

UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE – UERN
FACULDADE DE CIÊNCIAS EXATAS E NATURAIS – FANAT
DEPARTAMENTO DE INFORMÁTICA – DI

Roberval Gonçalves Moreira Filho

**HEURÍSTICA PARA DETERMINAÇÃO DE CAMINHO MÍNIMO COM PARADA
INTERMEDIÁRIA, APLICADA AO RESGATE MÉDICO DE URGÊNCIA**

MOSSORÓ - RN

2017

Roberval Gonçalves Moreira Filho

HEURÍSTICA PARA DETERMINAÇÃO DE CAMINHO MÍNIMO COM PARADA INTERMEDIÁRIA, APLICADA AO RESGATE DE MÉDICO DE URGÊNCIA

Monografia apresentada à Universidade do Estado do Rio Grande do Norte como um dos pré-requisitos para obtenção do grau de bacharel em Ciência da Computação, sob orientação da Prof. Dr. Francisco Chagas de Lima Júnior.

MOSSORÓ - RN

2017

Roberval Gonçalves Moreira Filho

Ficha catalográfica gerada pelo Sistema Integrado de Bibliotecas
e Diretoria de Informatização (DINF) - UERN,
com os dados fornecidos pelo(a) autor(a)

F488h Filho, Gonçalves Moreira Filho.
Heurística para determinação de caminho mínimo com parada
intermediária, aplicada ao resgate médico de urgência / Gonçalves
Moreira Filho Filho - 2017.
34 p.

Orientador: Francisco Chagas de Lima Junior.
Coorientadora: .
Monografia (Graduação) - Universidade do Estado do Rio Grande do
Norte, Ciência da computação, 2017.

1. Resgate médico de Urgência. 2. Roteamento de Ambulâncias. 3.
Método Heurístico. I. Chagas de Lima Junior, Francisco, orient. II. Título.

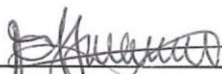
Roberval Gonçalves Moreira Filho

HEURÍSTICA PARA DETERMINAÇÃO DE CAMINHO MÍNIMO COM PARADA INTERMEDIÁRIA, APLICADA AO RESGATE MÉDICO DE URGÊNCIA

Monografia apresentada como pré-requisito para a obtenção do título de Bacharel em Ciência da Computação da Universidade do Estado do Rio Grande do Norte – UERN, submetida à aprovação da banca examinadora composta pelos seguintes membros:

Aprovada em: ___/___/___

Banca Examinadora



Prof. Dr. FRANCISCO CHAGAS DE LIMA JÚNIOR
Universidade do Estado do Rio Grande do Norte - UERN



Prof. Dr. CARLOS HEITOR PEREIRA LIBERALINO
Universidade do Estado do Rio Grande do Norte - UERN



Prof. Me. ANTÔNIO OLIVEIRA FILHO
Universidade do Estado do Rio Grande do Norte - UERN

Aos meus pais que sempre me incentivaram a estudar e fizeram grandes sacrifícios para eu concluir minha graduação.

AGRADECIMENTOS

Gostaria de agradecer primeiramente a Deus, pois sem ele eu não seria nada, aos meus amigos, cujo seus incentivos me motivaram a concluir quando eu mais pensei em desistir, ao meu aos meus professores em especial meu orientador, que forneceram conhecimento para eu crescer como grande profissional dentro da minha jornada acadêmica e aos meus familiares que me apoiaram e me suportaram nos momentos mais difíceis da graduação.

“Bem como o Filho do homem não veio para ser servido, mas para servir, e para dar a sua vida em resgate de muitos.”
Mateus 20:28

RESUMO

Atualmente, no cenário urbano, o número de veículos aumenta a cada dia e o número de acidentes cresce proporcionalmente, desta forma, um sistema de resgate médico de urgência, se faz necessário. É notável que o tempo de resposta do resgate, é diretamente influenciado pela qualidade do trânsito e pela rota utilizada. Neste contexto, este trabalho tem por objetivo desenvolver um método de otimização para minimizar o tempo de resposta no resgate médico de urgência, aplicado ao problema de roteamento de ambulâncias do Serviço de Atendimento Móvel de Urgência – SAMU. Tem como objetivos específicos desenvolver uma tecnologia que utiliza algoritmos heurísticos para oferecer uma melhor rota entre o SAMU e um determinado local de acidente e deste local a Unidade Pronto Atendimento -UPA o mais próximo levando em consideração a situação da rota no momento da ocorrência.

Palavras-chave: Resgate Médico de Urgência, Roteamento de Ambulâncias, Método Heurístico.

ABSTRACT

Currently, in the urban scenario, the number of vehicles increases every day and the number of accidents increases proportionally, so an emergency medical rescue system is necessary. It is notable that the response time of the rescue is directly influenced by the quality of the traffic and the route used. In this context, this work aims to develop an optimization method to minimize the response time in emergency medical rescue, applied to the ambulance routing problem of the Emergency Mobile Care Service - SAMU. Its specific objectives are to develop a technology that uses heuristic algorithms to offer a better route between SAMU and a certain accident site and from this location to the nearest Unit Ready Attendance UPA taking into account the situation of the route at the time of occurrence.

Keywords: Emergency Medical Rescue, Ambulance Routing, Heuristic Method.

LISTA DE SIGLAS

DENATRAN	-	Departamento Nacional de Trânsito
GIS	-	Geographic Information System
GPS	-	Global Positioning System
GSM	-	Global System for Mobile Communication
IPEA	-	Instituto de Pesquisa Econômica Aplicada
PCM	-	Problema do Caminho Mínimo
SAMU	-	Serviço de Atendimento Móvel de Urgência
SDSS	-	Spatial Decision Support System
UPA	-	Unidade Pronto Atendimento
VRP	-	Vehicle Routing Problem

LISTA DE FIGURAS

Figura 1 – Grafo para exemplificar o funcionamento do algoritmo de Dijkstra	20
Figura 2 – Solução para o PCM da figura 1 após aplicado o algoritmo de Dijkstra ...	21
Figura 3 – Pseudocódigo do algoritmo de Dijkstra.....	21
Figura 4 – Pseudocódigo do algoritmo de Floyd-Warshall.....	23
Figura 5 – Grafo para aplicar o algoritmo de Floyd-Warshall.....	23
Figura 6 – Pseudocódigo do método utilizada.....	26
Figura 7 – Cenário de simulação da heurística	26
Figura 8 – Rota do SAMU até o local do acidente	27
Figura 9 – Rota final fornecida pela Heurística	28
Figura 10 – Gráfico dos tempos de execução dado em milissegundos.....	30
Figura 11 – Gráfico com os custos associados as rotas fornecidas.....	30

LISTA DE TABELAS

Tabela 1 – Passo a passo do algoritmo de Dijkstra no grafo da figura 1	20
Tabela 2 – Matriz de custo do grafo da figura 4 com $k=0$	24
Tabela 3 – Matriz de custo do grafo da figura 4 com $k=3$	24
Tabela 4 - Testes executados	29

SUMÁRIO

1 INTRODUÇÃO	13
2 TRABALHOS RELACIONADOS.....	15
3 REFERENCIAL TEÓRICO.....	17
3.1 PROBLEMA DO CAIMNHO MÍNIMO.....	17
3.1.1 FORMULAÇÃO MATEMÁTICA DO PROBLEMA.....	18
3.2 ALGORITMO DE DIJKSTRA.....	19
3.2.1 FUNCIONAMENTO DO ALGORITMO.....	19
3.2.2 EXEMPLO DE FUNCIONAMENTO DO ALGORITMO.....	20
3.3 ALGORITMO DE FLOYD WARSHALL.....	22
3.3.1 FUNCIONAMENTO DO ALGORITMO.....	22
3.3.2 EXEMPLO DE FUNCIONAMENTO DO ALGORITMO.....	23
4 HEURÍSTICA UTILIZADA.....	25
4.1 PROBLEMÁTICA	25
4.2 MÉTODO ADOTADO.....	25
4.3 EXEMPLO DE EXECUÇÃO.....	26
5 RESULTADOS.....	29
6 CONSIDERAÇÕES FINAIS E TRABALHOS FUTUROS.....	32
REFERÊNCIAS	33

1 INTRODUÇÃO

Segundo o Departamento Nacional de Trânsito o Brasil tem uma frota de 87.364.144 veículos e a cidade de Mossoró - RN tem 127.368 veículos (dados do DENATRAN, fevereiro de 2015). Esta realidade nacional é consequência do desenvolvimento econômico, da oferta de crédito para aquisição de um meio de transporte particular, do crescimento populacional e da má qualidade do transporte público. Tais aspectos constituem fatores decisórios para o acúmulo de veículos nos centros urbanos. Acrescente ainda a este cenário uma malha viária mal dimensionada, pouca fiscalização, uma população de má educação no trânsito, e se obtém um cenário de caos.

De acordo com o Instituto de Pesquisa Econômica Aplicada (IPEA, 2011), a média de pessoas que enfrentam congestionamento mais de uma vez por dia no Brasil é de 20,5%, chegando na região Sudeste a 21,6% e na região Sul a 21,9%. Os constantes congestionamentos nas grandes cidades são responsáveis pelo alto grau de estresse dos motoristas, por um grande número de acidentes, pelo um elevado nível de poluição e por significativos prejuízos econômicos causados pelo alto consumo de combustível e pelo desperdício de tempo.

A aplicação de recursos tecnológicos para que o trânsito flua de forma rápida e segura tem sido atualmente motivo de investimento para os gestores públicos. Neste sentido, além de vários tipos de dispositivos eletrônicos (semáforos inteligentes, fiscalização eletrônica e sensores de velocidade, etc.) diversos softwares e algoritmos têm sido desenvolvidos, tais como sistema Siri (VILANOVA, 2004), e o Atefi (PIAI e CERVANTES, 2009), sendo, entretanto, estes softwares de alto custo para aquisição e que necessitam de treinamento especializado para utilização.

O Serviço de Atendimento Móvel de Urgência - SAMU tem como objetivo responder da melhor forma possível a toda situação de urgência que necessite de meios médicos, desde o primeiro contato telefônico até a liberação das vítimas ou seus encaminhamentos hospitalares. O sistema de recepção de demandas deve determinar e desencadear a resposta mais adequada para cada caso, assegurar a disponibilidade dos meios hospitalares, determinar o tipo de transporte exigido e preparar o acolhimento dos pacientes (TAKEDA et al., 2001)]. A função básica de um SAMU é responder de forma organizada, a fim de evitar o uso excessivo de recursos, a toda situação de urgência que necessite de meios médicos, desde o primeiro contato

telefônico até a liberação das vítimas ou seus encaminhamentos hospitalares. O sistema deve determinar e desencadear a resposta mais adequada para o caso, assegurar a disponibilidade dos meios hospitalares, determinar o tipo de transporte exigido e preparar o acolhimento dos pacientes (TAKEDA et al., 2001). Atualmente, na cidade de Mossoró, não existem condições ideais para que o SAMU consiga atender de forma satisfatória o seu propósito, as ruas não são adequadamente planejadas e o tráfego de veículos está em ascensão causando muito congestionamento.

Desta forma, o objetivo geral desse trabalho é desenvolver um método de otimização para minimizar o tempo de resposta no resgate médico de urgência, aplicado ao problema de roteamento de ambulâncias do SAMU da cidade de Mossoró. Tem como objetivos específicos reduzir o tempo de resposta no resgate médico de urgência, utilizar métodos heurísticos no auxílio a tomada de decisões e resolver o problema do caminho mínimo com parada intermediária e multiplicidade de localizações finais.

Os demais capítulos deste trabalho estão organizados como segue: o capítulo 2 apresenta o referencial teórico relativo ao problema do caminho mínimo, seu modelo formal e principais algoritmos, o capítulo 3 mostra qual o problema a ser resolvido bem como qual a solução que foi adotada e um exemplo que como aplica-la, no capítulo 4 temos uma explanação de alguns dos trabalhos dos últimos anos relacionados a otimização no resgate médico de urgência, o capítulo 5 indica os resultados da pesquisa enquanto no capítulo 6 são expostas as conclusões e possibilidades para trabalhos futuros.

2 TRABALHOS RELACIONADOS

Diversos trabalhos tiveram grandes contribuições para o roteamento de ambulâncias e de transportes em geral. No capítulo que se segue, teremos uma visão sobre alguns estudos que foram realizados nos últimos anos na área de VRP.

O trabalho de (DEREKENARIS et. al., 2001) trata do VRP usando uma integração entre o sistema de informação geográfica (do inglês *geographic information system* – GIS), o sistema de posicionamento global (do inglês *global positioning system* – GPS) e o sistema global para tecnologias de comunicação (do inglês *global system for mobile communication* - GSM) fornecendo com essas tecnologias uma maneira de prover uma rota com base em informações de tráfego em tempo real.

De semelhante modo, o trabalho de (PANAHI; DELAVAR, 2008) fornece um sistema espacial de apoio a decisões (do inglês *spatial decision support system* – SDSS) aplicado ao roteamento de ambulâncias usando GIS, o estudo realizou um comparativo entre os algoritmos dinâmicos de caminho mais curto e foi baseado no algoritmo de Dijkstra.

Outro trabalho muito interessante dos últimos anos na área de roteamento de ambulâncias é o trabalho feito por (JOTSHI; GONG; BATTÀ, 2009) que utiliza uma central de dados que coleta dados dos mais diversos meios como câmeras ou satélites para definir qual rota a ser tomada e seu diferencial está em selecionar qual ambulância será enviada ao local do acidente no caso de ter sido realizado um pre-diagnostico sobre o paciente.

(ELALOUF, 2012) propõe um algoritmo pseudo-polinomial, e um algoritmo de aproximação, ϵ -*approximation*, para encontrar o caminho mais curto dado um conjunto de arestas que possuem um valor aleatório que contém o tempo esperado para percorrer o caminho e a variância desse tempo.

No trabalho de (KRITZINGER et al., 2012) ele descreve uma avaliação experimental de um algoritmo de roteamento de veículos dependentes do tempo usando informações de tráfego em tempo real. As informações de tráfego são fornecidas por uma matriz de distância dependente do tempo, calculada pelo algoritmo de Dijkstra.

Com outras técnicas (MUSOLINO et al., 2013) nos apresenta um *framework* que projeta rotas ótimas para veículos de emergência que leva em conta as variações do tempo de viagem nas rodovias. Basicamente o *framework* trabalha com dois componentes: Um modelo de atribuição dinâmica, que simula a movimentação nas rodovias durante o dia e um modelo dinâmico de roteamento de veículos que projeta rotas ótimas.

3 REFERENCIAL TEÓRICO

Neste capítulo iremos abordar o modelo do problema bem como os algoritmos que foram utilizados na heurística desenvolvida.

3.1 PROBLEMA DO CAMINHO MÍNIMO

O problema do caminho mínimo (PCM) é um problema comum em diversas áreas como por exemplo em roteamento de veículos (Alvarenga, 2005), planejamento de caminho para robôs (Mathew et al., 2015), roteamento de pacotes IP (Kurose, 2013), etc. O PCM pode ser modelado em grafo e nada mais é do que determinar a melhor rota entre dois pontos.

Para determinar a melhor rota, muitas condições podem ser levadas em consideração como o tempo de percurso, a distância, o gasto associado ao caminho, etc. Chamamos estes agentes, genericamente, de custo. Muitos algoritmos podem ser utilizados para resolver o PCM como o algoritmo de Bellman-Ford, Algoritmo de Johnson, algoritmo de Floyd-Warshall, algoritmo de Dijkstra e o algoritmo A*.

Dreyfus (1969) apresenta 5 tipos distintos de PCM: Determinar o menor caminho entre dois nós específicos em uma rede; determinar o menor caminho entre todos os pares de vértices em uma rede; determinar o segundo, terceiro, etc. Menor caminho; determinar o percurso mais rápido através de uma rede com tempos de viagem dependendo da hora de partida; determinar o menor caminho entre dois pontos específicos onde é necessário a passagem por um nó intermediário.

Embora os problemas de caminho mínimo sejam relativamente fáceis de resolver, o projeto e a análise dos algoritmos mais eficientes para resolvê-los requerem considerável engenho (Ahuja et al., 1993). Neste trabalho iremos abordar os algoritmos de Dijkstra e o algoritmo de Floyd-Warshall para resolver o PCM entre dois pontos específicos com um nó intermediário entre eles e a possibilidade de mais de um nó final.

3.1.1 Formulação matemática do problema

Seja um grafo orientado $G1 = (V, E)$ que possui um conjunto de vértices V com n vértices e um conjunto de arestas E com m arestas. Cada aresta é representada por um par ordenado de vértices (i, j) e tem um custo C_{ij} , associado a cada aresta $(i, j) \in E$.

Em uma aresta $(i, j) \in E$, i é o antecessor de j e denomina-se como $p(j)$. Duas arestas são adjacentes se existe entre eles pelo menos um vértice em comum. A lista de adjacências de vértices $E(i)$ é o conjunto de vértices adjacentes ao vértice i , tal que $E(i) = \{j \in V : (i, j) \in E\}$.

O custo de um caminho orientado é a soma do custo de cada arco que pertence ao caminho. Méndez e Guardia (2008) propõem que o problema do caminho mínimo pode-se formular matematicamente como um problema de programação linear, dado a seguir:

Minimizar:

$$\sum_{(i,j) \in E} C_{ij} X_{ij}$$

Sujeito a:

$$\sum_{\{j:(i,j) \in E\}} x_{ij} - \sum_{\{j:(i,j) \in E\}} x_{ji} = \text{Fluxo na rede gerado pelo vértice } j$$

$$x \geq 0, \text{ para todo } (i, j) \in E$$

Para o estudo do problema do caminho mais curto neste trabalho, assume-se que:

- Todos os pesos dos arcos são inteiros.
- A rede contém um caminho orientado desde o nó fonte até os outros nós da rede.
- A rede não contém um ciclo com pesos negativo.

3.2 ALGORITMO DE DIJKSTRA

O algoritmo de Dijkstra teve sua primeira publicação no ano 1959 e foi criado por um matemático holandês chamado Edsger Dijkstra (Pares, 2016). O algoritmo resolve o problema do caminho mínimo fornecendo o menor custo entre o vértice A e o vértice B em um grafo ponderado $G = (V, E)$ com valores não nulo.

Geralmente o grafo G representa uma situação real e o algoritmo de Dijkstra permite a construção de uma rota ótima entre dois pontos quaisquer. Segundo Cormen (2009) com uma boa implementação, o tempo de execução do algoritmo de Dijkstra é menor que o do algoritmo de Bellman-Ford.

3.2.1 Funcionamento do algoritmo

Dado um grafo ponderado $G = (V, E)$ com valores não nulos, o algoritmo mantém um conjunto S de vértices cujos custos finais de trajetória mais curta da fonte S já foram selecionados, esses vértices são rotulados como permanente. Os outros vértices são rotulados como temporários.

Os vértices rotulados como permanente são aqueles cujo custo associado, representa o menor custo até o vértice. Aqueles rotulados como temporários, tem seu custo associado com o limite máximo de custo do caminho mínimo até o vértice. As regras de funcionamento podem ser analisadas a seguir:

Passo 1: Rotular como permanente o vértice S com custo 0 (já que não há custo para o vértice inicial para ir a ele mesmo) e os demais vértices com custo ∞ ;

Passo 2: Todos os vértices J ainda não rotulados adjacente a S devem receber uma nova rotulação temporária que será $i + C_{ij}$ onde i é o vértice rotulado como permanente no passo anterior, caso $i + C_{ij}$ seja superior ao novo valor, o custo antigo permanece;

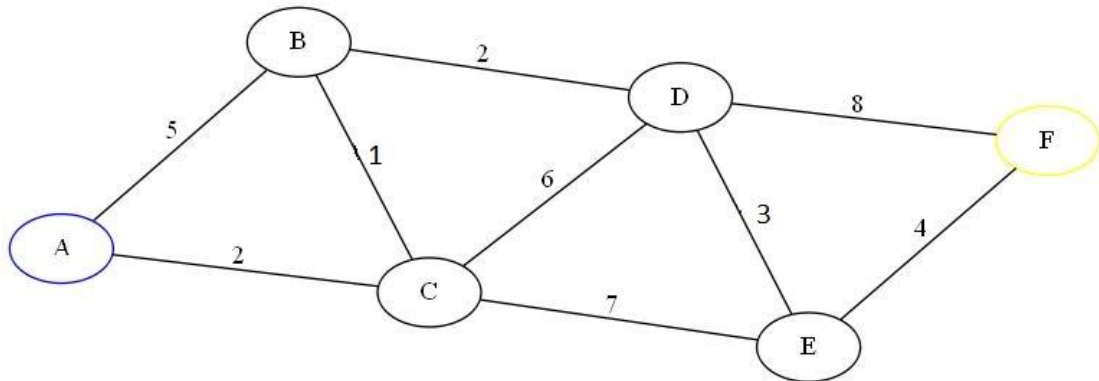
Passo 3: Selecionar e rotular como permanente o vértice J associado ao menor valor de custo encontrado da seção anterior;

Passo 4: Repete os passos anteriores até que o vértice terminal T seja rotulado como permanente ou que todos os vértices estejam rotulados como permanente.

3.2.2 Exemplo de funcionamento do algoritmo

Para exemplificar o funcionamento do algoritmo de Dijkstra, utilizaremos o grafo representado pela figura 1, a tabela 1 mostra o passo a passo para resolver o problema do caminho mínimo do vértice A ao vértice F.

Figura 1 – Grafo para exemplificar o funcionamento do algoritmo de Dijkstra



Fonte: Autoria própria

A figura 1 nos mostra um grafo ponderado que contém apenas valores não nulos associados as arestas, assim, o grafo apresentado corresponde aos pré-requisitos para que o algoritmo de Dijkstra possa ser executado corretamente.

A tabela a seguir nos apresenta seis colunas referentes aso vértices do grafo e uma sétima que nos mostra qual a ação realizada para se chegar ao determinado resultado:

Tabela 1 – Passo a passo do algoritmo de Dijkstra aplicado no grafo da figura 1

A	B	C	D	E	F	Ação
0	∞	∞	∞	∞	∞	RP – A
0	(5, A)	(2, A)	∞	∞	∞	Atualiza adjacências a i RP - C
0	(3, C)	(2, A)	(8, C)	(9, C)	∞	Atualiza adjacências a i, RP - B
0	(3, C)	(2, A)	(5, B)	(9, C)	∞	Atualiza adjacências a i, RP - D
0	(3, C)	(2, A)	(5, B)	(8, D)	(13, D)	Atualiza adjacências a i, RP - E
0	(3, C)	(2, A)	(5, B)	(8, D)	(12, E)	Atualiza adjacências a i, RP - F - Parar

Fonte: Autoria própria

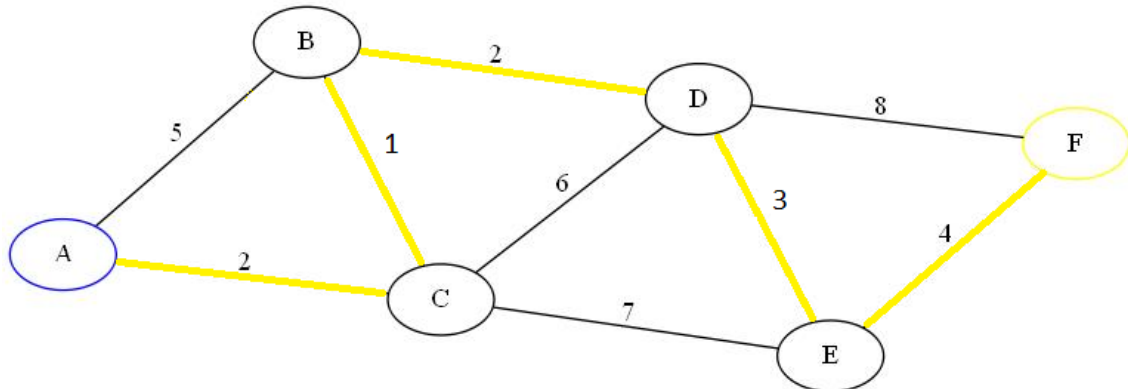
Onde:

RP: Rotula como permanente;

I: O vértice marcado como permanente no passo anterior.

O caminho mínimo encontrado pelo algoritmo é o vetor S contendo cada vértice que foi marcado como permanente, assim, a solução para o PCM da figura 1 é o caminho que pode ser visualizado na figura 2 e o seu custo é 12 que é o valor associado ao vértice F ao ser marcado como permanente.

Figura 2 – Solução para o PCM da figura 1 após aplicado o algoritmo de Dijkstra



Fonte – Autoria própria

A rota apresentada pela figura 2 é, partindo de A, C, B, D, E e por fim F. O pseudocódigo para o algoritmo de Dijkstra pode ser visualizado na figura 3 e os códigos usados neste trabalho incluindo o de Dijkstra são apresentados nos apêndices.

Figura 3 – Pseudocódigo do algoritmo de Dijkstra

```

começar
S := ∅
S̄ := N
d(i) := ∞ para cada nó i ∈ N
d(s) := 0
pred(s) := 0
se |S| < n fazer
começar
  Seja i ∈ S̄ um nó tal que d(i) = min{d(j) : j ∈ S̄}
  S := S ∪ {i}
  S̄ := S̄ - {i}
  para cada (i, j) ∈ A(i) fazer
    se d(j) > d(i) + cij então d(j) := d(i) + cij e pred(j) := i
  fim
fim
fim

```

Fonte: Mendéz (2008)

3.3 ALGORITMO DE FLOYD-WARSHALL

O algoritmo de Floyd-Warshall foi desenvolvido na forma como o conhecemos hoje por Robert Floyd (1962), mas anteriormente ele foi publicado em outra forma por Bernard Roy (1959) e por Stephen Warshall (1962). O algoritmo de Floyd Warshall tem a capacidade de devolver uma matriz contendo o custo do melhor caminho entre todos os vértices de um grafo $G = (V, E)$.

O algoritmo de Floyd-Warshall aceita custo negativo nas arestas, mas ele não aceita ciclos de peso negativo e é um algoritmo que usa programação dinâmica para encontrar o menor caminho entre os vértices.

3.3.1 Funcionamento do algoritmo

Para resolver o PCM entre todos os vértices, o algoritmo de Floyd-Warshall utiliza o conceito de vértice intermediário. O vértice intermediário em um caminho $S = \{v_1, v_2, \dots, v_f\}$ é qualquer vértice que não seja v_1 (vértice inicial) ou v_f (vértice final), ou seja, os vértices do conjunto $S = \{v_2, v_3, \dots, v_{f-1}\}$.

Seja o conjunto $V = \{1, 2, \dots, n\}$ todos os vértices do grafo $G = (V, E)$, considere o subconjunto $\{1, 2, \dots, k\}$ de vértices do conjunto V . O algoritmo de Floyd-Warshall busca o caminho mínimo P do vértice i ao vértice j onde todos os vértices intermediários k entre eles estão no subconjunto $\{1, 2, \dots, k-1\}$.

É importante lembrar que o caminho mínimo entre i e j não possui um mesmo vértice mais de uma vez e que o resultado depende se k é intermediário de P ou não onde:

Se k não é intermediário a P , então $C_{ij}(k) = C_{ij}^{(k-1)}$.

Se k é intermediário a P , então $C_{ij}(k) = \text{Min}[C_{ik}^{(k-1)} + C_{kj}^{(k-1)}]$.

Com base no que foi apresentado podemos definir a matriz de custo mínimo que o algoritmo retorna como:

$$C_{ij}(k) = \begin{cases} C_{ij} & \text{Se } k=0, \\ \text{Min}(C_{ij}^{(k-1)}, C_{ik}^{(k-1)} + C_{kj}^{(k-1)}) & \text{Se } k \geq 1 \end{cases}$$

Podemos analisar o pseudocódigo na figura 4:

Figura 4 – Pseudocódigo do algoritmo de Floyd-Warshall

```

FLOYD-WARSHALL( $W$ )
1   $n = W.rows$ 
2   $D^{(0)} = W$ 
3  for  $k = 1$  to  $n$ 
4      let  $D^{(k)} = (d_{ij}^{(k)})$  be a new  $n \times n$  matrix
5      for  $i = 1$  to  $n$ 
6          for  $j = 1$  to  $n$ 
7               $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$ 
8  return  $D^{(n)}$ 

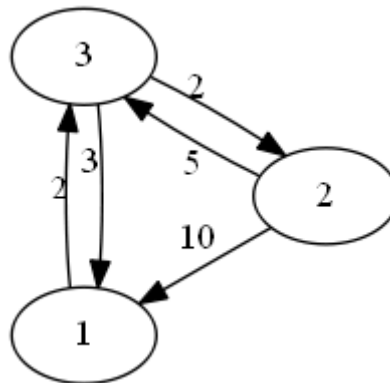
```

Fonte: Cormen (2009)

3.3.2 Exemplo de funcionamento do algoritmo

Consideremos a figura 4 e a tabela 2 para exemplificar o funcionamento do algoritmo:

Figura 5: Grafo para aplicar o algoritmo de Floyd-Warshall



Fonte: Autoria própria, elaborado com o software Graphviz.

A figura 4 contém um grafo direcionado e ponderado que não contém círculos com peso negativo de modo que é possível aplicar o algoritmo e obter resultados.

Tabela 2: Matriz de custo do grafo da figura 4 com $k=0$

		J		
		1	2	3
i	1	0	∞	2
	2	10	0	5
	3	3	2	0

Fonte: Autoria própria, elaborado com Graphviz

A tabela 2 relacionada a matriz de custo do grafo da figura 4, contendo o custo de cada aresta. Como podemos verificar, caso $i = j$ o custo se torna 0, onde não existe caminho o custo é ∞ . Depois da primeira da segunda e da terceira iteração onde k recebe o valor 1 e 2 a matriz não sofre alteração, a tabela 3 mostra como ela fica depois que k recebe 3:

Tabela 3: Matriz de custo do grafo da figura 4 com $k=3$

		J		
		1	2	3
i	1	0	4	2
	2	8	0	5
	3	3	2	0

Fonte: Autoria própria, elaborado com Graphviz

Como podemos observar a matriz sofreu duas alterações, onde o algoritmo encontrou uma rota de 1 para 2 passando pelo vértice 3, e melhorou a rota de 2 para 1 fornecendo uma rota com custo 8 que embora tenha mais vértices no caminho, possui um custo menor.

4 HEURÍSTICA UTILIZADA

Neste capítulo iremos abordar o problema bem como o método que foi proposto para solucioná-lo.

4.1 PROBLEMÁTICA

A situação trata basicamente do problema de roteamento de veículos (do inglês: *Vehicle Routing Problem* - VRP) aplicado aos veículos do SAMU para que eles cheguem a um determinado local de acidente no menor tempo possível e depois disso conduzir o(s) paciente(s) para a UPA mais próxima dado a situação apática da cidade, como o número limitado de postos do SAMU, transito caótico e ruas que não possuem um planejamento adequado.

O problema se trata de resolver o PCM com um vértice intermediário entre o vértice inicial e o vértice final, além disso o problema envolve outra variável que é a possibilidade de existência de vários vértices finais devido a imprevisibilidade da localização do acidente não podemos definir qual será o local em que o paciente deverá ser atendido.

4.2 MÉTODO ADOTADO

Para resolver o problema foi desenvolvida uma estratégia usando o algoritmo de Dijkstra e o algoritmo de Floyd-Warshall. A heurística funciona com bases nos seguintes passos:

1° - Dijkstra é executado para encontrar a melhor rota entre o SAMU até o local do acidente;

2° - A partir do local do acidente é necessário determinar qual a UPA mais próxima, para isso executamos Floyd-Warshall para encontrar o custo entre o local do acidente e as UPAs;

3° - Seleciona a UPA com o menor custo;

4° - Executa-se o algoritmo de Dijkstra mais uma vez para encontrar a rota até a UPA anteriormente selecionada.

A pseudocódigo para o algoritmo pode ser visualizado na figura 6:

Figura 6 - Pseudocódigo do método adotado

Algoritmo 1: FDR**Entrada:** Vértice Intermediário**Saída:** Solução S

```

1 início
2   DIJKSTRA(V-inicial, V-intermediario, Matriz-Distancia)
3   FLOYD(Matriz-Distancia) /* Encontra custos de V-finais */
4   SELECIONE menor V-final
5   DIJKSTRA(V-intermediario, V-final, Matriz-Distancia)
6 fim
7 retorna  $S$ 

```

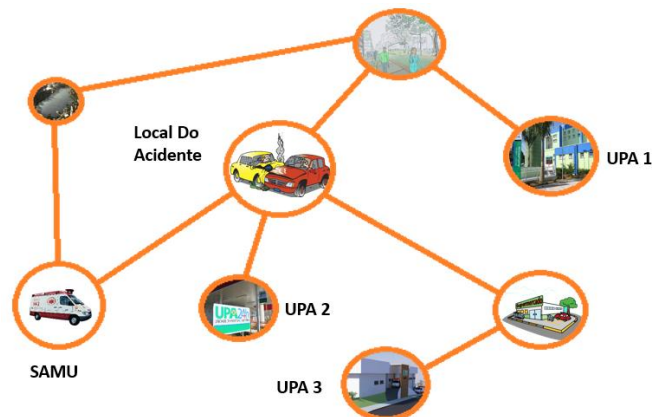
Fonte: Autoria própria

A figura 6 nos mostra de forma genérica o funcionamento do algoritmo desenvolvido onde V-inicial é o vértice inicial que representa a localização inicial da ambulância do SAMU, V-intermediário é o vértice de parada intermediária, ele representa o local da ocorrência de emergência e o V-final é o vértice de parada que representa a UPA para a qual aquele que sofreu a ocorrência deve ser encaminhado.

4.3 EXEMPLO DE EXECUÇÃO

Para exemplificar o funcionamento da heurística, foi criado um cenário que pode ser visualizado na figura 7, onde será realizada a execução dos procedimentos.

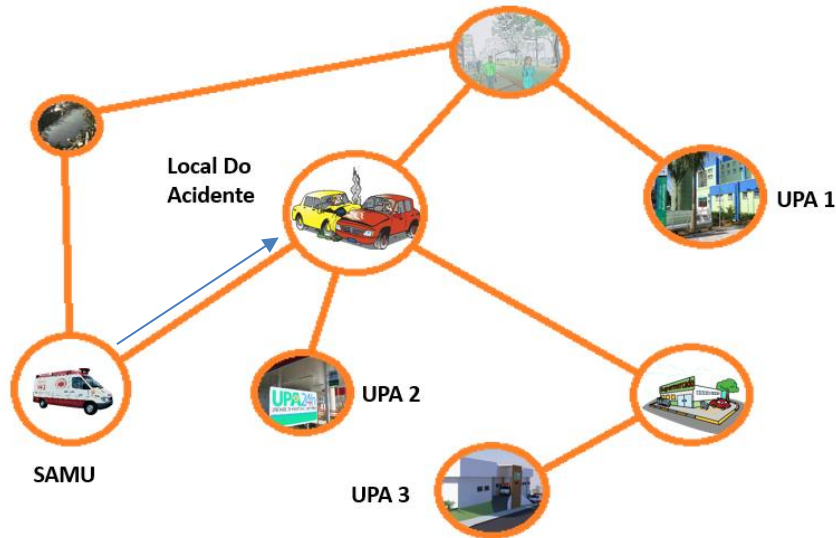
Figura 7 – Cenário de simulação da heurística



Fonte: Autoria própria com imagens retiradas do Google

A figura 7 nos apresenta um grafo cujo os vértices representam pontos de uma cidade, na imagem temos uma ambulância do SAMU, o local onde aconteceu um acidente e três UPAs. O primeiro passo para resolver o problema é a execução do algoritmo de Dijkstra para encontrar a melhor rota entre o SAMU e o local do acidente. A figura 8 apresenta a rota fornecida.

Figura 8 – Rota do SAMU até o local do acidente



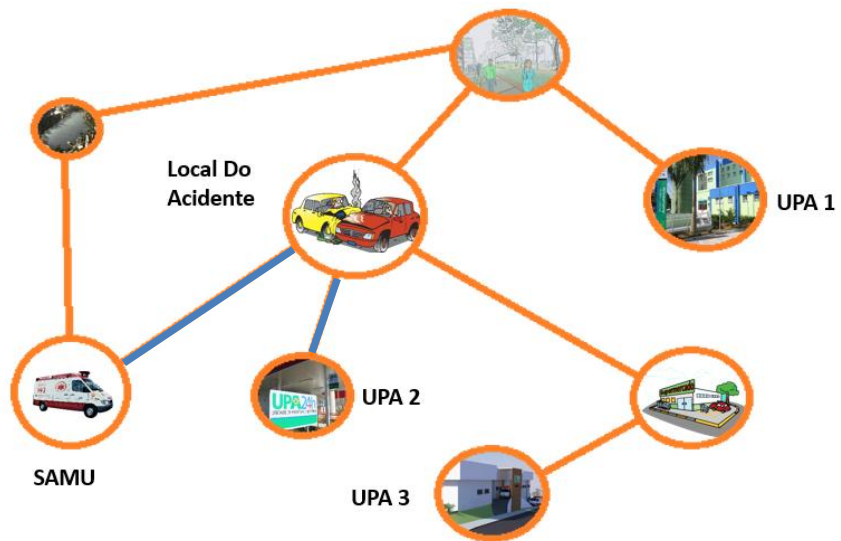
Fonte: Autoria própria com imagens retiradas do Google

A figura 8 indica a rota que o algoritmo de Dijkstra encontrou. Na sequência, executamos o algoritmo de Floyd-Warshall para encontrar a distância do local do acidente para as UPAs e logo em seguida comparamos os resultados para encontrar a UPA mais próxima.

Quando executamos o algoritmo de Floyd-Warshall ele retorna uma matriz que contém o custo entre todos os vértices do grafo (matriz de custo), essa matriz nos apresenta os custos até as UPAs: X para a UPA 1, Y para a UPA 2 e Z para a UPA 3 onde $Y < X < Z$, logo, a UPA que foi selecionada foi a 2.

Para finalizar a execução, Dijkstra é executado mais uma vez para encontrar a rota entre o local do acidente até a UPA que foi selecionada no passo anterior, este passo é necessário, pois o algoritmo de Floyd-Warshall não retorna o caminho até os vértices. A figura 9 indica a rota final que a heurística fornece.

Figura 9: Rota final fornecida pela Heurística



Fonte: Autoria própria com imagens retiradas do Google

5 RESULTADOS

Os testes foram realizados com dois grafos distintos, um contendo dez vértices e outro com quarenta. Foram realizadas dez execuções em cada instancia. Os resultados do experimento podem ser visualizados na tabela 4 onde o grafo A é o grafo de dez vértices contendo um ponto do SAMU e três possíveis UPAs. O grafo B representa o grafo com 40 vértices e possui cinco possíveis UPAs. O tempo de execução é dado em milissegundos.

Tabela 4 – Testes executados

	Tempo de Execução		Custo associado	
	Grafo A	Grafo B	Grafo A	Grafo B
Execução 01	9	47	668	1815
Execução 02	10	54	668	805
Execução 03	11	42	503	995
Execução 04	9	52	350	995
Execução 05	9	51	849	1640
Execução 06	9	53	728	510
Execução 07	9	45	875	1285
Execução 08	8	61	350	880
Execução 09	7	47	518	2145
Execução 10	10	52	518	1540
Tempo Máximo	12	61		
Tempo Mínimo	8	42		
Média	9.1	50.4		

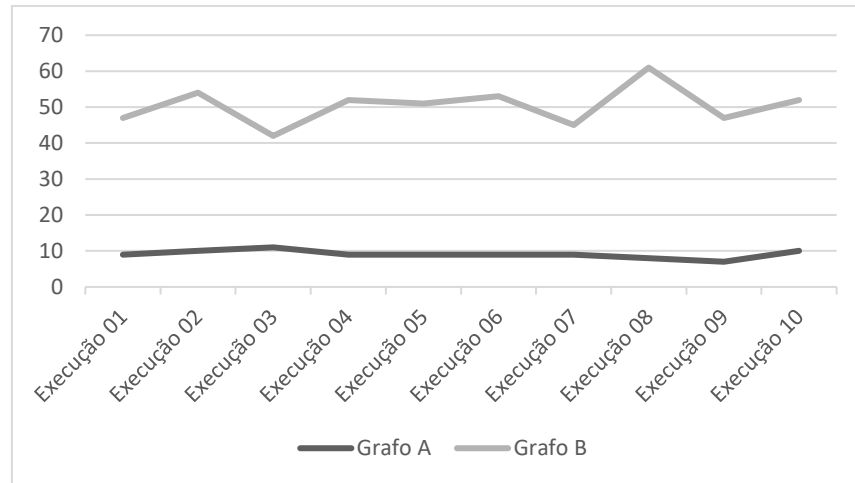
Fonte: Autoria própria

Podemos observar na tabela 4 que os valores dos testes quando executados no grafo com A, os valores são próximos, isso que dá devido à pequena quantidade de possibilidades, diferente de quando executados no grafo B que possui maior variância de resultados.

Em cada teste foi selecionado um local de acidente diferente e as UPAs nas duas instâncias foi alterado, os testes foram realizados em um *notebook* Dell com 6GB

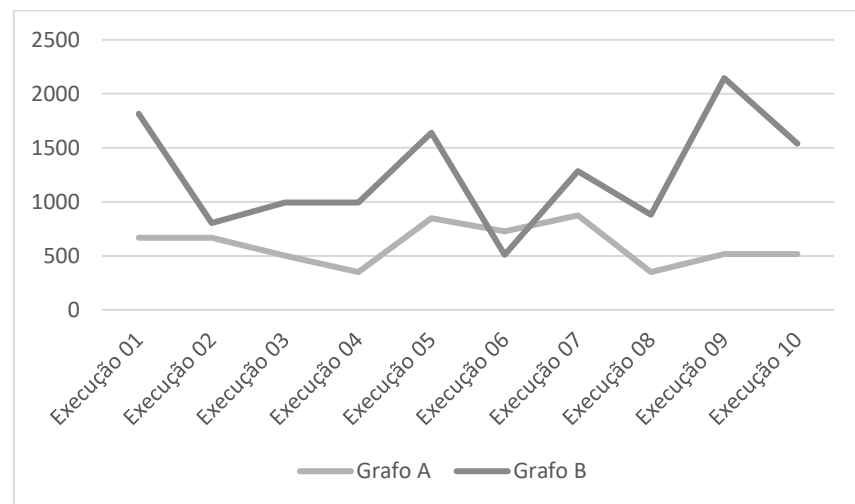
de memória RAM e com um processador Intel Core I5 com sistema operacional *Windows 10 home*.

Figura 10 -Gráfico dos tempos de execução dado em milissegundos



Fonte: Autoria própria

Figura 11 – Gráfico com os custos associados as rotas fornecidas



Fonte: Autoria própria

Nas figuras 9 e 10 podemos ver a representação gráfica dos resultados de teste, nos gráficos podemos observar que no grafo A que possui um número pequeno de rotas, possui pouca variação enquanto no grafo B os resultados obtiveram grande variação, principalmente em relação aos custos associados as rotas.

O *software* foi desenvolvido na linguagem de programação java e possui como entrada o local do acidente e como saída as rotas e custo para cada percurso (do SAMU até o local do acidente e do local do acidente para a UPA mais próxima) além do tempo de execução em milissegundos.

6 CONCLUSÃO E TRABALHOS FUTUROS

Embora os algoritmos estudados, a princípio, não fossem adequados a solucionar o problema no formato apresentado, a união de suas funcionalidades atendeu as expectativas e foi capaz de resolver o dilema da existência de um vértice intermediário entre o vértice inicial e os possíveis vértices finais. O algoritmo pode ser usado para salvar a vida das pessoas sabendo que em um resgate de urgência alguns segundos podem fazer a diferença entre a vida e a morte da vítima.

Como trabalhos futuros temos as seguintes possibilidades:

- Aplicar o algoritmo ao cenário da cidade de Mossoró ou alguma outra que tenha um SAMU;
- Utilizar um método de aquisição de tráfego em tempo real e alimentar o grafo com essas informações para fornecer uma rota melhor aos veículos do SAMU;
- Desenvolver a heurística em outras linguagens de programação para verificar qual oferece melhores tempos de resposta;
- Aplicar ao algoritmo, rotulamento de vértices para fornecer um sistema onde as ambulâncias possam enviar o paciente a um local de atendimento especializado no caso em que se encontra o paciente.

REFERÊNCIAS

- IPEA. Rapidez e custo influenciam na escolha do transporte. Instituto de pesquisas econômicas aplicadas – 2011. Disponível em:
<<http://www.ipea.gov.br/portal/index.php>> Acesso em: 6 abril 2015.
- VILANOVA, L. (2005), "SIRI - Um novo simulador para redes de semáforos.", Disponível em:
<http://meusite.mackenzie.com.br/professor_cucci/texto29.pdf>. Acesso em: 06 Abril de 2015.
- PIAI, J. C., CERVANTES, S. G. (2010), "Um modelo para tráfego urbano e suas otimizações." In: Congresso Brasileiro de Automática, XVIII. 12 a 16 set 2010, Bonito-MS.
- TAKEDA, R. A.; WIDMER, J. A.; MORABITO, R. (2011), "Uma proposta alternativa para avaliação do desempenho de sistemas de transporte emergencial de saúde brasileiros." Transportes, v. 9, n. 2, p. 9-27. 2011.
- ALVARENGA, G. B. (2005), "Um algoritmo híbrido para os problemas de roteamento de Veículos Estático e Dinâmico com Janela de Tempo."
- MATHEW, N.; SMITH, S. L.; WASLANDER, S. L. (2015), "Planning paths for package delivery in heterogeneous multirobot teams.", IEEE Transactions on Automation Science and Engineering, v. 12, n. 4, p. 1298-1308.
- KUROSE, J. F.; ROSS, K. W. (2013), "Redes de Computadores e a Internet 6ª edição.", São Paulo: Person.
- DREYFUS, S. E. (1969), "An appraisal of some shortest-path algorithms." Operations research 17.3.
- AHUJA, R. K.; MAGNANTI, T. L.; ORLIN, J. B. (1993), "Network flows: theory, algorithms, and applications."
- MÉNDEZ, Y. S.; GUARDIA, L. E. T. (2008), "Problema do caminho mais curto- Algoritmo de Dijkstra."
- PARES, R. (2016), "Algoritmo de Dijkstra.". Revista programar. Edição 53. 19 de dezembro. Disponível em: <<https://www.revista-programar.info/artigos/algoritmo-de-dijkstra>>. Acesso em: 3 Abril 2017.
- CORMEN, T. H. (2009), "Introduction to algorithms." MIT press.
- FLOYD, R. W. (1962), "Algorithm 97: shortest path." Communications of the ACM 5.6:345.

- Roy, B. (1959), "Transitivite et connexité".Comptes Rendus Hebdomadaires Des Seances De L Academie Des Sciences, 249(2):216–218.
- WARSHALL, S. (1962), "A theorem on boolean matrices", Journal of the ACM (JACM), 9(1):11–12.
- DEREKENARIS, G., et al. (2001), "Integrating GIS, GPS and GSM technologies for the effective management of ambulances." Computers, Environment and Urban Systems 25.3: 267-278.
- PANAHI, S., and M. R. DELAVAR. (2008), "A GIS-based dynamic shortest path determination in emergency vehicles." World applied sciences journal 3.1: 88-94.
- JOTSHI, A., GONG, Q., BATTA, R. (2009), "Dispatching and routing of emergency vehicles in disaster mitigation using data fusion." Socio-Economic Planning Sciences 43.1: 1-24.
- ELALOUF, A. (2012), "Efficient routing of emergency vehicles under uncertain urban traffic conditions."
- KRITZINGER, S., et al. (2012), "Using traffic information for time-dependent vehicle routing." Procedia-Social and Behavioral Sciences 39: 217-229.
- MUSOLINO, G., et al. (2013), "Travel time forecasting and dynamic routes design for emergency vehicles." Procedia-Social and Behavioral Sciences 87: 193-202.