

5

Variações e Heurísticas

Este capítulo apresentará algumas variações e heurísticas válidas para os algoritmos A1 e A2 descritos no Capítulo 4. As variações expostas estão baseadas na forma com que o procedimento inicial (linha 1) dos algoritmos cria a árvore geradora (*spanning tree*). As heurísticas atuam em dois pontos: na decisão de contrair o grafo e na decisão de adicionar mais de uma aresta na mesma iteração.

5.1

Variações na função Árvore Geradora

Duas diferentes funções foram implementadas nos algoritmos. Ambas retornam árvores geradoras aleatórias, isto é, não são, obrigatoriamente, máximas ou mínimas, e utilizam-se de recursão. Chamaremos uma função de *Árvore Geradora Estrela* e a outra de *Árvore Geradora DFS*, esta, por uma razão óbvia, utiliza o método de busca *Depth-first search* (Busca em Profundidade).

A função *Árvore Geradora Estrela* (AGE) busca gerar uma árvore com o maior número possível de ramificações, ou seja, ter uma quantidade grande de nós com três ou mais arestas incidentes. O objetivo é diminuir o caminho $i-j$ ao adicionarmos uma nova aresta ao grafo com o intuito de reduzir o número de execuções do algoritmo de fluxo máximo, e, possivelmente, o número de nós no grafo contraído. A Figura 5.1 ilustra um pseudocódigo do algoritmo AGE.

De modo contrário, a função *Árvore Geradora DFS* tende a gerar uma árvore com um número pequeno de ramificações, ou seja, a maioria dos nós possuindo, no máximo, duas arestas incidentes. Seu objetivo é obter um caminho $i-j$ maior, aumentando, assim, a possibilidade de adicionar mais de uma aresta por iteração. Isto se refletiria em um número reduzido de iterações no algoritmo, e, conseqüentemente, menos processos de contração e execuções do algoritmo de fluxo máximo. A Figura 5.2 ilustra um pseudocódigo do algoritmo DFS.

procedimento AGE (G, v);

```

1  Marque o vértice  $v$  como incluído;
2  para todas arestas  $e_k = (v, v_k)$  onde  $k = 1, 2, \dots$  faça
3      se  $v_k$  não está incluído
4          Marque  $v_k$  e  $(v, v_k)$  como incluídos;
5          Marque  $v$  como pai de  $v_k$ ;
6      finalize se
7  finalize para
8  para todas arestas  $e_k = (v, v_k)$  onde  $k = 1, 2, \dots$  faça
9      se  $v$  é pai de  $v_k$ 
10         Chame recursivamente AGE ( $G, v_k$ );
11     finalize se
12 finalize para
13 retorne  $T$ ;
finalize AGE;
```

Figura 5.1: Pseudocódigo do algoritmo AGE.

procedimento DFS (G, v);

```

1  Marque o vértice  $v$  como incluído;
2  para todas arestas  $e_k = (v, v_k)$  onde  $k = 1, 2, \dots$  faça
3      se  $v_k$  não está incluído
4          Marque  $v_k$  e  $(v, v_k)$  como incluídos;
5          Chame recursivamente DFS ( $G, v_k$ );
6      finalize se
7  finalize para
8  retorne  $T$ ;
finalize DFS;
```

Figura 5.2: Pseudocódigo do algoritmo DFS.

As Figuras 5.4 e 5.5 exemplificam, respectivamente, as aplicações das funções AGE e DFS à rede G da Figura 5.3. Para ambas as funções, o vértice 1 é o inicial; o primeiro filho é sempre o nó de número menor, o segundo filho é o nó com o segundo menor número, e assim por diante.

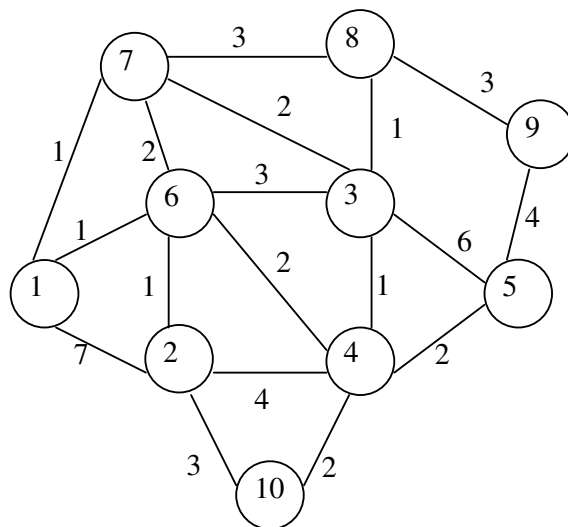


Figura 5.3: Rede G na qual será aplicada as funções das árvores geradoras.

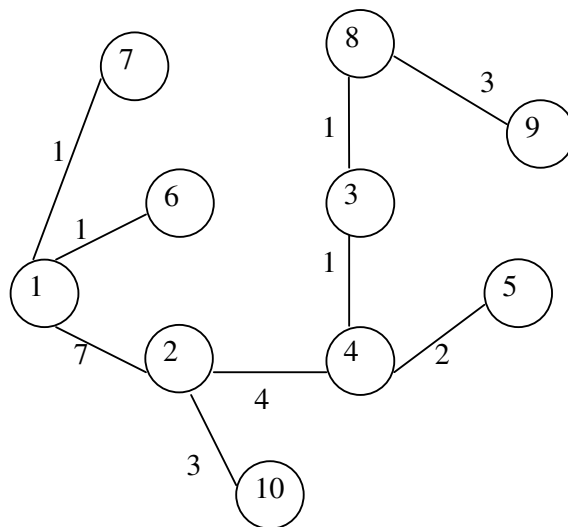


Figura 5.4: Resultado da aplicação da função *Árvore Geradora Estrela* em G .

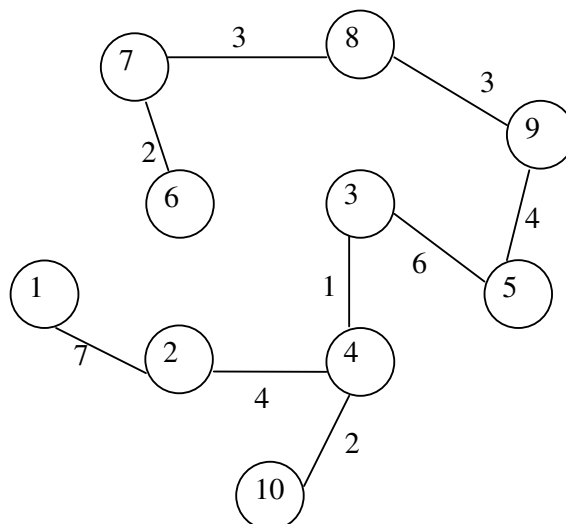


Figura 5.5: Resultado da aplicação da função *Árvore Geradora DFS* em G .

A diferença entre as duas funções é observável no exemplo das Figuras 5.4 e 5.5. Na árvore gerada por AGE, três vértices, a ser $\{1, 2, 4\}$, possuem, cada um, três arestas incidentes. O vértice $\{4\}$ é o único, na árvore gerada por DFS, com três arestas incidentes.

5.2 Heurísticas

Duas heurísticas foram elaboradas para o algoritmo, podendo as duas atuarem ao mesmo tempo. Elas são as seguintes tomadas de decisão:

1. Somente adicionar mais de uma aresta por iteração quando o caminho $i-j$ for suficientemente grande;
2. Somente contrair o grafo quando o número de nós do grafo contraído for suficientemente pequeno.

A primeira chamaremos de *Heurística de Adição de Arestas* (HAA) e a segunda de *Heurística de Contração do Grafo* (HCG).

A motivação por trás da HAA é não "gastar tempo" procurando novas arestas a serem adicionadas quando a possibilidade de encontrá-las é pequena

devido ao curto caminho $i-j$. Note que esta heurística não é aplicável ao algoritmo A1.

Ao mapear as subárvores T_a^b , o algoritmo obtém a quantidade de nós do grafo contraído. Caso esse número seja próximo do número de nós do grafo, não é eficiente gerar um novo grafo de tamanho relativamente igual ao já existente. Evitar isto é o objetivo da HCG.

A Figura 5.6 ilustra o pseudocódigo do algoritmo A2 com as heurísticas HAA (linha 9) e HCG (linha 13) implementadas.

procedimento A2HH (G);

```

1   $T \leftarrow$  crie uma árvore geradora (spanning tree) do grafo  $G$ ;
2   $G_0 = T$ ;
3   $CT_0 = G_0$ ;
4   $E' \leftarrow$  conjunto de arestas que estão em  $G$  mas que não estão em  $G_0$ ;
5  para  $e_k \in E'$  onde  $k = 1, \dots, m - n + 1$  faça
6      se  $e_k$  não está incluído no grafo  $G_{k-1}$ 
7          Adicione  $e_k$  ao grafo  $G_{k-1}$  criando  $G_k$ ;
8          Identifique caminho  $i-j$  em  $CT_{k-1}$ ;
9          se caminho  $i-j$  é suficientemente grande
10             Adicione  $e_w$  caso suas extremidades estejam contidas
                  no caminho  $i-j$ , para  $w = k + 1, k + 2, \dots, m - n + 1$ ;
11         finalize se
12         Mapeie subárvores  $T_a^b$  em  $CT_{k-1}$ ;
13         se número de nós do grafo contraído é suficientemente pequeno
14             Gere grafo contraído de  $G_k$ ;
15             Calcule, com algoritmo de Gusfield, árvore de cortes dos nós do
                  caminho  $i-j$  no grafo contraído;
16         se não
17             Calcule, com algoritmo de Gusfield, árvore de cortes dos nós do
                  caminho  $i-j$  no grafo  $G_k$ ;
18         finalize se
19         Substitua caminho  $i-j$  pela árvore de cortes em  $CT_{k-1}$  gerando  $CT_k$ ;
20     finalize se
21 finalize para
22 retorne  $CT$ ;
finalize A2HH;
```

Figura 5.6: Pseudocódigo do algoritmo A2 com as heurísticas HAA e HCG.