Universidad del Rosario

Matemáticas Aplicadas y Ciencias de la Computación

Nombres: *Juanita Robles y Camila Patarroyo*

Identificación de arbitraje con monedas

Teoría de grafos

Proyecto final

Grupo 1

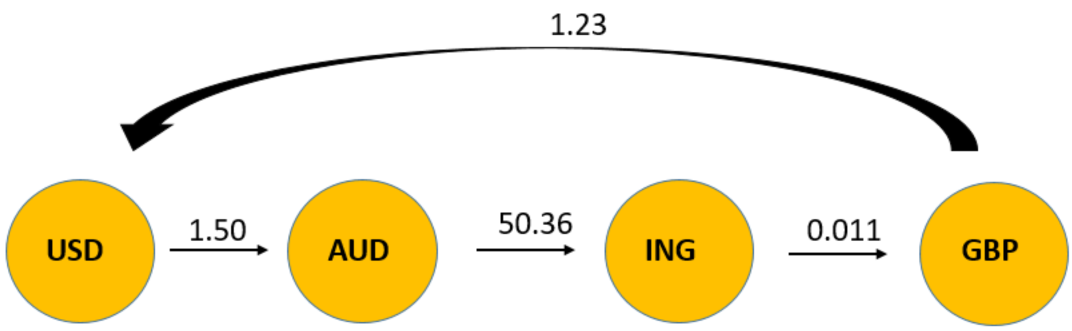
2022

Índice general

Capítulo 1

Descripción del problema y su contexto.

El arbitraje es la operacion que busca obtener ganancia con la diferencia de precios en dos mercados distintos. El mercado forex esta compuesto de diversas divisas que constantemente están siendo negociadas y cuentan con significativas diferencias en sus precios. Esto permite que se pueda realizar el arbitraje en este mercado. Para este caso específico la idea es encontrar varias monedas que al hacer una serie de compra y ventas entre ellas se genere una ganancia. Estas operaciones pueden ser representadas usando grafos dirigidos (cola sería la moneda que vendo y cabeza sería la que compro) y asignando pesos en las aristas que representan los precios.



Lo que la imagen representa son las transacciones que se generan al hacer arbitraje. Empezando en los dólares, yo tengo 1 dólar y lo intercambio (vendo) para poder comprar AUD a un precio de intercambio AUD/USD de 1.50. Una vez tenga los dólares australianos se intercambian (venden) por un precio de 50.3 INR/AUD para poder obtener rupias. Luego que tengo las rupias las intercambio a un precio de 0.011 GBP/INR consiguiendo así libras esterlinas. Finalmente cierro el ciclo al vender esas libras a un precio de 1.23 GBP/USD y consiguiendo finalmente los dólares de vuelta.

Haciendo los siguientes cálculos: $1{,}50 * 50{,}36 * 0{,}011 * 1{,}23 = 1{,}022 > 1$

Esto me dice que yo al inicio de todas mis operaciones tenía 1 dólar, después de todos los intercambios por los que pase, termine con 1.022 dólares. Lo que indica que la operación de arbitraje fue exitosa ya que terminé con más dólares que con los que empecé, teniendo una ganancia de 0.022 por cada dólar.

El objetivo de nuestro proyecto es encontrar estrategias de arbitraje en el mercado de las monedas, al construir un grafo dirigido con pesos y encontrar un camino hamiltoniano cuyo resultado sea una rentabilidad positiva, es decir mayor a 1.

Capítulo 2

Modelación y abstracción del problema.

2.1. Estructura de datos del gráfico

Para la modelación del problema es optimo usar los grafos dirigidos ponderados, ya que se pueden representar como una matriz de adyacencia.Donde la entrada en la fila i y la columna j es un número decimal o entero distinto de cero si y solo si la pareja (i, j) está en el gráfico. Cuando se indicar un peso en la matriz, se coloca en la entrada de la fila i, columna j.

2.2. Encontrar arbitraje

El arbitraje surgen cuando se determina un ciclo donde los pesos de las aristas cumplan la siguiente expresión:

$w1 * w2 * w3 * \ldots * wn > 1$

Podemos encontrar un ciclo de vértices tal que la suma de sus pesos sea negativa.Tomando el algoritmo negativo, esto se convierte en:

$(-\log(w1)) + (-\log(w2)) + (-\log(3)) + \ldots + (-\log(wn)) < 0$

Por lo tanto, podemos concluir que si podemos encontrar un ciclo de vértices tal que la suma de sus pesos sea negativa, entonces existe una oportunidad para el arbitraje de divisas. Afortunadamente, el algoritmo de Bellman-Ford es un algoritmo gráfico estándar que se puede usar para detectar fácilmente ciclos de peso negativo.

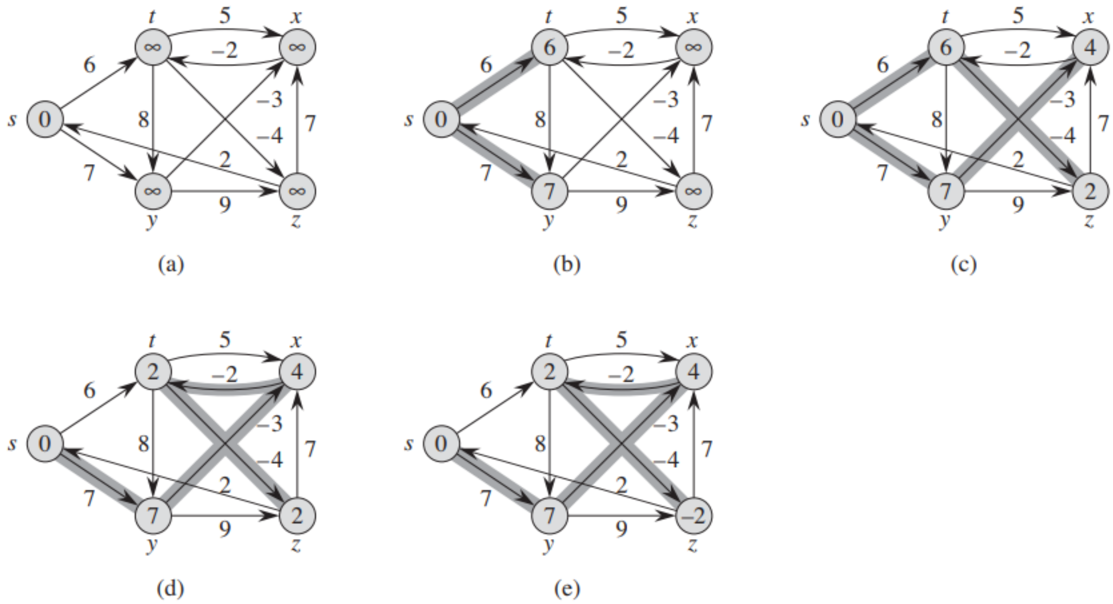
2.3. Algoritmo Bellman Ford

El algoritmo de Bellman-Ford resuelve el problema de los caminos más cortos en el caso general en el que los pesos de las aristas pueden ser negativas.El algoritmo devuelve un valor booleano que indica si hay o no un ciclo de peso negativo. Si existe tal ciclo, el algoritmo indica que no existe solución. Si no existe tal ciclo, el algoritmo produce los caminos más cortos y sus pesos. El algoritmo relaja las aristas para encontrar progresivamente el camino más corto.

```
BELLMAN-FORD(G, w, s)
1  INITIALIZE-SINGLE-SOURCE(G, s)
2  for i = 1 to |G.V| - 1
3      for each edge (u, v) ∈ G.E
4          RELAX(u, v, w)
5  for each edge (u, v) ∈ G.E
6      if v.d > u.d + w(u, v)
7          return FALSE
8  return TRUE
```

La imagen anterior muestra la ejecución del algoritmo Bellman-Ford en un gráfo con 5 vértices. Después de inicializar los valores la línea 1, cada iteración |V|-1 veces del ciclo for de las líneas 2 a 4 y consiste en relajar cada arista del grafo una vez.

En otras palabras, después de aplicar el algoritmo de Bellman-Ford en un gráfo, cada vértice mantiene el peso del camino más corto desde el vértice de origen hacia sí mismo.En cada iteración, todos los bordes se relajan si [w(i) + w(i, j) <w(j)] y el peso de cada vértice se actualiza en consecuencia. Después de la i-ésima iteración, el algoritmo encuentra todos los caminos más cortos que consisten en i-aristas como máximo.



El grafo dirigido del ejemplo de la imagen (a) es el grafo inicial, donde hay 4 infinitos en los vertices y,z,t y x, tambien cuenta con valores negativos en algunas aristas. La imagen (b) empieza el camino en el vertice s y tienen dos aristas adyacentes 't' y 'y', note que en 't' es infinito pero el valor de la arista es igual a 6 que es menor que infinito por lo que se deja el emnor valor, del mismo mono con la otra arista adyacente y con los otros vertices del grafo hasta encontrar el camino mas corto. La imagen (c), (d) y (e) representa esta seleccion del menor valor en cada arista.

Capítulo 3

Ejemplo de tamaño real.

https://github.com/Camila-pv/Identificacion-de-arbitraje-con-monedas.git

El link anterior de github se puede encontrar el archvio de python donde esta implementado el Algoritmo Bellman Ford en el cual retorna el ciclo con menor peso si existe y tambien un archivo de texto donde esta la matriz de los pesos de las monedas.

Para el ejemplo de tamaño real usamos una matriz 11x11 cuyos valores representan el peso de cada peso en el mercado:
1.'USD', 2.'EUR' 3.'JPY' 4.'GBP' 5.'CAD' 6.'AUD' 7.'NZD' 8.'CHF' 9.'DKK' 10.'NOK' 11.'SEK'

1	1	1.0532	0.007733	1.247	0.779	0.7012	0.6352	1.0072	0.1416	0.1034	0.100699
2	0.9493	1	0.73419	1.1839	0.7395	0.6657	0.603	0.9561	0.1344	0.0981	0.0956
3	129.3	136.18	1	161.236	100.703	90.671	82.129	130.211	18.3013	13.3644	13.0194
4	0.8017	0.84458	0.6201	1	0.6245	0.56229	0.5093	0.8075	0.1135	0.0829	0.0808
5	1.2839	1.35221	0.0099265	1.6009	1	0.9003	0.8155	1.2927	0.1817	0.1327	0.1293
6	1.4257	1.50185	1.1026	1.7782	1.1105	1	0.9058	1.4358	0.2018	0.1474	0.1436
7	1.5739	1.6579	0.01217	1.9629	1.2259	1.1038	1	1.5855	0.2228	0.1627	0.1585
8	0.9929	1.0458	0.7678	1.2382	0.7733	0.6963	0.6307	1	14.0528	10.2627	9.9981
9	7.0644	7.4409	5.4624	8.8094	5.502	4.9536	4.4873	7.1143	1	0.7303	0.71145
10	9.6707	10.1859	7.4788	12.0587	7.5316	6.78092	6.1418	9.739	1.3686	1	0.9739
11	9.9264	10.4553	7.6768	12.3776	7.7307	6.9603	6.3042	9.995	1.4047	1.026	1

Luego de ejecutar el programa en la terminal imprime el ciclo mas corto y con menor peso usando el algoritmo.El porcentaje resultante es la ganancia del peso incial de inversión y el ciclo de las monedas que se deberian cambiar para ganar ese porcentaje en especifico.En la imagen sigueinte nos muestra 3 ciclos de la matriz,note que el ciclo de la mitad genera una ganancia maxima de 0.0319039 con el ciclo DKK-USD-DKK.

Capítulo 4

Conclusiones.

Por medio de teoría de grafos, que es una rama de la matemática y las ciencias de la computación que estudia las propiedades de los grafos. Estas propiedades son muy utiles y con un amplio uso para aplicar en diferentes problemas de distitas areas. Como en la Identificación de arbitraje con monedas que encontramos que el uso de teoria de grafos era una herramiena muy grande para solucionar el problema de encontrar la mayor ganancia y como el provblema se puede representar de forma facil en un grafo y matriz.