Proposta de framework interativo para aprendizagem e resolução da problemática de caminho mínimo pelo algoritmo de Dijkstra e Programação Matemática

Miguel Ângelo Lellis Moreira
miguellellis@hotmail.com
UFF

Isaque David Pereira de Almeida
isaque.mestrado@gmail.com
UFF

Renata Raposo Del Vecchio
rrdelvecchio@id.uff.br
UFF

Marcos dos Santos
marcosdossantos_doutorado_uff@yahoo.com.br
IME

Carlos Francisco Simões Gomes
cfsg1@bol.com.br
UFF

Resumo:O estudo em questão apresenta uma proposta de framework computacional como modelo de análise ao problema de caminho mínimo. Neste contexto, a ferramenta computacional viabiliza a implementação de dois modelos matemáticos, algoritmo de Dijkstra e Modelagem por Programação inteira, ambos favoráveis como forma de resolução da problemática de identificar o caminho de menor custo em um dado grafo. Para o desenvolvimento do software, utilizou-se da linguagem R e seu pacote de desenvolvimento web Shiny, proporcionando a criação de uma interface interativa com o usuário e de acesso online. Como exemplo, apresenta-se um hipotético estudo de caso baseado na problemática de caminho mínimo, explorando os diferentes modelos de análise do conjunto de dados e seus possíveis tipos de resultados gerados por recursos numéricos e gráficos. Ao final do estudo, entende-se que dado framework tem um relativo potencial para resolução da problemática em questão, como também pode servir e apoio na aprendizagem de conceitos basilares em grafos, modelos heurísticos e programação matemática, presentes na Pesquisa Operacional.

Palavras Chave: Teoria dos grafos - Caminho mínimo - Dijkstra - Programação Inteira - Modelo computacional

1. INTRODUÇÃO

A Teoria dos grafos é um campo da Matemática destinada ao estudo de conjuntos de objetos interligados entre si, chamados de grafos, também reconhecidos como redes, formadas por vértices e arestas (FARIAS *et al*., 2021). Dada teoria foi gradualmente desenvolvida no século XIX, quando surgiram importantes aplicações em engenharia e em química. Sua importância cresceu muito a partir do século XX, com o surgimento das redes de energia elétrica e de telecomunicações, dos circuitos digitais e, por fim dos computadores (SANTOS *et al.*, 2018).

Conforme apresentado por Maquengo (2019), tem sido explorado o potencial da teoria dos grafos como suporte ao tratamento de redes viárias e elétricas destacando-se a importância da rede rodoviária urbana, modelando-se as redes por meio de um grafo onde cada interseção/subestações/posto de corte é considerado um vértice/nó e as arestas/ramos são trechos de estradas asfaltadas/linha elétricas ligando as interseções (RODRIGUES JUNIOR; SANTOS; FERNANDES, 2017).

Neste contexto, conforme abordado por Neves (2007), dentre os modelos de otimização combinatória, o problema do caminho mínimo tem sido considerada fundamental. Diversos problemas do mundo real podem ser modelados dessa forma, tais como: percurso de menor custo entre duas cidades, transmissão de dados em uma rede de computadores, reconhecimento de voz, segmentação de imagens entre outros (NEVES *et al*, 2021).

Os problemas de caminho mínimo em grafos, surgem quando se pretende determinar o caminho mais curto ou de menor custo entre vértices de uma rede. Os algoritmos associados à determinação de caminhos mais curtos são frequentemente aplicados quando se estudam problemas em redes como de transportes ou comunicações (SENNA *et al*., 2015).

Dentre as ferramentas de tratamento do problema de caminho mínimo, tem-se o algoritmo de *Dijkstra*, possibilitando sua resolução por meio da geração de árvores de caminhos mínimos, esclarecendo a rota menos custosa entre dois pontos de um grafo (BARROS; PAMBOUKIAN; ZAMBONI, 2007). Outra abordagem possível do problema de caminho mínimo é mediante otimização por Programação Matemática, obtendo um resultado otimizado em um dado contexto (AMORIM, *et al*., 2013).

Neste cenário, o estudo em questão apresenta um framework computacional que viabiliza o tratamento de problema de caminho mínimo tanto por meio do algoritmo de *Dijkstra*, quanto por Programação Inteira. Dado modelo proporciona uma interface interativa e de acesso online, auxiliando tanto na implementação do modelo quanto em sua aprendizagem.

2. FUNDAMENTÇÃO TEÓRICA

O Problema do Caminho Mínimo é muito conhecido na literatura da Pesquisa Operacional (PO). O objetivo é encontrar o caminho de custo mínimo conectando um nó de origem e um nó de destino em uma rede. As aplicações variam desde a computação da melhor rota em navegadores GPS e obtenção de rotas de risco mínimo para remessas de materiais perigosos até protocolos de roteamento em redes IP ou sem fio (GARCÍA-HEREDIA *et al*., 2021).

Segundo Yuan (2021), a crescente quantidade de dados gerados a partir de conexões de rede cada vez mais sofisticadas requer maior precisão e maior eficiência na localização dos caminhos mais curtos. De acordo com Nascimento (2020), o caminho mínimo entre duas cidades consiste na sequência de vias que resulta na menor distância. Não necessariamente o “menor” está associado à menor distância total percorrida; o conceito é mais genérico, e relacionado ao mínimo custo, considerando algum atributo quantificável, como, por exemplo, distância, tempo, risco etc.

2.1 ALGORITMO DE *DIJKSTRA*

O algoritmo proposto por *Dijkstra* (1959) trata o problema de caminhos mínimos em grafos sem arestas de peso negativo, o que não chega a ser restritivo na maior parte das aplicações. Desde então, o algoritmo tem sido refinado com o uso de estruturas de dados cada vez mais sofisticadas, reduzindo seu tempo de execução de pior caso (NEVES, 2007). De acordo com *Dijkstra* (1959), o processo também pode ser aplicado no caso em que o comprimento de uma ramificação depende da direção em que ela é percorrida, referenciando-se aos grafos direcionados.

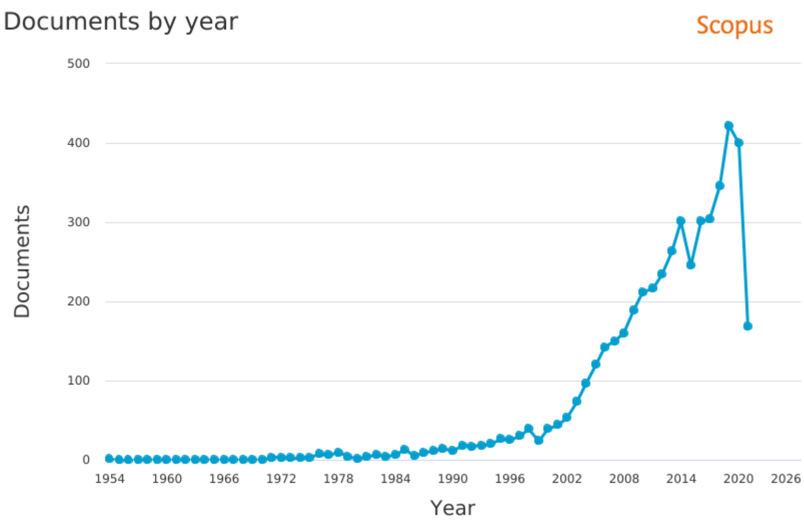
Uma vez conhecidos os pesos das arestas, uma forma de caminhar pelo grafo poderá ser definida. Um dos algoritmos mais conhecidos especializados em solucionar o caminho de custo mínimo é o algoritmo de *Dijkstra*. Sem reduzir o desempenho, este algoritmo é capaz de determinar o caminho mínimo, partindo de um vértice-fonte para todos os outros vértices do grafo (NASCIMENTO, 2020).

Firmino (2019) afirma que o algoritmo de *Dijkstra* é um processo interativo, no qual cada relação consiste em repetir um conjunto de passos até que o nó alvo seja atingido. Adicionando o nó inicial à fila de prioridade, realiza-se os seguintes passos:

a) obter nó da fila de prioridade com menor valor e que não tenha sido visitado;
b) verificar se o nó escolhido (n) é o nó alvo e encerra a busca, senão continua a busca;
c) explorar os nós descendentes do anterior escolhido (n), onde os nós descendentes são os nós conectados por uma aresta a esse nó n;
d) computar a distância acumulada do nó inicial aos filhos de n;
e) adicionar os nós filhos de n à fila de prioridade e marca o nó n como visitado.

Realizando uma pesquisa por "*Dijkstra algorithm*", na base SCOPUS, identificou-se 4819 documentos, indicando relativo crescimento após os anos 2000, conforme indicado na figura 1.

Figura 1: Relação de documentos relacionados ao algoritmo de *Dijkstra* na base SCOPUS



Fonte: Autores (2021)

2.2 PROGRAMAÇÃO INTEIRA

Segundo Amorim *et al*. (2013), a modelagem requer o desenvolvimento da habilidade de interação com o problema, recursos e o meio no qual está inserido. Os modelos matemáticos podem ser desenvolvidos por métodos e técnicas específicas como, por exemplo, Programação Linear, Teorias dos Estoques, das Filas ou dos Jogos, Simulação, Análise de Risco etc.

Baseando-se na utilização de programação matemática para resolução de problemas de caminho mínimo, pode-se utilizar de modelos de otimização inteira/binária, compreendendo as arestas do grafo como variáveis do problema, buscando assim o número mínimo de arestas entre dois vértices (HILLIER e LIEBERMAN, 2012). O sistema (1) apresenta uma modelagem genérica à problemática do caminho mínimo por programação inteira, onde i é vértice origem, j é vértice destino, e c o custo ou distância entre vértices.

$$\begin{gathered} Min\ z = \sum_{i=1}^{n} c_{ij} x_{ij} \\ S.a. \\ \sum x_{ji} - \sum x_{ij} = b_i \\ x_i \ldots x_n \geq 0 \end{gathered} \tag{1}$$

3. PROPOSTA DE FRAMEWORK

Conforme discutido na introdução, apresenta-se um modelo computacional baseado no tratamento de dados para resolução da problemática de caminho mínimo. Dado modelo apresenta a integração coleta e tratamento de dados, juntamente com a exposição de informação em base numérica e gráfica, sendo esta apresentada em estrutura de grafos direcionados.

Desenvolvido com base no *software* R (R CORE TEAM, 2021), a ferramenta computacional proporciona uma interação simples e intuitiva, podendo ser acessada de modo online, providenciando uma análise prática de um conjunto de dados relativos ao problema de caminho mais curto em um dado grafo.

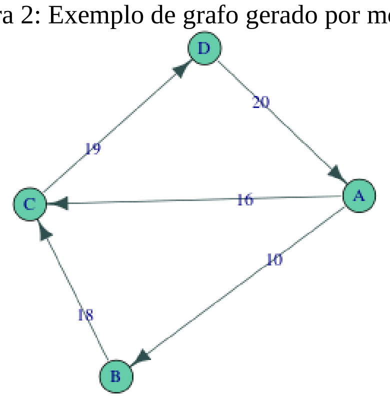
A entrada de dados na plataforma computacional é dada mediante arquivos .csv, comumente utilizados para o compartilhamento de dados entre softwares. Com base no conjunto de dados, o modelo esclarece os vértices do problema, viabilizando uma representação por grafo direcionado, utilizando o pacote *Igraph* (CSARDI *et al.*, 2020), proveniente do próprio software R. A tabela 1 e figura 2 apresentam a estrutura de entrada do modelo e grafo gerado.

Tabela 1: Exemplo de entrada em arquivos .csv

Origem	Destino	Pesos
A	B	10
A	C	16
B	C	18
C	D	19
D	A	20

Fonte: Autores (2021)

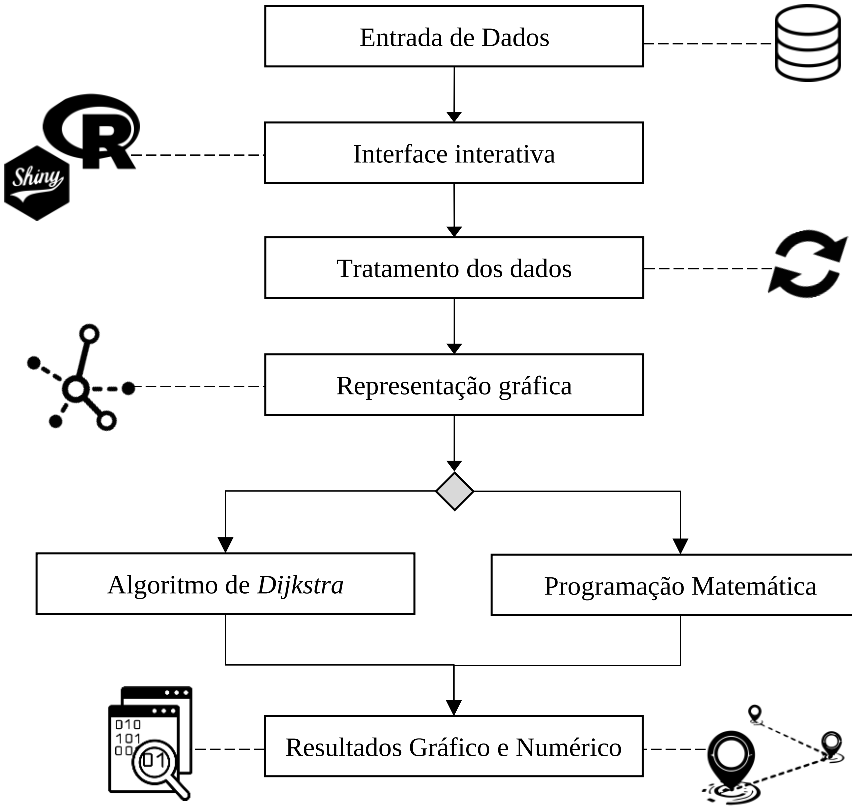
Figura 2: Exemplo de grafo gerado por modelo



Fonte: Autores (2021)

Dada a possibilidade de resolução do problema de caminho mínimo mediante diferentes tipos de algoritmos, o framework desenvolvido viabiliza dada resolução mediante a heurística de *Dijkstra* e por modelagem de Programação Inteira. Uma representação geral do modelo desenvolvido é exposta na figura 3.

Figura 3: Esquematização do framework proposto



Fonte: Autores (2021)

3.1 INTEGRAÇÃO DO ALGORITMO DE DIJKSTRA

A integração do algoritmo de *Dijkstra* ao modelo computacional é viabilizada pelo pacote *Igraph*, a partir do momento em que dado recurso possui conjuntos de funções destinado à resolução do algoritmo. A resolução por dado modelo apresenta a descrição gráfica dos vértices pertencentes ao caminho mínimo, descrição textual e apresentação de uma matriz indicando os caminhos mínimos possíveis entre todos os vértices do problema, conforme exposto na tabela 2, utilizando o conjunto de dados apresentado na tabela 1.

Tabela 2: Exemplo de matriz de todos caminhos mínimos possíveis

	A	B	C	D
A	0	10	16	35
B	57	0	18	37
C	39	49	0	19
D	20	30	36	0

Fonte: Autores (2021)

3.2 INTEGRAÇÃO DE PROGRAMAÇÃO INTEIRA

Utilizando os pacotes LpSolve (BERKELAAR, 2020) e LpSolveAPI (KONIS; SCHWENDINGER, 2020), é possibilitado a modelagem de problemas de programação matemática ao software. Em dado modelo, cada caminho indicado no conjunto de entradas é convertido em uma variável de decisão em um problema de otimização, onde busca-se a minimização da soma dos custos do caminho mínimo ótimo ao problema.

Desta forma define-se que cada variável $x_i = \{0\,,1\}$, indicando se o caminho será percorrido ou não, e seu respectivo custo c_i, buscando assim a minimização do problema, $Min\ Z = \sum_{i=1}^{n} x_i c_i$.

Quanto ao modelo computacional, é proporcionado ao usuário a obtenção do resultado, caso possível, de um caminho mínimo entre dois vértices indicados, assim como a geração de uma tabela contendo todas as variáveis do problema juntamente com o resultado binário da decisão e a impressão da modelagem matemática do problema de otimização, conforme indicado na figura 4.

Figura 4: Exemplo de modelagem matemática gerada pelo software

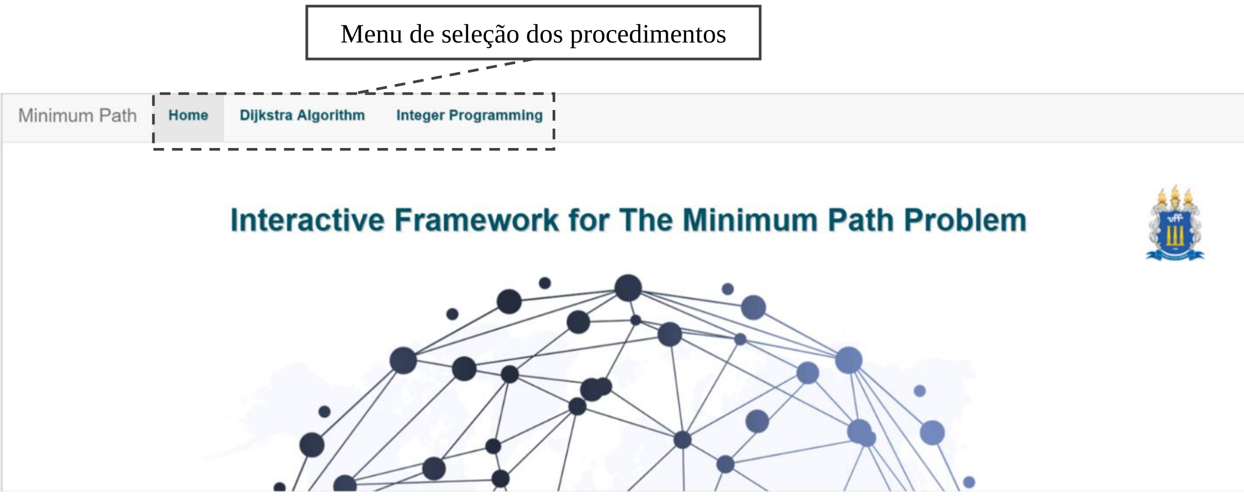
```
                    X...Objective.function...
min: +10 C1 +16 C2 +18 C3 +19 C4 +20 C5;
                          /* Constraints */
                               +C1 +C2 = 1;
                               -C1 +C3 = 0;
                           -C2 -C3 +C4 = 0;
                              R4: -C4 = -1;
                      /* Variable bounds */
                                   C1 <= 1;
                                   C2 <= 1;
                                   C3 <= 1;
                                   C4 <= 1;
                                   C5 <= 1;
                  /* Integer definitions */
                        int C1,C2,C3,C4,C5;
```

Fonte: Autores (2021)

3.3 INTERFACE COMPUTACIONAL

A interface de interação do modelo foi desenvolvida com base no pacote *Shiny* (CHANG *et al*., 2021), sendo este destinado à construção de ambientes computacionais interativos ao usuário, extinguindo a necessidade de conhecimento em programação computacional. A plataforma possui três páginas de acesso, sendo a primeira a página inicial do modelo, disponibilizando o acesso aos recursos do modelo, conforme exposto na figura 5.

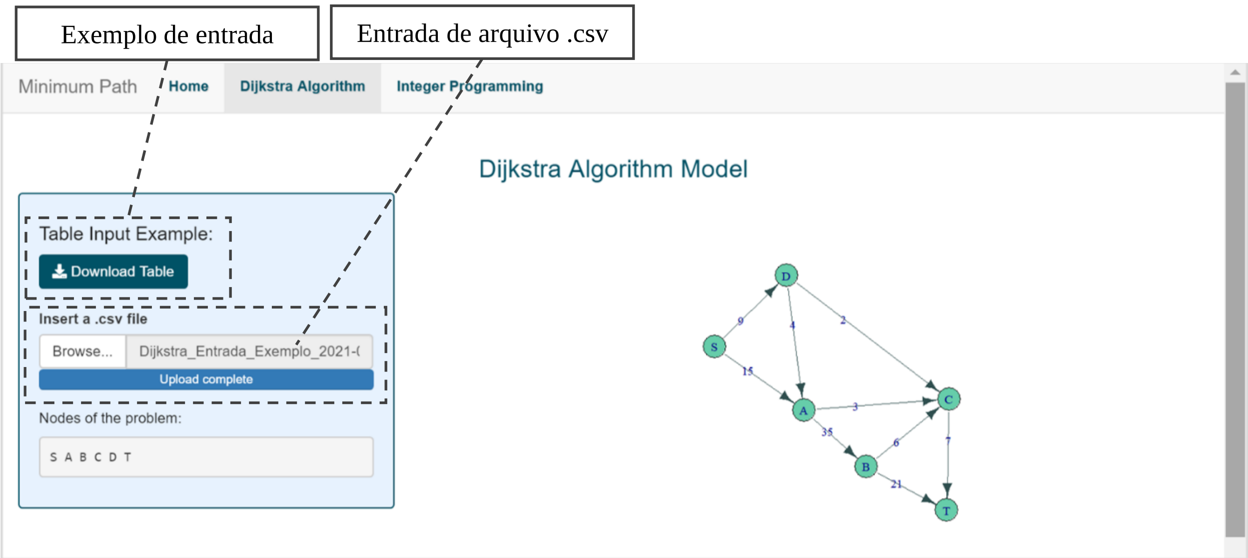
Figura 5: Página inicial da plataforma computacional



Fonte: Autores (2021)

Ambas as páginas de resolução do modelo possuem a mesma estrutura de entrada, possibilitando o salvar no sistema operacional um exemplo de entrada em formato .csv, a entrada pelo mesmo formato do arquivo e a exposição do grafo gerado, conforme indicado na figura 6.

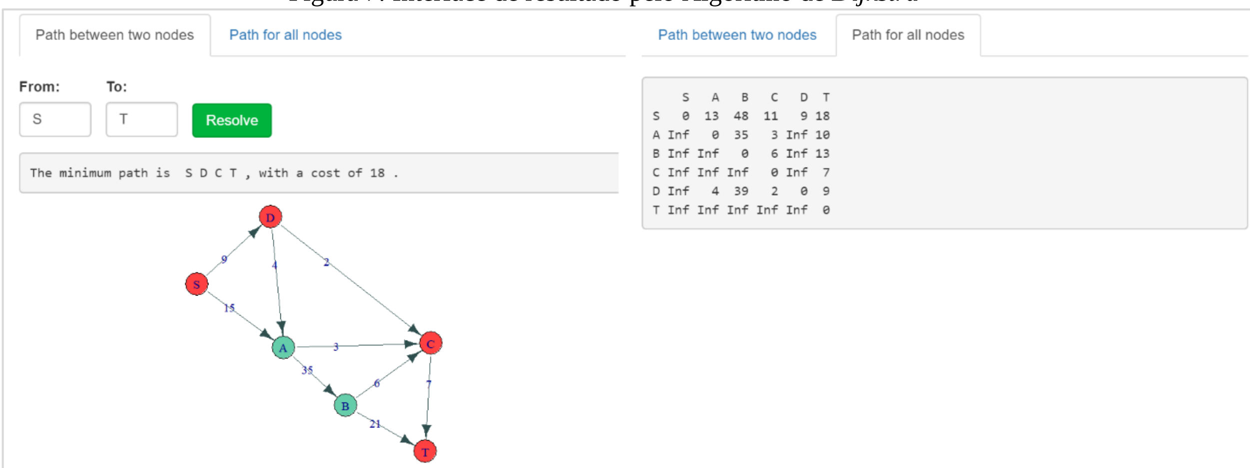
Figura 6: Interface de entrada de dados ao modelo computacional



Fonte: Autores (2021)

Para a resolução do Algoritmo de *Dijkstra*, basta indicar dois vértices do problema, que será obtido o caminho mínimo do problema. De forma adicional, a matriz indicando todos os caminhos mínimos possíveis do problema é apresentada na aba ao lado, conforme exposto na figura 7.

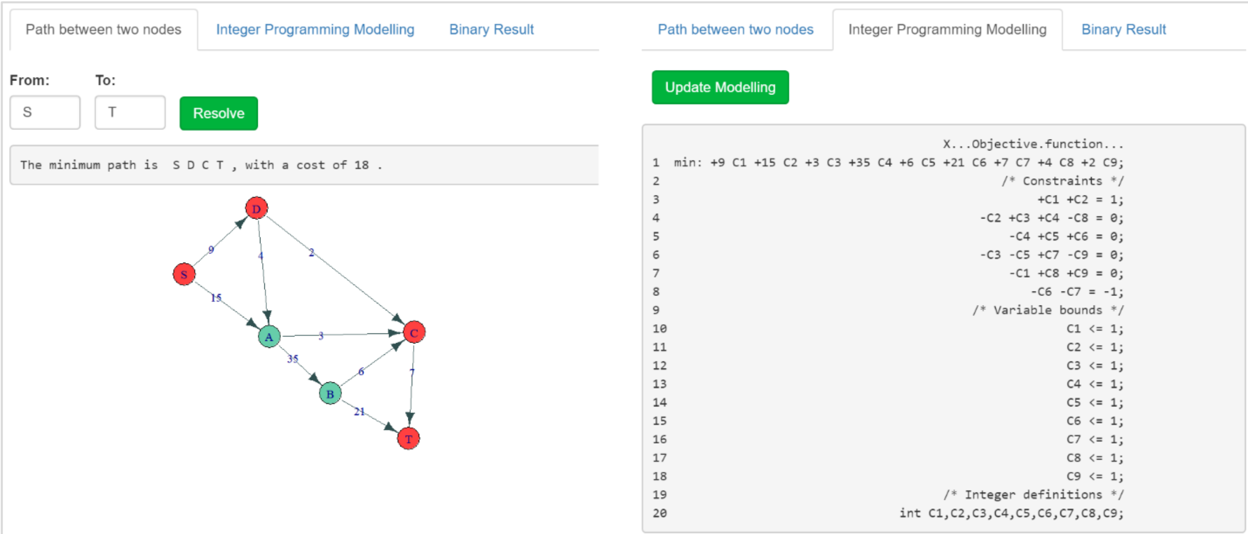
Figura 7: Interface de resultado pelo Algoritmo de *Dijkstra*



Fonte: Autores (2021)

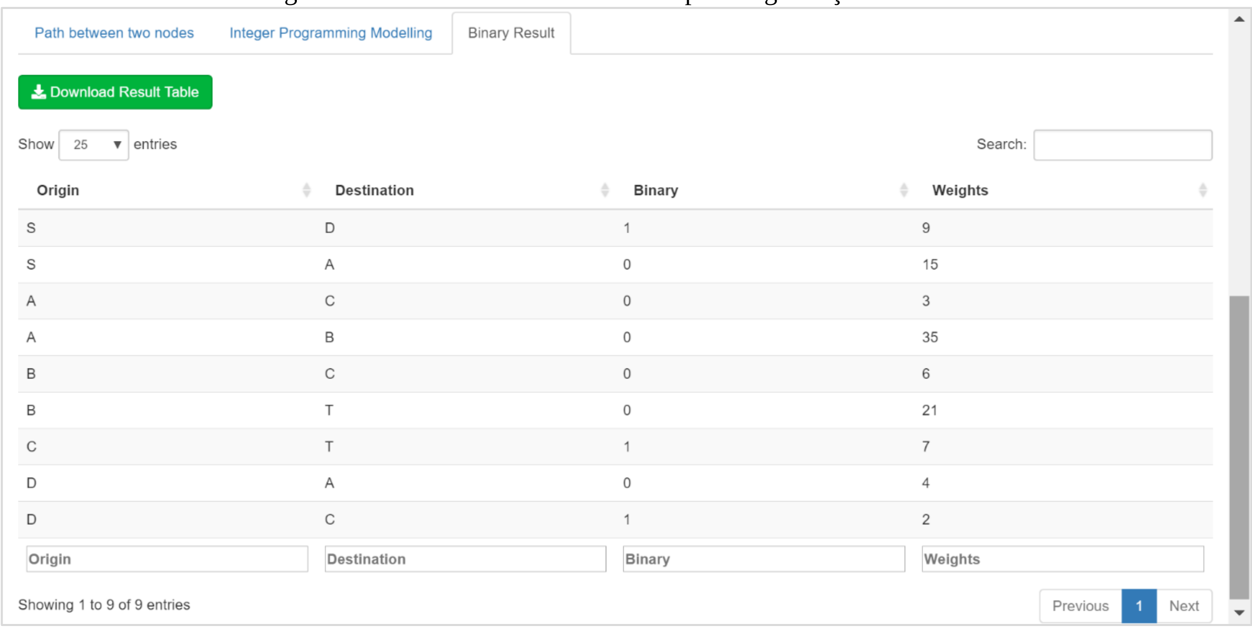
Para a resolução do problema mediante Programação Inteira, é necessário apenas indicar os vértices de origem e destino desejados, apresentando a solução ótima da problemática, sua respectiva modelagem matemática e por fim o detalhamento do resultado binário, apresentados nas figuras 8 e 9 respectivamente.

Figura 8: Interface de resultado por Programação Inteira



Fonte: Autores (2021)

Figura 9: Interface de resultado binário por Programação Inteira



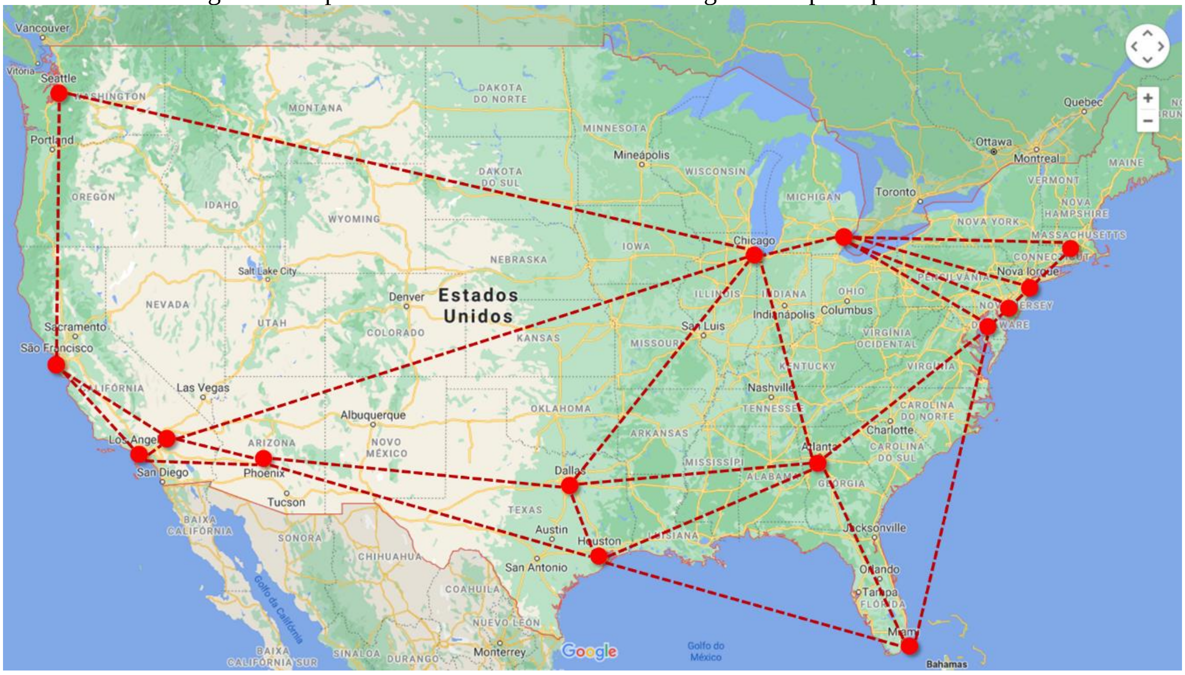
Origin	Destination	Binary	Weights
S	D	1	9
S	A	0	15
A	C	0	3
A	B	0	35
B	C	0	6
B	T	0	21
C	T	1	7
D	A	0	4
D	C	1	2
Origin	Destination	Binary	Weights

Fonte: Autores (2021)

4. APLICAÇÃO NUMÉRICA

Em prol de proporcionar uma implementação prática do modelo, utiliza-se de um estudo de caso hipotético baseado na análise de caminho mínimo entre um conjunto de cidades dos Estados Unidos. Para dada avaliação, definiu-se quinze cidades e vinte e seis caminhos possíveis entre duas cidades do modelo. A figura 9 e tabela 2 apresentam os dados respectivos à problemática.

Figura 9: Mapa dos Estados Unidos baseada em grafo das principais cidades



Fonte: Autores (2021)

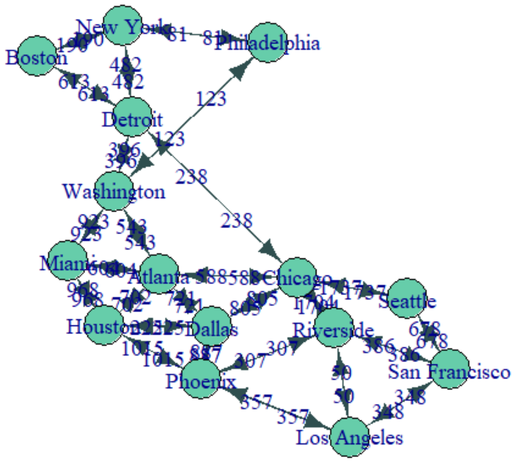
Tabela 2: Conjunto de entradas ao modelo computacional

Origem	Destino	Distância (km)
Seattle	Chicago	1737
Seattle	San Francisco	678
San Francisco	Riverside	386
San Francisco	Los Angeles	348
Los Angeles	Riverside	50
Los Angeles	Phoenix	357
Riverside	Phoenix	307
Riverside	Chicago	1704
Phoenix	Dallas	887
Phoenix	Houston	1015
Dallas	Chicago	805
Dallas	Atlanta	721
Dallas	Houston	225
Houston	Atlanta	702
Houston	Miami	968
Atlanta	Chicago	588
Atlanta	Washington	543
Atlanta	Miami	604
Miami	Washington	923
Chicago	Detroit	238
Detroit	Boston	613
Detroit	Washington	396
Detroit	New York	482
Boston	New York	190
New York	Philadelphia	81
Philadelphia	Washington	123

Fonte: Autores (2021)

Importando o conjunto de dados ao modelo computacional, possibilitou-se a representação em grafos do problema, juntamente com a determinação das respectivas distâncias como pesos, conforme exposto na figura 10.

Figura 10: Grafo importado ao modelo computacional



Fonte: Autores (2021)

Mediante o algoritmo de *Dijkstra* implementado no modelo, possibilitou-se a identificação dos caminhos mínimos entre dois pontos quaisquer e entre todos os pontos possíveis do grafo. Os valores de percursos mínimos entre os pontos do grafo são expostos nas tabelas 3 e 4.

Tabela 3: Distâncias mínimas entre todos os pontos do grafo (parte 1)

	Seattle	San Francisco	Los Angeles	Riverside	Phoenix	Dallas	Houston	Atlanta
Seattle	0	678	1026	1064	1371	2258	2386	2325
San Francisco	678	0	348	386	693	1580	1708	2301
Los Angeles	1026	348	0	50	357	1244	1372	1965
Riverside	1064	386	50	0	307	1194	1322	1915
Phoenix	1371	693	357	307	0	887	1015	1608
Dallas	2258	1580	1244	1194	887	0	225	721
Houston	2386	1708	1372	1322	1015	225	0	702
Atlanta	2325	2301	1965	1915	1608	721	702	0
Miami	2929	2676	2340	2290	1983	1193	968	604
Chicago	1737	2090	1754	1704	1692	805	1030	588
Detroit	1975	2328	1992	1942	1930	1043	1268	826
Boston	2588	2941	2605	2555	2543	1656	1639	937
New York	2457	2810	2474	2424	2355	1468	1449	747
Philadelphia	2494	2847	2511	2461	2274	1387	1368	666
Washington	2371	2724	2388	2338	2151	1264	1245	543

Fonte: Autores (2021)

Tabela 4: Distâncias mínimas entre todos os pontos do grafo (parte 2)

	Miami	Chicago	Detroit	Boston	New York	Philadelphia	Washington
Seattle	2929	1737	1975	2588	2457	2494	2371
San Francisco	2676	2090	2328	2941	2810	2847	2724
Los Angeles	2340	1754	1992	2605	2474	2511	2388
Riverside	2290	1704	1942	2555	2424	2461	2338
Phoenix	1983	1692	1930	2543	2355	2274	2151
Dallas	1193	805	1043	1656	1468	1387	1264
Houston	968	1030	1268	1639	1449	1368	1245
Atlanta	604	588	826	937	747	666	543
Miami	0	1192	1319	1317	1127	1046	923
Chicago	1192	0	238	851	720	757	634
Detroit	1319	238	0	613	482	519	396
Boston	1317	851	613	0	190	271	394
New York	1127	720	482	190	0	81	204
Philadelphia	1046	757	519	271	81	0	123
Washington	923	634	396	394	204	123	0

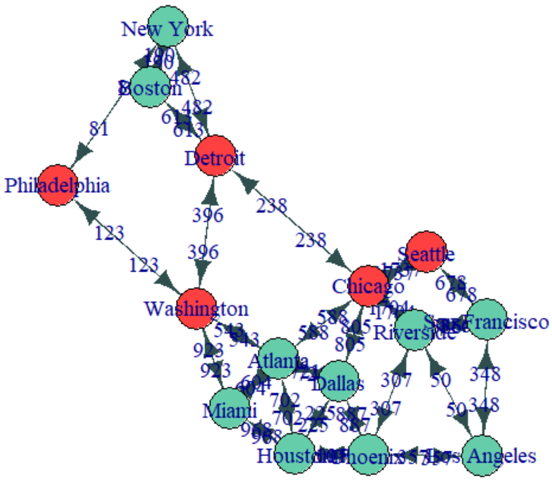
Fonte: Autores (2021)

Realizando uma análise entre as cidades de Seattle e Philadelphia, utilizou-se do modelo de Programação Inteira como ferramenta de resolução. Desta forma, a modelagem binária é desenvolvida com base nas seguintes restrições:

- A soma das arestas em que Seattle é ponto de origem, devem ser iguais a 1;
- A soma das arestas em que Philadelphia é ponto de destino, devem ser iguais a -1;
- A soma das arestas em que Philadelphia é ponto de origem, devem ser iguais a 0;
- As demais arestas, quando esta for anterior a um vértice recebe 1, posterior recebe -1, e a soma destas deve ser igual a 0.

Neste contexto obteve-se o caminho mínimo de 2494km, passando pelas cidades de Seattle, Chicago, Detroit, Washington e Philadelphia. A representação em grafos do resultado é apresentada na figura 11 e a indicação das variáveis ótimas são expostas na figura 12.

Figura 11: Grafo do resultado pelo modelo de Programação Inteira



Fonte: Autores (2021)

Figura 12: Resultado binário pelo modelo computacional de Programação inteira

Path between two nodes | Integer Programming Modelling | Binary Result

Download Result Table

Show 25 entries | Search:

Origin	Destination	Binary	Weights
Seattle	Chicago	1	1737
Chicago	Detroit	1	238
Detroit	Washington	1	396
Washington	Philadelphia	1	123
Seattle	San Francisco	0	678
San Francisco	Riverside	0	386
San Francisco	Los Angeles	0	348
Los Angeles	Riverside	0	50
Los Angeles	Phoenix	0	357
Riverside	Phoenix	0	307
Riverside	Chicago	0	1704

Fonte: Autores (2021)

Até o dado momento de desenvolvimento do estudo, não foi identificado limitações quanto à estrutura de uma problemática, viabilizando o tratamento de problemas simples e complexos, deste que os dados de entrada estejam conforme o formato de entrada exigido pelo modelo.

5. CONSIDERAÇÕES FINAIS

Dado estudo aborda o desenvolvimento de um modelo computacional como forma de apoio na aprendizagem e implementação do algoritmo de *Dijkstra* e Programação Matemática para a problemática de caminho mínimo. Como suporte de desenvolvimento, utilizou-se a linguagem R e seu pacote *Shiny* para interação em ambiente web.

O modelo proposto, viabiliza a entrada de dados de forma simples, viabilizando a representação gráfica, em grafos, dos pontos e caminhos de um dado problema. Respectivo ao algoritmo de *Dijkstra*, é viabilizado identificar o caminho mínimo entre dois vértices, também sendo esclarecidos os caminhos mínimos possíveis entre todos os pontos do conjunto de dados. Vale ressaltar que dado algoritmo baseia-se em um modelo heurístico, não necessariamente apresentando o resultado ótimo, mas sim um resultado satisfatório ao problema.

Quanto ao modelo de programação matemática, além de também proporcionar o esclarecimento do caminho de menor custo ao problema, também é viabilizado ao usuário a análise da modelagem matemática processada no modelo juntamente com a exposição das variáveis de caráter binário, levando a construção do percurso ótimo em contexto.

Neste cenário, pode-se concluir que dada plataforma tem grande potencial de inovação quanto a aplicação nos estudos e aprendizagem dos modelos integrados ao software, explorando não somente os resultados gerados, mas também a construção dos modelos de análise. Para estudos futuros, pretende-se realizar a exploração e desenvolvimento de novos modelos de análise por grafos e basilares no estudo da Pesquisa Operacional.

REFERÊNCIAS

AMORIM, G. D. A. F. *et al*. Pesquisa operacional: modelagem matemática do planejamento de culturas em uma fazenda familiar. XX Simpósio De Engenharia de Produção (SIMPEP). 2013.

BARROS, E. A. R.; PAMBOUKIAN, S. V. D.; ZAMBONI, L. C. Algoritmo de *dijkstra*: apoio didático e multidisciplinar na implementação, simulação e utilização computacional. International Conference on Engineering and Computer Education, São Paulo. 2007.

BERKELAAR, M. *LpsolveR package version* 5.6.15, 2020.

CHANG, W.; CHENG, J.; ALLAIRE J.; SIEVERT C.; SCHLOERKE B.; XIE Y.; ALLEN J. Shiny. (2021).

CSARDI, G. *et al*. *Igraph package version* 1.2.6, 2020.

DIJKSTRA, E. W. *A note on two problems in connexion with graphs*. Numerische Mathematik, v. 1, n. 1, p. 269–271, 1959.

FARIAS, N. F. DE S. *et al*. Dimensionamento de redes de distribuição de água utilizando Teoria dos Grafos. Brazilian Journal of Development, v. 7, n. 5, p. 52544–52561, 2021.

FIRMINO, A. S. Métodos de otimização aplicados ao problema de recuperação de contêineres. Teses de Doutorado - Ciência da Computação. Universidade Federal de Pernambuco. 2019.

GARCÍA-HEREDIA, D. *et al*. *A solution method for the shared resource-constrained multi-shortest path problem*. Expert Systems with Applications, v. 182, p. 115193, 2021.

HILLIER, F. S.; LIEBERMAN, G. J. Introdução à Pesquisa Operacional. 9ª Edição. São Paulo: Mcgraw Hill, 2012.

KONIS, K.; SCHWENDINGER, F. *LpsolveAPI R package version* 5.5.2., 2020.

MAQUENGO, G. L. Teoria dos grafos e aplicações: redes eléctricas e de transportes rodoviários. Dissertação de Mestrado. Universidade de Évora. 2019.

NASCIMENTO, R. A. Implementação de um algoritmo para solução do caminho de custo mínimo na logística aplicada ao transporte rodoviário de soja do estado de Mato Grosso. Brazilian Journal of Business, v. 2, n. 3, p. 3128–3141, 2020.

NEVES, C.N.R., SANTOS, M., GOMES, C.F.S. e QUINTAL, R.S. Utilização do método do caminho crítico em obras no Complexo Naval da Ilha do Governador. Diversitas Journal, 6(1), 396-410. 2021.

NEVES, P. T. Variações e aplicações do algoritmo de *Dijkstra*. Dissertação (mestrado) - Universidade Estadual de Campinas, Instituto de Computação, Campinas, SP. 2007. Disponível em: <http://www.repositorio.unicamp.br/handle/REPOSIP/276210>

R CORE TEAM. R: *A Language and Environment for Statistical Computing*. (2021).

RODRIGUES JUNIOR, L.; SANTOS, M.; FERNANDES, M. C. Aplicação do *travelling salesman problem* na roteirização das viaturas da Marinha do Brasil: uma abordagem da teoria dos grafos. IX SIMPROD, 2017.

SANTOS, M.; NETO, J. C.; OLIVEIRA, R.; WALKER, R. A.; & REIS, M. Conexidade e sinergia nas universidades brasileiras: uma abordagem a partir da teoria dos grafos. XXXVIII Encontro Nacional de Engenharia de Produção (ENEGEP). 2018

SENNA, P. *et al*. Ferramenta desenvolvida em Visual Basic for Applications (VBA) para o ensino de algoritmos de caminho mínimo. XLVII Simpósio Brasileiro de Pesquisa Operacional (SBPO). 2015.

YUAN, H. *et al*. *A new exact algorithm for the shortest path problem: An optimized shortest distance matrix*. Computers & Industrial Engineering, v. 158, p. 107407, 2021.