Problema de la mochila sin fraccionamiento

Realizado por: Samuel López Prieto

Índice:

Descripción de las cotas

Nota: los objetos están ordenados por su relación valor/peso de mayor a menor.

Cotas buenas:

Para la cota optimista lleno la mochila con los primeros objetos disponibles hasta que uno no quepa, entonces se licua ese objeto y se mete la parte proporcional que cabe en la mochila

La cota pesimista es igual que la optimista pero sin licuar el objeto que no cabe.

Ambas cotas tienen coste lineal en el número de objetos no explorados.

Cotas malas:

Para la cota optimista: beneficio actual + licuar el primer objeto disponible y llenar toda el espacio restante con el.

La cota pesimista es simplemente el beneficio actual.

Ambas cotas tienen coste constante.

Documentación

Elegir conjunto de datos

Los datos están almacenados en txt, para cargarlos hay que ir a a la función leerDatos y escribir el nombre del fichero deseado

```
void leerDatos(elem lista[], int & N) {
    ifstream myfile("datos1000.txt");
    if (myfile.is_open())
    {
        myfile >> N;
        for (int i = 0; i < N; i++) {
            myfile >> lista[i].valor;
            myfile >> lista[i].peso;
            myfile >> lista[i].valorReal;
        }
        myfile.close();
    }
    else std::cout << "Unable to open file";
}
```

Todos los ficheros tienen el mismo estilo de nombre datosX donde X es el número de objetos de ese archivo.

Los datos han sido creados aleatoriamente, después ordenados y almacenados en su txt desde la función generarDatos.

Para elegir cotas o factibilidad

Hay que ir a la función mochila y cambiar la función calcularCotasX donde X puede ser Buenas o Malas. Hay que cambiarlo dos veces, una antes de while y otra dentro del while cuando no se elige el objeto:

```
solucion mochila(const elem lista[],const int nObjetos) {
    priority_queue <Objeto, vector<Objeto>, CustomCompare> cola;
    solucion mejorSol;
    mejorSol.ben = 0;
    int nodosExplorados = 0;
    float optNueva,benefAnt,pesNueva, pesoAnt;
    List<int> l;
    Objeto nodo(0, 0, 0, 1, 0); //generar raiz

    calcularCotasBuenas(optNueva, pesNueva,nodo.k, nodo.pesoAct, nodo.benefAct
```

```
        //calcular nodo sin meter objeto
        calcularCotasBuenas(optNueva, pesNueva, nodo.k+1, nodo.pesoAct,
        if (optNueva >= mejorSol.ben) { //meter en la cola
            if (nodo.k == nObjetos-1) { // si es solucion
                mejorSol.sol = nodo.elems;
                mejorSol.ben = nodo.benefAct;
            }
```

Para elegir la función que usa cotas hay que llamar a mochila y para usar la función que solo usa factibilidad llamar a mochilaSoloFactible.

```
int main() {
    int nObjetos;
    elem lista[MAXOBJETOS];
    //generar_datos(lista);
    leerDatos(lista,nObjetos);
    mochila(lista, nObjetos);
    //mochilaSoloFactible(lista, nObjetos);
    return 0;
}
```

Análisis de resultados
40 objetos

Factibilidad:

```
La solucion esta formada por estos elementos:0 1 2 3 4 5 6 7 y tiene un benefici
o de: 222
Se han explorado 686772 nodos
El tiempo medio por nodo explorado es de: 0.0786169 ms
Tiempo hasta encontrar solucion:  53991.9 ms
```

Poda mala:

```
La solucion esta formada por estos elementos:0 1 2 3 4 5 6 7 y tiene un benefici
o de: 222
Se han explorado 672 nodos
El tiempo medio por nodo explorado es de: 0.0801789 ms
Tiempo hasta encontrar solucion:  53.8802 ms
```

Poda buena:

```
La solucion esta formada por estos elementos:0 1 2 3 4 5 6 7 y tiene un benefici
o de: 222
Se han explorado 96 nodos
El tiempo medio por nodo explorado es de: 0.0659411 ms
Tiempo hasta encontrar solucion:  6.33034 ms
```

Con solo 40 objetos ya se podan más de 650.000 nodos. La poda mala explora 7 veces más nodos que la poda buena.

50 objetos

Factibilidad:

```
La solucion esta formada por estos elementos:0 1 2 3 4 5 6 7 8 y tiene un benefi
cio de: 250
Se han explorado 1492498 nodos
El tiempo medio por nodo explorado es de: 0.0847914 ms
Tiempo hasta encontrar solucion:  126551 ms
```

Poda mala:

```
La solucion esta formada por estos elementos:0 1 2 3 4 5 6 7 8 y tiene un benefi
cio de: 250
Se han explorado 264 nodos
El tiempo medio por nodo explorado es de: 0.0870079 ms
Tiempo hasta encontrar solucion:  22.9701 ms
```

Poda buena:

```
La solucion esta formada por estos elementos:0 1 2 3 4 5 6 7 8 y tiene un benefi
cio de: 250
Se han explorado 100 nodos
El tiempo medio por nodo explorado es de: 0.0482323 ms
Tiempo hasta encontrar solucion:  4.82323 ms
```

Los nodos explorados pasan de 1 millón y medio con solo factibilidad, a 264 con la poda mala y 100 con la poda buena.

Curiosamente los tiempos por nodo son menores con la cota buena que con la mala. No se pueden considerar fiables porque el proceso solo se ha ejecutado 23 y 5 ms con las podas malas y buenas respectivamente. Hace falta ejecutarlos más tiempo.

500 objetos

Nota : No se incluye factibilidad porque tardaría demasiado.

Poda buena

```
La solucion esta formada por estos elementos:0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
15 16 17 18 19 21 22 23 25 y tiene un beneficio de: 7632
Se han explorado 1232 nodos
El tiempo medio por nodo explorado es de: 0.295744 ms
Tiempo hasta encontrar solucion:  364.357 ms
```

Poda mala

```
La solucion esta formada por estos elementos:0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
15 16 17 18 19 21 22 23 25 y tiene un beneficio de: 7632
Se han explorado 350832 nodos
El tiempo medio por nodo explorado es de: 0.768342 ms
Tiempo hasta encontrar solucion:  269559 ms
```

Se podan 350.000 nodos de una cota a la otra, lo que demuestra lo mala que es esa cota.

Ocurre una cosa curiosa y es que tarda menos tiempo por nodo usando la cota constante que la lineal, supongo que será porque aunque haya muchos objetos en estos ejemplos la lineal explora muy pocos. Por ejemplo en esta última ejecución con 500 objetos la mejor solución está formada por sólo 23 objetos, es decir caben muy pocos objetos, esto se podría arreglar aumentando el peso de la mochila. Pero esto genera otro problema, como la cota mala es muy mala tardaría demasiado en terminar y no puedo comparar los resultados. En resumen:la cota mala es demasiado mala como para poder probar el tiempo por nodo con muchos objetos.

Es posible que tarde menos por nodo con la poda buena porque la cola es de menor tamaño y se gestione más rápido.
Vamos a estudiar la media sin tener en cuenta la gestión de la cola(empezando el tiempo desde haber hecho pop en la pila)

200 nodos, cambiando la media

poda mala

```
MEDIA ES 0.0521594
La solucion esta formada por estos elementos:0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
15 16 17 18 20 y tiene un beneficio de: 6338
Se han explorado 32764 nodos
El tiempo medio por nodo explorado es de: 0.223435 ms
Tiempo hasta encontrar solucion:  7320.63 ms
```

poda buena

```
MEDIA ES 0.0377052
La solucion esta formada por estos elementos:0 1 2 3 4 5 6 7 8 9 10 11 12 13 14
15 16 17 18 20 y tiene un beneficio de: 6338
Se han explorado 1026 nodos
El tiempo medio por nodo explorado es de: 0.203438 ms
Tiempo hasta encontrar solucion:  208.727 ms
```

La nueva media “real”, sin contar la extracción de la cola es la que aparece al principio.
“ MEDIA ES …”

Si se compara con el tiempo medio por nodo se puede ver como la mayoría del tiempo de cada nodo se gasta en gestionar su cola de prioridad. Pasa de 0.2 ms a 0.03ms.
Aun así sigue dando menos tiempo por nodo la cota lineal que la constante.

Esto puede ser porque sigo contando el tiempo de hacer push en la cola. No voy a estudiar el tiempo sin los push en la cola porque creo que altera demasiado el rendimiento real de cada nodo, de hecho esto que acabo de hacer ya altera bastante el rendimiento real dado que la mayoría de tiempo se usa en gestionar la cola de prioridad.

Conclusiones

Factibilidad

Usar únicamente la factibilidad solo permite trabajar con conjuntos de datos muy pequeños, en mi caso no ha podido ni con 100 datos.
Esto pasa porque si todos los nodos son factibles tienes que explorar N! nodos(siendo N el número de elementos). 100! ya es 9.332622e+157

Además siendo N tan pequeño ni siquiera tardamos menos por nodo pues las cotas son lineales y constantes.

En conclusión, no se puede usar únicamente la factibilidad para problemas tan complejos, a no ser que tengamos poquísimos nodos explorables.

Cota mala

Usar una cota tan mala ayuda pero no mucho, hemos pasado de poder explorar 50 objetos a 500.

Esto pasa porque la cota pesimista es muy mala (beneficio actual) podando muy pocos nodos. Ni siquiera completa la solución, que sería el mínimo exigible, pero no lo he hecho porque eso lo he reservado para la cota buena pesimista.

Cota buena

Produce muy buenos resultados incluso con N grandes, la diferencia entre una cota mala y una buena es enorme, la mala solo puede manejar 500 datos y la buena maneja 10000, lo que son 100 veces más datos, hay que ajustar las podas lo máximo posible para obtener buenos resultados con estos algoritmos.

Tiempo medio por nodo

Si bien es cierto que la cota constante debería tardar menos por nodo que la lineal, esto no ocurre por varios motivos:

1. El número de objetos de los nodos no puede ser grande porque entonces la cota mala debería explorar demasiados nodos y tardaria demasiado en terminar
2. Al ser el número de objetos tan pequeños(<50) la diferencia entre una complejidad lineal y constante es prácticamente nula. Habría que comparar con muchos más, pero no se puede por 1.
3. Como la cota lineal explora muchos nodos menos, sus operaciones de añadir y extraer de la cola son mucho más rápidas que las de la cota constante. Y esto es lo que más tiempo consume en cada iteración del while. Como se puede ver cuando calculé la media sin tener en cuenta la extracción de la cola: El tiempo media por nodo pasa de 0.22 ms a 0.052 ms

Final

Importa mucho más lo buena que sea la cota que su complejidad, en este problema una lineal es muchísimo mejor que constante, y me atrevería a decir que para otros problemas, cotas cuadráticas o de peor complejidad pueden llegar a dar mejor rendimiento que cotas constantes que aproximen peor la solución real.