Relatório preliminar de Conceção e Análise de Algoritmos

Francisco Veiga, 201201604@fe.up.pt
João Cabral, up201304395@fe.up.pt
João Mota, 201303462@fe.up.pt
Faculdade de Engenharia da Universidade do Porto

março 2015

Conteúdo

1 Introdução

No contexto da unidade curricular de Conceção e Análise de Algoritmos foi solicitada a resolução de um problema relacionado com a distribuição de uma rede de fibra ótica pela rede habitacional de um determinado agregado populacional.

2 Problemas a abordar

Numa primeira fase é solicitada uma aplicação que receba como dados de entrada um mapa do referido agregado populacional e produza como saída uma representação gráfica, sob a forma de um gráfico, de uma distribuição ideal da rede de fibra ótica, minimizando o comprimento das ligações utilizadas. Estabelece-se ainda como restrição adicional que cada uma das casas cobertas não pode situar-se fora de uma determinada área a ser definida por uma distância máxima à central de onde parte a rede de fibra ótica.

Numa segunda fase é solicitado que a aplicação alargue o raio de ação de cobertura da rede de fibra ótica, procurando abranger uma maior área, sendo contudo necessário que a aplicação seja capaz de detetar áreas onde a cobertura providenciada por uma única central se revele insuficiente e seja capaz de indicar a necessidade de existirem novas centrais de distribuição da rede.

3 Formalização do problema

3.1 1º problema/fase

Inputs

- Um grafo $G = (V, E)$ conexo, onde V é o conjunto das casas e E o conjunto das suas ligações e para cada aresta $(u, v) \in E$ temos $w(u, v)$ representando a distância entre as arestas u e v e $d(p)$ como sendo a distância de um caminho $p = \langle v_0, v_1, \ldots, v_k \rangle$[1, p. 624];
- Uma distancia $l : l > 0$;
- Um vértice $s : s \in V$.

Outputs

- Um grafo $G_T = (V_T, E_T)$ tal que se verifique a condição[1, p. 643]:

$$\forall v \in V(\delta(s, v) \leq l \leftrightarrow v \in V_T)$$

$$\delta(s, v) = \begin{cases} \min\{d(p) : s \stackrel{p}{\leadsto} v\} & \text{se existe caminho entre s e v} \\ \infty & \text{caso contrário} \end{cases}$$

Função Objetivo

Seja

$$x(u, v) = \begin{cases} 1 & \text{se a aresta}(u, v) \in E_T \\ 0 & \text{caso contrário} \end{cases}$$

A função objetivo é[2]

$$\min \sum_{(u,v) \in E} w(u, v)x(u, v)$$

com as restrições

$$\sum_{(u,v) \in E} x(u, v) = |V_T| - 1$$

$$\sum_{(u,v) \in (S,S)} x(u, v) \leq |S| - 1 \quad \forall S \subseteq V_T$$

3.2 2º problema/fase

Inputs

- Um grafo $G = (V, E)$, desconexo ou não, onde V é o conjunto das casas e E o conjunto das suas ligações e para cada aresta $(u, v) \in E$ temos $w(u, v)$ representando a distância entre as arestas u e v e $d(p)$ como sendo a distância de um caminho $p = \langle v_0, v_1, \ldots, v_k \rangle$[1, p. 624];

Outputs

- Um grafo $G_T = (V_T, E_T)$;
- Um conjunto de grafos $D \subseteq G_T = \{d(V_D, E_D) : \forall d, e \in D(\nexists v \in V_T(v \in d \wedge v \in e)) \wedge \nexists d, e \in D(\exists (u, v) \in E_T(u \in d \wedge v \in e))\}$
- Um conjunto $S = \{s : \exists! d \in D(s \in d)\}$.

Função Objetivo

Seja

$$x(u, v) = \begin{cases} 1 & \text{se a aresta}(u, v) \in E_T \\ 0 & \text{caso contrário} \end{cases}$$

A função objetivo é[2]

$$\min \sum_{(u,v) \in E} w(u, v) x(u, v)$$

com as restrições

$$\sum_{(u,v) \in E} x(u, v) = |V_T| - 1$$

$$\sum_{(u,v) \in (S,S)} x(u, v) \leq |S| - 1 \quad \forall S \subseteq V_T$$

4 Ideias de solução

4.1 1º problema/fase

Aplicar o *algoritmo de Dijkstra*, um algoritmo ganancioso, para calcular o caminho mais curto para todos os pontos do grafo a partir da central de forma a aferir os vértices do grafo que se encontram dentro do raio de cobertura pretendido. Excluir do grafo todos os pontos cuja distância à central seja superior ao raio desejado.

De seguida usar o *algoritmo de Prim*, também ele um algoritmo ganancioso, para calcular uma árvore de expansão mínima do grafo resultante da operação anterior, resultando numa distância ótima de cabo a utilizar.

4.2 2º problema/fase

Para a segunda fase prevê-se um aumento do raio de abrangência para a central. Isto pode levar à existência de sub-grafos desconexos. É da nossa interpretação que a cada grafo desconexo se deve acrescentar uma central. O *algoritmo de Prim* só permite chegar à árvore de expansão mínima de um dos sub-grafos desconexos de G por cada vez que é corrido. Embora existam algoritmos que permitem calcular árvores de expansão mínima em grafos desconexos, gerando uma árvore em cada sub-grafo, o cálculo do conjunto D essencial para obter S implica necessariamente uma determinação destes subgrafos.

5 Métricas de avaliação

Para obter uma métrica da eficiência espacial e temporal dos algoritmos pretende-se aplicar chamadas às funções do sistema operativo que permitem obter informações acerca do tempo de execução e da ocupação de memória de um determinado código. Se no caso da eficiência temporal esta informação é de aplicação mais direta, podendo ser obtida imediatamente antes e depois da execução do algoritmo cuja eficiência se pretende medir, no caso da eficiência espacial torna-se essencial escolher pontos relevantes para efetuar a medição, dado que ao longo da execução de um programa a sua ocupação em memória pode variar drasticamente.

É ainda importante ter em conta para a medição do tempo de execução de um processo que as chamadas ao sistema que permitem obter a sua ocupação em memória têm um tempo de execução não nulo, pelo que convém efetuar as duas medições separadamente.

5.1 Variação das entradas

5.1.1 1º problema/fase

Torna-se importante variar o número de nós do grafo bem como o seu número de arestas separadamente, para aferir do impacto de cada uma das variáveis no tempo de execução de cada um dos algoritmos aplicados. Torna-se também importante variar o número de arestas a serem excluídas pela aplicação do algoritmo de Dijkstra antes do cálculo da árvore de expansão mínima.

5.1.2 2º problema/fase

Aplica-se a mesma necessidade de variar o número de grafos e o número de nós que se verifica na 1ª fase. Torna-se também importante variar a conectividade dos grafos, testando os algoritmos com um número variável de subgrafos desconexos, para avaliar o impacto que isso tem na performance de cada um.

Referências

[1] Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, third edition, 2009.

[2] Dorit S. Hochbaum. Network flows and graphs - lecture 25. Disponível em http://www.ieor.berkeley.edu/ ieor266/Lecture25.pdf (2015/03/26).