

Busca em Profundidade

(SCHWARZ, Gaston Adair. Grafos – Introdução e Processos de Busca. Apostila, 1998)

Considerações

Uma busca é dita em profundidade quando o critério de escolha do nó marcado obedecer a: “*dentre todos os nós marcados e incidentes a algum arco ainda não explorado, escolher aquele mais recentemente alcançado na busca*”.

A busca em profundidade é implementada com o auxílio de uma pilha, onde inserimos todos os nós gerados e retiramos sempre o de cima (último), quando for necessário.

A regra usada, portanto, para proceder uma busca é do tipo LIFO (*Last In First Out*), isto é, o último nó gerado será o primeiro nó a gerar seus sucessores.

Vamos usar, ainda, as definições de nó aberto e nó fechado. Convencionamos que o nó está aberto quando ele é gerado e colocado na fila. Quando o nó for retirado da fila, ele é dito *fechado* e é inserido numa lista de nós fechados. Expandir um nó significa gerar seus sucessores.

Este algoritmo é também conhecido na literatura como algoritmo de *busca vertical*.

A busca em profundidade explora cada caminho possível para chegar ao nó final, antes de tentar qualquer outro. Convém, em alguns casos, limitar o nível ou limite de profundidade de busca, para evitar uma pesquisa sem fim em determinado caminho ou um tempo muito grande de processamento que inviabilize todo o processo.

O limite de profundidade K pode ser interpretado como sendo “*o preço que se está disposto a pagar por uma solução viável, sem questionar a possível existência de outra solução mais econômica*”.

Uma vez que o limite de profundidade seja ultrapassado, assegura-se que a solução encontrada não tem custo adicional maior que R em relação à solução ótima.

Este algoritmo tem, portanto, aplicação limitada no que tange a obtenção do caminho ótimo.

Presta-se para a resolução de problemas extremamente complexos, no qual deseja-se obter uma solução viável, não necessariamente ótima.

Algoritmo de Busca em Profundidade

(*Depth-First Search Algorithm*)

Inicialização

1. Descrição detalhada de cada nó $x_i \in X$, incluindo nós iniciais $s \in S$ e terminais $t \in T$.
2. Definição de $\Gamma(x_i)$.
3. Criar três listas:
 A = lista de nós abertos = lista de S ;
 F = lista de nós fechados = ϕ ;
 V = lista auxiliar = ϕ .
4. Criar um conjunto de apontadores (predecessores) $P(x_i)$.
5. Fazer $g(s) = 0 \quad \forall s \in S$.
6. Fazer $P(s) = \phi \quad \forall s \in S$.
7. Inicializar:
 K = limite de profundidade;
 R = incremento no limite de profundidade.

Algoritmo (com limitação de profundidade)

1. Se $A \neq \phi$, vá para 2.
Senão, faça $A = V$, $K = K + R$ e $V = \phi$. Se $A = \phi$, pare com fracasso.
2. Remova v de A e inclua em F .
(escolha v seguindo a regra LIFO).
3. Se $g(v) > K$, faça $V = V \cup \{v\}$ e volte ao passo 1.
Se $v \in T$, pare com sucesso. Se não, gere $\Gamma(v)$. Se $\Gamma(v) = \phi$, volte a 1.

4. Para cada $m \in \Gamma(v)$ faça:

a) Se $m \notin A \cup F$:

$$P(m) = v;$$

$$g(m) = g(v) + C(v, m); \text{ e}$$

$$A = \{m\} \cup A.$$

b) Se $m \in A \cup F$:

Se $g(m) > g(v) + C(v, m)$, faça:

$$P(m) = v$$

$$g(m) = g(v) + C(v, m); \text{ e}$$

se $m \notin A$ faça $A = \{m\} \cup A$; a

se $m \in V$ faça $V = V - \{m\}$.

5. Volte a 1.

Saída de Resultados

1. Recuperar a solução através dos apontadores $P(x_i)$.
2. Emitir relatório.