UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

LEONARDO DA LUZ DORNELES

A GIS-based Methodology for Optimal Allocation of Acute Stroke Centers across the Globe: A Case Study in Rio Grande do Sul

Thesis presented in partial fulfillment
of the requirements for the degree of
Master of Computer Science

Advisor: Prof. Dr. Márcio Dorn

Porto Alegre
March 2024

CIP — CATALOGING-IN-PUBLICATION

da Luz Dorneles, Leonardo

A GIS-based Methodology for Optimal Allocation of Acute Stroke Centers across the Globe: A Case Study in Rio Grande do Sul / Leonardo da Luz Dorneles. – Porto Alegre: PPGC da UFRGS, 2024.

105 f.: il.

Thesis (Master) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR–RS, 2024. Advisor: Márcio Dorn.

1. Multi-objective optimization. 2. NSGA-II. 3. GIS. 4. Stroke. 5. Facility location problem. I. Dorn, Márcio. II. Título.

ABSTRACT

Acute stroke is a major global cause of disability, particularly in low- and middle-income countries, responsible for 87% of related disabilities. Improving access to acute stroke centers (ASCs) with limited resources requires an efficient approach to expanding their number. Computational methods find in facility location problems (FLPs) the abstraction that enables the optimization of resource assignment to demand points in a network. Diverse applications, including emergency department and ambulance station locations, can be modeled as FLPs. Our proposed methodology utilizes Geographical Information Systems (GIS) tools and public datasets along with FLPs to assess and enhance a region's current network of acute stroke centers. The proposed FLP is a multi-objective model designed to maximize population coverage, minimize the average distance to the nearest acute stroke center, and reduce the number of new centers required. We tested the application of the Nondominated-Sorting Genetic Algorithm II (NSGA-II) for the optimization of the ASC network. We applied this methodology to Rio Grande do Sul, Brazil, as a case study. Our findings revealed significant accessibility issues in areas outside the metropolitan region. The optimization results suggest that adding 38 hospitals with minimal infrastructure could significantly enhance the current ASC network. Furthermore, to highlight the benefits of employing facility location problems, we analyze how accessibility to acute stroke treatment could improve if the state utilized our model to build the ASC network from scratch. Finally, we analyze the suitability of NSGA-II optimization in the proposed model.

Keywords: Multi-objective optimization. NSGA-II. GIS. stroke. facility location problem.

Uma metodologia baseada em SIG para a atribuição ótima de centros de tratamento de AVC agudo pelo mundo

RESUMO

O AVC agudo é uma das principais causas de sequelas graves a nível mundial, particularmente nos países de baixo e médio rendimento, os quais são responsáveis por 87% desses números. Aumentar o número de centros de tratamento de AVC tendo recursos limitados requer uma abordagem eficiente. Métodos computacionais encontram nos problemas de localização de instalações (FLPs) a abstração que permite a otimização da atribuição de recursos a pontos de procura numa rede. Diversas aplicações, incluindo a localização de departamentos de emergência e estações de ambulâncias, podem ser modelados como FLPs. A metodologia que propomos modela um FLP integrado a ferramentas de Sistemas de Informação Geográfica (GIS) e conjuntos de dados públicos para avaliar e melhorar a atual rede de centros de AVC de uma região. O FLP proposto é um modelo multi-objetivo projetado para maximizar a cobertura populacional, minimizar a distância média até o centro de AVC mais próximo e minimizar o número de novos centros necessários. Nós testamos a aplicação do algoritmo NSGA-II para a otimização da rede de centros de AVC. Aplicamos esta metodologia ao Rio Grande do Sul, Brasil, como um estudo de caso. Nossos resultados revelaram problemas significativos de acessibilidade em áreas fora da região metropolitana do estado. Os resultados da otimização sugerem que a adição de 38 hospitais com infraestrutura mínima poderia melhorar significativamente a atual rede de centros de AVC. Além disso, para destacar os benefícios da utilização de problemas de localização de instalações, analisamos como a acessibilidade ao tratamento de AVC agudo poderia melhorar se o estado utilizasse nosso modelo para construir a rede de ASC a partir do zero. Finalmente, analisamos se a aplicação do NSGA-II na otimização é adequada para o modelo proposto.

Palavras-chave: otimização multi-objetiva, sistemas de informação geográfica, NSGA-II, AVC, problema de localização de instalações.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

ACO	Ant-Colony Optimization
AIS	Acute Ischemic Stroke
ARC	American Red Cross
ASC	Acute Stroke Centers
BRKGA	Biased-Random Key Genetic Algorithm
CFLP	Capacitated Facility Location Problem
EA	Evolutionary Algorithm
EMO	Evolutionary Multi-objective Optimization
ES	Evolution Strategies
FLP	Facility Location Problem
GA	Genetic Algorithm
GD	Generational Distance
GHSL	Global Human Settlement Layer
GP	Genetic Programming
GIS	Geographical Information System
HC	Hill-Climbing
HOT	Humanitarian OpenStreetMap Team
HRSL	High Resolution Settlement Layer
HV	Hypervolume
IVT	Intravenous Thrombolysis
ISEA3H	Icosahedral Snyder Equal Area aperture 3 Hexagon
LAP	Location-Allocation Problem
MCLP	Maximal Covering Location Problem
MOEA	Multi-Objective Evolutionary Algorithm

MOEA/D	Multi-Objective Evolutionary Algorithm Based on Decomposition
MSF	Médecins Sans Frontiéres
NSGA	Nondominated Sorting Genetic Algorithm
OR	Operations Research
ORS	OpenRouteService
OSM	OpenStreetMap
PSO	Particle Swarm Optimization
PMLP	p-median Location Problem
PCLP	p-center Location Problem
PSC	Primary Stroke Center
PSCLP	Partial Set Covering Location Problem
RS	Rio Grande do Sul
SA	Simulated Annealing
SPEA	Strength Pareto Evolutionary Algorithm
SCLP	Set Covering Location Problem
TOPSIS	Technique for Order Preference by Similarity to an Ideal Solution
TS	Tabu-Search
UFLP	Uncapacitated Facility Location Problem
VGI	Volunteered Geographic Information
ZCTA	ZIP Code Tabulation Area

CONTENTS

1 INTRODUCTION

Stroke is the second leading cause of global disability, with 87 percent of stroke-related disability occurring in low- and middle-income countries (Kayola et al., 2023). Patients in rural areas of countries like the United States (Hammond et al., 2020) and Brazil (Almeida et al., 2021) face limited accessibility to specialized care, leading to higher in-hospital mortality rates compared to their urban counterparts. Rapid access to adequate treatment during the acute phase of stroke can significantly reduce the probability of harmful patient outcomes (Gaynor et al., 2022).

The pressing need to increase the population's access to Acute Stroke Centers (ASCs) at the right time is evident, particularly in developing countries and rural areas. ASCs are equipped with specialized teams capable of stabilizing and providing immediate care to individuals experiencing acute strokes. Despite the urgency and relevance of improving ASC accessibility, there is a scarcity of solutions addressing this problem. These works use Facility Location Problems (FLPs) to enhance accessibility to stroke treatment (Leira et al., 2012; Vogel et al., 2023).

FLPs find applications across various domains, supporting different analyses, planning, management, and decision-making contexts (Murray et al., 2019). These models aim to optimize the placement of facilities, such as hospitals, within a given region. FLPs play a vital role in diverse contexts, including emergency care facilities (McNamara et al., 2020), solid waste management (Khan; Vaezi; Kumar, 2018), and ambulance service locations (Kaveh; Mesgari, 2019).

In many cases, works on FLPs employ classical models such as p-median, p-center, set-covering location, and maximal covering location (Ahmadi-Javid; Seyedi; Syam, 2017). These models typically adopt a single-objective optimization approach, primarily focusing on a singular criterion, such as maximizing population coverage or minimizing travel distances. However, it is increasingly recognized that this single-objective approach may not adequately capture the complex and conflicting nature of real-world instances of FLPs. Consequently, various researchers have embraced addressing these complexities by formulating FLPs as multi-objective problems (Farahani; Seifi; Asgari, 2010).

Various techniques have emerged to address multi-objective optimization problems, each with its own strengths and applications. These include matheuristics (Zhang et al., 2022), integer programming (Karatas; Yakici, 2018), and metaheuristics (Liu et

al., 2020). However, it is with Multi-Objective Evolutionary Algorithms (MOEAs) that multi-objective optimization has truly gained immense popularity. MOEAs, inspired by the principles of evolutionary biology, emulate natural selection to approximate the Pareto front of the problem. The Pareto front represents a collection of solutions that capture the trade-offs between competing objectives. MOEAs offer a powerful and versatile approach to multi-objective optimization, making them particularly well-suited for complex decision-making scenarios.

In modeling multi-objective FLPs effectively for real-world applications, incorporating a Geographic Information System (GIS) is indispensable. GIS tools provide a comprehensive suite of functionalities, enabling the input, storage, retrieval, manipulation, analysis, and output of spatial data (Wieczorek; Delmerico, 2009). When seamlessly integrated with computational optimization techniques, this set of GIS tools becomes an ideal platform for supporting decision-making activities in real-world scenarios. Notably, the contemporary days offer abundant geospatial information. For instance, platforms like OpenStreetMap[1] (OSM) provides access to vast crowd-sourced data, including detailed road networks and building information from around the world.

Besides the rise of new GIS technologies and demographic open data, the coverage of acute stroke treatment is still unknown in a large part of Brazil. In this work, we aim to assess the ASC network in Rio Grande do Sul (RS) state, Brazil, based on the population number that can travel to an ASC within 45 minutes and the average travel time that this population takes to reach the nearest ASC. Making this evaluation, we can verify whether RS follows the trend observed in low- and middle-income countries regarding acute stroke treatment. By mapping potential regional stroke centers, we recommend the placement of new ASCs based on a multi-objective FLP. Therefore, we add a cost objective to the evaluation to build a model that can offer decision-makers good solutions for different costs.

The current literature does not provide straight criteria to know which solutions are the most valuable in a Pareto front in this context. Domination-based MOEAs provide a framework that does not make assumptions about the Pareto front. Therefore, we will optimize the problem based on dominance, specifically, using the Nondominated Sorting Genetic Algorithm II (Deb et al., 2002) (NSGA-II), a classical approach that found success in many domains (VERMA; PANT; SNASEL, 2021). Then, we expose an evaluation of the efficacy of NSGA-II in this problem.

[1] https://www.openstreetmap.org (visited on 25/10/2023)

This work is organized into four chapters of background concepts: Chapter 2 presents the main concepts involved in single- and multi-objective optimization problems. Moreover it introduces metaheuristics and evolutionary algorithms, with focus on MOEAs; Chapter 3 gives a background of the FLPs foundations; finally, Chapter 4 presents some GIS tools that support this work. Chapter 5 and Chapter 6 are dedicated to the description of the methodology, including a summary of related works, and results. Lastly, Chapter 7 presents the conclusion and future works.

This study intends to shed light on the current accessibility for acute stroke treatment in RS and propose a possible improvement path. As observed by McNamara et al. (2020) and Ahmadi-Javid, Seyedi and Syam (2017), GIS is sparsely considered in healthcare facility locations. Therefore, it contributes to the useful direction of integrating GIS and FLPs to improve the catchment of time-sensitive emergency scenarios. Finally, we also evaluate the performance of NSGA-II in this application in terms of the quality of the obtained solutions. Our findings indicate that NSGA-II is a suitable alternative to optimize this model.

2 METAHEURISTICS

Many relevant optimization problems in science and industry have no exact methods that guarantee the finding of optimal solutions. This practical challenge prevents the application of exact methods for various problems. As opposed to exact methods, *metaheuristics* offer a powerful alternative capable of handling large-scale problem cases by providing satisfactory solutions in a reasonable amount of time. Metaheuristics encompass a diverse set of generic optimization algorithms that rely on stochastic operators. Due to their efficacy and flexibility, metaheuristics have gained significant popularity to tackle a broad spectrum of distinct and challenging problems.

This chapter is organized as follows. Section 2.1 and 2.2 present fundamental concepts of single- and multi-objective problems, respectively. These concepts include comparing solutions based on dominance, the notion of the Pareto front, and quality indicators for evaluating Pareto solutions. Section 2.3 delves into the fundamentals of metaheuristics, explaining the general attributes of these algorithms. Section 2.4 focuses on the Evolutionary Algorithms (EAs), introducing this framework's fundamental ideas and operators. Finally, Section 2.5 is dedicated to MOEAs, focusing on the renowned NSGA-II algorithm.

2.1 Optimization Problems

An optimization problem P can be defined by the triple $P = (S, f, \Omega)$, where S represents the problem search space, the set of all possible solutions defined by the decision variables $\{x_1, x_2, ..., x_n\}$. f stands for the *objective function* to be optimized. This function assigns a qualitative value to each solution $s \in S$, represented by a real number. f allows the comparison of any pair of solutions in the search space. Finally, Ω describes the set of constraints of the optimization problem (Talbi, 2009).

Optimization problems are divided into *continuous* and *combinatorial* problems, depending on whether the decision variables are continuous or discrete. The solution in a continuous optimization problem is generally a set of real numbers. In combinatorial optimization, we look for an object such as an integer, a permutation, or a graph in a countable set of solutions (Papadimitriou; Steiglitz, 1982).

The search space on combinatorial problems are not derivable. There is no such method as gradients or derivatives as we have in continuous problems to support the op-

timization. Therefore, combinatorial optimization is generally more challenging to solve than continuous problems (Peres; Castelli, 2021). The relevant combinatorial optimization problems are often NP-hard or NP-complete, having no deterministic method capable of solving them in polynomial time. The job scheduling problems, where tasks are allocated to resources in an optimal manner (Li et al., 2022), the FLPs, which aims to strategically place facilities to minimize costs (Ahmadi-Javid; Seyedi; Syam, 2017), and the vehicle routing problems, which aims to find the best set of routes that multiple vehicles must traverse to attend a set of locations (Konstantakopoulos; Gayialis; Kechagias, 2022), are just a few instances that highlight relevant real-world applications of combinatorial optimization.

Some real-world scenarios demand the optimization of more than a single problem simultaneously. Multi-objective optimization arises in this context. In such cases, the goal is not to identify a single best solution but to find a set of solutions representing a trade-off between the conflicting objectives.

2.2 Multi-objective Optimization Problems

There are diverse real-world situations that can be seen as optimization problems. For instance, suppose a person wants to buy a new mobile phone. This person is interested in the mobile phone that has the most extended battery life, but at the same time, it must be a cheap phone. Moreover, that person wants this device to be as small as possible. In this scenario, the person faces a classic example of a multi-objective optimization problem. Multi-objective optimization arises when multiple conflicting objectives or criteria must be considered simultaneously in decision-making. In the case of purchasing a mobile phone, this individual must balance three competing objectives: maximizing battery life, minimizing cost, and minimizing the phone's size.

This section intends to define multi-objective optimization problems and introduce multi-objective FLPs. The first section presents the general notation and definitions for these problems. Next, we define *Pareto Front*, the optimal set of solutions in a multi-objective space. Finally, it is presented a description on how FLPs fit in the multi-objective optimization context.

From Deb (2011), the general form of a multi-objective problem is:

$$\left.\begin{array}{lll} \text{Minimize/Maximize} & f_m(\mathbf{x}), & m = 1, 2, \ldots, M; \\ \text{subject to} & g_j(\mathbf{x}) \geq 0, & j = 1, 2, \ldots, J; \\ & h_k(\mathbf{x}) = 0, & k = 1, 2, \ldots, K; \\ & x_i^{(L)} \leq x_i \leq x_i^{(U)}, & i = 1, 2, \ldots, n. \end{array}\right\} \tag{2.1}$$

A solution $\mathbf{x} \in \mathbf{R}^n$ is a vector of n decision variables $\{x_1, x_2, ..., x_n\}^T$. g_j and h_k denote the inequality and equality constraints of the problem. The solutions that satisfy the variable bounds and both g_j and h_k constraints constitute the *feasible decision variable space* $S \in \mathbf{R}^n$. The functions $f_m : \mathbf{R}^n \rightarrow \mathbf{R}$ to be optimized are called *objective functions*. In multi-objective problems, there are two or more objective functions $(M \geq 2)$ that constitute a multi-dimensional *objective space* $Z \in \mathbf{R}^M$. Therefore, every feasible solution will map to an objective vector $\mathbf{f}(\mathbf{x}) = \{f_1(\mathbf{x}), f_2(\mathbf{x}), ..., f_M(\mathbf{x})\}^T$.

After placing the general definitions for a multi-objective optimization problem, we can proceed with how to compare the different vectors in the objective space.

2.2.1 Pareto Optimality

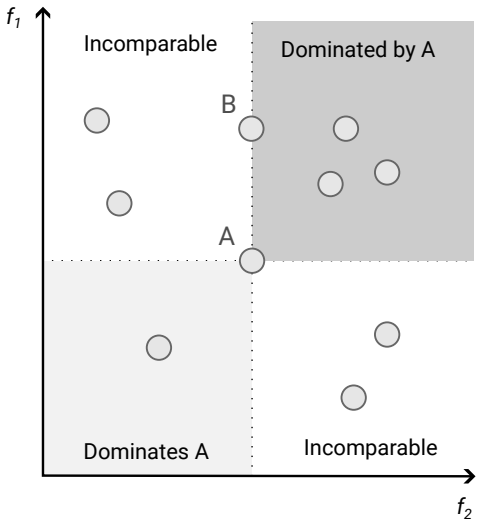
Due to the multiple conflicting objective functions, it is often impossible to find a single solution in S that is optimal for all M objectives simultaneously. Also, there is no *total ordering* in a multi-dimensional objective space, preventing a straightforward ranking of the set of solutions (Miettinen, 1998). For instance, although we can compare $\mathbf{f}(\mathbf{x}^{(1)}) = (1, 1)^T$ and $\mathbf{f}(\mathbf{x}^{(2)}) = (2, 2)^T$, it is not possible to evaluate if $\mathbf{f}(\mathbf{x}^{(3)}) = (2, 4)^T$ is better than $\mathbf{f}(\mathbf{x}^{(4)}) = (4, 2)^T$ because each solution is the best one in distinct objectives.

Hence, Z is said to be *partially ordered* since some elements in the set can be ordered, as illustrated by $\mathbf{f}(\mathbf{x}^{(1)})$ and $\mathbf{f}(\mathbf{x}^{(2)})$ in the example above. In multi-objective optimization, the term *domination* describes the ordering of the solutions. A solution $\mathbf{x}^{(1)}$ is said to dominate other solution $\mathbf{x}^{(2)}$, if both the following conditions are true (Deb, 2011):

1. The solution $\mathbf{x}^{(1)}$ is no worse than $\mathbf{x}^{(2)}$ in all objectives. Thus, the solutions are compared based on their objective function values.
2. The solution $\mathbf{x}^{(1)}$ is strictly better than $\mathbf{x}^{(2)}$ in at least one objective.

Figure 2.1 presents a set of solutions sampled over a two-dimensional objective

Figure 2.1: Domination relationships of a solution A.



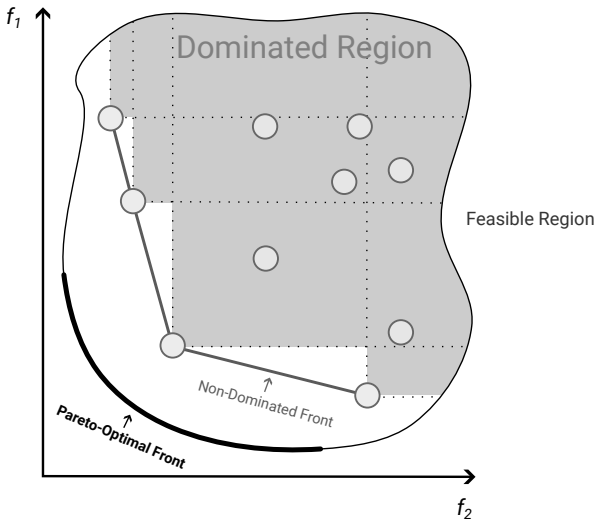
Source: Elaborated by the author. 2023.

space composed of f_1 and f_2. Both objectives must be minimized. The upper right quadrant[1] of A is the region where both f_1 and f_2 are more significant than the A's values. Therefore, all solutions that fall within this quadrant are dominated by A. Likewise, the solutions within the lower left quadrant dominate A. The solutions that are neither in the lower left nor upper right quadrants can not be compared to A since they are better than A in one of the objectives and worse in the other. Finally, besides B being not precisely within the region dominated by A, it is dominated by A because A is no worse than B concerning f_2 (it is equal), and it is strictly better than B in the objective $f1$.

The set of solutions not dominated by any other solution in the sample is called *non-dominated solutions*. If this set is non-dominated considering the whole search space S, the set is also *Pareto-Optimal*. Figure 2.2 illustrates these concepts. There is a feasible objective region composed of all solutions in S. The circles represent a sample of solutions in this space. Note that four different solutions form a non-dominated set. The upper right quadrants of these solutions create a *dominated region*, where the non-dominated front dominates all solutions within. In addition, the Pareto-Optimal Front is represented by the bold curved line in the Figure. It is the optima for this feasible region. Multi-Objective optimization algorithms, especially the Evolutionary Multi-Objective (EMO) class, pursue the Pareto-Optimal Front by trying to sample non-dominated solutions as close as possible to that front.

[1]The term quadrant fits when we have a two-dimensional objective space. However, we can call it *hypercube*, as the term generalizes for any number of dimensions.

Figure 2.2: The Pareto-Optimal Front and the Non-Dominated Front in the feasible objective space.



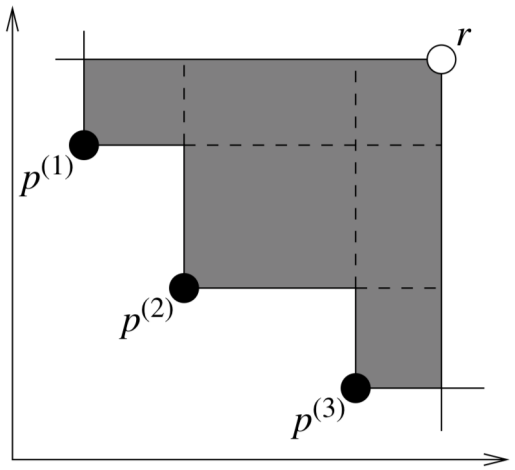
Source: Elaborated by the author. 2023.

Finally, it is noteworthy that there are different approaches to compare solutions in multi-objective optimization other than dominance. The classical methods are Objective Weighting, Distance Functions, and Min-Max Formulation (Srinivas; Deb, 1994). In summary, all these listed methods combine multiple objectives into one, which can be a profound drawback when decision-makers seek to analyze different alternatives before committing to a single solution. Nevertheless, it is essential to recognize that finding a non-dominated front does not guarantee that decision-makers have a high-quality set of solutions. Thus, distinct quality indicators of the obtained solutions are proposed in the literature.

2.2.2 Quality Indicators

There is no consensus among researchers on a single method for assessing the quality of a non-dominated set or selecting a unique solution within a Pareto front. A widely employed approach, particularly in cases with two or three objectives (as indicated by (Li; Chen; Yao, 2022)), involves visualizing the solution set. However, it is important to recognize that visual methods may not fully capture the aspects and differences among the solutions, particularly in problems with more than three objectives. Quality indicators have emerged as a valuable alternative to overcome the limitations of visual approaches. These indicators offer quantitative methods for comparing sets of solutions, overcoming

Figure 2.3: Hypervolume of three solutions in a two objectives minimization problem



Source: Extracted from Fonseca, Paquete and Lopez-Ibanez (2006)

the challenges associated with visualization methods (Li; Yao, 2019).

A quality indicator is a metric that maps a set of solutions (i.e., solution vectors) to a real number that represents one or more aspects of the quality of the solution set (Li; Chen; Yao, 2022). Examples of these aspects include the set's proximity to the Pareto-optimal front and the distribution of solutions within it.

The quality indicators can capture the *convergence*, *spread*, *uniformity*, and *cardinality* of a set of solutions. Convergence refers to the proximity of a set to the Pareto front; spread quantifies the region covered by the set; uniformity assesses the even distribution of solutions in the objective space; cardinality measures the number of non-dominated solutions within the set. The literature encompasses dozens of different quality indicators. The three most popular metrics, according to Li, Chen and Yao (2022), are described as follows:

- *Hypervolume*: Hypervolume (HV) is a prominent quality indicator widely recognized in the literature. It quantifies the hypervolume of the objective space enclosed by a set of solutions and a reference point. The HV of a set can be visualized as the union of hypercubes dominated by each solution within the set. For a visual representation of this concept, refer to Figure 2.3, which illustrates the hypervolume of three solutions in a two-objective space.

  The HV indicator for a set of solutions A and a reference point r can be defined as

$$HV(A) = \lambda \left(\bigcup_{a \in A} \{x | a \prec x \prec r\} \right) \tag{2.2}$$

where λ denotes the Lebesgue measure[2]. HV is biased to convex regions over concave regions since it prefers the *knee-regions*. The higher the HV, the better.

- *Generational Distance*: Generational Distance (GD) measures the quadratic mean of the Euclidean distance between each solution of a given set and the nearest point on the Pareto-optimal front. The GD for a solution set A is calculated by

$$GD(A, P) = \frac{1}{|A|} \left(\sum_{i=1}^{|A|} d\left(a_i, P\right) \right)^{\frac{1}{2}} \tag{2.3}$$

  where $d(a_i, P)$ is the Euclidean distance of a_i to its closest point in the Pareto front P. A smaller generational distance indicates that the solutions are closer to the Pareto-optimal front, signifying higher quality. However, this metric requires knowledge of an explicit target set, ideally the Pareto-optimal set, which can be a drawback in practice.

- *Inverted Generational Distance*: The Inverted Generational Distance (IGD) metric quantifies a similar aspect to GD but in an inverted manner. It measures the distance from points on the Pareto-optimal Front to their nearest points in a target set of solutions. Given a Pareto-optimal set denoted as P and a set of solutions A, the IGD is calculated as

$$IGD(P, A) = \frac{1}{|P|} \left(\sum_{i=1}^{|P|} d\left(P_i, A\right) \right)^{\frac{1}{2}} \tag{2.4}$$

  Lower IGD values indicate a closer approximation. Like GD, IGD also requires a target set of solutions for comparison.

Having introduced optimization problems in both single- and multiple-objective contexts and discussed methods for comparing solutions in a multi-objective space, we can now delve into a set of techniques known as metaheuristics, which can effectively address these complex optimization challenges.

[2]It is also called n-dimensional volume, n-volume, hypervolume, or simply volume.

2.3 Metaheuristic methods

Metaheuristics have emerged as a powerful and versatile approach for tackling complex optimization problems. These algorithms are distinguished by their capacity to find near-optimal solutions when exact methods are computationally infeasible. Due to their flexibility, metaheuristics are used in various fields, including computer science, engineering, bioinformatics, and operations research (Hussain et al., 2019). This section is reserved to present the main concepts of metaheuristics and how they relate to single- and multi-objective optimization problems.

Exact algorithms guarantee that a problem's optimal solution will be found in a finite amount of time. Heuristic methods use characteristics of a problem to guide a search through the problem's domain, hoping to find an optimal solution. However, unlike exact methods, heuristic algorithms do not have any guarantee of the optimality of their findings, returning near-optimal (or worse than optimal) solutions. Metaheuristics appears as one level higher in abstraction than heuristics -as the prefix *meta* suggests. That is, metaheuristics are not tied to specific problems but offer a general approach that can be adapted to various domains and scenarios. Therefore, we can define metaheuristics as a set of high-level procedures that provide guidelines or strategies for developing heuristic-based optimization algorithms (Sörensen, 2013).

Metaheuristics can be very general or high-level because they make weak or no assumptions about the problem to be optimized (Luke, 2013). These algorithms are designed on top of two main components: the representation of solutions and an objective function.

Representation of Solutions

The solutions' representation (or encoding) plays a major role in the efficiency and effectiveness of a metaheuristic. The representation must encompass all possible solutions for a given problem. Furthermore, a search path must connect any two solutions the encoding represents. The classical representations are listed as follows:

- *Binary*: In binary encoding, solutions are represented as binary digits (0s and 1s) strings. Each digit may correspond to a decision variable or solution component. This encoding is particularly useful for problems with discrete variables or deci-

sions, such as combinatorial optimization problems. For instance, the $0/1$ *knapsack problem* can be represented using the binary encoding (Tardos, 2005).

- *Vector of discrete values*: This representation uses vectors where each element corresponds to a decision variable that can take on discrete values. An example could be a scheduling problem where each vector element represents a time slot for a specific task.
- *Vector of real values*: In this representation, solutions are encoded as vectors of real numbers. It is useful for continuous optimization problems where the decision variables can take any real value within a specific range. It is a straightforward representation of any continuous optimization problem.
- *Permutation*: Permutation representation is employed for problems involving sequences or orders, such as the traveling salesman problem. Solutions are represented as permutations of elements, indicating the order in which they should be visited. Each position in the permutation corresponds to a specific location or task.

Objective Functions

The *objective function* (also *evaluation function* or *fitness function*) f maps any solution in the search space to a real value that describes the quality of a solution. f receives an encoded solution. Therefore, it might be necessary for the fitness function to first decode the solution before conducting the evaluation (Talbi, 2009).

The only way to compare two solutions in the context of metaheuristics is through their values on f. Therefore, metaheuristics conduct the exploration towards the optima or near-optima following the solutions that maximize or minimize the value yielded by f.

Single-solution and Population-based metaheuristics

Metaheuristics can be broadly categorized into single-solution (trajectory) and population-based algorithms (Hussain et al., 2019). These classes differ in their fundamental approach to exploring the solution space.

The single-solution metaheuristics are said to be exploitative. It refines a single solution iteratively by exploring its neighborhood in search of better solutions. The selection of a neighborhood function plays a crucial role in this class of metaheuristics.

This function will be responsible for mapping the possible moves that a current solution can take. An inadequate neighborhood function will make the method fail to solve the problem.

An example of a single-solution metaheuristic is the Simulated Annealing (SA) algorithm. SA starts with an initial solution and makes minor changes to the solution, gradually decreasing the magnitude of changes over time to find a global optimum. The quality of the solutions obtained by the moves guides the changes in the solution. Hill Climbing (HC) and Tabu Search (TS) are other prominent single-solution metaheuristics (Talbi, 2009).

Population-based metaheuristics maintain diverse solutions (population) and iteratively evolve this population over generations, converging the solutions to a specific region in the objective space. Therefore, a crucial aspect of the success of population-based metaheuristics is maintaining population diversity during the optimization, especially in early iterations. Population diversity is directly related to the convergence of the solutions (Del Ser et al., 2019). A failed mechanism to preserve diversity might result in the premature convergence of the solutions, which will prevent the algorithm from escaping a local-optimum region.

A prominent example of a population-based approach is the Evolutionary Algorithms (EAs). An EA starts by sampling an initial population. This population undergoes successive generations of mating and mutations, mimicking Darwin's theory of *natural selection*. Through this process, the best solutions tend to be preserved and improved over time. Particle Swarm Optimization (PSO), and Ant Colony Optimization (ACO) are other noteworthy population-based algorithms (Talbi, 2009).

The following section introduces one of the most prominent class of algorithms in metaheuristics: the Evolutionary Algorithms.

2.4 Evolutionary Algorithms

EAs are the class of metaheuristics that use the principle of natural selection for converging solutions to an optimal point. Since the 1980s, different EAs have emerged. The most prominent are Genetic Algorithms (GA), Evolution Strategies (ES), and Genetic Programming (GP). The common exclusive concepts that tie together all EAs are the following designing components (Talbi, 2009):

- *Selection Strategy*: This component is responsible for selecting solutions for mating and, consequently, selecting the information that will be passed into the generations. Selection strategies are usually composed of a heuristic that prefers to choose the best solutions among the population, but not always. In mechanisms such as the Tournament and the Roulette Selection (Luke, 2013), the algorithm selects parents randomly, but the solutions with more quality have more chances to be chosen for mating. In a different direction, the Biased-Random Key Genetic Algorithm (BRKGA) selection strategy always makes a mate by one elite individual, drawn from a pool of the best individuals in the population, and one non-elite individual, taken from a pool of the rest of the population (Resende, 2011).
- *Reproduction Strategy*: After selecting solutions for mating, the algorithm implements a reproduction strategy to recombine their *genetic code*. The *crossover* and *mutation* operators constitute the reproduction phase. The reproduction is a crucial component because it is through crossover and mutation operations that the algorithm traverses the solution space. Crossover performs a recombination of two solutions, mixing their genetic code (e.g., a recombination y of the vectors $x_1 = \{0, 1, 0\}$ and $x_2 = \{1, 0, 0\}$ could be $y = \{1, 1, 0\}$ or $y = \{0, 0, 0\}$, but never it would be $y = \{1, 1, 1\}$ because the third bit of both vectors is equal zero. The mutation operator is responsible for random perturbations in a solution. An example of a mutation operator is a random-bit flip with probability p. Taking the previous example, the algorithm could generate a vector $y = \{1, 1, 1\}$ from x_1 or x_2 if it randomly flipped the two zeros of the solutions.
- *Replacement Strategy*: The EA generates multiple new solutions, an *offspring*, during the reproduction. Therefore, the algorithm must select individuals from a pool composed of the current population and the offspring. An EA's replacement strategy will dictate which solutions will remain in the population from this pool.

Since its foundation, the EA framework has proved suited for diverse problems. However, when handling problems with multiple-conflicting objectives, we must extend EAs to a new particular class: the MOEAs.

2.5 Multi-Objective Evolutionary Algorithms

MOEAs represent a significant advancement in the field of evolutionary computation. They rely on specialized mechanisms for comparing solutions based on multiple criteria and maintaining them diverse regarding the multiple objectives, aiming to approximate an optimal Pareto Front properly. MOEAs can be divided into three categories: Domination-based, Indicator-based, and Decomposition-based (Liang et al., 2023; Trivedi et al., 2017):

- *Domination-based*: These algorithms operate on the principle of Pareto dominance, aiming to identify solutions that are not dominated by any other solution in the population. Classic examples include the Non-Dominated Sorting Genetic Algorithm (NSGA-II) (Deb et al., 2002) and the Strength Pareto Evolutionary Algorithm (SPEA2) (Zitzler, Eckart; Laumanns, Marco; Thiele, Lothar, 2001). These methods emphasize the preservation of diversity through elitism and nondominated sorting mechanisms.
- *Indicator-based*: In this category, algorithms evaluate the quality of solutions based on performance metrics or indicators. The goal is to approximate the Pareto front using limited indicators. Popular indicators include Hypervolume, Inverted Generational Distance, and Spread. IBEA (Zitzler; Künzli, 2004) is a well-known example that employs indicator-based approaches.
- *Decomposition-based*: Decomposition methods transform a multi-objective optimization problem into multiple single-objective subproblems, which are solved simultaneously. Multi-Objective Evolutionary Algorithm Based on Decomposition (MOEA/D) (Zhang; LI, 2007) fall within this category.

This work focuses on the NSGA-II algorithm, which belongs to the domination-based category. The following subsection describes this algorithm.

2.5.1 Nondominated Sorting Genetic Algorithm II

NSGA-II is among the most popular MOEA. It was published in 2002 in the paper "*A Fast and Elitist Multi-objective Genetic Algorithm NSGA-II*" (Deb et al., 2002). The primary motivation for creating the NSGA-II was to improve the inefficiencies of the prior NSGA algorithm. NSGA-II improved the computational complexity of the nondominated

sorting from $O(MN^3)$ to $O(MN^2)$ (where M is the number of objectives and N is the population size). Moreover, it introduced elitism to the NSGA and a prosperous new parameter-less strategy to preserve diversity in the population: the *crowding distance* calculation. These improvements were fundamental to making NSGA-II one of the most influential algorithms in evolutionary computation. Today, the original paper has over twenty thousand citations.

Fast Nondominated Sorting

NSGA-II sorts the population based on *nondomination ranks*. The solutions with the same rank are assigned to a *front*. The first front is the set F_1 of nondominated solutions of the population. We can find the first front by comparing all population solutions pairwise. Algorithm 1 describes the procedures to find the first front. S_p stores the solutions that a solution p dominates. n_p counts the number of solutions that dominate p. The procedure compares all solutions in the population. When a solution p dominates a solution q, the algorithm adds q to the set S_p. Whenever a solution q dominates p, n_p increases by one. Ultimately, all solutions in which n_p counts zero become part of the first front. This step has a computational complexity of $O(MN^2)$.

Algorithm 1: Computing First Front

```
for p ∈ P do
    S_p ← ∅
    n_p ← 0
    for q ∈ P do
        if p ≺ q then
            S_p ← S_p ∪ {q}
        end
        else if q ≺ p then
            n_p ← n_p + 1
        end
    end
    if n_p = 0 then
        p_rank ← 1
        F_1 ← F_1 ∪ {p}
    end
end
```

The second front is the set F_2 of nondominated solutions *when we remove the solutions of the first front* from the population. The third front is the set of solutions when we remove the individuals that are both within F_1 and F_2 from the population, and so

on. In a naive approach, the algorithm would require running Algorithm 1 without the solutions of the i^{th} front every time it needs to find the next front. Thus, the worst case would require $O(MN^3)$ computations (when there are N fronts with a single solution in each).

NSGA-II developed a new strategy to reduce the complexity of the nondominated sorting to $O(MN^2)$. The new algorithm keeps tracking, for each solution p, the number of solutions that dominate p and the set of solutions that p dominates. After computing the first front, the algorithm runs a procedure that computes all fronts in $O(MN^2)$. Algorithm 2 presents this procedure. The idea of the algorithm is to remove iteratively the solutions of the i^{th} non-dominated front of the sample. It starts removing the solutions from the front F_1. Therefore, all solutions dominated by F_1's solutions will have their n_q counter decreased by one (Line 6). If n_q becomes zero, q belongs to the next non-dominated front.

Algorithm 2: Computing All Fronts

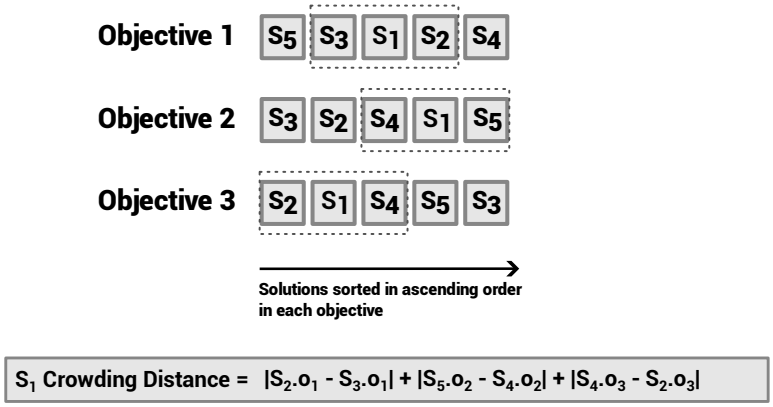
```
   /* S_p and n_q are set in the Algorithm 1                    */
 1 i ← 1
 2 while F_i ≠ ∅ do
 3 |   Q ← ∅ // Q stores the solutions of the next non-dominated front
 4 |   for p ∈ F_i do
 5 |   |   for q ∈ S_p do
 6 |   |   |   n_q ← n_q − 1
 7 |   |   |   if n_q = 0 then
 8 |   |   |   |   q_rank ← i + 1 // q belongs to the next non-dominated front
 9 |   |   |   |   Q ← Q ∪ {q}
10 |   |   |   end
11 |   |   end
12 |   end
13 |   i ← i + 1
14 |   F_i ← Q
15 end
```

Crowding Distance

Often, it is not helpful for a decision-maker to choose among slightly different solutions. Therefore, it is desirable for a MOEA that the solutions converge to Pareto-front while maintaining the set of solutions spread regarding the objective values. Hence, NSGA-II incorporates a property known as crowding distance to the individuals.

Crowding distance serves as a diversity-preserving mechanism within NSGA-II, enabling the algorithm to effectively manage the distribution of solutions across the Pareto

Figure 2.4: An example of the computation of the crowding distance of a solution S_1. The five solutions are sorted in each objective. The left and right neighbors of S_1 are taken for computing S_1 crowding distance.



Source: Elaborated by the author. 2023.

front. The fundamental idea is to evaluate the density of solutions surrounding each candidate solution. Solutions with higher crowding distances are favored, as they lie in sparser front regions (less dense regions), thereby preserving diverse and non-redundant solutions.

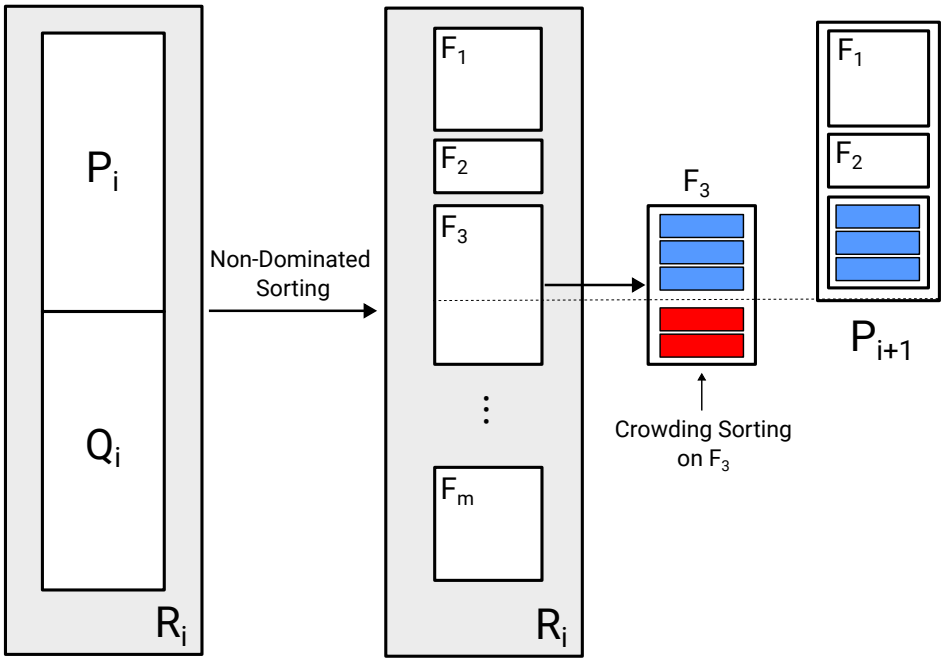
The distance of the solutions is accounted for in the objective space. Assuming m objectives, the crowding distance value of a solution i is the *Manhattan Distance* of the nearest neighbors $i + 1$ and $i - 1$ in each objective. Therefore, the crowding distance calculation requires sorting the population according to each objective function value in ascending order. NSGA-II assigns an infinite distance to the solutions with the smallest and largest function values to keep the extreme points for increasing diversity.

Figure 2.4 presents an example of the crowding distance calculation for a solution S_1. The example assumes that the objective values are already normalized. The solutions are sorted in ascending order for each objective. The nearest neighbors of S_1 are S_3 and S_2 considering the Objective 1; S_4 and S_5 for Objective 2; S_2 and S_4 for Objective 3. The distances of the neighbors are summed up as the equation in Figure 2.4. $S_i.o_j$ represents the value of the objective j from solution i.

Execution Flow

NSGA-II execution flow relies on selection, reproduction, and replacement as any EA. The initial generation G_0 is slightly different from the subsequent generations. It starts from a sampled initial population P_0 of size N. The first step on G_0 is sorting P_0 by

Figure 2.5: Execution flow in a generation G_{i+1}.



Source: Elaborated by the author. 2023.

non-dominance. From there, NSGA-II performs a tournament selection over P_0 to select parents for generating an offspring population Q_0 of size N.

All next generations perform the same operations. Let us look into a generation G_{i+1}. G_{i+1} starts by creating a pool of solutions $R_i = P_i \cup Q_i$ of size $2N$. Then, the algorithm computes all fronts $F_1, F_2, ..., F_m$ in R_i using the non-dominated sorting. Beginning with the first front, i.e., F_1, NSGA-II systematically selects solutions from each front such that the sum of their cardinalities fits within the population size limit N. If adding a new front exceeds the population size, the algorithm sorts this particular front using the crowding distance metric. It retains the best solutions that can be accommodated within the population.

Figure 2.5 illustrates this workflow. Adding the front F_3 in the new population would exceed the population size. Then, the algorithm sorted F_3 by crowding distance. Only the three best solutions are included in the new population P_{i+1}, while the remaining two are discarded. NSGA-II proceeds from generation to generation until a termination criterion is met. Termination criteria can include a fixed number of generations or a convergence threshold based on the problem requirements.

2.6 Final Remarks

In this chapter, we discussed the applications and the fundamentals of metaheuristics. We began by defining the essence of optimization problems, distinguishing between single and multi-objective variants, and introducing the concept of Pareto-optimality and the quality indicators for evaluating the proximity to the Pareto front in multi-objective problems. Then, the chapter reviewed basic concepts that glue all metaheuristics together, focusing on the EAs. The use of EAs for multi-objective optimization is very diffused in the literature, arguably one of the most used algorithms employed on such problems. A prominent example is the NSGA-II, which was deeply explained in this chapter based on its original publication.

As we conclude our exploration of metaheuristics and the foundational understanding of optimization problems, we can focus on a specific application area where these methodologies proved their effectiveness: Location-Allocation Problems (LAPs). LAPs are fundamental problems that often appear in Operations Research (OR) applications. The following chapter will cover classical formulations of LAPs, giving a historical view and formal definitions. Moreover, we explore different applications of LAPs as single- and multiple-objective problems.

3 FACILITY LOCATION PROBLEMS

FLPs can be formulated as optimization problems that must determine the optimal location of facilities, such as warehouses, hospitals, and communication centers (Galvão, 2004). As an active research area, the study on FLPs dates from the 60s. Since then, it has developed theoretical and practical applications in different fields, such as Economics, Geography, and Logistics. In today's data-rich environment, FLPs offer even more suitability for tackling real-world problems, making them a crucial area of interest in contemporary research and industry.

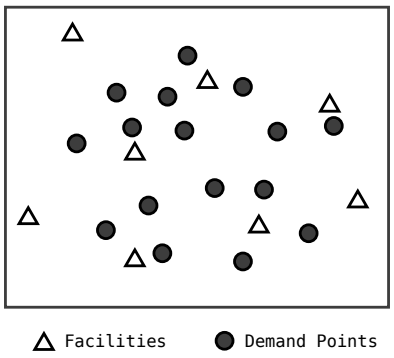
This chapter is structured as follows: Section 3.1 provides an introduction to FLPs, offering a historical overview of the field and introducing fundamental problems within this area. Section 3.2 defines and reviews individual location problems addressed in this work. Section 3.3 gives an overview of multi-objective FLPs through a review of relevant literature.

3.1 Fundamentals on FLP

Some specific geometry problems can be considered early precursors of FLPs, such as minimizing the sum of the distances in Euclidean space from one point to three additional points. However, the formalization of FLPs as a distinct scientific discipline traces its origins back to the 1960s. A notable early work in this field is the paper "Location-Allocation Problems" by Cooper (1963), which introduced one of the earliest formulations of FLPs as we understand them today. In Cooper's work, the problem aims to minimize the Euclidean distance of supply and demand points over a cartesian coordinate system. This formulation soon spread to weighted graphs when Hakimi (1965) introduced the p-*center* and the p-*median* location problems.

The p-*median* (PMLP) and the p-*center* (PCLP) Location Problems prioritize optimizing the distances between facilities and demand points. Hakimi initially proposed both problems under the context of distributing switching centers in telecommunications and police officers on highways. They differ slightly: the p-*median* seeks to identify p locations that minimize the average distance between supply and demand points, whereas the p-*center* is concerned with finding the p locations that minimize the maximum distance between supply and demand points. Both problems are location-allocation problems (LAPs) since, besides choosing facilities, the problem must allocate demand points

Figure 3.1: Instance of a facility location problem.



Source: Elaborated by the author. 2023.

to these facilities (Ahmadi-Javid; Seyedi; Syam, 2017).

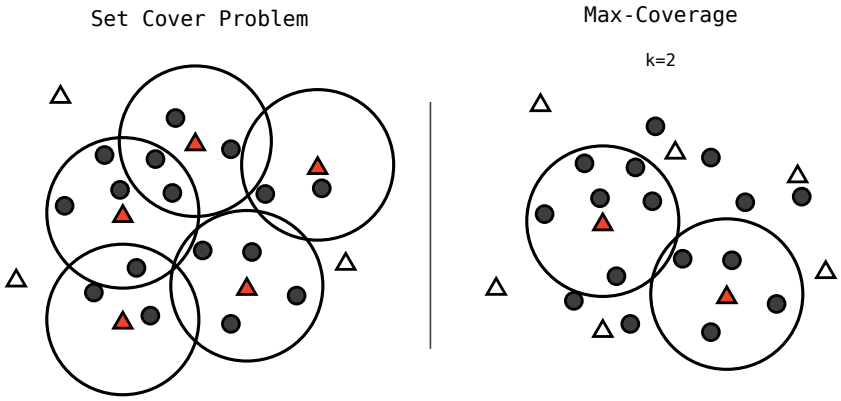
FLPs encompass a broader category than LAPs. Certain FLP models focus exclusively on covering demand points. These are known as Covering-Location Problems (CLPs). In CLPs, the core idea is that demand points must be within a certain distance of the nearest facility for effective service. Demand points meeting this proximity criterion are considered ‘covered‘ (García; Marín, 2019).

For better visualization of covering models, we bring the Figure 3.1 that illustrates an FLP instance. The triangles represent the facilities that must be established to cover the corresponding demand points represented by circles. Various problem variants can arise from this. For example, suppose we model the problem as the Maximal Covering Location Problem (MCLP) (Church; ReVelle, 1974). In that case, we allocate a fixed number of k facilities to maximize the supply of demand points. In contrast, if the goal is to find the minimum number n of facilities to allocate to cover all demand points, we formulate the problem as the Set Covering Location Problem (SCLP) (Toregas et al., 1971).

SCLP and MCLP are problem-specific versions of the well-studied Set Cover and the Maximum Coverage problems. Figure 3.2 presents the different solutions obtained from the different models. While the Set Cover instance finds a solution covering all demand points, the Max-Coverage, with $k = 2$, finds the two facilities covering the maximum number of demand points.

SCLP has been generalized as the Partial Set Covering Problem (PSCLP) on (Daskin; Owen, 1999). The facilities do not need to cover all demand points in this formulation but an exogenously specified percentage of demand points. Note that if this

Figure 3.2: Difference of running an optimization over the instance of Figure 3.1 as a set cover and max-coverage problem.



Source: Elaborated by the author. 2023.

percentage is set to 100%, the PSCLP becomes the SCLP.

In summary, the literature encompasses many FLP variants, each addressing different aspects of facility location optimization. Constraints to which the FLPs are subjected, such as the capacity of the facilities, are also essential factors in classifying this problem. For instance, attributing capacity or not to the facilities results in two different branches of FLPs: Uncapacitated Facility Location Problems (UFLPs) and Capacitated Facility Location Problems (CFLPs). The former addresses the problem of finding optimal suppliers' locations so that the cost to attend to the demand points is minimal. On the other hand, the latter includes a capacity constraint in which the suppliers must not surpass (Verter, 2011).

A review of various FLPs can be found in (Laporte; Nickel; Saldanha da Gama, 2019). The following section presents details and the formalization of selected FLPs that compose this work.

3.2 Classical Problems

This section explores classical problems, including set covering, partial set covering, maximum coverage, and the p-median problem. These foundational formulations are crucial in FLPs and are commonly encountered in the existing literature, serving as fundamental models for more complex formulations.

3.2.1 Set Covering Problem

The set covering problem is a well-known NP-hard combinatorial optimization problem that involves selecting the minimum number of sets from a larger collection of sets to cover all elements in a given universal set. In this problem, a set U of n elements is given, along with a collection S of m subsets of U, $S = \{s_1, s_2, ..., s_m\}$. Each subset s_i has an associated cost c_i. The goal is to select the subsets whose union covers all elements in U with a minimum cost. Formally, the set covering problem can be stated as follows:

- Given a set U of n elements and a collection S of m subsets of U with associated costs c_i, find a subset $C \subseteq S$ such that

$$\bigcup_{s_i \in C} s_i = U, \tag{3.1}$$

while minimizing the total cost

$$\min_{C \subseteq S} \sum_{c_i \in C} c_i. \tag{3.2}$$

Note that if $c_i = 1$ for all i, the minimization of the cost turns into the minimization of the number of facilities allocated.

One of the drawbacks of this model is that it fails to distinguish the importance of different demand nodes. For instance, if the problem has a population number associated with the demand nodes, there is no difference in that model of serving a node n_1 with one million people and a node n_2 of ten people. *It just needs to serve **all** nodes*. Thus, the minimum solution is often very sensitive to the network. As stated in Daskin and Owen (1999), the small addition of nodes to the demand set, even nodes with negligible weights, can significantly increase the minimum required facilities.

Furthermore, the solutions obtained from the SCLP model may be impractical when applied to real-world scenarios. Therefore, Daskin and Owen (1999) proposed a generalization of SCLP to better reflect real-world constraints and requirements: the so called partial set covering location problem.

3.2.2 Partial Set Covering Problem

The partial set covering location problem is a generalization of the set covering problem. Rather than finding a combination of facilities that covers the whole set U, the combination must cover U partially. Therefore, in addition to the set covering formulation, PSCLP has an exogenous variable k that denotes how many elements of U must be covered. The problem has the same optimization objective as the set covering, but the cardinality of the subset C must satisfy $|C| \geq k$. PSCLP can be formulated as follows:

- Given a set U of n elements, a collection S of m subsets of U with associated costs c_i, and an integer k, find a subset $C \subseteq S$ such that

$$\left| \bigcup_{s_i \in C} s_i \right| \geq k, \tag{3.3}$$

  while minimizing the total cost

$$\min_{C \subseteq S} \sum_{c_i \in C} c_i. \tag{3.4}$$

  Note that the set covering is the special case when $k = |U|$.

PSCLP is more capable than the SCLP of generating realistic solutions, especially when demand points are associated with population data (Daskin; Owen, 1999). Surprisingly, despite its practical applicability, PSCLP has received relatively little attention in the literature (Cordeau; Furini; Ljubić, 2019). In contrast, a very prominent model is the MCLP. It takes a different approach by making the number of located facilities an exogenous variable and maximizing the covered demand instead.

3.2.3 Maximum Coverage Problem

The maximum coverage problem is also a classical NP-Hard problem addressed in the combinatorial optimization area. In contrast with the set covering, the maximum coverage has to choose k different subsets in S that their union contains the maximum number of elements of the universal set U. A formal definition is stated as follows:

- Given a set U of n elements, a collection S of m subsets of U, and an integer k,

find the subset $S' \subseteq S$, where $|S'| \leq k$, such that

$$\max_{S' \subseteq S} \left| \bigcup_{s_i \in S'} s_i \right| \tag{3.5}$$

The Maximum Coverage Model is one of the most applied models for locating healthcare centers (Ahmadi-Javid; Seyedi; Syam, 2017). This model is attractive when considering population counts in demand points since it can differentiate points with large and small demand.

We can find this model in a study case of locating stroke centers in the state of Iowa, USA, while maximizing the population accessibility to the stroke centers (Leira et al., 2012). A recent work applied MCLP to maximize the local government areas with access to COVID-19 testing facility sites in Nigeria (Taiwo, 2021). Furthermore, MCLP is one of the objectives in a multi-objective methodology for finding suitable points for locating healthcare units in rural areas (Mishra et al., 2019).

3.2.4 *p*-median Problem

The p-median problem involves identifying the set of p points $X_p = \{x_1, x_2, ..., x_p\}$ within a network that minimizes the cumulative distance to all nodes. A network is a connected undirected graph $G = \{V, E\}$. A point x_i is any point along any edge of G, and might be a vertex of G. The problem is NP-hard (Kariv; Hakimi, 1979). The set X_p is a p-median of G if it minimizes the sum:

$$\min_{X_p \subseteq G} \sum_{v \in V} w(v)\, d(v, X_p) \tag{3.6}$$

The term $w(v)$ is a non-negative number representing the weight of v, $v \in V$. The distance $d(v, X_p)$ is defined by:

$$d(v, X_p) = \min_{1 \leq i \leq p} \{d(v, x_i)\} \tag{3.7}$$

where $d(v, x_i)$ is the length of a shortest path in G between a vertex v and a point x_i. Hence, $d(v, X_p)$ states that the problem considers that v are associated with the point whose distance is minimum.

In p-median location problems, the set X_p is a candidate set of facilities. The

vertices V are the demand points. PMLP differs from MCLP and PSCLP because it focuses on the distances to the facilities instead of their coverage.

Diverse works can be found using the p-median formulation. McNamara et al. (2020) proposes a generic benchmark to evaluate the emergency healthcare network of a region. One of their criteria is the average distance to the hospitals based on the p-median formulation. In addition, Zarate-Zapata et al. (2023) presents a model based on the PMLP to locate five vaccine centers to minimize the distance traveled by the population.

The list of works employing FLPs is endless. For more examples and a more extensive view of these problems, please refer to (Laporte; Nickel; Saldanha da Gama, 2019) and (Ahmadi-Javid; Seyedi; Syam, 2017).

3.3 Multi-objective Facility Location Problems

Due to the complexity of real-world applications, decision-makers might opt for models that concern different aspects of a location problem. The objectives that are usually addressed in FLPs are diverse (Farahani; Seifi; Asgari, 2010):

- Minimizing the total setup cost;
- Minimizing the longest distance from the existing facilities;
- Minimizing fixed cost;
- Minimizing total annual operating cost;
- Maximizing service;
- Minimizing average time/ distance traveled;
- Minimizing the number of located facilities;
- Maximizing responsiveness.

These objectives frequently conflict with each other and pose challenges in terms of measurement and prioritization within a model. Therefore, many works bring multi-objective approaches based on non-dominance to FLPs (Farahani; Seifi; Asgari, 2010). Bi-objective formulations, for instance, are easily found in the literature. Works such as Shi et al. (2020) and Bai, Chin and Zhou (2019) minimize the cost of placing facilities while maximizing coverage/service. Karasakal and Silav (2016) adds to the MCLP the objective of minimizing the maximum distance of uncovered nodes to their nearest facilities. Wang and Ma (2018) allocates nursing homes concerning the allocation cost

minimization and the total weighted distance traveled from the demand points to the facilities.

Other works go further, employing more than two objectives. Beheshtifar and Alimoahmmadi (2014) employs a p-median problem to allocate healthcare centers concerning three more objectives: minimizing unequal access to clinics, minimizing cost, and maximizing land-use suitability. Karatas and Yakici (2018) optimizes three classical formulations simultaneously: PMLP, MCLP, and PCLP.

3.4 Final Remarks

We introduced in this chapter the foundations of FLPs with a historical perspective, giving an overview of their applications. Moreover, different classes of FLPs were introduced, such as location-allocation and covering location problems. The chapter presented the definitions of classical problems and how they appear in the literature. We could explore the differences among covering formulations, including PSCLP, SCLP, and MCLP. Additionally, we introduce the p-median problem, giving a distance-based alternative for a FLP formulation. Finally, the last section of this chapter briefly describes various works in which multiple objective FLPs appear.

Equipped with models that are crucial for addressing real-world logistics challenges, such as healthcare optimization, the next step involves translating real-world scenarios into computational models. Modeling FLPs to address real-world challenges demands the seamless integration of GIS. These systems offer indispensable tools for manipulating geospatial data and accommodating factors such as real building locations, road networks, latitude/longitude coordinates, and regional boundaries. Integrating GIS and FLPs presents a powerful combination to enhance any service that demands spatial accessibility.

4 GEOGRAPHIC INFORMATION COMPUTATION

Geographic Information Systems encompass a wide array of functionalities for geographic analysis. GIS comes in different flavors nowadays. Some software offers a complete set of tools for geographic computation, including interactive visualization. Some libraries implement only part of the GIS functions in programming languages like Python. Another example of GIS tools include online services that offer routing algorithms in road networks. Routing algorithms are particular useful when modeling FLPs since these problems inherently involve graph structures. When facilities and demand points are interconnected through road networks, a graph structure emerges with roads as edges and target points as nodes. GIS tools can compute the relationships between these FLP entities by, for instance, establishing routes and measuring distances between them. In essence, there is invariably a GIS tool at the core of any FLP implementation concerned with solving a real-world problem.

This chapter aims to explain the basic concepts behind GIS and introduce the main libraries that drives the proposed study forward. It is structured into three sections. Section 4.1 provides an introductory segment, motivating and exposing fundamental concepts that constitute GIS. In Section 4.2, we introduce OpenRouteService (ORS), an open-source platform used primarily to calculate distances within a road network. Section 4.3 explores GeoPandas, a Python library to support geospatial computation in Pandas.

4.1 An Introduction to GIS

GIS comprises a suite of software tools designed for the analysis and modeling of *geographic data* (Wieczorek; Delmerico, 2009). Geographic data are spatial and are referenced to specific locations on the Earth's surface (McLafferty; Cromley, 2012). Typically, geographic data are represented using three basic geometric components: *points*, *lines*, and *polygons*. The representation of these components in GIS relies on a geographic coordinate system. Therefore, the x, y, and z coordinates of an Euclidean space will be translated to *longitude*, *latitude*, and *elevation* coordinates, respectively.

The GIS' key capabilities include spatial database management, visualization and mapping, and spatial analysis (McLafferty; Cromley, 2012):

- *Spatial Database Management*: This capability involves storing and managing spa-

tial objects and their associated attributes. These spatial objects can possess various attributes like length, name, and timestamps. Besides the regular database operations, one of the key features of a spatial database is its ability to retrieve data based on the spatial relationships among objects, such as whether they touch or intersect each other. Within GIS, the process of *georeferencing* is fundamental. For example, when incorporating a vector image (e.g., a polygon representing the boundaries of a city) into a GIS database, its points must be converted to a specific *coordinate reference system*. This process is commonly referred to as georeferencing.

- *Visualization and Mapping*: GIS offers powerful tools for visualizing data, creating maps representing geographic units, and displaying spatial analysis outcomes. An illustrative example of this capability is visualizing the distribution of healthcare facilities or other regional services. The visualization and mapping capabilities in GIS enable users to grasp the spatial relationships embedded in their databases, making it easier to interpret and to report results of different types of spatial analyses.
- *Spatial Analysis*: Spatial analysis within GIS involves the capability to query and manipulate spatial data to unveil meaningful patterns and insights of the data (Paramasivam; Venkatramanan, 2019). According to McLafferty and Cromley (2012), spatial analysis falls into five classes: measurement, topological analysis, network analysis, surface analysis, and statistical analysis. *Measurement* functions include distance, length, area, and volume measurements among the considered entities. Another important measurement feature is the spatial buffer, which identifies the area limited to a distance threshold around a specific entity (Wieczorek; Delmerico, 2009). *Topological analysis* pertains to the spatial relationships between the entities analyzed, retaining information about adjacency, overlay, and direction information such as the orientation of the streets. *Network analysis* involves analyzing the topology of entities, treating relationships as nodes and links, much like a graph. In this context, GIS can execute various network algorithms, including connectivity assessments, shortest-path algorithms, and location-allocation models. *Surface Analysis* is commonly used to analyze terrain data, particularly in the context of elevation models and land surfaces. *Statistical analysis* involves employing statistical methods to explore and analyze spatial data, allowing for the identification of trends, patterns, and correlations within geographic information.

Software tools with GIS capabilities encompass complete desktop applications, programming libraries, GIS services, and even relational databases designed explicitly for

managing geographic data. Notable examples of desktop applications include ArcGIS[1] and QGIS[2]. ArcGIS is a proprietary GIS software, while QGIS is a free and open-source alternative to ArcGIS. Both of these applications provide user-friendly point-and-click interfaces for GIS tasks. They offer various capabilities, allowing for various analyses, including modeling FLPs such as the maximal covering problem.

Desktop GIS applications offer options for customization and extending functionality through plugins. They also provide the ability to export spatial analyses to files other applications can read. However, working across multiple environments can be less desirable, and running custom analyses in these software applications may not be straightforward. An alternative approach involves using programming libraries with GIS capabilities, such as GeoPandas[3].

GeoPandas is a Python library that combines the capabilities of Pandas with geospatial data processing capabilities. It enables users to work with geospatial datasets in a tabular format, acting essentially as a GIS database. Additionally, it offers spatial data manipulation tools like overlays, spatial buffers, and various measurement functions. However, integration with other Python libraries within the data science ecosystem is essential to unlock the full range of GIS capabilities. For example, we can combine GeoPandas with NetworkX[4], a Python library for graph algorithms, to perform network analysis on GeoPandas data.

Another versatile approach is to leverage GIS services. Various APIs offer a range of GIS functionalities, especially network analysis on road networks, and provide interfaces for popular programming languages like Python. Notable examples include the Google Maps API[5], the Bing Maps API[6], and the OpenRouteService[7].

In summary, there are several options available today for performing geographic computations. In many works that model multi-objective FLPs using GIS, the primary choices are ArcGIS or QGIS. A typical approach involves calculating and exporting the distance matrix of the facilities and demand points within the GIS software, often ArcGIS. This generated file can then be used as input for further analysis within a different environment, as observed in previous studies (McNamara et al., 2020; Beheshtifar; Alimoahmmadi, 2014; Akgün; Erdal, 2019). This approach requires familiarity with the

[1]https://doc.arcgis.com/pt-br/ (visited on 25/10/2023)
[2]https://www.qgis.org (visited on 25/10/2023)
[3]https://geopandas.org/en/stable/ (visited on 25/10/2023)
[4]https://networkx.org/ (visited on 25/10/2023)
[5]https://developers.google.com/maps (visited on 25/10/2023)
[6]https://www.microsoft.com/en-us/maps/bing-maps/ (visited on 25/10/2023)
[7]https://openrouteservice.org/ (visited on 25/10/2023)

ArcGIS environment, and it does not offer an integrated pipeline for running the network analysis and the optimization model.

In our work, we take a different approach by integrating services of the ORS API into the Python environment, where we can leverage the functionalities of GeoPandas for comprehensive GIS capabilities within the Python environment. ORS API comprises very useful services for computing distances and travel times in a road network. The following section introduces the ORS API and its fundamentals.

4.2 OpenRouteService

ORS is an open-source project offering various geospatial services through an API. The project started in 2008 (Neis; Zipf, 2008), and currently is maintained by the Heidelberg Institute for Geoinformation Technology[8]. Its primary focus is network analysis functions, including route planning, geocoding, isochrone calculations, and distance matrix computations. ORS computations are performed using the OpenStreetMap database, which is the most important collaborative project to create a free and editable world map.

4.2.1 OpenStreetMap

OpenStreetMap is the world's largest Volunteered Geographic Information (VGI) platform, having complete and reliable geographic data produced by millions of people around the world (Anderson; Sarkar; Palen, 2019). The rise of an open geographic database, such as OSM, has brought a wide range of initiatives to production. For instance, organizations such as the American Red Cross (ARC) and Médecins Sans Frontiéres (MSF) contribute to the Humanitarian OpenStreetMap Team (HOT) for mapping the most disaster-risk places in the world (Herfort et al., 2021). In addition, big companies, such as Facebook and Amazon, rely on OSM data[9].

The core of the OSM data model is the *element*, which represents a physical or logical object in the real world. There are three types of elements in OSM:

- ***Nodes*** represent a single point in space, identified by a unique node ID identifier.

[8] https://heigit.org/ (visited on 25/10/2023)
[9] https://welcome.openstreetmap.org/about-osm-community/consumers/ (visited on 25/10/2023)

Nodes represent things like street lamps, benches, and fire hydrants.

- ***Ways***: are a sequence of connected nodes that form a linear feature, identified by a unique way ID. Ways represent things like roads, rivers, and buildings.
- ***Relations***: represent a collection of elements with some common feature, identified by a unique relation ID. Relations represent complex structures like public transportation systems or buildings with multiple floors.

In addition, all data elements (nodes, ways, and relations) can have associated *tags* that provide additional information about them. Tags are key-value pairs that describe the attributes of an element, such as the name of a road or the type of building. For example, a way element might have tags that indicate its name, speed limit, and number of lanes, while a building element might have tags that indicate its use, number of floors, and construction materials.

Road networks are expressed by the `highway` tag. The highway tag categorizes any road, route, or path that has been paved to allow travel by some conveyance[10]. The highway values that denote driving routes are *motorway*, *trunk*, *primary*, *secondary*, *tertiary*, and *unclassified*. Each value corresponds to a specific type of road. In addition to these main values, several other values denote specific types of roads, such as *residential* for roads in residential areas, *service* for roads that provide access to specific locations, and *track* for unpaved roads that are suitable for motor vehicles.

OpenRouteService calculations over the road network utilize the OSM tags on estimating travel times effectively. For trip driving by car, the API considers characteristics like the types, max speeds and the surface of highways.

4.2.2 ORS Travel Time Estimation

ORS can compute travel times for driving, walking, cycling, and wheelchair profiles. The driving profile has a cascading assessment of the base speed of the OSM ways. Generally, it first checks if the way exposes a speed value in the `maxspeed` tag. If so, 90% of this value is considered the way's base speed. However, if the way does not indicate a speed limit, ORS assigns the base speed depending on the way's `highway` tag. Table 4.1 presents the default speed for each type of way.

The base speeds might be further modified depending on other factors. These

[10]https://wiki.openstreetmap.org/wiki/Highways (visited on 25/10/2023)

Table 4.1: Base Speed by Highway Tag

Highway Tag	Base Speed (km/h)
motorway	100
motorway_link	60
motorroad	90
trunk	85
trunk_link	60
primary	65
primary_link	50
secondary	60
secondary_link	50
tertiary	50
tertiary_link	40
unclassified	30
residential	30
living_street	10
service	20
road	20
track	15

factors include the assignment of a based speed depending on the surface type of the way (based on the `surface` tag), speed reduction when driving in residential ways and roundabouts. A more detailed description can be found on the documentation[11].

The ORS services hide these details, providing a easy-to-use interface to the user. The API runs in a HTTP server, being possible to create an instance of the server in a local host. The services can be requested via standard HTTP methods, including GET and POST. Users can choose from various profiles to consider specific modes of transportation, including walking, cycling, driving by car, and public transportation.

4.2.3 ORS Services

FLPs are fundamentally tied to graph theory, making network analysis tools essential for their resolution. When optimizing the placement of stroke centers in a region, it involves facilitating the movement of individuals to healthcare facilities within a specific graph structure: the road network of that region. While there may be various routes that a person can take to reach a given stroke center, there is always a optimal one. OpenRouteService plays a vital role in solving FLPs in real road networks by identifying the

[11] https://giscience.github.io/openrouteservice/documentation/travel-speeds/Travel-Speeds.html (visited on 25/10/2023)

single best route for demand nodes to access the facilities, establishing relevant connections between these entities. ORS includes different services for network analysis, such as *directions*, *isochrones*, and *matrix*[12]:

- *Directions*: The directions service computes a route that passes through two or more points in the map. Figure 4.1a gives an example of a route through four distinct coordinates.
- *Isochrones*: Isochrones represent lines on a map or diagram that connect locations from which it takes the same time to reach a specific point.[13]. For example, if a facility has a 10-minute isochrone, any location within that line can reach the facility within 10 minutes of driving. Figure 4.1b presents an example of five isochrones covering different time ranges. Isochrones are particularly relevant to coverage-based FLPs, as they delineate areas where everything within the polygon is covered within a specific time frame.

  The ORS isochrones service receives a list of coordinates and a range time t. The service returns a list of shapes corresponding to the reachable region in t seconds from each coordinate. An optional parameter `location_type` defines whether the coordinates are starting or destination points.
- *Matrix*: A distance matrix contains the distances between pair elements within a set. In this context, each element represents a coordinate, and the distance can be measured in either time or distance traveled in the road network. Distance matrices are fundamental for modeling distance-based FLPs, such as p-median problems.

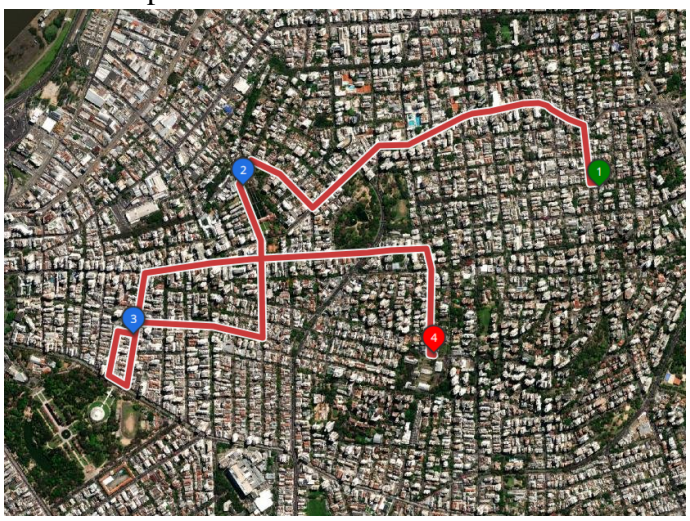
  The service takes a list of coordinates and returns the corresponding travel distances among the coordinates. Optionally, the client can specify which coordinates serve as source and destination nodes

All of the services provided by OpenRouteService return data in JSON[14] (JavaScript Object Notation) format. JSON is a lightweight data-interchange format that is easy to work with in various programming languages, including Python and Javascript.

However ORS is a valuable tool for analyzing road networks, its full potential is realized when coupled with the ability to interpret the results it provides. For example, merely generating isochrones for all facilities in an FLP may not be meaningful unless these isochrones are related to demand points in a meaningful way. In such scenarios, the

[12]https://openrouteservice.org/dev/#/API-docs/ (visited on 25/10/2023)
[13]Definition from the Collins Dictionary.
[14]https://www.json.org/json-en.html (visited on 25/10/2023)

Figure 4.1: Examples of the distance and isochrone ORS services.



(a) Routing through four distinct points.

(b) 2-, 4-, 6-, 8-, and 10-minutes isochrones.

Source: Screenshots taken by the author from <https://maps.openrouteservice.org>

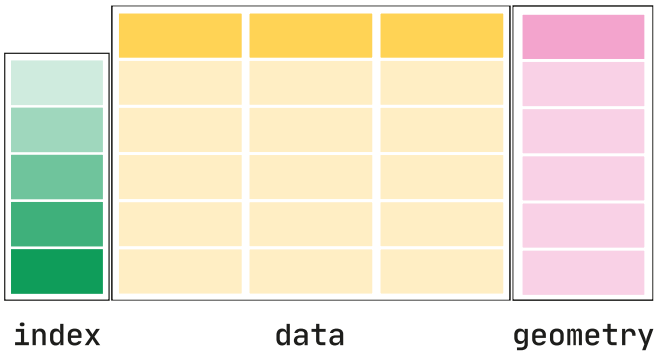
integration of a library for spatial and topological analysis becomes essential. In a Python environment, GeoPandas is an excellent choice for enhancing the analytical capabilities of a pipeline that utilizes ORS services.

4.3 GeoPandas

GeoPandas is an open-source project that enhances the capabilities of the widely used Python library Pandas[15] by incorporating geospatial computational support. In Pandas, a *DataFrame* is a data structure with a tabular format composed of rows (entries) and

[15]https://pandas.pydata.org/ (visited on 25/10/2023)

Figure 4.2: An illustration of a GeoDataFrame table.



Source: Extracted from <https://geopandas.org/> . Visited on 25/10/2023.

columns (attributes). Each of its columns is stored in memory as a Pandas *Series* object. A Series is a vector where all elements share the same data type. Therefore, Pandas store data column-wise in memory.

The GeoDataFrame is a specialized extension of a Pandas *Dataframe* that offers the unique ability to store geometry objects and conduct spatial operations. Figure 4.2 presents a representation of a GeoDataFrame. The index and data columns can be found in regular DataFrames. However, the geometry column, in pink, is specific to GeoPandas. This column is a Series, and its elements are Shapely[16] objects. *Shapely* is a Python package for the analysis and manipulation of geometry features. It primarily implements fundamental geometric object types such as points, lines, and polygons, aligning with the essential geometric components of GIS (as discussed in Section 4.1).

Like a database table, a GeoDataFrame offers a variety of functions for filtering data based on attribute values, grouping by a key, or merging two GeoDataFrames using a specific column. Furthermore, they can perform spatial operations based on the geometry column. Additionally, GeoDataFrames can perform various spatial operations based on the geometry column. For instance, by using the method `GeoDataFrame.area` we can compute the area of the geometric features. We can also measure the distance of multiple rows to a specific geometry using `GeoDataFrame.distance(object)`, among other capabilities.

[16]https://shapely.readthedocs.io/en/stable/manual.html (visited on 25/10/2023)

4.4 Final Remarks

In this chapter, we explored the fundamentals of GIS and how these systems empower real-world applications. We discussed the basic capabilities that makeup GIS, including spatial databases, visualization and mapping, and spatial analysis. Furthermore, we provided examples of GIS software, services, and libraries, discussing their advantages and applications in the context of FLPs. Subsequently, we introduced a powerful GIS service called OpenRouteService. To facilitate our understanding of ORS, we briefly introduced OpenStreetMap data since it serves as input to ORS. Lastly, we delved into the three ORS services that can support our work. In the last section, we saw that GeoPandas provides a complete set of spatial object management and analysis tools. This library can easily store data retrieved from diverse sources, including ORS. This integration provides all the necessary functionalities to build a FLP based on GIS.

In the following chapter, we will combine all the concepts presented in this work by describing our methodology for building an FLP to locate stroke centers optimally. Furthermore, we will present related works that use GIS and FLPs in the healthcare context. We will present the data we consider in our model and demonstrate how we use the Isochrone and Matrix ORS services along with GeoPandas to model a multi-objective FLP. Additionally, we will describe how to encode this model for optimization using NSGA-II, what objectives will be optimized, and the experiments to assess our methodology.

5 METHODOLOGY

Facility location models are crucial in identifying optimal regional locations for cost-effective solutions. However, applying these models to enhance treatment coverage, especially for time-sensitive diseases like strokes, should be more widespread. Stroke may cause several life-long disabilities and the death of various people that could not achieve treatment in adequate time. Unfortunately, this is a significant issue in low- and mid-income countries, where acute stroke treatment is often inadequately distributed. In the context of healthcare facility locations, multi-objective optimization problems naturally emerge. These problems involve modeling various objectives, including the cost of establishing a new healthcare unit, population coverage, capacity considerations, and travel distances for the population, among others. GIS plays a crucial role in these problems modeling since we are dealing with how people and essential services are spatially distributed in a region, and how people move around to reach these services.

In this chapter, we present a methodology that harnesses the integration of tools with GIS capabilities into the Python environment for modeling an FLP aimed at optimizing and assessing the network of acute stroke centers in regions where ASC data are available. We start the chapter by presenting a review of the related works in the literature. Therefore, Section 5.1 gathers relevant works that address the use of GIS and FLPs in the healthcare context. Following this review, we present in Section 5.2 how we model an FLP to improve acute stroke treatment accessibility. It describes which objectives we address, the demand points, the candidates set, and the ORS services we use. Next, Section 5.3 presents the optimization method, describing how we tackle the problem using NSGA-II. Finally, Section 5.4 presents the experiments to assess the proposed methodology.

5.1 Related Works

Solving FLPs to optimize healthcare networks has been the subject of extensive research for nearly four decades (Ahmadi-Javid; Seyedi; Syam, 2017). However, models based on GIS that incorporate existing infrastructure to evaluate and enhance aspects of a real-world healthcare network still need to be improved compared to the research on complex mathematical models tailored to specific scenarios (McNamara et al., 2020). In this section, we delve into existing works that harness the power of GIS in combination with FLPs in the healthcare context. We start by examining single-objective models, focus-

ing on addressing specific optimization objectives. Subsequently, we shift our attention to multi-objective models, which provide a more comprehensive approach to healthcare LAPs by simultaneously considering multiple objectives.

5.1.1 Single-Objective Models

Leira et al. (2012) evaluated the population coverage of existing Primary Stroke Centers (PSCs) and assessed the current certification process for stroke treatment hospitals in the USA—a voluntary procedure where hospitals aspiring to become certified stroke centers must express their interest to the relevant authorities. Their methodology involved aggregating the population of Iowa, USA, into ZIP code tabulation areas (ZCTAs), using the centroids of these areas as both facilities and demand points in a Maximum Coverage model. Using Microsoft's Bing Maps API, they constructed a time distance matrix to measure the distance between facility and demand points.

For solving MCLP, the authors adopted a greedy heuristic approach, selecting the facility with the most coverage in each iteration. The results revealed that the 12 existing self-initiated PSCs in the state covered approximately 37% of the population within a 30-minute travel time. However, if these PSCs had been strategically located using the MCLP, they could have covered up to 47.5% of the population. An additional 31 PSCs would be required to reach a 75% population coverage threshold. The study's findings highlight the importance of systematic planning for essential healthcare services like stroke treatment instead of the self-initiative approach.

Similarly, a study case for health planning in Chengdu, China, using the PSCLP is proposed by Deng, Zhang and Pan (2021). In this case, the authors assume systematic planning following China's National Development and Reform Commission guidelines, in which people should reach emergency care physicians within 15 minutes. The study aims to find the minimum number of hospitals to become part of the emergency medical system covering 90% of the population.

The work employs a genetic algorithm for selecting the minimum number of hospitals from 463 distinct candidates. The demand points are the population density of the region aggregated into gridded data from LandScan (Bhaduri et al., 2007). They used travel times as the distances among the entities in the model. A distance matrix that considers the travel spent by vehicles and on walks is computed using ArcGIS 10.6.

The study has shown that the 95 hospitals that provide emergency services cover

78.27% of the population. Moreover, it would require a minimum of 55 new hospitals to achieve the policy's goal. The results showed that the GA placed new hospitals in districts with worse access to emergency services.

5.1.2 Multi-Objective Models

In the previous section, we addressed works that predominantly focused on optimizing single-objective models. While these models have shown effectiveness, we advocate for incorporating multiple objectives in healthcare FLPs. Multi-objective optimization is better suited to encompass the intrinsic factors influencing healthcare demand, providing decision-makers with a more holistic perspective on the problem. In this section, we gather different studies in the literature that address this perspective.

Beheshtifar and Alimoahmmadi (2014) propose a multi-objective FLP for the placement of clinics in Region 17 of Tehran, Iran. The region is home to approximately 40,000 people, and there are no clinics. The study does not detail its dataset, but the optimization runs over 100 candidate facilities and 4466 demand points. The distance between the entities is based on the distance on the road network. Therefore, they do not use travel times. The FLP counts with four objective functions: (1) Minimizing transportation costs, based on the p-median model; (2) Minimizing unequal access to healthcare centers, based on the standard deviation of the distances; (3) Maximization of site suitability and compatibility of land-use, based on the distance of the candidates from factories, industrial sites and other features; (4) Minimization of cost of land purchase and facility establishment, based on the costs assigned to different areas in the city.

The study employed NSGA-II in the optimization, with 200 population size and 100 generations. For choosing the best solutions in the non-dominated front, they used the Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) metric (Behzadian et al., 2012). The results showed that they obtained different reasonable solutions depending on the importance of the objectives. However, they still need to choose a final one and provide a convergence analysis to assess the optimization's effectiveness.

Taking another direction, (McNamara et al., 2020) proposes a model for evaluating the current emergency department network of three regions in Canada. Note that the goal of the work is not to optimize an emergency department network but to measure how good it is. In addition to that, the evaluation criteria are based on two classical FLPs: the p-median and maximal coverage location model. They measure the average distance to

the nearest emergency department and the population that is covered within a range of 30, 40, 50, and 60 kilometers. Moreover, this work evaluates the emergency department redundancy and shortages in the regions.

The authors aggregate the population into the smallest geographic area for which the Canadian census disseminates data. Using ArcGIS, they measure the distance of the demand and facility points using the distance traveled via the road network. The discussion of results revolves around comparing the current ED network in the provinces. The authors benchmark the existing facilities' coverage of the population using the framework and explore various changes to the network, such as the implementation of collaborative emergency centers. The paper also discusses the study's limitations, such as the assumption that patients always seek care at the nearest emergency department and that the emergency departments have the same capacity due to a lack of adequate data.

In our work, we align with the recommendations by Leira et al. (2012), emphasizing the importance of utilizing FLPs to enhance accessibility to critical services such as stroke treatment. However, instead of maximizing the population coverage only, we propose an approach that optimizes population coverage and minimizes the average distance to the nearest ASC. The optimization method is also different since a greedy heuristic to an FLP tends to lose quality as the number of located healthcare units increases. Moreover, since we are dealing with a multi-objective FLP, aiming for a comprehensive view of the problem for a decision-maker, we require a more sophisticated method. We choose NSGA-II because this algorithm does not require prior knowledge of the optimal solutions while providing a spread non-dominance front. Finally, our work introduces a model for assessing the existing ASC network and any ED utilizing open-source data and GIS resources that can be readily applied to regions worldwide. In this respect, our work shares common ground with the approach presented by McNamara et al. (2020).

The proposed methodology is built over two axes: the problem's modeling and the optimization method. The former comprises the choice of objectives for evaluating ASC networks and the choice of data structures, facilities, and demand points to model the problem. The latter encompasses the choice of the optimization method, which is the NSGA-II, and the implementation of the objectives and codification of the solutions for the optimization.

5.2 Modeling the Problem

In this methodology, we aimed to create a practical model without excessive abstractions and variables, opting for simplicity that remains effective in real-world applications. Therefore, we sought the classical formulations of FLPs, which resulted in a multi-objective facility location problem incorporating PMLP, MCLP, and PSCLP characteristics. An aspect of an MCLP appears in our problem when we aim to maximize the population covered, of a PMLP when minimizing the average distance to the facilities, and of a PSCLP when we aim to minimize the cost instead of using a constant cost. The objectives of our model can be described as follows:

- *Accessibility*: Maximize the population covered by the facilities in the region within a range of 45 minutes of driving time. This threshold is based on a study conducted in Austria (Ferrari et al., 2018), which deemed 45 minutes an appropriate travel time.
- *Distance*: Minimize the average distance from demand points to their closest facilities. While the accessibility objective can find the facilities that maximize the population covered within 45 minutes, it does not look to the uncovered points. The distance objective will minimize the travel times in the whole network, shortening the distance to ASCs and mitigating harmful outcomes of living in non-covered regions.
- *Cost*: Minimize the number of new stroke centers. We assign a uniform cost of one for each candidate facility, as specific data regarding the cost of establishing a new stroke center is unavailable.

The problem at hand involves optimizing the combination of open facilities to achieve the best possible values for each objective. Naturally, there are trade-offs among these objectives. For instance, we cannot improve the accessibility or the distance objective without the increasing of cost. Accessibility is closely tied to the population density of the region, whereas the distance objective is contingent on the road networks. Consequently, we may find a configuration that improves both accessibility and distance simultaneously, but we may also encounter situations where these objectives are in conflict.

To evaluate both the distance and accessibility objectives, the model requires a predefined set of candidate facilities and a set of demand points.

Facility Candidate Set

The facilities are based on current and potential stroke center coordinates obtained from specialists in stroke treatment infrastructure. We gather information about stroke treatment protocol in healthcare units within Rio Grande do Sul. The data comprise two types of healthcare units. The first type includes the hospitals that currently have an acute stroke treatment protocol, specifically, the ones that treat acute ischemic stroke (AIS) with intravenous thrombolysis (IVT). AIS is characterized by a sudden loss of blood circulation in a vascular region of the brain, and the benefits of its treatment are time-sensitive (Powers, 2020). IVT is a typical treatment for AIS that can significantly reduce the outcomes of the stroke if administered within adequate time.

The second category of hospitals includes hospitals without an existing stroke treatment protocol but possessing essential infrastructure, such as non-contrast computed tomography scanners, monitored beds in the emergency department, and 24/7 laboratories. These hospitals can become stroke centers with the addition of specialized neurologists and an IVT-based stroke treatment protocol.

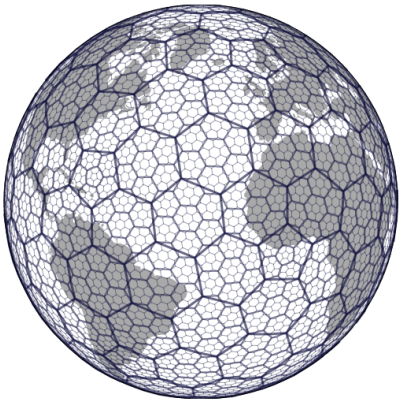
Demand Points

In this study, we define demand as the population within the region. Therefore, we sought a method to aggregate the population into static and discrete regions, creating a population density map. For this purpose, we found the publicly available Kontur Population: Global Population Density dataset to be an ideal fit. This dataset can generate a highly detailed population density map using the H3 indexing system, which we can then use to determine the demand coordinates in the region.

H3 API is a library that provides a geospatial indexing system that partitions the globe into a hexagonal grid. Therefore, every coordinate of the globe can be mapped to a hexagonal cell, making a discretization of the latitude/longitude coordinates system to a finite grid. Every H3 cell has a unique identification number. Figure 5.1 illustrates this grid system. It is based on the Icosahedral Snyder Equal Area aperture 3 Hexagon (ISEA3H) geodesic discrete global grid system (Sahr; White; Kimerling, 2003). Note that it has a hierarchical fashion so that the hexagonal cells can be subdivided into smaller and smaller cells.

The hierarchical relationships are achieved through the different *resolutions* of

Figure 5.1: H3 Hierarchical grid system



Source: Extracted From <https://www.uber.com/en-BR/blog/h3/>. Visited on 25/10/2023

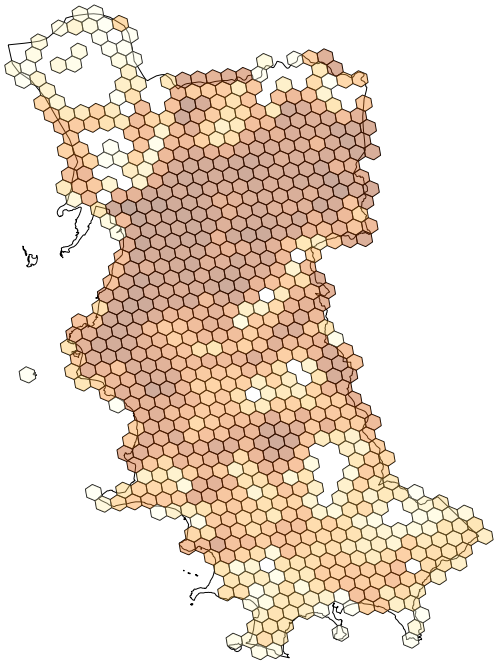
hexagonal grids. Table 5.1 displays the relation of the different H3 resolutions, the average area, and the number of hexagons. There are 16 different resolutions, with higher numbers representing finer grains of discretization. Each hexagon cell of resolution r contains seven hexagons of resolution $r + 1$; each cell of resolution $r + 1$ contains seven hexagons of resolution $r + 2$, and so on. This establishes parent / child relations among cells that can be retrieved using the proper H3 API methods.

Table 5.1: Hexagonal Grid Properties

Res	**Average Hexagon Area (km^2)**	**Number of hexagons**
0	4,357,449.416078381	110
1	609,788.441794133	830
2	86,801.780398997	5,870
3	12,393.434655088	41,150
4	1,770.347654491	288,110
5	252.903858182	2,016,830
6	36.129062164	14,117,870
7	5.161293360	98,825,150
8	0.737327598	691,776,110
9	0.105332513	4,842,432,830
10	0.015047502	33,897,029,870
11	0.002149643	237,279,209,150
12	0.000307092	1,660,954,464,110
13	0.000043870	11,626,681,248,830
14	0.000006267	81,386,768,741,870
15	0.000000895	569,707,381,193,150

During the course of this work, the version of the Kontur's dataset with finer resolution is with resolution eight hexagons. Kontur attributes a population number to each

Figure 5.2: Population density of Porto Alegre city according to Kontur's dataset.



Source: Elaborated by the author. 2023.

H3 cell based on overlapping the Global Human Settlement Layer (GHSL) with Meta's High-Resolution Settlement Layer (HRSL) population data where available. The dataset allows geospatial computation along a fine-grained population density map. The data is distributed as a GeoPackage (.gpkg) file, which can be read by GeoPandas. A description of the dataset's codebook is found below:

- fid [integer]: record order in data upload
- h3 [h3index/text]: H3 index of hexagon
- population [double]: total population inside a hexagon
- geom [geometry]: Polygon, EPSG:3857

In addition, we have data that describe the boundaries of a region. Therefore, we can filter all H3 cells within specific cities, states, or countries. Each H3 cell in this filter will map to a demand point in our facility location problem. Figure 5.2 is an example of the demand points within the boundaries of Porto Alegre city. The universe set $U = \{h_1, h_2, ..., h_m\}$ contains all cells, and each cell h_i has a population associated $p(h_i)$. In the figure, cells with darker colors have more population.

H3 API provides a method to get the latitude/longitude coordinates of the centroid and the vertices of a cell. We use their centroids as the demand points to compute the

distance in the road network. We use the cells to get the amount of the population covered by the facilities.

Isochrones and population coverage

Our method calculates the population covered by a facility by determining their associated isochrones. We can estimate the number of people within each facility's isochrone using the population density layer.

For computing the population coverage of each facility, we rely on the support of the ORS Isochrones Service. We compute an isochrone for each facility with the `location_type` indicating that they are destinations. The set of hexagonal cells $H \subseteq U$ that overlaps the isochrone of the facility d is considered to be covered by d. Summing up the population of all hexagonal cells $h \in H$ results in the population coverage $P(H)$ by d, as shown in Equation 5.1.

$$P(H) = \sum_{h \in H} p(h) \tag{5.1}$$

To find cells that overlap the isochrone, we rely on the R-Tree-based spatial indexing supported by GeoPandas. Geopandas' spatial index allows, among other things, the use of fast spatial queries to check if a Shapely geometry intersects a set of geometry features indexed by the R-Tree. In the context of this work, we build an R-tree from the population density H3 cells and query all cells that intersect an isochrone shape.

Distance Matrix

The optimization requires the driving time distance between the entities of the model (facilities and demand points) with respect to the distance objective. Therefore, we compute a travel distance matrix from the demand points to the facilities.

We use the ORS Matrix Service for this computation. The facilities' coordinates are the destination points. The coordinates of the demand cells centroids are the source points. Given the number of destinations n and sources m, the Matrix Service returns a JSON containing the metadata and a $m \times n$ matrix. An element $d_{j,i}$ in the matrix is the driving time, in seconds, from a source coordinate j to a destination coordinate i.

Flowchart

As we saw previously, we need to calculate the distance matrix that stores travel times between demand points and candidate facilities. We also need to compute the 45-minute isochrones of the facilities. Figure 5.3 provides an overview of the processes of computing both data structures. The process begins with two inputs: the population density data for the region, which generates the demand points, and the database of hospitals, which contains the candidate facilities within the model. The H3 API computes the coordinates of the H3 cells centroids. They subsequently serve as source points to the matrix service. Moreover, an R-tree is created from the H3 cells to create a spatial database. Thereby, we can query all sets that overlap the isochrones of each facility, generating a set of covered cells for each candidate.

Computing the necessary data structures, we can proceed with the description of the optimization method.

5.3 Optimization Method

In our method, we seek to find solutions for each number of candidate facilities. Following this idea, we can find a non-domination front that gives a set of optimized solutions either on accessibility or distance for each value of cost. Then, a decision-maker can evaluate which solution is more suitable according to their budget. Since this problem has only three objectives, domination-based MOEAs are suited for its optimization. Therefore, we choose NSGA-II as our optimization algorithm. As a prominent domination-based MOEA (refer to Section 2.5), NSGA-II can find a non-domination front with these characteristics straightforwardly. Whenever the algorithm finds a smaller-cost solution, it will remain in the population as a non-dominated and spread solution. Otherwise, if we opted for a decomposition-based MOEA, we would have to sample reference vectors in the objective space so that the algorithm could find solutions for each number of selected candidate facilities. This task is not trivial and would require a dedicated study.

Given the NSGA-II flexibility, we follow a straightforward process to model a problem to be optimized by the algorithm:

1. We encode the decision variables associated with the FLP, which in this case are candidate facilities, to solutions that NSGA-II can interpret;

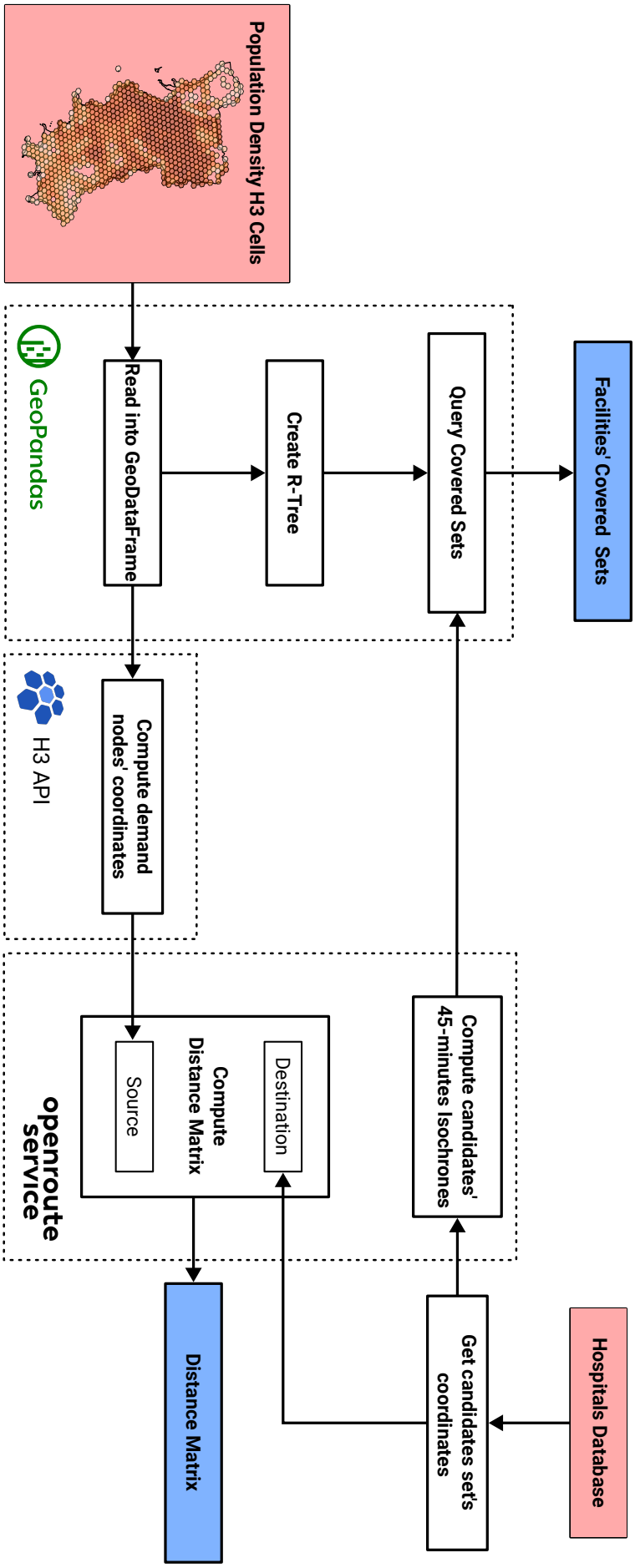


Figure 5.3: Flowchart to compute the necessary data structures for the model.

Source: Elaborated by the author. 2023.

2. We define objective functions that map these solutions to real numbers, representing their quality and enabling the algorithm to compare different solutions;
3. We need to define the parameters in which the algorithm will run.

Solution Encoding

In this problem, we have multiple candidate facilities that must be opened to find an optimal combination regarding the cost, accessibility, and distance objectives. Therefore, the number of decision variables of the problem is the number n of candidate facilities. The solutions are binary-encoded in this methodology. Therefore, a vector $\mathbf{x} = \{x_1, x_2, ..., x_n\}$ represents an individual, where each element x_i is either 0 (indicating the facility is closed) or 1 (indicating the facility is open/selected). Hence, an ASC network V comprises all indices in $\mathbf{x}$ equal to one.

Objective Functions

We described three objectives to optimize in the beginning of this section: accessibility, distance, and cost. We model these objectives as objective functions to be used in the NSGA-II as follows:

- *Accessibility*: each candidate i has a set $H_i \in U$ of covered hexagonal cells. The accessibility is then computed by summing up the population P from the union of all covered cells by the selected facilities in an ASC network V:

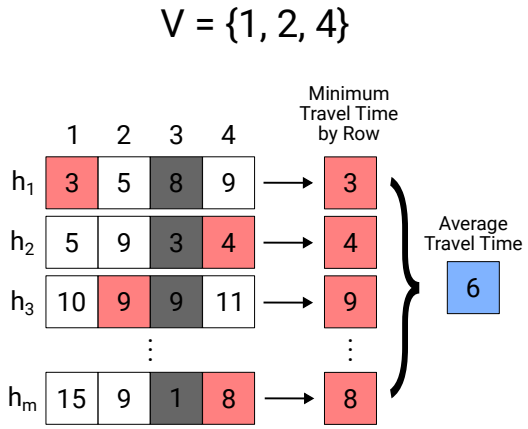
$$f_1(\mathbf{x}) = P(H), H = \bigcup_{v \in V} H_v \tag{5.2}$$

- *Distance*: To find the average travel time to the nearest ASC in a network described by V, we first find the row-wise minimum values of the distance matrix d. Then, we compute the average of these values. Each row j represents a demand point, while each column represents an ASC in V:

$$f_2(\mathbf{x}) = \frac{1}{m} \sum_{j=1}^{m} \min_{1 \leq i \leq n} \{d_{j,i}\} \tag{5.3}$$

Figure 5.4 illustrates this operation for a solution $V = \{1, 2, 4\}$. Note that the

Figure 5.4: Example of the *distance* objective computation for a solution $V = \{1, 2, 4\}$. The $m \times 4$ distance matrix represents a problem with 4 candidates and m demand points.



Source: Elaborated by the author. 2023.

computing does not consider the third column because the network V does not contain the third facility.

- *Cost*: The cost objective in this work is equal to the number of selected centers. The cost is computed by the sum over all elements in a solution vector $\mathbf{x}$, with a constraint to avoid a solution with zero selected facilities:

$$f_3(\mathbf{x}) = \sum_{i=1}^{n} x_i, \text{s.t} \sum_{i=1}^{n} x_i \geq 1 \tag{5.4}$$

5.4 Experiments

We present a methodology for assessing and optimizing acute stroke center networks based on accessibility, distance, and cost objectives. To validate the effectiveness of our methodology, we conduct a case study in Rio Grande do Sul, Brazil, utilizing data on the stroke treatment capabilities of hospitals within the state. Furthermore, we evaluate the optimization capabilities of the NSGA-II algorithm in the context of our proposed model. Therefore, we subject the optimization algorithm to testing in two distinct scenarios.

In the first scenario, we focus on efficiently expanding the existing ASC network by adding new stroke centers. Hospitals treating acute ischemic stroke with intravenous thrombolysis with established treatment protocols were categorized as ASCs. The set of candidate facilities comprises the potential stroke centers outlined in Section 5.2 of our

methodology. This scenario is particularly relevant because it addresses the pressing need to expand the ASC network in response to the increasing demand for stroke care in Rio Grande do Sul.

The second scenario involves a more comprehensive approach. Here, we reset the ASC network and consider the current stroke centers as candidates alongside the potential ones. This scenario is proposed because, thereby, we can assess the current distribution of ASCs within Rio Grande do Sul based on our objectives. Moreover, this scenario runs with more decision variables and a more complex search space, challenging NSGA-II even more.

We discuss the results using the accessibility and distance objectives by the cost of solutions. Since the cost objective is an integer, at least one solution lies in the Pareto-front for each cost value until the addition of a new hospital cannot improve the other objectives. We also discuss the results using the solutions' hypervolumes. The hypervolume metric quantifies the portion of the objective space dominated by a solution (refer to Subsection 2.2.2). The hypervolume metric gives a more comprehensive view of the best solution for all objectives. For computing the hypervolume of solutions, a normalization of the objective values is done, considering the minimum and maximum values possible for each scenario.

We use the NSGA-II implementation from Pymoo (Blank; Deb, 2020), a multi-objective optimization framework in Python. The parameters of NSGA-II runs are described in Table 2. Twenty independent replicates of all experiments were conducted to guarantee the statistical significance of the findings. Due to the considerable computational duration—requiring a whole day to complete twenty experiment replicates—the parameterization was derived empirically. This process involved executing a singular experiment using specific parameters, analyzing the outcomes, adjusting the parameters, and re-executing the experiment.

Table 5.2: NSGA-II parameters for the experiments.

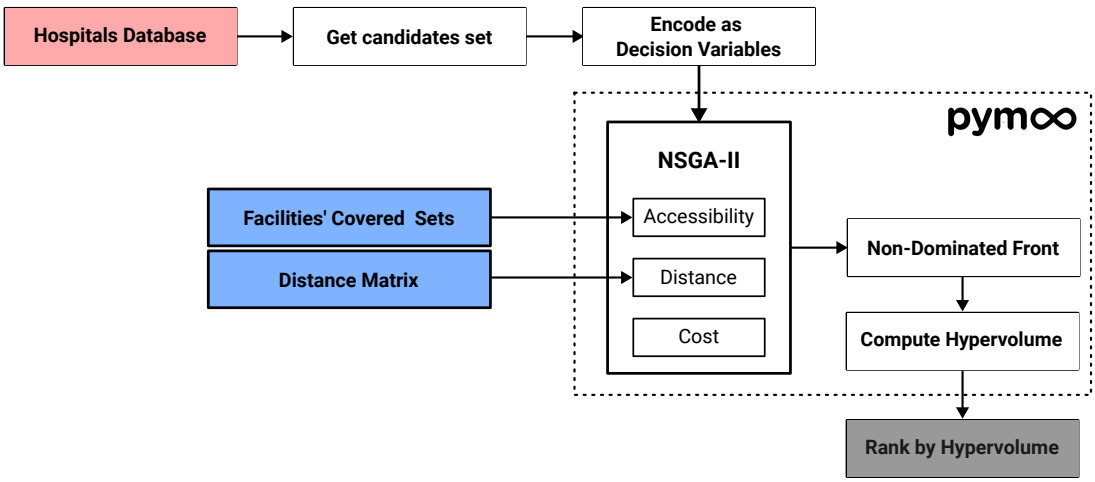
Parameter	**Value**
Chromosome Size	116 (Scenario 1) and 149 (Scenario 2)
Population Size	200 individuals
Generations	400
Initial Population	Binary Uniform Random Sampling
Crossover	Half Uniform Crossover
Mutation	Bitflip Mutation (p=1%)

To assess the NSGA-II optimizing the three proposed objectives, we compare to

baselines that intend to optimize the accessibility and the distance objectives. Two distinct methods are proposed as baselines, NSGA-II optimizing only cost and either accessibility or distance and a greedy heuristic to optimize either accessibility or distance. These comparisons provide a way to test the optimization under three objectives since both baseline methods try to improve to the limit of one of the functions. The comparison will also give insights into how the improvement of the different objectives affects the hypervolume of the solution.

Figure 5.5 gives an overview of how everything glues together. We use the candidates set to encode the solutions as described previously. Then, the facilities' covered sets and the distance matrix are used to compute the evaluation function in our model. NSGA-II optimization will find a non-dominated front. Finally, we compute the hypervolume of the front and the solutions individually. Ranking the solutions by their HV give us a reference of which solutions are more relevant in the optimization.

Figure 5.5: Flowchart including the optimization.



Source: Elaborated by the author. 2023.

5.5 Final Remarks

In this chapter we described our methodology. First we gave a background on how different works model FLPs along with GIS in the healthcare, discussing their methodologies and the objectives they optimize. Then, we detailed our work, describing the objectives we optimize to enhance a acute stroke center network. We discussed how we used the demand and candidate points to compute the distance matrix and the facilities' covered sets. Finally, we described how we model a NSGA-II for the problem, and the

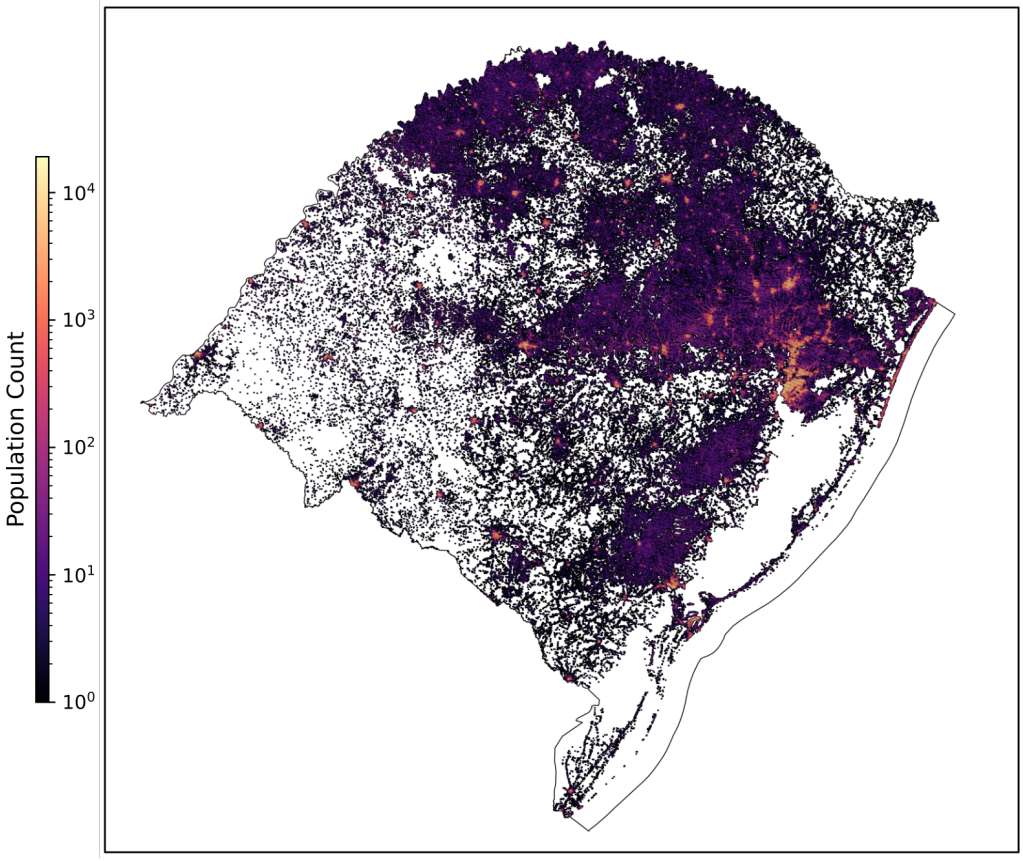
experiments we made to assess the whole methodology.

In the upcoming chapter, we will delve into the results of two experimental scenarios, aiming to provide an insightful perspective on the distribution of ASCs in RS, and and the effectiveness of FLPs in this context. The NSGA-II performance will be validated based on the hypervolumes of the solutions within the non-dominated front. Furthermore, we will compare the results obtained through NSGA-II with those achieved using greedy approaches and NSGA-II optimization considering only two objectives.

6 RESULTS

In this chapter, we present and discuss the results of the experiments, described in Section 5.4, for assessing the proposed methodology. As mentioned, our case study is in Rio Grande do Sul, which currently hosts 33 acute stroke centers and 116 potential stroke centers. Figure 6.1 illustrates the population density map of RS according to Kontur's dataset. The map comprises $175,221$ H3 cells, representing the demand points in our model. The dataset indicates a population of $10,986,852$ individuals, whereas the 2022 population census[1] reports $10,880,506$ residents. Therefore, Kontur's dataset presents 0.98% more population than the official number. We use Kontur's number when discussing the results.

Figure 6.1: Population density map of Rio Grande do Sul according to Kontur's dataset. The population in the cells are normalized in logarithmic scale. Hotter colors mean more population. White spaces mean no demand points.



Source: Elaborated by the author. 2023.

This chapter is organized into three sections. The first two sections focus on the outcomes achieved through the optimization, discussing how the obtained solutions can

[1] https://cidades.ibge.gov.br/brasil/rs/panorama (visited on 25/10/2023)

help to improve acute stroke treatment in Rio Grande do Sul. The final section shifts its focus to the quantitative analysis of the results, evaluating the overall effectiveness of the optimization process.

6.1 Effectivelly Increasing the Current ASC Network

We start the analysis by evaluating the existing acute stroke center network using the criteria outlined in our methodology, described in Section 5.2. Currently, are 33 ASCs in place, collectively providing a 45-minute population coverage of 63.79%. This means that approximately 7 million individuals reside within an adequate travel distance to access at least one ASC.

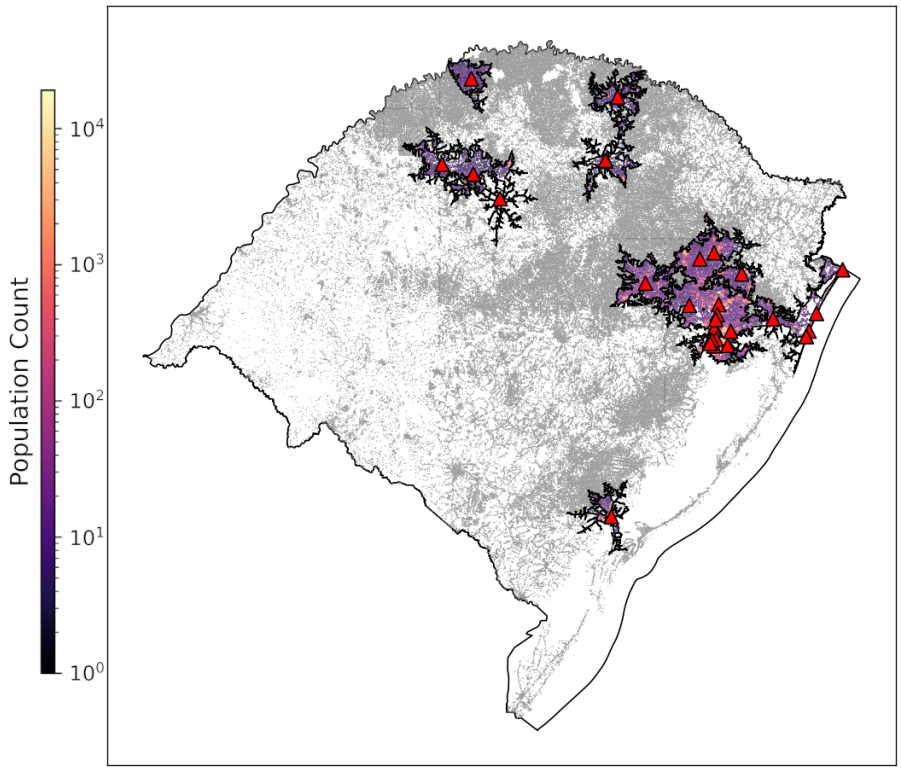
Figure 6.2a presents the 33 ASC along their 45-minutes population coverage. It is noteworthy to emphasize the population disparity between RS's eastern and western regions. The eastern region hosts the state's metropolitan area, which accounts for approximately 4 million residents. The metropolitan region comprises 17 of the 33 ASC (51.5% of all ASCs within the state). In contrast, none of the hospitals in the southwest region of Rio Grande do Sul have established protocols for treating acute strokes, indicating a geographical disparity in treatment access.

This inequality's bad outcomes become more evident in Figure 6.2b. It shows a random sample of 1000 demand points connected with their nearest ASC. It stands out in the figure that a significant amount of demand points are connected to very far stroke centers. Indeed, the average travel time to ASCs in this configuration is approximately 101.4 minutes, more than double the 45-minute threshold.

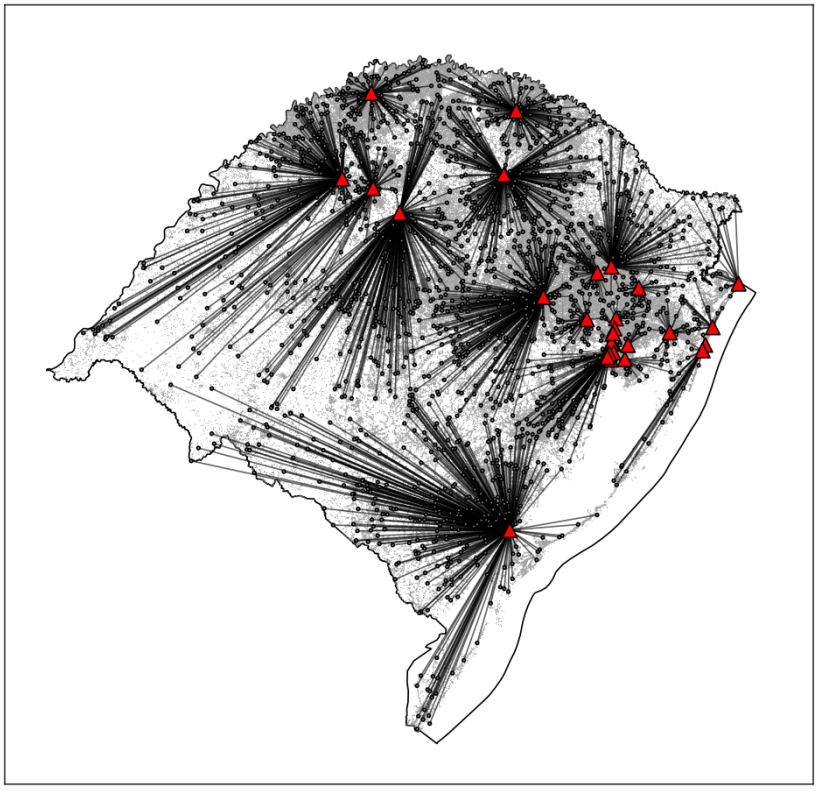
For a more detailed view of the current scenario, we bring Table 6.1. It details the percentage of population and demand points within different travel time intervals to reach their nearest ASC. Again, 63.79% of the population can reach an ASC within the 45-minute threshold. In contrast, the table shows that the number of covered demand points accounts for only 20.86% of the set. Therefore, most of this percentage is due to the metropolitan region being almost completely covered, a small but very populated area. Since the demand points represent regions with at least one person and equally distributed areas, we can perceive a substantial geographical disparity in access to stroke treatment. If we take the columns with more than 45 minutes of travel time, we perceive that $79,14\%$ of the demand areas have improper access to ASCs.

Established the current scenario in the state, we follow with the results of the

Figure 6.2: Stroke centers coverage and the nearest stroke centers from the demand points. Red triangles indicates a stroke center.



(a) Stroke centers coverage



(b) Fastest stroke centers to reach from a sample of 1000 demand points

Source: Elaborated by the author. 2023.

Table 6.1: Aggregation of the demand points and population by the travel time they take to reach their nearest ASC.

Travel Time Intervals (min)	**0-45**	45-90	90-135	135-180	180-450
Percentage of Population	**63.79%**	16.12%	10.03%	4.45%	5.61%
Percentage of Demand Points	**20.86%**	29.76%	24.17%	13.14%	12,07%

Source: Elaborated by the author. 2023.

optimization. For brevity of the analysis, we will discuss only the solution with the best hypervolume [2]. The hypervolume computing on this scenario is based on a reference solution that selects 116 hospitals, and covers 36% of the population, with 101.4 minutes of average distance to the nearest ASC. These values are selected in accordance to the worst possible values for each objective in this scenario. The ASC network with the highest hypervolume in the experiments adds 38 candidate locations to the 33 current ASCs. Therefore, with 71 ASC, the state can have a population coverage of $88,99\%$ and an average travel time to the nearest ASCs of approximately 49 minutes. Table 6.2 presents the updated RS coverage details, just as Table 6.1.

Table 6.2: Aggregation of the demand points and population by the travel time to reach their nearest ASC.

Travel Time Intervals (min)	**0-45**	45-90	90-135	135-180	180-450
Percentage of Population	**88.99%**	9.55%	1.28%	0.15%	0.02%
Percentage of Demand Points	**50.00%**	41.09%	7.84%	0.85%	0.21%

Source: Elaborated by the author. 2023.

Comparing both tables, we observe a considerable improvement of the ASC network concerning the population coverage. With 38 additional ASCs, the coverage can grow 25% in the state. Another noteworthy comparison is that, in the current ASC network, approximately 20% of the total population is more than one hour and a half from the nearest ASC. After the optimization, this number drops to only 1.45%.

The number of covered demand points also increased substantially, jumping from 20.86% to 50%. Also, a massive slice of 91% of the demand points became within one hour and a half from the nearest ASC. It is significant considering that before the optimization, only 50.62% falls within this range.

Finally, Figure 6.3 shows the map of RS with the optimized ASC network. The east region of the state was the only one that had not received a significant upgrade in the network since they already concentrate most ASCs. We can see in Figure 6.3b a better-distributed ASC network with shorter lines, which indicates shorter travel times to reach

[2] Refer to Table 8 in Supplementary Material for a list of the best solutions for each value of cost.

the nearest ASCs.

The number of 71 ASCs may look infeasible at first glance; however, we have good examples for comparison to show that we need to consider this number. For instance, we know that Austria, a country three times smaller in area than Rio Grande do Sul, has 39 acute stroke centers in the country (Ferrari et al., 2018). Another example is Germany, which comprises 335 acute stroke centers, despite having an area about $1.2\times$ larger than RS (Neumann-Haefelin et al., 2021).

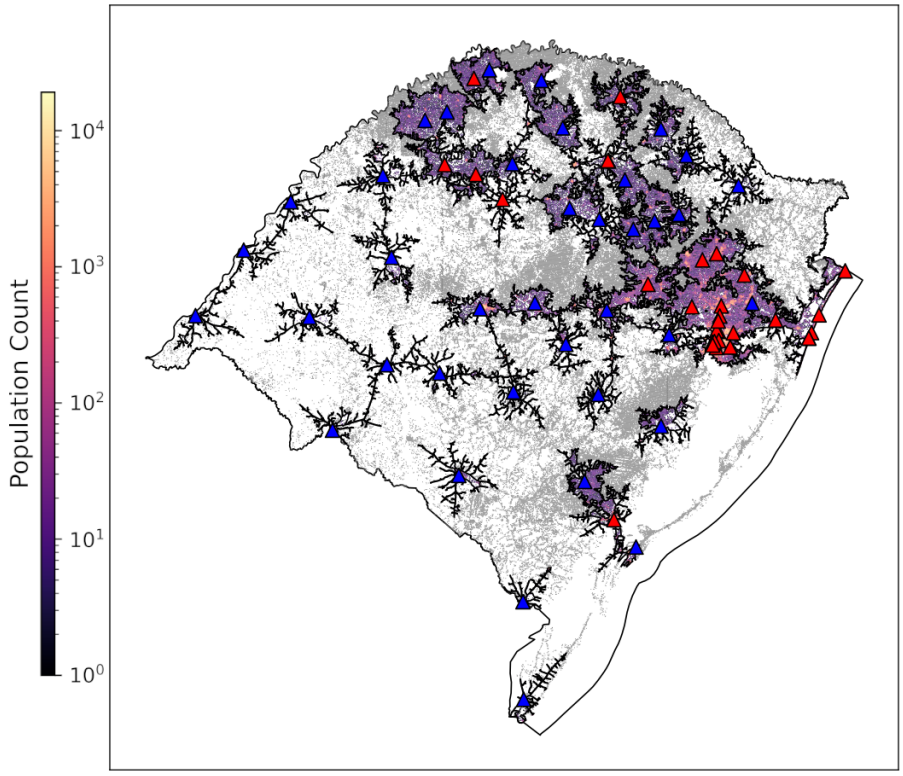
6.2 A Hypothetical Analysis: Resetting the Stroke Centers Network

The results in the last section showed evidence of how poor is the current ASC network in Rio Grande do Sul in the terms of accessibility and distance objectives. We presented a possible path to improve the network based on the solution with the highest hypervolume. Adding 38 potential stroke centers to the network can result in a configuration with $88,99\%$ of population coverage and 49 minutes in the distance objective. Moreover, to the best of the author's knowledge, RS does not use systematic planning to enhance the stroke treatment catchment. Essential services such as stroke treatment need more attention from the authorities, and the use of FLPs is essential.

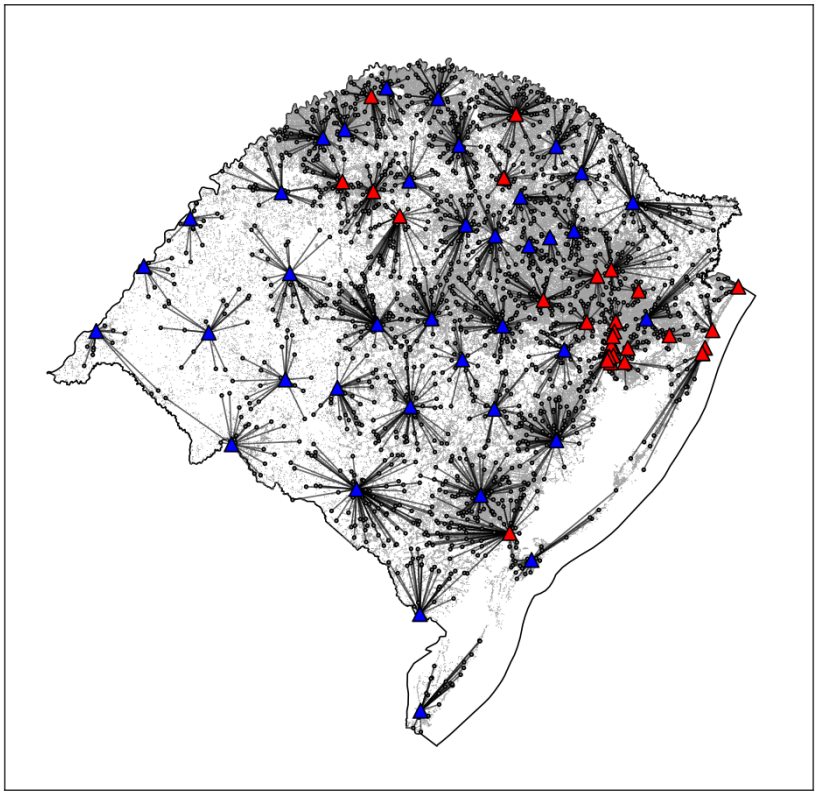
To demonstrate the benefits of a planned network expansion, we attempted resetting the current ASC network and locating stroke centers for an entirely new network with our model. Figure 1 compares the optimization results on the previous and present scenarios by the maximum hypervolume obtained for each number of candidates placed. Now, the reference point for computing the hypervolume is a solution that selects 149 candidates, and covers 0.0% of the population, with $507,03$ minutes of average distance to the nearest ASC. Both scenarios are normalized according to this new reference point. The red columns represent the optimization in the first scenario. The blue columns represent our new scenario that resets the ASC network. The respective numerical results can be found in Table 8 and Table 8 in the Supplementary Material.

By comparing the hypervolume of two solutions with equal values for the cost objective, we have a good indicator of the quality of the accessibility and distance of the solution. Let us look at the hypervolume of solutions with 33 ASCs in both scenarios. This number represents the actual number of ASCs in the state. We see a discrepant difference between the optimized and current networks. Indeed, with the same number of ASCs as today, RS could have a population coverage of 80.21% and an average travel

Figure 6.3: Stroke centers coverage and the nearest stroke centers from the demand points. Red triangles indicates a stroke center.



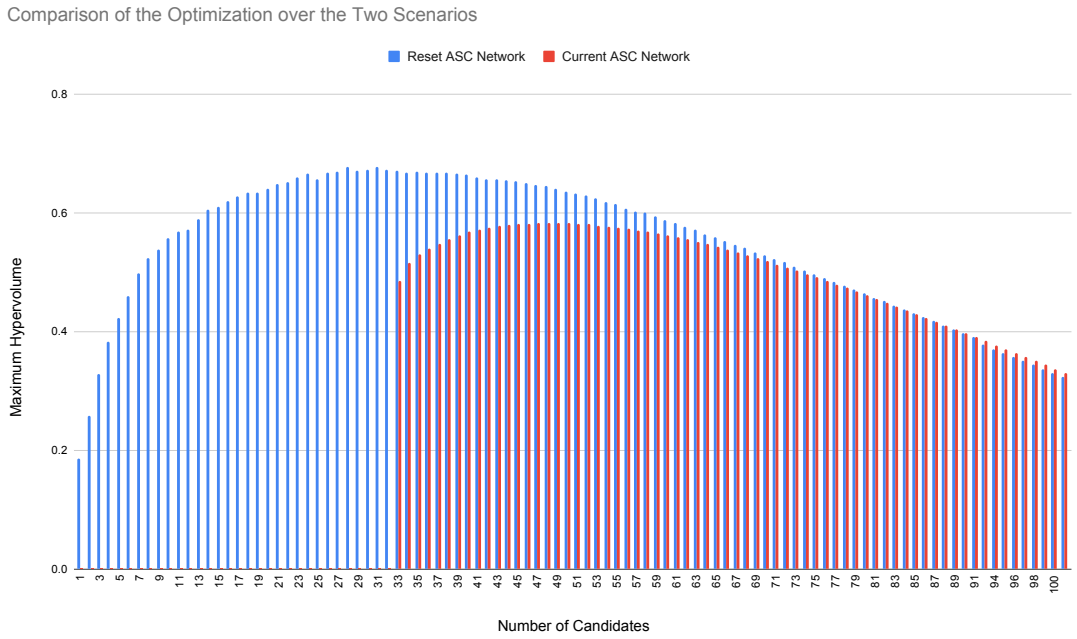
(a) Stroke centers coverage



(b) Fastest stroke centers to reach from a sample of 1000 demand points

Source: Elaborated by the author. 2023.

Figure 6.4: Hypervolume comparison of the optimization under the two scenarios.



Source: Elaborated by the author. 2023.

time of 60.6 minutes. These numbers represent a difference of 10.45% and 40.8 minutes, respectively, to the actual situation in the state.

Looking from another perspective, to achieve comparable results in terms of the accessibility and distance objectives as the best hypervolume solution in the first scenario (comprising 71 ASCs, 88.99% population coverage, and an average distance of 49 minutes), we need to select 13 fewer candidates in the reset network. With 58 ASCs, the state could cover 89.3% of the population, with 49.8 minutes on average to reach an ASC.

These results confirm the importance of a systematic planning using FLPs to improve the acute stroke center network in the state. However, the limitations of the model must also be taken into account. The model does not consider the capacity of healthcare units. Addressing this attribute, the redundancy of multiple stroke centers covering the exact locations is desirable to attend efficiently to the population, especially in high-populated regions.

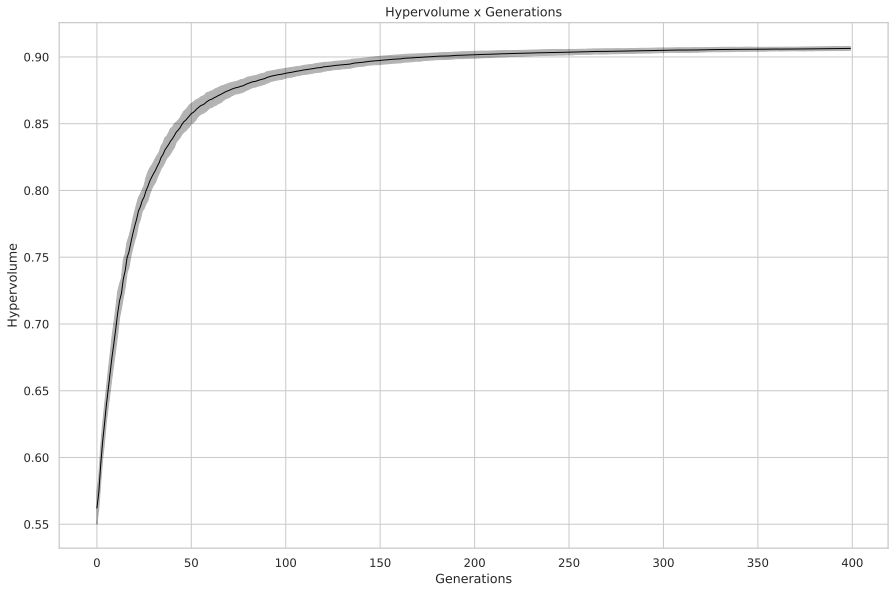
6.3 Analysis on the NSGA-II

Analyzing the 20 runs of the optimization, we can state that the NSGA-II can find a converged non-dominated set. Table 6.3 presents the average HV of the non-dominated

Table 6.3: Non-dominated set's average hypervolume on both scenarios.

Scenario	1	2
Hypervolume	0.825 ± 0.002	0.902 ± 0.002

Figure 6.5: Average hypervolume by generations in the second scenario.

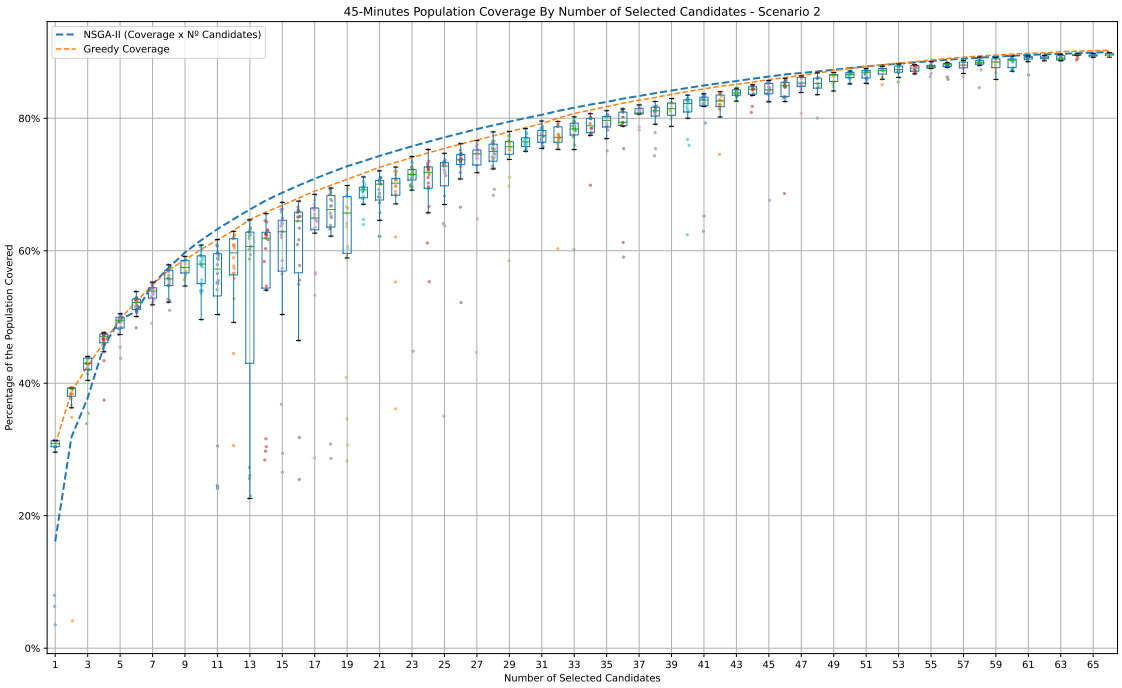


set on both optimization scenarios. The ideal point for measuring the HV was adding 1 hospital, population coverage of 90.626%, average traveled time of 46.43 minutes. The standard deviation of only 0.002 indicates that all runs found a set that dominates the objective space similarly. Figure 6.5 provides more information on how the hypervolume behaves through the generations. The plot presents the average hypervolume of the executions in the generations. We can visualize that the algorithm finds a *plateau*, which means that it cannot increase the HV occupied by the set of solutions.

The obtained HV cannot indicate alone the optimization's quality because we do not know the real optima points for the problem. However, we can evaluate the optimization by making further analysis. Our performance analysis center on the second scenario, given its higher complexity compared to the first. We have, therefore, filtered and displayed the results for the accessibility and distance objectives for each number of selected candidates in the form of box plots, presented in Figures 6.6 and 6.7, respectively. We bring the numerical results related to these plots in the Supplementary Material.

To provide a meaningful basis for comparison, Figure 6.6 incorporates a reference curve highlighting the outcomes of a greedy heuristic used to maximize population coverage. The greedy heuristic was the method used by Leira et al. (2012). It selects the candidate that improves at maximum the objective in each iteration. Additionally, the plot includes the average results of 20 runs of NSGA-II optimization with two primary objec-

Figure 6.6: 45-Minutes population coverage by number of selected candidates under the second scenario of optimization.

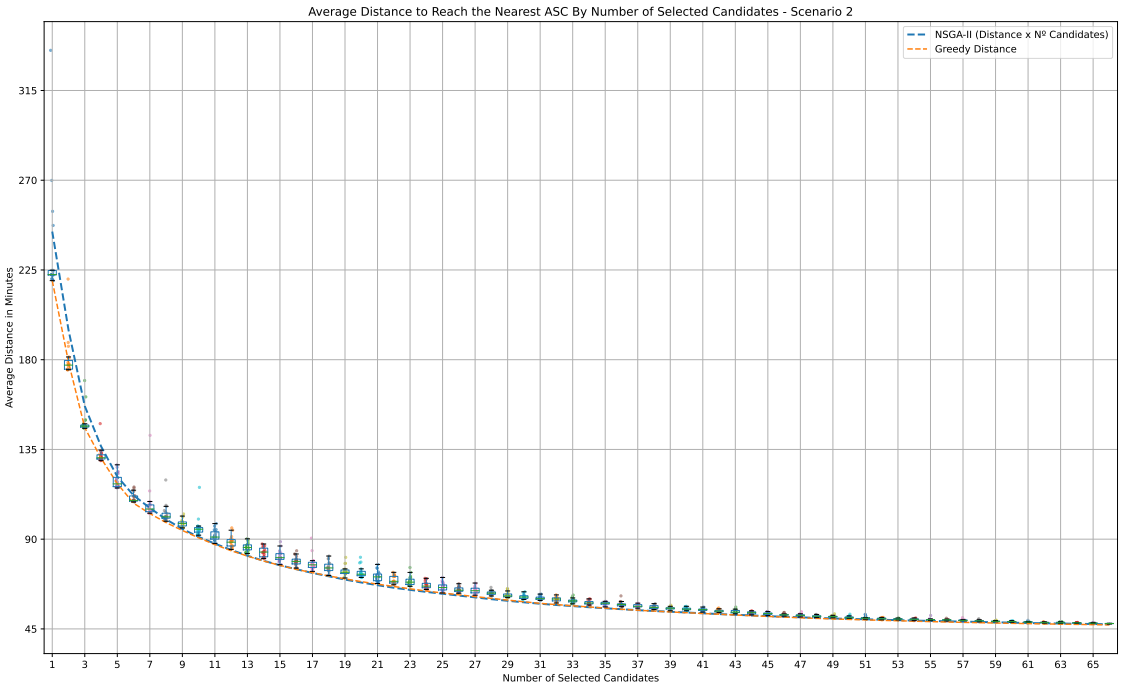


tives: cost and accessibility. Similarly, Figure 6.7 offers analogous comparisons, focusing on the distance objective instead of accessibility. Notably, in each of the 20 independent runs, NSGA-II identified solutions with the same number of selected candidates but different prioritization of objectives. The method might find solutions that dominate the distance and cost objectives while having a poor value for the accessibility objective. Since the algorithm tends not to overlap covered regions, the selected candidates become somehow distributed. Therefore, while not optimal, optimizing the coverage might improve the distance objective. The opposite is untrue since the distance objective does not consider the population.

Figure 6.6 provides valuable insights in terms of the efficacy of the different methods. Initially, the greedy heuristic achieves superior results up to a point, specifically with up to seven selected candidates. Beyond this threshold, the myopic nature of the greedy heuristic diminishes its effectiveness. Notably, NSGA-II with two objectives emerges as the standout performer from the seventh candidate selection onward, indicating its effectiveness in addressing this problem. In contrast, we observe that NSGA-II with three objectives, which is our primary focus, struggles to maximize coverage beyond the placement of eleven candidates. Furthermore, it's worth noting that the twenty independent executions exhibit greater divergence in their obtained solutions from 11 to 19 candidates.

Figure 6.7 exhibits a similar pattern to that seen in Figure 6.6. Here, both NSGA-II

Figure 6.7: Average drive distance by number of selected candidates under the second scenario of optimization.



with three objectives and the greedy heuristic perform well with a limited number of selected candidates, but their effectiveness diminishes as more candidates are added. However, a noteworthy difference is evident in this case: all the approaches effectively address the problem, and there are fewer discernible differences in the solutions obtained by these approaches.

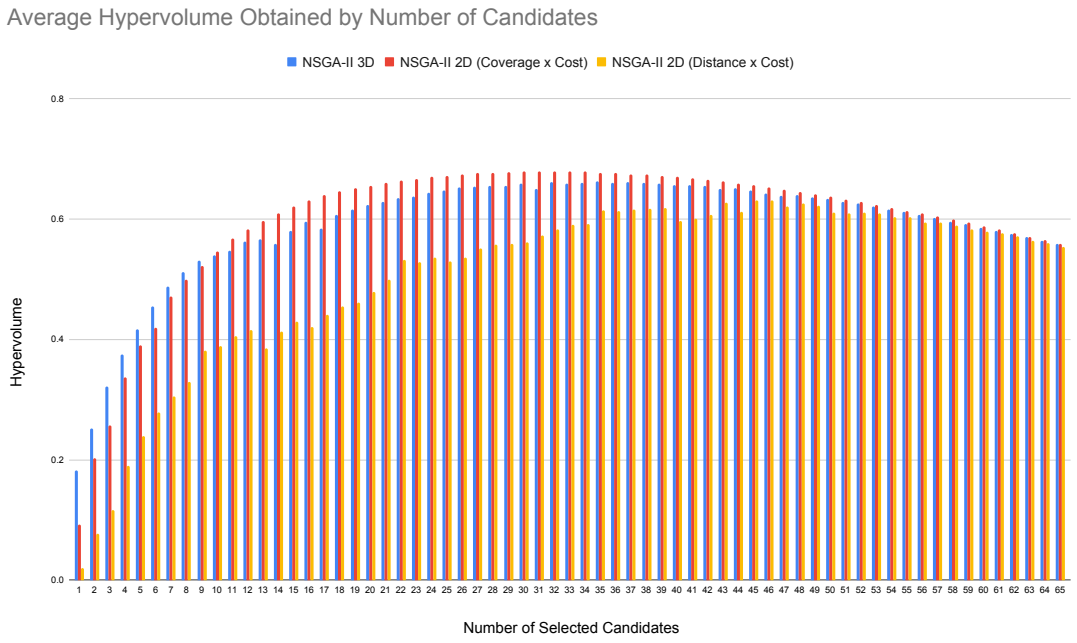
The difference in solution quality between the results depicted in Figure 6.6 obtained with NSGA-II using two objectives and three objectives raises pertinent questions about the effectiveness of the latter. It is a well-established fact that increasing the number of objectives in a problem can hinder the evolutionary process, as it leads to a higher frequency of nondominated solutions in the population, subsequently reducing selection pressure (Giagkiozis; Fleming, 2015). Nevertheless, a problem with three objectives should, in theory, remain suitable for NSGA-II.

For attesting the suitability, we can look to the hypervolume of the solutions. Figure 6.8 illustrates the average hypervolume obtained by the number of selected candidates in each approach. The pattern aligns with that observed in Figure 6.6. Beyond nine selected candidates, NSGA-II with three objectives, which is labeled as NSGA-II 3D in the plot, exhibits lower hypervolume compared to its counterpart focused on population coverage. The NSGA-II approach emphasizing distance optimization yields solutions with the lowest hypervolumes on average. Interestingly, NSGA-II 3D attains the best average

hypervolumes from one to nine candidates.

The focus on optimizing the distance objective leads to solutions with lower hypervolumes. Therefore, the method optimizing the three objectives finds solutions with lower HV when it focuses on the distance objective. The solutions found are not necessarily worse than the one found by NSGA-II focusing on coverage. They just optimized more the distance, harming the quality of coverage, which weights more in the HV metric. Indeed, if we look at the results of the supplementary material, we see that the hypervolume cannot fully capture the differences among the solutions.

Figure 6.8: Average maximum hypervolume comparison of three different approaches: NSGA-II with the three objectives; NSGA-II with two objectives (Accessibility and Cost); NSGA-II with two objectives (Distance and Cost).



Source: Elaborated by the author. 2023.

Table 6.4 exposes the objectives and hypervolume found by the different methods, fixing the number of selected candidates as 14. The values for all costs can be found in the supplementary material. Besides the optimization with three objectives finding lower hypervolumes than the optimization focusing on coverage, it found a middle term of accessibility and distance objectives. While naturally, the other methods found the best values for their respective focus. We can conclude that NSGA-II is effectively optimizing all objectives. However, the distance objective has less weight to the HV metric since the worst scenario distance is much higher than from the solutions found by the optimization.

Table 6.4: Comparison of the solutions with 14 candidates using different methods.

Method	Cost	Accessibility	Distance	Hypervolume
NSGA-II 3D	14	0.613 ± 0.075	5362.281 ± 275.955	0.558 ± 0.066
NSGA-II (Coverage)	14	**0.676** ± 0.004	5604.377 ± 66.156	**0.610** ± 0.004
NSGA-II (Distance)	14	0.444 ± 0.118	**4837.636** ± 47.256	0.413 ± 0.110

6.4 Final Remarks

In this chapter, we presented and discussed the results of the proposed experiments. First, we analyzed the actual scenario of Rio Grande do Sul regarding the stroke center network. From the perspective of our metrics, we have a poor stroke treatment catchment in the state. We found that the current network covers 63.79% of the population in adequate time, with most of the acute stroke centers located in the metropolitan region. The uncovered population is distant from treatment, with approximately 10% of the population more than 135 minutes away from the nearest ASC. When considering all demand points with equal importance, the situation is more dramatic. Only 20.86% of the demand points are covered adequately.

After exposing the current scenario, we analyzed the same metrics when we selected potential stroke centers to be ASCs. Since we had more than a hundred candidates, we used NSGA-II to select the best combination of selected candidates to improve the current network. The results showed great room for improvement in the current ASC network. With additional 38 candidates, the accessibility grows 25%, jumping to 88.99%. The average distance on this configuration diminishes from 101 to 49 minutes.

Furthermore, we proposed a scenario where we reset the current network to locate the ASCs from zero. With the new network, we achieved much better results for the evaluated objectives than the current network. The results strongly support the employment of FLPs in this context. In general, the experiments suggested that modeling as a three-objectives optimization problem is a good choice. NSGA-II can find high-quality solutions with balanced objectives, especially when locating a small number of candidates. However, NSGA-II struggled to consistently find solutions with the best HVs when more than nine candidates were selected.

7 CONCLUSION

Stroke stands as the second most prominent cause of global disability, a challenge that's especially pronounced in low- and middle-income countries. Through the assistance of GIS tools and access to public datasets, our research has revealed that Rio Grande do Sul is no exception to this underprivileged scenario. Currently, in Rio Grande do Sul (RS), approximately 63.79% of the population is covered by at least one ASC within a 45-minute drive, with an average travel time of 101.4 minutes to reach the nearest ASC. This revelation underscores the pressing necessity for systematically enhancing the ASC network within the state.

The main contribution of our work, therefore, is a new comprehensive methodology that not only identifies the deficiency in acute stroke treatment accessibility but also provides a viable course to address this issue. Our approach paves the way for establishing an enhanced and reliable ASC network in the state based on three objectives: accessibility, distance, and cost. Unlike approaches built on complex abstractions, which can result in overly specific models detached from the real world, our methodology is grounded in a realistic and straightforward framework. It relies on classical problems that, despite involving minimal variables, can provide decision-makers with powerful insights. Furthermore, the use of open-source resources in our proposed model means it can be easily adapted and applied to virtually any region across the globe. All that is needed is access to the following data:

- The OpenStreetMap road network of the region to compute the traveling times
- The Kontur's Global Population Density dataset
- The latitude/longitude coordinates of the region's current stroke centers and potential new stroke centers

While there is certainly room for improvement in NSGA-II to better handle the three proposed objectives, our model still yielded satisfactory results. Furthermore, our findings demonstrated that as the number of candidates to be placed increased, NSGA-II consistently outperformed the greedy heuristic regarding coverage, even when tackling all three objectives simultaneously.

An explicit limitation of this model is the absence of capacity attribution to the hospitals and the oversight regarding the potential advantages of hospital redundancy within the same covered region. Healthcare networks often thrive on multiple facilities in

an area to meet the demand effectively rather than relying on a single centralized facility. Future directions for research involve the development of a multi-objective model that introduces new objectives or constraints explicitly aiming to enhance network redundancy and adaptability. This direction could address the real-world need for multiple accessible ASCs in metropolitan regions like Porto Alegre's. Moreover, additional variables such as stroke incidence and prevalence of stroke risk factors can be included using public healthcare and demographic data.

Advanced strategies should be employed to address future directions in this model. It is well-known that hybridizing metaheuristics with problem-specific and local-search components improves the quality of the findings (Gandibleux; Ehrgott, 2005). Hence, it is fundamental to leverage problem characteristics such as the spatial distribution of candidates in the region. Employing clustering techniques can help reduce the number of decision variables in the problem (Saeidian et al., 2018) or facilitate the creation of local-search operators that capitalize on the spatial similarities among solutions (Corrêa; Chaves; Lorena, 2007).

REFERENCES

Ahmadi-Javid, A.; Seyedi, P.; Syam, S. S. A survey of healthcare facility location. **Computers & Operations Research**, v. 79, p. 223–263, 2017.

Akgün, I.; Erdal, H. Solving an ammunition distribution network design problem using multi-objective mathematical modeling, combined ahp-topsis, and gis. **Computers & Industrial Engineering**, v. 129, p. 512–528, 2019.

Almeida, P. F. de et al. Provision of specialized care in remote rural municipalities of the brazilian semi-arid region. **Rural and Remote Health**, Townsville QLD., v. 21, n. 4, p. 1–15, 2021.

Anderson, J.; Sarkar, D.; Palen, L. Corporate editors in the evolving landscape of openstreetmap. **ISPRS International Journal of Geo-Information**, v. 8, n. 5, 2019.

Bai, X.; Chin, K.-S.; Zhou, Z. A bi-objective model for location planning of electric vehicle charging stations with gps trajectory data. **Computers & Industrial Engineering**, v. 128, 2019.

Beheshtifar, S.; Alimoahmmadi, A. A multiobjective optimization approach for location-allocation of clinics. **International Transactions in Operational Research**, v. 22, n. 2, p. 313–328, 2014. Publisher: Wiley.

Behzadian, M. et al. A state-of the-art survey of topsis applications. **Expert Systems with Applications**, v. 39, n. 17, p. 13051–13069, 2012.

Bhaduri, B. et al. LandScan USA: a high-resolution geospatial and temporal modeling approach for population distribution and dynamics. **GeoJournal**, v. 69, n. 1-2, p. 103–117, 2007.

Blank, J.; Deb, K. Pymoo: Multi-objective optimization in python. **IEEE Access**, v. 8, p. 89497–89509, 2020.

Church, R.; ReVelle, C. The maximal covering location problem. **Papers of the Regional Science Association**, Taylor & Francis, v. 32, n. 1, p. 101–118, 1974.

Cooper, L. Location-allocation problems. **Operations Research**, INFORMS, v. 11, n. 3, p. 331–343, 1963.

Cordeau, J.-F.; Furini, F.; Ljubić, I. Benders decomposition for very large scale partial set covering and maximal covering location problems. **European Journal of Operational Research**, v. 275, n. 3, p. 882–896, 2019.

Corrêa, F. de A.; Chaves, A. A.; Lorena, L. A. N. Hybrid heuristics for the probabilistic maximal covering location-allocation problem. **Operational Research**, v. 7, n. 3, p. 323–343, 2007.

Daskin, M. S.; Owen, S. H. Two new location covering problems: The partial p-center problem and the partial set covering problem. **Geographical Analysis**, v. 31, n. 3, p. 217–235, 1999.

Deb, K. Multi-objective optimisation using evolutionary algorithms: An introduction. In: WANG, L.; NG, A. H. C.; DEB, K. (Ed.). **Multi-objective Evolutionary Optimisation for Product Design and Manufacturing**. London: Springer London, 2011. p. 3–34.

Deb, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **IEEE Transactions on Evolutionary Computation**, v. 6, n. 2, p. 182–197, 2002.

Del Ser, J. et al. Bio-inspired computation: Where we stand and what's next. **Swarm and Evolutionary Computation**, v. 48, p. 220–250, 2019.

Deng, Y.; Zhang, Y.; PAN, J. Optimization for Locating Emergency Medical Service Facilities: A Case Study for Health Planning from China. **Risk Management and Healthcare Policy**, p. 1791—1802, 2021.

Farahani, R. Z.; Seifi, M. S.; Asgari, N. Multiple criteria facility location problems: A survey. **Applied Mathematical Modelling**, v. 34, n. 7, p. 1689–1709, 2010.

Ferrari, J. et al. Experience from today for the stroke care of the future. **The CNS Journal**, Oruen, v. 4, n. 1, p. 84–91, 2018.

Fonseca, C.; Paquete, L.; Lopez-Ibanez, M. An improved dimension-sweep algorithm for the hypervolume indicator. In: **IEEE International Conference on Evolutionary Computation**. Vancouver, BC, Canada: IEEE, 2006. p. 1157–1163.

Galvão, R. D. Uncapacitated facility location problems: contributions. **Pesquisa Operacional**, v. 24, n. 1, p. 7–38, 2004.

Gandibleux, X.; Ehrgott, M. 1984-2004 – 20 years of multiobjective metaheuristics. but what about the solution of combinatorial problems with multiple objectives? In: COELLO, C. A. C.; AGUIRRE, A. H.; ZITZLER, E. (Ed.). **Evolutionary Multi-Criterion Optimization**. Berlin, Heidelberg: Springer, 2005. p. 33–46.

García, S.; Marín, A. Covering location problems. In: LAPORTE, G.; NICKEL, S.; SALDANHA DA GAMA, F. (Ed.). **Location Science**. Cham, Switzerland: Springer International Publishing, 2019. p. 99–119.

Gaynor, E. et al. Ambulance waiting and associated work flow improvement strategies: a pilot study to improve door-in-door-out time for thrombectomy patients in a primary stroke center. **Journal of NeuroInterventional Surgery**, v. 14, n. 6, p. 573–576, 2022.

Giagkiozis, I.; Fleming, P. Methods for multi-objective optimization: An analysis. **Information Sciences**, v. 293, p. 338–350, 2015.

Hakimi, S. L. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. **Operations Research**, v. 13, n. 3, p. 462–475, 1965.

Hammond, G. et al. Urban-Rural inequities in acute stroke care and In-Hospital mortality. **Stroke**, United States, v. 51, n. 7, p. 2131–2138, 2020.

Herfort, B. et al. The evolution of humanitarian mapping within the openstreetmap community. **Scientific Reports**, v. 11, n. 1, p. 3037, 2021.

Hussain, K. et al. Metaheuristic research: a comprehensive survey. **Artificial Intelligence Review**, v. 52, n. 4, p. 2191–2233, 2019.

Karasakal, E.; SILAV, A. A multi-objective genetic algorithm for a bi-objective facility location problem with partial coverage. **TOP**, v. 24, n. 1, p. 206–232, 2016.

Karatas, M.; Yakici, E. An iterative solution approach to a multi-objective facility location problem. **Applied Soft Computing**, v. 62, p. 272–287, 2018.

Kariv, O.; Hakimi, S. L. An Algorithmic Approach to Network Location Problems. II: The p-Medians. **SIAM Journal on Applied Mathematics**, v. 37, n. 3, p. 539–560, 1979.

Kaveh, M.; Mesgari, M. S. Improved biogeography-based optimization using migration process adjustment: An approach for location-allocation of ambulances. **Computers & Industrial Engineering**, v. 135, p. 800–813, 2019.

Kayola, G. et al. Stroke rehabilitation in low- and Middle-Income countries: Challenges and opportunities. **American Journal of Physical Medicine & Rehabilitation**, v. 102, n. 2S Suppl 1, p. S24–S32, 2023.

Khan, M. M.-U.-H.; Vaezi, M.; Kumar, A. Optimal siting of solid waste-to-value-added facilities through a gis-based assessment. **Science of The Total Environment**, v. 610-611, p. 1065–1075, 2018.

Konstantakopoulos, G. D.; Gayialis, S. P.; Kechagias, E. P. Vehicle routing problem and related algorithms for logistics distribution: a literature review and classification. **Operational Research**, v. 22, n. 3, p. 2033–2062, 2022.

Laporte, G.; Nickel, S.; Saldanha da Gama, F. **Location Science**. 2nd. ed. Cham, Switzerland: Springer International Publishing, 2019. XVII, 767 p. (Mathematics and Statistics).

Leira, E. C. et al. Primary stroke centers should be located using maximal coverage models for optimal access. **Stroke**, United States, v. 43, n. 9, p. 2417–2422, 2012.

Li, M.; Chen, T.; Yao, X. How to evaluate solutions in pareto-based search-based software engineering: A critical review and methodological guidance. **IEEE Transactions on Software Engineering**, v. 48, n. 5, p. 1771–1799, 2022.

Li, M.; Yao, X. Quality evaluation of solution sets in multiobjective optimisation: A survey. **ACM Comput. Surv.**, New York, NY, USA, v. 52, n. 2, 2019.

Li, X. et al. Survey of integrated flexible job shop scheduling problems. **Computers & Industrial Engineering**, v. 174, p. 108786, 2022.

Liang, J. et al. A Survey on Evolutionary Constrained Multiobjective Optimization. **IEEE Transactions on Evolutionary Computation**, v. 27, n. 2, p. 201–221, 2023.

Liu, Q. et al. Multi-objective metaheuristics for discrete optimization problems: A review of the state-of-the-art. **Applied Soft Computing**, v. 93, p. 106382, 2020.

Luke, S. **Essentials of Metaheuristics**. second. USA: Lulu, 2013. 242 p.

McLafferty, S.; Cromley, E. **GIS and Public Health**. Second. USA: The Guildford Press, 2012. 503 p.

McNamara, L. et al. An application and framework for evaluating emergency department networks using location analysis and geographic information systems. **Computers & Industrial Engineering**, v. 149, p. 106766, 2020.

Miettinen, K. **Nonlinear Multiobjective Optimization**. 1. ed. New York, NY: Springer, 1998. XXI, 298 p. (International Series in Operations Research & Management Science).

Mishra, S. et al. Geo-spatial site suitability analysis for development of health care units in rural india: Effects on habitation accessibility, facility utilization and zonal equity in facility distribution. **Journal of Transport Geography**, v. 78, p. 135–149, 2019.

Murray, A. T. et al. Commercial gis location analytics: capabilities and performance. **International Journal of Geographical Information Science**, v. 33, n. 5, p. 1106–1130, 2019.

Neis, P.; Zipf, A. Openrouteservice.org is three times "open": Combining opensource, openls and openstreetmaps. p. 248–251, 2008.

Neumann-Haefelin, T. et al. Zertifizierungskriterien für stroke-units in deutschland: Update 2022. **DGNeurologie**, Springer Medizin, v. 6, p. 438–446, 2021.

Papadimitriou, C.; Steiglitz, K. Combinatorial optimization: Algorithms and complexity. **IEEE Transactions on Acoustics, Speech, and Signal Processing**, v. 32, n. 6, p. 1258–1259, 1982.

Paramasivam, C.; Venkatramanan, S. Chapter 3 - An Introduction to Various Spatial Analysis Techniques. In: VENKATRAMANAN, S.; PRASANNA, M. V.; CHUNG, S. Y. (Ed.). **GIS and Geostatistical Techniques for Groundwater Science**. United Kingdom: Elsevier, 2019. p. 23–30.

Peres, F.; Castelli, M. Combinatorial optimization problems and metaheuristics: Review, challenges, design, and development. **Applied Sciences**, v. 11, n. 14, 2021.

Powers, W. J. Acute ischemic stroke. **New England Journal of Medicine**, v. 383, n. 3, p. 252–260, 2020.

Resende, M. G. C. Biased random-key genetic algorithms with applications in telecommunications. **TOP**, v. 20, n. 1, p. 130–153, 2011.

Saeidian, B. et al. Optimized location-allocation of earthquake relief centers using pso and aco, complemented by gis, clustering, and topsis. **ISPRS International Journal of Geo-Information**, v. 7, n. 8, 2018.

Sahr, K.; White, D.; Kimerling, A. J. Geodesic discrete global grid systems. **Cartography and Geographic Information Science**, v. 30, n. 2, p. 121–134, 2003.

Shi, J. et al. A bi-objective multi-period facility location problem for household e-waste collection. **International Journal of Production Research**, v. 58, n. 2, p. 526–545, 2020.

Srinivas, N.; Deb, K. Multiobjective optimization using nondominated sorting in genetic algorithms. **Evolutionary Computation**, v. 2, n. 3, p. 221–248, 1994.

Sörensen, K. Metaheuristics-the metaphor exposed. **International Transactions in Operational Research**, v. 22, n. 1, p. 3–18, 2013. Publisher: Wiley.

Taiwo, O. J. Maximal covering location problem (mclp) for the identification of potential optimal covid-19 testing facility sites in nigeria. **African Geographical Review**, v. 40, n. 4, p. 395–411, 2021.

Talbi, E.-G. **Metaheuristics: From Design to Implementation**. Hoboken, New Jersey, US: Wiley Publishing, 2009. 593 p.

Tardos, J. K. E. **Algorithm Design**. USA: Addison-Wesley Longman Publishing Co., Inc., 2005. 838 p.

Toregas, C. et al. The location of emergency service facilities. **Operations Research**, v. 19, n. 6, p. 1363–1373, 1971.

Trivedi, A. et al. A Survey of Multiobjective Evolutionary Algorithms Based on Decomposition. **IEEE Transactions on Evolutionary Computation**, v. 21, n. 3, p. 440–462, 2017.

VERMA, S.; PANT, M.; SNASEL, V. A Comprehensive Review on NSGA-II for Multi-Objective Combinatorial Optimization Problems. **IEEE Access**, v. 9, p. 57757–57791, 2021.

Verter, V. Uncapacitated and capacitated facility location problems. In: EISELT, H. A.; MARIANOV, V. (Ed.). **Foundations of Location Analysis**. New York, NY: Springer US, 2011. p. 25–37.

Vogel, N. E. et al. Optimized density and locations of stroke centers for improved cost effectiveness of mechanical thrombectomy in patients with acute ischemic stroke. **Journal of NeuroInterventional Surgery**, 2023.

Wang, S.; Ma, S. Efficient methods for a bi-objective nursing home location and allocation problem: A case study. **Applied Soft Computing**, v. 65, p. 280–291, 2018.

Wieczorek, W. F.; Delmerico, A. M. Geographic information systems. **Comput Stat**, Germany, v. 1, n. 2, p. 167–186, 2009.

Zarate-Zapata, A.-C. et al. Analysis and location of vaccination centers in the municipality of the state of puebla, méxico using the p-median problem. In: DESCHAMPS, F. et al. (Ed.). **Proceedings of the 11th International Conference on Production Research – Americas**. Cham: Springer Nature Switzerland, 2023. p. 1–15.

Zhang, C. et al. Exact and matheuristic methods for the parallel machine scheduling and location problem with delivery time and due date. **Computers & Operations Research**, v. 147, 2022.

Zhang, Q.; LI, H. MOEA/D: A multiobjective evolutionary algorithm based on decomposition. **IEEE Transactions on Evolutionary Computation**, v. 11, n. 6, p. 712–731, 2007.

Zitzler, E.; Künzli, S. Indicator-based selection in multiobjective search. In: **Parallel Problem Solving from Nature - PPSN VIII**. Berlin, Heidelberg: Springer, 2004. p. 832–842.

Zitzler, Eckart; Laumanns, Marco; Thiele, Lothar. **SPEA2: Improving the strength pareto evolutionary algorithm**. Computer Engineering and Networks Laboratory, Swiss Federal Institute of Technology (ETH), Zurich, Switzerland, 2001.

SUPPLEMENTARY MATERIAL

Table 7.1: The best hypervolume solutions for each number of selected candidate facilities in the first scenario.

Cost	Accessibility	Distance (in seconds)	Hypervolume
1	0.666	5449.720	0.0190
2	0.687	5259.689	0.0432
3	0.697	4987.419	0.0701
4	0.708	4829.259	0.0943
5	0.721	4725.665	0.1210
6	0.732	4602.061	0.1483
7	0.748	4575.710	0.1756
8	0.756	4413.657	0.2052
9	0.762	4285.503	0.2312
10	0.771	4205.965	0.2570
11	0.780	4143.440	0.2811
12	0.786	4043.969	0.3052
13	0.790	3918.710	0.3303
14	0.796	3813.461	0.3547
15	0.800	3719.682	0.3761
16	0.806	3654.869	0.3969
17	0.815	3633.696	0.4177
18	0.822	3598.940	0.4365
19	0.827	3551.839	0.4523
20	0.831	3499.995	0.4655
21	0.836	3467.317	0.4801
22	0.841	3421.040	0.4950
23	0.844	3367.114	0.5085
24	0.849	3322.254	0.5229
25	0.853	3276.453	0.5345
26	0.857	3231.333	0.5486
27	0.860	3182.744	0.5583
28	0.864	3162.277	0.5677
29	0.867	3119.938	0.5766

30	0.871	3098.325	0.5835
31	0.874	3078.312	0.5881
32	0.877	3054.469	0.5924
33	0.880	3041.156	0.5956
34	0.881	3021.836	0.5966
35	0.884	3014.367	0.5973
36	0.886	3004.131	0.5963
37	0.887	2975.811	0.5969
38	0.890	2968.419	**0.5974**
39	0.891	2957.262	0.5943
40	0.894	2954.965	0.5935
41	0.894	2936.819	0.5905
42	0.896	2934.598	0.5879
43	0.898	2924.727	0.5846
44	0.899	2914.615	0.5815
45	0.900	2913.178	0.5757
46	0.899	2894.200	0.5697
47	0.901	2888.635	0.5671
48	0.902	2884.757	0.5607
49	0.903	2871.224	0.5561
50	0.903	2865.908	0.5490
51	0.903	2856.650	0.5435
52	0.903	2857.265	0.5352
53	0.904	2845.080	0.5299
54	0.904	2832.528	0.5242
55	0.905	2834.866	0.5160
56	0.905	2828.589	0.5088
57	0.905	2823.839	0.5017
58	0.905	2822.067	0.4936
59	0.905	2820.969	0.4854
60	0.905	2818.427	0.4774
61	0.905	2815.614	0.4694
62	0.905	2813.523	0.4612

63	0.906	2810.757	0.4533
64	0.906	2807.143	0.4453
65	0.906	2807.603	0.4368

Table 7.2: Current ASC network results presented in Figure 6.4

Cost	Accessibility	Distance (in seconds)	Hypervolume
33	0.638	6084	0.4849
34	0.667	5457.042	0.5154
35	0.687	5259.689	0.5304
36	0.697	4987.419	0.5398
37	0.713	4961.068	0.5479
38	0.727	4857.474	0.5554
39	0.738	4699.314	0.5620
40	0.748	4575.710	0.5678
41	0.756	4413.657	0.5715
42	0.765	4325.776	0.5750
43	0.774	4263.251	0.5776
44	0.780	4143.440	0.5796
45	0.786	4043.969	0.5807
46	0.793	3997.009	0.5815
47	0.799	3897.538	0.5824
48	0.805	3792.288	0.5827
49	0.810	3733.471	0.5824
50	0.815	3637.207	0.5821
51	0.822	3598.940	0.5819
52	0.827	3551.839	0.5805
53	0.832	3528.006	0.5785
54	0.836	3467.317	0.5766
55	0.841	3421.040	0.5747
56	0.845	3384.804	0.5724
57	0.850	3353.745	0.5701
58	0.855	3331.216	0.5676
59	0.857	3231.333	0.5648
60	0.861	3215.278	0.5615
61	0.865	3168.527	0.5584
62	0.868	3146.915	0.5549
63	0.871	3098.325	0.5511

64	0.874	3078.312	0.5470
65	0.877	3064.998	0.5427
66	0.880	3041.156	0.5383
67	0.881	3021.836	0.5333
68	0.884	3014.367	0.5285
69	0.886	3004.131	0.5233
70	0.888	2994.784	0.5183
71	0.890	2968.419	0.5131
72	0.892	2966.412	0.5075
73	0.894	2954.965	0.5023
74	0.895	2950.269	0.4967
75	0.896	2934.598	0.4910
76	0.898	2924.727	0.4852
77	0.899	2914.615	0.4794
78	0.900	2913.178	0.4733
79	0.900	2907.348	0.4668
80	0.902	2895.045	0.4611
81	0.902	2884.757	0.4548
82	0.903	2871.224	0.4486
83	0.903	2865.908	0.4421
84	0.904	2864.444	0.4358
85	0.903	2857.265	0.4291
86	0.904	2845.080	0.4229
87	0.904	2832.528	0.4165
88	0.905	2834.866	0.4099
89	0.905	2828.589	0.4033
90	0.905	2823.839	0.3968
91	0.905	2822.067	0.3902
92	0.905	2820.969	0.3835
93	0.905	2818.427	0.3768
94	0.905	2815.614	0.3702
95	0.905	2813.523	0.3635
96	0.906	2810.757	0.3568

97	0.906	2807.143	0.3502
98	0.906	2807.603	0.3434
99	0.906	2803.499	0.3368
100	0.906	2807.506	0.3300

Table 7.3: Reset ASC network results presented in Figure 6.4

Cost	Accessibility	Distance (in seconds)	Hypervolume
1	0.313	15546.374	0.1858
2	0.330	10711.498	0.2577
3	0.411	10129.636	0.3275
4	0.447	8481.737	0.3827
5	0.480	7699.598	0.4229
6	0.530	7855.455	0.4602
7	0.545	6547.592	0.4972
8	0.570	6242.290	0.5235
9	0.585	6005.128	0.5383
10	0.609	5965.326	0.5571
11	0.617	5635.049	0.5681
12	0.624	5558.676	0.5723
13	0.647	5510.255	0.5899
14	0.656	5073.303	0.6045
15	0.663	4939.571	0.6094
16	0.675	4818.487	0.6189
17	0.685	4667.860	0.6273
18	0.694	4485.470	0.6345
19	0.698	4481.709	0.6343
20	0.711	4526.933	0.6399
21	0.721	4305.422	0.6487
22	0.726	4203.671	0.6512
23	0.742	4258.517	0.6590
24	0.753	4146.147	0.6658
25	0.747	4089.687	0.6569
26	0.762	3935.821	0.6681
27	0.766	3854.240	0.6689
28	0.779	3783.552	0.6765
29	0.780	3781.497	0.6716
30	0.784	3679.477	0.6721
31	0.796	3658.999	0.6768

32	0.795	3608.906	0.6718
33	0.802	3634.802	0.6712
34	0.807	3670.898	0.6684
35	0.811	3526.704	0.6699
36	0.814	3440.730	0.6682
37	0.820	3423.447	0.6679
38	0.825	3382.653	0.6671
39	0.830	3306.168	0.6665
40	0.835	3314.433	0.6643
41	0.837	3302.148	0.6601
42	0.840	3279.775	0.6570
43	0.845	3239.908	0.6560
44	0.851	3179.019	0.6551
45	0.857	3236.478	0.6525
46	0.860	3174.026	0.6502
47	0.863	3115.905	0.6475
48	0.869	3114.961	0.6451
49	0.869	3075.106	0.6400
50	0.871	3071.431	0.6350
51	0.875	3048.582	0.6321
52	0.878	3028.640	0.6285
53	0.882	3047.656	0.6241
54	0.881	3003.179	0.6178
55	0.885	3004.196	0.6145
56	0.884	2995.351	0.6070
57	0.886	2975.311	0.6028
58	0.893	2989.254	0.6000
59	0.892	2951.779	0.5938
60	0.894	2974.314	0.5880
61	0.894	2929.364	0.5825
62	0.895	2927.060	0.5765
63	0.897	2926.952	0.5709
64	0.897	2938.641	0.5643

65	0.898	2910.953	0.5588
66	0.899	2899.706	0.5531
67	0.899	2911.915	0.5463
68	0.900	2884.197	0.5407
69	0.900	2906.888	0.5337
70	0.901	2873.106	0.5283
71	0.902	2884.408	0.5217
72	0.904	2859.423	0.5166
73	0.903	2868.668	0.5090
74	0.903	2863.018	0.5027
75	0.904	2854.989	0.4965
76	0.904	2867.613	0.4895
77	0.904	2838.379	0.4834
78	0.904	2845.424	0.4766
79	0.905	2834.918	0.4705
80	0.905	2823.032	0.4642
81	0.904	2839.759	0.4568
82	0.905	2819.010	0.4508
83	0.905	2817.513	0.4442
84	0.905	2826.301	0.4372
85	0.905	2815.533	0.4307
86	0.906	2812.604	0.4242
87	0.906	2812.129	0.4175
88	0.905	2816.045	0.4104
89	0.905	2820.515	0.4038
90	0.906	2805.407	0.3974
91	0.905	2815.632	0.3902
93	0.906	2800.232	0.3772
94	0.905	2808.251	0.3703
95	0.905	2807.764	0.3635
96	0.906	2802.677	0.3571
97	0.906	2802.565	0.3504
98	0.906	2795.567	0.3436

99	0.906	2799.017	0.3369
100	0.906	2806.379	0.3300

Table 7.4: Results on the NSGA-II optimizing three objectives on the second scenario.

Cost	**Accessibility**	**Distance (in seconds)**	**Hypervolume**
1	0.307 ± 0.001	15546.464 ± 0.324	0.182 ± 0.001
2	0.325 ± 0.012	10857.386 ± 609.195	0.252 ± 0.001
3	0.412 ± 0.002	10582.183 ± 102.473	0.322 ± 0.001
4	0.438 ± 0.001	8496.667 ± 19.475	0.375 ± 0.001
5	0.481 ± 0.009	8102.921 ± 363.959	0.416 ± 0.002
6	0.519 ± 0.007	7640.326 ± 184.294	0.455 ± 0.004
7	0.546 ± 0.011	7046.561 ± 313.530	0.488 ± 0.005
8	0.563 ± 0.010	6490.970 ± 348.339	0.511 ± 0.010
9	0.583 ± 0.010	6235.185 ± 256.523	0.531 ± 0.009
10	0.592 ± 0.016	6053.636 ± 199.625	0.540 ± 0.014
11	0.599 ± 0.023	5833.819 ± 247.441	0.548 ± 0.020
12	0.616 ± 0.012	5701.461 ± 283.529	0.562 ± 0.012
13	0.620 ± 0.016	5467.916 ± 149.081	0.567 ± 0.015
14	0.613 ± 0.075	5362.281 ± 275.955	0.558 ± 0.066
15	0.637 ± 0.026	5157.489 ± 127.179	0.581 ± 0.024
16	0.655 ± 0.015	5031.463 ± 165.214	0.596 ± 0.014
17	0.644 ± 0.079	4864.287 ± 157.838	0.585 ± 0.071
18	0.669 ± 0.023	4720.647 ± 118.741	0.606 ± 0.021
19	0.683 ± 0.022	4664.786 ± 179.633	0.616 ± 0.020
20	0.693 ± 0.019	4530.695 ± 151.442	0.623 ± 0.017
21	0.701 ± 0.015	4395.935 ± 108.111	0.629 ± 0.014
22	0.710 ± 0.010	4288.085 ± 134.186	0.635 ± 0.010
23	0.717 ± 0.021	4200.218 ± 122.126	0.638 ± 0.019
24	0.728 ± 0.009	4138.198 ± 148.504	0.644 ± 0.009
25	0.736 ± 0.009	4081.042 ± 139.443	0.647 ± 0.009
26	0.744 ± 0.011	3935.385 ± 95.389	0.653 ± 0.010
27	0.750 ± 0.011	3894.835 ± 69.658	0.653 ± 0.010
28	0.756 ± 0.011	3867.534 ± 101.600	0.654 ± 0.010
29	0.761 ± 0.011	3794.852 ± 66.184	0.655 ± 0.009
30	0.770 ± 0.007	3748.641 ± 58.995	0.658 ± 0.007
31	0.765 ± 0.063	3696.720 ± 58.772	0.649 ± 0.054

32	0.784 ± 0.009	3646.136 ± 51.638	0.661 ± 0.008
33	0.786 ± 0.012	3580.772 ± 40.940	0.659 ± 0.010
34	0.793 ± 0.016	3565.642 ± 51.370	0.660 ± 0.013
35	0.802 ± 0.008	3524.860 ± 55.215	0.662 ± 0.007
36	0.805 ± 0.010	3477.770 ± 46.435	0.660 ± 0.009
37	0.813 ± 0.009	3444.832 ± 59.462	0.661 ± 0.008
38	0.817 ± 0.008	3402.060 ± 38.933	0.660 ± 0.007
39	0.822 ± 0.007	3375.400 ± 48.914	0.659 ± 0.006
40	0.826 ± 0.010	3330.756 ± 41.749	0.657 ± 0.009
41	0.832 ± 0.006	3291.778 ± 33.589	0.657 ± 0.005
42	0.837 ± 0.007	3270.954 ± 41.226	0.655 ± 0.005
43	0.839 ± 0.027	3252.970 ± 39.638	0.650 ± 0.021
44	0.846 ± 0.006	3220.890 ± 34.243	0.651 ± 0.005
45	0.850 ± 0.011	3195.072 ± 36.799	0.648 ± 0.008
46	0.850 ± 0.018	3166.316 ± 22.051	0.643 ± 0.014
47	0.852 ± 0.037	3142.771 ± 26.675	0.639 ± 0.028
48	0.862 ± 0.005	3118.230 ± 22.565	0.640 ± 0.004
49	0.864 ± 0.007	3101.773 ± 29.869	0.636 ± 0.005
50	0.869 ± 0.005	3080.851 ± 23.883	0.634 ± 0.004
51	0.871 ± 0.005	3068.084 ± 23.682	0.629 ± 0.004
52	0.875 ± 0.003	3052.376 ± 20.255	0.626 ± 0.003
53	0.877 ± 0.004	3038.213 ± 24.003	0.621 ± 0.003
54	0.879 ± 0.005	3025.756 ± 15.345	0.616 ± 0.004
55	0.881 ± 0.003	3013.240 ± 14.329	0.612 ± 0.002
56	0.883 ± 0.006	3001.356 ± 18.595	0.606 ± 0.004
57	0.886 ± 0.003	2988.590 ± 13.918	0.602 ± 0.002
58	0.886 ± 0.004	2977.184 ± 12.742	0.596 ± 0.003
59	0.888 ± 0.002	2971.934 ± 14.644	0.591 ± 0.002
60	0.889 ± 0.005	2961.123 ± 14.577	0.585 ± 0.004
61	0.892 ± 0.002	2950.873 ± 13.020	0.581 ± 0.002
62	0.894 ± 0.002	2939.771 ± 11.111	0.576 ± 0.002
63	0.895 ± 0.002	2929.516 ± 12.282	0.570 ± 0.002
64	0.896 ± 0.002	2925.990 ± 14.169	0.564 ± 0.001

65	0.897 ± 0.003	2914.597 ± 10.817	0.558 ± 0.002

Table 7.5: Results on the NSGA-II optimizing two objectives (Distance x Cost)

Cost	Accessibility	Distance (in seconds)	Hypervolume
1	0.032 ± 0.030	14926.831 ± 2201.224	0.020 ± 0.018
2	0.105 ± 0.116	11910.448 ± 1002.840	0.077 ± 0.085
3	0.142 ± 0.113	9577.985 ± 445.139	0.116 ± 0.092
4	0.219 ± 0.127	8346.306 ± 324.023	0.190 ± 0.110
5	0.264 ± 0.121	7418.012 ± 257.131	0.235 ± 0.108
6	0.306 ± 0.135	6842.214 ± 160.267	0.278 ± 0.123
7	0.326 ± 0.124	6468.303 ± 202.750	0.299 ± 0.115
8	0.357 ± 0.141	6107.558 ± 110.847	0.330 ± 0.130
9	0.411 ± 0.114	5811.360 ± 98.401	0.382 ± 0.106
10	0.417 ± 0.113	5569.894 ± 56.255	0.388 ± 0.105
11	0.435 ± 0.107	5363.893 ± 48.561	0.405 ± 0.100
12	0.446 ± 0.102	5170.940 ± 47.949	0.416 ± 0.095
13	0.413 ± 0.122	4993.383 ± 45.913	0.385 ± 0.114
14	0.444 ± 0.118	4837.636 ± 47.256	0.413 ± 0.110
15	0.462 ± 0.121	4699.516 ± 35.895	0.429 ± 0.112
16	0.454 ± 0.119	4579.662 ± 33.711	0.420 ± 0.110
17	0.478 ± 0.116	4464.348 ± 26.956	0.441 ± 0.107
18	0.494 ± 0.117	4362.569 ± 24.799	0.454 ± 0.108
19	0.503 ± 0.120	4261.819 ± 22.614	0.461 ± 0.110
20	0.524 ± 0.110	4172.053 ± 21.513	0.478 ± 0.101
21	0.550 ± 0.096	4089.617 ± 21.303	0.499 ± 0.087
22	0.590 ± 0.069	4010.646 ± 18.386	0.533 ± 0.062
23	0.588 ± 0.092	3945.870 ± 17.323	0.528 ± 0.083
24	0.599 ± 0.094	3883.644 ± 16.674	0.535 ± 0.084
25	0.597 ± 0.109	3827.753 ± 18.033	0.530 ± 0.096
26	0.607 ± 0.098	3771.350 ± 15.614	0.536 ± 0.086
27	0.626 ± 0.091	3715.831 ± 12.866	0.549 ± 0.080
28	0.640 ± 0.084	3660.942 ± 12.949	0.558 ± 0.073
29	0.645 ± 0.103	3611.716 ± 14.288	0.559 ± 0.089
30	0.651 ± 0.098	3564.932 ± 15.192	0.561 ± 0.085
31	0.670 ± 0.096	3520.696 ± 13.354	0.573 ± 0.082

32	0.687 ± 0.091	3482.099 ± 16.772	0.583 ± 0.077
33	0.701 ± 0.088	3448.646 ± 15.800	0.591 ± 0.074
34	0.707 ± 0.091	3414.323 ± 17.280	0.592 ± 0.076
35	0.741 ± 0.059	3382.921 ± 16.132	0.615 ± 0.049
36	0.745 ± 0.066	3353.821 ± 19.292	0.613 ± 0.055
37	0.753 ± 0.060	3324.744 ± 15.714	0.616 ± 0.049
38	0.761 ± 0.058	3298.284 ± 16.803	0.617 ± 0.047
39	0.769 ± 0.061	3276.360 ± 16.881	0.618 ± 0.049
40	0.748 ± 0.084	3252.890 ± 17.362	0.597 ± 0.067
41	0.759 ± 0.083	3229.207 ± 15.207	0.600 ± 0.066
42	0.773 ± 0.080	3204.426 ± 14.437	0.606 ± 0.063
43	0.807 ± 0.038	3182.333 ± 16.460	0.628 ± 0.030
44	0.794 ± 0.069	3163.746 ± 14.899	0.612 ± 0.053
45	0.826 ± 0.011	3141.768 ± 13.766	0.631 ± 0.008
46	0.834 ± 0.011	3121.920 ± 13.835	0.631 ± 0.008
47	0.827 ± 0.058	3105.987 ± 14.924	0.621 ± 0.044
48	0.842 ± 0.012	3087.151 ± 11.657	0.626 ± 0.009
49	0.844 ± 0.018	3070.839 ± 13.707	0.622 ± 0.013
50	0.838 ± 0.048	3056.695 ± 12.052	0.611 ± 0.035
51	0.843 ± 0.042	3045.019 ± 10.527	0.609 ± 0.030
52	0.853 ± 0.017	3030.942 ± 11.751	0.610 ± 0.012
53	0.860 ± 0.013	3017.665 ± 11.914	0.609 ± 0.009
54	0.859 ± 0.015	3005.705 ± 9.757	0.602 ± 0.011
55	0.868 ± 0.008	2995.144 ± 9.745	0.603 ± 0.006
56	0.864 ± 0.016	2984.201 ± 8.300	0.594 ± 0.011
57	0.873 ± 0.009	2973.627 ± 8.488	0.594 ± 0.006
58	0.876 ± 0.010	2963.144 ± 8.645	0.589 ± 0.007
59	0.875 ± 0.012	2952.789 ± 7.303	0.583 ± 0.008
60	0.879 ± 0.007	2941.925 ± 6.766	0.579 ± 0.005
61	0.884 ± 0.008	2930.892 ± 5.546	0.576 ± 0.005
62	0.888 ± 0.007	2923.146 ± 7.960	0.572 ± 0.005
63	0.885 ± 0.007	2915.174 ± 7.264	0.564 ± 0.005
64	0.889 ± 0.007	2907.446 ± 6.321	0.560 ± 0.004

65	0.890 ± 0.008	2899.812 ± 7.227	0.554 ± 0.005

Table 7.6: Results on the NSGA-II optimizing two objectives (Accessibility x Cost)

Cost	Accessibility	Distance (in seconds)	Hypervolume
1	0.161 ± 0.117	15997.461 ± 707.425	0.093 ± 0.069
2	0.319 ± 0.114	14167.960 ± 964.462	0.203 ± 0.070
3	0.378 ± 0.111	12786.303 ± 1943.116	0.257 ± 0.069
4	0.456 ± 0.021	11432.316 ± 1231.908	0.337 ± 0.014
5	0.496 ± 0.006	10097.266 ± 883.627	0.391 ± 0.017
6	0.507 ± 0.076	9019.684 ± 815.162	0.418 ± 0.063
7	0.549 ± 0.012	7979.979 ± 462.955	0.471 ± 0.016
8	0.574 ± 0.008	7534.390 ± 444.258	0.499 ± 0.014
9	0.597 ± 0.005	7266.122 ± 486.812	0.522 ± 0.014
10	0.616 ± 0.004	6728.365 ± 613.237	0.546 ± 0.016
11	0.633 ± 0.004	6283.110 ± 397.969	0.568 ± 0.011
12	0.648 ± 0.004	6060.652 ± 133.065	0.582 ± 0.005
13	0.662 ± 0.004	5838.519 ± 113.221	0.596 ± 0.004
14	0.676 ± 0.004	5604.377 ± 66.156	0.610 ± 0.004
15	0.688 ± 0.004	5416.447 ± 83.339	0.620 ± 0.004
16	0.699 ± 0.003	5206.409 ± 84.772	0.631 ± 0.004
17	0.709 ± 0.004	5010.127 ± 95.002	0.640 ± 0.004
18	0.718 ± 0.003	4933.962 ± 89.114	0.646 ± 0.004
19	0.728 ± 0.003	4845.006 ± 67.978	0.651 ± 0.004
20	0.735 ± 0.004	4768.905 ± 60.765	0.655 ± 0.004
21	0.743 ± 0.003	4662.627 ± 79.548	0.660 ± 0.004
22	0.751 ± 0.002	4565.769 ± 81.297	0.664 ± 0.003
23	0.758 ± 0.002	4495.760 ± 114.856	0.667 ± 0.004
24	0.765 ± 0.002	4389.249 ± 78.616	0.670 ± 0.003
25	0.771 ± 0.002	4330.877 ± 83.347	0.672 ± 0.004
26	0.777 ± 0.002	4230.710 ± 72.710	0.674 ± 0.003
27	0.784 ± 0.002	4143.924 ± 70.560	0.677 ± 0.003
28	0.789 ± 0.003	4104.070 ± 63.681	0.677 ± 0.003
29	0.795 ± 0.003	4038.672 ± 68.663	0.678 ± 0.004
30	0.800 ± 0.003	3962.229 ± 58.697	0.679 ± 0.003
31	0.805 ± 0.003	3884.619 ± 59.069	0.679 ± 0.003

32	0.811 ± 0.003	3830.093 ± 67.246	0.679 ± 0.003
33	0.816 ± 0.003	3784.568 ± 55.029	0.679 ± 0.003
34	0.821 ± 0.003	3727.978 ± 67.850	0.679 ± 0.003
35	0.825 ± 0.002	3687.341 ± 65.343	0.677 ± 0.003
36	0.830 ± 0.003	3639.980 ± 55.322	0.676 ± 0.003
37	0.834 ± 0.003	3601.772 ± 53.055	0.674 ± 0.003
38	0.838 ± 0.003	3534.507 ± 54.497	0.673 ± 0.003
39	0.842 ± 0.003	3485.420 ± 63.134	0.672 ± 0.004
40	0.846 ± 0.002	3447.426 ± 40.201	0.670 ± 0.002
41	0.850 ± 0.002	3405.872 ± 38.305	0.667 ± 0.002
42	0.853 ± 0.002	3366.999 ± 42.719	0.665 ± 0.002
43	0.856 ± 0.002	3335.919 ± 34.240	0.662 ± 0.002
44	0.860 ± 0.003	3297.278 ± 37.272	0.659 ± 0.003
45	0.863 ± 0.002	3272.768 ± 37.382	0.656 ± 0.002
46	0.866 ± 0.002	3250.453 ± 44.066	0.653 ± 0.002
47	0.868 ± 0.002	3227.720 ± 46.715	0.649 ± 0.002
48	0.871 ± 0.002	3198.652 ± 29.527	0.645 ± 0.002
49	0.873 ± 0.002	3172.849 ± 37.878	0.641 ± 0.002
50	0.876 ± 0.003	3146.074 ± 31.192	0.637 ± 0.003
51	0.878 ± 0.003	3126.201 ± 33.107	0.633 ± 0.003
52	0.880 ± 0.002	3112.231 ± 27.643	0.628 ± 0.002
53	0.882 ± 0.002	3095.160 ± 20.551	0.623 ± 0.002
54	0.884 ± 0.003	3077.579 ± 28.972	0.619 ± 0.002
55	0.886 ± 0.003	3061.883 ± 27.375	0.614 ± 0.002
56	0.888 ± 0.002	3051.616 ± 21.933	0.609 ± 0.002
57	0.890 ± 0.002	3026.444 ± 20.816	0.604 ± 0.002
58	0.891 ± 0.002	3014.060 ± 13.728	0.599 ± 0.001
59	0.893 ± 0.002	2999.697 ± 11.126	0.594 ± 0.001
60	0.894 ± 0.001	2992.029 ± 13.523	0.588 ± 0.001
61	0.896 ± 0.001	2978.828 ± 13.189	0.582 ± 0.001
62	0.896 ± 0.001	2969.573 ± 12.750	0.577 ± 0.001
63	0.897 ± 0.001	2965.302 ± 14.416	0.570 ± 0.001
64	0.898 ± 0.001	2953.738 ± 10.365	0.565 ± 0.001

65	0.899 ± 0.001	2944.168 ± 9.016	0.559 ± 0.001

APPENDIX A - RESUMO EXPANDIDO

O acidente vascular cerebral (AVC) agudo é uma das principais causas de sequelas graves em nível mundial, particularmente nos países de baixa e média renda, os quais são responsáveis por 87% desses números. O acesso a tratamento em uma janela de tempo adequada após a incidência de um AVC diminui significativamente a probabilidade de ocorrência de sequelas graves em pacientes. Considerando a baixa cobertura de tratamento em países de baixa e média renda, é imperativo iniciar iniciativas para aumentar o número de centros de tratamento de AVC de forma eficiente. Métodos computacionais encontram nos problemas de localização de instalações (FLPs) a abstração que permite a otimização da atribuição de recursos a pontos de procura numa rede. Diversas aplicações, incluindo a localização de departamentos de emergência e estações de ambulâncias, podem ser modelados como FLPs.

Propomos uma metodologia que pode ser aplicada ao planejamento de uma rede de centros para tratamento de AVC em uma região a fim de obter, com custo mínimo, uma cobertura de tratamento que maximiza a população atendida em tempo adequado, minimiza o tempo de deslocamento médio da população até o centro de AVC mais próximo. Definimos que centros de AVC são aqueles hospitais que possuem protocolo ativo para tratamento do AVC isquêmico agudo, o qual é o tipo de AVC agudo com maior incidência na população. Com a intenção de ampliar a disponibilidade de tratamento, colaboramos com especialistas da área para identificar hospitais que, embora disponham de infraestrutura mínima para tratar AVC, não o realizam devido à falta de especialistas. Esses hospitais são considerados potenciais centros de AVC em nossa metodologia. No contexto dos FLPs, é imperativo definir candidatos a pontos de oferta e demanda. No nosso modelo, os potenciais centros de AVC compõem o conjunto de candidatos. Os pontos de demanda são estabelecidos pela população regional, dividida em hexágonos de aproximadamente 1km². Utilizando ferramentas de Sistemas de Informação Geográfica (GIS), calculamos o tempo de deslocamento de cada ponto de demanda para cada centro de AVC existente e potencial. Com esses dados em mãos, otimizamos o problema empregando o algoritmo *Non-dominated Sorting Genetic Algorithm II* (NSGA-II), considerando os três objetivos que consideramos essenciais para o eficiente planejamento de uma rede de centros de AVC em uma região.

Analisando a literatura, observamos uma escassez de trabalhos avaliando a cobertura de tratamento de redes de centros de AVC. Inclusive, não há estudos do tipo no

Brasil. Portanto, propomos um estudo de caso no estado de Rio Grande do Sul, Brasil. Nesse estudo, tivemos como objetivo avaliar:

1. A cobertura de tratamento atual no Rio Grande do Sul e as melhorias potenciais descritas pelo método de otimização com base em nossa metodologia;
2. O quão boa a rede otimizada em relação à rede atual do estado
3. Se o algoritmo NSGA-II é capaz de encontrar soluções boas para o problema descrito

Para o objetivo (1), elaboramos um cenário de experimentos que considera os centros atuais de AVC no Rio Grande do Sul. Portanto, todos centros potenciais adicionados à rede durante a otimização, são concatenados a uma solução que contém os centros atuais. Medimos a cobertura de tratamento atual no Rio Grande do Sul, e com 33 centros de AVC no estado, apenas $63,79\%$ da população está localizada a uma distância adequada de um centro. Além disso, o tempo médio de deslocamento da população para o centro de AVC mais próximo é de $101,4$ minutos. Após rodar a otimização, encontramos que é possível aprimorar esses números para uma cobertura de $88,99\%$ da população e tempo médio de deslocamento de 49 minutos selecionando 38 potenciais centros de AVC no estado para disponibilizarem tratamento.

Em relação ao objetivo (2), comparamos a otimização do primeiro cenário, que inclui os centros de AVC atuais no estado, com um cenário de otimização que não necessariamente irá incluir esses centros. Esse cenário considera como candidatos tanto os centros de AVC atuais como os potenciais. Portanto, é um cenário que refaz a rede de centros de AVC do estado do zero. Os resultados mostraram que com os mesmo número de centros de hoje em dia, é possível obter uma configuração que cobre $80,21\%$ da população, com tempo médio de deslocamento de $60,6$ minutos. Para obter $89,3\%$ de cobertura e $49,8$ minutos de tempo de deslocamento médio, precisariam de 58 centros; no primeiro cenário 71 centros foram necessários para obter valores parecidos. É evidente pelos resultados obtidos que um planejamento sistemático de localização de centros de AVC pode ser muito benéfico à população.

Por fim, o terceiro objetivo do estudo foi avaliar se o algoritmo NSGA-II providenciava uma otimização adequada para o problema de três objetivos propostos. Não é possível saber exatamente quais são as soluções ótimas para o problema. Portanto, comparamos as soluções encontradas com as soluções obtidas heurística gulosa para o problema de maximização da população coberta e o de minimização da distância mé-

dia. Como as soluções de métodos gulosos se distanciam da solução ótima em soluções que selecionam mais hospitais, aplicamos o NSGA-II otimizando dois objetivos também para os problemas de maximização da população coberta e o de minimização da distância média. Os resultados mostraram que o NSGA-II, quando otimizando três objetivos obtem soluções pelo menos tão boas quanto as heurísticas gulosas para cada objetivo. No entanto, NSGA-II que tenta apenas maximizar a cobertura populacional minimizando o número de hospitais selecionados obtém soluções com mais hipervolume as soluções que o NSGA-II otimizando os três objetivos encontra para mais de nove hospitais selecionados.

O estudo tem algumas limitações como a ausência de atribuição de capacidade aos hospitais e o fato do modelo não considerar os benefícios de redundância de hospitais cobrindo a mesma região. Além disso, a adição de métodos de busca local ao NSGA-II pode melhorar os resultados e, portanto, é um caminho de trabalho futuro. A adição de outras variáveis no modelo a partir de dados de incidência de AVC na região e dados demográficos são outros potenciais aprimoramentos.