

Comparação entre os Algoritmos de Busca pela Vizinhança de Von Neumann ou de Moore para Geração do Mapa de Distâncias em um Ambiente Construído*

Comparison between Search Algorithms by Von Neumann or Moore Neighborhood for Generation of the Distances Map in a Built Environment

Henrique Costa Braga¹

Gray Farias Moita²

Paulo Eduardo Maciel de Almeida³

Resumo

Existem vários algoritmos de busca na literatura, mas os poucos dedicados a simulações de grandes edificações geralmente carecem de maior detalhamento, e também de uma análise quantitativa dos erros gerados pela sua utilização na obtenção dos mapas de distância. Assim, este trabalho apresenta o passo-a-passo de um algoritmo de busca *pathfinder* especificamente apropriado para simulações em ambientes construídos, considerando duas variações do mesmo em função da vizinhança pesquisada: Von Neumann ou Moore. As duas variações do algoritmo apresentado foram computacionalmente implementadas e diversos experimentos realizados, de modo a se conhecer melhor várias de suas características como o mapa de distâncias gerado e os erros inerentes. Verificou-se que o algoritmo aqui apresentado possui diversas características importantes como simplicidade lógica, funcionamento automático, independência tanto do tamanho, quanto do leiaute interno ou externo da edificação a ser pesquisada, e baixo custo computacional para uma aplicação não dinâmica. Entretanto, a variação, considerando a vizinhança de Moore, forneceu os melhores resultados pelo menor erro na determinação das distâncias (erro médio de +5, 8% e pontual máximo de +7, 9% nos exemplos estudados).

Palavras-chave: Algoritmo de busca. Vizinhança de Von Neumann. Vizinhança de Moore. Ambiente construído. Mapa de distâncias.

*Submetido em 03/02/2016 – Aceito em 10/04/2016

¹Doutorando em Modelagem Matemática e Computacional. Programa de Pós-graduação em Modelagem Matemática e Computacional do CEFET-MG, Brasil – bragaseg@yahoo.com.br

²Doutor em Aeronáutica. Programa de Pós-graduação em Modelagem Matemática e Computacional do CEFET-MG, Brasil – gray@dppg.cefetmg.br

³Doutor em Engenharia. Programa de Pós-graduação em Modelagem Matemática e Computacional do CEFET-MG, Brasil – pema@lsi.cefetmg.br

Abstract

There are various search algorithms in the literature, but the few dedicated to large buildings simulations typically lack details and also a quantitative analysis of the errors generated by their use. So this paper presents a step- by-step description of a pathfinder search algorithm specifically suitable for simulations in built environments, considering two variations of the neighborhood of search: Von Neumann and Moore. The two variants of the presented algorithms were computationally implemented and various experiments were performed in order to better understand various characteristics such as the distance map and inherent errors. It was found that the algorithms have several important features such as logical simplicity, automatic application, independence of both size and internal or external layout of the building and low computational cost for a non-dynamic application. However, the use of the search algorithm considering the Moore's neighborhood achieved more regular and consistent results (a medium error of +5.8% and a punctual maximum error of +7.9% in the examples studied).

Keywords: Search algorithm. Von Neumann neighborhood. Moore neighborhood. Built environment. Distance maps.

1 INTRODUÇÃO

A simulação e modelagem computacional da movimentação de pessoas ou materiais em ambientes construídos são ferramentas de grande valor para melhor se conhecer e otimizar os sistemas em que estão inclusos (MA et al., 2013; BRAGA et al., 2014). São aplicáveis a áreas tão distintas como logística, pesquisa operacional, roteirização, robótica, automação, jogos (de natureza séria ou lúdica), segurança, dentre outras. Uma situação corriqueira é se determinar um melhor percurso a ser adotado por uma pessoa, considerando um posicionamento inicial qualquer no ambiente e um destino específico desejado, que pode ser tanto um setor como um almoxarifado ou depósito, uma rota de fuga ou uma saída de emergência (KO et al., 2007; RONCHI; NILSSON, 2013; KULIGOWSKI et al., 2013). Esse melhor caminho, normalmente considerado como sendo o roteiro que permite o deslocamento com o menor percurso (ANDAYESH; SADEGHPOUR, 2014), pode ser encontrado pela análise de mapas de distâncias (THOMPSON; MARCHANT, 1995).

O mapa de distâncias é uma matriz suficientemente refinada para representar o ambiente em estudo, que contém, em cada um dos seus elementos que podem servir como rota de encaminhamento, o valor da distância desse elemento a outro elemento (ou conjunto de elementos) que represente o objetivo a ser alcançado (uma saída de emergência por exemplo). Estando inicialmente em qualquer lugar do ambiente, basta selecionar aquele elemento imediatamente vizinho ao seu, que possua a menor distância indicada no mapa de distâncias, e se transferir para ele. Depois, continue pesquisando e se transferindo para aquele elemento vizinho que sempre contenha a menor distância ao objetivo. Repetindo esse processo de pesquisa aos elementos imediatamente vizinhos, e sempre se transferindo para aquele com a menor distância indicada no mapa de distâncias, em um número finito de passos irá se atingir o objetivo perpassando pela rota (ou por uma das rotas) de menor comprimento. Muitas vezes o mapa de distâncias é representado por faixas de cores (THOMPSON; MARCHANT, 1995), onde cada faixa de cor representa uma faixa de distanciamento fixa em relação ao ponto objetivo.

Os mapas de distâncias podem ser obtidos pela aplicação direta de alguns algoritmos de busca e exploração. Existe uma grande gama de algoritmos de busca e exploração, mas de modo geral algoritmos de busca podem ser considerados como estratégias de inteligência computacional para a resolução de problemas e exploração (RUSSELL; NORVIG, 2004). A classe de algoritmos de busca e exploração especificamente aqui tratada também é chamada de algoritmos *pathfinder*, pois visam encontrar ou contribuir na escolha de um melhor caminho ou roteiro geralmente entre dois pontos distintos de um ambiente (STOUT, 1996; CHUI; SHI, 2011).

Esses algoritmos de busca também podem ser incorporados em ferramentas tecnológicas de apoio ao projeto arquitetônico. As ferramentas são capazes de fornecer maior suporte ao projetista de um ambiente, incentivando sua criatividade pela liberdade e capacidade de se verificar alternativas, e o impacto quantitativo dessas na edificação (KOWALTOWSKI et al., 2010). Estudos de casos relacionados ao projeto arquitetônico envolvendo a seleção de rotas

(BOULMAKOUL; MANDAR, 2011; SILVA et al., 2012) podem se beneficiar desse tipo de algoritmo. É verdade que a realidade prática da utilização de recursos tecnológicos como esse pode, ainda, estar afastada da vivência cotidiana de vários escritórios de arquitetura, mas, certamente a divulgação e a operacionalização de estudos nessa área contribuem para diminuir essas distâncias.

Existem diversos algoritmos de busca da classe *pathfinder* já consolidados e também suas variações (HART et al., 1968; STOUT, 1996; KOENIG et al., 2004; DELLING et al., 2009). Um desses algoritmos de busca mais utilizados e estudados é o algoritmo de Dijkstra (DIKSTRA, 1959; SNIEDOVICH, 2006). O algoritmo de Dijkstra pode ser usado com sucesso, por exemplo, na seleção de melhores roteiros para tráfego de veículos. Entretanto, a aplicação direta do algoritmo de Dijkstra, para indicação de melhores rotas de movimentação em simulações de ambientes construídos, poderá resultar em grandes erros nas trajetórias indicadas em ambientes maiores (SOLTANI et al., 2002).

Portanto, não existe um único melhor algoritmo *pathfinder* (CHUI; SHI, 2011), tornando-se eventualmente necessária a adaptação de algoritmos já estabelecidos para melhor se atender a uma demanda específica. Adicionalmente, pela análise da literatura (THOMPSON; MARCHANT, 1995; LIGHTFOOT; MILNE, 2003; PELECHANO et al., 2008) fica claro que algoritmos logicamente simples que se aplicam, especificamente, a soluções para se escolher melhores rotas de movimentação em edificações carecem tanto de um maior detalhamento, como de uma melhor compreensão quantitativa dos erros produzidos pela aplicação desses.

Assim, dada a carência de detalhamento desses algoritmos para obtenção dos mapas de distâncias na literatura, e de pouca informação sobre os erros gerados, a contribuição deste trabalho é apresentar o passo-a-passo de um algoritmo de busca e exploração da classe *pathfinder* em duas variações de vizinhanças diretas de pesquisa especialmente adequadas para gerar o mapa de distâncias de um ambiente construído, assim como identificar, quantitativamente, os erros resultantes de suas aplicações.

Essas vizinhanças são comumente denominadas de vizinhança de Von Neumann e vizinhança de Moore (WOLFRAM, 1983). Para ambas as variações são realizadas diversas simulações e os valores obtidos são criticamente comparados. São considerados os valores do custo de movimentação (distância a ser percorrida) entre pontos de um ambiente com a geração do respectivo mapa de distâncias. Desse modo, se tem um algoritmo detalhado e o conhecimento de seu comportamento, da influência que suas variações de vizinhanças de pesquisa têm sobre o resultado final e, quantitativamente, os erros passíveis de serem gerados pela sua aplicação.

Este trabalho está organizado na seguinte ordem: esta Seção 1 contempla a introdução com a motivação e o escopo do trabalho; a Seção 2 aborda a descrição e o detalhamento do algoritmo de busca com suas variações de vizinhanças de pesquisa; a Seção 3 apresenta resultados de simulações computacionais, permitindo conhecer o desempenho do algoritmo de busca e comparar suas variações; a Seção 4 discute alguns pontos do algoritmo e os resultados obtidos anteriormente e na Seção 5 são listadas as principais conclusões encontradas.

2 DESENVOLVIMENTO

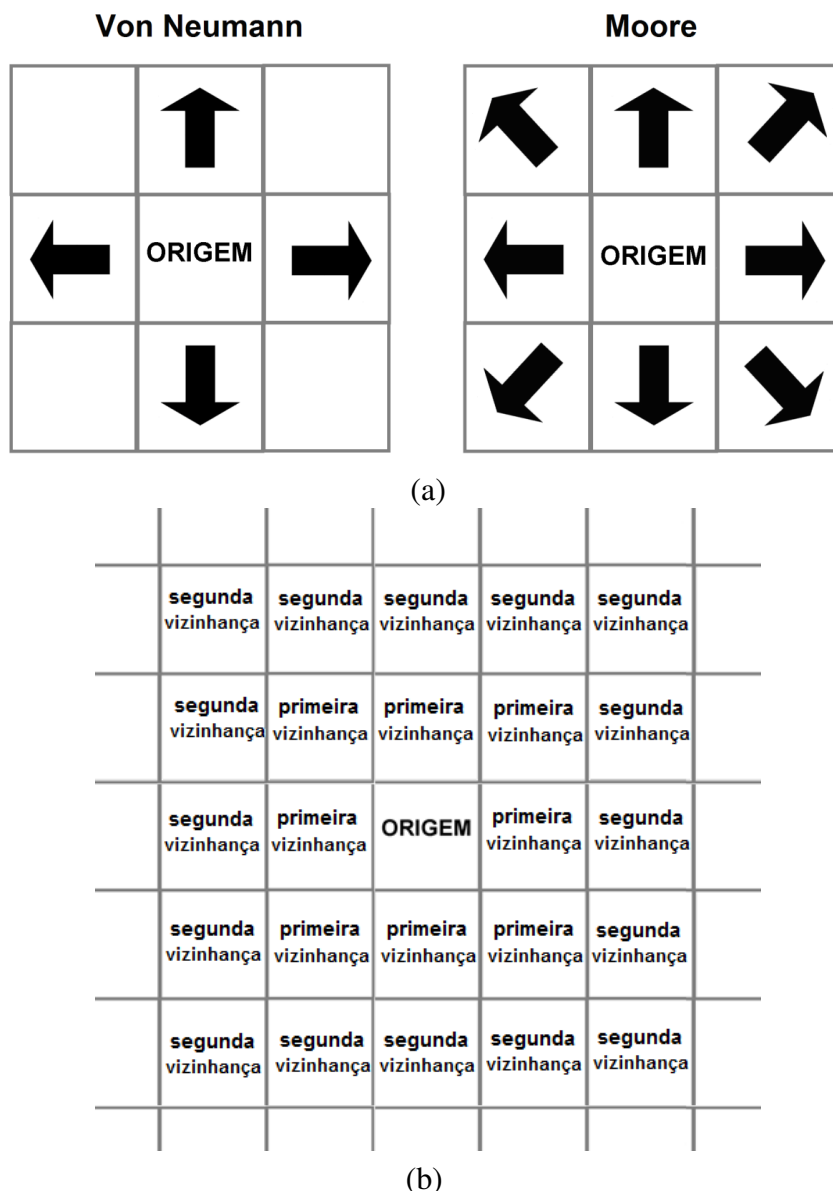
Aqui é detalhada, passo-a-passo, toda a parte conceitual do algoritmo para ambas as variações de vizinhança de busca. Não é objetivo apresentar um modelo de código acabado, mas sim, descrever cuidadosamente toda a lógica envolvida, de modo a possibilitar que um desenvolvedor interessado possa elaborar por si só o seu próprio programa.

2.1 Vizinhanças de busca

Na busca ou movimentação pela primeira vizinhança de Von Neumann (VVN), a partir de um ponto do ambiente, é permitido se pesquisar diretamente em apenas quatro direções (considerando uma vista de topo, estas direções são para cima, para baixo, para a direita e para a esquerda). Já na busca ou movimentação pela primeira vizinhança de Moore (VM), é permitido, além dos movimentos diretos já citados, movimentações e pesquisas também nas diagonais.

A Figura 1a ilustra essas duas possibilidades de movimentações a partir de uma vista de topo da primeira vizinhança no ambiente discretizado de uma casa qualquer de origem, também chamada casa base de varredura. A Figura 1b identifica as células que compõem a primeira e a segunda vizinhança genérica de busca em relação à casa base (origem). Neste trabalho, para todas as variações do algoritmo, apenas a primeira vizinhança de uma casa base será considerada a cada iteração do algoritmo de busca.

Figura 1 – Em (a), as possibilidades de pesquisa pela vizinhança de Von Neumann (VVN) e de Moore (VM) e, em (b), a representação da primeira e segunda vizinhança em relação à origem (ou casa base)



Fonte: Elaborada pelos autores

2.2 Discretização do ambiente

Inicialmente, o ambiente construído a ser simulado deve, em sua vista de topo, ser discretizado na forma de uma matriz retangular bidimensional denominada matriz ambiente (MA). Na matriz MA, o índice das respectivas linhas e colunas representa um determinado posicionamento no mundo real. A distância no mundo real entre duas linhas ou colunas consecutivas é denominada resolução (r). Quanto menor o valor de r , mais refinados tendem a ser os resultados da simulação, mas, maior será também o custo computacional associado.

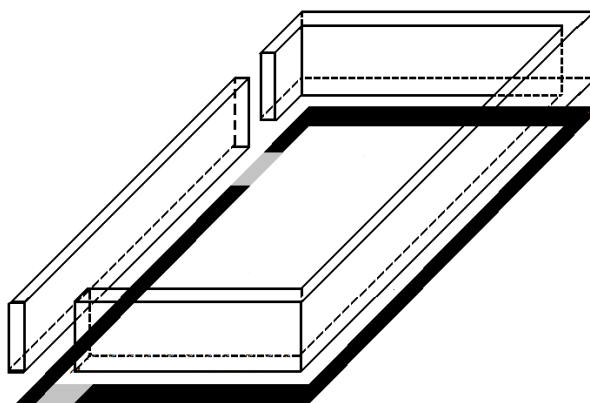
Adicionalmente, os valores contidos nos elementos de MA estão diretamente associados a um significado no mundo real. Por exemplo, o valor “0” em MA indica que esse respectivo

elemento no mundo real está vazio e pode ser usado como passagem; o valor “-1” em MA indica que este pertence a uma parede; o valor “-2” em MA indica que este elemento pertence a um dos objetivos da simulação (uma saída ou sala específica), e o valor “-3” em MA indica que este elemento representa uma área externa ao ambiente.

Uma vez discretizado o ambiente com a geração da matriz MA, é, então, criada uma matriz linha, denominada Matriz de Busca Primária (MBP), contendo em cada elemento, o posicionamento (índices) de todos os elementos de MA pertencentes ao objetivo (ou que contém o valor “-2”).

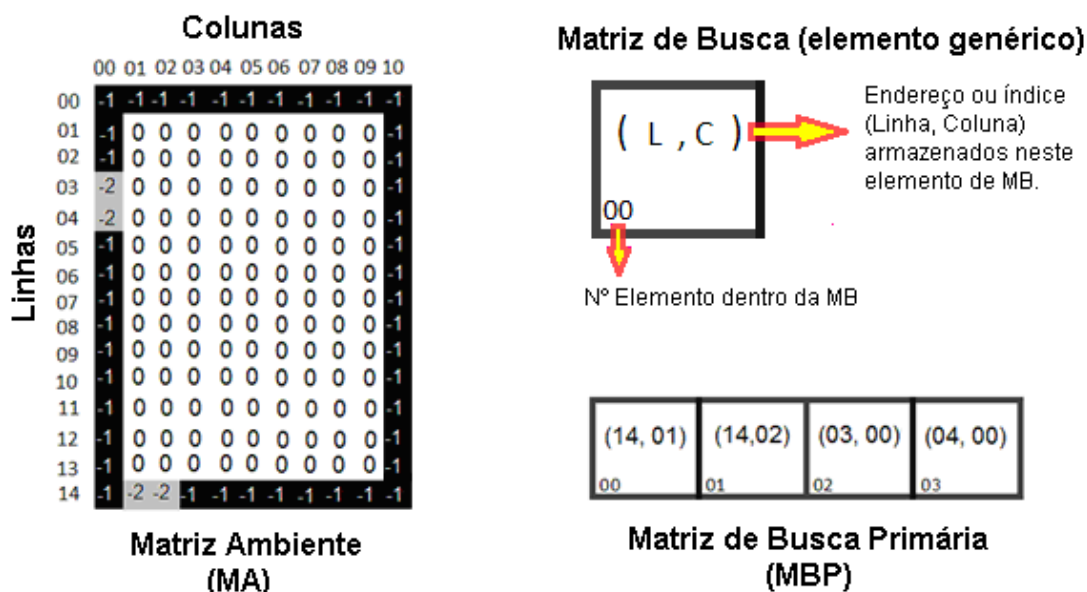
A Figura 2 apresenta uma representação em perspectiva de um ambiente construído na forma retangular com duas saídas distintas e a Figura 3 apresenta as respectivas matrizes MA e MBP iniciais obtidas para o ambiente da Figura 2.

Figura 2 – Representação de um ambiente construído em perspectiva (sem escala)



Fonte: Elaborada pelos autores

Figura 3 – Matrizes MA (a) e MBP iniciais obtidas (b) para o ambiente da Figura 2



Fonte: Elaborada pelos autores

Depois de criada a MBP, é também gerada uma matriz com as mesmas dimensões da

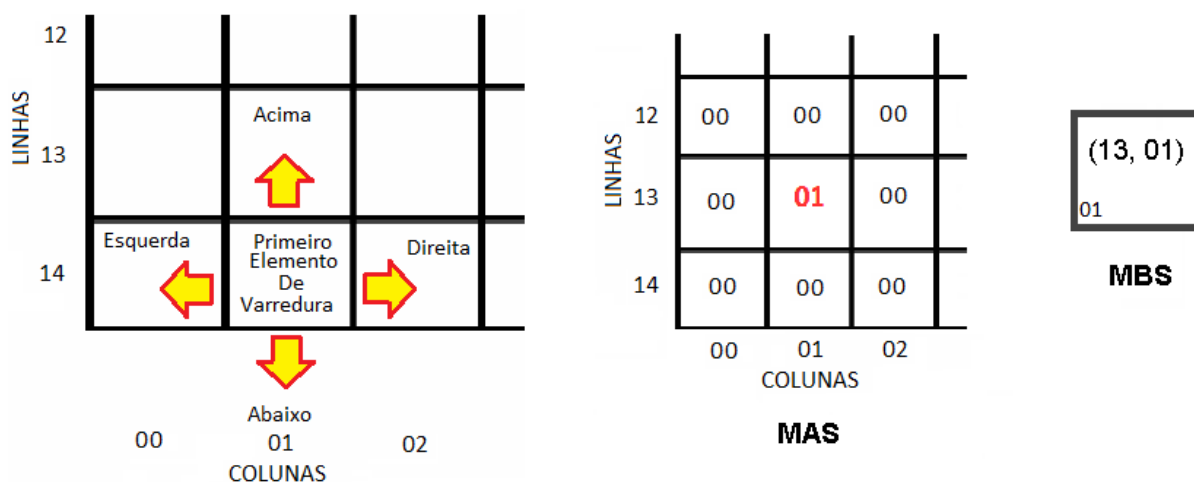
matriz MA, denominada matriz ambiente secundário (MAS), mas com todos os seus elementos contendo inicialmente o valor “0”.

2.3 Algoritmo de busca segundo a vizinhança de Von Neumann (VVN)

Finda a discretização, o algoritmo de busca se inicia pela pesquisa de cada elemento de MBP. A ordem da pesquisa em MBP não é relevante. Considerando o primeiro elemento de MBP indicado na Figura 3b, ele contém o índice base de busca inicial (linha 14, coluna 01), ou seja, esse é um endereço que, em MA, representa um objetivo. Todos os elementos de MA (ou em MAS), que são os primeiros vizinhos de Von Neumann (VVN) desse índice base, estão a uma distância “1” desse objetivo. Assim, todos esses elementos vizinhos em MAS terão o seu respectivo valor alterado pelo valor de MAS da casa base ou origem incrementado em “1” (como neste caso é a primeira busca o valor passará de “0” para “1”).

Entretanto, esse incremento no valor de MAS não vai ocorrer caso esse elemento vizinho do índice base pertença a uma parede, a um outro objetivo; ou, obviamente, esteja fora do ambiente (a matriz MA original é usada para esta verificação). Quando um elemento de MAS tem seu valor alterado, o índice desse elemento alterado é adicionado em uma nova matriz linha, denominada Matriz de Busca Secundária (MBS). Na Figura 4, é mostrada a representação desse primeiro passo para o ambiente da Figura 2. A matriz MA inicialmente criada nunca será alterada.

Figura 4 – Em (a), a representação da primeira vizinhança de busca em MA e, em (b) e em (c), as respectivas novas matrizes MAS (visão parcial) e MBS formadas após esta primeira busca pela VVN do primeiro elemento base de MBP



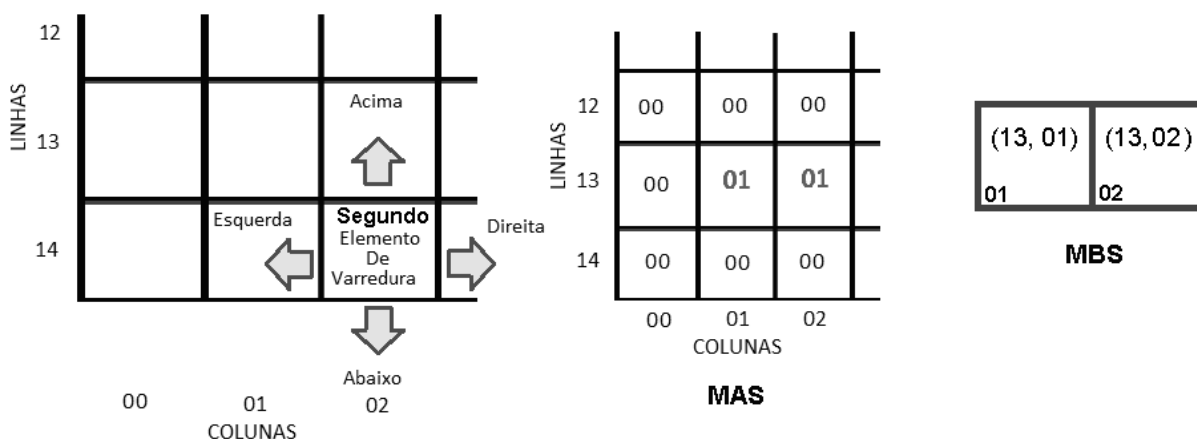
Fonte: Elaborada pelos autores

Terminada a busca sobre os quatro primeiros vizinhos pela VVN do primeiro índice base da matriz MBP, a busca vai prosseguir sobre o próximo elemento de MBP, no caso (14, 02) ou linha 14 e coluna 02 em MA. Repetindo o raciocínio, todos os elementos em MAS, que são os primeiros vizinhos de Von Neumann desse novo elemento base, terão o seu valor de MAS

anterior substituído pelo valor de MAS do novo elemento base incrementado em “1” (exceto, novamente, os pertencentes a paredes, objetivos ou externos).

Todos os endereços de MAS que sofreram alterações no seu valor são adicionados na nova matriz linha MBS em formação, cujo tamanho vai sendo incrementado com a busca. Na Figura 5, pode se ver a representação desse segundo passo. Ressalta-se que, caso algum novo valor de MAS seja eventualmente igual ou superior a um valor não nulo de MAS já existente no elemento vizinho que estiver sendo verificado, essa substituição de valores não vai se concretizar, e esse índice não será adicionado à matriz MBS em formação.

Figura 5 – Em (a), a representação da busca em MA considerando o segundo elemento de varredura de MBP e, em (b) e em (c), as respectivas novas matrizes MAS (visão parcial) e MBS em formação



Fonte: Elaborada pelos autores

Finda essa nova busca sobre os quatro primeiros vizinhos do segundo índice base em MBP, a busca vai prosseguir sobre o terceiro elemento de MBP, e assim, sucessivamente, até que todos os elementos de MBP tenham sido verificados. Terminada essa fase, a MBP original é apagada e uma nova MBP contendo o conteúdo de toda a matriz final MBS é criada. Após essa transferência e substituição, a matriz MBS é feita nula. Nesse momento, se tem uma nova MBP, com os endereços dos novos elementos base de busca em MA, uma nova MBS nula e uma matriz MAS em formação.

Uma nova rodada de pesquisa sobre a atual MBP é então realizada, com a contínua formação da matriz MAS e, assim, sucessivamente. O processo irá terminar quando, após realizar uma verificação integral de uma MBP, não ocorrer nenhuma alteração em qualquer valor de MAS, ou seja, a matriz final MBS da rodada antes da sua transferência para a MBP seja nula. Nesse momento, todo o ambiente foi integralmente pesquisado.

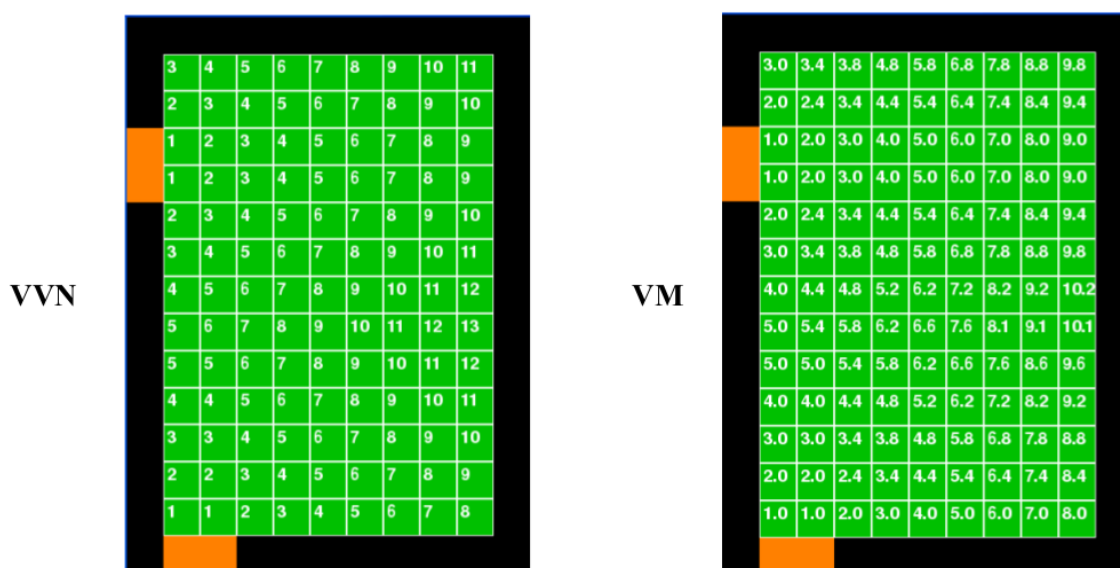
Para finalizar, todos os valores negativos na matriz MA são transferidos, na mesma posição em que se encontram em MA, para a matriz MAS. A matriz MAS então formada contém, em todos os seus elementos não nulos, o respectivo número de elementos (ou distância) que esse elemento está da saída ou objetivo desejado mais próximo, segundo a VVN, além de ter identificado com o valor “0” os objetivos (ou saída), com “-1” as paredes ou obstáculos e, caso exista em MA, com “-2” a parte externa do ambiente.

2.4 Algoritmo de busca segundo a vizinhaça de Moore (VM)

Essa versão do algoritmo é similar à anterior em todos os aspectos. A diferença é que a vizinhaça considerada do elemento base identificada na matriz MBP; neste caso, será toda a primeira vizinhaça, inclusive os primeiros vizinhos em diagonal ao elemento da casa base de varredura. Quando a vizinhaça considerada for a dos elementos acima, abaixo, à direita e à esquerda, o valor substituído como o novo eventual valor de MAS continuará sendo “1” acrescido pelo valor de MAS do elemento de origem. Mas, se o vizinho estiver numa das diagonais, o valor a ser substituído como sendo o novo eventual valor de MAS será “ $\sqrt{2}$ ” (considerado como 1,4142) acrescido pelo valor de MAS do elemento de origem.

A Figura 6 apresenta, para o ambiente construído da Figura 2, as respectivas matrizes MAS finais estilizadas obtidas com as duas variações do algoritmo de busca (VVN e VM). Na Figura 6, as distâncias pontuais (Dp) em número de elementos entre o objetivo mais próximo e cada uma das células do ambiente discretizado estão indicadas sobre os mesmos. Ainda na Figura 6, as paredes do ambiente estão representadas em preto e os objetivos em laranja, sem indicação de distâncias.

Figura 6 – Representação das matrizes MAS finais estilizadas obtidas para o ambiente da Figura 2 com as distâncias (em número de elementos) de qualquer elemento em seu interior e o objetivo mais próximo



Fonte: Elaborada pelos autores

3 RESULTADOS

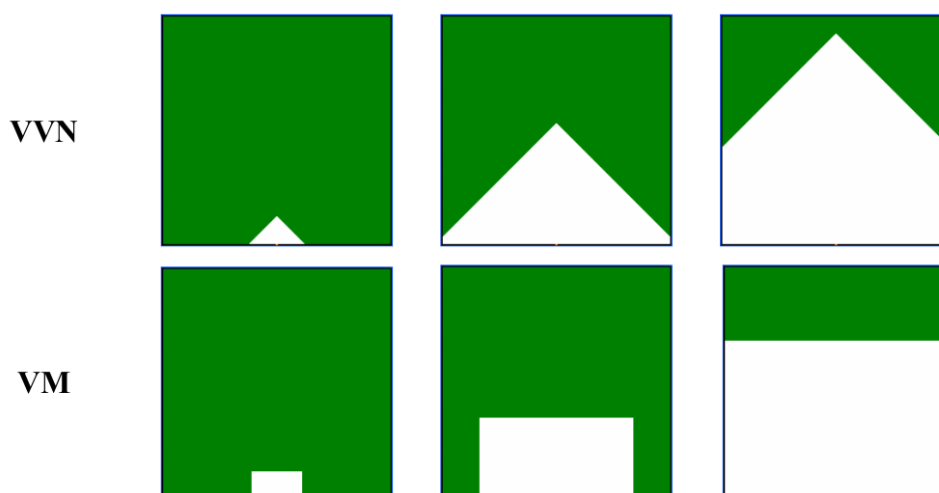
As variações dos algoritmos aqui apresentados foram implementadas computacionalmente em um computador pessoal (sistema operacional Microsoft Windows XP versão 2002, 32 bits, processador Intel Pentium Dual Core, 2 GHz de processamento, 3,24 GB de RAM) na

linguagem Python 3.4.

Inicialmente, foram simulados ambientes em forma de simples quadrados, sem separações internas, com paredes apenas externas de um elemento de espessura e um único objetivo (saída) de dois elementos de largura posicionado no centro de um dos lados. Esses ambientes quadrados têm largura externa variando de 10 elementos a 1.000 elementos (ou com área formada por 100 a 1.000.000 de células) de modo a cobrir uma grande faixa de áreas podendo ser representativo para a maioria das edificações.

A Figura 7 apresenta imagens de momentos da simulação para ambas as variações do algoritmo, indicando o perfil de sua evolução no ambiente quadrado com 1000 elementos de lado. Em verde, está a região ainda não verificada pelo algoritmo e, na cor branca, a região já pesquisada. Inicialmente, toda a parte interior do ambiente está em verde. Com o andamento da simulação, a parte já analisada do ambiente (inicialmente vizinha à saída ou objetivo que se encontra no centro do lado inferior) muda sua cor para branco. A região em branco cresce com a evolução do algoritmo, até que todo o interior do ambiente fique na cor branca, indicando que todo o ambiente foi analisado (a matriz MAS final foi obtida).

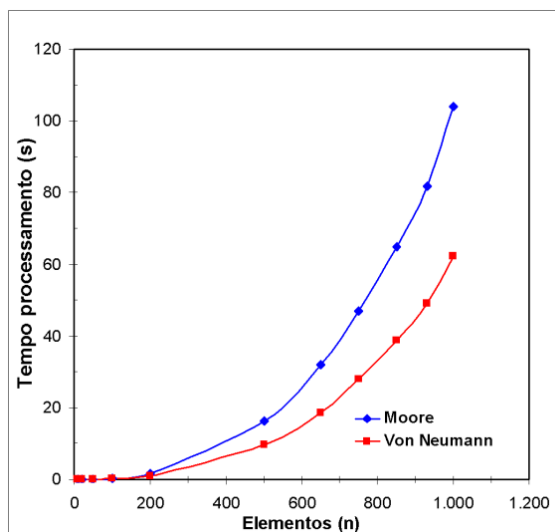
Figura 7 – Perfil da evolução do algoritmo de busca sobre o ambiente quadrado com 1000 elementos de lado, onde a cor branca identifica a área já pesquisada e a saída está localizada no centro do lado inferior



Fonte: Elaborada pelos autores

A Figura 8 mostra a variação do custo computacional (em tempo de processamento informado por uma rotina introduzida no próprio programa, estando o uso da CPU em torno de 50%) para as simulações dos ambientes quadrados com as duas variações do algoritmo de busca. Baseado na Figura 8, pode-se inferir informações sobre a complexidade apresentada pelo algoritmo.

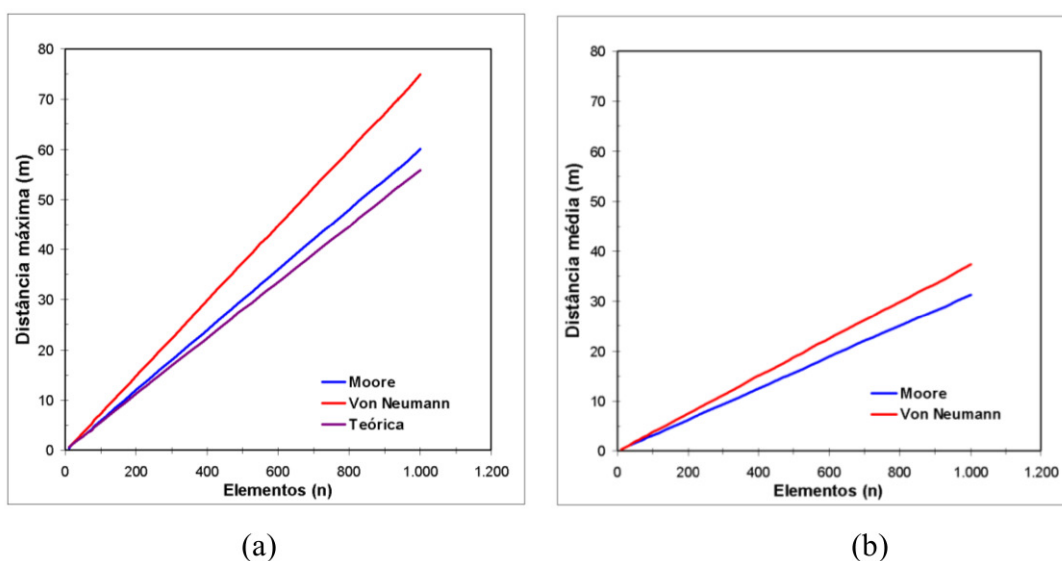
Figura 8 – Tempo de processamento em função do número de elementos de cada lado para todas as simulações dos ambientes quadrados e para as duas variações do algoritmo de busca



Fonte: Elaborada pelos autores

Adicionalmente, considerando uma resolução (r) de 5 cm, foi obtido o valor da maior distância entre um ponto do ambiente e o objetivo ($D_{\text{máx}}$) e o valor da distância média entre todos os pontos do ambiente e o objetivo mais próximo ($D_{\text{méd}}$). O valor teórico para o $D_{\text{máx}}$, obtido pela distância euclidiana contínua do ponto mais afastado da saída independente de qualquer discretização, foi manualmente calculado para cada simulação para fins de comparação. Na Figura 9, são apresentados os resultados obtidos de $D_{\text{máx}}$ e $D_{\text{méd}}$ para ambas as variações do algoritmo para os ambientes quadrados com até 1000 elementos de lado.

Figura 9 – A variação da distância máxima (a) e da distância média (b) para todas as simulações dos ambientes quadrados em função do número de elementos de cada lado

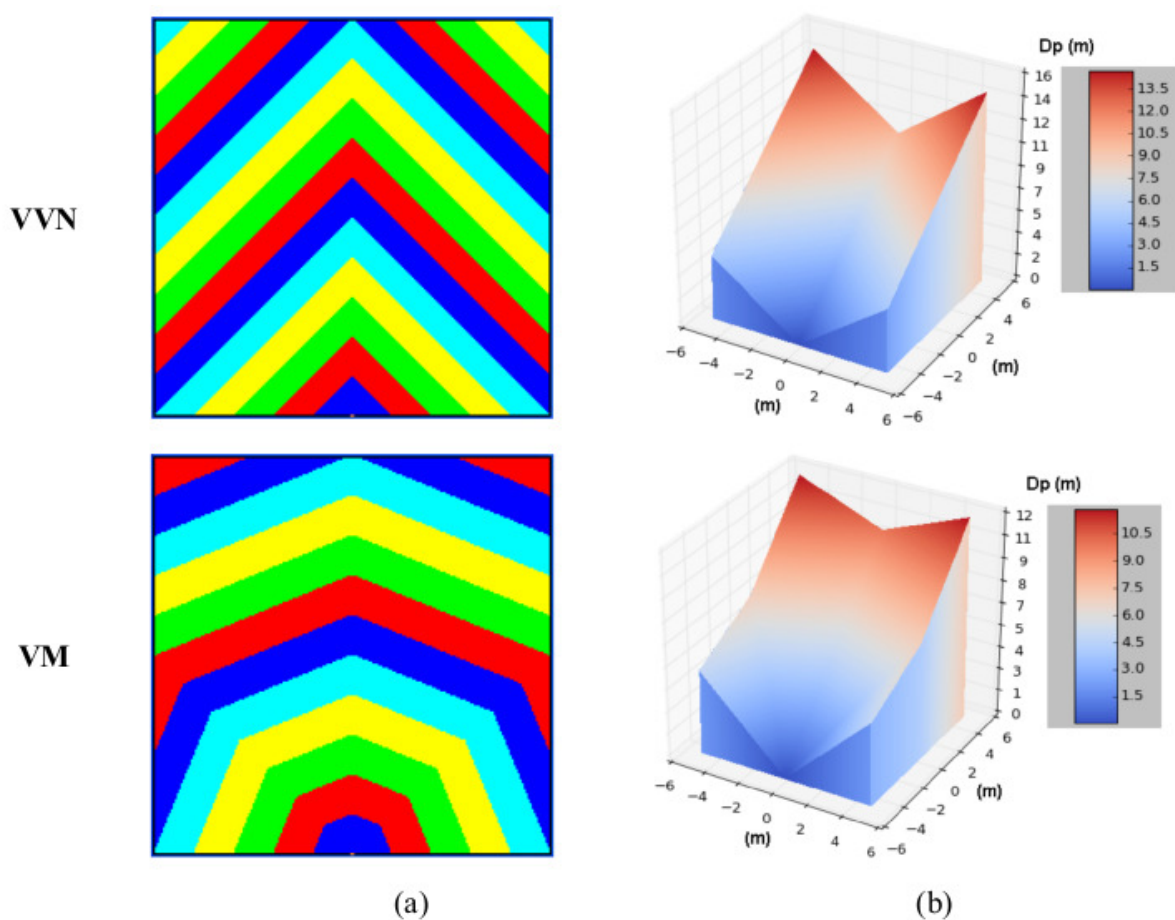


Fonte: Elaborada pelos autores

Também foi gerado para cada simulação dos ambientes quadrados o respectivo mapa

de distâncias. Complementando-os, foram elaborados gráficos de distância em três dimensões (3D), mostrando o valor de D_p em função de sua posição no ambiente. Como exemplo, a Figura 10a apresenta o mapa de distâncias em faixas de cores; e a Figura 10b, o respectivo gráfico de D_p em 3D para as simulações do ambiente quadrado com nº de 200 elementos de lado (ou 100 m² de área) e para as duas variações do algoritmo de busca. Nessa Figura 10a, as diferentes cores de faixas no mapa de distâncias indicam um distanciamento de 1 m entre si, estando o objetivo (ou saída) localizado no centro do lado inferior do ambiente. Na Figura 10b, o valor de D_p é mostrado em função da sua posição no ambiente, onde o ponto (0,0) indica o centro do ambiente.

Figura 10 – Representações do mapa de distâncias para as variações do algoritmo de busca, em faixas (a) e em 3D (b), do ambiente quadrado de 200 elementos de lado (ou 100 m² de área com r de 5 cm)



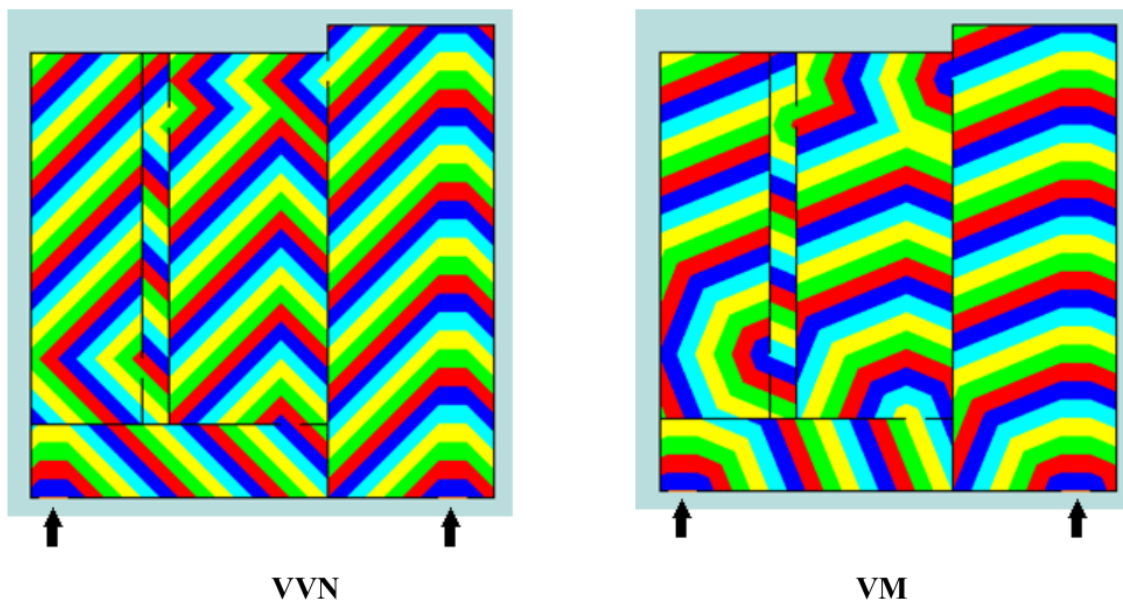
Fonte: Elaborada pelos autores

O algoritmo de busca como detalhado, pela sua lógica de funcionamento e por todos os testes realizados, aparenta poder ser diretamente usado (sem interferência humana) em ambientes com qualquer grau de sofisticação no seu leiaute, tanto interna quanto externamente, de qualquer tamanho, desde que de um único pavimento. Como exemplos, se apresentam, na Figura 11, o mapa de distâncias para simulações pelas duas variações do algoritmo de busca sobre um ambiente mais sofisticado.

A Figura 11 apresenta um ambiente irregular com aproximadamente 620 m² de área,

r de 5 cm, duas saídas (indicadas pelas setas) e várias salas internas (paredes horizontais e verticais em preto). Na Figura 11, cada uma das tonalidades de cor das faixas internas equivale a 1 m de distanciamento da saída (em laranja) mais próxima e a região em volta do ambiente, em azul claro, representa uma região externa a este ambiente.

Figura 11 – Mapa de distâncias para as simulações de um ambiente irregular ($\approx 620 \text{ m}^2$ de área), várias salas internas e dois objetivos (duas saídas distintas indicadas pelas setas)



Fonte: Elaborada pelos autores

4 DISCUSSÕES

O algoritmo apresentado, diferentemente de algoritmos como o de Dijkstra, não é do tipo guloso, pois todo o ambiente é sempre pesquisado, não havendo nunca alguma área eventualmente não verificada. Não há possibilidade de algum outro melhor percurso ter passado despercebido. Essa é uma característica positiva para que o algoritmo possa ser usado na simulação de ambientes construídos genéricos com uma população não previamente posicionada, permitindo total liberdade nas simulações do projetista. Sempre será gerado o melhor mapa de distâncias para todo o ambiente, havendo somente o erro inerente à discretização e ao formato da pesquisa.

O custo computacional para se pesquisar todo o ambiente é compensado pela necessidade da pesquisa ser feita uma única vez para cada edificação, não importando quantas simulações serão posteriormente, realizadas e nem como as mesmas sejam planejadas, desde que não se altere a parte física da edificação (a matriz MAS é gerada uma única vez). Obviamente, se o leiaute da edificação for modificado, uma nova matriz MAS deve ser gerada antes de se realizar novas simulações. Assim o algoritmo descrito neste trabalho é especialmente apropriado para ser usado de modo não concomitante com a simulação, mas sim, previamente à mesma, onde o

mapa de distâncias é então salvo na memória para ser utilizado sempre que necessário. Entretanto, se ressalta que não há nenhum impedimento, exceto pelo tempo de processamento, para seu uso também dinâmico.

Existem algoritmos de busca aplicados a simulações no ambiente construído que precisam de alguma interferência humana na obtenção de bons resultados (PELECHANO et al., 2008). Entretanto, o algoritmo apresentado é totalmente automático. Uma vez discretizado o ambiente, nenhuma interferência humana é necessária para a obtenção da matriz MAS ótima.

Muitos dos principais algoritmos de busca são do tipo em profundidade, o que também não é o caso deste. Neste algoritmo, a busca se dá inicialmente em largura e, somente após toda a primeira vizinhança considerada (VVN ou VM) de todos os elementos objetivos terem sido pesquisadas, é que se evolui em profundidade. A busca do algoritmo evolui a partir do(s) objetivo(s), de modo conjunto para o restante do ambiente. Por isso, na Figura 7, a área em branco (indicando a região pesquisada) se inicia no entorno dos objetivos (saída) e vai evoluindo paulatinamente até abranger todo o ambiente.

Observa-se também, na Figura 7, que a evolução da variação do algoritmo segundo VVN sobre o ambiente construído sendo pesquisado apresenta uma forma macroscópica triangular, ao passo que a evolução da variação do algoritmo segundo VM apresenta uma forma macroscópica retangular. Essas diferentes formas visualizadas de evolução do algoritmo sobre o ambiente são plenamente coerentes com o formato de busca das células vizinhas para cada uma das variações dos algoritmos (Figura 1a).

Ressalta-se que a pesquisa sobre a matriz MBP pode ser feita de modo sequencial simples, nem mesmo importando se a matriz MBP é tratada como uma pilha, ou como uma fila.

Desconsiderando os valores absolutos indicados para os tempos de processamento, pois esses dependem da arquitetura e da configuração computacional particular das simulações (*hardware* e *software*), na Figura 8, se observa que a variação do algoritmo segundo VM apresenta um tempo de processamento cerca de 64% superior ao tempo de processamento obtido para a variação segundo VVN. Considerando que, a cada iteração, a variação do algoritmo segundo VM pesquisa 8 casas vizinhas à base, enquanto a variação VVN pesquisa apenas 4 casas vizinhas à base (Figura 1a), esse aumento no tempo de processamento já era esperado.

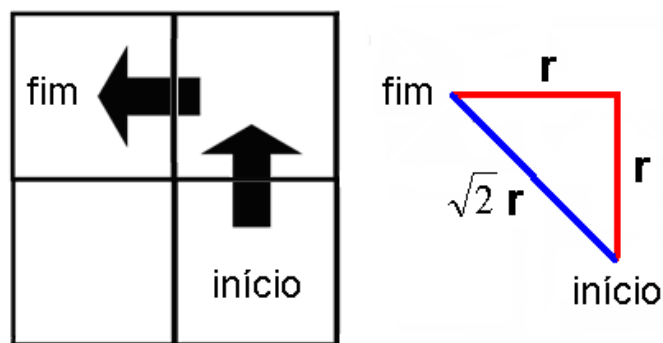
Na Figura 9, se observa que as variações do algoritmo apresentam erro em relação à determinação da distância máxima teórica, sendo o erro pela VVN superior ao erro obtido pelo VM. Esses erros estão associados à discretização do ambiente, e, principalmente, à forma de pesquisa e movimentação sobre o ambiente discretizado. O valor da distância média obtido pela variação VVN também foi superior ao obtido pela aplicação da variação VM.

Ressalva-se que, caso o agente que irá percorrer a rota a ser gerada pelo mapa de distâncias apenas realizar movimentações em ângulos de 90° (não se locomova nas diagonais), a utilização da variação VM não se aplica e somente deve ser usado o algoritmo pela variação VVN.

O gráfico com o mapa de distâncias (Figura 10) permite conhecer o perfil de evolução das distâncias em relação ao objetivo em cada ponto do ambiente. O algoritmo conforme a VVN

do algoritmo é similar (Figura 1a), mas, nas diagonais, existe grande diferença, já que, pelo algoritmo, segundo VVN, é necessária a movimentação de duas casas de distância (r) para se locomover, na verdade, uma distância de apenas $\sqrt{2}r$ (Figura 13). Independente da variação do algoritmo utilizado, o erro, quando não nulo, será sempre positivo, ou seja, o mapa de distâncias sempre indicará valores iguais ou superiores aos reais.

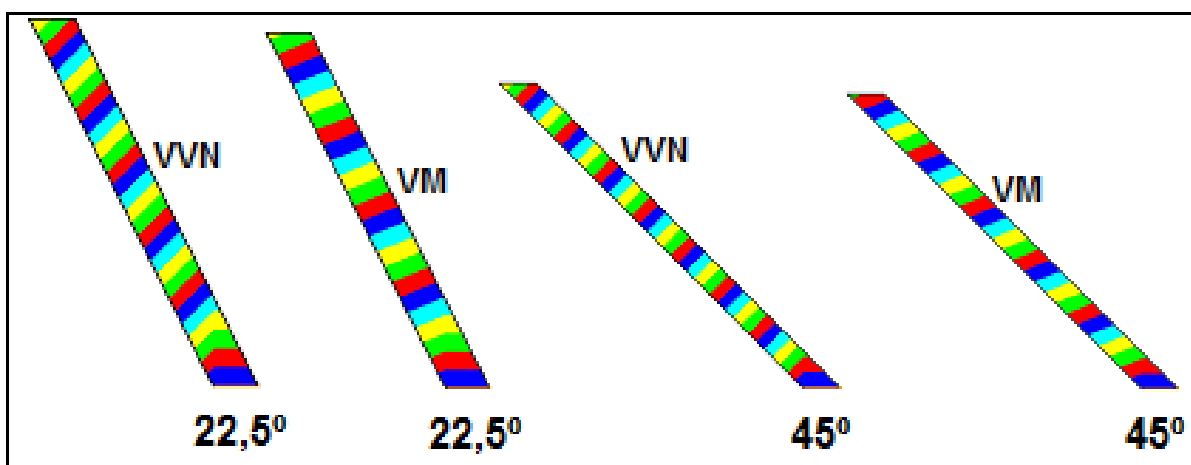
Figura 13 – Representação esquemática da movimentação entre casas diagonalmente vizinhas



Fonte: Elaborada pelos autores

Na Figura 12, pela identificação das regiões onde os perfis de distância das variações do algoritmo se aproximam (ou se afastam), pode-se notar que a pior situação (maior erro entre o mapa de distância gerado e o real) ocorre em ambientes cuja discretização esteja a múltiplos de 45° e não múltiplos de 90° do posicionamento desse ambiente para o algoritmo, conforme VVN, e para múltiplos de $22,5^\circ$ e não múltiplos de 45° para o algoritmo conforme VM. Para melhor ilustrar estes casos, na Figura 14, são mostrados os mapas de distância em ambientes, exatamente, com essa característica (defasados em $22,5^\circ$ e 45° na matriz MA).

Figura 14 – Representação dos mapas de distância para ambientes cuja geometria faz com a matriz MA um ângulo de $22,5^\circ$ ou de 45° para ambas variações do algoritmo (saída em toda a face inferior)



Fonte: Elaborada pelos autores

Na Tabela 1, são mostradas as diferenças percentuais entre os valores obtidos pelo mapa de distâncias e os ideais para os ambientes da Figura 14 e também, para comparação, os valores

obtidos para os ambientes quadrados sem repartições internas com uma saída pontual central utilizados para a geração da Figura 9.

Os resultados da Tabela 1 são muito favoráveis à utilização da variação segundo VM em detrimento da variação pela VVN. Esse erro de até quase + 42% na determinação da distância máxima indica que o algoritmo segundo VVN, simplesmente, não deve ser utilizado diretamente em simulações e modelagens mais rigorosas, pelo menos, não sem algum algoritmo adicional de suporte para corrigir ou minimizar esse efeito.

Tabela 1 – Erros obtidos na determinação da distância máxima (Dmáx) e da distância média (Dméd) em relação ao valor ideal teórico para os ambientes da Figura 14 e para um ambiente quadrado

Erro (%)	Ambiente a 22,5°		Ambiente a 45°		Ambiente quadrado (saída central)	
Distância	VVN	VM	VVN	VM	VVN	VM
Dmáx	+ 34,3	+ 7,9	+ 41,8	0	+ 33,8	+ 7,7
Dméd	+ 28,6	+ 7,3	+ 39,7	+ 0,8	+ 26,1	+ 5,7

Fonte: Elaborada pelos autores

Já a utilização da variação do algoritmo segundo VM irá resultar num erro máximo de +7,9% na determinação das distâncias, mas esse erro pontual é dependente da geometria do ambiente e da posição considerada. Ainda para a variação do algoritmo segundo VM, para o ambiente a 45°, o erro médio encontrado foi de apenas +0,8% e nulo para a determinação da distância máxima. Para o ambiente quadrado, o erro médio foi de +5,7%.

5 CONSIDERAÇÕES FINAIS

Neste artigo, foi detalhado o passo-a-passo da lógica por trás de um algoritmo de busca da classe *pathfinder*, onde são gerados mapas de distâncias em um ambiente construído, permitindo sua aplicação posterior como suporte para simulações e modelagens das mais diversas, envolvendo a movimentação humana ou de materiais em um ambiente construído.

O algoritmo foi apresentado considerando a busca sobre duas variações de vizinhanças de pesquisa, ora pela primeira vizinhança de Von Neumann (VVN), ora pela primeira vizinhança de Moore (VM). Ambas as variações apresentadas foram implementadas computacionalmente e apresentaram as seguintes características positivas: grande capacidade de generalização (permite a simulação de ambientes com qualquer leiaute interno ou externo, desde que de um único pavimento), baixo tempo de processamento (considerando uma aplicação não dinâmica) mesmo em grandes ambientes (foram simulados ambientes formados por até 1.000.000 de células), simplicidade lógica, capacidade de produzir resultados com certa similaridade com o teórico e total automação na sua aplicação.

Observou-se, no experimento, que o valor do erro máximo no percurso em termos de distância percorrida é dependente da geometria do ambiente e da posição considerada. Verificou-se, também, nos experimentos conduzidos, que a variação do algoritmo de busca pela VVN

pode apresentar um erro pontual entre 0% a até + 41,8% em relação aos valores reais teóricos de distância, enquanto a variação do algoritmo pela VM, o erro pontual pode ser entre 0% a até + 7,9. Esses são comportamentos favoráveis ao uso da variação do algoritmo pela vizinhança de Moore em detrimento do uso somente da vizinhança de Von Neumann para geração dos mapas de distâncias.

Como próxima etapa para este trabalho, propõe-se adaptar o algoritmo de busca aqui apresentado para gerar mapas de distâncias também em ambientes de múltiplos andares, passando por escadas e rampas. Essa é uma melhoria importante para que este algoritmo possa ser mais amplamente utilizado, pois muitas das edificações mais relevantes possuem mais de um pavimento.

AGRADECIMENTOS

Os autores agradecem a CAPES por todo o apoio recebido.

REFERÊNCIAS

- ANDAYESH, M.; SADEGHPOUR, F. A. Comparative Study of Different Approaches for Finding the Shortest Path on Construction Sites. **Procedia Engineering**, v. 85, p. 33–41, 2014.
- BOULMAKOUL, A.; MANDAR, M. Fuzzy Ant Colony for Virtual Pedestrian Simulation. **The Open Operational Research Journal**, v. 5, p. 19–29, maio 2011.
- BRAGA, H. C. et al. Simulação da Movimentação de Pessoas em Situações de Emergência: aspectos ergonômicos e computacionais com autômatos *Fuzzy* e sua aplicação ao projeto arquitetônico. **Ambiente Construído**, v. 14, n. 2, p. 61–77, abr./jun. 2014.
- CHUI, X.; SHI, H. A*-based Pathfinding in Modern Computer Games. **International Journal of Computer Science and Network Security**, v. 11, n. 1, p. 125–130, 2011.
- DELLING, D. et al. Engineering Route Planning Algorithms. In: LERNER, J.; WAGNER, D.; ZWEIG, K. A. (Ed.). **Algorithmics of Large and Complex Networks: design, analysis, and simulation**. Springer, 2009. p. 117–139. LNCS 5515.
- DIJKSTRA, E. W. A Note on Two Problems in Connexion. **Numerische Mathematik** 1, p. 269–271, 1959.
- HART, P. E.; NILSSON, N. J.; RAPHAEL, B. A Formal Basis for the Heuristic Determination of Minimum Cost Paths. **IEEE Transactions of System Science and Cybernetics**, v. 4, n. 2, p. 100–107, 1968.
- KO, S. Y.; SPEARPOINT, M. J.; TEO, A. Trial Evacuation of an Industrial Premises and Evacuation Model Comparison. **Fire Safety Journal**, v. 42, n. 2, p. 91–105, mar. 2007.
- KOENIG, S. et al. Incremental Heuristic Search in AI. **AI Magazine**, v. 25, n. 2, p. 99–112, 2004.
- KOWALTOWSKI, D. C. C. K.; BIANCHI, G.; PAIVA, V. T. Methods that May Stimulate Creativity and Their Use in Architectural Design Education. **International Journal of Technology and Design Education**, v. 20, n. 4, p. 453–476, 2010.
- KULIGOWSKI, E.; PEACOCK, R. D.; AVERILL, J. Modeling the Evacuation of the World Trade Center Towers on September 11, 2001. **Fire Technology**, v. 49, n. 1, 2013.
- LIGHTFOOT, T. J.; MILNE, G. J. Modelling Emergent Crowd Behaviour. In: AUSTRALIAN CONFERENCE ON ARTIFICIAL LIFE, 1, 2003, Canberra. **Anais...** Canberra, 2003. p. 159–169.
- MA, J. et al. Modeling Pedestrian Space in Complex Building for Efficient Pedestrian Traffic Simulation. **Automation in Construction**, v. 20, p. 25–36, 2013.
- PELECHANO, N.; ALLBECK, J.; BADLER, N. **Virtual Crowds: methods, simulation, and control**. [S.l.]: Morgan & Claypool Publishers, 2008.
- RONCHI, E.; NILSSON, D. Fire Evacuation in High-rise Building: a review of human behaviour and modeling research. **Fire Science Review**, v. 2, n. 7, p. 21, nov. 2013.
- RUSSELL, S.; NORVIG, P. Busca com informação e exploração. In: INTELIGÊNCIA ARTIFICIAL. Rio de Janeiro: Elsevier, 2004. cap. 4, p. 94–133. 11ª reimpressão.

SILVA, T. et al. Análise do Tempo de Evacuação Total de um Cinema por meio da Aplicação de Simulações Computacionais. **Revista Ingeniería Industrial**, v. 11, n. 1, p. 5–16, 2012.

SNIEDOVICH, M. Dijkstra's Algorithm Revisited: the dynamic programming connexion. **Control and Cybernetics**, v. 35, n. 3, p. 599–620, 2006.

SOLTANI, A. R. et al. Path Planning in Construction Sites: performance evaluation of the Dijkstra, A*, and GA search algorithms. **Advanced Engineering Informatics**, v. 16, n. 4, p. 291–303, 2002.

STOUT, B. Smart Moves: intelligent pathfinding. **Game Developer Magazine**, p. 28–35, out./nov. 1996.

THOMPSON, P. A.; MARCHANT, E. W. A Computer Model for the Evacuation of Large Building Populations. **Fire Safety Journal**, v. 24, p. 131–148, mar./apr. 1995.

WOLFRAM, S. Statistical Mechanic of Cellular Automata. **Reviews of Modern Physics**, v. 55, n. 3, p. 601–644, 1983.