

Aplicación del método Backtracking para la resolución del problema de la empresa *Clover Inversions*

CRISTIAN EDUARDO ESPINOZA SILVA

Universidad de Santiago de Chile
cristian.espinoza.s@usach.cl

12 de diciembre de 2018

Resumen

En el presente informe se muestra el problema y la solución correspondiente al segundo laboratorio del curso de Algoritmos Avanzados. La solución se elabora usando la técnica de backtracking. Se realizó un análisis de los resultados obtenidos, viendo la eficacia y eficiencia de la solución. En donde obtuvimos un orden de ejecución de $O(n) = (n)^2$. Además, a modo de conclusión se logra el objetivo de resolver el problema que presentaba la empresa Clover Inversions, mediante un algoritmo eficaz y con el orden de ejecución mencionado anteriormente.

I. INTRODUCCIÓN

En el presente Paper de la asignatura de algoritmos avanzados recae en la aplicación de uno de los tópicos que se estudia, en esta oportunidad tendremos la aplicación de la resolución de problema basada en "Backtracking". La resolución de problema mencionada anteriormente la aplicaremos al problema planteado en esta ocasión, el cual consiste en maximizar el beneficio de la empresa con un capital inicial debido a que ha crecido exponencialmente la cantidad de inversiones, pero no de la misma forma la utilidad de la empresa, también lo podemos encontrar como el problema de la "Mochila". Finalmente, tendremos un análisis que contendrá la eficiencia y eficacia del algoritmo planteado, además de dar a conocer técnicas que ayudarían a mejorar los aspectos mencionados anteriormente, con el fin de concluir finalmente si el algoritmo propuesto es efectivamente eficiente.

I. Objetivo Principal

Desarrollar un algoritmo que, mediante backtracking resuelva el problema planteado, el cual consiste en maximizar la utilidad de la empresa, obteniendo el beneficio mayor con un capital inicial. Las utilidades y sus respectivos

costos serán entregados mediante un archivo de texto, al igual que la cantidad que se dispone de capital inicial y el número total de inversiones disponibles. Por otro lado, tendremos que la solución será entregada mediante otro archivo de texto que contendrá la cantidad de utilidad generada y el detalle de cada una de las inversiones que se optaron para obtener dicho resultado.

II. Objetivo Secundario

1. Obtener las inversiones que maximizan la utilidad de la empresa.
2. Responder las preguntas principales de un algoritmo, tales como: ¿Se detiene?, ¿Cuándo se detiene?, ¿Es eficiente?, ¿Se puede mejorar?, ¿Existen otros métodos?.
3. En el caso de la eficiencia, estimar el orden de complejidad que adopta el algoritmo implementado.

III. Estructura del informe

En el informe se presentan la descripción del problema, luego un marco teórico que define conceptos clave que se utilizarán a lo largo del informe, se procede a describir y analizar la

solución obtenida para el problema para finalmente, hacer un análisis de esta respondiendo las preguntas anteriormente planteadas, haciendo énfasis en la eficiencia del algoritmo.

II. DESCRIPCIÓN DEL PROBLEMA

La empresa de inversiones internacional clover inversiones es reconocida a nivel mundial por sus grandes ganancias en este rubro de realizar inversiones y generar un máximo de utilidad para la empresa. Aunque esta pasando por un momento difícil ya que se encuentra enfrentando un problema que se esta desarrollando exponencialmente, debido a que crecen la cantidad de inversiones que se pueden realizar y no esta pudiendo realizar la máxima utilidad para las empresas y eso mantiene preocupado a los inversionistas. Debido a esto es que se nos pide realizar un algoritmo utilizando la técnica de backtraking que logre nuevamente obtener el máximo de utilidad para la empresa realizando un análisis minucioso en cada una de las inversiones que están disponibles. Esta solución se entregara en un archivo de texto, donde se indicara la utilidad máxima que se logro obtener con un capital inicial dado, dejando en claro que inversiones fueron las escogidas para llegar a una utilidad máxima.

III. MARCO TEÓRICO

1. Algoritmo: Secuencia de instrucciones finita que permite encontrar la solución a un determinado problema.
2. Fuerza bruta: Consiste en una forma de resolución de problemas en la cual se deben encontrar todos los candidatos posibles a solución, y luego dentro de ese conjunto encontrar el subconjunto que cumpla las condiciones dadas.
3. Backtracking: Es una estrategia para encontrar soluciones a problemas que satisfacen restricciones. Dejando de recorrer una solución si ya supero el limite de la solución óptima que se tiene hasta el momento.

4. Eficiencia de un algoritmo: Corresponde a las propiedades del algoritmo que nos permite identificar cuanto tiempo tarda un algoritmo en completarse y cuantos recursos usara. Por lo tanto una mayor eficiencia significara que el algoritmo usara menos recursos y tardara menos tiempo en resolver un problema.
5. Tiempo de ejecución: Expresión algebraica que indica el tiempo en que demora ejecutar un algoritmo, en función de la entrada de este. Las instrucciones mas básicas poseen un tiempo de ejecución constante 1.
6. Orden de complejidad: Corresponde a la estimación del tiempo de ejecución usando la cota superior asintótica de esta.
7. Lenguaje de programación C: Lenguaje de programación imperativo-procedural lanzado en 1972. Es un lenguaje de nivel medio que destaca por el manejo manual de memoria, poder crear tipos de datos compuestos y estructuras, además de las características propias de su paradigma.
8. Números Binarios: Es un sistema de numeración en el que los números se representan utilizando solamente dos cifras: cero y uno (0 y 1).

IV. DESCRIPCIÓN DE LA SOLUCIÓN

La cantidad de inversiones que se leen desde el archivo de entrada se guardan dentro de una lista de inversiones la cual es representada como una estructura y contiene como atributo la utilidad y costo respectivo. Además, tenemos que tener en cuenta que se guarda el capital inicial con el cual se consta y el numero de inversiones disponibles para poder consumir y conseguir la utilidad máxima para la empresa.

1. Idea: Se desea tratar el problema planteado que recae en la obtención máxima de utilidad para las empresas analizando cada una de las inversiones que se tiene disponible. Para abordar el problema se realiza una creación de una estructura que representa la utilidad y costo respectivo que tiene la inversión, esto nos permitirá tener

Costo	Utilidad
250000	1000
600000	6000
450000	4000
300000	2000
100000	4500

Figura 1: Representación gráfica de la entrada

un manejo de los datos de una forma mas directa. Con respecto a las combinaciones posibles ocuparemos una representación gráfica no muy común basada en formato de números binarios, lo cual nos permitirá realizar cada una de las combinaciones posibles de las inversiones, debido a que la cantidad máxima de combinaciones la podemos obtener con la siguiente formula:

$$2^n, n = \text{Numerodeinversiones} \quad (1)$$

La cual nos permitirá esa cantidad de números con su representación binaria, además así pudiendo representar cada una de las inversiones como una posición del número binario, donde el "1" nos representara que se encuentra presente dicha inversión y en caso contrario que no esta presenta. Al mismo momento que se está creando la combinación binaria de cada uno de los números binarios se irá comparando con la utilidad máxima que se tiene hasta el momento (Se asume que la utilidad máxima que se genera es con la combinación numero 1 de los números binarios), así solamente guardando dicha combinación si entrega una utilidad mayor a la empresa sin sobrepasar el capital inicial. Finalmente, se procederá a crear el archivo de salida que contendrá la utilidad entregada, el capital inicial y enumerar las inversiones que se utilizaron para llegar a esa utilidad máxima.

2. Funciones y procedimientos: A continuación dejaremos en claro cada una de las funciones y algoritmos utilizados para llegar a la solución del problema, tenemos que tener en cuenta que las funciones uti-

lizadas se separan en 3 sub-secciones y la función principal, que recaen en :

- a) Funciones auxiliares: Funciones que se encargan de resolver sub-problemas que se obtienen al ir desarrollando cada uno sucesos que se enfrentan para llegar a la solución final del problema.
- b) Funciones de manejo de archivo: Funciones que se encargan de cargar y escribir los archivos que se ocupan para el comienzo y finalización del programa.
- c) Funciones de manejos de matrices: Funciones que se encargan de crear y mostrar las inversiones leídas en el archivo de texto.
- d) Función principal : Función que se encarga de realizar el llamado a cada una de las funciones para obtener el resultado final, a continuación dejamos el pseudocódigo que se utilizo para su implementación:

```

auxiliary = []
i = 0
numberCombinations = mypow(2,numInversiones)
auxUtility = []

#Asignamos mochila vacia (0 0 0 ...)
maximunUtility = convertToBinary(0,numInversiones)
maxUtility = benefitGranted(investments)

#Empezamos a comparar las combinaciones
Mientras i < numeroCombinaciones

    auxiliary = convertToBin(i,numeroCombinaciones)
    auxUtility = benefitGranted(investments)

    Si auxUtility.utility > maxUtility.utility
        maximumUtility = auxiliary;
        maxUtility = auxUtility;

return maximumUtility

```

A continuación se deja la explicación del pseudocódigo

- 1) Podemos observar que se declaran al comienzo cada una de las variables que serán utilizadas durante el transcurso del algoritmo.
- 2) Se asigna una utilidad máxima para poder utilizar como comparador (primera posición) durante el algoritmo.
- 3) Se realiza la creación de los números binarios mientras el nume-

ro sea menor al numero de combinaciones posibles.

- 4) Se realiza la conversión del número actual, binario y se calcula la utilidad que entrega, teniendo en cuenta que no sobrepase el capital inicial.
- 5) Se compara si dicha utilidad calculada es mayor a la que se tiene actualmente para reemplazarla, en caso contrario se mantiene la misma.
- 6) Finalmente, se entrega la inversión con el número en formato binario que entrega la mayor utilidad a la empresa.

Revisando cada sentencia de declaración que hacemos tenemos que dentro del ciclo, se realizan llamados a dos funciones que cada una tiene un $O(n)$, por lo tanto tenemos que el tiempo de ejecución obtenido es de $T(n) = 3 * n^2 + 13 * n + 7$ y el orden de nuestra solución queda en $O(n^2)$

V. ANÁLISIS DE LA SOLUCIÓN Y RESULTADOS

1. Análisis de la solución

- a) ¿El algoritmo se detiene?: Si se detiene, al momento de obtener la combinación de inversiones que entrega la utilidad máxima para la empresa.
- b) ¿Resuelve el problema?: Si resuelve el problema, ya que luego de verificar cada una de las combinaciones binarias, entrega como resultado la que maximiza la utilidad.
- c) ¿Es eficiente?: Es más eficiente que ocupar fuerza bruta, ya que se descartan enseguida los estados que no sean mejor que la solución ya encontrada, sin embargo, aun tiene un orden de complejidad ineficiente.
- d) ¿Se puede mejorar?: Si, agregando una comparación al comienzo del código que verifique de inmediato si la mochila completa cumple con un

costo menor al capital inicial, debido que es sabido que mientras más elementos hayan en la mochila mayor será la utilidad que se le otorgará a la empresa.

- e) ¿Existe otro método?: Si, se puede ocupar programación dinámica la cual nos va a asegurar la utilidad máxima en su absoluto, debido a que comienza su resolución de atrás hacia adelante teniendo en consideración los resultados anteriores con el fin de ahorrar cálculos computacionales, provocando un tiempo de ejecución menor.

2. Complejidad de la solución

A continuación tendremos una tabla donde se verá reflejado cada uno de los órdenes que adopta cada una de las funciones implementadas en el programa realizado.

Función	Orden - $O()$
printCurrent	1
freeMemory	1
myPow	n
convertToBinary	n
benefitGranted	n
loadInvestments	n
saveInvestments	n
backtracking	N^2
createInvestments	1
printInvestments	n
main	n

Figura 2: Tabla que contiene orden de cada una de las funciones

VI. MODO DE EJECUCIÓN

La secuencia de pasos para ejecutar el programa son los siguientes:

1. Primero nos dirigimos a la carpeta principal y ejecutamos el makefile desde la consola de la siguiente forma *make*.

2. Luego nos dirigimos a la carpeta en donde se creo el archivo ejecutable de la siguiente forma *cdBin*.
3. A continuación, ejecutamos el siguiente comando *./main.out*.
4. Luego, el programa consta con un menú que ayuda a la utilización del mismo.

VII. TRAZA DE LA SOLUCIÓN

A continuación realizaremos el paso a paso que va a realizando nuestro algoritmo implementado para poder abordar el problema descrito anteriormente:

1. Paso: Primero realizamos la lectura del archivo, de donde obtendremos el capital inicial, la cantidad de inversiones y cada una de las inversiones disponibles, lo cual nos permitirá crear una lista de inversiones, la cual se representa mediante una estructura que contiene su utilidad y costo respectivo. Podemos decir que al terminar dicho proceso, tenemos cargado en nuestro programa la cantidad de inversiones disponibles para poder combinar y obtener la utilidad máxima para la empresa.
2. Paso: Realizamos el calculo de la cantidad máxima de combinaciones que tendremos, esto lo obtenemos de la siguiente manera:
 - a) Calculamos la cantidad máxima de posibles combinaciones que podemos realizar con el numero de inversiones disponibles que dispones, esto lo obtenemos aplicando la formula 1 descrita anteriormente.
 - b) Luego de obtener el numero total de combinaciones posibles que tendremos, realizamos la representación de cada uno de ellos con el fin de obtener las distintas combinaciones que se tendrán, teniendo en cuenta que "1"nos indica que esa inversión esta considerada dentro de esa combinación y en caso contrario "0"no se encuentra considerada. Revisando una representación de lo mencionado anteriormente tenemos que:
 - 1) Utilidad: 25 | Costo: 1000
 - 2) Utilidad: 30 | Costo: 4000
 - 3) Utilidad: 35 | Costo: 3000
 Donde considerando las 3 utilidades anteriores tenemos una aplicando la formula 1 una cantidad total de 8 combinaciones posibles las cuales nos van a representar cada una de las combinaciones posibles con un total de 3 inversiones:
 - 1) 1 2 3
 - 2) 0 0 0 = Utilidad: 0 | Costo: 0
 - 3) 0 0 1 = Utilidad: 35 | Costo: 3000
 - 4) 0 1 0 = Utilidad: 30 | Costo: 4000
 - 5) 0 1 1 = Utilidad: 65 | Costo: 7000
 - 6) 1 0 0 = Utilidad: 25 | Costo: 1000
 - 7) 1 0 1 = Utilidad: 60 | Costo: 4000
 - 8) 1 1 0 = Utilidad: 55 | Costo: 5000
 - 9) 1 1 1 = Utilidad: 90 | Costo: 8000
 Donde tenemos que el primer numero de la lista anteriores nos indica si esta presenta o no dicha inversión en las combinaciones anteriores.
 - c) Luego asumimos que la primera combinación es la que entrega la utilidad máxima a la empresa con el fin de tener un elemento comparador, además cada vez que se este creando los números binarios descritos anteriormente se procede a verificar que no supere el capital inicial con el cual se esta trabajando. Además, de ser menor el costo al costo inicial, se procede a verificar si entrega una utilidad mayor a la que se tiene actualmente como utilidad máxima de ser afirmativo lo anterior se realiza el cambio nombrando como la combinación que entrega el mayor beneficio a la empresa.
 - 1) Capital inicial: 4000
 - 2) 0 0 0 = Utilidad: 0 | Costo: 0
 La nueva combinación creada es la siguiente:
 - 1) 0 0 1 = Utilidad: 35 | Costo: 3000

Donde podemos observar que entrega una utilidad mayor y no supera el costo inicial declarado anteriormente, lo cual nos hace nombrar a esta combinación como la **nueva que entrega la utilidad máxima parcialmente**. Luego analizamos la siguiente combinación que recaen en:

1) 0 1 0 = Utilidad: 30 | Costo: 4000

Donde podemos observar que no supera el costo inicial declarado al comienzo, por ende, esta combinación es descartada. A continuación analizamos la siguiente combinación:

1) 0 1 1 = Utilidad: 65 | Costo: 7000

Donde podemos observar que entrega una utilidad mayor, pero supera el capital inicial, por ende, nuevamente tenemos que esta combinación es descartada. Analizamos la siguiente combinación:

1) 1 0 0 = Utilidad: 25 | Costo: 1000

2) 1 0 1 = Utilidad: 60 | Costo: 4000

Como podemos observar ninguna de las combinaciones anteriores supera el capital inicial, pero en la primera entrega una utilidad menor a la que actualmente es la mayor, en cambio tenemos que la segunda entrega una utilidad mayor por ende tenemos que es la **nueva combinación que entrega la utilidad máxima para la empresa**. Analizando las ultimas combinaciones que quedan tenemos lo siguiente:

1) 1 1 0 = Utilidad: 55 | Costo: 5000

2) 1 1 1 = Utilidad: 90 | Costo: 8000

Tenemos que ambas combinaciones quedan descartadas por el primer criterio que es superar el capital inicial, por ende podemos concluir que la **combinación que nos entrega la utilidad máxima para la empresa** luego de analizar cada una de las opciones recaen en la siguiente:

1) 1 0 1 = Utilidad: 60 | Costo: 4000

d) Por ultimo, lo que realizamos es escribir el archivo el cual contiene en su primera linea el capital invertido seguido de la utilidad que le entrega a la empresa, para terminar dejando un historial de cada una de las inversiones seleccionadas para obtener a utilidad máxima para la empresa.

VIII. CONCLUSIÓN

La solución entregada cumple efectivamente con lo pedido en el enunciado, es decir que este entrega la utilidad máxima para la empresa y su respectivo costo, además de cumplir con el requisito de usar backtracking para encontrar la solución. El algoritmo propuesto responde correctamente a las preguntas de análisis del algoritmo.

En cuanto a la eficiencia de este, el algoritmo tiene un orden de complejidad bastante alto, el cual es el siguiente :

1. Analizamos cada una de las funciones mostradas en la figura 2, llegamos a un análisis que acotando superiormente, el orden de complejidad de la solución propuesta es de:

$$O(n) = (n)^2 \quad (2)$$

El algoritmo recae en un orden de complejidad aceptable para la complejidad del problema, debido a su orden polinómico.

Podemos observar que backtracking es mas eficaz que Fuerza bruta, además de entregar una eficiencia mejor. Sin embargo, se debe tener en cuenta que en el ámbito de la programación este método implementado requiere de un análisis mas minucioso que fuerza bruta para llegar a una solución correcta del problema.

Por ultimo, se debe tener en consideración que el laboratorio se desarrollo y llego a prueba en un Sistema Operativo Linux.

REFERENCIAS

- [M. Villanueva, 2002] Algoritmos: Teoría y aplicaciones.