

Laboratorio N°3: Método Goloso

ANDRÉS MUÑOZ

Profesora: Mónica Villanueva

Ayudante: Patricio Vargas

Compiled May 8, 2017

En la siguiente investigación se desarrolla la técnica de resolución de problemas denominada *Goloso*. Con esta técnica se va creando la solución paso a paso, eligiendo un elemento perteneciente a un conjunto dado, que sea el mejor dado un criterio y que cumpla la propiedad definida al agregarlo a la solución. Esta técnica es aplicada a un problema en particular que consiste en indicar el efectivo mínimo que debe ser entregado por un cajero a un cliente. Además, se implementan esta técnica en lenguaje de programación C para ejemplificar y analizar sus características y comportamiento para este problema.

1. INTRODUCCIÓN

El ser humano ha inventado un sin número de cosas útiles, pero sin lugar a duda, muchos de estos inventos generan un sin fin de problemas. Para solucionar estos diferentes problemas, existen varios métodos, algunos mejores que otros, que conllevan a una implementación de una solución. Esta solución puede entregar una o muchas soluciones validas, pero que sea válida no garantiza que sea la óptima.

En este informe se implementará un tipo de algoritmo, el cual será comparado con los otros vistos en las entregas anteriores. Para llevar a cabo esta comparación, se analizará el tiempo de ejecución que demora en entregar una solución en función de una entrada.

En el segundo capítulo, se da a conocer el problema planteado, explicando las diferentes restricciones. Luego en el tercer capítulo, se darán a conocer los diferentes conceptos técnicos necesarios para entender a profundidad lo planteado en este documento. En el cuarto capítulo se describe la solución que se dará al problema, aquí es donde se explica cómo se implementa un algoritmo Goloso y su tiempo de ejecución para el problema planteado. Posteriormente en el quinto capítulo se analizan los resultados obtenidos, esto se refiere a evaluar contestando unas series de preguntas el algoritmo implementado anteriormente, así como también compararlo con el algoritmo implementado en la experiencia anterior. En el siguiente capítulo se muestran tablas las cuales representan una traza de la implementación para un caso particular. Finalmente se dará a conocer lo objetivos logrados, además de interpretar lo investigado e implementado a lo largo de este proceso.

2. DESCRIPCIÓN DEL PROBLEMA

El banco regional de Concepción tiene problemas de atención al público y sus clientes se quejan constantemente de la lentitud con que avanzan las filas de las cajas.

Las quejas siempre hacen que una institución pierda prestigio y a la vez, no sea recomendado por los mismos clientes. Esto conllevaría a que muchos clientes se cambien de banco prefiriendo uno con mejor servicio de atención al público.

Al parecer el problema de la lentitud en la que avanzan las filas son los cajeros, ya que no son lo suficientemente rápidos para calcular el vuelto que deben entregar después de un pago y además no siempre lo hacen correctamente, es decir, muchas veces devuelven una cantidad errónea de dinero, lo cual provoca peleas y retrasos en la atención.

3. MARCO TEÓRICO

- **Lenguaje de programación:** Es un lenguaje formal diseñado para realizar procesos que pueden ser llevados a cabo por máquinas como las computadoras.
- **Algoritmo:** Conjunto ordenado y finito de operaciones para hallar la solución de un tipo de problemas. Un algoritmo puede ser implementado en diferentes lenguajes de programación.[1]
- **Método Goloso:** Es un método de resolución de problema, el cual para formar la solución, va agregando elementos uno por uno, donde cada uno de esos es el mejor según un criterio y que al agregar ese elemento a la solución se sigue cumpliendo una propiedad definida previamente.

4. DESCRIPCIÓN DE LA SOLUCIÓN

La solución para disminuir las quejas de los clientes, es diseñar e implementar un programa en el que se indique a los cajeros el vuelto que deben entregar y la cantidad mínima de efectivo para hacerlo.

El programa realizado, cumple con el sistema monetario con el cual rige la sucursal, el cual es el chileno. Además, el programa fue escrito en lenguaje de programación C y utilizando la técnica de resolución de problemas denominado Goloso. A continuación, se procede a explicar cómo funciona esta técnica para el caso particular de este problema.

Primero es necesario calcular el vuelto que se debe entregar al cliente, el cual se obtiene de la resta entre lo que entrega el cliente para pagar y lo que debe pagar. Luego es necesario guardar la cantidad que se tiene de cada billete o moneda. Estos dos pasos previos serán utilizados para realizar la técnica Goloso:

Se irá agregando el efectivo de mayor valor que se tenga disponible y que sea menor o igual al vuelto restante que se debe entregar. Esto quiere decir que cada vez que se agregue efectivo a la solución, se restará el valor de éste al vuelto. Finalmente se muestra al usuario de la aplicación la cantidad agregada de cada billete o moneda a la solución, ya que este sería el mínimo efectivo que se debe entregar. Para entender de mejor manera la técnica consultar apuntes [2].

A continuación, se muestra una tabla de todas las funciones implementadas para llevar a cabo la solución, en esta tabla se especifica el tiempo de ejecución y orden para cada función. Cabe recalcar que la función en la cual se implementa el concepto de Goloso tiene el nombre de *calculate*, la cual se encarga de calcular el efectivo mínimo a entregar. También mencionar que dentro de esta función subyace la función *initDeliveredCoins*, la cual reinicia el arreglo en el cual se guardarán la cantidad de cada efectivo que se debe entregar. Además, subyace la función *showChange*, que muestra en la consola la cantidad que se debe entregar de cada efectivo.

Table 1. Tiempos y orden de ejecución de funciones

Función	Tiempo	Orden
existsFile	$T(n) = c$	$O(1)$
loadValues	$T(n) = c + cn$	$O(n)$
initDeliveredCoins	$T(n) = c + cn$	$O(n)$
showChange	$T(n) = c + cn$	$O(n)$
addCash	$T(n) = c + cn$	$O(n)$
calculate	$T(n) = c + cn + cn^2$	$O(n^2)$

En la siguiente figura se muestra el código en el cual se implementa el algoritmo de Goloso:

```
// SE APLICA METODO GOLOSO
while (i<10){
    while (change>=valueCoins[i] && numberCoins[i]>0){
        deliveredCoins[i]+=1;
        numberCoins[i]-=1;
        change -= valueCoins[i];
    }
    i++;
}
// FIN METODO GOLOSO
```

Fig. 1. Algoritmo implementado técnica Goloso

5. ANÁLISIS DE LOS RESULTADOS

Analizando el algoritmo Goloso implementado, se puede decir que siempre **termina**, esto quiere decir que no realiza bucles infinitos. También mencionar que cuando este algoritmo finaliza lo hace con la **solución**, la cual en este caso particular corresponde a la cantidad mínima de efectivo para un vuelto que se debe entregar a un cliente. Cabe recalcar que el tiempo de ejecución del algoritmo depende de la combinación entre cantidad de efectivo que se tenga que entregar, la cantidad de efectivo que se tenga en la caja, y la cantidad de tipos de efectivo que se tengan. Esto quiere decir que una combinación distinta de estos datos podría dar el mismo tiempo de ejecución.

Teniendo en cuenta la tabla 1, se obtiene que el orden de la función *calculate* es de n^2 , esto quiere decir que es muy eficiente. Principalmente este orden está dado por un ciclo anidado en otro, ya que es necesario verificar si se puede agregar cada una de las monedas a la solución y por cada una de éstas se va agregando una unidad hasta que no se pueda agregar más.

Es necesario mencionar que para este problema en particular, el método Goloso garantiza el óptimo, cabe recalcar que esto es debido a que el archivo de entrada esta ordenado desde el billete de mayor valor, hasta la moneda de menor valor. En el caso de que no esté ordenado el archivo de entrada, el programa no entregaría la solución correcta ni óptima. Para solucionar este problema, lo que se debe hacer, es ordenar el efectivo de mayor a menor valor, transformando la entrada al algoritmo implementado. Este arreglo al programa afectaría al tiempo de ejecución, pero no al orden de las funciones ya que de todas formas la función *calculate* sería la de mayor orden, ya que el mejor algoritmo de ordenamiento es de orden $n \log(n)$

6. TRAZA DE LA SOLUCIÓN

Las siguientes tablas muestran una traza del algoritmo de Goloso, el cual ejemplifica de manera concreta como se va generando la solución. En este ejemplo el vuelto es de \$35500.

Table 2. Trazo de arreglo de cantidades disponibles

Variable	T=0	T=1	T=2	T=3	T=4
Vuelto	35500	15500	5500	500	0
cd[0]	20	19	19	19	19
cd[1]	40	40	39	39	39
cd[2]	50	50	50	49	49
cd[3]	100	100	100	100	100
cd[4]	45	45	45	45	44
cd[5]	1000	1000	1000	1000	1000
cd[6]	750	750	750	750	750
cd[7]	500	500	500	500	500
cd[8]	90	90	90	90	90
cd[9]	100	100	100	100	100

Table 3. Trazo de arreglo de cantidades entregadas

Variable	T=0	T=1	T=2	T=3	T=4
Vuelto	35500	15500	5500	500	0
ce[0]	0	1	1	1	1
ce[1]	0	0	1	1	1
ce[2]	0	0	0	1	1
ce[3]	0	0	0	0	0
ce[4]	0	0	0	0	1
ce[5]	0	0	0	0	0
ce[6]	0	0	0	0	0
ce[7]	0	0	0	0	0
ce[8]	0	0	0	0	0
ce[9]	0	0	0	0	0

Como se puede apreciar en la tabla 2, se van descontando las cantidades de efectivo disponible, mientras que en la tabla 3, se van agregando unidades de efectivo que debe ser entregado.

Al finalizar el algoritmo en $T = 4$, se observa que $ce[0] = 1$, $ce[1] = 1$, $ce[2] = 1$ y $ce[4] = 1$, lo que quiere decir este resultado es que se debe entregar un billete de \$20000, un billete de \$10000, un billete de \$5000, y una moneda de \$500 respectivamente, sumando un total de 35500 que es igual al vuelto que se debe entregar con el efectivo mínimo.

7. CONCLUSIONES

Para concluir este informe se pudo observar que se cumplieron todos los diferentes objetivos. El objetivo principal fue realizar un programa que indique la cantidad mínima de efectivo que debe entregar un cajero a un cliente, donde se utilizó la técnica de resolución de problema denominada *Goloso*.

Posteriormente se analizó el algoritmo implementado y se llegó a la conclusión de que es eficiente para este problema, ya que entrega la primera solución encontrada, a diferencia de los métodos de backtracking y fuerza bruta, los cuales generan muchas soluciones y todas las soluciones respectivamente. Además, la solución entregada es óptima cuando los valores del efectivo están ordenados de mayor a menor, como se dijo en el capítulo de análisis de resultados, incluso se dio una forma de cómo mejorar el programa y que dicha condición no sea necesaria.

Finalmente, se ha llevado a la práctica distintos conceptos vistos en clases, de esta forma se ha comprendido de una forma mejor sobre éstos. Con esta tercera entrega, es posible decir, que el óptimo de una solución dado un algoritmo implementado, depende netamente del problema. En este problema en particular al utilizar la técnica *Goloso*, se obtiene el óptimo, pero generalmente a través del método *Goloso* no se obtiene.

REFERENCES

1. "Definición de algoritmo," <http://conceptodefinicion.de/algoritmo>. [Online; Último acceso 17-Abril-2017].
2. M. V. Ilufi, *Algoritmos: Teoría y aplicaciones* (2002).