

Trabalho Prático 1: The city is on fire!

Gabriel Henrique Souto Pires {gabrielpires@ufmg.br}

1 Introdução

Neste problema são dados um conjunto de vértices que representam os bairros de uma cidade e as arestas que representam as ruas que ligam os bairros. Você é um bombeiro preguiçoso e gostaria de trabalhar o mínimo possível, então ao sair de um corpo de bombeiros para outro seria ideal evitar passar pelos caminhos onde a probabilidade $P(u, v)$ de ter um incêndio é alta entre os bairros u e v , sendo que $P(u, v) = P(v, u)$ e todas as ruas são de mão dupla. Durante o trajeto, também é necessário ficar a uma distância de no máximo k bairros de algum corpo de bombeiros, dessa forma outros bombeiros podem vir apagar o incêndio no seu lugar.

A tarefa neste TP é descobrir o caminho que respeita as restrições descritas acima, ou seja, um caminho que passe por bairros que tenham a menor probabilidade de incêndio possível tal que a probabilidade total de incêndio entre o bairro de saída e o de chegada seja mínima e o caminho passe sempre por vértices que estejam a uma distância máxima k de algum corpo de bombeiros.

2 Solução do Problema

A probabilidade $P(u, v)$ de ter um incêndio em dado trecho pode ser interpretada como o peso das arestas entre os vértices do grafo que são os bairros da cidade. Desta forma, o caminho mais curto é aquele em que o peso total das arestas é mínimo, ou seja, a probabilidade de incêndio é menor. Para resolver o problema, foi criada uma lista¹ de adjacência para representar o grafo (os vértices e as arestas que ligam os vértices adjacentes a eles). A lista de adjacência é basicamente um vetor com uma posição para cada vértice, cada posição do vetor contém uma lista encadeada onde são inseridos os vértices adjacentes ao vértice referente à posição atual do vetor.

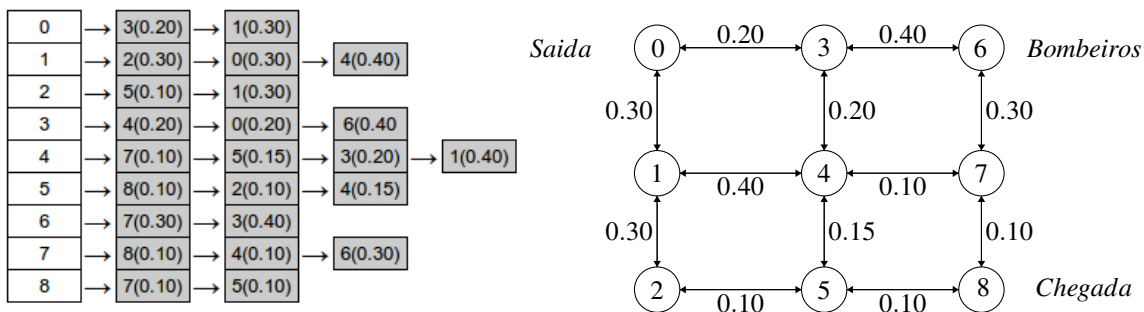


Figure 1: Representação do grafo (direita) em forma de lista de adjacências (esquerda)

Uma vez que os vértices que estão a uma distância k maior que a máxima permitida não podem ser percorridos, o menor caminho nunca passará por eles, então esses vértices devem ser desconsiderados na hora de se procurar a solução no grafo. Isso foi feito com uma *busca em largura* onde para cada vértice com corpo de

¹Os TADs Lista, Fila e Heap utilizados neste TP foram feitos usando como referência o código desenvolvido pelos professores Thiago Noronha e William Schwartz durante a disciplina de AEDS 2 no semestre passado.

bombeiros uma busca é feita e apenas os vértices a uma distância máxima k do vértice de origem são marcados como válidos em um vetor. Dessa forma, ao se rodar o algoritmo que acha o caminho mínimo, os vértices inválidos não são nem testados, o que torna o algoritmo mais rápido e evita que caminhos mínimos que passam por vértices inválidos sejam considerados no final da execução. Como visto no pseudo código abaixo, uma fila é utilizada

Algoritmo 1 Pseudo código do loop principal da busca em largura

```
while (!filaVazia){
    if (distPai < k){
        for (Percorre a lista(vertice[i]) até o final da lista){
            enfileira(nodoAtual, distPai+1);
            vertsValidos[nodoAtual] = 1;
        }
    }
    desenfileira(fila);
    i = frenteFila(fila);
}
```

Com os vértices válidos marcados no vetor, o grafo foi submetido ao algoritmo de *Dijkstra* que acha o menor caminho entre dois vértices em um grafo onde as arestas não tem peso negativo e como as arestas no problema em questão não terão valor negativo uma vez que a probabilidade de incêndio varia de 0 a 1, isso não será um problema.

O *Dijkstra* recebe como parâmetro o grafo em forma de lista de adjacência, o vértice de origem, o número de quarteirões, o vetor de vértices válidos gerado pela busca em largura e o número do vértice de chegada. Na função os vértices válidos são percorridos a fim de se encontrar o melhor caminho entre o vértice de origem e o de chegada. Nessa função, cada vez que se chega até um vértice todas as suas arestas são inseridas em um *Heap* na forma de P_N , ou melhor, $P(s,u) * (1 - P(u,v))$, sendo que s é o vértice de saída, u é o vértice atual e v é o próximo vértice a ser percorrido. Esta probabilidade representa a chance de não haver incêndio no caminho até a origem. Desta forma é possível saber qual aresta percorrer em $O(1)$ já que basta apenas retirar a maior aresta do Heap (Max-Heap) e seguir a partir dela repetindo o processo até que se ache a solução.

Ao final do algoritmo, a maior probabilidade de não haver incêndio é armazenada na posição de um vetor de probabilidades referente ao vértice de chegada. Essa posição é então exibida na tela no formato $1 - prob[c]$, que representa a probabilidade de incêndio no melhor caminho possível até o vértice de chegada.

3 Análise Teórica do Custo Assintótico

3.1 Análise Teórica do Custo Assintótico de Tempo

3.2 Análise Teórica do Custo Assintótico de Espaço

4 Análise Experimental do Custo Assintótico

5 Conclusão