0 \\
  0 \\
  0 \\
  22.02510638 \\
  4.21434997 \\
  0
\end{bmatrix}
\]

This solution set indicates that the same overall pattern of exchanges exists as in Case 1. It is observed, once again, that those exchanges with maximum trading limits are at their maximums, and that the values assigned to \(x_{20}\) and \(x_{42}\) are increasingly large.

The exchange pattern that is implied by the nonzero variable assignments in Solutions 4.1 and 4.2 is shown in Figure 4. Note that the exchange path taken in Figure

![Figure 4: Currency Exchange Pattern for Solution 4](image)

4 is much simpler than those illustrated in Figures 1, 2, and 3. The lack of limit on
JPY leads $x_{04}$ ($\text{JPY} \rightarrow \text{USD}$) to be the only input to USD. Additionally, many of the seemingly circular exchanges observed in previous solution sets have been eliminated, leading to a much more streamlined and simple exchange pattern.

<table>
<thead>
<tr>
<th>Solution Set</th>
<th>Profit Earned</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Case 1</td>
</tr>
<tr>
<td>2</td>
<td>$9,040.06</td>
</tr>
<tr>
<td>4</td>
<td>$10,880.83</td>
</tr>
</tbody>
</table>

Table 7: Profit Earned in Solution with Corresponding USD and EUR Trade Limits

More relevant to this solution set, however, is the fact that the profit generated by this particular exchange situation exceeds the profit of those solutions with comparable USD and EUR limits (Solutions 2.2 and 3), as well as Solution 4.1, which has a similar trade limit structure to Case 2 shown here. This is well illustrated in Table 7, which lists the profit earned for Solution Sets 2 and 4, grouping the corresponding cases together. General observations about the solution sets are made in the next section.

2.6 Discussion

The results presented in the previous section can be codified into general observations:

1. The higher the maximum trading limits are, the higher the possible profit generated from arbitrage is.
2. The fewer maximum trading limits that exist, the higher the possible profit generated from arbitrage is.

The most important characteristic of these observations is that they seem to make sense when applied to real-world situations. For example, it seems probable that higher maximum trading limits would allow for higher possible profit generated through currency arbitrage. In this light, the solutions presented in the last section are helpful and appear to be reliable.

There are a few very important remarks about solution sets presented. The first would be that the profits generated in each case seem too high. In any case, the profits generated in Solution 4 are not realistic because a market situation in which no trading limits exist is not realistic. However, even the profits generated in the most conservative case, Solution 1, seem higher than would be possible in a realistic market. This is, in part, because the currency linear program formulated in Section 2.3 used mid-market exchange rates. Recall from Section 1.1 that mid-market rates lie between bid and ask rates. These rates do not account for the bid-ask spread, which represents the cost of trading with the market maker. Most dealers will charge an additional commission on top of the bid-ask spread, which cuts into profits even further.[7] This essentially means that the currency LP formulated in section 2.3 ignores transaction costs, and that any profits are therefore inflated. Another important observation is that the presence of transaction costs could feasibly alter the currency exchange path.
taken, as the solution set may tend towards fewer trades in order to minimize the overall cost. A last observation about the solution sets is while they did generate significant profits in outright dollar amounts, the profits generated are actually quite small when compared with the initial capital input required. For instance, even the highest profit generated by Solution 4.4, $12,488.85, represents only a .12% return on the initial investment. A typical investor will not have an available starting capital of 10 million USD, and therefore would generate even lower returns. As a last and very important note, since exchange rates change almost instantaneously in today’s foreign exchange market, an investor would have virtually no opportunity to actually go through the steps of formulating, running, and executing results found by a Currency LP model, as shown here. The rates he used to formulate the model would be irrelevant before solutions could even be found. While the exchange paths recommended by a Currency LP model might still be profitable seconds or even a minute later, it is unlikely that the maximum profit would be the same; it is more likely that profit generated through arbitrage would be negligible, if it still even existed at all. Assuming an investor could overcome this hurdle, however, he would only be attracted to such a Currency Arbitrage scheme because it is virtually risk-free if executed properly and because the gains are instantaneous, not because of the potential to generate an unusually high return.

While the exercise in this paper was much more focused on the theoretical aspects of Linear Programming and understanding how the concepts in the field of Linear Programming can be applied to a real-world problem, there are actually several academic papers that focus solely on developing methods for detecting arbitrage opportunity through currency exchange. Since rates change almost by the second, it is a necessary feature of these methods that the calculations be very fast; many of academic texts focus on decreasing calculation time or finding techniques that can efficiently process more than three currencies, as some academic evidence suggests that including more than three currencies increases the chance for profit.[17] Most of the papers use some method of Network Flows to formulate and solve their linear programs. Network Flow models consider the items in the LP to be nodes, while the activities are paths. A path consists of an ordered set of branches such that each node in the ordered set is the end point for only two branches in the set. The exception to this would be the first and last nodes, where the sequence begins and terminates. If the first and last nodes are the same, then the path is called a loop. [11] The currency problem is easy to define in this manner, and the graphic visual representation of each currency as a node and each exchange as a path makes formulating large problems much easier to comprehend. However, the definition of path above would preclude many of the exchanges seen in the solution sets in this paper, since each node (currency), with the exception of the node of origin, can be the endpoint for only two paths (exchanges). This would preclude the circular exchange patterns seen in the solution sets presented in the previous section, and would result in a much less complicated exchange path. One subcategorization of network flows is the maximum flow problem, which seeks to achieve the maximal flow through each of the nodes in a given network, so that the output of the system is maximized relative to input. This could also be generalized as
the minimum cost flow, which seeks to achieve the flow through the nodes such that the cost is minimized. The Generalized Circulation Problem (GCP) is the generalization of the maximum flow problem, and specifies that the relationships between the amount of flow entering and leaving a node is linear. Because this definition very closely corresponds with the situation outlined in the Currency LP, GCPs are often used in detecting currency arbitrage.

Soon and Ye implement a network flow solving method and use a type of linear programming called binary integer programming to find arbitrage opportunities with quick computational speed. The primary difference between their model and other models is that the binary integer program solution returns only values of 0 and 1, where a 0 indicates that a particular currency path is not taken, i.e. that that trade is not made, and a 1 indicates that the path is taken. The simplicity of the solutions ensures quick computational speed. They also highlight how the currency problem may be formulated as a GCP problem. Fernandez-Pérez, Fernández-Rodríguez and Sosvilla-Rivero use a genetic algorithm, which belongs to a class of adaptive search and optimization techniques, to analyze up to 14 currencies with very fast computational time, and find that the more currencies were involved, the higher the opportunity for arbitrage. Fleischer and Wayne focus their paper on finding quick solving methods for generalized flow problems, such as the generalized maximum flow, the generalized minimum cost and maximum flow. Christofides, Hewins, and Salkin use a graph theoretic approach and apply network flow and maximum flow formulation methods to a number of arbitrage problems in their paper, including space and time arbitrage.

Other papers focus solely on exchange rate arbitrage over a period of time. Jones' paper makes use of network flows to facilitate exchange rate and covered interest arbitrage, which involves moving money into currencies with the most favorable interest rate for investing. He does this first with two currencies and one period of time $t$, and then moves on to multiple currencies over multiple periods of time. Additionally, his model accounts for transaction costs and is therefore a more realistic simulation. Cantú García and Espinosa use very similar techniques to solve the specific exchange rate arbitrage problem between USD and Mexican Pesos.

The summary above provides only a basic outline of some academic papers on the topic. While some of the texts focus much more on the theoretical basis of the network flow, GCP, or maximum flow problem, others specifically aim to solve a currency arbitrage problem very similar to the one discussed in this paper. Yet another group aims to solve currency arbitrage problems involving multiple time periods, which may be a more realistic way to generate profit through arbitrage, given the quickly changing exchange rates that exist in today's market. This paper aimed only to explore the field of linear programming, the theoretical concepts of arbitrage, and to apply knowledge of both theories to an applied problem involving five currencies. However, a short discussion of the existing literature, both more theoretical and more applied in nature, is crucial for readers wishing to further investigate the topic.

For reference, Goldberg, Plotkin and Tardos discuss the GCP and minimum cost problems, as well as solving methods, in their paper in great detail.
3 Conclusion

This paper gave an overview of the foreign exchange market, the concept of arbitrage, and of the principles of Linear Programming. An extended example of pricing via arbitrage was given, and The Arbitrage Theorem was stated and later proven using concepts of Linear Programming. The terminology and techniques regarding foreign exchange and Linear Programming were then applied to a currency arbitrage problem involving the top five most traded currencies: US Dollar, Euro, Japanese Yen, British Pound, and Swiss Franc. The steps involved in formulating the linear program for the currency arbitrage problem were explained in detail. The problem was solved using the Microsoft Excel computer solver, but the theoretical concepts of the Simplex method were detailed to facilitate better understanding of the steps taken by automated linear program solvers. Four distinct solution sets, distinguished by different maximum trading limits, were presented and general observations regarding those solutions were made. It was found that increasing trading limits, as well as reducing the number of trading limits, increased the possibility for profit-generation through currency exchange. These observations were then relativized in the discussion section, which also provided a summary of other important currency arbitrage models.

References


REFERENCES


