

SpeedMail: Correio Prioritário (tema 9)

Concepção e Análise de Algoritmos



Grupo X Turma 7

Luís Fernandes - up201706910@fe.up.pt

Nelson Gregório - up200900303@fe.up.pt

Paulo Moutinho - up201704710@fe.up.pt

18 de Abril de 2019

Descrição do tema	2
Identificação do problema	2
Formalização do problema	3
Dados de entrada	3
Dados de saída	4
Restrições	4
Função objetivo	4
Perspectiva de solução	5
Divisão do problema em subproblemas:	5
Pré-processamento	5
Procedimento a cada fase	5
Casos de utilização	6
Considerações e situações a averiguar	6
Conclusão	7

Descrição do tema

Uma empresa pretende estabelecer um sistema inteligente, onde encomendas são recolhidas num ponto de origem indicado pelo cliente e transportadas, por veículos da empresa, até aos seus respectivos pontos de destino.

Sendo o número de veículos limitado é preciso encontrar uma rota que minimize o tempo necessário para transportar as encomendas, tendo em conta possíveis obstáculos.

Identificação do problema

Dada a localização do centro de apoio, o número e respetivas capacidades de veículos disponíveis, a dispersão dos pontos e precedência de recolha face à entrega, é preciso configurar rota(s) de forma a minimizar o tempo necessário para satisfazer todas as encomendas. Existe também uma restrição que obriga a fazer a recolha de uma encomenda antes da sua entrega.

Vamos então modelar a rede de estradas através de um grafo, onde as arestas serão as ruas e os vértices as junções entre elas. As arestas podem ser unidirecionais ou bidirecionais de forma a representar o sentido do trânsito das ruas. Os pontos de recolha, de entrega e os centros de apoios são também representados por vértices no grafo.

Minimizar o custo da rota é redutível ao problema do **Caixeiro Viajante (TSP)**, sendo portanto **NP-difícil**, mas não **NP**. Isto é, é tão difícil quanto qualquer outro problema NP, no entanto não é possível encontrar a solução ótima (rota mais curta neste caso) em tempo polinomial.

Embora seja fácil calcular todas as rotas possíveis e encontrar a mais curta, a natureza combinatória do problema faz com que ao aumentar o número de vértices, a quantidade de rotas a verificar cresça com o fatorial desse número (vértices). É importante notar que certas restrições específicas ao nosso problema, diminuem o número máximo de rotas possíveis a verificar.

Soluções sub-ótimas são calculáveis através de algoritmos gananciosos com complexidade temporal aceitável face ao número de vértices.

Formalização do problema

Dados de entrada

→ V - conjunto de vértices (cruzamentos de ruas e pontos de interesse).

Cada ponto terá os seguintes parâmetros associados:

◆ Nome (termo identificativo)

◆ Coordenadas (localização geográfica)

→ E - conjuntos de arestas (ruas)

◆ Peso W - cada estrada terá associado um peso (depende do comprimento da estrada, grau de acessibilidade, etc)

→ $P(V_i, V_f)$ - tuplo que define os vértices de origem e destino de uma encomenda.

→ $G(V, E)$ - grafo dirigido pesado, em que os vértices representam os vários pontos no mapa e as arestas representam as estradas (unidirecionais ou bidirecionais) que ligam os vários vértices.

→ N - Conjunto de veículos representados pela sua capacidade

→ C - Vértice que define o centro de apoio

Dados de saída

- **R** - Conjuntos de vértices (ordenados) que representa a(s) melhor(es) rota(s) entre todos os pontos de recolha e entrega de encomendas
- ◆ **T** - Soma dos pesos de todas as arestas de uma rota em **R**

Restrições

- $\forall e \in E : W(e) > 0$
- $|N| > 0 \wedge \forall n \in N : n > 0$
- $T > 0$, pois é uma soma entre pesos (**W**) também positivos
- Uma encomenda só pode ser entregue se já foi recolhida
- O primeiro e último elementos de **R** são o centro de apoio
- Só é possível recolher uma encomenda se houver espaço para ela no veículo.

Função objetivo

A solução deste problema é obtida minimizando o peso total **T** das arestas percorridas em cada rota de **R** satisfazendo todas as encomendas pendentes. Assim sendo, função objetivo será a seguinte:

$$h = \min(\sum_{i=1}^{|R|} T(r)), r \in R$$

Perspectiva de solução

Divisão do problema em subproblemas:

1. Um veículo com espaço ilimitado.
2. Um veículo com espaço limitado.
3. Um veículo com espaço limitado e um centro de apoio
4. Múltiplos veículos com espaço limitado e um centro de apoio

Pré-processamento

Calcular distâncias de todos os vértices para todos os outros vértices usando o algoritmo **Floyd-warshall**

Procedimento a cada fase

Subproblema 1)

1. Determinar rota com menor custo (força bruta VS algoritmo ganancioso)
2. Minimizar custo (tempo de viagem/distância percorrida) da rota

Subproblema 2)

1. Determinar rotas (semelhante ao subproblema 1) tendo em atenção a capacidade máxima do veículo
2. Minimizar custo

Subproblema 3)

1. Determinar rotas (semelhante ao subproblema 2) tirando partido do centro de apoio.
2. Minimizar custo

Subproblema 4)

1. Determinar configuração de rotas, utilizando múltiplos veículos, minimizando o custo total destas
2. Minimizar custo (tempo de viagem/distância percorrida) de cada rota

Casos de utilização

O software que propomos criar vai, potencialmente, criar um grafo a partir de pontos (guardados num ficheiro em disco).

Vai calcular as distâncias entre todos esses pontos (este cálculo pode ser feito apenas uma vez, com o seu resultado também guardado em disco).

Irá apresentar um menu com opções para:

- introduzir/alterar/remover encomendas
- definir a localização do centro de apoio
- definir o tamanho e capacidades da frota de veículos disponíveis
- calcular rotas baseadas em diferentes critérios.

A visualização do grafo será feita recorrendo a material de apoio fornecido, inicialmente ao [GraphViewer](#), e posteriormente ao [OpenStreetMap](#)

Considerações e situações a averiguar

Com o intuito de avaliar qualitativamente as rotas calculadas, tencionamos colecionar métricas (obtidas através de geração aleatória de encomendas) do custo, sob a forma de tempo. Para isso, pensamos em definir uma velocidade média para o veículo que junto com a distância entre pontos permite calcular esse tempo.

Será também contabilizado um tempo constante para simular o ato de recolher/entregar uma encomenda. É assumido que a configuração de encomendas no veículos é feito de acordo com a sua ordem de entrega, daí ser constante.

Tencionamos também estudar a utilidade de um algoritmo (pesquisa em profundidade/largura) para avaliar a acessibilidade a um determinado ponto, contra tentar calcular imediatamente o caminho para ele, uma vez que o grafo representa uma zona urbana, com nós bastante interligados (possivelmente poucos casos de reduzida acessibilidade).

Outra ideia é, mediante uma avaliação de acessibilidade ao centro de apoio, um valor é atribuído a cada ponto e é usado como peso para definir prioridade de visita.

Mais uma vez, com um centro de apoio localizado no centro urbano, é esperado que a acessibilidade diminua radialmente a partir desse centro.

Na fase de múltiplos veículos, avaliar métodos que permitam atos de recolha e entrega na mesma rota contra rotas dedicadas a recolha ou entrega. A noção pré-implementação é que a variação de tempo do processo todo (colocar pedido até entregar encomenda) será menor com rotas dedicadas, algo que pode ser desejável pois mostra consistência do serviço. No processo de escolha de encomendas a serem entregues, é importante controlar o número de vezes que uma encomenda é adiada, de forma a manter “constante” o tempo de entrega entre todas.

Conclusão

O problema a resolver, embora simples, é computacionalmente complexo, e constitui uma área actual de investigação.

O principal obstáculo que antevemos é dificuldade em escalar qualquer um dos parâmetros, particularmente a área da zona servida.

Com estas noções, e as considerações previamente indicadas, consideramos o estudo dos métodos utilizados e a sua aplicação ao problema mais relevante que a qualidade das soluções que se obterá.

Todos os membros participaram de equivalente forma para a redação do relatório, não havendo diferenças na participação individual de cada um.

Nota:

Por lapso da nossa parte, não guardamos a letra do nosso grupo.

Infelizmente, não está também registada essa informação no documento "Grupos de Trabalho" do moodle.

Deste modo, colocamos X como letra.

Pedimos desculpa por qualquer inconveniência que isto possa causar.