



ALGORITMO PARA GERAÇÃO DO MENOR CAMINHO DISCRETO SUAVE A SER PERCORRIDO ATÉ A SAÍDA DE UM AMBIENTE

Henrique Costa Braga

Gray Farias Moita

Paulo Eduardo Maciel de Almeida

bragaseg@yahoo.com.br

gray@dppg.cefetmg.br

pema@lsi.cefetmg.br

Programa de Pós-Graduação em Modelagem Matemática e Computacional

Centro Federal de Educação Tecnológica de Minas Gerais, CEFET-MG

Av. Amazonas 7675, Nova Gameleira, Belo Horizonte, MG, CEP 30510-000, Brasil

Resumo. *Uma aplicação da simulação computacional da movimentação de pessoas está na avaliação das melhores rotas e da dinâmica de fluxo durante o abandono de um ambiente em uma situação de emergência e pânico, tal como a causada por um incêndio. Para isto, podem ser usados diversos algoritmos de busca como suporte. Muitos destes algoritmos, que possuem simplicidade lógica e fácil implementação computacional, fornecem perfis de deslocamento com a possibilidade da ocorrência de variações bruscas de direcionamento não realistas, muitas vezes até ortogonais. Neste trabalho, será detalhado o algoritmo BR (busca referenciada) que, além de ter simplicidade lógica e possuir relativo baixo custo computacional (para uma aplicação não dinâmica), é capaz de gerar um perfil de deslocamento humano sem a ocorrência de variações bruscas artificiais, devido ao fornecimento de resultados muito próximos do melhor teórico. O algoritmo BR pode ser utilizado em ambientes de quaisquer geometrias internas e externas, com uma ou múltiplas saídas. As vantagens da utilização do algoritmo proposto será evidenciada pela comparação com resultados obtidos por outros algoritmos. Assim, apresenta-se um algoritmo eficiente e com resultados de qualidade que pode ser introduzido em programas de alta performance para a modelagem da movimentação humana em situações de evacuação de ambientes.*

Palavras-chave: Algoritmo de busca, Simulação, Evacuação, Movimentação de pessoas, Pathfinder.

INTRODUÇÃO

O estudo da movimentação de pessoas ou materiais em ambientes construídos é importante para melhor conhecer os sistemas ou processos em que estão inclusos e facilitar a realização de melhorias aos mesmos. A simulação e a modelagem computacional da movimentação nestes ambientes são ferramentas de grande valor para este estudo, aplicáveis a áreas tão distintas como logística, pesquisa operacional, otimização, roteirização, robótica, automação, jogos (de natureza séria ou lúdica), segurança, dentre outras. Uma relevante aplicação da simulação da movimentação em um ambiente construído está em se determinar os percursos a serem adotado pelos ocupantes de uma edificação rumo à saída em caso de uma evacuação por motivo de emergência, como a provocada por uma situação de incêndio ou pânico. Simulações da movimentação humana muitas vezes se utilizam de algoritmos de busca como suporte à modelagem (THOMPSON, MARCHANT, 1995; PELECHANO *et al.*, 2008; BRAGA *et al.*, 2014).

Pela relevância, são inúmeros os algoritmos de busca existentes e também suas variações (SOLTANI *et al.*, 2002; KOENIG, S. *et al.*, 2004; SNIEDOVICH, 2006), cada um com suas vantagens e desvantagens. Assim, não existe uma solução única geral, já que cada caso tem suas particularidades e muitas vezes se torna necessário a adaptação de algoritmos já estabelecidos para melhor se atender a demanda específica da simulação. Em trabalho anterior (BRAGA *et al.*, 2015), foram apresentados dois algoritmos de busca sobre o ambiente construído a ser simulado, sendo o primeiro através de uma pesquisa sobre a primeira Vizinhança de Von Neumann (VVN) e o segundo pela pesquisa sobre a primeira Vizinhança de Moore (VM). Em ambos algoritmos, o ambiente é tratado na sua forma discreta e a busca se dá a partir das saídas e cobrindo todo o ambiente.

Estes algoritmos (VVN e VM) geram, para todo o ambiente, uma matriz de custos de movimentação (ou de distâncias aparentes) que pode ser usada como um indicativo das melhores rotas a serem seguidas rumo à saída. Além disso, estes algoritmos apresentam várias características positivas, tais como a sua simplicidade lógica, fácil implementação computacional, generalidade (aplicam-se a ambientes construídos de qualquer geometria interna ou externa), total automação da sua aplicação (não necessita de intervenção humana) e tempo de processamento relativamente baixo (para uma aplicação não dinâmica).

Entretanto, a discretização do ambiente associada ao método empregado para evolução dos algoritmos implementados segundo VVN e VM geram erros nos valores da matriz de custo de movimentação que, em muitos casos, podem ser significativos. Outra característica é que mesmo se teoricamente existir um único melhor percurso rumo à saída, os erros de discretização e de evolução do algoritmo geram vários percursos distintos com o mesmo custo. Finalmente, o percurso indicado rumo a saída não é tão suave como seria esperado, havendo muitas variações bruscas na direção de movimentação, inclusive ortogonais. A variação do algoritmo VM (que leva em consideração movimentos diagonais) apresentou resultados muito melhores e mais coerentes que os obtidos considerando o algoritmo VVN; entretanto, mesmo pela Vizinhança de Moore, os caminhos indicados ainda aparentam serem artificiais.

Desta maneira, melhorias tornam-se necessárias nos algoritmos implementados, visando se manter todas as vantagens destes, porém minimizando os erros e gerando perfis de deslocamento mais suaves e coerentes. Um algoritmo com estas características permitirá o desenvolvimento de aplicações, tais como softwares para simulação da evacuação de ambientes, mais realísticos e de alta performance.

1 DESENVOLVIMENTO

1.1 Discretização do ambiente

Como primeiro passo, o ambiente construído contínuo alvo da simulação deve ser, em sua vista de topo, discretizado na forma de uma matriz retangular bidimensional denominada de matriz ambiente (MA). Nesta matriz MA cada elemento, cuja posição é indicada pela sua linha e coluna na matriz MA, representa um determinado posicionamento no mundo real. A distância física representada no mundo real entre duas linhas ou colunas consecutivas é denominada de resolução (r). Quando menor o valor de r mais precisa será a simulação, mas maior será o custo computacional associado.

Os valores contidos em cada um dos elementos da matriz MA estão diretamente associados a um significado no mundo real. Por exemplo, o valor “0” na matriz MA indica que este respectivo elemento no mundo real está vazio e pode ser usado como passagem; o valor “-1” na matriz MA indica que este pertence a uma parede ou obstáculo; o valor “-2” na matriz MA indica que este elemento pertence a um dos objetivos da simulação (uma saída ou sala específica); e, o valor “-3” na matriz MA indica que este elemento pertence a uma região externa ou não considerada do ambiente (apesar de pertencer a matriz MA, não participa da simulação). Na Fig. 1 se apresenta um exemplo de um ambiente construído e da respectiva matriz MA gerada. Esta matriz MA inicial não é alterada durante a simulação.

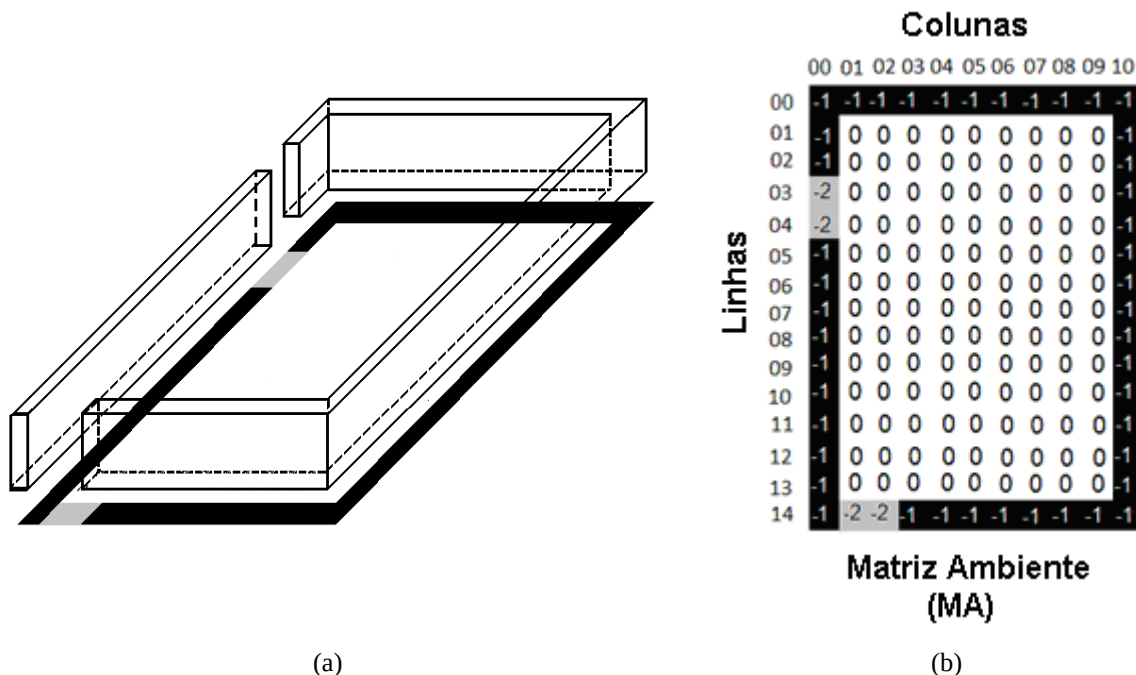


Figura 1. Representações. em (a) de um ambiente e sua vista de topo em perspectiva (sem escala) e em (b) de sua respectiva matriz MA onde “0” indicam espaço vazios, “-1” as paredes e “-2” as saídas

1.2 Matrizes de suporte

Discretizado o ambiente com a geração da matriz MA são então criadas adicionalmente quatro distintas matrizes de suporte. Duas possuem as mesmas dimensões (linhas e colunas) da matriz MA, mas com todos seus elementos contendo inicialmente o valor “0”. Estas duas matrizes dimensionalmente semelhantes à matriz MA são denominadas de matriz distância

aparente (MDA) e de matriz de direcionamento e referência (MDR).

A terceira matriz de suporte será denominada de matriz de busca primária (MBP). A matriz MBP possui duas linhas e inicialmente tantas colunas quanto forem os elementos de saídas (elementos de valor “-2”) existentes na matriz MA. Na primeira linha da matriz MBP, cada elemento será associado a um elemento de saída da matriz MA. Para esta associação, os índices (linhas e colunas) da matriz MA que contém as saídas serão colocados um a um nos elementos da primeira linha da matriz MBP. Assim, cada elemento da matriz MBP estará associado a um elemento de saída da matriz MA. A sequência ou ordem no qual a associação entre a matriz MA e a matriz MBP é realizada não interfere no resultado final do algoritmo.

Por ser o primeiro passo do algoritmo, cada elemento da segunda linha da matriz MBP conterá os mesmos valores do respectivo elemento da primeira linha da matriz MPB de sua mesma coluna. A Fig. 2 apresenta a matriz MPB inicialmente gerada para o ambiente da Fig. 1. Finalmente, a quarta matriz de suporte será denominada de matriz de busca secundária (MBS) e terá sempre duas linhas, inicialmente nulas.

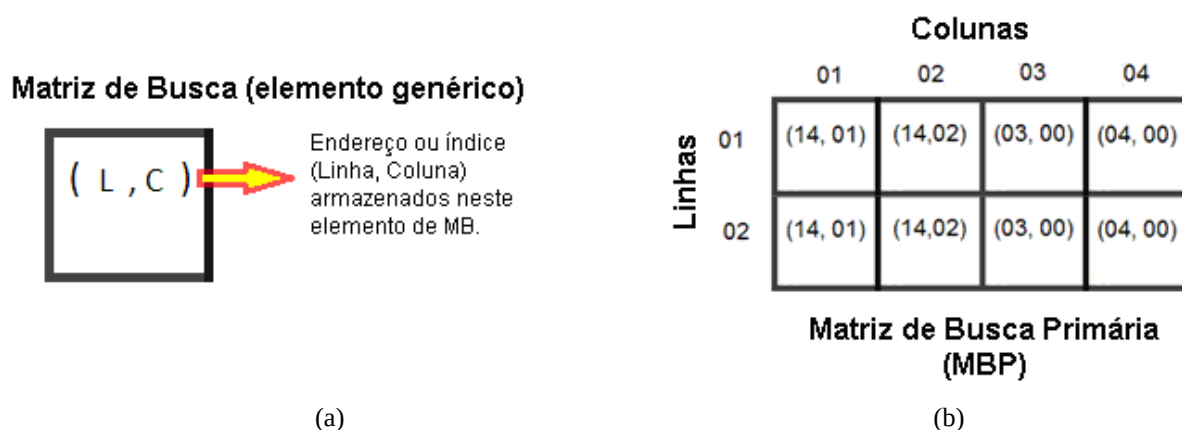


Figura 2. Representações em (a) de um elemento genérico da matriz MBP, e em (b) a respectiva matriz MBP inicial gerada para o ambiente da Fig. 1

1.3 O algoritmo de busca referenciada – primeiro laço

Geradas as quatro matrizes de suporte (MDA, MDR, MBP e MBS), o algoritmo de busca se inicia pela pesquisa sobre cada elemento da primeira linha da matriz MBP. Considerando o primeiro elemento da primeira coluna da matriz MBP indicado na Fig. 2b, ele contém o que se denominará índice base de varredura inicial, no caso a linha 14 e coluna 1, ou seja, este é um endereço que na matriz MA representa uma saída ou objetivo. Todos os elementos da matriz MA que são os primeiros vizinhos de Von Neumann (os elementos imediatamente acima, abaixo, a direita e a esquerda) deste objetivo indicado serão verificados.

O primeiro passo da verificação consiste em se determinar se na matriz MA estes elementos vizinhos são válidos. Somente atendem ao critério de validade aqueles elementos que na matriz MA não pertençam a paredes, obstáculos ou saídas, nem que sejam externos ao ambiente ou externos a própria matriz MA.

Quando um elemento pesquisado da matriz MA é considerado válido, o índice deste elemento (linha e coluna) é adicionado na primeira linha da matriz MBS. Como se está no primeiro laço do algoritmo, a segunda linha da mesma coluna da matriz MBS terá o mesmo índice base de varredura inicial (no caso a linha 14 e coluna 1). Na Fig. 3 está representado este primeiro passo de busca e a respectiva matriz MBS, gerada a partir das matrizes da Fig. 2.

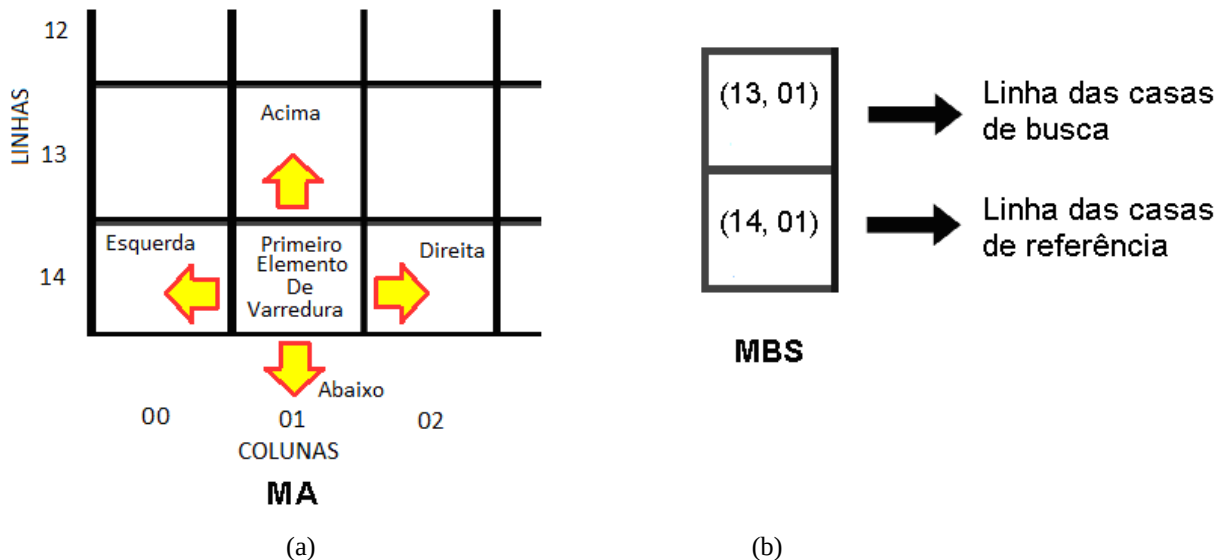


Figura 3. Representações em (a) da primeira vizinhança de busca na matriz MA, e em (b) da respectiva matriz MBS final após esta primeira busca sobre o primeiro elemento da matriz MBP

Nota-se que os elementos da matriz MA à esquerda, à direita e abaixo do primeiro elemento de varredura não foram adicionados a matriz MBS porque não atenderam ao critério de validade (no caso destes elementos, os mesmos pertencem a uma parede, ou à própria saída, ou estão fora da matriz MA). Consequentemente, somente o elemento da matriz MA localizado acima da casa de varredura atendeu ao critério de validade e teve seu índice transferido para a matriz MBS.

Os elementos da primeira linha da matriz MBP serão todos pesquisados, e mantendo o mesmo raciocínio, os respectivos elementos da matriz MA, considerados como válidos, serão adicionados em novos elementos na primeira linha da matriz MBS. Terminada a varredura por sobre a matriz MBP, a matriz MBS estará então completa. Adicionalmente, neste primeiro laço, todos os elementos válidos estão a um elemento de distância da saída. Assim, todos os elementos na matriz MDA cuja posição está indicada na primeira linha da matriz MBS terão seu valor alterados de “0” para “1”.

Finalmente, agora na matriz MDR, todos estes mesmos elementos alterados na matriz MDA terão o respectivo valor contendo o endereço (linha e coluna) do elemento da saída a que estão a uma casa de distância. Na Fig. 4 está a representação de parte da matriz MDA, de parte da matriz MDR alterada, e de toda a matriz MBS final após este primeiro laço de verificação sobre toda a MBP inicial.

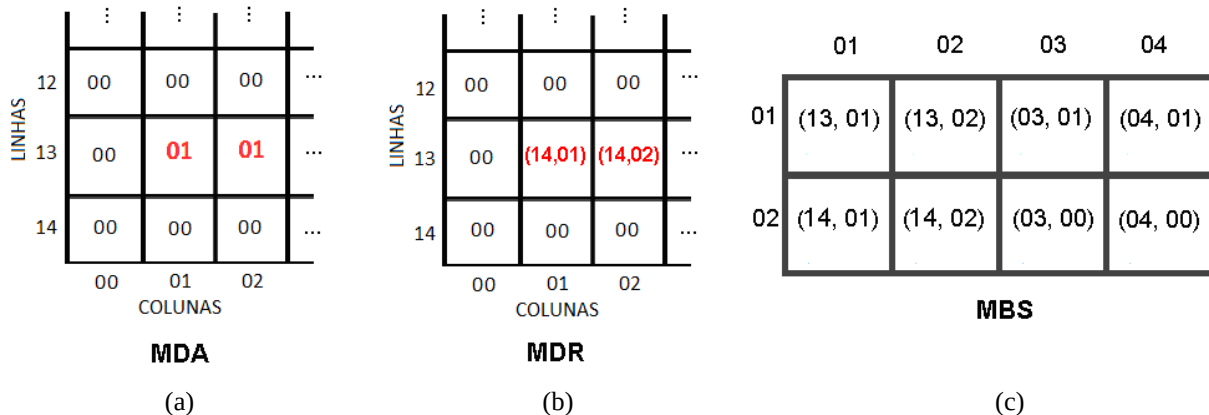


Figura 4. Representações, em (a) e (b) repectivamente de parte atualizada da matriz MDA e da matriz MDR, e em (c) da matriz MBS final após a primeira busca sobre toda a MBP inicial

Nesse momento, a matriz MDA passa a conter, em todos os seus elementos que não possuem o valor “0”, a distância (em número de elementos) que este respectivo elemento está da saída mais próxima. Ainda, a matriz MDR passa a conter, em todos os seus elementos que não possuem o valor “0”, o endereço (índice linha e coluna) da saída mais próxima do respectivo elemento.

Terminada este primeiro laço de busca por toda a matriz MBP inicial, toda esta matriz MBP é então substituída pela matriz MBS final, ou seja, uma nova matriz MBP é criada com os mesmos valores e dimensões (número de colunas e linhas) da matriz MBS final. Feita esta troca de valores, a matriz MBS é então novamente feita nula.

As linhas da nova matriz MBP terão denominações particulares, relativas à função de cada uma. A primeira linha da matriz MBP será denominada de linha das casas de busca e a segunda de linha das casas de referência.

1.4 O algoritmo de busca referenciada – próximos laços

O próximo laço será realizado sobre a nova matriz MBP (antiga matriz MBS final da etapa anterior). Considerando o início a partir da primeira coluna da matriz MBP (Fig. 4c), o primeiro índice das casas de busca (primeira linha) indica a linha 13 e a coluna 1, e o índice da respectiva casa de referência (segunda linha) também indica a linha 13 e a coluna 1.

Uma nova verificação então se dará sobre a matriz MA considerando as casas vizinhas, segundo Von Neumann, do índice indicado pela casa de busca. Caso alguma casa vizinha da matriz MA não seja válida (pertença a paredes, obstáculos, saídas ou esteja fora do ambiente válido ou da própria matriz MA), esta casa não será considerada. Caso seja uma casa válida, a partir desta segunda rodada, tem-se também que verificar a adequação da mesma. Sendo considerada adequada é então calculada sua D_{pv} (distância pontual da casa sendo pesquisada vizinha a casa base de varredura).

Alteração das matrizes de suporte

Tendo sido calculado o D_{pv} , verifica-se então se o mesmo é menor que o valor já contido na matriz MDA para esta mesma casa. Se o valor de D_{pv} for menor que o valor já contido na matriz MDA, ou se na matriz MDA estiver com o valor “0”, esta casa válida é agora também considerada como adequada. Caso contrário, a casa, apesar de válida, é considerada como não adequada (o algoritmo nunca pode “piorar” um valor anteriormente encontrado). Se a casa for considerada não adequada, nenhuma alteração é feita nas matrizes MDA, MDR e MBS.

Se a casa for considerada adequada, o índice desta casa vizinha (linha e coluna) é adicionado na primeira linha de uma nova coluna da matriz MBS. Caso o critério de linearidade tenha sido atendido, na segunda linha desta mesma coluna da matriz MBS será colocado o índice (linha e coluna) contido na segunda linha da coluna da matriz MBP que está sendo analisada. Caso o critério de linearidade tenha sido reprovado, na segunda linha desta mesma coluna da matriz MBS será colocado o índice (linha e coluna) relativo à casa de varredura.

Ainda se a casa for considerada adequada, o valor de D_{pv} será transferido substituindo o valor da matriz MDA da casa vizinha que está sendo analisada, e o valor do elemento da segunda linha da nova matriz MBS que acabou de ser adicionado será também transferido,

substituindo o valor da matriz MDR da mesma casa vizinha que está sendo analisado.

Finalizando os laços

Todas as etapas anteriores são repetidas para todos os elementos de varredura da casa base considerada, e depois para todas as casas bases relacionadas na primeira linha da matriz MBP. Terminada a verificação, a matriz MBP será apagada e nova matriz MBS será então feita como sendo a nova matriz MBP. Após esta substituição a matriz MBS é feita nula. Este processo se repete continuamente, até que ao final de alguma rodada após a verificação de todos os elementos de varredura de todas as casas base relacionadas na matriz MBP em uso, não se encontre nenhuma casa válida e adequada.

Neste momento, a matriz MBS que no início do laço estava nula continuará nula ao fim do mesmo. Isto indica que todo o ambiente já foi inteiramente analisado e que a busca está completa. Assim, a matriz MDA final e a matriz MDR final têm respectivamente, em todos os seus elementos não nulos, a distância em números de elementos da saída mais próxima e o indicativo da melhor direção a ser seguida rumo à saída.

Este algoritmo descrito foi implementado computacionalmente em linguagem Python v.3.4 e será denominado de algoritmo de busca referenciada (BR).

2 RESULTADOS

Inicialmente são simulados ambientes em forma de quadrados, sem separações internas, com paredes de um elemento de espessura e um único objetivo (saída) de dois elementos de largura posicionado no centro de um dos lados. Estes ambientes quadrados tem largura externa variando de 10 elementos a 1.000 elementos.

Na Fig. 5 se apresenta uma imagens de momentos da simulação indicando o perfil de evolução do algoritmo no ambiente de 200 elementos de lado (vista de topo), onde em verde está a região ainda não verificada pelo algoritmo e em branco a região já pesquisada. A saída do ambiente da Fig. 5 está localizada no centro do lado inferior.

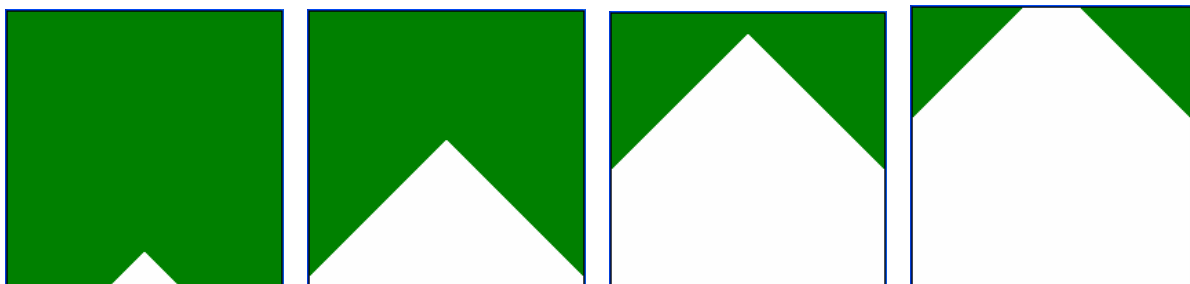


Figura 5. Diferentes estágios do perfil de evolução do algoritmoBR sobre o ambiente de 200 elementos (vista de topo). A região em branco identifica a área já pesquisada e a verde a não pesquisada. A saída se localiza no centro do lado inferior

Na Fig. 6 se apresenta a variação do custo computacional em tempo de processamento (s) para as simulações de todos estes ambientes quadrados de 10 a 1000 elementos de lado. Para comparação, foram incorporadas na Fig. 6 os tempo de processamento com a mesma configuração computacional para os mesmos ambientes simulados por algoritmos diretos de busca conforme VM e VVN.

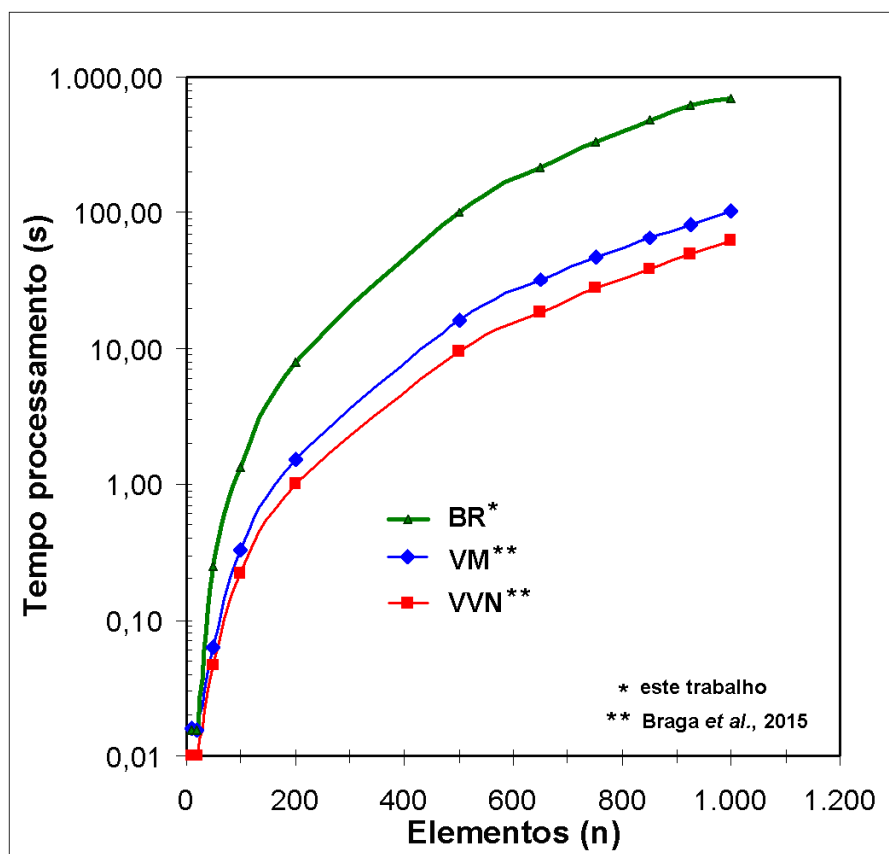


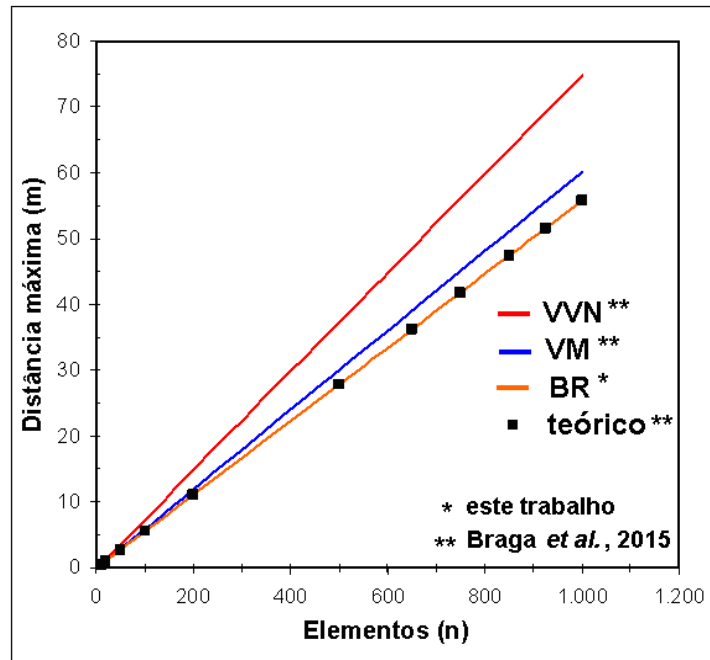
Figura 6. Tempo de processamento em função do número de elementos para todas as simulações dos ambientes quadrados e diversos algoritmo de busca

Ainda para estes ambientes quadrados entre 10 e 1000 elementos de lado, considerando a resolução r como sendo de 5 cm, foram obtidos, através da matriz MDA, o valor da maior distância entre um ponto do ambiente e o objetivo ($D_{\text{máx}}$), assim como o valor da distância média entre todos os pontos do ambiente e o objetivo mais próximo ($D_{\text{méd}}$).

O valor teórico para o $D_{\text{máx}}$ exato também foi calculado para cada simulação para fins de comparação. Na Fig. 7 se apresentam os resultados obtidos de $D_{\text{máx}}$ e $D_{\text{méd}}$ e para comparação também foram adicionados os valores obtidos pelos algoritmos de busca direta conforme VM e VVN.

Adicionalmente, para se facilitar a identificação e visualização dos perfis de evolução dos valores de distância aparente em relação a saída mais próxima ao longo de todo o ambiente, são gerados os respectivos mapas de distância (THOMPSON & MARCHANT, 1995). O mapa de distância é uma espécie de representação gráfica estilizada da matriz MDA, onde cada faixa de cor representa uma faixa de distanciamento em relação ao ponto objetivo ou saída.

(a)



(b)

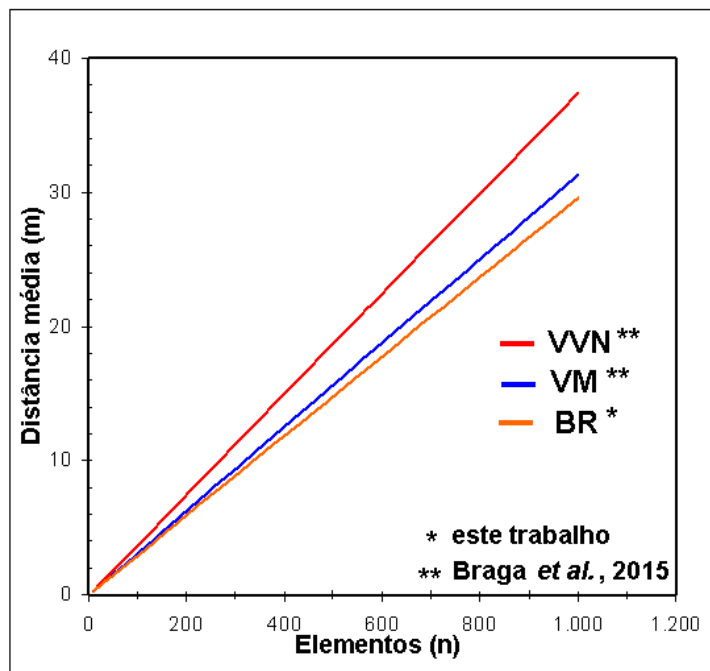


Figura 7. A variação da distância máxima (a) e da distância média (b) para todas as simulações dos ambientes quadrados considerando $r = 5$ cm

Assim, na Fig. 8 estão mostrados os mapas de distância para o ambiente quadrado com n de 200 elementos (ou 100 m^2 de área) e para o ambiente quadrado com n de 1000 elementos (2500 m^2 de área). Para comparações, os mapas de distância de ambos os ambientes foram gerados pelo algoritmo BR e também pelos algoritmo de busca simples, segundo VM e VVN.

As diferentes cores de faixas no mapa de distâncias da Fig. 8 sempre indicam um distanciamento de 1 m crescente em relação à saída, sendo que o objetivo (ou saída) está localizado no centro do lado inferior (vista de topo) do ambiente.

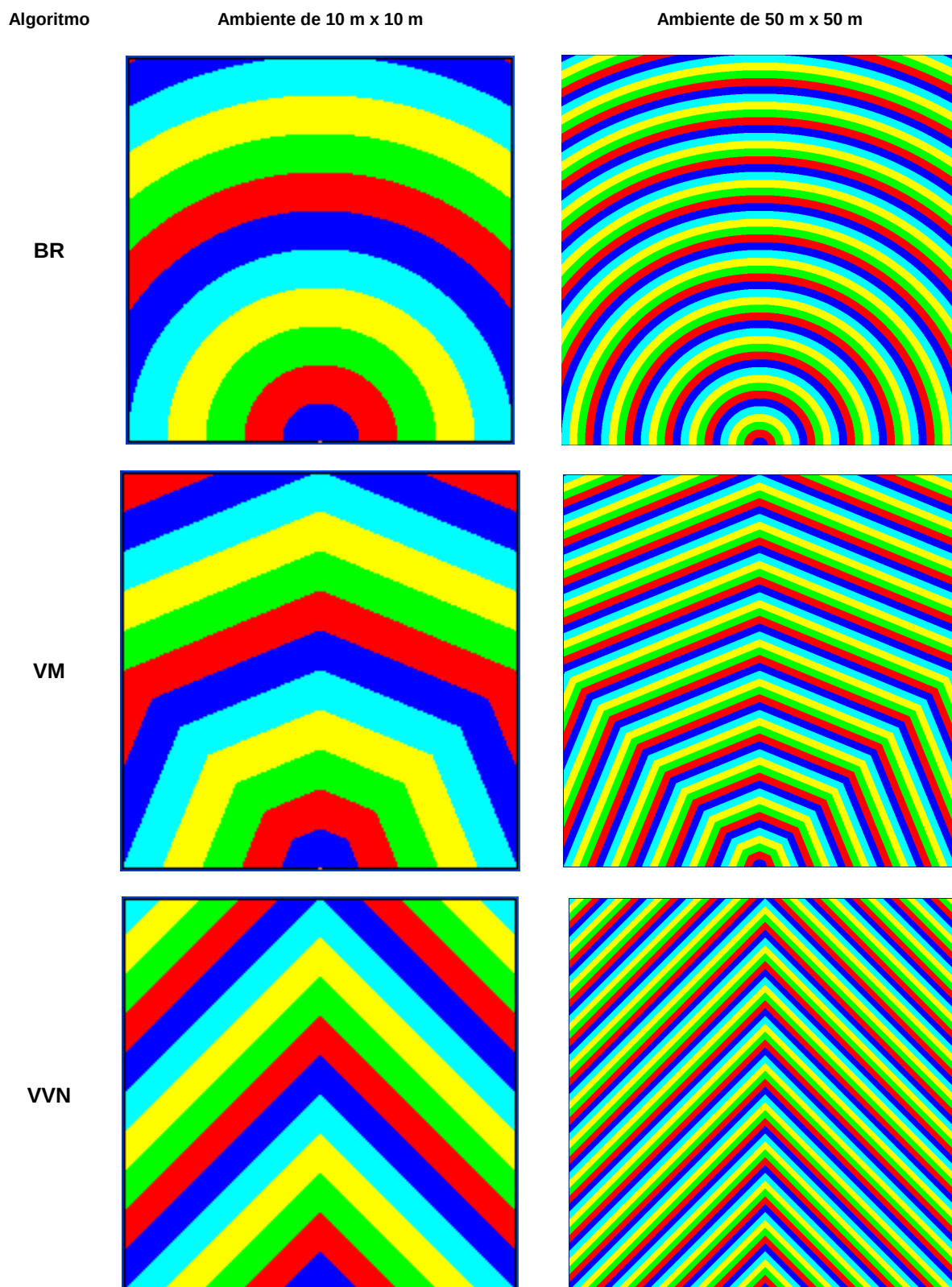


Figura 8. Representação dos mapas de distância gerados para os três algoritmos (BR, VM, VVN) para ambientes quadrados, sendo o da esquerda de 100 m² de área e o da direita de 2500 m² de área

Finalmente, para se mostrar a influência no direcionamento e a questão da possibilidade da geração de rotas ortogonais artificiais de deslocamento, foram realizadas simulações no ambiente da Fig. 1. Considerando um posicionamento inicial qualquer, o encaminhamento

ótimo indicado pelos algoritmos será aquele que propiciar se alcançar o objetivo seguindo a trilha com o menor valor da distância aparente (D_p). A matriz MDR faz esta indicação. Na Fig. 10a são apresentadas as rotas de movimentação até a saída geradas para o ambiente da Fig. 1 pelo algoritmo BR e pelos algoritmos de busca simples segundo VM e VVN.

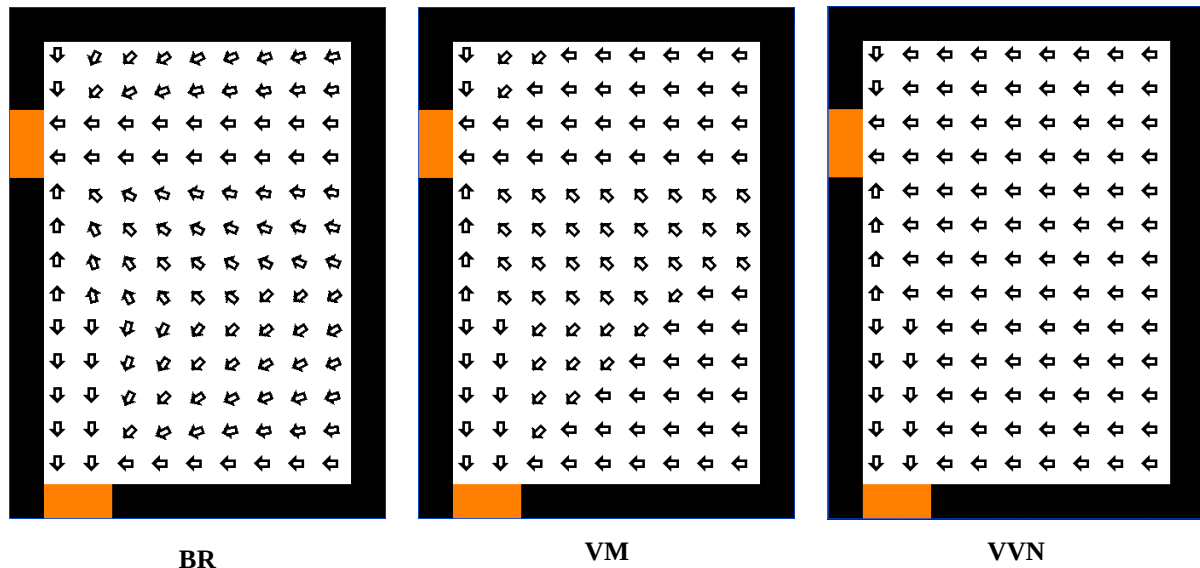


Figura 10. Vista de topo do ambiente da Fig. 1, estilizado com setas indicando o melhor caminho a ser escolhido até o objetivo mais próximo (em laranja), para os três algoritmos (BR, VM e VVN)

3 DISCUSSÕES

A forma gráfica da evolução do algoritmo de busca suave sobre o ambiente apresentada pela Fig. 5 era esperada devido a este algoritmo seguir segundo uma primeira vizinhança de Von Neumann. Entretanto, as semelhanças com o algoritmo segundo VVN param por aí, já que o algoritmo BR utiliza a todo momento a casa referênciada (segunda linha da matriz MBP) como balizadora tanto do distanciamento (matriz MDA) como do direcionamento (matriz MDR), e não somente os primeiros vizinhos da casa de busca (primeira linha da matriz MBP) como fazem os algoritmos de busca simples.

Pela Fig. 6 verifica-se que, para um mesmo ambiente e requisitos computacionais, o tempo de processamento do algoritmo BR é muito superior ao tempo de processamento dos algoritmos de busca simples VM e VVN (aproximadamente uma ordem de grandeza a mais para ambientes quadrados sem repartições internas). Este é um ponto negativo do algoritmo BR, trazendo maiores dificuldades a eventuais aplicações dinâmicas do mesmo. Entretanto, para aplicações não dinâmicas, como pode acontecer em programas de simulação de evacuação de ambientes (THOMPSON, MARCHANT, 1995; BRAGA *et al.*, 2014), este tempo gasto no processamento não é um impeditivo e os valores alcançados são aceitáveis.

Pela Fig. 7a, verifica-se que os valores de $D_{\max}$ fornecidos pela algoritmo BR são similares aos valores teóricos calculados para aqueles mesmos ambientes, mas cerca de 8% inferiores aos valores fornecidos pelo algoritmo de busca simples conforme uma VM e cerca de 34% inferiores pelo algoritmo de busca simples conforme uma de VVN. Esta é uma grande qualidade a favor do algoritmo BR, a alta precisão dos valores de distância obtidos, mesmo quando comparados com os valores teóricos, sem a necessidade de nenhum tipo de correção. Em relação à $D_{\text{méd}}$, a diferença entre os valores obtidos pelos algoritmos BR, VM e VVN

mantêm esta mesma diferença relativa (Fig. 7b).

Devido a qualidade dos resultados, o perfil das curvas dos mapas de distância gerado pelo algoritmo BR (Fig. 8) é geralmente macroscopicamente diferenciável, sem variações bruscas, como se fosse similar a arcos de circunferência. Já os mapas de distância para estes mesmos ambientes gerados pelos algoritmos VM e VVN (Fig. 8) são formados, mesmo macroscopicamente, por segmentos de retas contínuas e não colineares.

Para melhor visualizar as diferenças nos perfis, na Fig. 11 todos os mapas de distância para o ambiente de 100 m² foram superpostos em uma única representação estilizada. As semi-retas vermelhas foram obtidas do mapa de distância conforme VVN, as retas azuis do mapa de distância conforme VM, e ao fundo desbotado está o mapa de distância obtido pelo algoritmo BR.

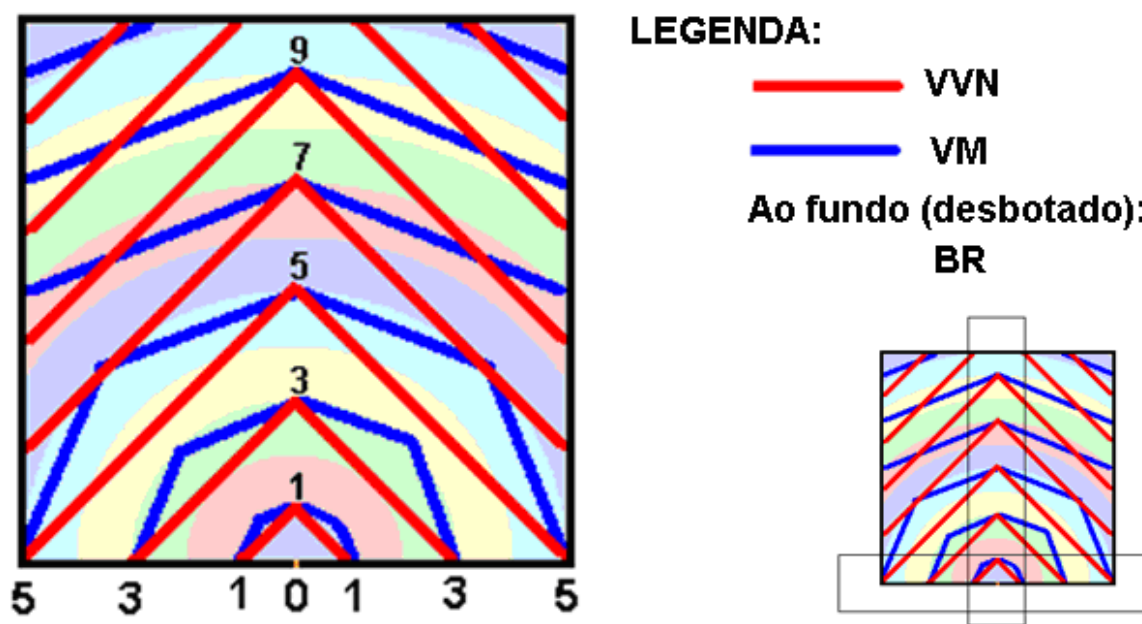


Figura 11. Representações de algumas das linhas de distância geradas da Fig. 8 obtidas por todos os algoritmos. Em miniatura a direita a reprodução desta imagem com a identificação das áreas (retângulos) onde a discrepância entre as distâncias obtidas pelos algoritmos é menor

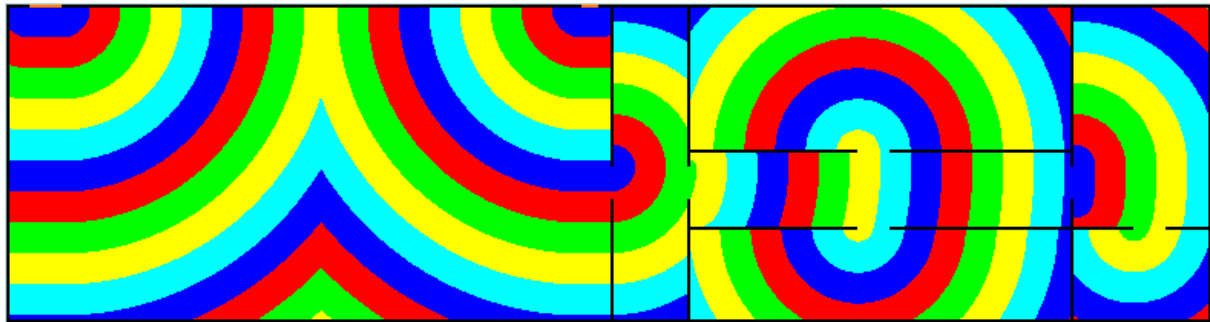
O primeiro conjunto de segmento de retas azuis e vermelhas contínuo (o conjunto com o número 1 posicionado perto da interseção de ambos) indica a posição localizada a 1 m da saída (indicada pelo número 0). Exceto pelas interseções, os segmentos vermelhos estão mais próximos da saída que os segmentos azuis. Isto indica que para um ponto qualquer no ambiente (exceto nas interseções), o algoritmo segundo VVN fornece distâncias maiores que o valor de distância fornecido pelo algoritmo segundo VM.

Considerando o mapa de distância desbotado ao fundo, gerado pelo algoritmo BR, os valores de distância gerado para todos os três algoritmos são macroscopicamente os mesmos nas interseções, mas fora das interseções, o valor relativo ao distanciamento de 1 m da saída obtido pelo algoritmo BR está ainda mais distante da saída que os obtidos pelos algoritmos segundo VVN e VM. Isto fica mais evidente ao se observar as linhas de distância de 3 m e de 5 m. Esta característica é coerente com as discrepâncias entre os valores de distâncias máximas obtidos (Fig. 7).

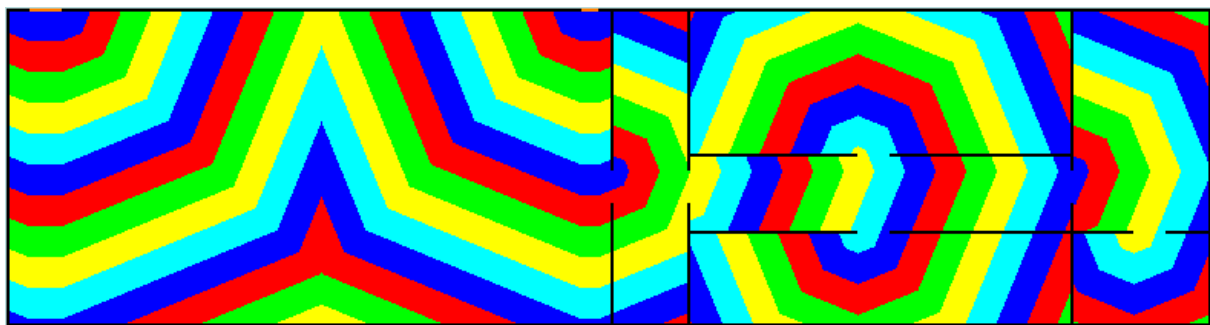
Os algoritmos segundo a busca simples incorrem em erros devido a discretização. Em um caso extremo, para uma movimentação de duas casas diagonais na matriz ambiente

segundo VVN são necessárias duas movimentações, resultando num distanciamento de 2 casas, enquanto o valor correto seria de $\sqrt{2}$ casas de aumento no distanciamento entre estas casas. O algoritmo BR minimiza enormemente este efeito, como se estivesse mesclando o mundo discreto, com suas vantagens computacionais, e o mundo contínuo, com suas vantagens físicas.

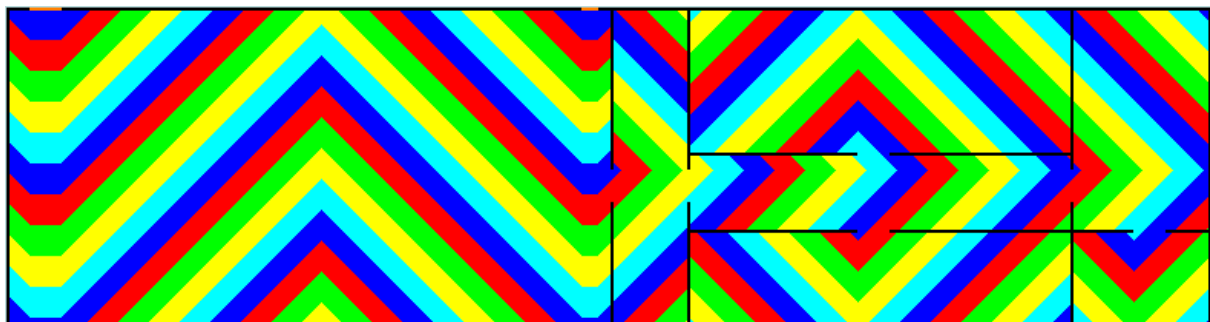
Para visualizar melhor ainda mais este efeito, na Fig. 12 se tem os mapas de distância obtidos para um ambiente mais sofisticado (com várias passagens internas) gerado pelo algoritmo BR e também pelos VM e VVN. Cada faixa de cor representa um distanciamento de 1 m, e são duas saídas independentes localizadas na sala maior (em laranja)



Algoritmo BR: Dmáx de 27,0 m; Dméd de 12,0 m e tempo de processamento de 54 s.



Algoritmo VM: Dmáx de 27,5 m; Dméd de 12,4 m e tempo de processamento de 11 s.



Algoritmo VVN: Dmáx de 30,3 m; Dméd de 13,9 m e tempo de processamento de 8 s.

Figura 12. Representação esquemática da movimentação entre casas vizinhas

No ambiente da Fig. 12 as diferenças verificadas entre os valores da distância máxima e da distância média dos algoritmos foi menor que as diferenças encontradas anteriormente (Fig. 7). Isto aconteceu porque a específica geometria irregular retangular da Fig. 12, tanto interna quanto externa (ambientes formados por retângulos posicionados em ângulos múltiplos de 90° em relação às coordenadas do ambiente) minimizam os erros do efeito da discretização, que por outro lado pode ser acentuado em outros formatos.

A pior situação (maior diferença entre os valores de distância entre os algoritmos) obtida ocorreu em ambientes cuja discretização esteja a múltiplos de 45° do posicionamento deste

ambiente para o algoritmo conforme VVN e para múltiplos de $22,5^\circ$ e não múltiplos de 45° para o algoritmo conforme VM. A Fig. 11b corrobora estas observações pela identificação das regiões onde os perfis de distância dos algoritmos se aproximam (ou se afastam). A Fig. 13 ilustra estes casos e a Tab. 1 apresenta os respectivos valores encontrados de $D_{m\acute{a}x}$, $D_{m\acute{e}d}$ e suas diferenças entre os ambientes a $22,5^\circ$ e a 45° para os três algoritmos.

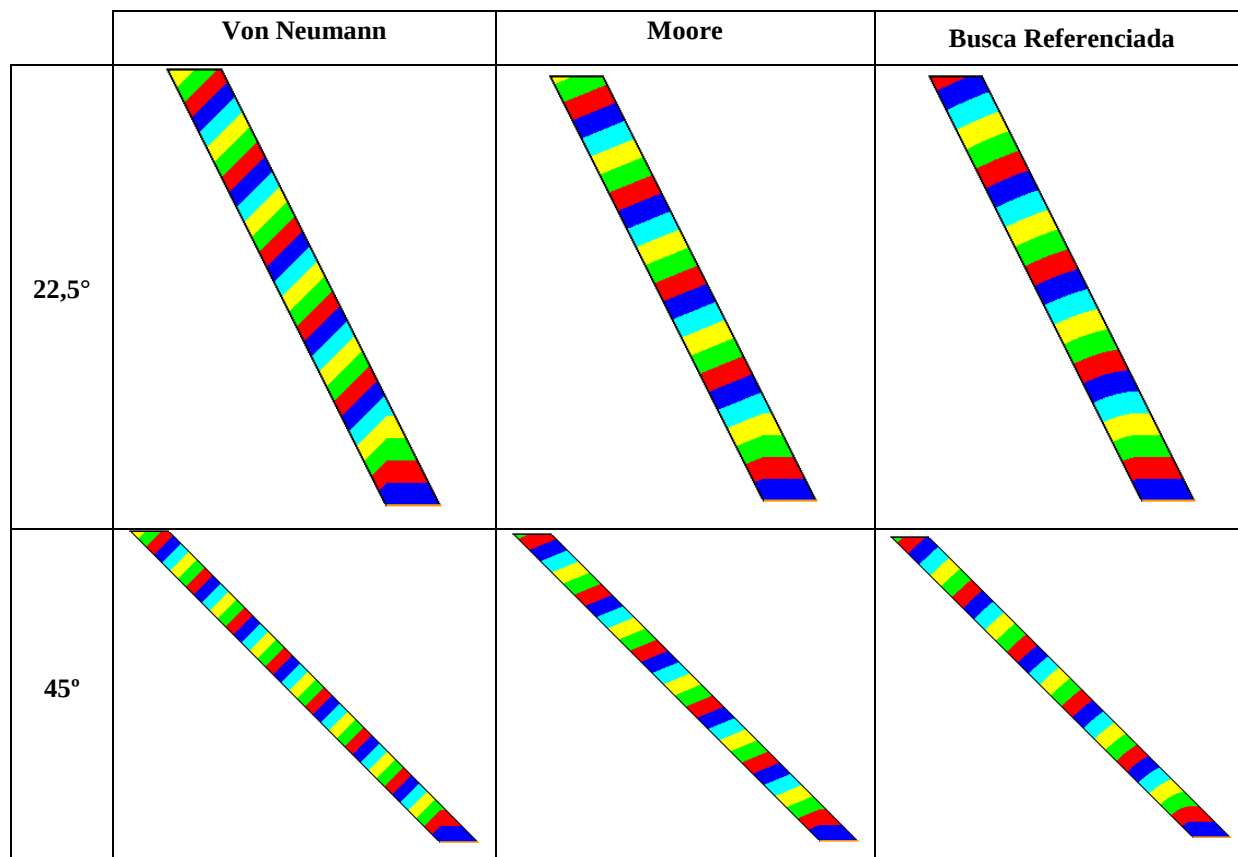


Figura 13. Representação dos mapas de distância para ambientes cuja geometria faz com a matriz MA um ângulo de $22,5^\circ$ ou de 45° obtidos para todos os algoritmos. Cada faixa de cor representa um distanciamento de 1 m em relação à saída, identificada em laranja na parte inferior dos ambientes

Tabela 1. Valores de $D_{m\acute{a}x}$, $D_{m\acute{e}d}$ e do tempo de processamento obtidos para todas as simulações da Fig. 13. As diferenças identificadas entre as distâncias estão em relação ao valor gerado pelo algoritmo BR

	Ambiente a $22,5^\circ$			Ambiente a 45°		
	VVN	VM	BR	VVN	VM	BR
$D_{m\acute{a}x}$ (m)	29,0	23,3	21,6	38,7	27,3	27,3
Dif $D_{m\acute{a}x}$ (%)	+ 34,3	+ 7,9	-	+ 41,8	0	-
$D_{m\acute{e}d}$ (m)	13,5	11,3	10,5	18,3	13,2	13,1
Dif $D_{m\acute{e}d}$ (%)	+ 28,6	+ 7,3	-	+ 39,7	+ 0,8	-
Tempo (s)	1,4	1,5	2,3	1,5	1,5	2,4

4 CONSIDERAÇÕES FINAIS

O algoritmo de busca referenciada (BR) descrito neste trabalho apresenta diversas características positivas como relativa simplicidade lógica, fácil implementação computacional, capacidade de generalização (independe da geometria interna ou externa do ambiente), baixo tempo de processamento (considendo sua aplicação não dinâmica) e total automação na sua aplicação (independe de intervenção humana). Ademais, em comparação com os algoritmo de busca simples pela primeiras vizinhança de Von Neumann ou de Moore, apresenta resultados próximos do ideal em todo o ambiente e com perfis de mapas de distância sem o aparecimento de variações bruscas artificiais de direcionamento (níveis em formato aproximadamente circular). Assim, o algoritmo de busca referenciada é potencialmente interessante para ser utilizado em programas de alta performance de simulação da movimentação humana, inclusive em um contexto de evacuação de ambientes.

Agradecimentos

Os autores agradecem ao CEFET-MG e a CAPES pelo suporte recebido.

REFERÊNCIAS

- Braga, H.C., Moita, G.F., Camargo, F. e Almeida, P.E.M., 2014. 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, pp. 61-77.
- Braga, H.C., Moita, G.F. e Almeida, P.E.M., 2015. Análise comparativa de algoritmos de busca para simulação e modelagem do melhor caminho inicial de percurso em um ambiente construído. 1º EINEPRO – Encontro Interestadual de Engenharia de Produção. *Anais ...*, São João da Barra: UCAM, 12p.
- Pelechano, N., Allbeck, J. e Badler, N., 2008. *Virtual crowds: methods, simulation, and control*, Morgan & Claypool Publishers.
- Koenig, S. et al., 2004. Incremental heuristic search in AI. *AI Magazine*, v. 25, n. 2, pp. 99-112.
- Russell, S. e Norvig, P., 2004. Busca com informação e exploração, In: *Inteligência Artificial*, Cap. 4, Elsevier, 11ª reimpressão, Rio de Janeiro, pp. 94-133.
- Sniedovich, M., 2006. Dijkstra's algorithm revisited: the dynamic programming connexion, *Control and Cybernetics*, v. 35, n. 3, pp. 599-620.
- Soltani, A.R., Tawfik, H., Goulermas, J.Y. e Fernando, T., 2002. Path planning in construction sites: performance evaluation of the Dijkstra, A*, and GA search algorithms. *Advanced Engineering Informatics*, v. 16, n. 4, pp. 291-303.
- Thompson, P.A. e Marchant, E.W., 1995. A computer model for the evacuation of large building populations, *Fire Safety Journal*, v. 24, pp. 131-148.