

FERRAMENTA DESENVOLVIDA EM VISUAL BASIC FOR APPLICATIONS (VBA) PARA O ENSINO DE ALGORITMOS DE CAMINHO MÍNIMO

Pedro Senna

CEFET/RJ UnED Nova Iguaçu
Estrada Adrianópolis 1.317, Santa Rita, Nova Iguaçu, RJ
pedro.sennavieira@gmail.com

Andreia Macedo Gomes

CEFET/RJ UnED Nova Iguaçu
Estrada Adrianópolis 1.317, Santa Rita, Nova Iguaçu, RJ
andreiamg@hotmail.com

Augusto Reis

CEFET/RJ
Avenida Maracanã, 229, Maracanã, Rio de Janeiro, RJ
acreis_1@yahoo.com.br

Julio Cesar Guimarães

Faculdade Meridional (IMED)
Rua Senador Pinheiro, 304, Cruzeiro, Passo Fundo, RS
juliofcguimaraes@yahoo.com.br

RESUMO

Este trabalho tem como objetivo apresentar a ferramenta desenvolvida em ambiente Excel/VBA para ensinar o algoritmo de Dijkstra aos alunos. A ferramenta possui recursos didáticos visuais que visam a motivar o aluno na medida em que torna possível a execução e correção de exercícios de fixação, assim como a aplicação da ferramenta em pequenos projetos reais de Pesquisa Operacional.

PALAVRAS CHAVE: Grafos, Algoritmo Dijkstra, Ensino

Área principal: EDU

ABSTRACT

This work aims to present a tool developed using Excel / VBA environment which has the purpose of teaching Dijkstra's algorithm to students. The tool has visual teaching resources which can bring extra motivation to students as it makes possible the implementation and correction of practice exercises, as well as the application of the tool in small real projects of Operational Research.

KEYWORDS: Graphs, Dijkstra Algorithm, Teaching

Main Area: EDU

1. Introdução

A Pesquisa Operacional consiste em uma área do conhecimento com notória e crescente importância. Vários algoritmos e técnicas exatas de resolução de problemas são implementadas nos pacotes de otimização disponíveis, fornecendo aos pesquisadores de P.O. uma grande gama de ferramentas. Entretanto, quando são buscadas ferramentas que auxiliem no ensino das técnicas, não são encontradas muitas opções de softwares. Wang e Li (2013) criticam a falta de flexibilidade nas aulas de Pesquisa Operacional e o fato de muitas vezes as aulas não serem dadas no computador. Neste sentido, este trabalho busca trazer uma contribuição para a literatura acadêmica e para o ensino da Pesquisa Operacional em cursos de Graduação ou Pós-Graduação ao oferecer uma ferramenta de ensino de algoritmos, mais especificamente do algoritmo de Dijkstra.

Uma busca na base SCOPUS pelo string “*Operations Research*” revela um quantitativo de mais de 33.000 resultados. No entanto, quando o termo “*Operations Research*” é buscado juntamente com o termo “*Teaching*”, os resultados caem drasticamente para cerca de 300 trabalhos e ainda que seja possível perceber leve tendência de crescimento, a publicação destes trabalhos ainda é bastante sazonal e raramente são encontrados em periódicos.

Com uma análise fina nestes 300 trabalhos percebe-se ainda que a quantidade de trabalhos que se preocupam com o desenvolvimento de uma ferramenta didática para que os alunos possam vislumbrar a aplicação de alguma técnica é irrisória. Assim, nota-se que a pesquisa em P.O. não tem dado a devida importância ao desenvolvimento de ferramentas de ensino.

Entre alguns trabalhos que tratam do ensino de Pesquisa Operacional incluem-se Fonseca e Pinto (2013) que apresentam um sistema Web de apoio para cursos de Programação Linear e Pesquisa Operacional. Pinto e Borenstein (2006) apresentam um sistema para resolução de PPL's usando as fases 1 e 2 do método Simplex. Alguns artigos mostram inovações no sistema educacional como por exemplo o trabalho de Roeder e Roeder (2013) onde as autoras mostram uma experiência de ensino de Pesquisa Operacional no ensino médio, cujo principal objetivo é mostrar aos jovens estudantes as aplicações práticas da Álgebra Linear,

No que diz respeito ao ensino de P.O. de forma ampla, Yang et al. (2012) discutem métodos de ensino de pesquisa operacional e ressaltam a importância do uso de métodos computacionais para o ensino das técnicas.

Cochran (2012) cita alguns parâmetros que em sua visão deveriam guiar o ensino de forma ampla:

- Estimular o interesse dos estudantes
- Promover que os estudantes obtenham uma compreensão profunda do tema tratado.
- Desenvolver o pensamento crítico dos estudantes
- Encorajar os estudantes a terem pensamentos independentes
- Construir a autoconfiança dos estudantes

No que diz respeito, mais especificamente à Pesquisa Operacional, o autor cita pontos como:

- Identificar e resolver problemas interessantes e relevantes
- Demonstrar o valor da Pesquisa Operacional

Shaohong et al. (2010) afirmam que o “corpo” principal do ensino é formado por professores e estudantes. Os autores afirmam ainda que dado a dificuldade do ensino de algoritmos, é necessário mobilizar o entusiasmo destes estudantes de forma que eles virem protagonistas em

uma sala de aula. O uso do Excel e do VBA pode servir como um motivador para os alunos, haja vista que em grande parte de seus estágios e empregos eventualmente terão que usar tais softwares. Ainda, o desenvolvimento de ferramentas de ensino em Excel/VBA também foi utilizado por Mesquita e Hernandez (2006), no desenvolvimento de uma ferramenta para ensino da teoria das filas para os alunos.

Neste sentido, objetivando oferecer uma contribuição para as ferramentas disponíveis para o ensino de Pesquisa Operacional, foi implementado em Excel/VBA um programa computacional que gera um grafo, e calcula o menor caminho entre dois pontos escolhidos. De forma bem visual e prática, a ferramenta tem como objetivo despertar nos estudantes o gosto pela programação e pelo estudo de algoritmos aplicáveis a grafos.

2. Algoritmos de caminho mínimo

Os Problemas de caminho mínimo em grafos surgem quando se pretende determinar o caminho mais curto ou mais barato entre um ou vários pares de nós 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.

Ahuja, Magnanti e Orlin. (1988) já afirmavam que projetar e testar algoritmos eficientes para problemas de caminho mais curto são uma importante área de pesquisa na otimização de redes. Existem, na literatura, inúmeros algoritmos eficientes para resolver problemas de caminho mais curto, um dos mais conhecidos é o algoritmo de Dijkstra. Na atualidade, vários trabalhos têm sido encontrados (Galán-García et al, 2014; Sedeño-Noda e Raith, 2015; Parra, Runio e Daze, 2014; Uchida, Takematsu e Honda, 2014; Biswas, Alam, e Doja, 2014; Murota, e Shioura, 2014) utilizando adaptações no algoritmo de Dijkstra com o intuito de contornar desvantagens ou melhorar a sua aplicação para diversas áreas.

A aplicação do algoritmo de Dijkstra se dá em problemas de caminho mais curto de um nó inicial (origem) para todos os demais vértices em grafos de custos positivos nos arcos e se baseia num processo de rotulação de nós. Ahuja, Magnanti e Orlin (1988); Goldbarg e Goldbarg (2012) descrevem detalhadamente o algoritmo de Dijkstra.

Conforme descrito pelos autores, seja um grafo $G = (N, A)$, onde N é o conjunto de nós (ou vértices) do grafo e A é o conjunto de arestas (ou arcos). Considere um comprimento (ou custo) de arco c_{ij} associado com cada arco $(i, j) \in A$. Outros dois conjuntos são considerados no algoritmo, o conjunto C , que armazena os nós fechados, e o conjunto O , que armazena os nós abertos. Os nós fechados são aqueles cujo caminho da origem até ele é conhecido e os nós abertos são os de caminho desconhecido. O pseudocódigo do algoritmo de Dijkstra é:

Ler $G = (N, A)$, e $D = [c_{ij}]$, onde c_{ij} é o custo da aresta (i, j)

$dis[1] \leftarrow 0$

$rot[1] \leftarrow \infty$

Para $i \leftarrow 2$ até n **Faça**

$dis[i] \leftarrow \infty$

$rot[i] \leftarrow 0$

Fim_Para

$O \leftarrow \{N\}$

$$C \leftarrow \infty$$

Enquanto $F \neq N$ **faça**

$$s \leftarrow j \in O, \text{ tal que } dis[i] \text{ é m\u00ednimo dentre os elementos de } O$$

$$C \leftarrow C \cup \{s\}$$

$$V \leftarrow V \setminus C$$

Para $i \in V$ **faça**

$$p \leftarrow \min \{dis[i], (dis[s] + c_{ij})\}$$

se $p < dis[i]$

$$dis[i] \leftarrow p$$

$$rot[i] \leftarrow s$$

Fim_Se

Fim_Para

Fim_Enquanto

Dois vetores s\u00e3o criados pelo algoritmo, o vetor *dis* que armazena a dist\u00e2ncia entre a origem at\u00e9 o v\u00e9rtice que est\u00e1 sendo analisado, e o veto *rot*, que armazena o v\u00e9rtice anterior no caminho da origem at\u00e9 o v\u00e9rtice que est\u00e1 sendo analisado. Esses dois vetores v\u00e3o fornecer os valores dos r\u00f3tulos que cada v\u00e9rtice carrega. Os r\u00f3tulos possuem dois valores, primeiro valor \u00e9 do vetor *rot* e o segundo valor \u00e9 do vetor *dis*. Cada elemento *i* do vetor *dis* armazena a dist\u00e2ncia entre a origem e o v\u00e9rtice de \u00edndice *i* e cada elemento *i* do vetor *rot* guarda o v\u00e9rtice anterior ao v\u00e9rtice *i* no caminho da origem ao v\u00e9rtice *i*. Na etapa de inicializa\u00e7\u00e3o, o algoritmo atribui zero para a dist\u00e2ncia da origem at\u00e9 ela mesma, ou seja, $dis[1]=0$ e atribui infinito para $rot[1]$. Neste momento inicial, os demais r\u00f3tulos tem valor zero atribu\u00eddo ao vetor *rot* e valor infinito atribu\u00eddo \u00e0 dist\u00e2ncia entre eles at\u00e9 a origem, representando que a dist\u00e2ncia entre eles at\u00e9 a origem ainda n\u00e3o foi calculada ou n\u00e3o existe. O conjunto de n\u00f3s abertos recebe *N* e o conjunto de n\u00f3s fechado, inicialmente, \u00e9 vazio. Itera\u00e7\u00f5es v\u00e3o ocorrendo at\u00e9 que todos os n\u00f3s entrem para lista de n\u00f3s fechados (*C*) (GOLDBARG e GOLDBARG, 2012).

A cada itera\u00e7\u00e3o, um v\u00e9rtice *s* \u00e9 removido do conjunto de n\u00f3s abertos (*A*) e \u00e9 inclu\u00eddo na lista de n\u00f3s fechados (*C*). Este v\u00e9rtice *s* escolhido \u00e9 aquele que possui menor valor $dis[s]$. Todos os seus vizinhos ou sucessores comp\u00f5e o conjunto *V*. Cada elemento *i* do conjunto *V* \u00e9 examinado, onde a dist\u00e2ncia da origem at\u00e9 cada um deles \u00e9 calculada e em seguida \u00e9 verificado se essa dist\u00e2ncia \u00e9 maior que a dist\u00e2ncia da origem at\u00e9 *s* mais o valor da aresta c_{si} . Se confirmado, o caminho da origem at\u00e9 o v\u00e9rtice *i* que passa pelo vetor *s* \u00e9 menor que o caminho anterior encontrado entre a origem e *i*. Sendo assim, o valor $dis[i]$ \u00e9 atualizado para $dis[s] + c_{si}$ e $rot[i]$ \u00e9 atualizado com *s*. Desta maneira, o primeiro valor do r\u00f3tulo passa a ser *s* e o segundo valor do r\u00f3tulo passa a ser o valor do caminho da origem at\u00e9 *s* mais o valor da aresta que liga *s* at\u00e9 o v\u00e9rtice *i* que est\u00e1 sendo examinado (GOLDBARG e GOLDBARG, 2014).

A figura 1 mostra o exemplo de um grafo rotulado onde os v\u00e9rtices em cinza s\u00e3o os que j\u00e1 foram examinados e fazem parte da lista *C*, os v\u00e9rtices brancos fazem parte do conjunto *O* e o v\u00e9rtice “a” \u00e9 a origem. O v\u00e9rtice que est\u00e1 sendo examinado encontra-se em destaque, neste caso o v\u00e9rtice “f”. Nesta itera\u00e7\u00e3o pode-se observar que os v\u00e9rtices “a”, “b”, “c”, “d” e “e” j\u00e1 foram examinados e seus r\u00f3tulos modificados. Pode-se observar que quem rotulou “f” foi o v\u00e9rtice “b”

e não o vértice “d”, pois entre “b” e “d”, o que apresentava menor caminho até “f” era o “b”. O vértice “j” apresenta zero no primeiro valor do rótulo e infinito no segundo valor, indicando que ainda não foi examinado.

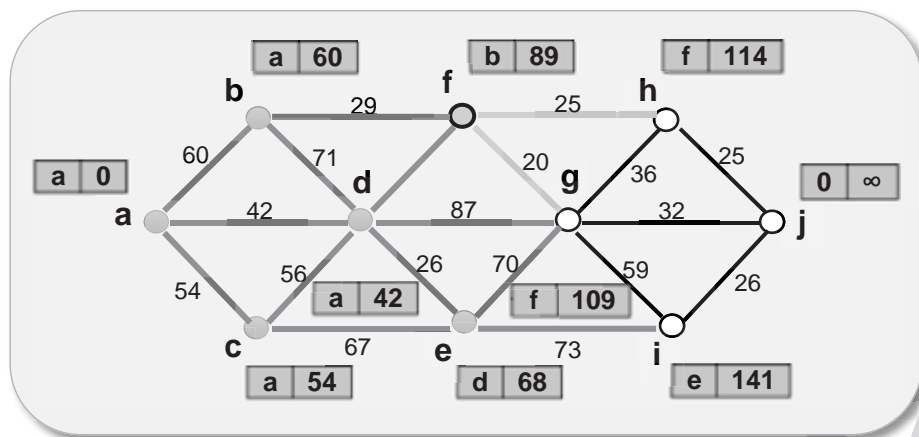


Figura 1 – exame do vértice “f” (Goldbarg e Goldbarg, 2012)

No momento em que “f” está sendo examinado, seus vizinhos, “g” e “h”, entram para o conjunto V e a distância da origem até cada um deles é calculada. Neste exemplo, ambos os rótulos, “g” e “h”, são atualizados mostrando que os caminhos até os dois passam pelo vértice “f”. Observa-se que entre os dois, “g” tem o menor caminho até a origem passando por “f”. Assim prossegue o algoritmo até que todos os vértices sejam examinados. O menor caminho entre o vértice de origem “a” a um vértice qualquer do grafo pode ser recuperado através dos valores de *rot* de cada vértice. Neste exemplo da figura 1, quando todas as iterações forem concluídas, torna-se possível saber que o menor caminho da origem “a” até o vértice “j” é o caminho a-b-f-h-j.

O tempo computacional para o algoritmo de Dijkstra pode ser dividido no tempo requerido para duas operações básicas, selecionar nós e atualizar distâncias. Em uma iteração, o algoritmo requer $O(n)$ tempo para percorrer todos os nós do conjunto de nós abertos e $O(n)$ para atualizar o rótulo de distância do rótulo adjacente. Assim, no geral, esta implementação do algoritmo de Dijkstra roda com uma complexidade $O(n^2)$ onde n é o número de vértices do grafo (AHUJA, MAGNANTI e ORLIN, 1988).

3. A ferramenta proposta

3.1. Funcionamento da ferramenta

A ferramenta foi desenvolvida em ambiente Excel/VBA, objetivando trazer aos alunos um ambiente que é familiar para a maioria dos estudantes de engenharia e mostrar como o software também pode ser aproveitado para a visualização de um grafo e da aplicação de algoritmos de caminho mínimo. A ferramenta construída é simples e possui, basicamente, duas telas. A primeira é visualizada na figura 2:

Entrada de dados

Índice	Nó	Beta	Distância	Coordenadas			Alfa	Origem	Destino
1	1	2	111.8033989					1	6
2	1	3	150	1	50	150	1		
3	2	1	111.8033989	2	100	50	3		
4	2	3	141.4213562	3	200	150	6		
5	2	4	200	4	300	50	9		
6	3	1	150	5	50	200	12		
7	3	4	141.4213562	6	100	300	13		
8	3	5	150.113883						
9	4	3	141.4213562						
10	4	5	291.5475947						
11	4	6	320.1562119						
12	5	6	111.8033989						
13	6	5	111.8033989						

Origem	Destino
1	6

Gerar Rede

Rodar Dijkstra

Figura 2 - Tela inicial

Como pode ser visto na figura 2, os principais elementos são tabela contendo a conectividade dos nós. O índice consiste em uma numeração da lista de nós, as colunas “Nó” e “Beta” indicam os nós que são conectados diretamente por um arco. “Distância” é um parâmetro de entrada que contém os comprimentos de cada arco. A tabela “Coordenadas” contém as coordenadas x e y de cada nó, assim como a coluna “Alfa”, que contém os índices dos nós para serem usados pelo algoritmo. Nas células “Origem” e “Destino” são colocados os *inputs* de origem e destino conforme o desejo do estudante para que a ferramenta possa calcular o caminho mínimo entre os nós determinados. Uma vez que o estudante terminou de preencher as tabelas (eventualmente o Excel poderia receber uma base já pronta) ele pode clicar no botão “Gerar Rede” para visualizar a rede de nós e arcos montada automaticamente. A figura 3 mostra a rede montada com os respectivos nós e arcos cadastrados pelo usuário da ferramenta. A rede é plotada em uma planilha a parte.

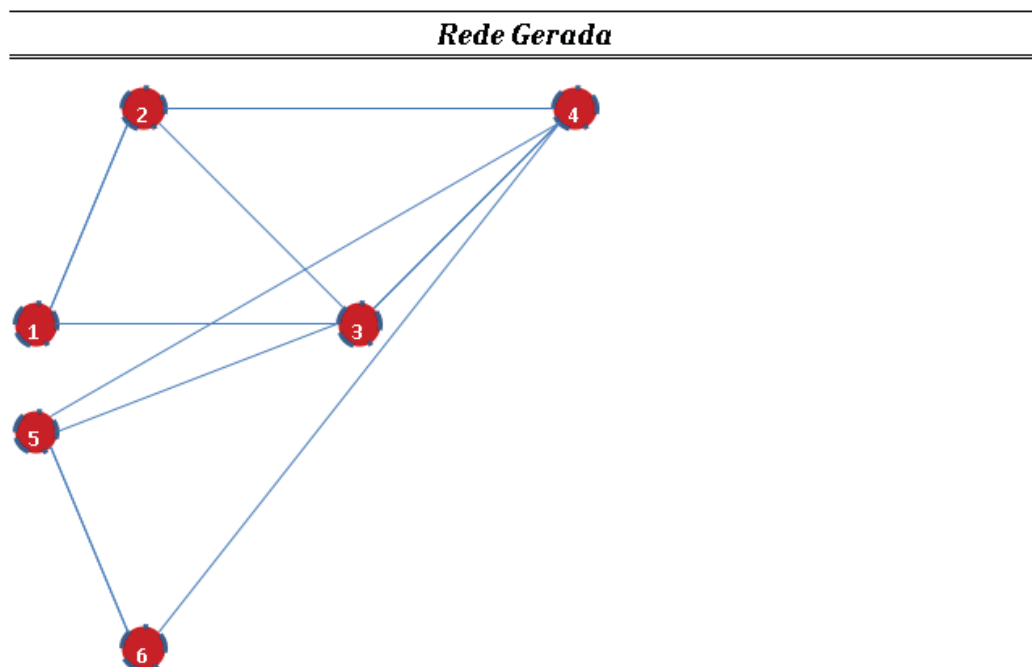


Figura 3 - Rede montada

A próxima etapa consiste em apertar o botão “Rodar Dijkstra”, que calculará o caminho mínimo entre os dois pontos selecionados e plotará na rede como na figura 4:

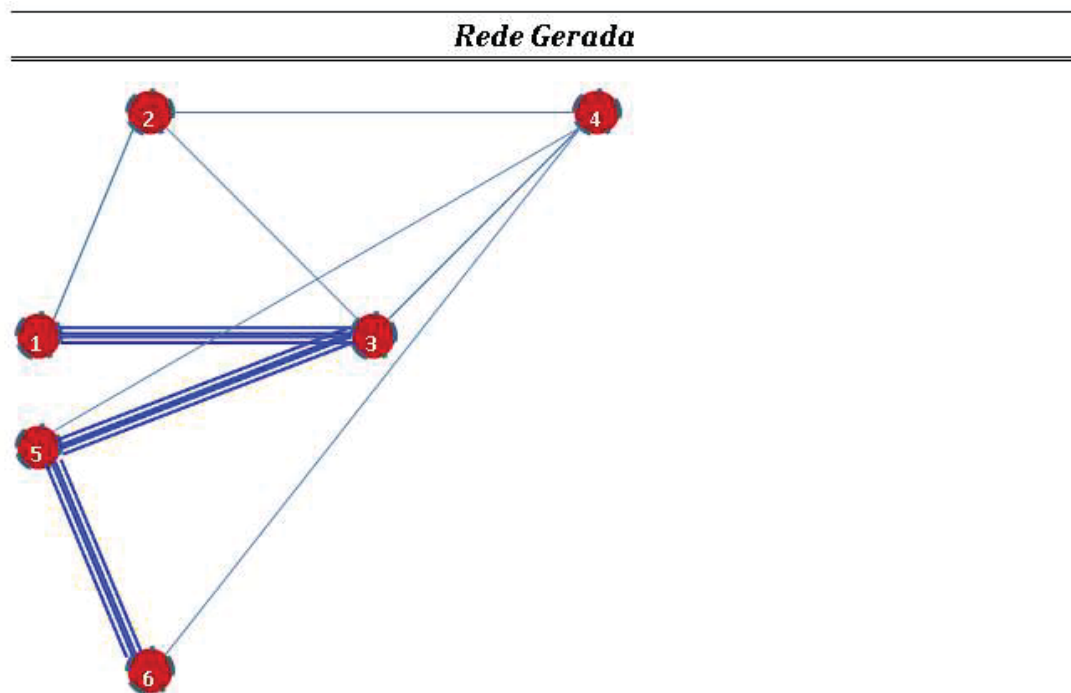


Figura 4 - Rede gerada com caminho mínimo destacado

3.2. Possibilidades de trabalho em sala de aula e pequenos projetos

A partir dos recursos gráficos oferecidos podem ser trabalhados alguns projetos com turmas de diferentes disciplinas:

- a) Disciplinas básicas de programação – Neste conjunto de disciplinas, que em geral estão posicionadas em períodos iniciais de cursos de engenharia, o professor pode trabalhar a programação utilizando um algoritmo que já possui funcionalidade prática perceptível aos alunos. O professor pode sugerir que seus alunos peguem coordenadas de algumas cidades, vejam as estradas que as conectam e sugerir que plotem na rede. Ao final podem calcular o caminho mínimo entre dois pontos quaisquer. O professor pode, também, trabalhar com os alunos a tradução do pseudocódigo em linhas de código, no caso da ferramenta, o *Visual Basic for Applications*. Os alunos podem depurar o código linha por linha e verificar como está sendo implementado.
- b) Disciplinas de Logística e Cadeias de Suprimentos – Aqui há algumas possibilidades, entre elas, comparação de cálculo de custos logísticos por uma rota qualquer e a rota ótima. Podem ser feitas comparações, por exemplo, do algoritmo de Dijkstra com outros e eventualmente melhorias na eficiência do algoritmo.
- c) Pesquisa Operacional – Aqui, provavelmente a disciplina com mais aderência à ferramenta. Mais especificamente no ensino de grafos, possibilita aos alunos resolver listas de exercícios sobre o algoritmo e verificar se fizeram da maneira correta, conferindo o gabarito pela ferramenta.

Assim, quanto às possibilidades de trabalho, é possível concluir que a ferramenta construída pode ajudar tanto alunos iniciantes quanto alunos em matérias mais avançadas de cursos de engenharia. A multidisciplinaridade e a possibilidade de mostrar aos alunos do ciclo básico as possibilidades do curso é outra vantagem importante. Ainda, o estudante poderá rodar a

ferramenta em qualquer máquina que possua o *MS Office*. Como a grande maioria das empresas possui o pacote *Office*, é possível, inclusive, que a ferramenta seja utilizada em pequenos projetos. Como o código é aberto, outros algoritmos podem ser implementados de acordo com as necessidades da empresa.

4. Conclusões

O trabalho objetivou trazer uma aplicação do algoritmo Dijkstra em plataforma Excel/VBA para facilitar o ensino da técnica aos estudantes. A vantagem da ferramenta é possibilitar a resolução de pequenas redes onde os alunos imediatamente poderão ver a aplicação e os resultados. A maneira visual de trabalhar o conteúdo serve de estímulo para gerar aprendizado e motivar o interesse do aluno a trabalhar na área. A ferramenta pode ser trabalhada em conjunto com disciplinas como Programação e Logística, contribuindo para a integração dos conteúdos. Em cada uma dessas disciplinas novos módulos podem ser criados para a ferramenta, com novas aplicações em algoritmos de roteirização, entre outros.

Referências

- Ahuja, R. K.; Magnanti, T. L. e Orlin, J. B.** Network Flows (1988).
- Biswas, S. S.; Alam, B. e Doja, M. N.** (2014). A refinement of Dijkstra's algorithm for extraction of shortest paths in generalized Real Time-Multigraphs. *Journal of Computer Science* 10, 593-603.
- Cochran, J. J.** (2012). You want them to remember? Then make it memorable! Means for enhancing operations research education, *European Journal of Operations Research*, 219, 659-670.
- Fonseca, F. F. e Pinto, L. L.** (2013). Uma ferramenta de apoio para professores e estudantes em cursos de Pesquisa Operacional, *Anais do XLV SBPO*, 824-831.
- Galán-García, J. L.; Aguilera-Venegas, G.; Galán-García, M. A. e Rodríguez-Cielos, P.** (2014). A new Probabilistic Extension of Dijkstra's Algorithm to simulate more realistic traffic flow in a smart city, *Applied Mathematics and Computation*.
- Goldbarg, M. C. e Goldbarg, E.** (2012). *Grafos: Conceitos, algoritmos e aplicações*. Rio de Janeiro: Elsevier.
- Mesquita, M. A. e Hernandez, A. E.** (2006). Discrete-Event Simulation of queues with spreadsheets: a teaching case, *Proceedings of the 2006 Winter Simulation Conference*, 2277 – 2283
- Murota, K. e Shioura, A.** (2014). Dijkstra's algorithm and L-concave function Maximization. *Math. Programming*, 163-177.
- Parra, O. J. S.; Rubio, G. L. e Daza, B. S. R.** (2014). Dijkstra Algorithm Based On A Reliability Model. *International Conference on Information Society*, 45-49.
- Pinto, L. L. e Bornstein, C. T.** (2006). Um Software Interativo para o Algoritmo Simplex em Programação Linear, *Anais do XXXVIII SBPO*, 421-431.
- Roeder, T. M. e Roeder, K. N.** (2013). An experiment in teaching Operations Management to sixth graders using simulation, *Proceedings of the 2013 Winter Simulation Conference - Simulation: Making Decisions in a Complex World, WSC*, 3600-3608.
- Sedeño-Noda, A. e Raith, A.** (2015). A Dijkstra-like method computing all extreme supported non-dominated solutions of the biobjective shortest path problem. *Computers & Operations Research* 57, 83-94.

Shaohong, Y.; Lichao, F.; Baoxiang, L. e Nan, J. (2010). Some Thoughts on "Algorithm Design and Analysis" Teaching Reform, 2nd International Conference on Industrial Mechatronics and Automation, 595-597

Uchida, K.; S. Takematsu e Honda, J. (2014). Tsunami Simulation Based on Dijkstra Algorithm. International Conference on Network-Based Information Systems, 114-119.

Wang, Y. M. e Li, Q. (2013). The Research on the Teaching Model of Operations Research, *Advanced Materials Research*, 860, 2991-2994.

Yang, Y.; He, Y.; Zheng, S. e Wang, L. (2012). Study on teaching methods of Operations Research, Proceedings of the 2012 International Conference on Cybernetics and Informatics Lecture Notes in Electrical Engineering, 1513-1517.

