Concepção e Análise de Algoritmos

Tema 5 “Planeamento de Férias”

Partindo de uma cidade pré-definida (vértice), com um aeroporto próximo, pretende-se visitar um conjunto de locais de interesse (vértices obrigatórios). Existem também locais cuja visita não é obrigatória, e podem ser vistos como auxiliares para completar o percurso de uma forma mais vantajosa (rápida).

Cada percurso entre cidades tem um custo (tempo) associado (consideramos igual o custo de A para B e de B para A), e cada local de visita obrigatória tem também um custo associado, que é o tempo de visita. Existe um limite diário de horas despendidas entre viagens e visitas definido. O que se pretende é, encontrar um percurso que minimize o número de horas necessário para visitar todos os locais de interesse.

Quando se atinge o limite de horas diárias disponível, é necessário que o local possua alojamento. Os locais de interesse podem ou não possuir este atributo, e os locais auxiliares são obrigatoriamente deste tipo.

Para resolver o problema, pretende-se utilizar numa primeira fase programação dinâmica, onde para cada cidade se obtém o menor custo de viagem para qualquer outra, tendo em conta todas as cidades disponíveis. Segue-se a aplicação do algoritmo do Caixeiro Viajante.

Partindo da cidade com aeroporto próximo, utiliza-se uma pesquisa do tipo gananciosa, neste caso, dirige-se para a cidade cuja visita é obrigatória (e ainda não foi visitada) cujo percurso tem o menor custo associado, e marca-se como visitado. Quando se atinge o limite de horas diárias disponíveis ou não existirem percursos com um custo associado igual ou inferior ao tempo ainda disponível, procura-se por um caminho para um local de repouso (local que pode já ter sido visitado), cujo custo de viagem também não exceda o tempo disponível. Em caso de insucesso (não existir a partir da cidade actual nenhuma hipótese que permita a continuar a procura), se o local onde nos encontramos não possuir alojamento, é necessário regredir para o vértice anterior e explorar (caso exista) outro percurso.

A cada local visitado, adiciona-se à fila que guarda o melhor percurso.

A cada tempo diário esgotado, é incrementado o número de dias.

Formalização:

G= (V,A), V é o Set de N cidades a serem visitadas. A é o set de caminhos possíveis de percorrer.

Ca= (Cij) é o custo em horas associado a cada percurso i para j.

Cv= (Cj) é o custo em horas da visita à cidade j, se esta ainda não foi visitada.

M é o número máximo de horas diárias disponíveis.

Pj é a cidade onde se encontra.

Ud é o número de horas gastas no dia.

D é o número de dias acumulados.

R é uma cidade com aeroporto.

Xj é 1 se a cidade ainda não foi visitada e é de visita obrigatória, 0 para outros casos.

Dados de entrada:

Ca.

Cv.

Ud.

Dados de saída:

Ud.

Objectivo:

Maximizar diariamente ∑(i, j) Xj

Restrições:

Assegurar que a primeira viagem é feita a partir de uma cidade com aeroporto.

(Ud=0, D=0) → Pj = R.

Assegurar que a última posição é um local com aeroporto.

∑(i, j) Xj = 0 → Pj = R.

Garantir que não são ultrapassadas o tempo máximo diário disponível.

M – Ud≥ 0.