Pesquisa aplicada à evacuação

Relatório Final

Mestrado Integrado em Engenharia Informática e Computação

Inteligência Artificial

Turma 2:
Carlos Freitas - up201504749
Luís Martins - up201503344
Vicente Espinha - up201503764

Faculdade de Engenharia da Universidade do Porto
Rua Roberto Frias, sn, 4200-465 Porto, Portugal

20 de Maio de 2018

Conteúdo

1 Objetivo

O objetivo deste trabalho é determinar um plano de evacuação de um conjunto de turistas retidos numa montanha, em que estão disponíveis veículos de transporte localizados em n pontos/locais estratégicos. Para além do número de veículos e da sua localização, é também conhecido o local do abrigo para onde devem ser evacuados os turistas. Os veículos possuem capacidade limitada e só existe um veículo em cada um dos n pontos estratégicos considerados. Determinar-se-á um bom percurso para cada ambulancia para evacuar todos os turistas no menor tempo, considerando que todas as ocorrências se dão em simultâneo. Se o veículo de transporte não possuir capacidade suficiente para transportar todos os turistas que se encontram num determinado local, poderá efetuar mais que uma viagem, ou poder-se-á optar por usar um segundo veículo, conforme o que for mais eficiente.

Este trabalho tem ainda como objetivo a utilização e compreensão aprofundada do algoritmo de A* para resolver o problema em questão, sendo que foi implementado também o algoritmo Dijkstra por forma a poder comparar resultados.

2 Especificação

2.1 Algoritmo

Por forma a não existir uma explosão combinatória de casos a analisar, o grupo estabeleceu as seguintes fases de decisão do algoritmo para escolha dos veículos de transporte e da ordem dos pontos de recolha:

- Escolha de veículo

Inicia-se o algoritmo pela escolha do melhor veículo disponível. Para tal, são tidos em conta os seguintes aspetos com diferentes pesos:

- É calculado o ***centróide*** dos pontos nos quais existem turistas para resgatar, sendo determinada a distância entre cada um dos veículos existentes e o centróide. Este fator tem um peso de 50%;
- É calculada a ***média + desvio padrão*** do número de passageiros existentes em cada um dos pontos a resgatar, com o objetivo de determinar, de entre os veículos existentes, qual aquele cuja capacidade melhor se adapta à distribuição de turistas a resgatar (considera-se a soma do desvio padrão por forma a valorizar a escolha de veículos com capacidade excedentária, em detrimento de veículos com capacidade insuficiente). Apesar de à partida este fator poder não ser muito determinante, permitirá em alguns casos escolher um veículo que, embora seja um pouco pior nos outros parâmetros em consideração, se adapte melhor à distribuição do número de montanhistas a resgatar, e como tal permita reduzir o número de viagens necessárias para resgatar todos os montanhistas. Este fator tem um peso de 10%;
- É tida em consideração a ***distância percorrida pelo veículo até ao momento da escolha***, uma vez que, e considerando que todas ocorrências se dão em simultâneo, apenas após percorrer essa distância este poderá executar o resgate que se segue. Esta consideração é importante para que se possa desvalorizar veículos que já tenham resgatado turistas anteriormente, e que pelo facto das ocorrências se darem em simultâneo poderão não ser a melhor escolha. Este fator tem um peso de 40%.

Todos estes parâmetros são normalizados para valores entre 0 e 1, considerando como 1 o maior valor do respetivo parâmetro. Esta adaptação é importante para que possam ser somados de acordo com os seus pesos e posteriormente comparados. Assim para cada veículo o seu valor final para comparação será:

$$valor = 0.5 * \frac{|posicao\_veiculo - centroide|}{dmax} + 0.1 * \frac{|capacidade\_veiculo - (media + desvio\_padrao)}{cmax} + 0.4 * \frac{distancia\_percorrida\_veiculo}{dpmax}$$

em que:

- dmax: posição do veículo que se encontra mais longe do centróide - posição do centróide
- cmax: maior diferença entre o número de passageiros que um veículo leva e a soma da média com o respetivo desvio_padrão da distribuição de montanhistas a resgatar
- dpmax: distancia percorrida pelo veículo com maior distancia percorrida até ao momento

Esta escolha é um pouco *naive* uma vez que, apesar de ter em conta alguns parâmetros da distribuição dos pontos de resgate e respetivo número de montanhistas a resgatar, não está determinado à partida qual o ponto de resgate para onde o veículo se irá deslocar e como tal, o veículo escolhido, apesar de ser globalmente o melhor, poderá não o ser para o ponto de resgate para onde irá. Esta é, no entanto, uma boa abordagem, pois era necessário limitar alguns parâmetros, para que fosse possível obter uma solução válida em tempo útil.

- Caminho a percorrer pelo veículo escolhido

Após a escolha do melhor veículo a utilizar no socorro, é necessário determinar quais os pontos pelos quais este irá passar para resgatar turistas. São então tidas em conta as seguintes considerações:

- É criado um grafo em que os seus nós representam apenas os locais de resgate(com pessoas), o ponto de refúgio, e do local do veículo escolhido. As arestas representam o percurso entre estes nós no mapa principal. Assim, o custo das arestas neste grafo minimizado, são calculadas a partir de outro cálculo do algoritmo A*.

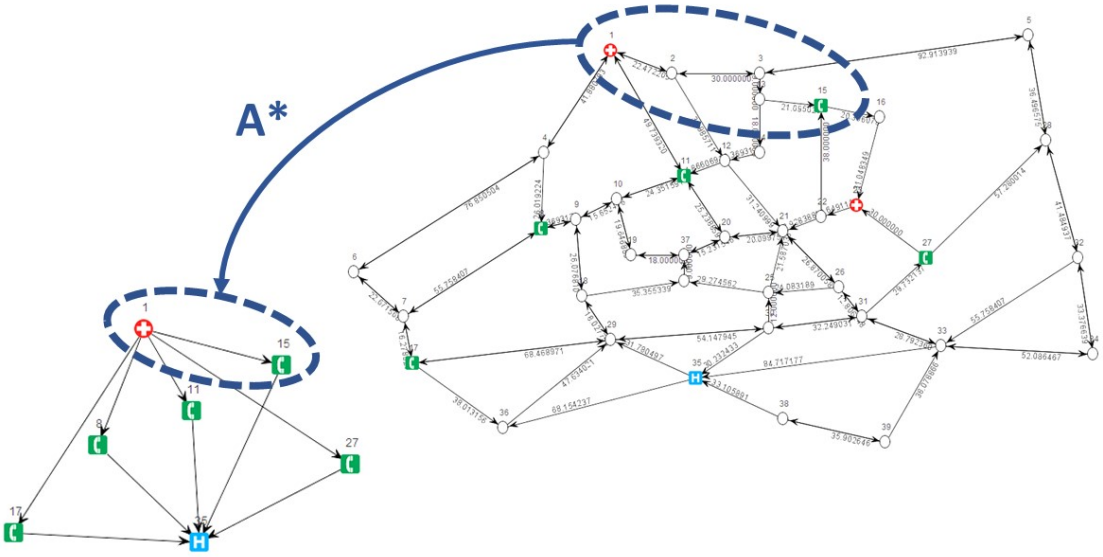

Figura 1: A* de cálculo do caminho entre 2 pontos

- Procede-se então à escolha do próximo ponto para onde o veículo se dirigirá relativamente ao grafo externo, através da seguinte função de transição:

$$f(n) = g(n) + h'(n)$$

Sendo g(n) o custo para chegar desde o ponto incial até ao ponto n e h'(n) a heurística dada pela distância de manhattan desde o ponto n até ao ponto final.

Para o cálculo de g(n) são considerados então os seguintes parâmetros:

- *espaço disponível no veículo*, após a recolha total ou parcial dos turistas existentes no ponto n. Este valor está compreendido entre 0 e 1 e é igual a divisão entre o número de lugares disponíveis após recolha em n e a capacidade total do veículo (80%);
- *cálculo da distância que é necessário percorrer desde o ponto atual até n* segundo a aplicação de um novo algoritmo A* (20%).

  Estes parâmetros são importantes para que o ponto escolhido seja aquele que ponderadamente se encontra mais perto do ponto inicial mas ao mesmo tempo tira proveito da capacidade do veículo tentando preenchê-lo.

  Em relação à aplicação do segundo algoritmo A* (do percurso entre os nós), o algoritmo terá como heurística a distância de Manhattan entre o ponto atual do veículo e o próximo potencial ponto do algoritmo demonstrado na figura abaixo.

- Condição de paragem

Após a escolha de um veículo e do ponto de resgate por este irá passar as respetivas estruturas são atualizadas, diminuindo o número de lugares disponíveis no veículo em consideração, e o número de pessoas por resgatar no ponto escolhido, incrementando a distância percorrida pelo veículo.

O algoritmo executa iterativamente até que não existam mais pontos de resgate com pessoas.

3 Desenvolvimento

Ferramentas e Ambiente de Desenvolvimento

O projeto foi desenvolvimento unicamente no sistema operativo Windows 10. A linguagem de programação utilizada foi C/C++, com o uso de Eclipse Kepler como Ambiente de Desenvolvimento de Trabalho (IDE).

Visto que o tema trabalhado é baseado em grafos que representam mapas, foi utilizado uma ferramenta gráfica denominada GraphViewer, que representa grafos, cedida na cadeira de Cálculo e Análise de Algoritmos .

Estrutura da Aplicação

Para a implementação dos algoritmos e heuristicas associadas foram então criadas algumas classes. Assim existem as seguintes estruturas:

Emergencia A classe Emergência é a base do programa. É a partir dela que são chamadas as funções de criação (a partir dos ficheiros de input necessários - ficheiros de texto), visualização e animação dos grafos com o apoio da ferramenta GraphViewer. Esta classe é também responsável pela invocação das funções responsáveis pela aplicação dos algoritmos/heurísticas.

Graph A classe Graph representa um grafo e é constítuida por outras classes como Edges, que representa as arestas de um vértice para o outro, e Vertex que representa os vértices do grafo.

No A classe Nó representa, tal como o Vertex, um ponto do grafo, sendo que é nesta que são mantidas as informações relativas à posição e numero de pessoas a resgatar no respetivo ponto.

Path Classe que representa o percurso necessário percorrer para ir de um ponto até outro. Esta classe mantém por isso uma lista de nós que constituem o caminho, bem como o distancia total do mesmo.

Hospital Classe que representa um No do grafo onde se encontra um ponto de refúgio para onde devem ser deslocados os montanhistas socorridos. Esta classe extende a classe No.

Veiculo Classe que representa um veículo e mantem as estruturas relativas à capacidade, nó onde se localiza bem como distância já percorrida por este.

As restantes classes, são apenas auxiliares.

Detalhes relevantes da implementação

Para a obtenção da melhor solução possível recorreu-se a vários métodos, tais como, o uso do cálculo do centróide de forma a determinar qual o(s) melhor(es) veículo(s) para resgatar os montanhistas presentes nos vários pontos de resgate. É também relevante referir o uso de dois algoritmos A*, um para calcular o melhor percurso de um ponto ao outro (por exemplo, de um ponto de resgate ao ponto de refúgio) e outro para calcular qual o ponto de resgate para onde um veículo, após ter sido selecionado, se deve dirigir, de forma a melhor aproveitar a sua posição relativa e a sua capacidade.

4 Experiências

4.1 Teste de integridade

Após a implementação das heurísticas e algoritmos propostos foram realizados alguns testes para verificação da plausabilidade dos mesmos, tendo-se concluído que apesar de não se obter o resultado ótimo os valores obtidos são razoavelmente bons. Esses testes foram os seguintes:

- Verificar se eram utilizados diferentes veículos no salvamento de turistas, tendo em conta o algoritmo descrito anteriormente neste relatório(pág.4).
- Verificar se um veículo seria utilizado no plano de evacuação de turistas, apesar de já ter feito pelo menos uma passagem pelo ponto de resgate e ter ido ao hospital (algoritmo descrito na pág.4).
- Verificar se um veículo poderia passar por um ponto de resgate, não socorrer todos os turistas, deixando alguns no local, e outro veículo resgatar as pessoas que faltam.

Consideremos o exemplo presente na Figura 2, onde se observa uma representação do mapa, contendo três veículos de resgate, um ponto de evacuação com 15 turistas e um hospital. Nesta representação, encontram-se os ids dos nós do grafo junto a estes. Também é possível verificar a distancia euclidiana entre dois nós por cima da aresta que os une. Os nós onde se encontravam os véiculos, o ponto de resgate e o hospital estão pintados, maioritariamente, de vermelho, verde e azul, respetivamente.

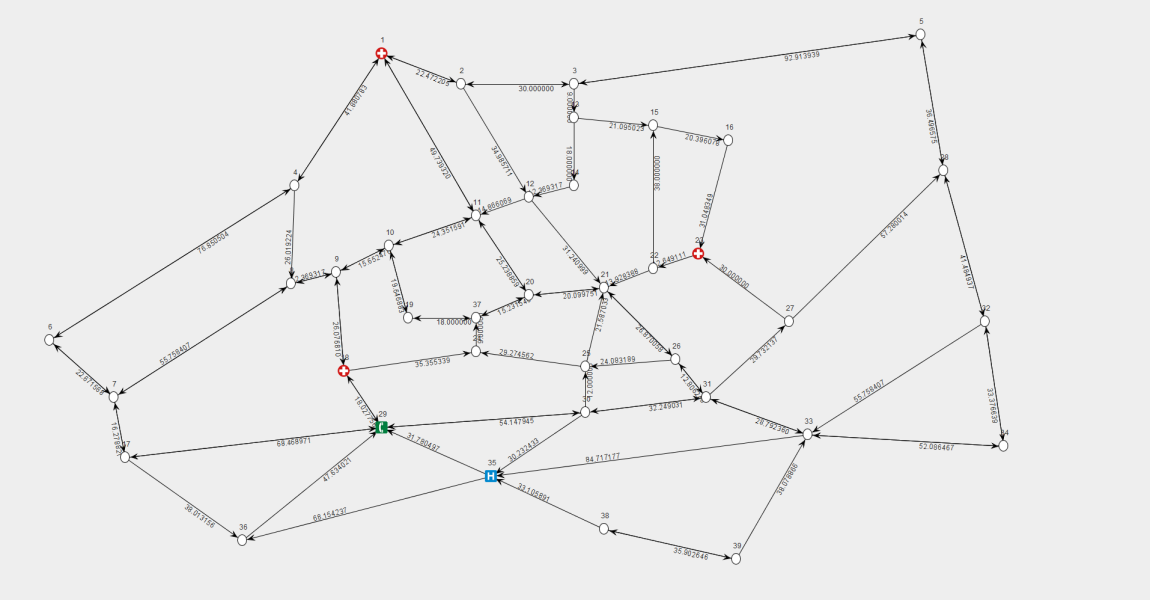

Figura 2: Grafo inicial

Iniciando o programa, este vai verificar qual o véiculo mais apropriado para socorrer, em primeiro lugar, os turistas. Para isso, são efetuados os cálculos explicados anteriormente neste relatório (pág.4) e obtemos os resulados observáveis na Figura 3, sendo o veículo escolhido o que obtiver um menor "Final result".

```
                    ID:1
Distancia:1
Tempo percorrido:0
Capacidade:1
Final result: 0.6

ID:23
Distancia:0.967379
Tempo percorrido:0
Capacidade:0.818182
Final result: 0.565508

ID:18
Distancia:0.182099
Tempo percorrido:0
Capacidade:0.909091
Final result: 0.181958

Veiculo escolhido nr 3 do No 18
```

Figura 3: Dados primeira iteração

Como já seria previsível, neste exemplo, o veículo selecionado na primeira iteração encontra-se no nó com id 18, devido à sua proximidade ao ponto de resgate e de ser dos que consegue levar mais pessoas (5). O trajeto efetuado pelo veículo pode ser observado na Figura 4, verificando que este passa pelo ponto de resgate (nó com id 29) e , de seguida, se dirige para o hospital, através do camninho mais curto.

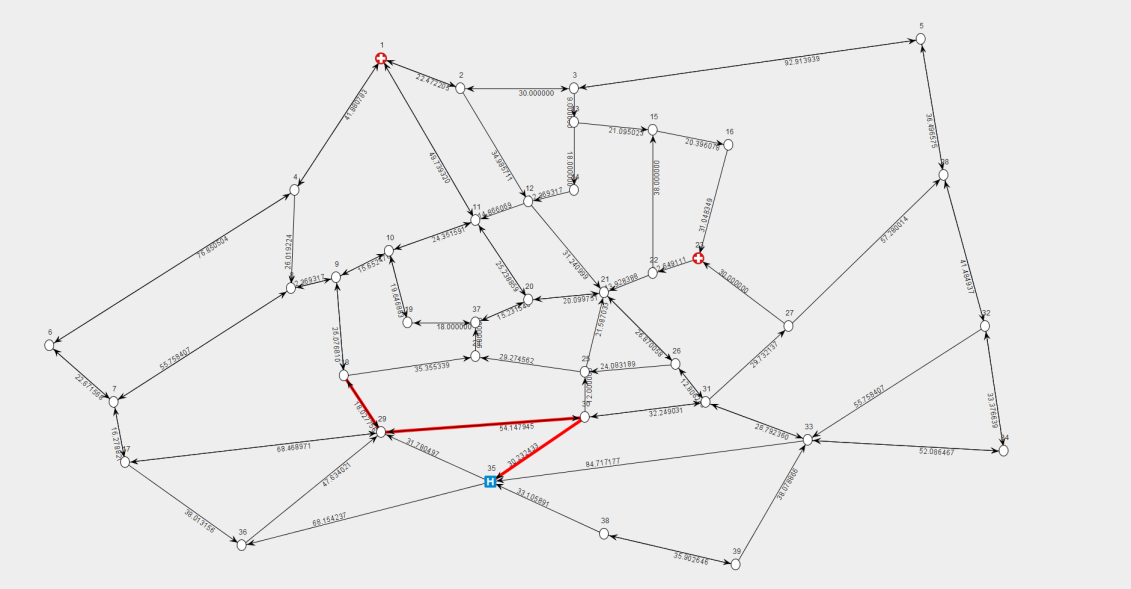

Figura 4: Grafo primeira iteração

Concluída a primeira iteração, é calculada a segunda, obtendo os resultados observados na Figura 5. Nesta iteração, já é tida em conta a distancia percorrida anteriormente pelo veículo, ou seja, apesar do veículo selecionado na primeira iteração, que no fim desta se encontra no hospital, parecer o mais apropriado para ser o selecionado outra vez, verifica-se que o melhor veículo é o que se encontra no nó com id 23. Como todos as ambulâncias do salvamento são chamadas ao mesmo tempo, o segundo veículo chega ao ponto de resgate antes do primeiro passar por lá uma segunda vez.

```
ID:1
Distancia:1
Tempo percorrido:0
Capacidade:1
Final result: 0.6

ID:23
Distancia:0.967379
Tempo percorrido:0
Capacidade:0.666667
Final result: 0.550356

ID:35
Distancia:0.321015
Tempo percorrido:1
Capacidade:0.833333
Final result: 0.643841

Veiculo escolhido nr 2 do No 23
```

Figura 5: Dados segunda iteração

O trajeto efetuado pelo veículo pode ser observado na Figura 6, passando pela segunte sequência de ids de nós: 23, 22, 21, 26, 31, 30, 29, 30 e 35. No fim desta iteração, a ambulância permanece no nó respetivo ao hospital, tal como na iteração anterior, tendo resgatado mais 6 turistas, faltando, desta forma, 4 turistas no ponto de resgate.

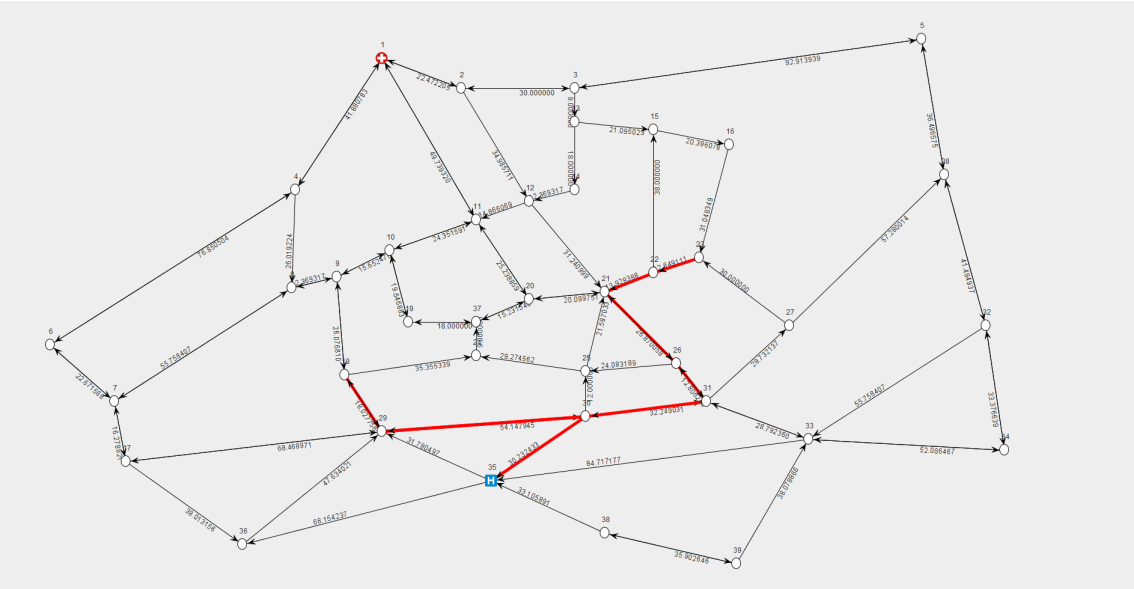

Figura 6: Grafo segunda iteração

Na terceira iteração, através da Figura 7, verifica-se que dois dos três veículos se encontram no hospital e que já têm distância percorrida. Concluí-se que o veículo escolhido desta vez é o véiculo utilizado na primeira iteração, pois apesar de estar à mesma distância do ponto de resgate que o segundo veículo, e de esta ser inferior ao veículo no nó com id 1, tem uma menor distância anterior percorrida.

```
ID:1
Distancia:1
Tempo percorrido:0
Capacidade:0
Final result: 0.5

ID:35
Distancia:0.321015
Tempo percorrido:1
Capacidade:1
Final result: 0.660508

ID:35
Distancia:0.321015
Tempo percorrido:0.432045
Capacidade:0.5
Final result: 0.383326

Veiculo escolhido nr 3 do No 35
```

Figura 7: Dados última iteração

O trajeto efetuado pelo veículo pode ser observado na Figura 8, passando pela segunte sequência de ids de nós: 35, 29, 30 e 35. Assim, concluíem-se as iterações, pois já todos os turistas foram resgatados, visto que a capacidade de este último veículo era superior ao número de pessoas restantes no ponto de evacuação (5 >4).

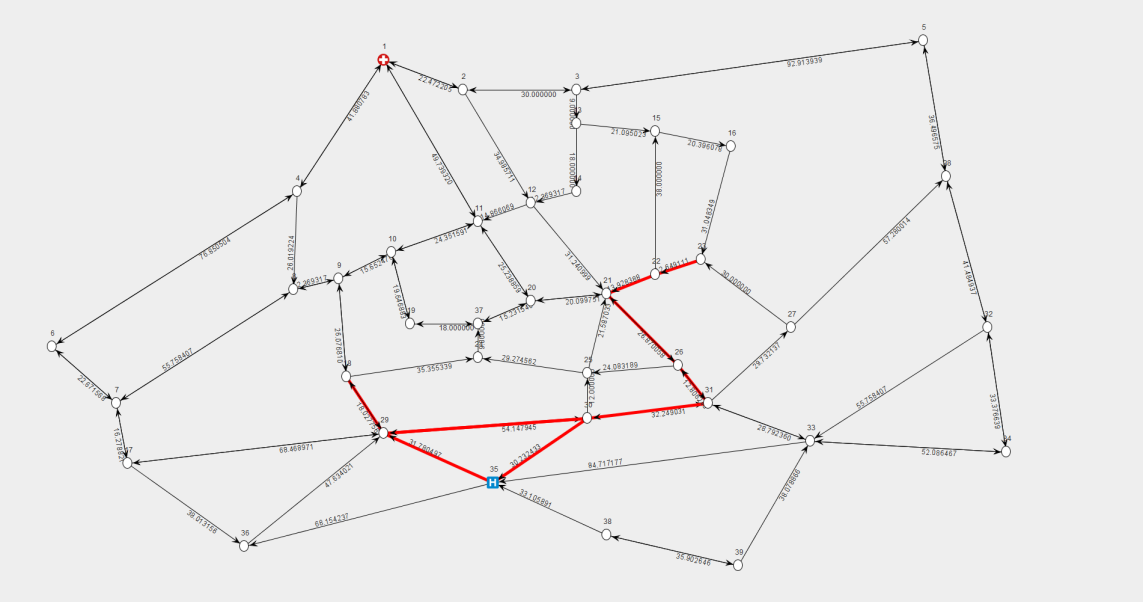

Figura 8: Grafo última iteração

Neste exemplo, foi possível realizar todos os testes acima apresentados, verificando, desta forma, a plausibilidade dos algoritmos e heuríticas propostas.

4.2 Comparações

Para comparação dos algoritmos de escolha realizaram-se alguns testes. Para tal, foi utilizado um mapa com 500 nós para ser possível obter resultados significativos e que possibilitem ser comparados. De forma a avaliar a eficiência do algoritmo efetuado em comparação com outros algoritmos foram realizadas as seguintes experiências:

4.2.1 Experiência 1

Objetivo

Existência de apenas um ponto de resgate em que nenhum veículo tem capacidade capaz de suportar todos os montanhista que necessitam de serem socorridos, e como tal o algoritmo escolheu em iterações distintas, utilziar uma veículo ainda não utilizada ou reutilizar uma já utilizada

Resultados

Tempo A*(ns)	Tempo Dijkstra(ns)
124947000	140575000
109359000	109322000
93696000	124948000
124949000	93697000

Deste modo em média:

Tempo A*(ns)	Tempo Dijkstra(ns)
113237750	117135500

4.2.2 Experiência 2

Objetivo

Experiência de bastantes pontos de resgate estando as pessoas a resgatar dispersas pelos vários pontos. Este teste tem o objetivo de comparar o tempo necessário por cada algoritmo quando executados bastantes vezes. Isto permitirá, de uma forma fácil, analisar a diferença dos algoritmos no que toca a tempo necessário para execução.

Resultados

Tempo A*(ns)	Tempo Dijkstra(ns)
484321000	468694000
421821000	468695000
405256000	468695000
390623000	484374000

Deste modo em média:

Tempo A*(ns)	Tempo Dijkstra(ns)
425505250	472614500

4.2.3 Experiência 3

Objetivo

Experiência de vários pontos de resgate havendo grande concentração de pessoas a resgatar nos vários pontos.

Resultados

Tempo A*(ns)	Tempo Dijkstra(ns)
984329000	1031247000
1015582000	968706000
937453000	999957000
943821000	1015582000

Deste modo em média:

Tempo A*(ns)	Tempo Dijkstra(ns)
970296250	1003873000

4.3 Análise de resultados

Após a realização das várias experiências e analisando os resultados obtidos, é posssível concluir que a utilização do algoritmo A* permite poupar por vezes bastante tempo, não invalidando que pontualmente o tempo necessário para execução por parte do algoritmo dijkstra não possa ser melhor. Ainda que num mapa com apenas 500 nós as variações de tempo não sejam muito expressivas, quando escalado o problema, a utilização do algoritmo A* pode permitir poupar bastante tempo.

5 Conclusão

Com a realização deste projeto, o grupo conclui que, independentemente da heurística usada para tentar obter uma solução do problema, é impossível garantir uma solução ótima em tempo útil, e como tal será sempre necessário restringir à partida algumas das variáveis em causa. No entanto, com a utilização das heurísticas e algoritmos selecionados, é possível obter uma solução plausível de ser aplicada num cenário real. O grupo conclui, assim, que a resolução deste problema, apesar de aparentar ser de pouca complexidade, se trata, de facto, de um problema difícil, uma vez que para encontrar a melhor solução seria necessário analisar uma explosão combinatória de hipóteses. Deste modo, será sempre necessário definir, à partida, alguns parâmetros (neste caso, optou-se por selecionar a ambulância a utilizar em cada iteração).

6 Melhoramentos

Para obter melhores resultados, o grupo pensa que seria possível adaptar a heurística, melhorando-a através da verificação da possibilidade de socorrer montanhistas de diferentes nós na mesma viagem de um veículo.

7 Recursos

Referências

[1] https://paginas.fe.up.pt/~eol/IA/1718/trabalho.html - Site da unidade curricular IART

[2] https://en.wikipedia.org/wiki/A*_search_algorithm - Página referente ao algoritmo

[3] https://www.redblobgames.com/pathfinding/a-star/implementation.html#cplusplus - Página referente ao algoritmo

[4] http://theory.stanford.edu/~amitp/GameProgramming/Heuristics.html#S - Página de heurísticas recomendadas para o algoritmo A*

[5] https://en.wikipedia.org/wiki/Euclidean_distance - Informação sobre a distância Euclidiana

[6] http://www.eclipse.org/downloads/packages/release/Kepler/SR2 - IDE utilizado para realização do projeto

8 Apêndice

8.1 Manual do utilizador

- Numa primeira fase, o utilizador poderá escolher se quer que aplicação use o caso em que existe apenas um ponto de resgate ou o grafo com vários pontos de refúgio.

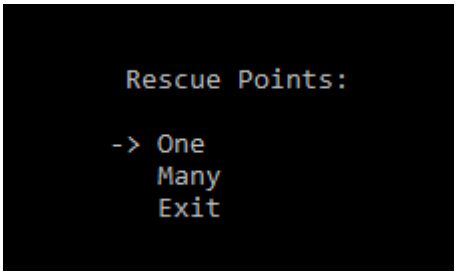

Figura 9: Menu inicial

- Após o utilizador escolher o tipo de situação que pretende simular, é aberto a janela GraphViewer com a representação do mapa.

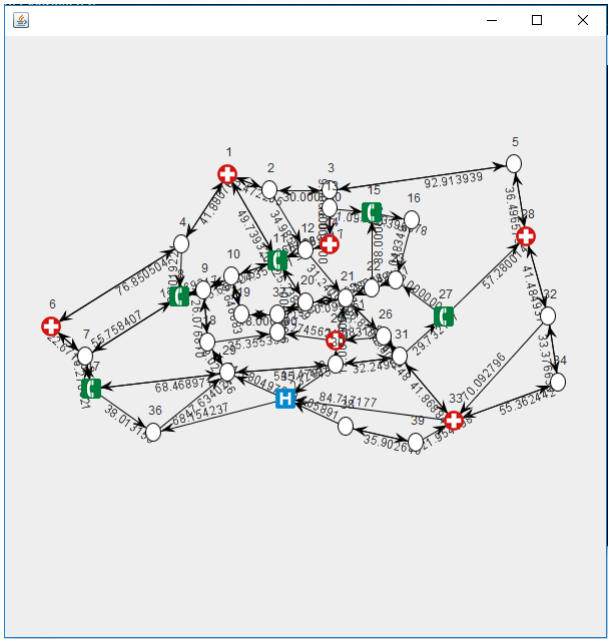

Figura 10: GraphViewer

- O utilizador é apresentado com o segundo menu em que poderá escolher o algoritmo que irá solucionar o problema, Dijsktra ou A*.

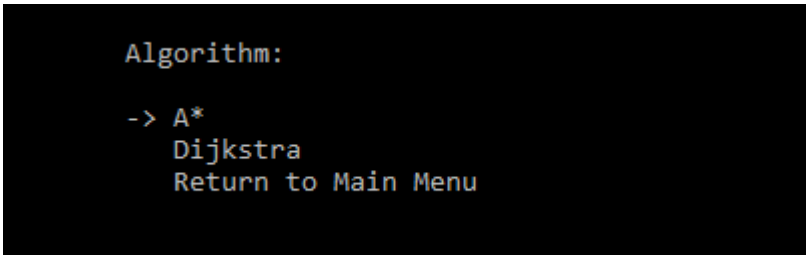

Figura 11: GraphViewer

8.2 Participação dos elementos do grupo

- Carlos Freitas: 33%
- Luís Martins: 33%
- Vicente Espinha: 33%