

Simulação de Aplicação de Armadilhas no Combate ao *Aedes aegypti*

Alessandro M. Baldi¹, Eduardo Zambon¹, Patrícia D. Costa¹, Eduarda M. S. Montiel²

¹Departamento de Informática – Universidade Federal do Espírito Santo (UFES)
Av. Fernando Ferrari, 514 – Goiabeiras – 29.075-910 – Vitória – ES – Brazil

²Departamento de Medicina – Universidade Estadual de Ponta Grossa (UEPG)
Av. General Carlos Cavalcanti, 4748 – Uvaranas – 84.030-900 – Ponta Grossa – PR – Brazil

alessandro.baldi@aluno.ufes.br, {zambon, pdcosta}@inf.ufes.br

eduarda.montiel@gmail.com

Abstract. *The Aedes aegypti mosquito is the main vector of Dengue Fever, Zika, Chikungunya and Yellow Fever diseases. This article describes a computer simulation of the mosquito behavior and of the mosquito traps effectiveness in a predefined geographic setting. The computer simulation models three fundamental aspects: (i) the reproduction of mosquitoes in foci, (ii) the growth of mosquito population, and (iii) the combat of vector through traps. The main goal of this simulation is to verify the most suitable places for the deployment of traps to combat mosquitoes, in order to decrease the chances of an epidemic outbreak. The main result of this work is a tool that contributes to the planning of vector combat, improving the effectiveness of prevention.*

Resumo. *O mosquito Aedes aegypti é o principal vetor das doenças Zika, Dengue, Chicungunha e Febre Amarela. Este artigo descreve a pesquisa em simulação computacional sobre o comportamento deste mosquito e sobre a eficácia de armadilhas em um certo cenário geográfico predefinido. A simulação computacional modela três aspectos fundamentais: (i) a reprodução dos mosquitos em focos, (ii) o crescimento da população de mosquitos, e (iii) o combate do vetor através de armadilhas. O objetivo principal desta simulação é a verificação dos locais mais adequados para a implantação de armadilhas com o propósito de combater os mosquitos, diminuindo as possibilidades de uma epidemia. Como resultado do trabalho foi desenvolvida uma ferramenta que contribui para o planejamento do combate ao vetor, melhorando a eficácia das ações de prevenção.*

1. Introdução

A dinâmica da transmissão de doenças virais como Dengue, Febre Amarela, Chicungunha e Zika é complexa e tem motivado pesquisadores a realizar estudos sobre os múltiplos fatores relacionados à circulação e persistência dessas doenças nas comunidades humanas [Morato et al. 2005, Vasconcelos 2015]. O aparecimento e o ressurgimento dessas doenças arbovirais transmitidas pelo *Aedes aegypti* e *Aedes albopictus* continuam a ser uma grande ameaça nos trópicos e subtrópicos [Braga and Valle 2007, Sivagnaname et al. 2012]. Um dos desafios que o Brasil enfrenta atualmente é a vigilância

epidemiológica para reconhecer precocemente áreas com transmissão de doenças e minimizar o impacto na população [Vasconcelos 2015, Johnson et al. 2017], uma vez que a associação entre os índices atualmente utilizados e a transmissão dessas doenças não têm provado ser satisfatoriamente preditivas das epidemias [Sivagnaname et al. 2012].

A expectativa dos pesquisadores da área é de que índices preditivos confiáveis e úteis possam vir a ser desenvolvidos com a ajuda de ferramentas entomológicas eficientes e apropriadas. Embora se tenha registrado algum progresso neste domínio, ainda há muito a aprender para garantir uma vigilância epidemiológica mais eficaz [Sivagnaname et al. 2012, Morato et al. 2005]. Nesse sentido, o objetivo principal deste artigo é descrever um sistema de simulação computacional utilizando a ferramenta *Groove* (*G*Raph based *O*bject-*O*riented *V*erification) [Ghamarian et al. 2012, Rensink 2003] de modo a permitir uma análise dos locais mais adequados para a implantação de armadilhas para o controle populacional do mosquito *Aedes*.

2. O Combate ao Mosquito *Aedes*

As doenças arbovirais estão emergindo e ressurgindo em diferentes partes do mundo. A incidência de Dengue, por exemplo, aumentou 30 vezes com o crescimento da expansão geográfica para novos países. Estima-se que 50 milhões de casos de Dengue ocorrem anualmente e aproximadamente 2,5 bilhões de pessoas vivem em países endêmicos para Dengue [Sivagnaname et al. 2012, Johnson et al. 2017]. O controle vetorial tem sido considerado um importante aliado na prevenção e controle das infecções. Como sugerido em [Sivagnaname et al. 2012], reduzir a densidade dos vetores para níveis baixos é a única medida atualmente disponível para prevenir a transmissão de doenças como a Dengue. Para a Organização Mundial da Saúde (OMS) há uma necessidade urgente de novas ferramentas de controle de vetores para o equilíbrio das populações de *Aedes* [WHO 2017].

Algumas novas e promissoras ferramentas de controle de vetores são objeto de pesquisa e atualmente estão sendo testadas em campo para seu uso em intervenções de saúde pública. Dentre essas ferramentas, destacam-se os materiais tratados com inseticida (ITMs), que consistem em redes insecticidas de longa duração, cortinas e tapeçarias; e as armadilhas de oviposição letais (*ovitraps*), que atraem e matam as fêmeas que realizam a oviposição no local, assim como seus descendentes que ficam presos na armadilha [Lok et al. 1977, Johnson et al. 2017]. Essas armadilhas necessitam de visitas periódicas dos agentes de saúde para a contagem manual dos vetores. Com isso, é possível identificar localidades com maior concentração dos mosquitos para realizar ações efetivas e centralizadas no combate às doenças transmitidas pelos vetores.

A simulação proposta neste artigo modela o comportamento do mosquito *Aedes* e da armadilha para o mosquito. Dessa forma, consegue-se simular ações de combate e verificação da população de mosquitos em uma determinada região. As próximas seções detalham o cenário de simulação e a implementação da simulação.

3. Cenário da Simulação

Utilizando-se o panorama de uma determinada região (um bairro ou uma cidade, por exemplo), o cenário da simulação é formado por casas, mosquitos, ovos, agentes de combate ao vetor e armadilhas.

As casas representam os locais em que os mosquitos podem fazer a oviposição e que podem possuir focos (criadouros do mosquito) ou armadilhas. No caso do mosquito fazer a oviposição em focos, há a evolução dos ovos para novos mosquitos após 20 dias [Maimusa et al. 2016]. De outra forma, a oviposição em armadilhas faz com que o mosquito seja morto e os ovos não evoluam para novos mosquitos [WHO 2017, Sivagnaname et al. 2012]. O agente de saúde, ao fazer uma verificação dois dias após a oviposição do mosquito, consegue observar que há ovos e que é necessário um controle vetorial do mosquito na região em que a armadilha está presente [Dzul-Manzanilla et al. 2016]. O controle consiste em fumacê e visita domiciliar, que faz com que ovos, mosquitos e focos presentes na região sejam exterminados [Dzul-Manzanilla et al. 2016].

Os mosquitos sobreviventes permanecem fazendo a oviposição por até 37 dias [Maimusa et al. 2016]. Durante a sua vida, os mosquitos ficam livres para sobrevoar e fazer a oviposição em até 100 metros do seu ponto de nascimento [Dzul-Manzanilla et al. 2016]. Fenômenos climáticos também estão no cenário, de forma que os focos combatidos possam voltar após uma determinada chuva, por exemplo.

A simulação foi construída baseando-se em fatos científicos reais, de forma que se aproxima ao que acontece na vida real. Nas próximas seções, há o detalhamento da ferramenta utilizada e implementação da simulação.

4. A Ferramenta *Groove*

O *Groove* (GRaph based Object-Oriented VERification) é um *software* desenvolvido na Universidade de Twente (Holanda), utilizado para verificação e simulação de sistemas. O *software* utiliza grafos para representar os estados do sistema e aplica regras de reescrita (produção/transformação) de grafos para modelar a dinâmica do sistema [Rensink 2003, Rensink and Zambon 2017].

A ferramenta possui um editor para criar regras de produção de grafos, um simulador para visualizar as transformações dos grafos introduzidas pelas regras de produção e um gerador para explorar espaços de estados automaticamente [Rensink and Zambon 2017]. Adicionalmente, o *Groove* trabalha com o formato aberto de arquivo GXL (*Graph eXchange Language*), que provê uma forma programática padrão para salvar os respectivos grafos.

Inicialmente, para se modelar uma simulação no *Groove* é necessária a implementação de um cenário inicial como um grafo de entrada. Os grafos manipulados pelo *Groove* são formados por nós e arestas direcionadas. Arestas são anotadas com *rótulos* que caracterizam a relação entre os nós. Além disso, nós podem possuir atributos (valores inteiros e Booleanos) que descrevem certas propriedades de cada nó.

Os grafos são transformados pelo *Groove* através de regras. Uma regra também consiste de um grafo, mas elementos (nós e arestas) da regra são subdivididos em subgrafos *L* e *R*. O subgrafo *L* pode ser visto como a pré-condição para a aplicação da regra, e o grafo *R* como o efeito da aplicação. Uma regra é aplicada sobre um grafo hospedeiro *G* (isto é, o grafo que sofrerá a transformação). Essa aplicação requer encontrar uma correspondência de *L* em *G*, substituindo-a pelo subgrafo *R*, formando um novo grafo *H* [Ramos et al. 2016].

A figura 1 apresenta a sintaxe e semântica do *Groove* na aplicação de regras. O papel dos nós e arestas em uma regra é identificado pelo tipo de tracejado na aresta e no entorno do nó. Elementos em tracejado fino (o nó indicado por “nó3” e a aresta com rótulo “aresta2” na figura 1) são elementos que devem estar presentes no estado atual da simulação e que são removidos quando a regra é aplicada. De forma similar, elementos em traço comum fino (“nó1”, “nó2” e “aresta1”) também são elementos que devem estar presentes no estado atual mas estes são preservados pela aplicação da regra. Elementos em tracejado grosso (“nó5” e “aresta4”) não podem ocorrer no estado atual, caso contrário a regra não é aplicável. Finalmente, elementos em traço comum grosso (“nó4” e “aresta3”) são criados pela regra.



Figura 1. Sintaxe e Semântica de Regras de Reescrita no Groove

5. Modelagem da Simulação

Essa seção apresenta os detalhes da simulação implementada no *Groove*. O cenário de simulação é feito em rodadas, nas quais as ações dos mosquitos e dos agentes são realizadas. Ao final de cada rodada, um contador de dias é incrementado até que um dado limite seja atingido.

A figura 2 exibe os tipos de nós utilizados na simulação. O elemento “casa” (figura 2a) possui arestas “voar” que ligam outros elementos “casa”, indicando a possibilidade do mosquito voar de uma casa a outra. Também há possibilidade de uma casa possuir nós “ovos” (figura 2d) ou “mosquitos” (figura 2c). “Mosquito” (figura 2c) possui uma aresta indicando a casa em que nasceu, dois atributos Booleanos indicando se já realizou oviposição e voou em uma rodada e um inteiro indicando a idade do mosquito em dias. O elemento “agente” (figura 2b) foi especificado de forma a sempre estar em alguma casa e possui um contador de visitas para cada casa visitada. “Ovos” (figura 2d) apresenta a idade em dias e um controle da atualização da idade por rodada.

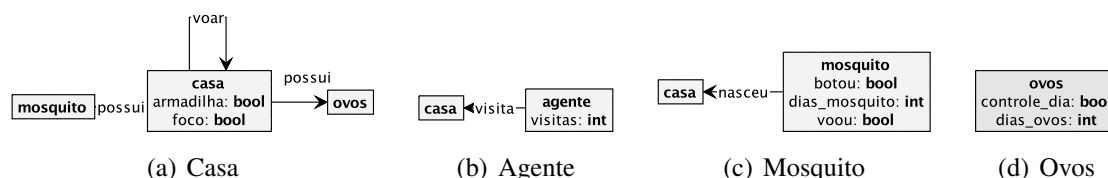


Figura 2. Tipos

5.1. Cenário Inicial

De acordo com o que foi apresentado na seção 4, inicialmente é definido o cenário de execução da simulação, especificando as casas, as conexões entre as casas para vôo do mosquito e a localização de mosquitos/ovos iniciais. As figuras 3a e 3b, mostram um exemplo simples de cenário inicial, utilizando uma casa com um foco, uma casa com uma

armadilha e uma casa com um mosquito. Ao lado há a respectiva modelagem no *Groove*. Convém observar que na modelagem há a indicação do local em que o mosquito nasceu (figura 3b) e os locais em que o mosquito pode voar, indicados pelas arestas “voar”.

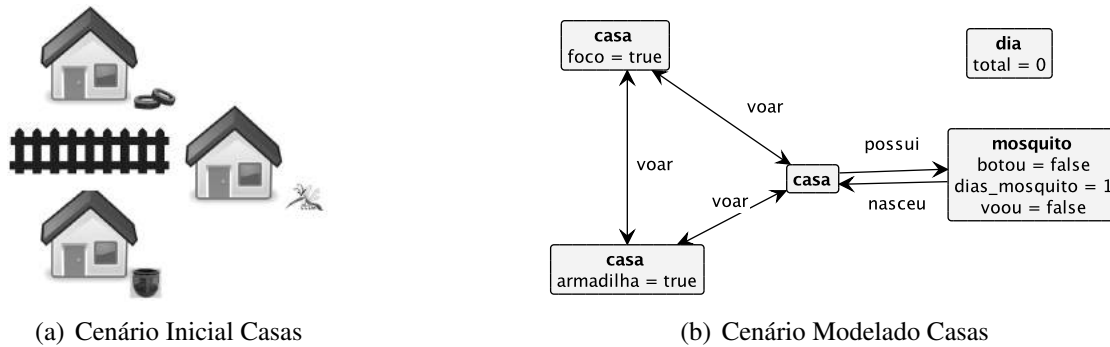


Figura 3. Cenário Inicial

Vale notar que cada caso de estudo requer uma modelagem de cenário inicial nova. Assim, para melhorar a usabilidade do protótipo, foi feita uma integração do *Groove* com o *OpenStreetMap*, uma plataforma aberta e colaborativa de mapeamento das ruas que possui uma API para utilização gratuita [OpenStreetMap 2017]. Utilizando a ferramenta “Exportar” presente na interface principal do *OpenStreetMap*, é possível extrair partes de um mapa para o formato OSC.XML. Tal formato indica os “waypoints” (coordenadas geográficas de pontos específicos no mapa) de todas as construções em uma determinada área. Uma casa com um plano quadrado, por exemplo, possui quatro “waypoints” no *OpenStreetMap* com as coordenadas de cada um dos vértices.

A integração com o *Groove* se dá a partir da conversão para arquivos GXL, formato de armazenamento utilizado pelo *Groove*. É feito um cálculo central do “waypoint” de cada uma das construções no mapa, com a longitude e latitude conhecida para o centro de cada uma das construções, sendo cada construção transformada em um nó “casa”. Utilizando também a longitude e latitude do centro, é calculada a distância para outros nós, criando as arestas “voar” entre as casas que se encontram a menos de 100 metros de distância. Com essa integração é possível verificar o avanço do *Aedes aegypti* utilizando dados do mundo real, facilitando o trabalho de modelagem do grafo inicial.

A figura 4 representa a região da Universidade Federal do Espírito Santo no *OpenStreetMap*. A mesma região foi convertida para o *Groove* (figura 5), indicando todas as construções da área (nós) e as possibilidades de vôo do *Aedes aegypti* entre as construções (arestas “voar”). Convém destacar que embora o grafo fique bastante confuso visualmente, isto não representa um problema pois o grafo é manipulado de forma automática pela ferramenta, e a conversão de formatos também é automática.

Utilizando o grafo inicial apresentado na figura 5, foi desenvolvido um estudo utilizando o simulador para a colocação de armadilhas em determinadas áreas. Na simulação, colocou-se um único mosquito *Aedes aegypti* na construção que fica ao centro do mapa: o restaurante universitário. As armadilhas na simulação foram manualmente colocadas a uma certa distância do local em que o mosquito nasceu. Essa distância foi variada durante os experimentos, conforme descrito na seção 6.



Figura 4. OpenStreetMap

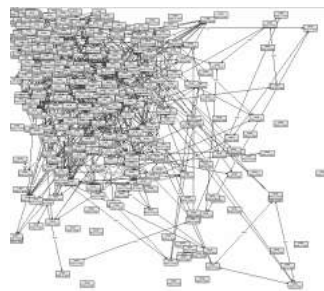


Figura 5. Exportação para o Groove

5.2. Regras da Simulação

Na simulação há um controle temporal do que está acontecendo por rodada em dias, impedindo por exemplo que um mesmo nó “mosquito” bote ovos duas vezes no mesmo dia. O controle é feito a partir de um conjunto de regras que faz a contagem de dias, obrigando que todas as outras regras sejam executadas antes da próxima sequência de dia.

De uma forma geral a simulação segue o seguinte algoritmo.

enquanto dia atual $\leq$ total dias simulação:

 A cada dia:

 Atualização do dia dos ovos/mosquitos;

 Vôo do mosquito, deposição dos ovos.

 Após 2 dias de ovos na armadilha:

 Surgimento de agentes e combate aos focos/ovos/mosquitos.

 A cada 15 dias: volta dos focos desativados (simulação de chuva).

 Após 20 dias de postura dos ovos: eclosão de novos mosquitos.

 Após 38 dias: falecimento do mosquito.

São apresentadas aqui quatro regras da simulação, de forma a ilustrar a implementação feita no *Groove*. Por limitações de espaço, as demais regras não são mostradas nesse texto mas são similares às selecionadas para discussão. O conjunto completo de regras da simulação está disponível no endereço apresentado na seção 8.

O vôo do mosquito é regido por regras de vôo, uma delas é a “Voo”, mostrada na figura 6a. Um mosquito é selecionado pela regra e o seu vôo é indicado através da remoção e inclusão da aresta “possui” de uma casa antiga para outra casa. Aqui vale destacar que a seleção do mosquito e das casas é feita pelo *Groove* de forma aleatória, a cada aplicação da regra. Com isso, é possível obter comportamentos variados em diferentes execuções da simulação. Ainda na regra da figura 6a, a idade do mosquito é incrementada após o seu voo, e um *flag* Booleano é atualizado para impedir que um mesmo mosquito seja selecionado para voar novamente nesta mesma rodada.

A regra “Eclode”, exibida na figura 6b, especifica que cada nó “ovos” com 20 dias de rodada vira um nó “mosquito”, ligado por uma aresta “nasceu” ao nó “casa” de nascimento. Novamente por limitações de espaço para apresentação, a figura 6b na verdade exhibe somente um fragmento da regra, ilustrando a criação de um único mosquito a partir de um conjunto de ovos. Na regra completa implementada de fato na simulação, são criados 10 novos mosquitos. Um novo nó “mosquito” é criado com três atributos: um Booleano “botou” para verificação se o determinado mosquito já colocou ovos no dia, um Booleano “voou” para verificação de vôo no dia e um inteiro “dias_mosquito” para controle da idade do mosquito.

“Armadilha” é a regra apresentada na figura 6c que faz o controle das armadilhas, criando um nó “agente” quando o nó “ovos” está no terceiro dia, através do controle no inteiro “dias_mosquito”. O agente possui um inteiro “visitas” que faz o controle decrescente de visitas em cinco nós (ou seja, em casas distantes no máximo 500 metros).

Uma das regras de combate do agente é a “Combate Ovos” que está apresentada na figura 6d. A regra faz a exclusão do nó “ovos” quando o nó “agente” estiver no nó “casa” respectivo. Além do combate a ovos, existem duas regras de combate semelhantes que combatem mosquitos e focos.

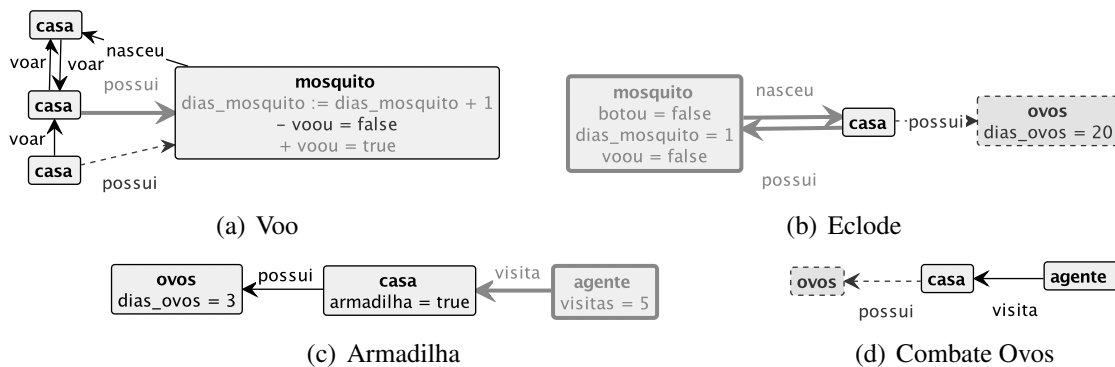


Figura 6. Exemplos de Regras da Simulação

5.3. Critérios de Parada

Conforme descrito na seção anterior, a simulação no *Groove* pode ser configurada para parar em um determinado dia e apresentar o cenário completo deste dia. Além disso, a simulação também termina quando uma das duas condições abaixo for satisfeita.

1. Mosquito e ovos totalmente combatidos: quando os mosquitos e ovos não existem mais no cenário, ou seja, quando não houver nenhum nó “mosquito” ou “ovos”, a execução é interrompida pois o mosquito foi erradicado.
2. Mosquito ou ovos em todas as casas: a execução é interrompida quando há um cenário de epidemia, ou seja, quando há um nó “mosquito” ou “ovos” em todas as casas.

6. Resultados

O simulador foi configurado para executar o ambiente durante 30 dias em uma exploração do espaço de estados de forma linear randômica [Ghamarian et al. 2012]. Foram executados testes com nenhuma armadilha, uma a três armadilhas posicionadas a 100 metros e uma a duas armadilhas a 200 metros do ponto de origem do mosquito. A tabela da figura 7 mostra o resultado da média de 10 execuções completas (por 30 dias) da simulação. Nas colunas “Mosquito” e “Ovos” há o pior caso, isto é, o caso em que não houve total combate do mosquito *Aedes aegypti*. Já a coluna “Combate Total” exhibe o número médio de dias necessários para o combate total (erradicação) do mosquito. Esse resultado representa o melhor caso e também foi calculado como a média entre 10 execuções da simulação que terminaram com erradicação.

Armadilhas	100 metros			200 metros		Sem Armadilhas	
	Mosquitos	Ovos	Combate Total	Mosquitos	Ovos	Mosquitos	Ovos
1	3,72	339	4,7 dias	46,6	2057	70,5	3692,5
2	0,8	93	3,9 dias	36,2	1875		
3	0	39	3 dias				

Figura 7. Tabela Geral de Resultados

Os valores da tabela mostram que o aumento do número de armadilhas reduziu a população de mosquitos e ovos, tanto para as distâncias de 100 quanto de 200 metros. No cenário de 100 metros é interessante notar que o combate total foi efetuado e que diminuiu à medida em que mais armadilhas foram incluídas. Para 200 metros, o combate total não foi encontrado durante as execuções.

Outro resultado interessante é que a disposição das armadilhas e dos mosquitos no cenário alteram significativamente os resultados finais. A armadilha posicionada a uma distância de 200 metros apresenta um resultado pior que 100 metros, dando liberdade ao mosquito na continuação da reprodução, aumentando o risco de uma epidemia no cenário.

É interessante notar que os resultados mostram que há uma diminuição da população de mosquitos e ovos independentemente da posição ou do número de armadilhas, ou seja, apenas uma armadilha faz diferença no combate do mosquito.

7. Trabalhos Correlatos

Uma simulação utilizando o *Groove* também pode ser vista na modelagem de uma colônia de formigas [Ghamarian et al. 2012]. Nesse trabalho, essa simulação foi utilizada como *benchmark* do desempenho da ferramenta, e mostrou que o *Groove* pode ser uma interessante ferramenta para construção de simuladores.

Uma armadilha da dengue do tipo *ovitrap* pode ser vista no desenvolvimento de sistemas geográficos para acompanhamento dos agentes de saúde [Moraes et al. 2015]. Neste trabalho, os autores desenvolveram um sistema em que o usuário faz o monitoramento da armadilha e envia os dados para um mapa. Utilizando a ideia destes autores, algo similar poderia ser integrado ao simulador aqui descrito, incluindo a recepção de dados em tempo real, com obtenção de informações das armadilhas e da cidade para que os agentes de saúde consigam descobrir uma melhor forma na distribuição das armadilhas, prevendo uma possível epidemia do vetor.

Uma outra abordagem utilizando a tecnologia no combate à dengue é [Fiorini et al. 2016] em que os autores utilizam a mineração em redes sociais para a aplicação dos dados no combate ao mosquito *Aedes aegypti* e *Aedes albopictus*. Utilizando a mesma ideia dos autores, dados da mineração poderiam ser utilizados neste trabalho de forma a integrar dados do cenário de entrada para fazer previsões utilizando o simulador.

Em [Bacani et al. 2015] é feita uma modelagem para a dinâmica populacional do *Aedes aegypti* via simetrias de Lie. Esse texto é outro trabalho descrevendo a dinâmica vital e espacial de vôo do *Aedes aegypti* e sua relação com a população humana. As equações dessa pesquisa poderiam ser integradas no simulador aqui descrito para maior precisão do vôo do mosquito.

Em [Lima et al. 2014] é apresentado um *framework* especializado na simulação do *Aedes aegypti*. Construído sobre o *TerraMe* com dados geográficos, o *framework* permite a construção de intervenções para auxiliar no combate à dengue. O *framework* leva em conta a simulação do ciclo de vida do mosquito, densidade demográfica humana, mobilidade humana, terreno da região e transmissão da dengue [Lima et al. 2014]. Por outro lado, o *framework* não leva em conta a intervenção por armadilhas como especificada no presente artigo.

8. Conclusão e Trabalhos Futuros

A modelagem da simulação captura de forma simples o comportamento do mosquito *Aedes aegypti*, vetor de infecções virais. De acordo com nossos conhecimentos, não existem outras ferramentas que simulem o comportamento epidemiológico do *Aedes aegypti* e levem em conta a utilização de uma armadilha como a desta pesquisa. Além disso, a implementação no *Groove* pode ser adaptada a diferentes contextos e pesquisas. Dessa forma, órgãos responsáveis pela saúde e controle vetorial podem fazer o uso da ferramenta para analisar áreas mais vulneráveis, planejando ações e inserindo novas informações no simulador.

Vale destacar que é possível construir cenários de simulação reais e complexos, baseados em informações de mapas e outras bases de dados. Isso foi ilustrado pela integração feita com o *OpenStreetMap* proposta por este artigo. De forma similar, é possível criar programas adicionais que façam a extração de dados de outras fontes (informações de agentes de saúde, mineração de dados, etc.) e convertam os dados em arquivos GXL para simulação no *Groove*. A mineração de dados de notícias, por exemplo, pode informar regiões com o vetor e adicioná-las automaticamente ao simulador, aumentando a precisão na simulação.

Em um próximo passo da pesquisa, há o interesse de se tornar a simulação mais precisa, com a inclusão de novos elementos do contexto, tais como: condições do tempo (temperatura, clima, etc), hora do dia, quantidade populacional, movimentação populacional, etc. Além desses elementos, busca-se a obtenção também de resultados a respeito da transmissão das doenças ao inserir vírus nos vetores, o que tornaria possível a análise da propagação de doenças. Finalmente, ao invés da colocação manual das armadilhas para simular os acontecimentos, o simulador também poderia levantar os melhores pontos na região para a inserção das armadilhas.

A simulação do *Aedes aegypti* em um cenário com armadilhas através da ferramenta *Groove* mostrou-se importante para o desenvolvimento de uma pesquisa mais ampla na área. Ela favorece a integração de informações a respeito de uma região com a possibilidade de instalação da armadilha em locais adequados. A implementação da simulação no *Groove* e do código que faz a conversão de mapas do *OpenStreetMap* para grafos estão disponíveis em código aberto no link <https://github.com/alexnede/AedesSimulator>.

Referências

Bacani, F., Freire, I. L., Maidana, N. A., and Torrisi, M. (2015). Modelagem para a dinâmica populacional do *aedes aegypti* via simetrias de Lie. *Proceeding Series of the Brazilian Society of Computational and Applied Mathematics*, 3(1).

- Braga, I. A. and Valle, D. (2007). *Aedes aegypti*: histórico do controle no Brasil. *Epidemiologia e serviços de saúde*, 16(2):113–118.
- Dzul-Manzanilla, F. et al. (2016). Indoor resting behavior of *aedes aegypti* (diptera culicidae) in Acapulco, Mexico. *Journal of Medical Entomology*.
- Fiorini, S. T. et al. (2016). Vigilância entomológica da dengue, zika e chikungunya: Uma solução baseada em redes sociais e dispositivos móveis. *16º Workshop de Informática Médica*.
- Ghamarian, A. H., de Mol, M., Rensink, A., Zambon, E., and Zimakova, M. (2012). Modelling and analysis using GROOVE. *International journal on software tools for technology transfer*, 14(1):15–40.
- Johnson, B. J., Ritchie, S. A., and Fonseca, D. M. (2017). The state of the art of lethal oviposition trap-based mass interventions for arboviral control. *Insects*, 8(1):5.
- Lima, T. et al. (2014). A framework for modeling and simulating *aedes aegypti* and dengue fever dynamics. In *Proceedings of the 2014 Winter Simulation Conference*, pages 1481–1492. IEEE Press.
- Lok, C., Kiat, N., and Koh, T. (1977). An autocidal ovitrap for the control and possible eradication of *aedes aegypti*. *The Southeast Asian journal of tropical medicine and public health*, 8(1):56–62.
- Maimusa, H. A., Ahmad, A. H., Kassim, N. F. A., and Rahim, J. (2016). Age-stage, two-sex life table characteristics of *aedes albopictus* and *aedes aegypti* in Penang Island, Malaysia. *Journal of the American Mosquito Control Association*, 32(1):1–11.
- Moraes, L. et al. (2015). Sistema web de baixo custo geolocalizador para prevenção de casos da dengue. *Seminário Integrado de Software e Hardware*.
- Morato, V. et al. (2005). Infestation of *aedes aegypti* estimated by oviposition traps in brazil. *Revista de saude publica*, 39(4):553–558.
- OpenStreetMap (2017). OpenStreetMap. <http://www.openstreetmap.org>.
- Ramos, A. S., Zambon, E., and Boeres, M. C. S. (2016). Aplicação de métodos de busca guiada na exploração do espaço de estados de gramáticas de grafos. In *Anais do XLVIII - Simpósio Brasileiro de Pesquisa Operacional*. SOBRAPO – Sociedade Brasileira Pesquisa Operacional.
- Rensink, A. (2003). The GROOVE simulator: A tool for state space generation. In *International Workshop on Applications of Graph Transformations with Industrial Relevance*, pages 479–485. Springer.
- Rensink, A. and Zambon, E. (2017). GROOVE - GRaphs for Object-Oriented VERification. <http://groove.cs.utwente.nl>.
- Sivagnaname, N., Gunasekaran, K., et al. (2012). Need for an efficient adult trap for the surveillance of dengue vectors. *Indian Journal of Medical Research*, 136(5):739.
- Vasconcelos, P. F. d. C. (2015). Doença pelo vírus zika: um novo problema emergente nas américas? *Revista Pan-Amazônica de Saúde*, 6(2):9–10.
- WHO (2017). World Health Organization: Dengue Control. <http://www.who.int/denguecontrol/research/en/>.