Universidade Estadual de Campinas
Instituto de Computação

Carlos Victor Dantas Araújo

Dengue Control

CAMPINAS
2025

Carlos Victor Dantas Araújo

Dengue Control

Tese apresentada ao Instituto de Computação da Universidade Estadual de Campinas como parte dos requisitos para a obtenção do título de Doutor em Ciência da Computação.

Thesis presented to the Institute of Computing of the University of Campinas in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Supervisor/Orientador: Prof. Dr. Fábio Luiz Usberti

Este exemplar corresponde à versão da Tese entregue à banca antes da defesa.

CAMPINAS
2025

Na versão final esta página será substituída pela ficha catalográfica.

De acordo com o padrão da CCPG: “Quando se tratar de Teses e Dissertações financiadas por agências de fomento, os beneficiados deverão inserir no formulário de solicitação da ficha catalográfica as informações sobre agência e número do processo pelo qual receberam o auxílio, porém essa informação não ficará visível na ficha, será utilizada para fins estatísticos. Os auxílios recebidos devem também ser citados na seção “Agradecimento” seguindo os padrões exigidos pelas agências de fomento.”
e
“caso a tese de doutorado seja feita em Cotutela, será necessário informar na ficha catalográfica o fato, a Universidade convenente, o país e o nome do orientador.”

Na versão final, esta página será substituída por outra informando a composição da banca e que a ata de defesa está arquivada pela Unicamp.

Acknowledgements

Os agradecimentos devem ocupar uma única página.

Resumo

O resumo deve ter no máximo 500 palavras e deve ocupar uma única página.

Abstract

The abstract must have at most 500 words and must fit in a single page.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

Dengue is a mosquito-borne viral infection primarily transmitted by *Aedes aegypti* mosquitoes. The disease, caused by any of the four distinct serotypes of the Dengue virus, poses a significant global public health challenge [63]. Environmental and ecological changes, such as rising temperatures, increased urbanization, and changing precipitation patterns, have expanded the range of *Aedes aegypti*, contributing to the greater geographic spread of Dengue. Since its emergence in the late 18^{th} century in Asia and the Pacific, Dengue has become endemic in many regions, with approximately half of the world's population currently living in areas at risk [25, 47]. Factors such as rapid population growth, unplanned urban development, inadequate sanitation, and healthcare inequality play a central role in the persistence and resurgence of dengue.

In the last year, 2024, Brazil reported approximately 10.239.883 cases of Dengue, accounting for $78,62\%$ of the world's cases and $92,10\%$ of the South American cases [7]. Of the reported cases, 10.231.692 were confirmed in the laboratory, 8.191 were classified as severe, and 6.161 resulted in death. According to the epidemiological report of the Brazil Health Departments up to April 20, 2024, more than 11 states and 465 cities declared a state of emergency [20]. By 28 February 2025, Brazil had reported 662.224 cases of Dengue, which represents 86.43% of the World's cases. Figure 1.1 displays a graph that shows the number of Dengue cases in the world, South America, and Brazil from 1980 to 2024.

The disease imposes substantial social and economic burdens, affecting not only the health and well-being of individuals but also the broader economic productivity and healthcare system. The financial impact of Dengue in Brazil is profound, for example, annual expenditures on Dengue prevention and control exceed BRL 1.6 billion [47]. The direct costs include healthcare expenses for hospitalization, medical treatment, and public health campaigns aimed at controlling mosquito populations [49]. Indirect costs, such as lost productivity due to illness and the long-term effects of severe Dengue cases, add to the financial strain. Moreover, the social consequences are equally alarming, with communities enduring the disruption of daily life, increased anxiety over disease outbreaks, and the loss of lives.

Despite the severity of the situation, most Brazilian municipalities continue to make crucial decisions in combating Dengue without the support of advanced computational tools [51]. The current approach is largely reactive, relying on traditional mosquito control

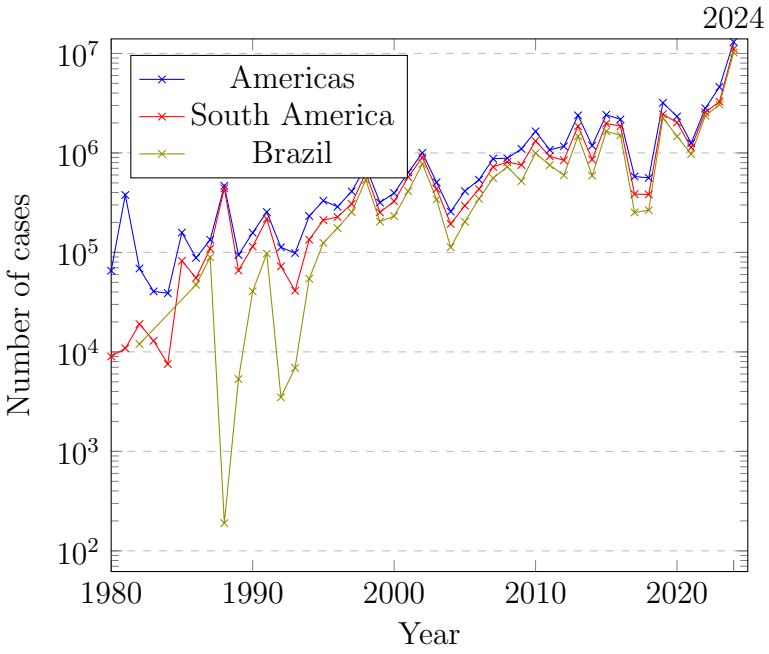


Figure 1.1: Dengue reported cases from 1980 to 2024 [52].

methods, including eliminating adult mosquitoes, their breeding sites and public health interventions. These actions are often insufficient to address the complexity and scale of the problem. This lack of computational support means that decisions regarding resource allocation, strategic planning, and the deployment of interventions are not optimized, leading to possible inefficiencies and potentially exacerbating the spread of the disease [27, 68, 58].

The most common approach to combating Dengue spread is employing chemical control methods, such as insecticides, which manage the mosquito population in both the larval and adult stages [51]. The World Health Organization (WHO) provides technical and operational standards for pesticide experts to ensure the safe use of insecticides in public health. These standards specify the active ingredients and dosages for various treatments. During disease outbreaks, emergency responses in urban areas usually involve the dispatch of spraying vehicles to apply insecticides (see Figure 1.2). It is essential to use insecticides carefully and responsibly in vector control activities, as indiscriminate use can have significant environmental impacts and contribute to developing resistance in mosquitoes [45, 67].

The sprayed insecticide has no residual effect and it is strongly influenced by wind and obstacles along the streets. The best effect is achieved when the densest insecticide cloud is at a distance of at most 100 meters from the equipment [51]. As this distance is crossed, the effectiveness decreases, as a consequence of droplet drift influenced by factors of the environment. The cloud dispersion is illustrated in Figure 1.3.

Insecticide application instructions are generally based on ideal topology conditions, locality structure, and favorable winds. The operation is often affected by unpaved roads, the presence of high walls, and high vegetation, in addition to headwinds. The application methodology must consider these limitations to obtain a good impact on the vector

Figure 1.2: Nebulizer equipment attached to vehicle [60].

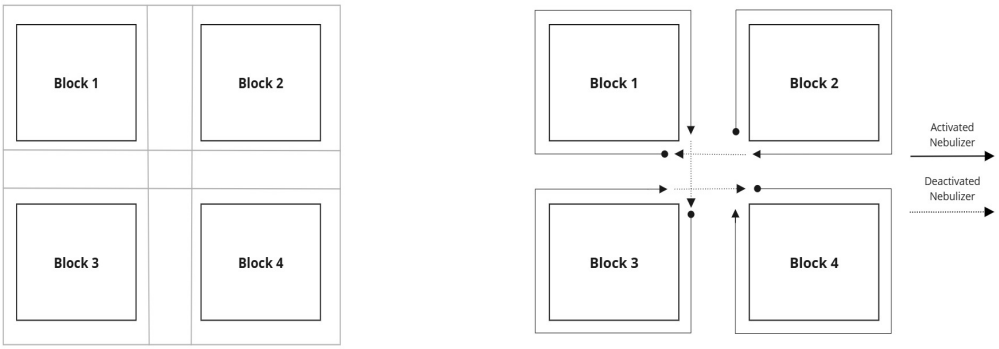
Figure 1.3: Cloud dispersion of the application [51].

population. Once a spraying vehicle begins to service a city block, it must sequentially cover all surrounding streets in a clockwise direction. The traversal ensures that the insecticide fog forms a continuous barrier, preventing mosquitoes from escaping. The clockwise direction is due to real operational factors, as the nebulizer equipment points to the right side of the vehicle.

Figure 1.4a shows an example of a map, with four blocks to service, and Figure 1.4b presents a spraying route for these blocks where the nebulizer is activated in the black dots and follows the direction of the arrow, starts by serving Block 2, goes to Blocks 1, 3 and 4, respectively.

As a result, health authorities with limited budgets must make two key decisions: first, selecting which city blocks should receive insecticide spraying to maximize vector suppression; and second, optimizing vehicle routes to ensure the efficient use of available resources. In this context, there is a clear and urgent need to integrate computational models and decision-support systems into Dengue control strategies. Such tools can help municipalities better predict outbreaks, optimize resource use, and implement more effective mosquito control measures.

In operations research, routing problems are typically classified based on where the service is performed. When services are provided at specific locations (nodes), they fall



(a) Street blocks example. (b) Spraying route example.

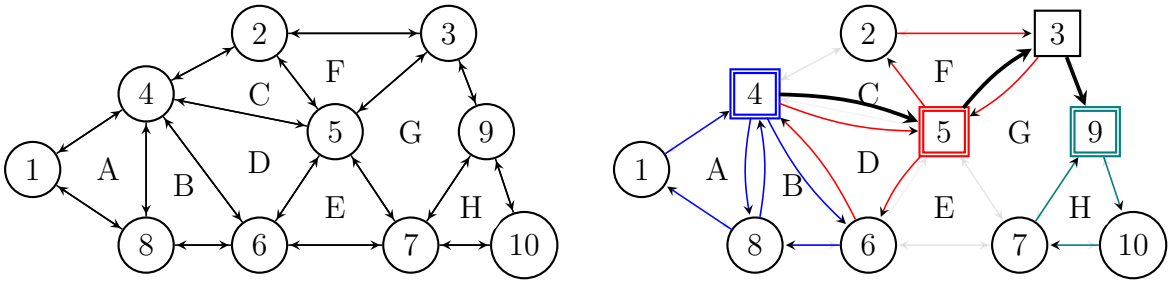
Figure 1.4: Vehicle pattern with nebulizer [51].

under Vehicle Routing Problem (VRP) [13]. When services are conducted along edges (or arcs), they are categorized as Arc Routing Problem (ARP) [18]. The routing challenge in Dengue control exhibits the characteristics of both VRP and ARP. Each city block can be represented as a *super-node* (similar to a VRP), while spraying occurs along the surrounding arcs, aligning with ARP features. Given this hybrid structure, we introduce and explore a new problem, titled the City Block Routing Problem (CBRP).

The CBRP aims to optimize the servicing of city blocks within an urban street network by assigning spraying vehicles. However, due to limited operational resources, only a subset of city blocks can be serviced. A city block is considered serviced if a spraying vehicle completely encircles it without detours, and each serviced block contributes a predefined benefit. Thus, the primary objective of the CBRP is to identify the subset of blocks whose servicing yields the highest aggregate benefit, alongside determining the optimal vehicle routes to achieve this objective within the available resources.

The Figure 1.5 illustrates an example of an input graph (city map) and a corresponding CBRP feasible solution. In Figure 1.5a, the circles represent the graph nodes, while the labels from A to H indicate the city blocks. Figure 1.5b depicts a feasible solution to the CBRP. In this solution, the nodes used in the route are marked as squares and connected by black arrows (nodes 4, 5, 3, and 9). Squares with a double border denote nodes that act as starting points to serve at least one block. The atended blocks are represented by the arcs connected to the double square with the same color, for example: node 4 (blue) serves blocks A (arcs 4→8, 8→1 and 1→4) and B (arcs 4→6, 6→8 and 8→4). Node 5 (red) serves blocks D and F, node 3 does not serve any block, and node 9 (teal) serves block H.

Given the complex and dynamic nature of urban environments, particularly in public health scenarios like Dengue control, incorporating stochastic elements into the CBRP is essential to enhance the realism and applicability of the solution. Some unpredictable behavior of mosquitoes and the spread of Dengue cases during seasons of the years highlight the need for a stochastic version of the CBRP. This leads to solutions that better reflect the variability of actual deployment scenarios, improving the adherence of the model to real-world challenges. Moreover, by accounting for probabilistic factors, stochastic approaches can prioritize the most critical city blocks under uncertain resource availability,



(a) Initial graph and blocks. (b) Route and attended blocks.

Figure 1.5: CBRP instance (a) and solution (b).

thereby increasing the effectiveness and resilience of vector control interventions.

The CBRP extends beyond its application in vector control, providing a robust framework to address a variety of urban logistics challenges. These challenges include waste collection, postal delivery, and reading of utility meters, where the primary goal is to efficiently service city blocks while managing both spatial and resource constraints. The flexibility of the CBRP makes it particularly valuable in densely populated urban environments, where the efficiency of routing decisions significantly affects both service effectiveness and operational costs.

Besides the definition of control actions [30, 35], it is important to develop visualization and decision support tools to assist the health departments. A clear understanding of the likely progression of Dengue cases can significantly enhance short-term resource allocation strategies [51]. Generating accurate predictive models for Dengue spread, especially in urban areas is highly challenging since mosquito behavior is inconsistent and depends on numerous factors. These include the availability and location of breeding sites, the mosquito population size and infection rate, the timing and location of insecticide application, frequency of rainfall, various climate conditions, and the interactions between mosquitoes, breeding sites, and human populations.

In this thesis, we developed a complete set of methodologies focused in the application of CBRP to optimize insecticide-spraying routes for Dengue control. Those methodologies compreend exact and heuristic approaches for deterministic and stochastic versions of the CBRP, instances based on real data, robust simulation based on multi-agent systems and a simulation-optimization framework that could assist health departments. This application is particularly pertinent to many Brazilian cities, where seasonal outbreaks of mosquito-borne diseases demand resource-efficient, targeted vector control strategies. By optimizing spray routes, the CBRP not only improves operational efficiency but also maximizes coverage of high-risk areas, ultimately supporting public health initiatives aimed at reducing Dengue transmission.

Chapter 2

Preliminary Concepts

In this chapter, we introduce the fundamental concepts necessary for understanding this thesis. In this work, we employ basic concepts from Combinatorial Optimization, which are assumed to be known. If the reader deems a review necessary, we recommend the textbook by Nemhauser and Wolsey [50], which covers this topic with a focus on Integer Linear Programming (ILP), one of the main tools used in this work. Basic concepts related to graph theory are also assumed to be known. Should the reader require a refresher, the material can be found in standard textbooks on the subject, such as Diestel [21].

2.1 Notations and Definitions

Let $G = (V, A, B)$ be a weighted and directed graph, where $V = \{1, \ldots, n\}$ is the set of vertices, $A = \{(i, j) : i, j \in V, i \neq j\}$ is the set of m arcs, and $B = \{b : b \subseteq V\}$ is the set of blocks. In each arc, the first vertex is the source and also the predecessor of the second vertex in the ordered pair, which is known as the destination. Each block b has an associated set of arcs $B(i, j) \subseteq A$ that can be serviced by a vehicle.

We now present a formal definition of the CBRP. Let a Plannar Graph, extracted from a real citymap, be represented as a weighted directed graph G. Each node in V represents the intersection of at least two streets and has a list of blocks $b \subseteq B$ that are associated with it. Each arc $(i, j) \in A$ has a deadheading time $t_{i,j} > 0$, a service time $t'_{i,j} > 0$ such that $t_{i,j} \leqslant t'_{i,j}$ and the block $b \in B$ that is associated with it. The input to the CBRP is defined as:

- $G = (V, A, B)$ is a weighted directed planar graph with the blocks associated with it;
- $p_b : B \to \mathbb{N}$ is a function that returns the profit for each block $b \in B$;
- $t_{i,j} : A \to \mathbb{N}$ is a function that returns the deadheading time for each arc $(i, j) \in A$;
- $t'_{i,j} : A \to \mathbb{N}$ is a function that returns the service time for each arc $(i, j) \in A$;
- T is the time limit for the vehicle to travel and service the blocks;

- $V(b)$ is the set of nodes that are associated with the block $b \in B$;
- $B(i)$ is the set of blocks that are associated with the node $i \in V$;
- $B(i,j)$ is the set of blocks that are associated with the arc $(i,j) \in A$;
- δ_i^+ is the set of arcs with destination i;
- δ_i^- is the set of arcs with source i;

The CBRP is introduced as a general framework for addressing the routing of spraying vehicles and other city block servicing problems. The objective of the CBRP is to determine an optimal traversal that services a subset of blocks within a network, maximizing the total collected benefit from each serviced block. A vehicle traverses the graph G following a route that can serve a subset of blocks within a given time limit T. This work considers two types of solution routes, depending on whether the vertices can be visited more than once: *Walk-based route*, in which any vertex (or arc) can be visited multiple times and *Path-based route*, in which no vertex appears more than once. An optimal route (walk or path) is one that maximizes the total prize collected from the serviced blocks while respecting the vehicle time limit T.

To facilitate modeling, we augment the graph G by introducing a dummy depot 0, from which the route originates, and a set of arcs $\{(0,i),(i,0) : \forall i \in V\}$ with times $t_{i,0} = t_{0,i} = 0$, $\forall i \in V$. Thus, we define the modified graph as $G' = (V' = V \cup \{0\}, A' = A \cup \{(0,i),(i,0) : \forall i \in V\})$.

We now present key properties of the CBRP. First, due to the operational restrictions, once a block b starts being serviced at a given node i, it must be fully encircled before the vehicle moves to another block. This leads to the following property:

Property 1. *A block can be serviced if at least one of its nodes is visited.*

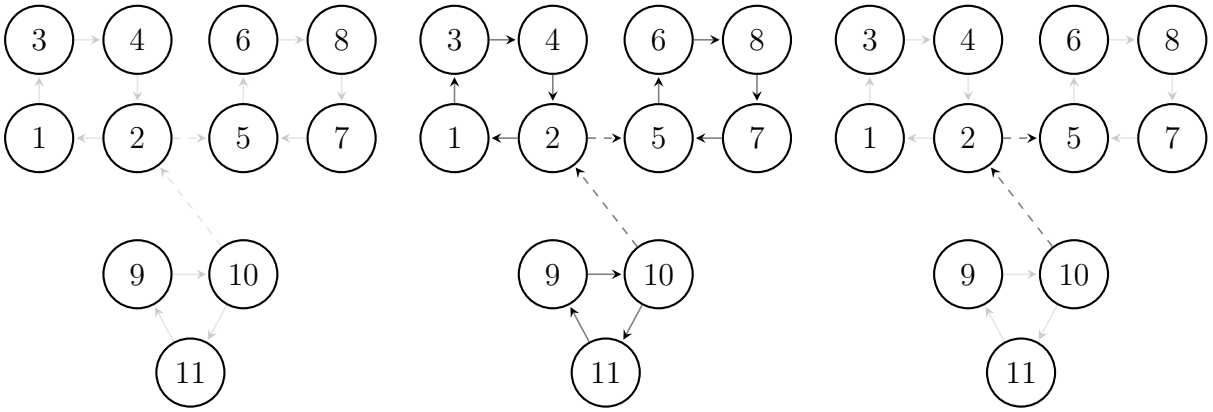
From Property 1, when formulating the problem, it is not necessary to explicitly require the vehicle to completely traverse a block's perimeter in order to count it as serviced. Instead, servicing can be achieved by visiting at least one node within the block and accounting for the corresponding service time. This property is valid due to the definition of the blocks B and the fact that the vehicle must service all blocks in a clockwise direction.

Figure 2.1 illustrates an example of this strategy, where servicing is achieved without requiring a full traversal of the block's perimeter.

Assuming the implicit servicing of a block as allowed by Property 1, additional properties can be established concerning optimal solutions. Let $V_B = \bigcup_{b \in B} V(b)$ be the set of nodes belonging to some city block, and let $S^* = (v_1, \ldots v_n)$ represent an optimal route.

Property 2. *There exists an optimal route S^* with both extremities in V_B.*

Proof. Suppose S^* does not have both extremities in V_B. By removing the prefix and/or suffix of S^* whose nodes do not belong to V_B, we obtain another solution that preserves the total collected prize and does not increase the total time spent, since $t_a \geqslant 0$. □



(a) Instance Example. (b) Explicity Block Servicing. (c) Implicit Block Servicing.

Figure 2.1: Strategies for servicing blocks.

2.2 Deterministic Optimization Models

Given an instance for the CBRP, consider a path-based solution on the original planar graph, in which no arc or vertex is visited more than once. The following binary decision variables are introduced for the model:

- $x_{ij} \in \{0, 1\}$ is a binary variable that indicates whether the arc $(i, j) \in A'$ is included in the route ($x_{ij} = 1$) or not ($x_{ij} = 0$);
- $y_{ib} \in \{0, 1\}$ is a binary variable that indicates whether node $i \in V$ is selected as the starting point for serving block $b \in B(i)$ ($y_{ib} = 1$) or not ($y_{ib} = 0$).

A Path-CBRP formulation is defined as follows:

$$\text{(Path-CBRP)} \quad \max \sum_{i \in V} \sum_{b \in B} p_b y_{ib} \tag{2.1}$$

subject to:

$$\sum_{i \in V} x_{0,i} = \sum_{j \in V} x_{j,0} = 1 \tag{2.2}$$

$$\sum_{i \in V} x_{i,j} - \sum_{k \in V} x_{j,k} = 0 \qquad \forall j \in V \tag{2.3}$$

$$\sum_{i \in V(b)} y_{ib} \leq 1 \qquad \forall b \in B \tag{2.4}$$

$$\sum_{j \in \delta^-(i)} x_{i,j} \geq y_{ib} \qquad \forall b \in B, i \in V(b) \tag{2.5}$$

$$\sum_{(i,j) \in A} x_{i,j} t_{i,j} + \sum_{i \in V} \sum_{b \in B} y_{ib} t_b^{'} \leq T \tag{2.6}$$

$$\sum_{(i,j) \in A(S)} x_{i,j} \leq |V(S)| - 1 \qquad \forall S \subseteq V \tag{2.7}$$

$$x \in \mathbb{B}^{|A'|} \tag{2.8}$$

$$y \in \mathbb{B}^{|V'|\times|B|}. \tag{2.9}$$

The Path-CBRP formulation generates a closed path that starts and ends at the depot. The objective function (2.1) maximizes the total prize collected from each serviced block. Constraints (2.2)-(2.3) enforce the start and end of the route at the depot, and the flow conservation at each node, i.e. the number of arcs entering a node equals to the number of arcs leaving it. Constraints (2.4) and (2.5) ensure that exactly one node is selected as the starting point for servicing a block. Constraints (2.6) impose a time limit, accounting for differences in time spent while servicing and traveling. Constraints (2.7) prevent the formation of subcycles. Finally, constraints (2.8) and (2.9) define the domain of the decision variables.

Since the subcycle elimination constraints are exponentially large in relation to the input size, there are two most common approaches to solve Path-CBRP: the first is to apply the subcycle elimination procedure to integer solutions obtained during the solver branch and bound. The second is to replace the constraint by a more compact set of constraints based on the Miller-Tucker-Zemlin (MTZ) formulation.

To implement the MTZ approach, we introduce the following additional variable:

- $w_a \in \mathbb{R}$ is a real-valued variable representing the accumulated time along the arc $(i, j) \in A'$.

Using this variable, it is possible to replace the constraints (2.7) by the following set of constraints:

$$w_{j,k} \geq w_{i,j} + x_{i,j}t_{i,j} - (2 - x_{j,k} - x_{i,j})T \qquad \forall (i,j) \in A, (j,k) \in A, j \in V \tag{2.10}$$

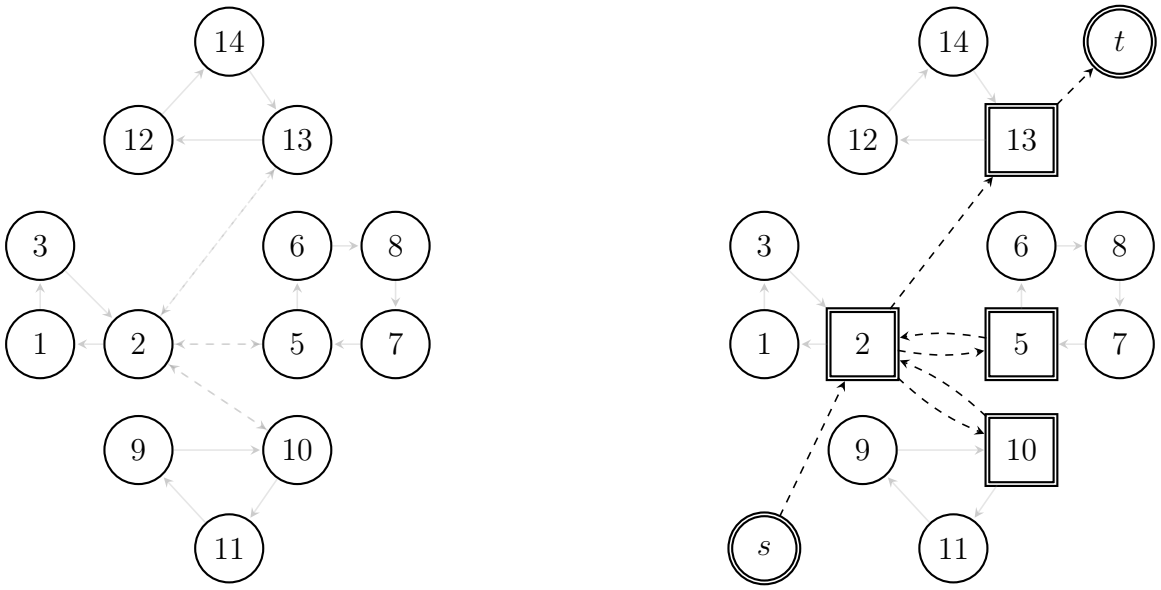
$$w_{i,0} \leq T \qquad \forall i \in V. \tag{2.11}$$

Constraints (2.10) compute the accumulated time at each arc, while constraints (2.11) enforce an upper bound on the MTZ variable. This reformulated model maintains an equivalent set of integer feasible solutions to Path-CBRP while significantly reducing the number of constraints, which now grows polynomially with respect to the input size. However, the fractional feasible space may widen, since the number of cuts reduced from exponential to polynomial. This formulation is referred to as Path-CBRP-MTZ.

2.2.1 Walk-based Formulation

The possibility of visiting a vertex more than once is a significant difference between the Path-CBRP and the Walk-CBRP. This allows for more flexibility in the solution, and achieve values always greater than or equal to the Path-CBRP. Figure 2.2 presents a feasible route for the Walk-CBRP, that is not feasible for the Path-CBRP. In the example, the route starts at the depot, then visits node 2, and goes to node 5, then goes back to 2, does the same to node 10 and finally ends the route going to node 13 and depot.

Given an instance for the CBRP, consider a walk-based solution on the original planar graph, in which an arc or vertex could be visited more than once. The following binary decision variables are introduced for the model:



(a) Instance Example. (b) Walk Solution.

Figure 2.2: Walk Feasible Solution Example.

- $x_{ij} \in \mathbb{Z}$ is an integer variable that indicates the number of times that the arc $(i, j) \in A'$ is traversed in the route;
- $y_b \in \{0, 1\}$ is a binary variable that indicates whether block $b \in B$ is attended ($y_b = 1$) or not ($y_b = 0$).

A Walk-CBRP formulation is defined as follows:

$$\text{(Walk-CBRP)} \quad \max \sum_{b \in B} p_b y_b \tag{2.12}$$

subject to:

$$\sum_{i \in V} x_{0,i} = \sum_{j \in V} x_{j,0} = 1 \tag{2.13}$$

$$\sum_{i \in V} x_{i,j} - \sum_{k \in V} x_{j,k} = 0 \qquad \forall j \in V \tag{2.14}$$

$$\sum_{i \in V(b)} \sum_{j \in \delta^-(i)} x_{i,j} \geq y_b \qquad \forall b \in B \tag{2.15}$$

$$\sum_{(i,j) \in A} x_{i,j} t_{i,j} + \sum_{b \in B} y_b t_b^{'} \leq T \tag{2.16}$$

$$\sum_{(i,j) \in \delta^+(S)} x_{i,j} \geq x_{k,l} + x_{p,q} - 1 \qquad \forall S \subseteq V, (k,l) \subseteq S, (p,q) \nsubseteq S \tag{2.17}$$

$$x \in \mathbb{Z}^{|A'|} \tag{2.18}$$

$$y \in \mathbb{B}^{|B|}. \tag{2.19}$$

The objective (2.12) maximize the profit collected. Constraints (2.13) ensure that

the route starts and ends at the depot. Constraints (2.14) ensure the equity of arcs in each used node. Constraints (2.15) ensure the connectivity to at least one node to any attended block. Constraints (2.16) ensure that the route and the block service time does not exceed the time limit. Constraints (2.17) ensure that the route does not contains disconnected subtours. Variables (2.18) and (2.19) ensure the domain of the variables.

We define, for every pair of arcs $a' = (k, l)$ and $a'' = (p, q)$, the family of separating sets as

$$\text{cut}(a', a'') = \{S \subset V \mid \{k, l\} \subseteq S,\ \{p, q\} \nsubseteq S\}.$$

That is, every set $S \in \text{cut}(a', a'')$ contains both endpoints of a' and does not contain the endpoints of a''. If two arcs a' and a'' are selected simultaneously, then for every $S \in \text{cut}(a', a'')$ there must be an outgoing arc from S in the solution.

2.2.2 Walk-Solution Using Transitive Closure of the Graph

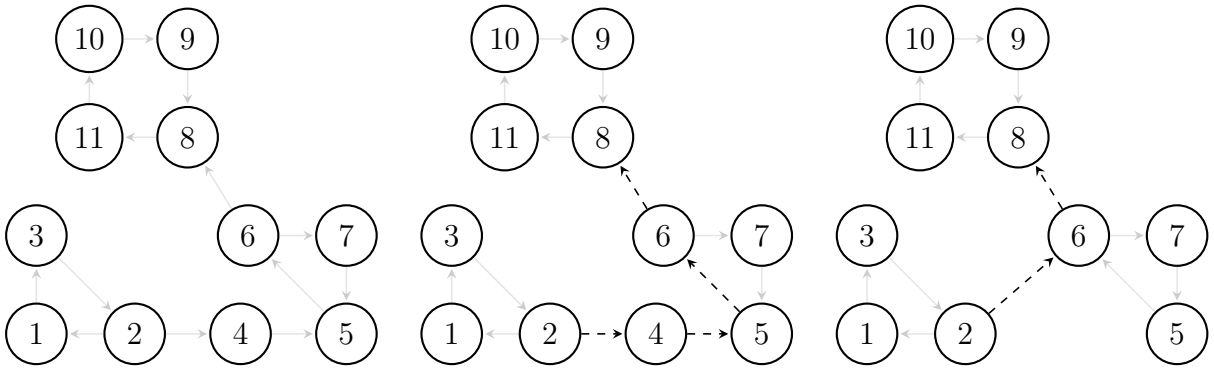
In this section, we present an adaptation that consists in the adaptation of original planar graph that allow us to compute implicit walks using a formulation presented in Section 2.2. Our approach is to construct a new graph in which blocks are transformed into nodes. Let:

- $G' = (V', A')$: be the graph with the node and arcs to the artificial depot $s = 0$.
- $P_{i,j}$: be the shortest path from node i to node j in G where $t_{i,j} = t_{P_{i,j}}$, and $t_{i,0} = t_{0,i} = 0$ for all $i \in V$.
- $V_B = \bigcup_{b \in B} V(b)$: be the set of nodes belonging to some city block.
- $V_B^0 = V_B \cup \{0\}$: represent the set of block nodes along with the artificial depot 0.
- $A_B = A(V_B)$ and $A_B^0 = A(V_B^0)$: be the sets of arcs associated with V_B and V_B^0, respectively.

The construction of G_B^0 (blocks-as-nodes graph) follows from Lemma 2, which considers only nodes in V_B from the original graph, and from the definition of $P_{i,j}$, ensuring that the shortest paths between two nodes in V_B are preserved. Together, these properties guarantee that using G_B^0 preserves all optimal solutions of the CBRP.

Figure 2.3 illustrates the rationale for using graph G_B. Specifically, Figure 2.3a presents a CBRP instance, while Figure 2.3b shows the corresponding solution in the original graph, where the dashed arcs represent the non-spraying travels. Finally, Figure 2.3c represents the same solution mapped onto the transformed graph structure.

This adaptation in the original graph allows us to use any CBRP fomulation presented in Section 2.2 in G_B and generate a implicit walk solution, i.e., a solution that is a path in G_B, however when the arcs represented by shortest paths are converted to real arcs in the original graph G, the solution could turn into a walk. However, this adjustment demands the creation of all arcs between nodes in V_B, fact that could increase significantly the size of G_B according to the number of blocks in the instance.



(a) Instance Example. (b) Solution on Graph G. (c) Solution on Graph G_B.

Figure 2.3: Solution with Transitive Closure.

2.3 Lagrangean Relaxation

Lagrangean Relaxation (LR) is a well-known decomposition method used to solve combinatorial optimization problems. The main idea of LR is to remove complicated constraints from the mathematical model and transfer them to the objective function by assigning them weights (known as Lagrange multipliers), which penalize their violation in any solution. It can be shown that the cost of an optimal solution to the LR always provides a dual bound for the optimal value of the original problem. A primal bound can be obtained by checking the feasibility of a solution returned by the relaxed model and computing the value of the original objective function for that solution. An important step in LR is determining the values of the Lagrange multipliers that yield the best dual bound. For this purpose, the subgradient method can be employed, which is an iterative procedure in which the multipliers are updated until they converge to their optimal values. For a minimization problem, this method can be interpreted as the maximization of the lower bound obtained from the relaxed model based on appropriate choices of multipliers [8].

LR is particularly convenient for problems that, apart from a subset of complicating constraints, can be efficiently solved. For instance, consider the following ILP model:

$$
\begin{aligned}
&\text{(IP)} && z = \min cx \\
&\quad \text{s.t.} && Ax \geq b, \\
& && Dx \geq d, \\
& && x \in \mathbb{Z}^n_+.
\end{aligned}
$$

Assuming $Dx \geq d$ is the set of complicating constraints, removing it yields $z' = \min\{cx : x \in X\}$, where $X = \{x \in \mathbb{Z}^n_+ : Ax \geq b\}$, which is an easier problem and referred to as the relaxed problem. Two facts can be observed. First, z' is a lower (dual) bound on z, since the feasible region of x is larger, so the optimal value of z' is less than or equal to that of z. Second, the optimal solution in X may not satisfy the constraints in $Dx \geq d$. Based on these observations, the idea is to move the complicating constraints

into the objective function, penalizing their violation using a vector $u \in \mathbb{R}^m_+$. This results in the Lagrangian relaxation problem (Lagrangian Primal Problem (LPP)), which can be written as:

$$\begin{aligned} \text{LR}(u) \qquad & z(u) = \min cx + u(d - Dx) \\ \text{s.t.} \qquad & x \in X, \\ & u \in \mathbb{R}^m_+. \end{aligned}$$

The following proposition establishes the relationship between $z(u)$ and z.

Proposition 1. *Let $z(u) = \min\{cx + u(d - Dx) : x \in X\}$. Then, $z(u) \leq z$ for all $u \geq 0$.*

The penalty u_i associated with constraint $D_i x \geq d_i$ is called the Lagrange multiplier for that constraint. We now define the following problem: determine the set of multipliers that provide the best (i.e., greatest) dual bound $z(u)$. To find these values, we must solve the Lagrangian dual problem (Lagrangian Dual Problem (LDP)), described as:

$$\text{LD} \qquad w = \max\{z(u) : u \geq 0\}.$$

The Lagrangian dual can be solved using the subgradient optimization method, which is based on the following result.

Proposition 2. *A function $g : \mathbb{R}^n \to \mathbb{R}$ is concave if and only if, for every $\bar{x} \in \mathbb{R}^n$, there exists $s \in \mathbb{R}^n$ such that $g(\bar{x}) + s(x - \bar{x}) \geq g(x)$ for all $x \in \mathbb{R}^n$.*

Thus, at point $\bar{x}$, it is necessary to choose a direction in which to move to increase $g(x)$. From Proposition 2, we know that if $g(x) > g(\bar{x})$, then $s(x - \bar{x}) > 0$. That is, moving an appropriate amount in the direction of s from $\bar{x}$ will increase the value of g. Therefore, we must find a vector s that satisfies Proposition 2. When g is differentiable at $\bar{x}$, we can take $s = \nabla g(\bar{x})$, i.e., the gradient of g at $\bar{x}$. However, when g is not differentiable, Proposition 2 still guarantees the existence of a vector s such that $s(x^* - \bar{x}) > 0$ for an optimal point x^*. This means it is possible to move from $\bar{x}$ a small step in the direction of s to get closer to an optimal point, even if g does not increase. Before finding such a vector s, we define the notions of subgradient and subdifferential.

Definition 1. *Let $g : \mathbb{R}^n \to \mathbb{R}$ be a concave function. A vector s is a subgradient of g at $\bar{x}$ if and only if $s(x - \bar{x}) \geq g(x) - g(\bar{x})$ for all $x \in \mathbb{R}^n$. The subdifferential $(\delta g(\bar{x}))$ of g at $\bar{x}$ is the set of all subgradients at that point.*

An immediate consequence of the above is the following:

Proposition 3. *If g is concave and $0 \in \delta g(x^*)$, then $g(x^*) = \max\{g(x) : x \in \mathbb{R}^n\}$, that is, x^* is an optimal solution.*

Therefore, in theory, to maximize a concave function g, it is sufficient to start from any point and iteratively take small steps in the direction of a subgradient at that point

until 0 belongs to the subdifferential of the current point—that is, the current point is optimal. The results below allow us to apply this theory to the Lagrangian relaxation technique for ILP.

Proposition 4. *$z(u) = \min\{cx + u(d - Dx)\}$ is concave.*

The next result shows how to compute a subgradient of $z(u) = \min\{cx + u(d - Dx) : x \in X\}$ at point u.

Proposition 5. *Let $\bar{x} \in X$ such that $z(u) = c\bar{x} + u(d - D\bar{x})$. Then, $(d - D\bar{x})$ is a subgradient of $z(u)$ at u.*

Based on the previous definitions and results, we can now present a procedure to minimize a concave function for which a subgradient is known at all points in its domain. For this purpose, a high-level pseudocode of the subgradient method is presented in Algorithm 1.

The algorithm takes three input parameters: 'threshold', 'maxIter', and 'updatePi'. The first two are stopping criteria. The 'threshold' indicates the maximum gap value for the solution to be considered optimal, terminating the execution. The 'maxIter' limits the maximum number of iterations of the subgradient method. The third parameter, 'updatePi', is a counter for updating the value of π, i.e., when 'updatePi' consecutive iterations occur without improvement in the dual bound. Lines (1)–(5) represent the initialization of variables: m is the main loop counter, and ρ counts consecutive iterations without improvement in z_{UB}. The variables z_{LB} and z_{UB} are the lower and upper bounds, respectively.

The symbol α represents the vector of Lagrange multipliers, and θ is the subgradient vector. First, the Lagrangian primal problem is solved using the current multipliers, and this solution is used to obtain the subgradient. If the relaxed model returns a better feasible solution (lower in value), z_{UB} is updated; if the solution is feasible for the original problem, z_{LB} is updated with the value of the original objective function, disregarding the cost of the multipliers. The subgradient vector (θ^m) is then calculated, and its norm (n^m) is used to compute the step size (s^m) in the subgradient direction. The Lagrange multipliers are then updated for the next iteration. Line (26) ensures non-negativity of the multipliers.

In this work, we developed three Lagrangean relaxations based on the Path-CBRP-MTZ model. An important detail to consider is the resolution of the lagrangean relaxations when the constraints (2.5), (2.6) and (2.11) are dualized, being treated in the LPP, i.e., when this problem becomes a shortest path problem with resource constraints and a selection/Knapsack problem on y variables.

2.4 Optimization Under Uncertainty

This Section presents the basic concepts of Stochastic Programming and is strongly based on the textbook by Birge and Louveaux [9].

Two-Stage Stochastic Programming models with recourse deal with uncertain data, represented by random variables. A decision must be made at the present time; we call

Algorithm 1: Subgradient Method (Minimization Problem)

Input: threshold, maxIter, updatePi
Output: z_{LB}, z_{UB}

$m \leftarrow \rho \leftarrow 0$;
$z_{LB} \leftarrow 0$;
$z_{UB} \leftarrow$ cost of a feasible solution;
$\alpha^0 \leftarrow \theta^0 \leftarrow 0$;
$\pi^0 \leftarrow 2$;
while $(\frac{z_{UB}-z_{LB}}{Z_{UB}}) \leq$ *threshold or* $m <$ *maxIter* **do**
  $x \leftarrow$ solution of $LR(\alpha^m)$;
  $z^m \leftarrow$ cost of the objective function of x ;
  if $z^m > z_{LB}$ **then**
    $z_{LB} \leftarrow z^m$;
  end
  $z_f^m \leftarrow$ original objective function value of solution x;
  if x *is feasible and* $z_f^m < z_{UB}$ **then**
    $z_{UB} \leftarrow z_f^m$;
    $\rho \leftarrow 0$;
  else
    $\rho \leftarrow \rho + 1$;
  end
  if $\rho =$ *updatePi* **then**
    $\pi \leftarrow \pi/2$;
    $\rho \leftarrow 0$;
  end
  $\theta^m \leftarrow$ subgradient(x);
  $n^m \leftarrow$ norm(θ^m);
  $s^m \leftarrow \pi^m \frac{(z_{UB}-z^m)}{(n^m)^2}$;
  $\alpha^{m+1} \leftarrow \max(0, \alpha^m + s^m\theta^m)$;
  $m \leftarrow m + 1$;
end

this the first-stage (or first-phase) decision, and at a later time, after its implementation, the uncertainties are revealed. Then, based on the first-stage decision and the revealed uncertain data, we can make a corrective (recourse) decision, called the second-stage (or second-phase) decision [9].

The uncertain data of the problem are represented by a random vector ξ. We are interested in the case where ξ is a discrete random variable, that is, it can take values from a finite set $\Xi = \{\xi_1, \ldots, \xi_k\}$. We denote each possible random event by $\omega \in \Omega$. Alternatively, ξ can be represented as a function of $\omega \in \Omega$, with $\xi(\omega)$ being the value taken by the random variable when event ω occurs.

First-stage decision variables are represented by vector x, of dimension n_1, while second-stage decision variables are represented by the random vector y, of dimension n_2. We denote by $y(\omega)$ the second-stage decision variables associated with the realization of random event ω.

A traditional two-stage stochastic programming problem, also called a Recourse Problem (RP) or *here-and-now* problem, since a decision must be made in the present before the uncertainties are revealed, is formulated as:

$$\begin{aligned} \max \quad & c^T x + \mathbb{E}_\xi \left[\max q(\omega)^T y(\omega) \right] \\ \text{s.t.} \quad & Ax = b \\ & T(\omega)x + Wy(\omega) = h(\omega) \\ & x \geq 0 \\ & y(\omega) \geq 0 \end{aligned} \tag{2.20}$$

The first-stage data matrices and vectors A, b, and c have dimensions $m_1 \times n_1$, m_1, and n_1, respectively. Once uncertainties are revealed through the realization of event $\omega \in \Omega$, $T(\omega)$, $h(\omega)$, and $q(\omega)$, with dimensions $m_2 \times n_1$, m_2, and n_2, respectively, become known. To represent the problem's uncertainties as a single random vector, we define

$$\xi(\omega)^T = \left(q(\omega)^T,\ h(\omega)^T,\ T_{1\cdot}(\omega), \ldots, T_{m_2\cdot}(\omega) \right),$$

for every $\omega \in \Omega$, where $T_{i\cdot}(\omega)$ denotes the i-th row of matrix $T(\omega)$. We are interested in the class of *fixed recourse problems* [9], in which the matrix W, of dimension $m_2 \times n_2$, represents the second-stage actions and does not depend on the realization of the random event, and is thus known before making the first-stage decision.

Note that the objective function in (2.20) has a deterministic term $c^T x$ and the expected value of $q(\omega)^T y(\omega)$ over all realizations ω. What makes solving a two-stage stochastic program difficult is that, for each $\omega \in \Omega$, determining $y(\omega)$ requires solving a Linear Programming (LP) (or ILP) problem. To make this more explicit, an alternative notation for the same problem is known as the *deterministic equivalent program*. For a given realization ω and first-stage decision x, we define:

$$Q(x, \xi(\omega)) = \max_y \left\{ q(\omega)^T y \;\middle|\; Wy = h(\omega) - T(\omega)x,\ y \geq 0 \right\},$$

as the second-stage value function. Then, the expected second-stage value for the first-stage decision x is:

$$Q(x) = \mathbb{E}_\xi \left[Q(x, \xi) \right].$$

With this, the deterministic equivalent program can be formulated as:

$$\begin{aligned} \max \quad & c^T x + Q(x) \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0. \end{aligned} \tag{2.21}$$

2.4.1 Expected Value of Perfect Information (EVPI)

The Expected Value of Perfect Information (EVP) measures the maximum amount a decision-maker would be willing to pay to have perfect foresight of the future. This

allows determining to what extent a more accurate forecast would improve the quality of the obtained solution. Consider:

$$\begin{aligned} \max \quad & z(x, \xi(\omega)) = c^T x + Q(x, \xi(\omega)) \\ \text{s.t.} \quad & Ax = b \\ & x \geq 0, \end{aligned} \tag{2.22}$$

which is the optimization problem for a given scenario $\xi(\omega)$. We assume that for every $\xi(\omega) \in \Xi$, this problem is neither infeasible nor unbounded.

Suppose there is an efficient way to obtain an optimal solution $\bar{x}(\xi(\omega))$ to (2.22) for scenario $\xi(\omega)$, and let $z(\bar{x}(\xi(\omega)), \xi(\omega))$ be its objective value. The so-called *wait-and-see* (WS) solution is given by:

$$WS = \mathbb{E}_\xi \left[z(\bar{x}(\xi), \xi) \right]. \tag{2.23}$$

That is, for each scenario $\xi(\omega)$, we solve (2.22) to optimality and take the expected value of the objective over all $\omega \in \Omega$. The RP, defined by (2.20) or (2.21), can be written as:

$$RP = \max_x \mathbb{E}_\xi \left[z(x, \xi) \right]. \tag{2.24}$$

By definition, the EVP is the difference between the wait-and-see and here-and-now solutions:

$$EVPI = WS - RP.$$

2.4.2 Value of the Stochastic Solution

A practical difficulty of the wait-and-see approach is that we must solve a subproblem for each scenario $\xi(\omega)$. A simpler approach is to replace the random vector ξ with its expected value $\bar{\xi} = \mathbb{E}[\xi]$. This problem is known as the Expected Value Problem (EVP), or *mean value problem* [9]. This corresponds to a simpler problem than the recourse problem, since we only solve for a single future scenario. The objective of the EVP is:

$$EV = \max z(x, \bar{\xi}). \tag{2.25}$$

Let $\bar{x}(\bar{\xi})$ be an optimal solution to (2.25). Given the stochastic nature of the problem, in general $\bar{x}(\bar{\xi})$ may not be close to the optimal solution of the recourse problem (2.20). However, this solution is useful for defining the Value of Stochastic Solution (VSS), which measures the quality of the solution in terms of (2.20). We define the expected value of the EVP solution as:

$$EEV = \mathbb{E}_\xi \left[z(\bar{x}(\bar{\xi}), \xi) \right]. \tag{2.26}$$

This is equivalent to solving the recourse problem while fixing the first-stage decision

obtained from the EVP. Given the EEV, the VSS is:

$$VSS = RP - EEV,$$

which indicates the loss incurred by adopting only the expected scenario $\bar{\xi}$ compared to the full recourse problem (2.20). This corresponds to the cost of ignoring uncertainty. The values WS (2.23), RP (2.24), and EEV (2.26) satisfy:

$$EEV \leq RP \leq WS.$$

Thus, for a maximization problem, EEV and wait-and-see serve, respectively, as lower and upper bounds for the problem of interest, the recourse problem.

2.5 Stochastic Optimization Models

The stochastic version of the CBRP, called in this work as Stochastic City Block Routing Problem (SCBRP), computes, in the first-stage, a route to nebulize the blocks in the scenario 0 (present), in which the objective is to maximize the profit considering a bonus related to the impact on the reduction of future cases for each scenario in the second-stage. The second-stage represents a set of independent routes that nebulize independent blocks in each future scenario.

In addition to the deterministic data, a set of k future scenarios $\Omega = \{\omega_1, \ldots, \omega_k\}$, with ξ^ω being the probability of scenario $\omega \in \Omega$ to occur. The profit of block b in the present is p_b^0, and the profit in each future scenario is p_b^ω. This value represents the number of cases predicted for that city block in the scenario ω. However, the profit of each block is affected by the first-stage, with a parameter α being a reduction factor in the number of cases in a block nebulized in the first-stage. For example, if $\alpha = 1$, then for all block $\{b \in B : y_b^0 = 1\}$ will have a 0 profit in all second-stage scenarios.

Let $\Omega' = \{0, \omega_1, \ldots, \omega_k\}$ be the augmented set of scenarios, with 0 representing the present scenario and $\omega_1, \ldots, \omega_k$ representing the k future scenarios. The following binary decision variables are introduced for the SCBRP model:

- $y_b^\omega \in \{0, 1\}$: binary variable valued as 1 if the block $b \in B$ is nebulized in the scenario $\omega \in \Omega'$ ($y_b^\omega = 1$) or not ($y_b^\omega = 0$).
- $x_a^\omega \in \{0, 1\}$: binary variable valued as 1 if the arc $a \in A'$ is used in the route in the scenario $\omega \in \Omega'$ ($x_a^\omega = 1$) or not ($x_a^\omega = 0$).
- $z_b^\omega \in \mathbb{R}$: real variable that represents the maximum profit of a block $b \in B$ in a scenario $\omega \in \Omega'$ ($z_b^\omega \in \mathbb{R}_+$).

The SCBRP formulation is defined as follows:

$$\text{(Path-SCBRP)} \max \overbrace{\sum_{b \in B} y_b^0 (p_b^0 + \alpha \sum_{\omega \in \Omega} \xi^\omega p_b^\omega)}^{\text{First-Stage}} + \overbrace{\sum_{\omega \in \Omega} \xi^\omega \sum_{b \in B} z_b^\omega}^{\text{Second-Stage}} \tag{2.27}$$

subject to:

$$z_b^\omega \leq y_b^\omega((1-\alpha)p_b^\omega) + (1-y_b^0)(\alpha p_b^\omega) \qquad \forall b \in B, \omega \in \Omega \qquad (2.28)$$

$$z_b^\omega \leq y_b^\omega p_b^\omega \qquad \forall b \in B, \omega \in \Omega \qquad (2.29)$$

$$\sum_{i \in V} x_{0,i}^\omega = \sum_{j \in V} x_{j,0}^\omega = 1 \qquad \forall \omega \in \Omega' \qquad (2.30)$$

$$\sum_{i \in V} x_{i,j}^\omega - \sum_{k \in V} x_{j,k}^\omega = 0 \qquad \forall j \in V, \omega \in \Omega' \qquad (2.31)$$

$$\sum_{j \in \delta^-(i)} x_{i,j}^\omega \geq y_b^\omega \qquad \forall b \in B, i \in V(b), \omega \in \Omega' \qquad (2.32)$$

$$\sum_{(i,j) \in A} x_{i,j}^\omega t_{i,j} + \sum_{b \in B} y_b^\omega t_b^{'} \leq T \qquad \forall \omega \in \Omega' \qquad (2.33)$$

$$\sum_{(i,j) \in A(C)} x_{i,j}^\omega \leq |V(C)| - 1 \qquad \forall C \subseteq V, \omega \in \Omega' \qquad (2.34)$$

$$x \in \mathbb{B}^{|A'|*|\Omega'|} \qquad (2.35)$$

$$y \in \mathbb{B}^{|B|*|\Omega'|} \qquad (2.36)$$

$$z \in \mathbb{R}_+^{|B|*|\Omega|} \qquad (2.37)$$

The objective function (2.27) maximizes the total profit. When a block is attended in the first-stage, the final value is the profit of the block in scenario 0 plus α times the profit for all scenarios multiplied by the probability of the scenario ω to occur. Besides the profit associated in first-stage, the second-stage maximizes the profit for the residual number of cases (after the α reduction for apply the route in scenario 0) in each block and for every future scenario $\omega \in \Omega$.

Constraints (2.28) ensure that the profit of a block b in the scenario ω is $(1-\alpha)p_b^\omega$, if $y_b^0 = 1$ or p_b^ω when $y_b^0 = 0$. Constraints (2.29) guarantee that the value of z_b^ω is 0 when $y_b^\omega = 0$ and p_b^ω otherwise. The Constraints from (2.30) to (2.34) are the same as the constraints from (2.2) to (2.7) of the Path-CBRP formulation, but for independent routes for all scenario $\omega \in \Omega'$. Constraints (2.35)-(2.37) define the domain of the decision variables.

Considering the Path-SCBRP formulation, with the changes to consider the stochastic nature of the problem, it is also possible to adapt the deterministic models from Section 2.2 to generate feasible solutions for the SCBRP, i.e., creating the k routes to nebulize the blocks in the scenario 0 and each future scenario. The further references for the SCBRP will follow the same pattern presented for the deterministic formulations: Path-SCBRP, Path-SCBRP-MTZ, Walk-SCBRP and Walk-SCBRP-MTZ.

2.6 Multi-Agent-Based Simulation

Epidemiological models aim to answer how a certain disease spreads in a region and how it may appear in regions not initially considered susceptible. Since simulation models have limitations regarding the real world, they commonly attempt to represent a phenomenon either at a global (macro) or local (micro) scale.

Macro-simulation is a deterministic approach that focuses on representation at a global level, neglecting individual characteristics such as sex, age, address, and interactions between individuals. Although easy to interpret and implement, when using a macro model, one must be aware of these limitations since certain individual characteristics may be crucial to disease propagation [54]. On the other hand, micro-simulation models contain a particular representation for each individual.

Multi-agent systems commonly represent complex systems with characteristics such as non-linearity and multiple levels of abstraction [28]. This system contains agents that perceive and act on the environment, reacting to situations and making decisions based on cognitive abilities. Typically, predicting the global behavior of MABS is impractical because individual interactions between agents lead to emergent global structures.

MABS encompass various types tailored to simulate complex systems where multiple autonomous entities interact dynamically. One prevalent type is task-based MABS, focusing on agents completing specific tasks or goals within a system. These simulations are crucial in studying decentralized systems like logistics and manufacturing.

Furthermore, spatial MABS integrates geographic information and spatial relationships, essential for modeling urban planning, environmental systems, and epidemiology. Adaptive MABS introduces agents capable of learning and adapting behaviors over time, necessary for studying adaptive systems such as artificial intelligence and robotics. Lastly, hybrid MABS combines features from multiple types, offering versatility to model various real-world phenomena across domains like economics, biology, and engineering. Each kind of MABS contributes uniquely to understanding complex systems and addressing challenges in predictive modeling and policy analysis [11].

Due to the inherent non-deterministic nature of MABS, multiple simulation runs are required to achieve statistically significant results and accurately characterize system behavior. The complex interactions between input parameters and output behaviors in MABS often make it challenging to establish explicit relationships and formally validate models. To overcome this, researchers frequently employ expert validation, as seen in [3, 54], where epidemiological experts were consulted to evaluate the model's realism and reliability.

Regardless of the scale, it is possible to use compartmental theory, where the main idea is to represent the states of individuals [14]. In our Dengue spread simulation, we adopted the following states: susceptible (S), exposed or latent (E), infected (I), and recovered (R) as illustrated in Figure 2.4. Individuals who have never been exposed to the disease are in state S. If an individual carries the virus but does not yet transmit it or show symptoms, they are in state E. Symptoms appear when they reach state I, which also allows the individual to transmit the infection. In the R state, individuals are considered cured and immune to new infections.

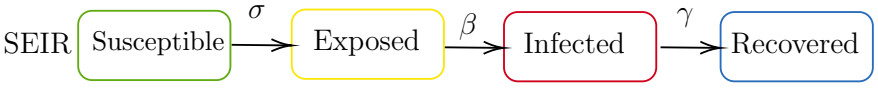


Figure 2.4: Diagram of a dynamic compartmental model.

The compartment flow is based on the order of the letters [2]. For example, an

SEIR model states that individuals are initially susceptible to the virus or disease, then they become exposed, infected, and finally recover. The transition between states is not mandatory and depends on the model's internal parameters, which aim to be as coherent as possible with reality. The choice of a model depends primarily on the intrinsic characteristics of the disease. The literature has proposed a set of different models considering various types of compartments, such as **SEI** [70, 55], **SEIR** [70, 55, 46, 19], and **SIR** [29, 56, 64], among others. Classical mathematical models are mainly based on compartmental theory, and a system of Ordinary Differential Equations (ODE) describes the changes between states [19].

2.7 Simheuristics

Simulation methods are widely used to analyze complex real-life systems under uncertainty, which is typically modeled through probability distributions [42]. These techniques are applied in diverse domains such as manufacturing, logistics, transportation, healthcare, finance, smart cities, and telecommunications. While simulation provides valuable insights into system performance under different scenarios, it does not inherently offer optimization capabilities, especially for large-scale and NP-hard Stochastic Combinatorial Optimization Problem (SCOP) [37]. To address this, simulation is often combined with metaheuristics, which can efficiently explore large solution spaces and deliver high-quality solutions in reasonable computational times.

The hybridization of simulation and optimization, known as simulation-optimization, encompasses various methods including mathematical programming, metaheuristics, and machine learning. In such approaches, simulation can serve multiple roles: evaluating objective functions or constraints in stochastic settings, generating feasible solutions, or enhancing analytical models [36]. Surrogate models derived from simulation outputs can further accelerate the optimization process. This integration enables a balance between modeling realism and computational tractability, making simulation-optimization a first-choice strategy for tackling complex stochastic problems [37].

Given a deterministic optimization problem, it is reasonable to assume that its optimal and near-optimal solutions will exhibit satisfactory performance when moderate levels of uncertainty are introduced [37]. Formally, for a solution s, let $\det(s)$ denote its deterministic performance and $\text{stoch}(s)$ its stochastic performance (estimated via simulation). Under moderate uncertainty, these values are expected to be strongly and positively correlated. However, this correlation diminishes as uncertainty increases, and for large variances, robust or reliable solutions may be preferable to those merely maximizing expected performance.

Another important observation concerns the use of the best deterministic solution $\det(s^*)$. In many optimization problems, the optimal objective value degrades as uncertainty increases, particularly when asymmetric effects are present, where detrimental realizations of random variables outweigh beneficial ones. Let s^* denote an optimal or near-optimal deterministic solution, and s^{**} an optimal or near-optimal stochastic solution. Then, for minimization problems, it often holds that $\det(s^*) \leq \text{stoch}(s^{**}) \leq \text{stoch}(s^*)$,

while for maximization problems, $\text{stoch}(s^*) \leq \text{stoch}(s^{**}) \leq \det(s^*)$. Thus, deterministic optimal values provide natural lower and upper bounds for their stochastic counterparts [37].

Figure 2.5 shows the default logic behind simheuristic concept. Given a stochastic optimization problem, its deterministic counterpart is considered. This can be done, for instance, by replacing all random variables by their expected values, which leads to a simplified version of the stochastic problem in which uncertainty is not considered. Then, a metaheuristic component is employed to efficiently search inside the solution space, thus generating increasingly better solutions to the deterministic version of the problem. Each time the metaheuristic generates a "promising" solution (i.e., one that is likely to perform well in a scenario under uncertainty), this solution is sent to the simulation component in order to assess its performance in the stochastic environment.

The simulation component returns not only estimates of statistics, such as the mean, variance, and percentiles associated with the proposed solution, but also probabilistic information that can be useful in risk or reliability analyses, as well as information about the status of different system variables. Since simulation is a time consuming procedure, it is usually a good idea to employ a short number of simulation runs every time a new promising solution is evaluated at this stage, otherwise the simulation computing time might jeopardize the time required by the metaheuristic to converge to near-optimal solutions. Some useful recommendations to speed up these computations can be found in Rabe [57].

The feedback provided by the simulation can then be processed by a machine learning component and then used to: (i) update and adjust the metaheuristic parameters to better explore the solution space and increase the chances of obtaining new solutions with a high stochastic performance; (ii) build a classification or prediction model able to identify new promising solutions with a high accuracy, thus avoiding wasting time in simulating solutions that will offer sub-optimal values under uncertainty conditions; and (iii) develop a surrogate model that, at least partially, substitutes time-consuming simulations when estimating the value of the stochastic objective function or probabilistic constraints associated with a new promising solution proposed by the metaheuristic component

As a result of this first stage, a reduced list of "elite" solutions is obtained. According to our initial estimates, each of these solutions show a high performance in a scenario under uncertainty. Now, in order to increase the accuracy of our estimates, a more intensive simulation is executed on each of the elite solutions. Of course, variance reduction techniques, such as the use of common random numbers, can be employed here to speed up these simulations, which can also be run in parallel processors. As a final stage, the simulation outcomes might be employed to perform a risk or reliability analysis on each of these elite solutions. This analysis might enrich the decision-making process with detailed probabilistic information describing the stochastic behavior of each elite solution.

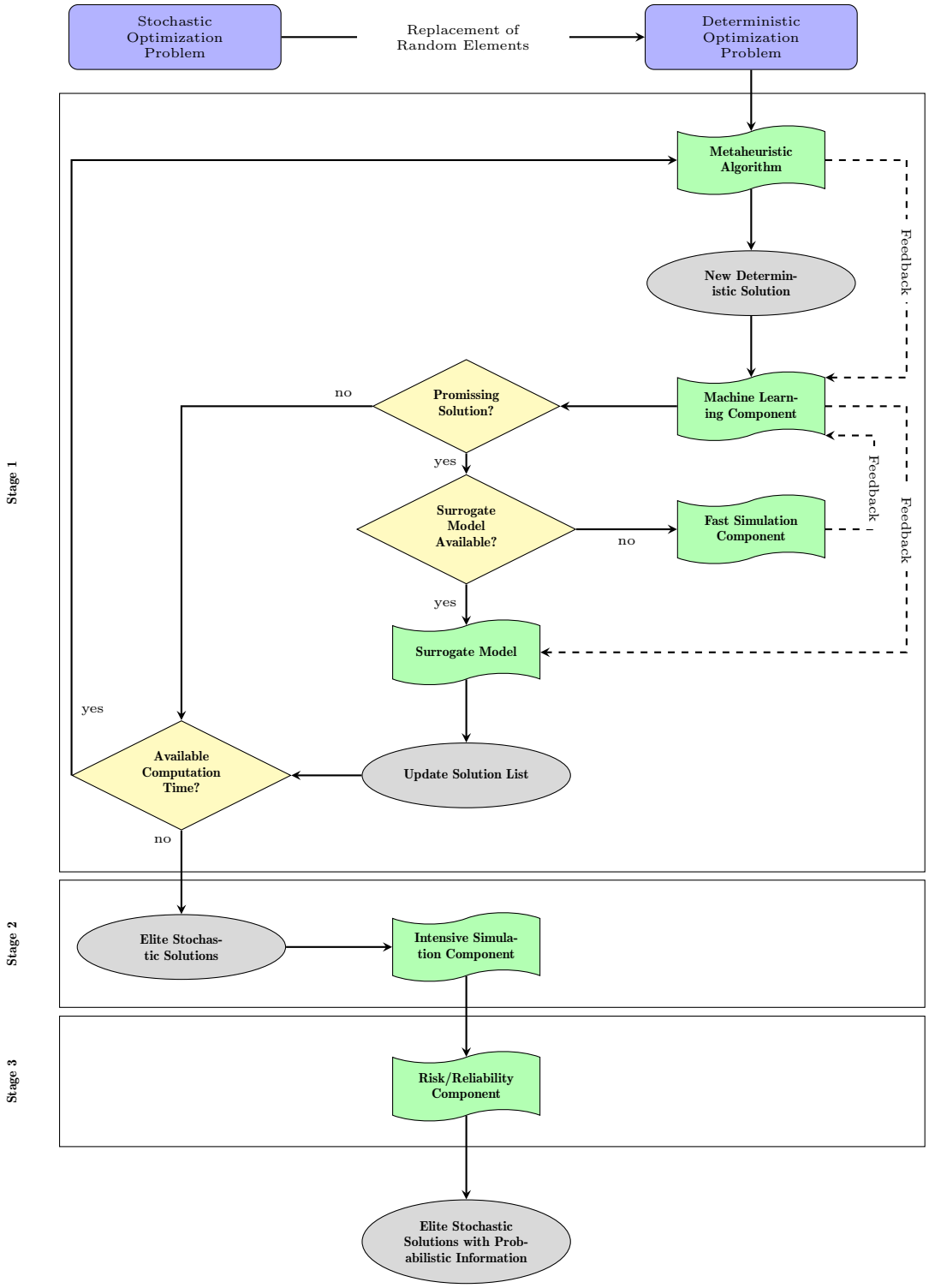


Figure 2.5: Logic Behind Simheuristics [37].

Chapter 3

Literature Review

This section presents the existing literature related to the main topics considered in this paper. The focus is on methodologies that simulated the spread and impact of Dengue and other arboviral diseases, along with computational methodologies for support decisions in vector control.

3.1 Operations Research

An extensive number of works in the literature use Machine Learning (ML) in the context of Dengue [62, 61, 32, 59, 5]. Some of the main problems addressed are generating Dengue diagnoses based on the patient's symptoms and clinical patterns, distinguishing Dengue and its types in the early stage of disease progression and, Dengue fever predictions in certain regions based on a set of factors, such as precipitation, rain, humidity, temperature, and others.

Lowe et al. [41] presented a modeling procedure to quantify the benefits of including climate information in a Dengue model for the 76 provinces of Thailand, from 1982-2013. The authors considered a set of factors that have a statistically significant contribution to the relative risk of Dengue. The system provided an advanced warning that enabled the implementation of a more effective surveillance system. According to the authors, the proposed framework was flexible to be applied in any geographical setting, generating a more global approach to assessing the impact of climate variability and climate change on Dengue risk.

Handam and Kilicman [33] developed a mathematical model of Dengue transmission considering the effect of the temperature on the transmission dynamics. The work computes a primary reproduction number (R_0) considering the known aspects of the *Aedes* mosquito population and the experimental data of the Dengue transmission in Malaysia. Ultimately, the authors provide numerical evidence that the Dengue virus can spread in temperate areas that are no longer limited to tropical and subtropical regions.

Maneerat and Daudé [44] presented the Model of Mosquito Aedes (MOMA), a spatial agent-based simulation model of the Aedes female mosquito. This model aims to produce statistical data on mosquito behaviors and population dynamics that are difficult to obtain through field surveys, such as population densities in various geographical and

climatic conditions. The paper gives a detailed description of the multiple components of the model and the approach to calibrate and validate it. The data in the model represent a developing neighborhood in Delhi, India, processed using a Geographical Information System (GIS). The results indicate a significant relationship between urban topology, human densities, and adult mosquito flight.

Some previous works have reported multi-agent system approaches in endemics. For example, in [3], the author created a module able to trace a relationship between the spread of the Dengue virus and commercial routes in Asia. Likewise, Damien et al. [54] presented a model that encapsulated the characteristics of the north of Vietnam to describe the spread of the H1N5 virus. Hence, it seems reasonable to use the same approach for the study in Brazil, as in [19], that developed a MABS and a classical model based on ordinary differential equations to simulate the Dengue virus propagation adopting some simplifying assumptions.

Cantane et al. [15] presented a mathematical model to estimate the rate of population growth of mosquitoes. The authors observed that, without a control action, the population of mosquitoes could rapidly increase in short periods. The results of the experiments provide estimates of the life cycle of the mosquitoes are in perfect condition in nature and the model allows to measure an approximate number of mosquitoes for each scenario.

The authors of [23], [26], and [34] presented methodologies for Dengue control, including mathematical models that describe the population dynamics of the Aedes aegypti. A Genetic Algorithm (GA) was developed to obtain optimal strategies using an integration of different types of control. A Variable Neighborhood Search (VNS) determines optimized control strategies for the aquatic phase, which is when the mosquitoes are larva and pupa. The authors argue that the control during the aquatic phase reduces the environmental impact and the use of chemical control on large scale. The authors indicated that the presented methodologies have the potential as a tool for supporting decision-makers against arboviruses.

The works of Negreiros et al. [49, 48, 47], to the best of our knowledge, are the first papers that apply optimization techniques for solving routing problems in the context of mosquito control. These works focus on the computational tool called Web Dengue, aimed at helping the Dengue control managers by providing a better visualization and coordination of control procedures. Among the methodologies, the authors developed an ILP formulation to compute a periodical scheduling of vehicles. A GRASP heuristic was also developed to solve a real-life instance corresponding to the location of Praia de Iracema, Fortaleza.

To the best of our knowledge, Andrade et al. [4] are one of the first to present routes for spraying vehicles in Dengue control. The work developed an Integer Programming (IP) formulation based on the rural postman problem, that is a routing problem that consider mandatory visit to a set of arcs. The instance set is based on Dengue outbreak locations reported for the city of Campinas, São Paulo, and the set of required arcs consider a complete region. The results show that the methodology is effective to obtain optimal routes for instances with more than 2500 vertices and 5500 arcs in less than one hour of computational time.

The work of [6] (accepted for publication), proposed the Dengue Prize-collection Arc

Routing Problem (Dengue-PARP), where the focus is to generate routes for spraying vehicles. In Dengue-PARP, each arc of a street block has a prize value related to the number of dengue cases. The Dengue-PARP objective is to maximize the prize collected to create a route that covers a subset of blocks respecting a limited amount of resources. The work presented an IP formulation and a set of 39 real-world instances based on the cities of Alto Santo and Limoeiro do Norte, Ceará. The methodology achieved fast optimal results for 31 instances containing up to 1212 nodes and feasible solutions for the other 8.

3.2 Dengue Spread Simulation

In [12], the authors propose a MABS to assess the pupal productivity of the *Aedes Aegypti* mosquito. The model considers the productivity of water-holding containers, defined as the number of adult mosquitoes each site can produce. The findings indicate that factoring in pupal productivity is crucial for effective dengue control and prevention strategies. Consequently, they assert that prevention measures should incorporate pupal productivity and recognize that the proximity of containers can enhance productivity, thereby heightening the risk of transmission.

[44] presented the Model of Mosquito Aedes (MOMA), a MABS that explores the population dynamics of the *Aedes aegypti* female mosquito. This model produced statistical data on mosquito behaviors and population dynamics that are difficult to obtain through field surveys, such as population densities in various geographical and climatic conditions. The paper gives a detailed description of the model and its calibration process. The data employed by the model is from a developing neighborhood in Delhi, India, processed using a Geographic Information System (GIS). Results showed a strong relationship between urban topology and adult mosquito flight range: in areas with limited open space, mosquitoes tend to remain close to their site of birth. Thus, densely built-up areas and wide roads act as barriers, restricting mosquito movement and access to resources.

[40] compare two intricate, spatially explicit, stochastic models that analyze the population dynamics of *Aedes aegypti*. While both models account for the mosquito's biological and ecological attributes, their complexity and underlying assumptions differ. The authors evaluate the models' predictions within two distinct climatic environments: the tropical and mildly seasonal climate of Iquitos, Peru, and the temperate, highly seasonal environment of Buenos Aires, Argentina. The models were employed to simulate killing larvae and/or adult mosquitoes. The results reveal considerable differences in the models' predictions regarding population recovery, likely attributable to differing assumptions about larval development and density dependence – that is, the negative impact of higher larval density on development time and survival due to resource competition.

[54] focuses on the East-West economic corridor in Southeast Asia, where a notable correlation between corridor opening and dengue fever cases has been observed, though causality remains undetermined. To address this, authors employ an MABS model integrating dengue dynamics, climate data, economic mobility, and health policies. The approach decomposes these factors into sub-models linked to form an integrated model,

and it also proposes strategies to manage data gaps in modeling. Simulation results show the influence of increased mobility and varied control policies on rising dengue cases.

In [16], various control strategies and recommendations for the vaccination campaign are assessed. The proposed mathematical model integrates mechanical and chemical control. Mechanical control is determined by the environmental support capacity, influenced by a discrete function that represents the elimination of breeding sites. Chemical control involves the application of insecticides and larvicides. The authors conclude that eradicating the dengue epidemic is only possible with the introduction of an immunizing vaccine, as in some scenarios, vector control measures alone are not sufficient to eliminate the disease, which may persist or reemerge even after infected mosquitoes are removed.

[19] presented a work to simulate the dengue virus propagation using a MABS approach. Inspired by compartmental models in epidemiology, the agent-based model was implemented on the GAMA platform alongside a classical model based on ordinary differential equations. Despite some simplifying assumptions, comparing the outputs of the two models validated the approach, indicating that the model could serve as a foundation for developing more refined models in the future.

In [43], the authors expand the conventional SEIR mathematical model by introducing a MABS to analyze the interactions between hosts and vectors. They estimate the growth of the vector density based on reproductive behavior and simulate agent interactions for virus transmission within a spatiotemporal context, predicting disease spread in a specified area over time. Their simulation results reveal expected dengue cases and their movement patterns, which can assist in the early detection of epidemic outbreaks. The authors demonstrate a similar trend between the real and simulated data. This similarity is quantified by a Root Mean Square Error (RMSE) of 0.064.

A model comprising eight distinct compartments was presented in [1]. This model captures the dynamics of dengue fever transmission while incorporating interventions such as personal protective measures, larvicide, and adulticide applications. The calibration of this model is based on data from the dengue outbreak that occurred in Johor, Malaysia, in 2012, utilizing the least-squares method. Optimal control theory is applied to effectively explore the synergistic effects of combined control strategies to mitigate the spread of dengue fever.

An ODE model taking into account multiple strains of dengue virus is presented in [69]. This model allows the evaluation of human vaccination effectiveness considering the decline and failure of immunity. A sensitivity analysis was conducted, and the influence of different parameters on the basic reproduction number was measured. The findings suggest that early-stage vaccination of humans is more beneficial.

[33] developed a deterministic mathematical model of dengue transmission, incorporating the effect of temperature. The model exhibits a disease-free equilibrium (DFE),a state in which dengue eventually disappears from the population, when the basic reproduction number (R0) is below 1. However, it also displays a backward bifurcation, meaning that even when R0 is less than 1, the disease may persist under certain conditions, such as high initial numbers of infected individuals. Using entomological and experimental data from Malaysia, the model evaluates R0 at different temperatures, peaking at 32°C. The model's solutions show oscillatory behavior in the number of cases, which are smoothed

when using an alternative mathematical formulation that incorporates memory effects and gradual change, known as a fractional-order model. This suggests that dengue can spread in temperate areas and that the fractional-order approach is a promising alternative for modeling dengue transmission dynamics.

In [31], the authors propose a MABS to simulate mosquito population dynamics by incorporating key biological and behavioral characteristics of Aedes aegypti. The model also introduces parameters to represent human responses to awareness campaigns focused on eliminating water containers. These parameters include: the effectiveness of bucket-emptying actions, the frequency of these actions, and a delay parameter that models the time until emptied containers become refillable and suitable again for oviposition. Through numerical simulations, the study analyzes how different combinations of these protocols impact both adult and aquatic mosquito populations. The results indicate the existence of a critical effectiveness threshold above which mosquito populations can be significantly reduced.

[53] present a MABS for dengue fever developed using the Multi-Agent Research and Simulation (MARS) framework, a platform for modeling and simulating agent-based models in C#. The model is designed considering climatic and environmental factors and is contextualized in the region of Dar es Salaam, Tanzania. However, the study is limited to the conceptual and design stages, without performing simulation experiments or analyzing the actual impact of these factors on disease dynamics.

[17] introduce a hybrid model that combines statistics and MABS to analyze mosquito population dynamics. A central step in their methodology involves calibrating a key model parameter using spatiotemporal abundance estimates generated by a Generalized Additive Model (GAM), a statistical model that captures non-linear relationships between mosquito abundance and environmental factors such as temperature and precipitation. Utilizing this calibrated parameter alongside literature-derived values in the MABS, the authors explore the dynamics of the mosquito population and the effects of insecticide spraying on adult mosquitoes. The model is validated using a large dataset of 176.352 field observations of adult *Aedes aegypti*, collected from mosquito traps between 1999 and 2011 in Iquitos, Peru. Their results suggest that the success of insecticide spraying campaigns strongly depends on achieving high coverage and repeated applications, as targeting only a fraction of households proves insufficient.

[66] propose a comparative study of three implementations of a MABS designed to simulate the dynamics of *Aedes aegypti* mosquitoes and the transmission of vector-borne diseases. The models were implemented using different platforms – Mesa-Geo, Repast Simphony and Repast HPC – to evaluate their performance in terms of computational efficiency, scalability, memory usage, and suitability for incorporating geospatial and epidemiological factors in an urban context. The authors conclude that while Mesa-Geo facilitates model development and visualization, Repast HPC offers superior scalability and performance, making it more appropriate for large-scale simulations aimed at informing public health decision-making.

This research introduces a Multi-Agent-Based Simulation (MABS) approach with novel contributions compared to previous studies. Our model integrates geographical data to define population density, position agents within street blocks, and dynamically

determine the initial population size. Using historical case notifications, we generate the simulation's starting scenario by estimating the initial number of infected individuals and calibrating parameters based on the specific city. The objective is to project short- to mid-term future scenarios (up to 14 Epidemiological Weeks (EW)), assisting cities of different scales in making timely decisions to prevent and mitigate epidemic seasons while optimizing resource allocation. Thus, this methodology aims to provide more localized and detailed insights into Dengue propagation, considering the specific dynamics and characteristics of smaller urban areas in Brazil.

Table 3.1 highlights the key differences between the papers cited in this section, including our work. If a paper includes the specified property or data listed in a column, the corresponding cell in the table is marked with a ✓.

The analysis of the table shows that our approach stands out in two key aspects: The location of agents in street blocks that contributes to the simulation environment closer to real urban settings. In this work, human agents have fixed home and work locations within specific blocks, enabling more realistic mobility patterns. This mapping also supports flexible spatial granularity, as the simulation can be adapted to different levels of detail by adjusting the number and size of blocks or by focusing on specific city regions or neighborhoods. The second element is the use of short-range historical notifications to define the initial simulation scenario. This decision replicates the current procedure adopted by the public health authorities to evaluate current epidemiological conditions, such as the number of dengue cases in recent weeks and take preventive actions accordingly.

Table 3.1: Summary of the literature on Dengue simulation.

Paper	**Agent-based Simulation**	**Use Mortality and Birth Rates**	**Map agents into Street blocks**	**Use** GIS **Data**	**Start Simulation From Real Notifications**	**Validate Results with Historical Data**
[12]	✓	✓	—	—	—	—
[44]	✓	✓	—	✓	—	—
[40]	—	—	—	✓	—	✓
[54]	✓	✓	—	✓	—	—
[16]	—	✓	—	—	—	—
[19]	✓	—	—	—	—	—
[43]	✓	—	—	—	—	✓
[1]	—	✓	—	✓	—	—
[69]	—	✓	—	—	—	—
[33]	—	—	—	✓	—	✓
[31]	✓	✓	✓	—	—	—
[53]	✓	✓	—	✓	—	—
[17]	✓	✓	—	✓	—	—
[66]	✓	✓	—	✓	—	—
This paper	✓	✓	✓	✓	✓	✓

Chapter 4

Methodologies

4.1 Dengue Virus Spread Simulation

This section presents the design and implementation of the proposed MABS following the compartmental theory [2]. The model adopts a SIR compartment for human agents and SEI for mosquitoes. Birth and death rates are considered for the mosquito population, while the human population remains constant throughout all simulations. Transitions from susceptible to infected (or exposed) occur based on interactions between a susceptible agent and an infected agent of the opposite species.

Table 4.1 presents the description of the model parameters as well as their chosen values. Most parameter values are based on those explored in the works of [19] and [24]. The other parameters, especially those that established the generation of new mosquitoes, are adjusted with experiments discussed in Section **??**.

Table 4.1: Static parameters used in the proposed MABS.

Parameter	**Description**	**Value**
γ_H	Human recovery rate per day	0.146
a	Daily rate of bites	0.168
b	Fraction of infectious bites	0.600
γ_M	Mosquito latency rate per day	0.143
c	Mosquito susceptibility to dengue	0.526
ρ	The percentage of eggs that produce female mosquitoes	0.125
ϕ	Mosquito daily oviposition rate	0.020
ω	Carrying capacity of the mosquito	3.000
δ	Mosquito daily mortality rate in aquatic phase	0.066
σ	Mosquito daily maturation rate	0.100
μ_M	Mosquito daily mortality rate in adult phase	0.010
Δ_M	Daily rate of eggs to female adult mosquitoes	0.125

Each value in Table 4.1 represents the probability applied to agents in the simulation in each cycle. The parameter γ_H denotes the likelihood of a human transitioning from Infected to Recovered. The parameter a indicates the chance of a human being bitten by a nearby mosquito, while b represents the probability that a bite from an infected

mosquito is infectious. The transition of a mosquito from exposed to infected depends on γ_M. If a mosquito bites an infected human, the probability of the mosquito becoming infected is given by c. The parameter ρ signifies the number of eggs that hatch into female mosquitoes, ϕ is the likelihood of a mosquito laying eggs due to an interaction with a breeding site, and ω represents the maximum number of eggs deposited per interaction. The parameter δ indicates the probability of an egg dying, while σ is the probability of it developing into an adult susceptible mosquito. Finally, μ_M is the adult mosquito mortality rate.

4.1.1 MABS Agents

The model has four types of agents that could have some interactions with each other: mosquito, human, egg, and breeding site. The sizes of the human and breeding site populations do not change during the simulation. The mosquito agent is divided into two phases: aquatic and adult. The breeding site agent represents a point on the map that allows mosquitoes to lay eggs to proliferate new individuals for the population, independent of the current state. The mosquitoes in the starting population and those generated during the simulation are associated with a breeding site.

This work does not consider vertical transmission, i.e., mosquitoes do not transmit the virus to eggs in the breeding sites. This information implies that all mosquitoes changing from the aquatic to adult phase in a breeding site start as susceptible. A simplified flowchart of interactions is presented in Figure 4.1, followed by a more detailed description.

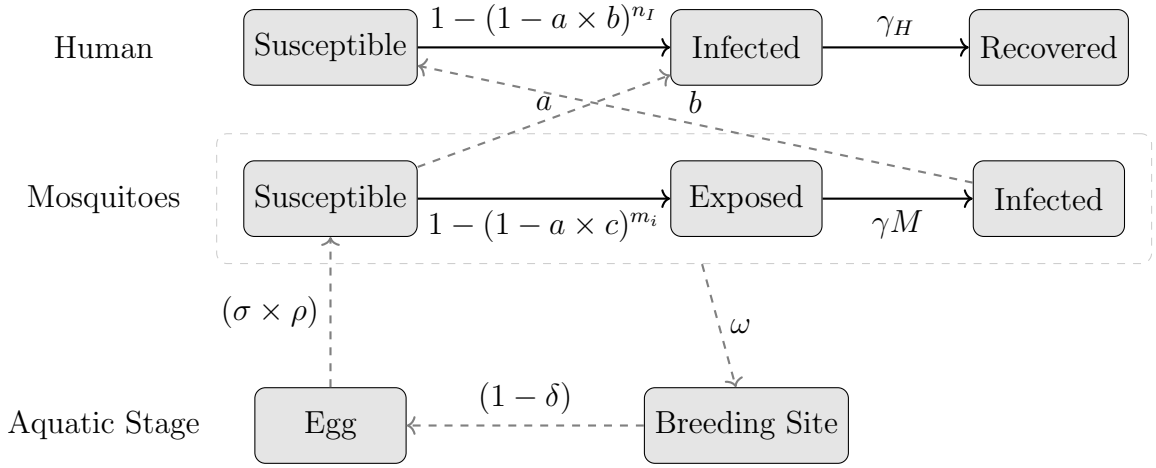


Figure 4.1: Simplified agent interaction flowchart.

- **Human Agent:**
  - **Location**: For each human, two blocks are randomly defined to represent work and resting places. The exact point inside of each block is also defined randomly.
  - **Move**: Considering the basic people's routines, they cycle every day between two locations. It is assumed that humans have a residence and a destination

point, which may be, for example, school or work. The movement is based on daily hours; each human has a "*start_work_time*" that is a time between 5 AM and 8 AM, indicating the time when the agent starts to move from home to its destination, setting the current objective to "working". The "*end_work_time*" follows the same pattern, representing the return home between 4 PM and 7 PM, changing the current objective to "resting".

  - **Change to Infected state**: Changing a human agent from susceptible to infected requires interaction with infected mosquitoes. The higher the number of infected mosquitoes nearby, the greater the chance of this state transition. Assume n_I as the number of infected mosquitoes within a 1-meter distance and the probability of a human being infected by the bite of an infected mosquito. From these values, consider $(a \times b)$ as the probability of a human becoming infected. Thus, the chance of changing from a susceptible to an infected state is given by $1 - (1 - a \times b)^{n_I}$.
  - **Change to Recovered state**: Changing from infected to recovered depends on the parameter γ_H.

- **Breeding Site Agent.**
  - **Location**: Each breeding site is located at a random point inside a street block.
  - **Create new eggs**: When a mosquito lays an egg at a breeding site, this interaction creates a new Egg agent in the simulation. This new egg is associated with the breeding site, and this decision impacts the movement range of the adult mosquito.

- **Egg Agent.**
  - **Location**: All Egg agents are located inside breeding sites.
  - **Change to Adult phase**: The change from aquatic to adult phase depends on the maturation rate (σ), and the successfully hatched egg rate (ρ). The probability of an egg changing to the adult stage is given by $\sigma \times \rho$.
  - **Aquatic Phase Death**: The death of an egg in the aquatic phase depends on the mortality rate (δ).

- **Mosquito Agent.**
  - **Location**: Each mosquito is associated with a Breeding Site, and its starting location is a random point inside a circle with a 100-meter radius, centered on the breeding site.
  - **Move**: Each mosquito has a 20% chance of remaining stationary during the cycle. If the agent decides to move, this action is carried out randomly and at a random distance from the current location. However, the destination is always close to the starting point, limited by a radius of 100 meters from the associated breeding site.

- **Oviposition**: The frequency and quantity of eggs that mosquitoes lay depend on the number of breeding sites nearby (specifically, within one meter of the current location), the oviposition rate (ϕ), and the biotic capacity of the mosquitoes (ω). Each mosquito selects a random breeding site, and based on ϕ, the mosquitoes can lay a random number of eggs ranging from 1 to ω.
- **Change to Exposed state**: changing from susceptible to exposed depends on the number of infected humans nearby. The larger this number, the greater the chance of changing the state. Considering a as the average bite rate per day, c as the probability of the mosquito being infected by the virus, and m_I as the number of infected people in the vicinity of the mosquito, the probability of a mosquito changing from susceptible to infected is given by $1 - (1 - a \times c)^{m_I}$.
- **Change to Infected state**: The transition from the exposed to the infected state in mosquitoes is governed by the parameter γ_M.
- **Die**: The death of the mosquitoes can occur during any state, and the chance is given by the parameter μ_M.

The flow of time in the simulation is based on cycles. Each cycle represents a moment in time during which all agents perform their specific actions. In this work, each cycle represents a time skip of 12 hours. Since the simulation runs over real dates, the starting hour of the initial date of the simulation is 05:30 AM. This decision ensures that all cycles occur at 05:30 AM/PM, which are moments of thermal inversions known as the most active time for mosquitoes. In addition, these hours divide the human population into work and home locations.

The model implementation was developed in the GAMA Platform[1] [65], a tool designed specifically for MABS. GAMA operates using a modeling language called GAML (Gama Modeling Language), which is an agent-oriented language. This means that everything active in the model can be represented in GAMA as an agent. GAMA provides an interface to manipulate input/output data from the simulation, which enables integration with other tools that interact with the agents of the simulation. Figure 4.2 illustrates a geospatial agent-based simulation of dengue transmission dynamics in *Alto Santo*, Brazil, modeled in the GAMA platform using OSM data. Human populations are represented as yellow dots, red dots denote *Aedes aegypti* mosquito population and black nodes are breeding sites. Potential breeding sites are excluded from this visualization layer, as their simulated locations are confined to street blocks. This simplification ensures visual clarity while retaining the model's focus on human-mosquito spatial interactions.

[1] `https://gama-platform.org/`

Figure 4.2: Geographic view in GAMA platform.

4.2 Lagrangean Relaxation

4.3 Deterministic Heuristics

4.4 Stochastic Heuristics

4.5 Data Analysis

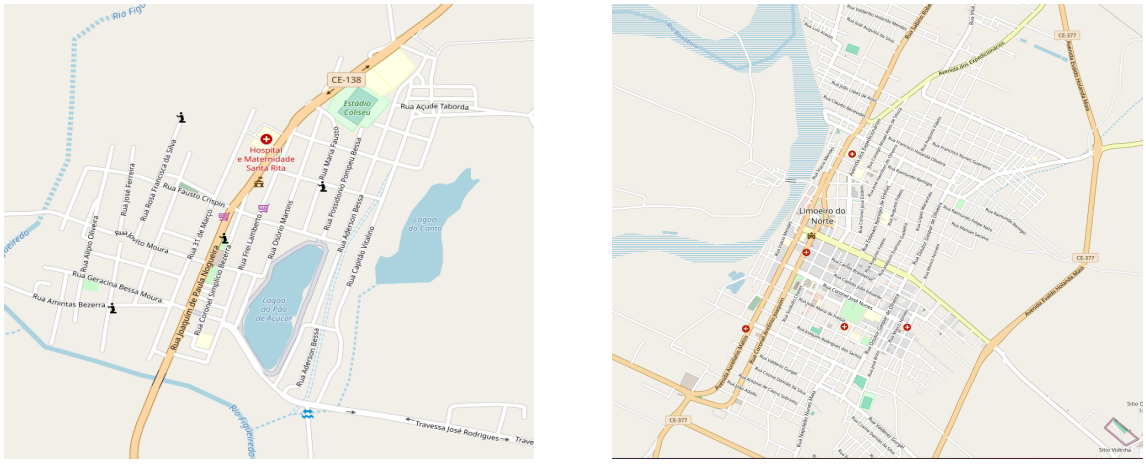
4.5.1 Limoeiro do Norte and Alto Santo Cities

This section presents a analysis of the notified cases from the cities of *Alto Santo* and *Limoeiro do Norte* in the state of Ceará, Brazil. These notifications are from 2015 to 2022, and they were obtained through the city health departments through the Information System for Notifiable Disease (SINAN) [39]. The data is organized in tables that contain 157 information fields from the medical service sheets. From *Alto Santo* city, we obtained 630 notifications in the data from 2016 to 2022. *Limoeiro do Norte* city has reported 3829 Dengue and 414 Chikungunya cases from 2015 to 2021. The data from both cities gives a total of 4873 cases reported in the SINAN.

According to the Brazilian Institute of Geography and Statistics (IBGE)[2], in the last demographic census carried out in 2022, the *Alto Santo* resident population was 14,155 people. In *Limoeiro*, this value increases to 59,560 residents, more than four times the

[2]`https://www.ibge.gov.br/`

population of *Alto Santo*. The urban area of *Alto Santo* is only around 2 km^2, while *Limoeiro* has almost 15 km^2. The city centers extracted from the OSM are presented in Figure 4.3.



(a) Map of Alto Santo downtown.

(b) Map of Limoeiro downtown.

Figure 4.3: Maps extracted from OSM.

Figure 4.4 shows the number of notifications clustered by the final classification for the complete SINAN dataset, including options such as positive for Dengue or Chikungunya, Dengue with alarm signals, severe Dengue, discarded cases, and cases pending closure. According to the current health department of *Alto Santo*, there was underreporting of cases in the SINAN during the initial years. This can be observed, for example, in Figure 4.5, which presents the number of notifications grouped by year for each city. In *Alto Santo*, in 2017, more than 200 notifications remained pending.

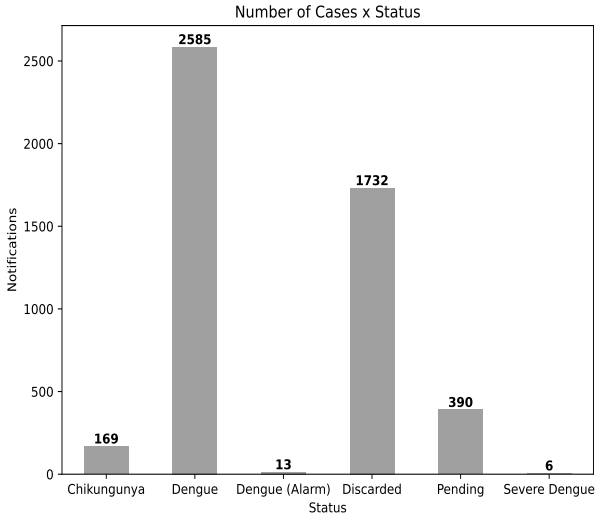
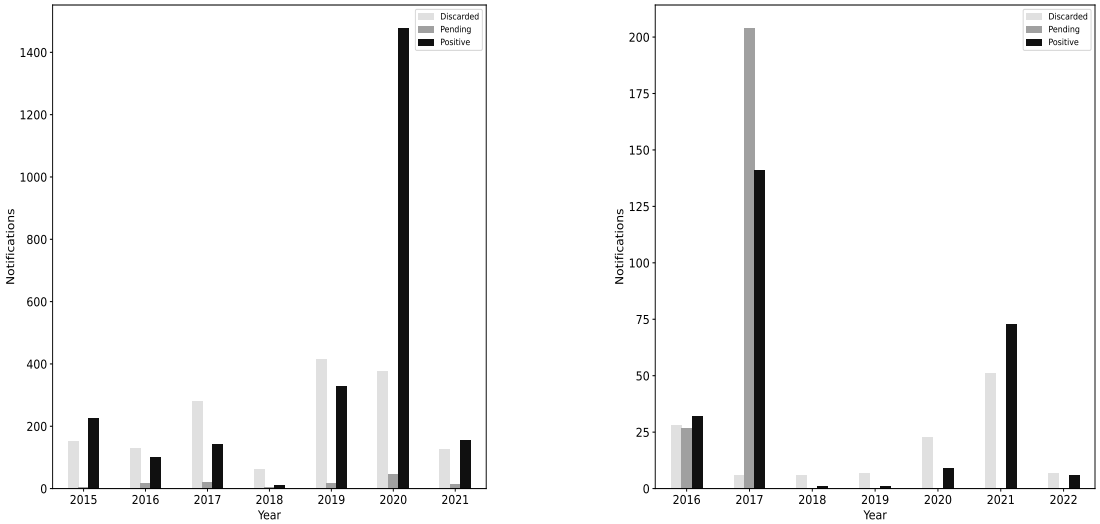


Figure 4.4: Number of notifications considering both cities under study.

Analyzing the data presented in Figure 4.5, except for 2018 in *Limoeiro do Norte*, there are at least 150 positive cases each year, with the highest number of cases occurring in 2019 and 2020. The year of 2020 was considered an epidemic year, reporting more than 1450 positive cases. In *Alto Santo*, more than half of the total notifications were reported only in 2017, with more cases pending than confirmed positive or negative. The number of positive notifications in all other years is lower than 75 and almost no cases in the two years after 2017. According to conversations with the *Alto Santo* health department, except for 2017, none of the years were characterized as epidemic seasons. The number of cases increased only in 2021.



(a) Notifications from Limoeiro.

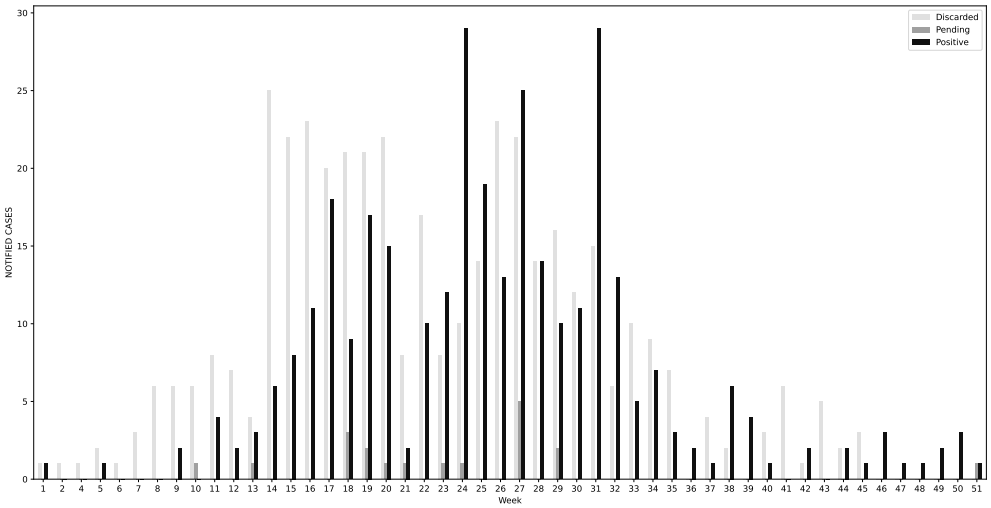
(b) Notifications from Alto Santo.

Figure 4.5: Number of notifications per year.

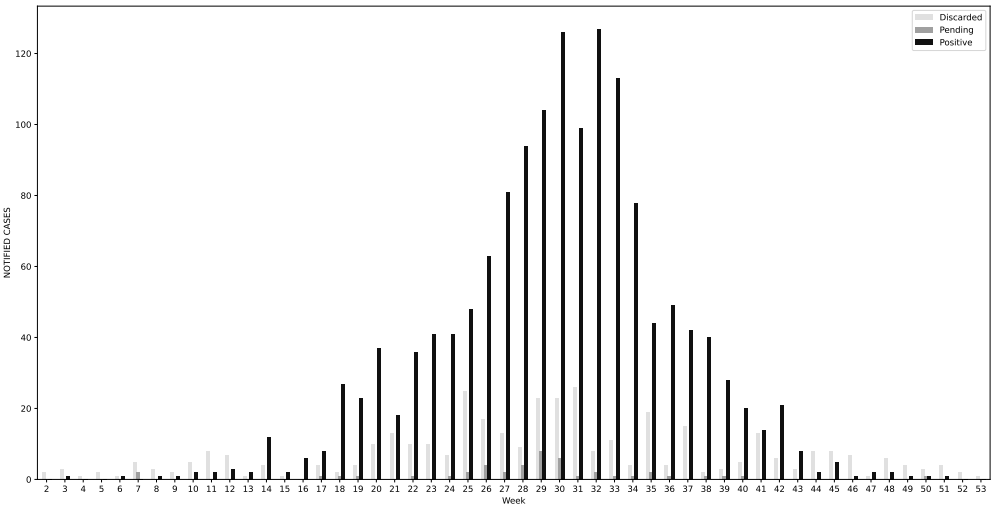
Considering the total number of notifications and their distribution over the years, it is evident that *Limoeiro do Norte*, with a larger population, also experiences a higher number of Dengue and Chikungunya cases every year. To analyze the distribution of cases in the same granularity of time used by the health departments, we selected the two years with the most notifications for each city, specifically 2019/2020 for *Limoeiro* and 2017/2021 for *Alto Santo* (Figures 4.6 and 4.7, respectively). These figures depict the distribution of notifications for each Epidemiological Week (EW) with at least one notification.

In *Alto Santo*, during 2017, dengue cases were reported within the first 16 epidemiological weeks (EW), peaking at 25 cases in week 4. Based on the application, we can classify these notifications as either positive or negative, given the high number of confirmed dengue cases during the same period. In 2021, cases were recorded between EW 14 and 48, with a lower weekly incidence, reaching a peak of 9 confirmed cases in week 21.

From *Limoeiro*, in 2019, cases are present in almost all weeks with peaks of positive cases occurring between weeks 24 and 32, ranging from 10 to 29 cases. There is an



(a) Notifications from Limoeiro - 2019.



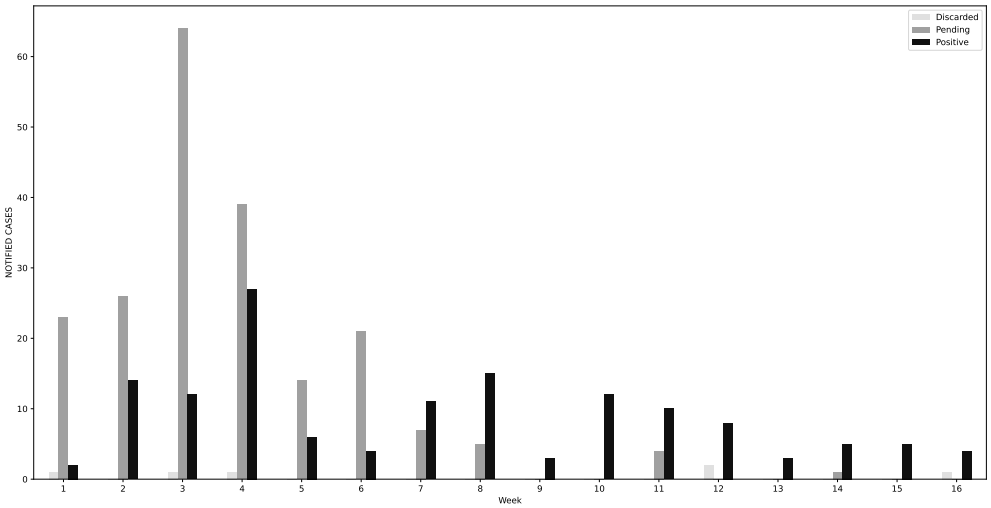
(b) Notifications from Limoeiro - 2020.

Figure 4.6: Number of notifications per week - Limoeiro do Norte.

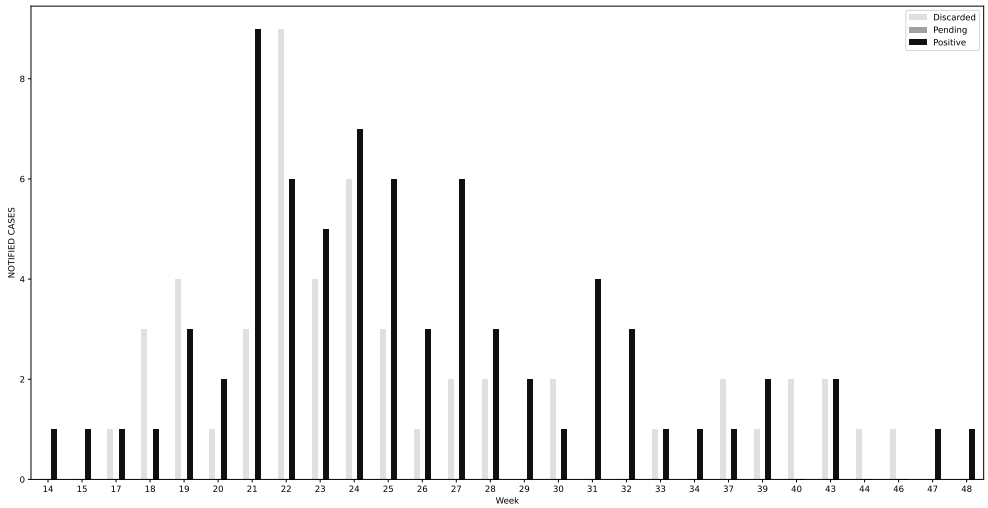
epidemic season from weeks 20 to 38 of 2020, where the number of cases is at least 40, and for 8 consecutive weeks after the 27 EW, the number of cases ranges from 80 to 123.

Two points need to be highlighted to understand this dataset. First, underreporting likely occurred as infected individuals may not have pursued hospital care due to: *(1)* Knowing Dengue is seasonal, individuals often self-diagnose when experiencing common disease symptoms, *(2)* adherence to clinically recommended protocols without formal testing, or *(3)* mild symptoms that did not disrupt daily activities. Second, the absence of spatiotemporal records on vector-control interventions (e.g., insecticide campaigns and larval habitat removal) obscures their impact on mosquito populations and localized transmission dynamics. These limitations collectively contribute to the observed fluctuations in notifications.

In both cities, public health strategies lack integration with digital surveillance tools or predictive systems. Unfortunately, they rely on reactive containment measures initiated only after spikes in confirmed dengue cases. This firefighting approach - common



(a) Notifications from Alto Santo - 2017.



(b) Notifications from Alto Santo - 2021.

Figure 4.7: Number of notifications per week - Alto Santo.

across resource-constrained municipalities in Brazil - fails to optimize prevention due to three key gaps: *(i)* delayed detection of epidemiological trends, *(ii)* absence of data-driven risk mapping to prioritize high-risk zones, and *(iii)* ad hoc resource allocation that overlooks transmission dynamics. Consequently, interventions often occur too late to curb outbreaks effectively, perpetuating cycles of suboptimal public health outcomes.

4.6 Statistical Analysis of Results

4.7 Instance Generation

To identify (label) each city block, we employ a graph-theoretical procedure that corresponds to finding the faces of a planar embedding [22]. A planar embedding, also called a rotation system, is defined for a planar graph by assigning, for each vertex ($i \in V$), a cyclic ordering $C(i)$ of its neighbors (typically in clockwise order). Intuitively,

this defines the clockwise order of streets around each intersection. This rotation system can be computed using the geometric slope of each incident arc [38]. As an illustrative example, consider the graph in 4.8. Its planar embedding can be described by the function C as shown in Table 4.2.

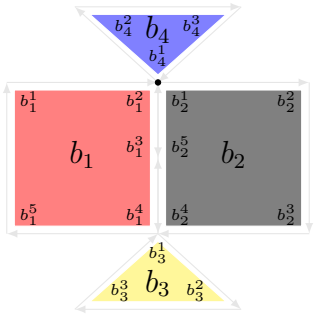


Figure 4.8: Example of non-isolated street-blocks.

Table 4.2: Connections for each city block

City Block	Connections
$C(b_1^1)$	(b_1^5, b_1^2)
$C(b_1^2)$	$(b_1^3, b_1^1, b_4^2, b_4^3, b_2^2)$
$C(b_1^3)$	(b_1^4, b_2^1)
$C(b_1^4)$	$(b_3^3, b_1^5, b_2^5, b_2^3, b_3^2)$
$C(b_1^5)$	(b_1^1, b_1^4)
$C(b_2^2)$	(b_2^1, b_2^3)
$C(b_2^3)$	(b_2^4, b_2^2)
$C(b_3^2)$	(b_3^3, b_3^1)
$C(b_3^3)$	(b_3^1, b_3^2)
$C(b_4^2)$	(b_4^3, b_4^1)
$C(b_4^3)$	(b_4^1, b_4^2)

With combinatorial embedding C and arc set A, we identify all bounded and unbounded faces of the planar graph using a traversal procedure formalized in Algorithm 2. Each face corresponds to a cyclic sequence of arcs that form the boundary of a region enclosed by streets (i.e., a city block). The algorithm iteratively traverses unused arcs in the graph by following the clockwise ordering defined by C, marking each arc as visited once it has been assigned to a face. Each traversal terminates upon returning to the starting arc, thus completing a cycle (face). This procedure ensures that every face is uniquely defined by a sequence of arcs traversed in the clockwise direction relative to the embedding. We note that every planar embedding has one unbounded 'outer' face encircling the whole graph. This outer face does not correspond to any physical city block, so we exclude it from the set of labeled blocks.

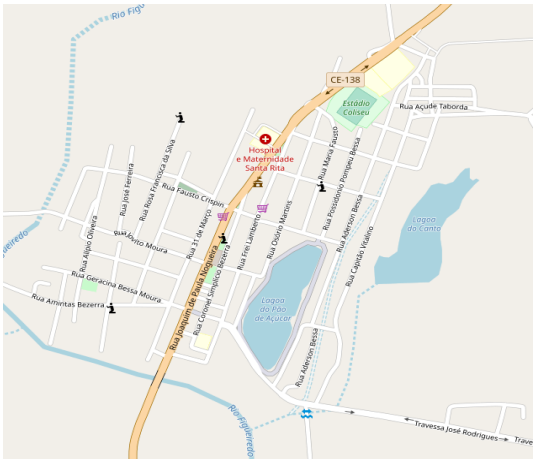
4.7.1 Real-case instances

We collected real-world dengue notification data from 2015 to 2021 through the municipal health departments of Alto Santo and Limoeiro do Norte, located in the state

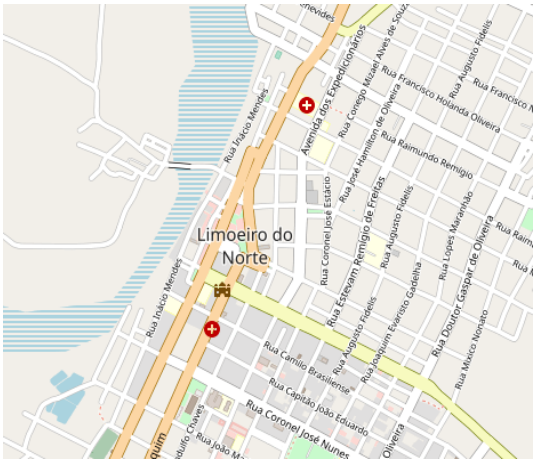
Algorithm 2: Face-finding algorithm for planar embedding

Require: Planar graph $D = (V, A)$, combinatorial embedding C
Ensure: Set of directed cycles $\mathcal{F}$ representing faces
Initialize $\mathcal{F} \leftarrow \emptyset$, and mark all arcs in A as unvisited
for each unvisited arc $a = (u, v) \in A$ **do**
  Set $a_0 \leftarrow a$, and mark a as visited
  repeat
    Append a to f
    Let w be the next vertex after u in $C(v)$ (i.e., clockwise successor of u)
    Set $a \leftarrow (v, w)$; mark a as visited
    Update $u \leftarrow v$, $v \leftarrow w$
  until $a = a_0$
  Add f to $\mathcal{F}$
end for
return $\mathcal{F}$

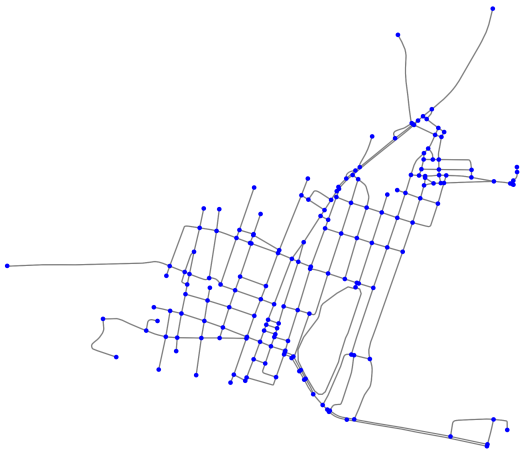
of Ceará, Brazil. The data were extracted from the SINAN [39]. During this period, Alto Santo reported 582 cases (2016-2021), and Limoeiro do Norte reported 4,243 cases (2015-2021), totaling 4,825 confirmed dengue cases between the two cities. Our computational benchmark consists of 39 instances based on these notifications. The addresses of reported cases were geolocated using the OSMnx library [10], which provided latitude and longitude coordinates. Each case was mapped to its corresponding city block, allowing us to integrate the case data with the street network. Figure 4.9 illustrates both the geographic maps and the corresponding graph models of Alto Santo and Limoeiro do Norte, as derived from OSM. Figure 4.3 summarizes the key characteristics of the benchmark used in the experimental evaluation.



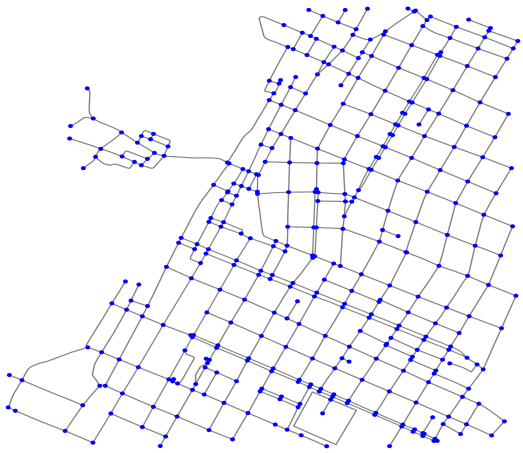
(a) Map of Alto Santo from OSM.



(b) Map of Limoeiro do Norte from OSM.



(c) Graph of Alto Santo.



(d) Graph of Limoeiro do Norte.

Figure 4.9: Representation of the two cities used in the benchmark.

Table 4.3: Benchmark main characteristics.

Instances	$\|V\|$	$\|A\|$	$\|B\|$	Notified Cases	Blocks with Cases (%)
AS-1000-2016	179	499	73	35	13.70
AS-1000-2017	179	499	73	194	36.99
AS-1000-2018	179	499	73	196	36.99
AS-1000-2019	179	499	73	196	36.99
AS-1000-2020	179	499	73	204	36.99
AS-1000-2021	179	499	73	218	36.99
AS-2000-2016	253	684	88	26	9.09
AS-2000-2017	253	684	88	174	29.55
AS-2000-2018	253	684	88	174	29.55
AS-2000-2019	253	684	88	174	29.55
AS-2000-2020	253	684	88	181	29.55
AS-2000-2021	253	684	88	195	30.68
AS-3000-2016	353	946	114	33	8.77
AS-3000-2017	353	946	114	198	24.56
AS-3000-2018	353	946	114	198	24.56
AS-3000-2019	353	946	114	198	24.56
AS-3000-2020	353	946	114	206	24.56
AS-3000-2021	353	946	114	225	24.56
LN-1000-2015	372	1063	152	18	7.89
LN-1000-2016	372	1063	152	26	10.53
LN-1000-2017	372	1063	152	50	17.11
LN-1000-2018	372	1063	152	51	17.11
LN-1000-2019	372	1063	152	66	20.39
LN-1000-2020	372	1063	152	163	29.61
LN-1000-2021	372	1063	152	168	31.58
LN-2000-2015	980	2866	443	174	6.77
LN-2000-2016	980	2866	443	337	9.03
LN-2000-2017	980	2866	443	527	12.19
LN-2000-2018	980	2866	443	543	12.64
LN-2000-2019	980	2866	443	875	14.22
LN-2000-2020	980	2866	443	2183	20.99
LN-2000-2021	980	2866	443	2316	21.67
LN-3000-2015	1212	3497	517	169	5.22
LN-3000-2016	1212	3497	517	336	7.54
LN-3000-2017	1212	3497	517	519	10.83
LN-3000-2018	1212	3497	517	533	11.03
LN-3000-2019	1212	3497	517	860	12.38
LN-3000-2020	1212	3497	517	2176	18.96
LN-3000-2021	1212	3497	517	2315	19.92

Chapter 5

Computational Experiments

5.1 Multi-Agent-Based Simulation

This section presents the results obtained from the MABS described in Section **??**. The implementation uses GAMA Platform version 1.9.2, the OSMnx Python Library [10] 1.9.3 was used to extract the map from OSM, and the street blocks are computed using the procedure described in Section **??**. The code is available in this public repository on Github[1].

This study explores different combinations of parameters from Table 5.1 to identify the optimal configuration for each city based on the average number (min and max) of cases per week in multiple simulation runs. The precision and reliability of the results are assessed through the Pearson correlation coefficient (r), Mean Absolute Error (MAE), and the analysis of the simulated endemic channel.

Table 5.1: Parameters for the initial state of the simulation.

Parameter	Description	Value
p	Number of people per m^2	0.006 - 0.01
H_b	Number of people in block b	$p \times A_b$
H_0	Starting size of human population	$\sum_{b=1}^{B} H_b$
M_h	Number of mosquitoes per human	0.50 - 3.00
M_b	Percent of infected mosquitoes in a block without notifications	0.05 - 0.30
M_b^i	Percent of infected mosquitoes in a block with at least one notification	0.40 - 0.90
BS	Number of potential Breeding Sites	50 - 500

5.1.1 Data Processing

First, the objective is to establish the initial population size by computing the p values. This work calculates the area of each street block as a polygon using the Shapely[2] and Geopandas[3] Python libraries with data imported from OSM. Since the investigation

[1] `https://github.com/cvaraujo/dengue-cbrp-mabs`
[2] `https://shapely.readthedocs.io/en/stable/`
[3] `https://geopandas.org/en/stable/`

in this work focuses mainly on urban areas, and both cities have vast territorial extensions that are predominantly composed of rural areas, their overall population densities reported in IBGE are low. For this reason, this work restricts the area considered in the population density calculation to the city center and strictly adjacent zones. However, there are no population data available at the level of blocks or neighborhoods, and alternative sources were needed to improve the values that estimate the resident population in the cities and then define the initial population parameters for the simulation.

The choice of using a parameter (p) defined as persons per square meter (m^2) ensures a good level of generalization for the simulation, as the presence of this value allows the population to remain proportionally adjusted for different experiments changing the size of the city map. This can be particularly useful, for example, if we wish to simulate the evolution of dengue cases in a single neighborhood.

Based on data provided by OSM, the urban area considered in Alto Santo covers approximately 0.9 km^2, with an estimated population of 9,000 people living in the city centers. This yields a population density of $p = 0.01$ people per m^2. For Limoeiro, the approximate urban area has 6 km^2 and a population of 36000, which corresponds to a value of $p = 0.006$ people per m^2. Since this MABS positions the individuals within blocks, the starting human population consists of the sum of p multiplied by the area of the corresponding block. For Alto Santo, using a radius of 700 m inside OSMnx, 73 street blocks are generated and the starting human population is $H_0 = 5688$. In Limoeiro, with a large urban area, the radius of 2000 m provides 563 blocks and an initial population $H_0 = 29914$.

The starting size of the mosquito population is determined by $m = H_b \times M_h$, where the number of infected mosquitoes in the block is $M_b \times m$ for blocks without cases and $M_b^i \times m$ otherwise. The starting number of potential breeding sites is related to the parameter BS and the blocks. For each block, if there is at least one positive case, then there is at least one breeding site. At the end of the initial scenario creation, if the breeding sites located are fewer than BS, then the remaining were placed in random blocks.

The MABS initialize using epidemiological data from a specified start date, incorporating historical notifications from the preceding week as confirmed cases within the simulation blocks. This methodology is in line with standard public health surveillance practices. To validate accuracy and quality, the simulated case projections are systematically compared with actual weekly reported cases over a 90-day evaluation period (equivalent to 180 simulation cycles after initialization). The selected starting dates are selected based on epidemiological periods within a reasonable number of notifications, as explored in Section **??**.

The parameters tested were $M_h = [0.5, 1, 2, 3]$, $M_b = [0.05, 0.1, 0.15, 0.2, 0.3]$, $M_b^i = [0.4, 0.5, 0.6, 0.7, 0.8]$, and $BS = [50, 100, 200, 300, 500]$. The analysis aggregates the results of 100 independent simulation trials for each parameter configuration and start date, ensuring statistical robustness in assessing model performance. The default maps of Alto Santo and Limoeiro consider OSMnx radius of 700 m and 2000 m, respectively.

An independent set of initial dates was defined for each city. The objective of the parameter exploration is to determine the best configuration based on similarity with historical real-world data, which were mapped to the same map segment used in the

simulation. One set of parameters is considered better than another using the number of "wins" for the dates on which these three criteria, in the following order of precedence, were better: (1) higher correlation between the simulated mean and the real data; (2) lower MAE between the simulated mean and the historical data; (3) number of weeks in which the real case notifications fall within the Simulation Endemic Channel, i.e., the range between the highest and lowest number of simulated cases in a given week. The concept of endemic channel is used by public health departments to assess how the number of cases in the current EW compares with the historical in previous years.

5.1.2 Results for Alto Santo

To determine the start dates for the experiments, we selected the two years with the highest number of positive or pending notifications (2017 and 2021). The cases were then grouped by epidemiological week, as shown in Figure 4.7. The objective is to focus on periods with the highest number of cases and consistency between adjacent weeks, while avoiding intervals with few or no cases. The experiments explore different starting dates within these periods to analyze the simulation's behavior when initiated at distinct moments along the endemic curve. Only historical notifications that can be assigned to a block in the simulation are considered, ensuring a fair comparison.

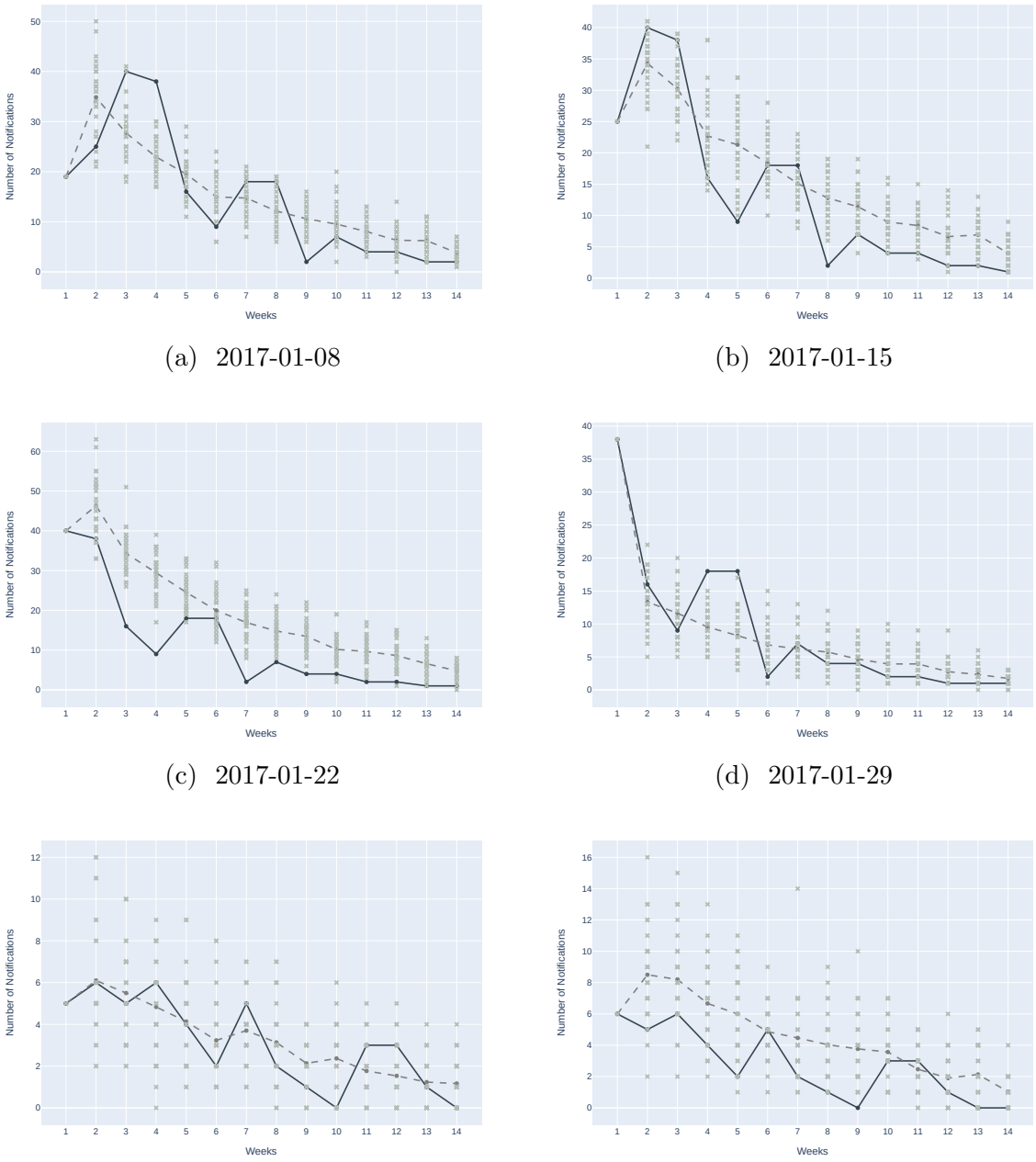
For better visualization of the results achieved by the simulator, Figure 5.1 presents the results for the best configuration considering the following starting dates: *(1)* 2017-01-08, *(2)* 2017-01-15, *(3)* 2017-01-22, *(4)* 2017-01-29, *(5)* 2021-05-30, and *(6)* 2021-06-06. The X-axis represents the week number, the Y-axis represents the number of cases, the solid black line represents the total real notifications in the week, the dashed line represents the average number of cases across simulations with the same starting scenario, and the light gray "×" markers represent the number of cases for a given simulation run.

The results represented in Figure 5.1 correspond to the variable parameters $M_h = 1$, $BS = 50$, $M_b = 0.05$ and $M_b^i = 0.4$, which is the combination that achieved the best results considering the highest correlation values (r) among all configurations for *Alto Santo*. More detailed information on these results is discussed later in this section. Table 5.2 shows the values of MAE, the correlation of the average simulated cases with the real notification (r) and the confidence interval (CI), in addition to the graphical results presented in Figure 5.1.

Table 5.2: Statistical values from the best configuration - *Alto Santo* city.

Date	**MAE**	r	CI_{min}	CI_{max}
2017-01-08	5.67	**0.83**	0.61	0.94
2017-01-15	5.20	**0.93**	0.82	0.97
2017-01-22	8.40	**0.90**	0.75	0.96
2017-01-29	2.80	**0.92**	0.80	0.97
2021-05-30	0.94	**0.84**	0.62	0.94
2021-06-06	1.92	**0.77**	0.49	0.91

All experiments strongly correlate with real cases, achieving correlation coefficients of



(a) 2017-01-08

(b) 2017-01-15

(c) 2017-01-22

(d) 2017-01-29

(e) 2021-05-30

(f) 2021-06-06

Figure 5.1: Comparison of simulation and real cases from *Alto Santo* for different starting dates.

at least 0.77 with positive confidence intervals. This strong correlation indicates that the average number of simulated cases per week closely matches the real number of cases. The endemic channel generated in each image of 5.1 has a difference of at most 10 cases from the real notifications when they are outside the endemic channel for a given EW, which represents a small deviation from reality.

5.1.3 Results for Limoeiro

The years 2019 and 2020 are selected for *Limoeiro* city, and the number of cases per week is shown in Figure 4.6. Figure 5.2 presents the results for the following starting dates: *(1)* 2019-04-14, *(2)* 2019-05-05, *(3)* 2020-06-21, *(4)* 2020-07-05, *(5)* 2020-07-19, and *(6)* 2020-07-26. Due to the extreme difference in the number of cases between the two years for *Limoeiro*, where 2020 has more than 1000 notifications than 2019, it was not possible to define a set of parameters that fit well for both years with such different samples and behavior. Figures 5.2a and 5.2b show the best results obtained with the parameters $M_h = 0.5$, $BS = 100$, $M_b = 0.05$ and $M_b^i = 0.4$. Figures 5.2c to 5.2f present the results with the following parameters set $M_h = 1$, $BS = 500$, $M_b = 0.20$ and $M_b^i = 0.9$.

Table 5.3 shows the values of MAE, the correlation of the average simulated cases with the real notification (r), and the confidence interval of the results presented in Figures 5.2.

Table 5.3: Statistical values from the best configuration - *Limoeiro* city.

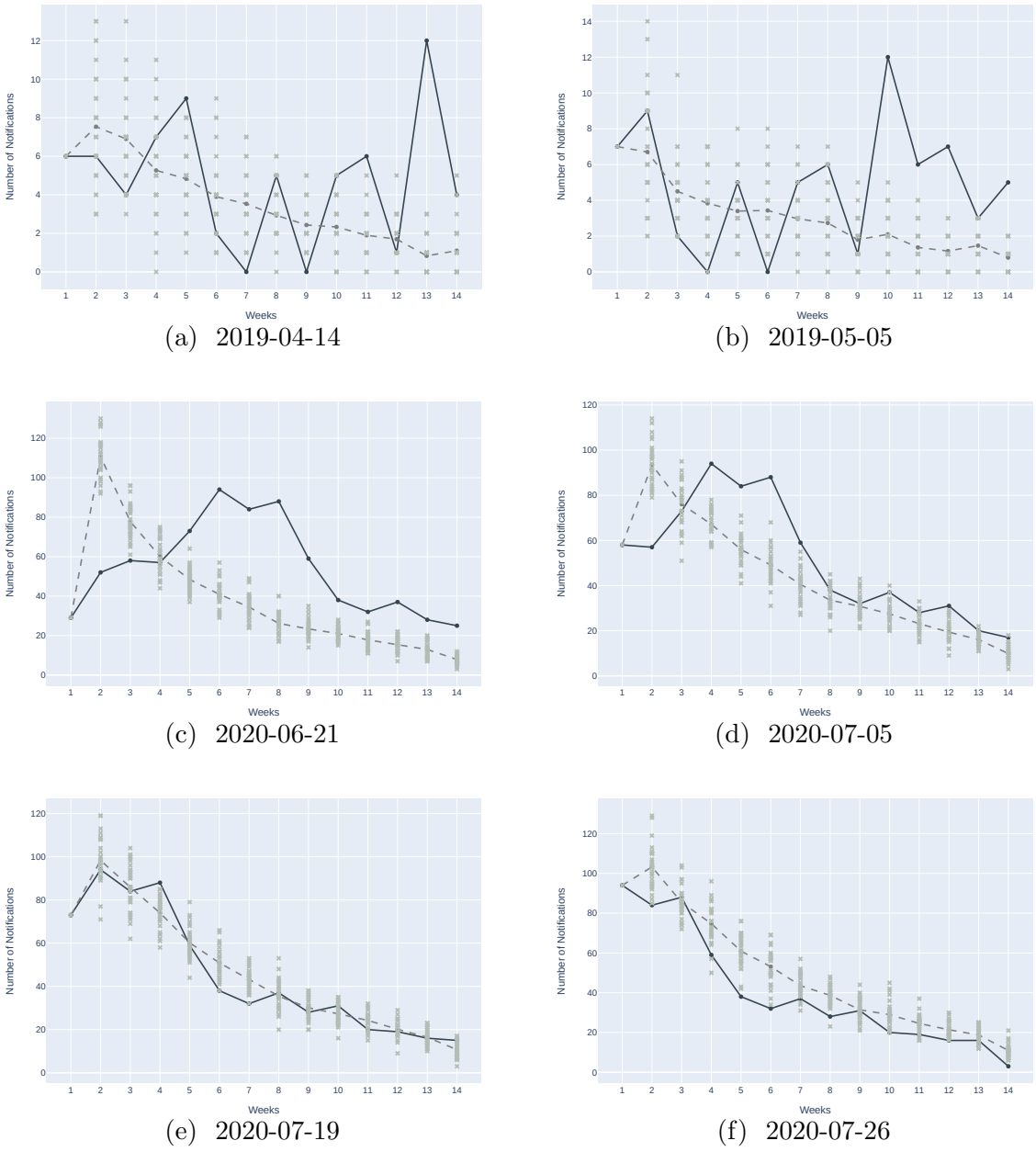
Date	**MAE**	r	CI_{min}	CI_{max}
2019-04-14	2.99	0.06	-0.41	0.51
2019-05-05	3.28	0.09	-0.38	0.53
2020-06-21	27.97	0.30	-0.18	0.67
2020-07-05	13.90	**0.76**	0.45	0.90
2020-07-19	4.49	**0.97**	0.93	0.99
2020-07-26	9.27	**0.96**	0.90	0.99

In 2019, the analysis of *Limoeiro* City's epidemiological data revealed a weak correlation (values approaching zero), as indicated by the experimental results. However, the associated MAE metrics were modest, ranging from 2.99 to 3.28. Figures 5.2a and 5.2b illustrate the weekly case distribution, highlighting irregular temporal patterns despite the low overall incidence (with a peak of 12 notifications in the highest week). In particular, only 4 of the 14 analyzed EW exceeded the boundaries of the simulated endemic channel.

Discrepancies between observed and expected case counts were predominantly minor (1-2 cases), except for the 10^{th} and 13^{th} weeks, which each recorded 12 notifications and deviated by 9 cases from the channel's upper limit. This suggests that while the statistical correlation was weak, most of the weekly case counts aligned with or remained close to the endemic channel. The results show that even in low-incidence settings, localized fluctuations can occur without systematically breaching expected epidemiological thresholds.

Figures 5.2c and 5.2d represent two simulations started two weeks apart. In the first scenario, starting in 2020-06-21, the model yields a weak correlation ($r = 0.3$), as it anticipates a case peak in the initial weeks, while the observed data show a delayed surge in the sixth week. This temporal misalignment probably reflects limitations in model calibration, including insufficient historical data and an inability to integrate time-dependent variables that influence case trajectories.

In contrast, the second simulation, considering the start date of 2020-07-05 and the initiation of the simulation two weeks later, shows a strong correlation ($r = 0.76$) with



(a) 2019-04-14 (b) 2019-05-05

(c) 2020-06-21 (d) 2020-07-05

(e) 2020-07-19 (f) 2020-07-26

Figure 5.2: Comparison of simulation and real cases from *Limoeiro* for different starting dates.

a positive confidence interval (CI), indicating a significantly improved alignment with real-world trends. Here, the later start date probably allowed the model to incorporate critical early phase data, enhancing its predictive accuracy. These results highlight the sensitivity of epidemiological models to initialization timing and the importance of adaptive calibration to account for dynamic real-world factors.

Figures 5.2e–5.2f demonstrate near-perfect correlation coefficients ($r = 0.97$ and $r = 0.96$, respectively), reflecting an exceptional alignment between simulated and observed case trends. These simulations were initialized during peak cases in incidence periods for both cities, the highest recorded in the notification dataset. The proximity

of these starting dates to the epidemiological peak likely allowed the model to capture critical transmission dynamics, resulting in highly accurate projections. This outcome confirms robustness when calibrated during periods of increased disease activity, where the underlying patterns may be more pronounced and predictable.

5.1.4 Simulation Quality Assessment

Taking into account all the results presented, it is possible to highlight the strengths of our MABS that achieve good results for all simulations in *Alto Santo*. These outcomes present a strong correlation for all cases and a good fit of the real data inside the simulated endemic channel. The MABS also presents a strong correlation for half of the simulations for *Limoeiro* and a good adjustment of the endemic channel to contain the historical progression of notifications.

In some instances, the simulation's inability to accurately capture real-world data may be attributed to unmodeled factors. For example, the model currently does not account for the influence of climate, vector control efforts, under-reporting, population density, and geographic features like vacant land and proximity to water bodies. These factors can significantly impact disease transmission dynamics, however, may not be fully represented in the current simulation model.

Although the current MABS framework cannot dynamically adapt parameters in response to real-time data, it remains a robust analytical tool when combined with the calibration of strategic parameters and domain expertise. By integrating the principles of MABS with epidemiological insights specific to dengue transmission, researchers can identify optimal parameter configurations that produce highly accurate simulations. Inclusion of real-time data streams or detailed geographic information could further improve the precision of the model. However, such enhancements are not prerequisites for generating meaningful results.

Thus, it is possible to claim that the proposed MABS is a reliable and accurate simulation methodology, capable of generating scenarios that account for both casual disease spread and endemic seasons in the cities of *Alto Santo* and *Limoeiro*, even with limitations in accurate and essential information. The methodology can be further enhanced with additional data, and the simplified programming language facilitates the integration of new features and information into the MABS. This approach can be used as a visualization and decision support tool, primarily considering the generated endemic channel and the fact that it is easy to simulate for other cities assuming the existence of basic geographic information and historical data to improve the best fit of the variable parameters.

Chapter 6

Conclusions

Bibliography

[1] Afeez Abidemi and Nur Arina Bazilah Aziz. Optimal control strategies for dengue fever spread in johor, malaysia. *Computer Methods and Programs in Biomedicine*, 196:105585, 2020.

[2] Marcos Amaku, Franciane Azevedo, Marcelo Burattini, Francisco Coutinho, Luis Fernandez Lopez, and Eduardo Massad. Interpretations and pitfalls in modelling vector-trasmitted infections. *Epidemiology and infection*, 143:1–13, 2014.

[3] Edouard Amouroux, Stéphanie Desvaux, and Alexis Drogoul. Towards virtual epidemiology: An agent-based approach to the modeling of h5n1 propagation and persistence in north-vietnam. In *Intelligent Agents and Multi-Agent Systems*, pages 26–33, Berlin, Heidelberg, 2008. Springer Berlin Heidelberg.

[4] Matheus Andrade, Fábio Luiz Usberti, and Rafael Kendy Arakaki. Routing vehicles for dengue control. In *LIII Brazilian Symposium on Operational Research (SBPO)*, pages 1–12, Paraíba, João Pessoa, 2021. SOBRAPO.

[5] Annalisa Appice, Yulia R. Gel, Iliyan Iliev, Vyacheslav Lyubchich, and Donato Malerba. A multi-stage machine learning approach to predict dengue incidence: A case study in mexico. *IEEE*, 8:52713–52725, 2020.

[6] Carlos V. Dantas Araújo, Matheus Andrade, Fábio Luiz Usberti, and Rafael Kendy Arakaki. A prize-collecting approach for the dengue arc routing problem. In *LIV Brazilian Symposium on Operational Research (SBPO)*, pages 1–12, Minas Gerais, Juiz de Fora, 2022. SOBRAPO.

[7] Ariel Esteban Bardach, Herney Andrés García-Perdomo, Andrea Alcaraz, Elena Tapia López, Ruth Amanda Ruano Gándara, Silvina Ruvinsky, and Agustín Ciapponi. Interventions for the control of aedes aegypti in latin america and the caribbean: systematic review and meta-analysis. *Tropical Medicine & International Health*, 24(5):530–552, 2019.

[8] John E. Beasley. Lagrangian relaxation. In Colin R. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Problems*, chapter 6, pages 243–303. John Wiley & Sons, Inc., New York, NY, USA, 1993.

[9] John R. Birge and Franois Louveaux. *Introduction to Stochastic Programming*. Springer Publishing Company, Incorporated, 2nd edition, 2011.

[10] Geoff Boeing. Osmnx: New methods for acquiring, constructing, analyzing, and visualizing complex street networks. *Computers, Environment and Urban Systems*, 65:126–139, 2017.

[11] Eric Bonabeau. Agent-based modeling: Methods and techniques for simulating human systems. *Proceedings of the National Academy of Sciences of the United States of America*, 99 Suppl 3:7280–7, 2002.

[12] Francisco Borges, Albert Gutierrez-Milla, Remo Suppi, Emilio Luque, and Marylene De Brito Arduino. An agent-based model for assessment of aedes aegypti pupal productivity. In *2015 Winter Simulation Conference (WSC)*, pages 159–170, 2015.

[13] Kris Braekers, Katrien Ramaekers, and Inneke Van Nieuwenhuyse. The vehicle routing problem: State of the art classification and review. *Computers & industrial engineering*, 99:300–313, 2016.

[14] Fred Brauer. *Compartmental Models in Epidemiology*, pages 19–79. Springer Berlin Heidelberg, 2008.

[15] R. Daniela Cantane, C. Ariane Cristino, A. Rogério Oliveira, A. R. Marcos Fernandes, E. M. Paulo Ribolla, and A. S. Jayne Neto. O desenvolvimento da população do aedes aegypti aplicado ao modelo de otimização no controle da dengue. In *Anais do XLVII Simpósio Brasileiro de Pesquisa Operacional (SBPO)*, pages 2961–2972, Porto de Galinhas, Pernambuco, 2015. SOBRAPO.

[16] S. A. Carvalho, da Silva S. O., and I. D. C. Charret. Mathematical modeling of dengue epidemic: control methods and vaccination strategies. *Theory in Biosciences*, 138:223–239, 2019.

[17] S. M Cavany, A. L. España, G. Lloyd, G.M. Vazquez-Prokopec, H. Astete, L. A. Waller, U. Kitron, A. C. Scott, T. W. Morrison, and T. A. Reiner, R. C. Jr. Perkins. Fusing an agent-based model of mosquito population dynamics with a statistical reconstruction of spatio-temporal abundance patterns. *PLoS Comput Biol*, 19(4), 2023.

[18] Angel Corberán, Richard Eglese, Geir Hasle, Isaac Plana, and José María Sanchis. Arc routing problems: A review of the past, present, and future. *Networks*, 77(1):88–115, 2021.

[19] Letícia da Silva Rodrigues, Sóstenes Gutembergue Mamedio Oliveira, Luiz Fernandez Lopez, and Jaime Simão Sichman. Agent based simulation of the dengue virus propagation. In Mario Paolucci, Jaime Simão Sichman, and Harko Verhagen, editors, *Multi-Agent-Based Simulation XX*, pages 100–111, Cham, 2020. Springer International Publishing.

[20] Brasil Health Department. Vinte estados apresentam tendência de estabilidade ou queda na incidência de dengue, 2024.

[21] Reinhard Diestel. *Graph Theory*. Springer, 2005.

[22] Reinhard Diestel. *Graph theory*. Springer (print edition); Reinhard Diestel (eBooks), 2024.

[23] Célia A. dos Reis, Helenice de O. Florentino, Diego Cólon, Suélia R. Fleury Rosa, and Daniela R. Cantane. An approach of the exact linearization techniques to analysis of population dynamics of the mosquito aedes aegypti. *Mathematical Biosciences*, 299:51–57, 2018.

[24] Ananya Dwivedi, Ram Keval, and Subhas Khajanchi. Modeling optimal vaccination strategy for dengue epidemic model: a case study of india. *Physica Scripta*, 97:1–21, 2022.

[25] Rafaelle C. G. Fares, Souza P. R., Añez G., and Rios M. Epidemiological scenario of dengue in brazil. *BioMed Research International*, 2015:1–13, 2015.

[26] Helenice O. Florentino, Daniela R. Cantane, Fernando L.P. Santos, Célia A. Reis, Margarida V. Pato, Dylan Jones, Marianna Cerasuolo, Rogério A. Oliveira, and Luiz G. Lyra. Genetic algorithm for optimization of the aedes aegypti control strategies. *Pesquisa Operacional*, 38:389–411, 12 2018.

[27] Angus Forbes and Peter Griffiths. Methodological strategies for the identification and synthesis of 'evidence' to support decision-making in relation to complex healthcare systems and practices. *Nursing Inquiry*, 9(3):141–155, 2002.

[28] Bernardo Furtado, Patricia Sakowski, Marina Tóvolli, William Rand, Miguel Fuentes, James Gentile, Chris Glazner, Matthew Koehler, Jaime Sichman, Masaru Yarime, Ali Kharrazi, Claudio Tessone, Orlando Gomes, Herbert Dawid, Luís Bettencourt, Yaneer Bar-Yam, Bernardo Mueller, Dick Ettema, Michael Jacobson, and Marcos Silva. *Modeling Complex Systems for Public Policies*. Ipea, 2015.

[29] Antonio Gloria, Muhammad Umar, Zulqurnain Sabir, Muhammad Asif Zahoor Raja, K. S. Al-Basyouni, S. R. Mahmoud, and Yolanda Guerrero Sánchez. An advance computing numerical heuristic of nonlinear sir dengue fever system using the morlet wavelet kernel. *Journal of Healthcare Engineering*, 2022:9981355, 2022.

[30] Héctor Gómez-Dantés and Janine Ramsey Willoquet. Dengue in the americas: challenges for prevention and control. *Cadernos de saúde pública*, 25:S19–S31, 2009.

[31] Ana Alicia Gramajo, Karina Laneri, and María Fabiana Laguna. Mosquito populations and human social behavior: A spatially explicit agent-based model. *Phys. Rev. E*, 106:034–405, 2022.

[32] Gleicy Hair, Flávio Nobre, and Patrícia Brasil. Characterization of clinical patterns of dengue patients using an unsupervised machine learning approach. *BMC Infectious Diseases*, 19:1–11, 2019.

[33] Nur 'Izzati Hamdan and Adem Kilicman. The development of a deterministic dengue epidemic model with the influence of temperature: A case study in malaysia. *Applied Mathematical Modelling*, 90:547–567, 2021.

[34] O. Florentino Helenice, R. Cantane Daniela, Célia A. dos Reis, Diego Colón, and Suélia S. R. F Rosa. Algoritmo de busca em vizinhança variável para determinação de controle otimizado para a população de aedes aegypti. In *XXXVII CNMAC*, pages 1–7, São José dos Campos, São Paulo, 2018. Proceeding Series of the Brazilian Society of Computational and Applied Mathematics.

[35] Qinlong Jing and Ming Wang. Dengue epidemiology. *Global Health Journal*, 3(2):37–45, 2019.

[36] Angel A. Juan, Javier Faulin, Scott E. Grasman, Markus Rabe, and Gonçalo Figueira. A review of simheuristics: Extending metaheuristics to deal with stochastic combinatorial optimization problems. *Operations Research Perspectives*, 2:62–72, 2015.

[37] Angel A. Juan, Yuda Li, Majsa Ammouriova, Javier Panadero, and Javier Faulin. Simheuristics: An introductory tutorial. In *2022 Winter Simulation Conference (WSC)*, pages 1325–1339, 2022.

[38] Philip Klein and Shay Mozes. Optimization algorithms for planar graphs. `https://www.planarity.org/`, 2011.

[39] Josué Laguardia, Carla Magda Allan Domingues, Carolina Carvalho, Carlos Rodrigo Lauerman, Eduardo Macário, and Ruth Glatt. Sistema de informação de agravos de notificações (sinan): desafios no desenvolvimento de um sistema de informações em saúde. *Epidemiologia e serviços de Saúde*, 13:135–146, 2004.

[40] Mathieu Legros, Marcelo Otero, Victoria Romeo Aznar, Hernan Solari, Fred Gould, and Alun L. Lloyd. Comparison of two detailed models of aedes aegypti population dynamics. *Ecosphere*, 7(10):1500–1515, 2016.

[41] Rachel Lowe, Bernard Cazelles, Richard Paul, and Xavier Rodo. Quantifying the added value of climate information in a spatio-temporal dengue model. *Stochastic Environmental Research and Risk Assessment*, 30:2067–2078, 2015.

[42] Thomas Lucas, David Kelton, Paul Sanchez, Susan Sanchez, and Ben Anderson. Changing the paradigm: Simulation, now a method of first resort. *Naval Research Logistics (NRL)*, 62, 2015.

[43] Imran Mahmood, Mishal Jahan, Derek Groen, Aneela Javed, and Faisal Shafait. An agent-based simulation of the spread of dengue fever. In Valeria V. Krzhizhanovskaya, Gábor Závodszky, Michael H. Lees, Jack J. Dongarra, Peter M. A. Sloot, Sérgio Brissos, and João Teixeira, editors, *Computational Science – ICCS 2020*, pages 103–117, Cham, 2020. Springer International Publishing.

[44] Somsakun Maneerat and Eric Daudé. A spatial agent-based simulation model of the dengue vector aedes aegypti to explore its population dynamics in urban areas. *Ecological Modelling*, 333:66–78, 2016.

[45] World Health Organization. World health organization. dengue: Guidelines for diagnosis, treatment, prevention and control, 2009.

[46] Xin-You Meng and Chong-Yang Yin. Dynamics of dengue fever model with unreported cases and asymptomatic infected classes in singapore, 2020. *Journal of Applied Analysis & Computation*, 13(2):782–808, 2023.

[47] M. Negreiros, N. Maculan, and A. Xavier. Projeto webdengue: Evolução e recursos para implantação. Technical Report 1, UFRJ, 2020.

[48] Marcos Negreiros, Adilson Xavier, Airton Xavier, Nelson Maculan, Philippe Michelon, Jose Lima, and Luis Andrade. *Optimization Models, Statistical and DSS Tools for Prevention and Combat of Dengue Disease*, chapter 7, pages 115–160. IntechOpen, 2011.

[49] Marcos José Negreiros, Airton Fontenele Sampaio Xavier, José Wellington de Oliveira Lima, Adilson Elias Xavier, Nelson Maculan, and Philippe Michelon. Integração de sistemas computacionais e modelos logísticos de otimização para prevenção e combate à dengue. *Pesquisa Operacional*, 28:1–27, 2008.

[50] George L. Nemhauser and Laurence A. Wolsey. *Integer and Combinatorial Optimization*. Wiley-Interscience, New York, NY, USA, 1988.

[51] Brazilian Department of Health. Diretrizes nacionais para a prevenção e controle de epidemias de dengue. `https://bvsms.saude.gov.br/bvs/publicacoes/diretrizes_nacionais_prevencao_controle_dengue.pdf`, 2009. Accessed: 05-22-2022.

[52] World Health Organization. Epidemiological update dengue and other arboviruses, 2024.

[53] Luba Pascoe, Devotha Godfrey Nyambo, Karen Bradshaw, and Thomas Clemen. Agent-based modeling and simulation for transmission dynamics and surveillance of dengue: Conceptual and design model. In *2023 IEEE AFRICON*, pages 1–6, 2023.

[54] Damien Philippon, Marc Choisy, Alexis Drogoul, Benoit Gaudou, Nicolas Marilleau, Patrick Taillandier, and Quang Chi Truong. Exploring trade and health policies influence on dengue spread with an agent-based model. In *Multi-Agent Based Simulation XVII*, pages 111–127. Springer International Publishing, 2017.

[55] Puntipa Pongsumpun, Jiraporn Lamwong, I-Ming Tang, and Puntani Pongsumpun. A modified optimal control for the mathematical model of dengue virus with vaccination. *AIMS Mathematics*, 8(11):27460–27487, 2023.

[56] T. Prasetyo, R. Saragih, and D. Handayani. Genetic algorithm to optimization mobility-based dengue mathematical model. *International Journal of Electrical and Computer Engineering*, 13(4):4535–4546, 2023.

[57] M. Rabe, M. Deininger, and A.A. Juan. Speeding up computational times in simheuristics combining genetic algorithms with discrete-event simulation. *Simulation Modelling Practice and Theory*, 103:102089, 2020.

[58] Abdur Rais and Ana Viana. Operations research in healthcare: a survey. *International Transactions in Operational Research*, 18(1):1–31, 2011.

[59] Dhiman Sarma, Sohrab Hossain, Tanni Mittra, Md. Abdul Motaleb Bhuiya, Ishita Saha, and Ravina Chakma. Dengue prediction using machine learning algorithms. In *8th R10 Humanitarian Technology Conference (R10-HTC)*, pages 1–6. IEEE, 2020.

[60] SESA. Nebulizer attached in vehicle. `https://www.saude.ce.gov.br/2022/02/16/notificacoes-de-casos-de-arboviroses-crescem-\e-sesa-inicia-operacao-fumace-no-cariri`, 2025. Accessed: 2025-06-17.

[61] Kashish Ara Shakil, Shadma Anis, and Mansaf Alam. Dengue disease prediction using weka data mining tool. *ArXiv*, 1502:1–25, 2015.

[62] Kamran Shaukat Dar, Nayyer Masood, Sundas Mehreen, and Ulya Azmeen. Dengue fever prediction: A data mining problem. *Journal of Data Mining in Genomics & Proteomics*, 06(03):2–5, 2015.

[63] Donald Shepard, Eduardo Undurraga, Yara Halasa, and Jeffrey Stanaway. The global economic burden of dengue: A systematic analysis. *The Lancet Infectious Diseases*, 16:935–941, 2016.

[64] Akhil Kumar Srivastav, Vanessa Steindorf, Nico Stollenwerk, and Maíra Aguiar. The effects of public health measures on severe dengue cases: An optimal control approach. *Chaos, Solitons & Fractals*, 172:113–577, 2023.

[65] Patrick Taillandier, Benoit Gaudou, Arnaud Grignard, Quang-Nghi Huynh, Nicolas Marilleau, Philippe Caillou, Damien Philippon, and Alexis Drogoul. Building, composing and experimenting complex spatial models with the gama platform. *GeoInformatica*, 23(2):299–322, 2019.

[66] Maria Sofía Uribe, Mariajose Franco, Luisa F. Londoño, Paula Escudero, Susana Álvarez, and Rafael" Mateus. Comparing three agent-based models implementations of vector-borne disease transmission dynamics. In Marta Tabares, Paola Vallejo, Biviana Suarez, Marco Suarez, Oscar Ruiz, and Jose Aguilar, editors, *Advances in Computing*, pages 330–349, Cham, 2024. Springer Nature, Switzerland.

[67] WHO. List of who prequalified vector control products, 8 2020. Accessed: 4-24-2021.

[68] Xiaolan Xie and Mark A. Lawley and. Operations research in healthcare. *International Journal of Production Research*, 53(24):7173–7176, 2015.

[69] Ling Xue, Xue Ren, Felicia Magpantay, Wei Sun, and Huaiping Zhu. Optimal control of mitigation strategies for dengue virus transmission. *Bulletin of Mathematical Biology*, 83:8, 2021.

[70] Ling Xue, Hongyu Zhang, Wei Sun, and Caterina Scoglio. Transmission dynamics of multi-strain dengue virus with cross-immunity. *Applied Mathematics and Computation*, 392:1–24, 2021.