DECSAI
Universidad de Granada

Problema del Viajante del Comercio

Técnicas Backtracking y Branch and Bound

ugr Universidad de Granada

ÍNDICE

ENUNCIADO DEL PROBLEMA

Dado un conjunto de ciudades y una matriz con las distancias entre todas ellas, un viajante debe recorrer todas las ciudades exactamente una vez, regresando al punto de partida, de forma tal que la distancia recorrida sea mínima. Más formalmente, dado un grafo G, conexo y ponderado, se trata de hallar el ciclo hamiltoniano de mínimo peso de ese grafo.

SOLUCIÓN TEÓRICA. BACKTRACKING

ALGORITMO:

- Comienza por la ciudad 1 (cualquiera)
- Intenta meter en el tour la siguiente ciudad no visitada
- Si no quedan ciudades por visitar, el algoritmo hace backtracking proponiendo una ciudad válida para el nivel anterior

IMPLEMENTACIÓN BACKTRACKING

```
double suma(list<int> rama, double* afinidades,int n){
  double sum=0;
  list<int>::iterator it;
  list<int>::iterator next_it;
  for(it=rama.begin(); it!=prev(rama.end());++it){
    next_it=it;
    ++next_it;
    sum += afinidades[(*it)*n+(*(next_it))];
  }
  sum+= afinidades[(*(rama.begin()))*n+(*(prev(rama.end())))];
  return sum;
}

list<int> BacktrackingTSP(list<int> rama, double* afinidades, list<int> sinusar,double& minima_suma, int n, double& cont, double& total)
{
  int tam = sinusar.size();
  double sum=suma(rama,afinidades,n);
  if(tam <= 1 && minima_suma>sum && sum>0){
    rama.splice(rama.end(),sinusar);
    minima_suma=sum;
    return rama;
  }else{
    list<int>::iterator it;
    list<int> res;
    list<int> aux2;
    for(it = sinusar.begin(); it != sinusar.end(); ++it){
      int aux = *it;
      it = sinusar.erase(it);
      rama.push_back(aux);
      aux2 = BacktrackingTSP(rama, afinidades, sinusar,minima_suma, n, cont, total);
      if(aux2.size() != 0){
        res = aux2;
      }
      it = sinusar.emplace(it,aux);
      rama.remove(aux);
    }
    return res;
  }

}
```

RESULTADOS Y GRÁFICAS (dataset custom.tsp)

DIMENSION: 11

1	38.24	20.42
7	38.42	13.11
6	37.56	12.19
5	33.48	10.54
9	41.23	9.1
10	41.17	13.05
3	40.56	25.32
2	39.57	26.15
4	36.23	23.12
8	37.52	20.44
11	36.08	-5.21

El tamaño del recorrido es: 46.8093

TIEMPO: 11 nodos, 6.38558s

RESULTADOS Y GRÁFICAS (dataset custom.tsp)