D3S: Decision Support System for Sectorization

Abstract

Sectorization problems refer to dividing a large set, area, or network into smaller parts concerning one or more objectives. A decision support system (DSS) is a relevant tool for solving these problems, improving optimisation procedures, and finding feasible solutions more efficiently. This paper presents a new web-based Decision Support System for Sectorization (D3S). D3S is designed to solve sectorization problems in various areas, such as school and health districting, designing sales territories and maintenance operation zones, or political districting. D3S bridges the gap between sectorization problems and a state-of-the-art decision support tool due to its broad design. The paper aims to present generic and technical attributes of D3S by providing detailed information regarding the problem-solution approach (based on Evolutionary Algorithms), objectives (most common in sectorization), constraints, structure, and performance.

Keywords: Sectorization, Districting, Resectorization, Decision Support System, Evolutionary Algorithms

1. Introduction

Sectorization, also known as districting or territory design, refers to dividing a whole into smaller parts guided by certain objectives and respecting specific restrictions. Sectorization problems are motivated by several different applications, such as political districting, designing sales territories or maintenance operations zones, school and health districting, designing police zones, forest planning, or air-traffic services. Among all these application fields, Kalcsics et al. (2005) highlight two of them – political districting and service territory design – as the main applications. The nature of these two applications corresponds to the sectorization of geographic areas (polygons) and the sectorization of the single points/coordinates (e.g. customers, service centres).

The literature recognises several primary objectives in the context of sectorization, including equilibrium (balance or equity in sectors), compactness (density within sectors), and contiguity (the ability to move between all basic units without leaving a single sector) (Bozkaya et al., 2003; Benzarti et al., 2013; Rodrigues and Ferreira, 2015; Xie and Ouyang, 2016; Martinho et al., 2017). Additionally, restrictions such as capacity limitations within sectors, the requirement for distinct sectors between two basic units (boundaries), and the necessity for two basic units to be in the same sector (neighbourhood) are acknowledged as crucial considerations for deriving feasible solutions to these problems. These objectives and restrictions differ from one application to another or among decision-makers (DMs). Such diversity creates obstacles in the solution procedures and difficulties in defining the problem clearly. In addition to these differences, a real-world problem can also be very large. Finding a reasonable solution is difficult when several objectives, constraints and the size of the problems are considered. Thus, advanced computer support is needed to solve these problems.

A decision support system (DSS) is a computerised information system designed to solve complex problems and assist DMs in decision processes. DSS are important for policy-making due to their analytical strength. Besides, multiple users can access DSS simultaneously and benefit from its advantages in relaxing the cognitive, spatial, economic and temporal limitations (Holsapple, 2008). Moreover, a DSS is vital in improving time and cost efficiency due to finding faster solutions under different conditions (scenarios), providing adequate performance assessment, and empowering users' decisions even if the problem is not well-characterised (Ramachandra et al., 2006). Finally, these tools are innovative, trustworthy and satisfactory for the DMs.

This work presents a DSS called D3S (Decision Support System for Sectorization) to solve sectorization problems. D3S is a solver-oriented web-based DSS that comprises several problem solvers designed to optimise a specific case defined by the DM. Earlier efforts to build a DSS to solve sectorization problems were focused on a specific type. However, to the best of our knowledge, no study has developed a comprehensive DSS that covers diverse cases. D3S addresses this lack by filling the gap between sectorization problems and an inclusive decision-support tool with state-of-the-art methods.

D3S was developed as a generalised system covering a wide range of sectorization applications. The solution approach is based on metaheuristics, namely, Non-dominated Sorting Genetic Algorithm-II (NSGA-II) and Genetic Algorithm (GA). The corresponding algorithms developed are adapted to the applications considered. D3S allows multiple objectives (balance, compactness, and contiguity), restrictions (capacity, boundary, neighbour, similarity) and routing, supporting the decision-making process under various scenarios and applications. In D3S, sectorization problems are characterised by four main groups. Each group represents a different identity. The services provided by D3S are named after these groups: (i) basic sectorization problems, (ii) sectorization problems with service centres/facilities, (iii) resectorization problems and (iv) dynamic sectorization problems. Given these aspects, the advantages and superiority of D3S over existing tools can be summarised as follows: (i) its solver-oriented design excel in addressing sectorization problems by efficiently exploring and generating optimal solutions, (ii) its accessibility and user-friendly design simplifies widespread use and efficient decision-making by a diverse range of DMs, and (iii) its unified framework to address various sectorization problems, fostering a comprehensive understanding through categorisation into four generalised groups, facilitates effective problem-solving across various applications.

The remainder of this paper is structured as follows. Section 2 reviews the literature focusing on different sectorization applications, and DSS used to try to address them. Section 3 shows the architecture of D3S; Section 4 broadly explains the optimisation methods and adaptations implemented for each D3S service, and Section 5 shows its user interface. Section 6 describes the computation of objectives, restrictions and routing, while in Section 7 a case study on a Portuguese delivery company is solved using the D3S. Finally, Section 8 concludes.

2. Literature Review

Sectorization covers several problem types related to the design of geographic areas or districts considering one or more objectives. The oldest application field is political districting, which seeks to apportion territory among voters to ensure unbiased electoral processes impartially. A common strategy involves periodic resectorization of political districts in subsequent elections, wherein input data is updated to reflect changes in geographic areas. Designing sales and service territories is an important aspect of sectorization. It involves solving sub-problems, such as deciding where to locate service centres or facilities (known as location analysis) and defining areas for maintenance operations, technical facilities, or clients.

Several sectorization applications directly pertain to the distribution of essential human life resources. For instance, issues surrounding the deployment of power networks aim to distribute internet, mobile networks, or electricity across defined districts. Similarly, waste collection and water distribution enhance living standards and urban infrastructure. Notably, sectorization finds relevance in optimising mobility and transportation. Additionally, health, schooling and police districting are problems that often emerge beneath sectorization. Health districting optimizes caregiving services to patients or ambulance relocation. School districting focuses on the allocation of students to schools. Similarly, police districting endeavours to define service areas for police patrolling. These problems commonly involve considerations of some criteria and restrictions, such as equilibrium, compactness, contiguity, boundary definition, capacity constraints, and desirability (see Bozkaya et al., 2003; Jahuira and Cuadros-Vargas, 2003; Kalcsics et al., 2005; Camacho-Collados and Liberatore, 2015; Lei et al., 2016; Bouzarth et al., 2018; Kim, 2018; Yanık et al., 2019; Hajiali et al., 2022). Incorporating these criteria aims to minimise travel costs, distances between sectors, and variations in work hours among employees.

Throughout the years, diverse attempts have been made to develop Decision Support Systems (DSS) in various formats to facilitate solution procedures for different sectorization applications.

These DSS are integral to aiding decision-making processes and can be broadly categorized into web-based and desktop-based deployment formats.

2.1. Web-based systems

Web-based DSS (Wb-DSS) offers accessibility, flexibility, and real-time updates, enhancing remote collaboration and decision-making. Their scalability and customisation capabilities make them effective in complex decision scenarios, improving domain efficiency.

Exploring Wb-DSS within various application domains has yielded noteworthy insights and advancements. Noorian and Murphy (2017) devised a Wb-DSS utilising a GA grounded in graph theory to address territory design problems. The main objective was to ensure sector balance by adhering to region boundaries and neighbour restrictions, while accommodating multiple objectives. In a distinct context, Pan et al. (2016) contributed to the field by developing a Wb-DSS tailored for scheduling the daily and long-term water distribution network in Milan, with an emphasis on sectorizing the water system and orchestrating pump operations within Milan City. The solution strategies employed metaheuristics, specifically NSGA-II and AMGA2 (Archive-based Micro Genetic Algorithm). A different facet of Wb-DSS application emerged in the healthcare sector, as detailed by Eveborn et al. (2006). Addressing healthcare systems' rising costs and labour intensiveness, particularly in home care provision, the authors introduced a DSS named Laps Care. This system facilitated the development of visiting schedules for care providers, considering various restrictions and soft objectives. The decision-making process involved a set partitioning model and a repeated matching algorithm. Furthermore, Rest and Hirsch (2022) presented a Wb-DSS specifically tailored for home health care (HHC) services, showing scalability, manageability of heavy workloads, and customisation capabilities for constraints and objective functions. Incorporating Causal-Loop-Diagrams (CLDs) provided insights into the impacts of epidemics, blackouts, heat waves, and floods on the HHC system. A Tabu Search metaheuristic formed the basis of the DSS, addressing the intricacies of urban regions. Finally, Hajiali et al. (2022) advanced the field by developing a model-driven online DSS for ambulance relocation, considering real-time changes. It had two objectives: minimising travel time and maximising patients' demand coverage. Linear programming was the underlying solution method, showcasing the model's adaptability to dynamic conditions.

2.2. Desktop-based systems

A comprehensive exploration of desktop-based DSS (Db-DSS) reveals diverse applications and methodologies across various domains. One way of developing Db-DSS is to create a built-in to be used in an existing tool. Bozkaya et al. (2011) introduced 'DistrictBuilder,' a spatial DSS integrated within ArcGIS software designed for resectorizing political districts. Employing the Tabu search algorithm, the system addressed multiple objectives, meeting districting criteria such as population balance, contiguity, compactness, and respect for natural boundaries. In the context of snow removal, Labelle et al. (2002) developed built-in within MapInfo GIS software, incorporating the two-resource assignment problem (TRAP) heuristic. The aim was to rapidly sectorize cities during snow-related emergencies, providing an interactive solution for decision-makers. Furthermore, Bergey et al. (2003) employed Genetic Algorithms (GA) to create a Microsoft Excel built-in DSS for electric power districting. The GA configurations ensured contiguity in the distribution network, and the resulting DSS featured a visualisation tool for decision-makers.

Another common approach beneath desktop-based systems is micro-computer-based or PC-based DSS. Moynihan et al. (1995) presented a micro-computer-based DSS utilising simulation and heuristic-based solutions in logistic and distribution network planning. Users could evaluate various alternatives under different scenarios and objectives, considering variable and fixed costs associated with manufacturing, selling, and distributing products. Moreover, Ferland and Guénette (1990) developed a micro-computer-based DSS with heuristic methods addressing school district problems. Users had the flexibility to interact with the system, modifying solutions based on objectives and constraints such as contiguity, school capacity, population, and student numbers. A spatial micro-computer-based DSS for the sectorization of nursing-related services is developed by Begur et al. (1997). They integrated geographic information system software with scheduling heuristics and databases, streamlining nurse deployment decisions and improving work balance among nurses. Finally, Camacho-Collados and Liberatore (2015) introduced the Predictive Police Patrolling DSS (P3-DSS) as a PC-based system in police districting employing a Greedy Randomized Adaptive Search Procedure (GRASP) algorithm aimed to create balanced sectors for police patrolling by efficiently distributing police cars/officers within service territories. The model incorporated random search and local search algorithms, offering a dynamic approach to police patrolling.

2.3. The Contribution of D3S

Both desktop-based and web-based systems are good options to build a DSS. However, web-based systems can be accessible through any device without installation, configuration challenges, or licensing restrictions.

Thus, D3S was designed as a web-based DSS, standing out as a comprehensive solver-oriented tool designed to address sectorization problems, which aims to solve a diverse range of problems, unlike previous DSS efforts that focused on specific types. The advantages of the solver-oriented design of D3S can be summarised as follows: (i) the problems are well segregated which allows updates in one solver to be independent of the others (ii) it can solve multiple problems simultaneously and (iii) it improves user experience due to predefined user input forms per solver.

D3S contributes significantly to the literature by collecting diverse sectorization problems under a single umbrella and offering an efficient tool.

3. Decision Support System for Sectorization

D3S is a non-commercial, unique and inclusive decision support tool developed to support decision-makers for sectorization problems, designed in a website format to make it easily accessible to users. Anyone can access the D3S platform through the following address: --[1]. Its implementation required three main steps: design of problem solvers, design of database and design of user interface. Implementations were done in Python on the Django web framework. Figure 1 shows the high-level architecture of D3S.

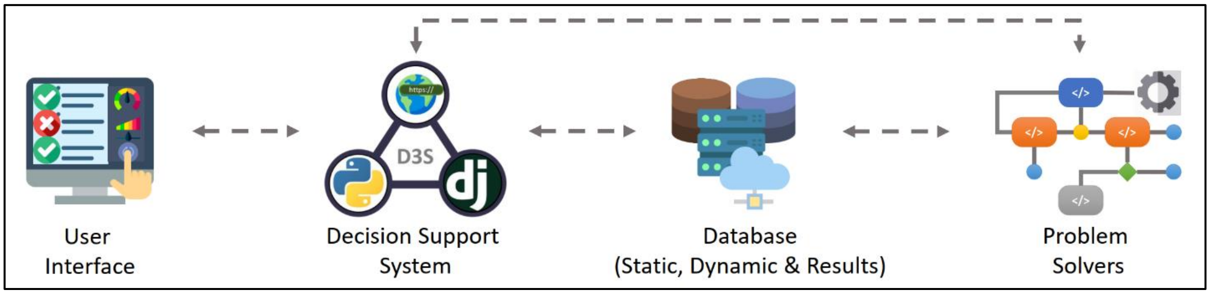

Figure 1: System Architecture of D3S (source: own elaboration)

D3S is a solver-oriented DSS that contains 19 problem solvers, each designed to solve a particular class of problems. The user's request indicates which problem solver is adequate (Holsapple, 2008). Modifications occurred regarding these solution methods (GA and NSGA-II) differences.

The database includes all the information regarding D3S and stores three types of data: static, dynamic, and results. The static data is unchangeable data, and available for all users rendered in the user interface. For instance, information regarding the specifications of the problems is determined through several surveys in D3S. The data regarding the logic behind the questions in these surveys occupies a static database component. Dynamic data changes from one user to another, being mostly the data presented by the user to the D3S. For example, the answers to the survey or the problem instance can be considered dynamic. Finally, the results are the data stored in the database after the optimisation, composed of solutions found by the algorithms and their visualisation.

---

[1] Website link removed for blinded review purposes; link available upon request

4. Solution Approach

The solution approach embedded in the D3S is based on metaheuristic methods, namely, GA (Goldberg and Holland, 1988) and NSGA-II (Deb et al., 2002). Metaheuristics find solutions in a reasonable time when real-size instances and several objectives are in question (Andria et al., 2015). GA is selected for single-objective optimisation, while NSGA-II is selected for multi-objective optimisation. GA represents an approximate and adaptable approach, effectively addressing practical problems characterised by incomplete rigour in data and objectives. Capitalising on this flexibility, various authors have employed GA in the literature to successfully tackle sectorization and related problems (see Konak, 2012; Agustín-Blas et al., 2009; Bacao et al., 2005). Similarly, NSGA-II has demonstrated robust performance in handling up to three objectives, as highlighted by several authors in the literature who utilised it in the resolution of sectorization and similar problems (see Pan et al., 2016; Farughi et al., 2020; Zhang et al., 2019; Zou et al., 2016). These algorithms were modified to adapt to the concerned applications, as explained in Section 4.3.

Since both methods are based on Evolutionary Algorithms (EA), they share some common concepts and steps. The EA are population-based metaheuristics. The population comprises a finite set of solutions, named individuals, evaluated according to their performance on objective function(s). These algorithms search for superior individuals over generations and try to evolve the population to a better plateau.

In D3S, a matrix form binary grouping (MFBG) genetic encoding system is used to form the solutions due to its adequacy on the sectorization problems (Ozturk et al., 2023). MFBG is a binary matrix where columns represent the sectors, and rows are the basic units (e.g., coordinates or polygons). When a basic unit is assigned to a sector, the corresponding row and column intersection takes the value of one. The restrictions of this encoding system are: (i) each basic unit can be part of only one sector, (ii) each sector has to have at least one basic unit.

It is possible to see the format of the MFBG genetic encoding system below.

$$MFBG = \left[x_{ij}\right]_{I \times J}, \text{ where } x_{ij} = \begin{cases} 1 & \text{if basic unit } i \text{ is assigned to sector } j \\ 0 & \text{otherwise} \end{cases}$$

In EA, completing one whole generation (iteration) takes several steps, such as selection, crossover, mutation and elitism. Selection means picking two individuals as parents to mate and generate offspring. In D3S, both GA and NSGA-II follow the tournament selection method. Two random solutions are picked and evaluated regarding their superiority, selecting the first parent as the one that performs better. This process repeats to select the second parent. This way, all individuals in the population are matched to be mated[2]. The crossover step is a term used to define the process of mating. Depending on the selected crossover method, one or two offspring could be generated from two parents. In D3S, a multi-point crossover is used. The randomly selected rows are switched between the parents to generate the two offspring. The crossover used in D3S is illustrated below.

$$\begin{bmatrix} x^1_{1j} \\ \mathbf{x^1_{2j}} \\ x^1_{3j} \\ \vdots \\ \mathbf{x^1_{ij}} \\ \vdots \\ x^1_{nj} \end{bmatrix} \Longleftrightarrow \begin{bmatrix} x^2_{1j} \\ \mathbf{x^2_{2j}} \\ x^2_{3j} \\ \vdots \\ \mathbf{x^2_{ij}} \\ \vdots \\ x^2_{nj} \end{bmatrix} \longrightarrow \begin{bmatrix} x^1_{1j} \\ x^2_{2j} \\ x^1_{3j} \\ \vdots \\ x^2_{ij} \\ \vdots \\ x^1_{nj} \end{bmatrix} \quad \begin{bmatrix} x^2_{1j} \\ x^1_{2j} \\ x^2_{3j} \\ \vdots \\ x^1_{ij} \\ \vdots \\ x^2_{nj} \end{bmatrix}$$

Crossover is followed by the mutation implemented only in the offspring population. The mutation aims to create some randomness in the population to minimise the possibility of being stuck in local minima. The mutation method changes the assignment of basic units from one sector to the other according to a given probability. After mutation, offspring are merged into the main population. Finally, elitism is used to eliminate the worst-performed individuals from the population. This step keeps the population size steady and leads the population to a better scale by eliminating the worst-performing solutions.

[2] The superiority of an individual is evaluated differently in GA and NSGA-II. These differences are explained in sections 4.1 and 4.2, respectively.

The parameters necessary for the EA are the number of solutions in the population (Pop_{size}), the number of generations (Gen) and the mutation probability (P_{mut}). The parameters used in D3S for both GA and NSGA-II are presented in Table 1.

Table 1: EA parameters used in D3S

Pop_{size}	Gen	P_{mut}
100	1500	0.01

Carefully selecting parameters is crucial in EA, taking into account specific performance metrics. Ozturk et al. (2022) identified three performance metrics (error ratio, inverted generational distance, and spacing) suitable for evaluating the quality of the solution space when applying EA to address sectorization problems. Following multiple test runs with varying population sizes (from 50 to 100) and mutation probabilities (up to 10%), higher population sizes and smaller mutation probabilities produce superior performance across those three metrics. For this reason, the D3S uses a population size (Pop_{size}) of 100 and a mutation probability (P_{mut}) of 1%. Additionally, tests were done to find the appropriate number of generations using the instances provided by Ozturk et al. (2022). Tests were run for different generation numbers, keeping constant population size and mutation rates, revealing that 1500 generations yield the best results across the three performance metrics mentioned above.

The main difference between GA and NSGA-II is how the solutions are evaluated. The rest of this section details the specific differences between these two algorithms.

4.1. Genetic Algorithm

Algorithm 1 shows the pseudocode of the GA used in D3S with its most generic form with implemented adaptations when specific cases require it.

In GA, each solution is evaluated by its score in the fitness function using a weighted composite single objective function of the solutions' performance on one or more objectives. Equation (1) illustrates the fitness function used in the GA of D3S, where $F(x)$ represents the fitness value of solution

x, where $f_i(x)$ and w_i (for i = 1,...,h) respectively are the normalised value and the weight of the i^{th} objective.

$$F(x) = w_1 f_1(x) + w_2 f_2(x) + ... + w_h f_h(x), \ \ where \ \sum_{i=1}^{h} w_i = 1 \qquad (1)$$

Algorithm 1 Pseudocode of GA

```
Generate N feasible solutions and insert into Population (Pop_size = N)
Evaluate the solutions according to the fitness function
Generation := 0
while Generation < Gen do
    while Pop_size < N × 2 do
        Select parents through tournament selection
        Create two offsprings using crossover
        Pop_size := Pop_size + 2
    end while
    Mutate offsprings (for selected P_mut)
    Merge offsprings into the population
    Evaluate the solutions according to the fitness function
    while Pop_size > N do
        Delete the worst solution from the population according to F(.)
        Pop_size := Pop_size − 1
    end while
    Keep the best solution
    Generation := Generation + 1
end while
```

When a weighted composite single objective function is used, normalisation is necessary if the measurement units of the objectives are different. In D3S, the min-max normalisation method sets all the values between 0 and 1. Furthermore, the weighting procedure is handled using the Analytical Hierarchy Process (AHP), a preference-based weighting method presented by Saaty (1977, 1980, 2008). In D3S, the weighting scheme was standardised[3] for all users, expecting to compare the objectives regarding their relative importance for the problem from 1 (objectives i and j are equally important) to 9 (objective i is extremely more important than objective j) as defined in the

[3] While not all D3S users may possess the expertise to determine precise weights for each objective, they likely have a general sense of the relative priorities. Therefore, a standardised weighting scheme is employed within the AHP framework instead of directly soliciting precise weights from users.

fundamental comparison scale by Saaty (1977)[4]. Each objective is compared with each other unilaterally in AHP. Thus, $\frac{h(h-1)}{2}$ comparisons are required for h objectives. The performance of the solutions is evaluated in the weighted composite single objective function.

4.2. Non-dominated Sorting Genetic Algorithm - II

Algorithm 2 presents the pseudocode of the NSGA-II used in D3S in its most generic format. The adaptations for specific cases are implemented on Algorithm 2 as needed.

Algorithm 2 Pseudocode of NSGA-II

```
Generate N feasible solutions and insert into Population (Pop_size = N)
Evaluate the solutions according to the selected objectives
Compute μ(.)
Calculate CD for each solution
Generation := 0
while Generation < Gen do
   while Pop_size < N × 2 do
      Select parents through tournament selection
      Create two offsprings using crossover
      Pop_size := Pop_size + 2
   end while
   Mutate offsprings (for selected P_mut)
   Merge offsprings into population
   for each solution x in the current population do
      Evaluate the solutions according to the selected objectives
      Compute μ(.)
      Calculate CD for each solution
   end for
   while Pop_size > N do
      Delete the worst solutions from the Pop according to frontier or CD
      Pop_size := Pop_size − 1
   end while
   for each solution x in the population do
      Evaluate the solutions according to the selected objectives
      Compute μ(.)
      Calculate CD for each solution
   end for
   Generation := Generation + 1
   Output the Pareto frontier results
end while
```

[4] The fundamental comparison scale of AHP can be found in (Saaty, 1977, 1980)

NSGA-II is a multi-objective optimisation method where the solutions are evaluated for each objective separately and simultaneously. Equation (2) shows a generalised illustration of this process, where μ is the objective vector. Moreover, x is a vector of decision variables in the feasible region of the solution space X.

$$Min \qquad \mu(x) = (f_1(x), f_2(x), \ldots, f_h(x)) \;\; subject\ to \qquad x \in X \qquad (2)$$

The solutions are ranked in Pareto frontiers according to their performance in each objective. These ranks are established according to two parameters in NSGA-II: (i) set of dominating members, and (ii) domination count. The former contains the solutions that a solution dominates. If a solution is better than another solution in at least one objective while not being worse in the other objectives, that solution is considered dominant over the other solution. The latter, on the other hand, counts how many other solutions dominate a solution. These two parameters are computed for each solution in the population. The solutions not dominated by any other solution are allocated on the first Pareto frontier. The solutions dominated only by the solutions on the first Pareto frontier are allocated in the second Pareto frontier. The solutions that are dominated only by the solutions in the first and second Pareto frontier form the third Pareto frontier. And so forth, until all solutions are allocated to Pareto frontiers. The rank of the Pareto frontier, to which a solution belongs, represents how well that solution performed compared to the other solutions in the population. Thus, a solution from a lower rank is considered better than a solution from an upper rank when minimisation is considered. In addition, solutions in the same Pareto frontier are considered equally good. Nevertheless, the NSGA-II specific concept called Crowding Distance (CD) helps to compare solutions from the same Pareto frontier. More precisely, the CD shows the concentration of solutions in the objective space. Since it is desired to have diversified solutions, the higher the CD, the more representative and better the solution will be. In the selection procedure of NSGA-II, the information regarding the Pareto frontiers' rank and the CD value of each solution are necessary. If both randomly selected solutions are in the same Pareto frontier, then the parent is decided regarding their CD values. Or, if one of the randomly selected solutions is in a better Pareto frontier, it is selected as the parent. This information is also used in the elitism part of the algorithm at the end of each generation.

4.3. Solution Methods Adaptation for D3S Services

The presented solution methods were adapted to fit within the addressed application requirements, and while they have their specifications, it is possible to find common ground among them. For instance, problems that aim just to sectorize geographic areas or points, problems that consider service or distribution centres, or problems that aim to modify a given solution can be considered similar due to their characteristics. By this logic, the sectorization problems were grouped by similarity, creating four groups: basic sectorization, sectorization with service centres, resectorization and dynamic sectorization, comprising the four D3S services introduced in detail in the following subsections.

4.3.1. Basic Sectorization Problems

Basic sectorization problems do not consider any predetermined service centres/facilities and do not plan to determine one. They aim to sectorizing groups of points (coordinates) or geographic areas by considering specified objectives and restrictions.

Both GA and NSGA-II are implemented in the basic sectorization case. If the user asks for multiple trade-off solutions from the D3S, NSGA-II runs automatically. On the other hand, if a single solution is ordered, GA is activated. In the basic sectorization case, no specific algorithm modifications were needed. Figure 2 shows the possible problem specifications that can be solved under this service.

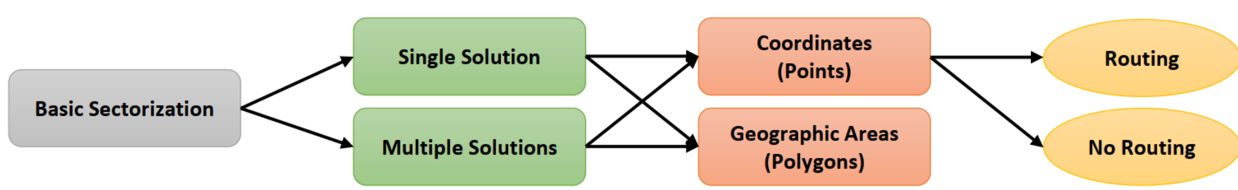

Figure 2: Basic sectorization problem tree

4.3.2. Sectorization Problems with Service Centres

Sectorization problems with service centres consider predetermined service centres or plans to determine them. When service centres are in question, some possible scenarios exist for concretising the problem. In D3S, these possibilities are grouped into four cases, as presented in Figure 3.

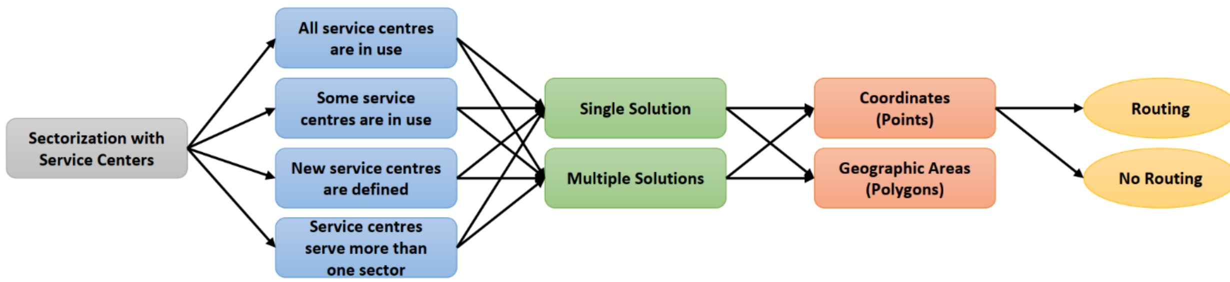

Figure 3: Sectorization with service centres problem tree

In the first case, all the service centres specified in the data must be used, sectorizing the territory for the given number of centres. The only modifications required were on the genetic encoding system, where previously a column represented a sector, now representing a service centre. In the second case, only a given number of the specified service centres will operate. Some algorithm modifications were needed to allow D3S to deal with this case (see Algorithm 3 for systematised modifications). The user provides the number of service centres to be used, and D3S performs several scenarios where different service centre combinations are tested, and tries to find the best sectorization for the tested scenarios. The number of scenarios is defined regarding the total number of service centres given in the data. If the number of total service centres is five or less, all the service centre combinations were tried in a different scenario. Otherwise, one-third of the total number of service centres is used as the number of scenarios. A scenario is created by randomly selecting the required number of service centres from the set of all given service centres, taking into account their distance from each other. This way, the service centres at a small distance were avoided from operating simultaneously when it is the case. Moreover, the scenario is considered valid if the total capacity of the selected service centres is bigger than the basic units' total quantity (demand).

Algorithm 3 Modifications for the case of some service centres (SCs) in use

Number of potential SCs = N_p
Number of SCs to be used = N_f ($N_f < N_p$)
Number of scenarios = S
for each scenario **do**
    while TRUE **do**
        Randomly select N_f SCs from the set of potential SCs
        if Total capacity of N_f SCs $\geq$ Total quantity (demand of basic units) **then**
            FALSE
        else
            TRUE
        end if
    end while
    Store the selected SCs
    Run the solution method by considering the selected SCs
    if GA **then**
        Keep the best solution
    else if NSGA-II **then**
        Keep the first Pareto frontier solutions
    end if
end for
if GA **then**
    Take all the best solutions
    present the the best solution with the best fitness value
else if NSGA-II **then**
    Put all the first Pareto frontier solutions in a solution pool
    Rank the pool solutions into Pareto frontiers according to their performance
    Present the final first Pareto frontier solutions to the user
end if

The algorithm (either GA or NSGA-II) is run for each valid scenario. In the case of GA, all the best solutions found for the different scenarios are taken and evaluated according to their fitness value. The best is presented to the user. In the case of NSGA-II, all the first Pareto frontier solutions found for the different scenarios are taken and put in a solution pool, where they are re-ranked into Pareto frontiers for the selected objectives. Finally, the solutions ending on the first Pareto frontier are presented to the user as the best trade-off solutions.

The third case is about defining new service centres of facilities. If the user does not have predefined service centres and plans to define possible locations to establish service centres, D3S provides these locations to support the user's decision-making process. A few modifications were needed in the solution methods for this case, as summarised in Algorithm 4. Namely, the ideal locations for the potential service centres are computed and presented to the user, considering the sectorization obtained in the best solution(s) found by GA or NSGA-II. These locations are the centroids of the sectors if the routing is not considered. If the routing is required, then the basic unit closest to the centroid of a sector is selected as its service centre.

The last case of sectorization with service centres occurs when these centres may serve more than one sector each. For instance, if the service centre represents a police station, the police officers of this police station should be divided into sectors to be serviced. In this case, some modifications are also implemented in the algorithms. These alterations are summarised as follows. First, basic sectorization is considered; thus, all the basic units are sectorized, ignoring service centres. Then, the sectors are assigned to the predetermined service centre(s), and solutions are presented to the user.

Algorithm 4 Modifications for the case of new defined service centres (SCs)

```
if GA then
    Find the SCs as the ideal locations of the sectorization represented in the
    best solution
else if NSGA-II then
    for each solution in the first Pareto frontier do
        Find the SCs as the ideal locations of the sectorization represented in
        the solution
    end for
end if
```

As in basic sectorization, both GA (single solution) and NSGA-II (multiple solutions) are valid approaches to solving all cases of sectorization problems.

4.3.3. Resectorization Problems

Resectorization problems correspond to redesigning a solution that is already in use. If the user wants to explore other possible alternatives or changes, and adaptations are needed to address a new reality, resectorization may be requested. Both basic sectorization and sectorization with service centres problems can be subject to resectorization. Figure 4 shows the problem tree of resectorization.

Figure 4: Resectorization problem tree

In sectorization with service centres, resectorization is done only for the first case where all the service centres are in use. Any other cases were not considered. Both GA and NSGA-II are valid solution methods for resectorization problems[5]. Algorithm 5 describes the steps of resectorization.

Algorithm 5 Modifications for resectorization

```
Define the adaptation (Completely or Partially)
if Completely then
    Run the Algorithm 1 or Algorithm 2 without considering the given solution
    and present a new solution to the user
else
    Minimum similarity level = MinSim
    Run all the steps of Algorithm 1 or Algorithm 2
    for each solution x in the population do
        Compute the similarity between solution x and the given solution
        if similarity < MinSim then
            Delete the solution x
            Generate a feasible solution and insert it into Population to keep
            Pop_size equal
        end if
    end for
end if
```

When resectorization is required, it is generally desired to keep some degree of similarity to the given solution. For that reason, D3S proposes two options to the user: (i) change *completely* or (ii) change *partially* the given solution. The new solution is kept as similar as possible to the previous

[5] In practical business applications, resectorization scenarios often involve more complex restrictions (e.g., removing some other existing centres, or adding new service centres). While the resectorization service in D3S may seem limited in directly handling these more complex cases, users can iteratively analyze such scenarios. For instance, to reduce service centres, users can start with the second case of the second service, achieve the desired reduction, and then proceed with resectorization. Conversely, users can opt for the third case of the second service to add new service centres, and input them into the resectorization, accommodating scenarios that may involve keeping specific centres open, resulting in a modified solution.

one for a *partial* change. That is achieved by randomly generating the initial solutions (population), using the user-specified minimum desired level of similarity (MinSim) as a filter, replacing those with a similarity level below MinSim. The calculation of *similarity* is described in Section 6.2.3.

4.3.4. Dynamic Sectorization Problems

Dynamic sectorization problems consider changes in the instance over time. Thus, it is necessary to adapt the solution from one period to the following one.

D3S specifically allows changes in the basic units' quantity (i.e., demand) for a given percentage, and adjusts the solution in the next period. The time horizons are considered discrete. The user defines the type of distribution (uniform or normal) and the percentage change (quantity). D3S follows Equation (3) for each basic unit regarding the DM's preferences.

$$L_i^{t+1} \sim U\left(l_i^t - \frac{\Delta\%}{100} l_i^t, \quad l_i^t + \frac{\Delta\%}{100} l_i^t\right) \quad or \quad L_i^{t+1} \sim N\left(l_i^t, \quad \frac{\frac{\Delta\%}{100} l_i^t}{3}\right) \tag{3}$$

Here, L_i^{t+1} is the random variable representing the quantity of basic unit i in period $t + 1$, l_i^t is the quantity of the basic unit i at period t, and $\Delta\%$ is the percentage change allowed to the quantity. Moreover, U and N respectively represent the uniform and the normal probability distributions. The expressions inside the brackets are the parameters of the corresponding distribution: the minimum and the maximum quantity values for the uniform distribution, and the mean quantity and the standard deviation for the normal distribution. That basic unit is permanently dropped from the instance if the returned quantity is below zero. The purpose is to adapt the solutions considering the changes in the instance from one moment to the next. Thus, the solution found in moment t is used to generate the population in moment t+1. For this reason, only GA is implemented as a solution method for dynamic sectorization problems[6].

[6] NSGA-II gives several trade-off solutions as equally good. In dynamic problems, the initial population in $t + 1$ is created depending on the best solution found in t. When several solutions are considered equally

Basic sectorization problems and all the cases of sectorization problems with service centres can be subject to dynamic problems, as represented in Figure 5.

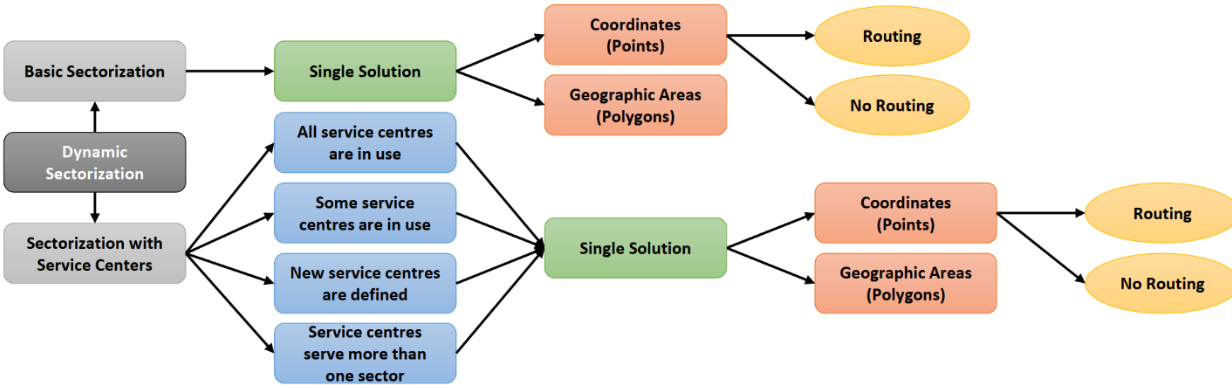

Figure 5: Dynamic sectorization problem tree

The modification in the pseudocode for dynamic problems is represented in Algorithm 6.

Algorithm 6 Modifications for dynamic sectorization

Define T: number of periods
Generate N feasible solutions and insert into Population ($Pop_{size} = N$)
Evaluation of the solutions according to the fitness function
$Period := 0$
while $Period < T$ **do**
    if $Period > 0$ **then**
        Consider the demands of $Period$
        Generate N feasible solutions from the best solution found in period $Period - 1$ and insert them into Population ($Pop_{size} = N$)
        Evaluate the solutions according to the fitness function
    end if
    Complete a whole generation as specified in Algorithm 1
    $Period := Period + 1$
end while

Specifically for the case in which one service centre can serve more than one sector, the number of sectors each service serves must be defined initially and cannot be changed from one period to the next. Thus, to solve such cases dynamically over time, the sectors are first assigned to the service centres and then the basic units are allocated to the sectors.

good, it is impossible to pick one as best to guide the population in the next period. Thus, only GA is implemented in dynamic problems.

5. User Interface

D3S is designed with a user-friendly layout. Access to D3S is free but requires user registration. Once registered, the user logs on at any time and has access to all the functionalities of D3S. The details of using the D3S are briefly described in the remainder of this section. The user interface and D3S steps are available in a video, at https://www.youtube.com/watch?v=MjNwrLg4__s&t=16s.

5.1. Problem Identification

As already mentioned, D3S is organised into four categories of sectorization problems. The category fitting the user's problem is identified by filling in a survey. The questions are designed to capture the main features of the problem, as illustrated in Figure 6.

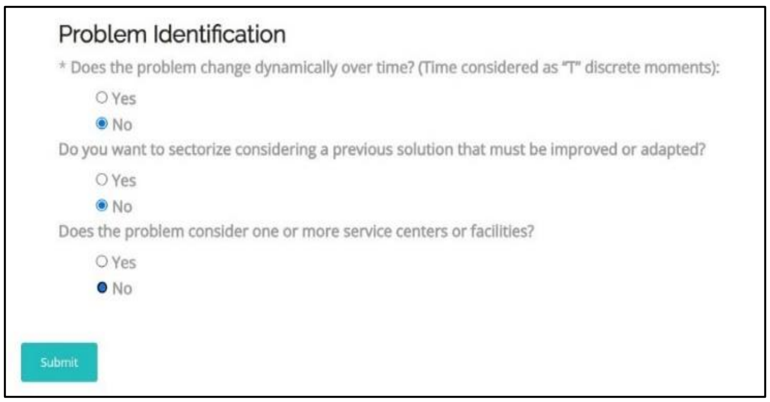

Figure 6: Survey for problem identification in D3S

The user is then directed to a more detailed survey about the particular type of sectorization problem. The user must also select the desired solution method and the objectives to be considered.

5.2. Data Upload and Preprocessing

After the problem specifications, the user must upload the data into D3S. The system gives the instructions for the submission. The data structure is explained in detail, and D3S has a data template available for download. Figure 7 illustrates the oversee for data upload.

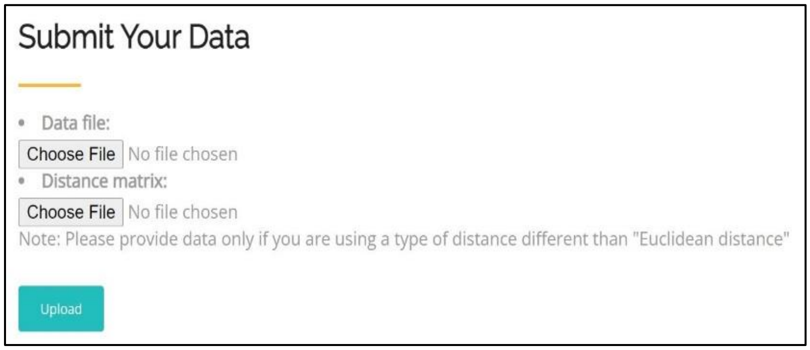

Figure 7: Oversee for data submission in D3S

As seen in Figure 7, the user can upload two data types. The first one is obligatory and should include the problem instance. The second one is optional, and the user should provide this data only if the problem under consideration uses a type of distance other than Euclidean. This file should include the distances between basic units in a matrix format, where diagonals are zeros.

5.3. Results Representation and Visualisation

Once D3S has finished processing the problem, the results are available to the user. Each user has an exclusive page to visualise all the submissions he/she has made to date in the system. Each submission has a unique ID, and the user can access the corresponding submission date and time, survey answers, uploaded data, and results provided by the D3S. The performance of the solutions and the visual representations[7] of the sectorization are also available to the user.

Moreover, all relevant data are presented to users in each service. These are (i) the assignment of each basic unit to each sector or service centre, (ii) the selected service centres in the case where only some of the service centres are in use among all given centres, (iii) the newly defined service centres, (iv) the routing if the user requests it, (v) the level of similarity in the resectorization, (vi) the changes in demand and points in the dynamic cases.

[7] Two Python packages - matplotlib (if the basic units are coordinates) and folium (if the basic units are polygons) - are used for visualisation.

6. Objectives and Restrictions

D3S handles several specifications for each sectorization type, as Figure 8 illustrates. This flexibility helps D3S to support decision makers in solving their problems under various scenarios.

		Balance (Quantity or Distance)	Compactness (Furthest or All)	Contiguity	Boundary	Neighbour	Similarity	Capacity	Routing
Basic Sectorization Problems	Geographic Areas (polygons)	Yes	Yes	Yes	Yes	Yes		Yes	
	Coordinates (points)	Yes	Yes	Yes	Yes	Yes		Yes	Yes
Sectorization with Service Centers	Geographic Areas (polygons)	Yes	Yes	Yes	Yes	Yes		Yes	
	Coordinates (points)	Yes	Yes	Yes	Yes	Yes		Yes	Yes
Resectorization Problems	Geographic Areas (polygons)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	
	Coordinates (points)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Dynamic Sectorization Problems	Geographic Areas (polygons)	Yes	Yes	Yes	Yes	Yes		Yes	
	Coordinates (points)	Yes	Yes	Yes	Yes	Yes		Yes	Yes

Figure 8: Specifications available by sectorization problem

6.1. Objectives

D3S is designed to allow the most common objectives in sectorization problems: balance (regarding quantity or distance), compactness (regarding the furthest element or all the elements) and contiguity. The next sections describe their mathematical expressions in detail.

6.1.1. Balance Regarding Quantity

Balance aims to create equity across sectors. Balance regarding quantity may refer to the basic units' demands, labour hours, or weights. This objective avoids quantity diversity among sectors by minimising their standard deviation. The expression for balance regarding quantity is in Equation (4) (adapted from Rodrigues and Ferreira (2015)).

$$std'_{B_q} = \sqrt{\frac{1}{K-1}\sum_{j=1}^{K}(q_j - \bar{B}_q)^2} \qquad (4)$$

$$\text{where} \quad \bar{B}_q = \frac{\sum_{j=1}^{K} q_j}{K} \quad \text{and} \quad q_j = \sum_{i=1}^{N_j} x_{ij} \times q_i, \quad \forall j = 1, \dots, K$$

Here, q_j is the total quantity in sector j, where x_{ij} is the basic unit i with quantity q_i located in sector j. $\bar{B}_q$ is the mean quantity of the K sectors. Finally, std'_{B_q} is the standard deviation from the mean quantity.

6.1.2. Balance Regarding Distance

Balance regarding distance aims to minimise the diversity of distances within the various sectors by minimising their standard deviation. Such an objective may be essential when the problem requires travelling between basic units, such as police patrolling or home healthcare services. Equation (5) gives the expression for balance regarding distance.

$$std'_{B_d} = \sqrt{\frac{1}{K-1}\sum_{j=1}^{K}(d_j - \bar{B}_d)^2} \qquad (5)$$

$$\text{where} \quad \bar{B}_d = \frac{\sum_{j=1}^{K} d_j}{K} \quad \text{and} \quad d_j = \sum_{i=1}^{N_j}\sum_{l=1;l<i}^{N_j} d_{il}, \quad \forall j = 1, ..., K$$

Here, d_j is the total distance within sector j, where d_{il} is the distance between basic units i and l located in sector j and N_j is the number of basic units in sector j. $\bar{B}_d$ is the mean distance of the K sectors. Finally, std'_{B_d} is the standard deviation from the mean distance.

6.1.3. Compactness Regarding Furthest Distance

Compactness refers to density in sectors. It is more desirable to have circle-shaped sectors than long sectors. Equation (6) shows the mathematical definition of compactness regarding the furthest distance.

$$C_f = \sum_{j=1}^{K} dist(o_j, x_{ij}^F) \qquad (6)$$

In Equation (6), o_j shows the mass centroid of sector j, and x^F_{ij} is the furthest basic unit to the centroid. When C_f is larger, the compactness of the sectors is considered worse.

6.1.4. Compactness Regarding All Distances

Compactness regarding all distances also aims to create well-shaped and dense sectors. The mathematical definition of compactness regarding all distances is presented in Equation (7).

$$C_a = \sum_{j=1}^{K} \sum_{i=1}^{N_j} dist(o_j, x_{ij}) \tag{7}$$

In Equation (7), o_j is the mass centroid of sector j, x_{ij} is a basic unit i located in sector j, and N_j is the number of basic units in sector j. A smaller C_a refers to better compactness as the basic units are closer to the centroid of the sector.

6.1.5. Contiguity

Contiguity refers to the ability to move between the basic units in the same sector. A sector is considered fully connected if it can move between all the basic units in the same sector without crossing to other sectors. D3S uses the contiguity definition from Rodrigues and Ferreira (2015).

The existence of paths between the basic units located in the same sector is represented by the symmetric square matrix M^j in Equation (8). By definition, there is no path to the basic unit itself.

$$M^j = \left[m_{wi}^j\right]_{N_j \times N_j} \tag{8}$$

$$\text{where} \quad m_{wi}^j = \begin{cases} 1 & \text{if there is a path between the basic units w and i in sector j} \\ 0 & \text{otherwise} \end{cases}$$

The level of contiguity c_j of sector j is given in Equation (9): the numerator is the number of valid paths in sector j, and the denominator is the total number of possible paths in sector j. If all the basic units are connected, then c_j takes the value of one.

$$c_j = \frac{\sum_{w=1}^{N_j} \left(\sum_{i=1}^{N_j} m_{wi}^j\right)}{N_j \times (N_j - 1)} \tag{9}$$

The mean contiguity $\bar{c}$ among all the sectors, in Equation (10), varies between 0 and 1, where N is the total number of basic units ($N = \sum_{j=1}^{K} N_j$). Contiguity is given by $(1 - \bar{c})$ since it should be minimised, and higher values of $\bar{c}$ denote better connectivity.

$$\bar{c} = \frac{\sum_{j=1}^{K} c_j \times N_j}{N} \tag{10}$$

6.2. Restrictions

Besides several objectives, D3S allows the selection of the most common restrictions in sectorization problems. The definitions and details are presented in the remainder of this subsection.

6.2.1. Capacity

Capacity constraints are common restrictions to be respected in sectorization problems. Solutions should be generated accordingly when there is limited space in the sectors. In D3S, any solution that exceeds the sector's capacity is considered infeasible and deleted from the population. If the violation of the capacity constraint occurs during crossover or mutation, the step is repeated until a feasible solution is obtained.

6.2.2. Boundary and Neighbour

Boundary constraints refer to basic units that should not be in the same sector due to nature's conditions, like two basic units located on opposite sides of a river. Neighbour information refers to the opposite of the boundary. More precisely, if two basic units are inseparable and must be in the same sector, they must be considered neighbours. D3S accepts boundary and neighbour restrictions simultaneously. To avoid errors, the user must provide consistent information to D3S.

6.2.3. Similarity

Similarity constraints are usually imposed in resectorization problems. For a solution given by the user and a new solution provided by D3S, the similarity measure in Equation (11) is used by D3S to calculate the proportion of basic units that stay in the same sectors.

$$Similarity = \frac{\sum_{j=1}^{K}\sum_{i=1}^{N} x^g[i,j] \times x^p[i,j]}{N} \tag{11}$$

Given the matrix form binary structure of the genetic encoding system used in the solution methods of D3S, a simple similarity measure is used to calculate the percentage of similarity of the provided solution to the given solution. In Equation (11), $x^g[i,j]$ is the matrix cell corresponding to basic unit i and sector j in the given solution. Likewise, $x^p[i,j]$ is the matrix cell corresponding to basic unit i and sector j in the provided solution. If the cell has the value one for both the given and the provided solution, then it means the basic unit is assigned to the same sector, and thus the multiplication is one. Alternatively it is zero if the basic unit is assigned to different sectors.

6.3. Routing

The routing starts from the service centres if they are defined, and starts from a random basic unit when not. A route is obtained for each sector using a greedy nearest neighbour algorithm that iteratively moves to the closest basic unit from the current one. D3S uses Euclidean distances unless the user submits an alternative distance data, as explained in Section 5.2.

7. Case Study: Sectorization of a Parcel Delivery Company

Here follows a brief reference to a case study related to a parcel delivery company located in the district of Porto. The real data includes 7956 basic units (as coordinates) to be served. The company provided no data regarding service centres or routes. Nonetheless, these types of problems usually consider service centres that serve the customers in their area using the shortest routes. This problem is defined as sectorization with service centres and aims to define potential locations for them and sectorizing the given area. The customers to be served (basic units) are divided into 50 sectors, and a different service centre will serve each. Also, Euclidean distances are considered to calculate the routes within each sector.

Figure 9 shows the D3S result. A single solution is provided through GA with equally given weights to three objectives (balance regarding quantity, compactness regarding furthest distance, and contiguity), with a fitness score of 0.5543. The 'X's indicate the proposed service centres location and the points are the basic units given in the data.

Together with the solution visualization, the D3S always provides additional information (illustrated in Figure 10) about sectors assigned to each basic unit within sublists, where the first element represents the ID of the basic unit and the second element represents the ID of the sector.

The second information given by D3S is regarding the 'Routes'. Routing information is only presented if the user asks for it. The routes are also presented within sublists. Each sublist gives the order for routing between the IDs of the basic units composed of a sector, shown for each sector, including the elements allocated.

Finally, the service centres' information is given under the title 'Service Centres Used (indexed or coordinates)', providing information regarding the locations of defined service centres. Each sublist includes the latitude and longitude information of each service centre defined for each sector, sequentially.

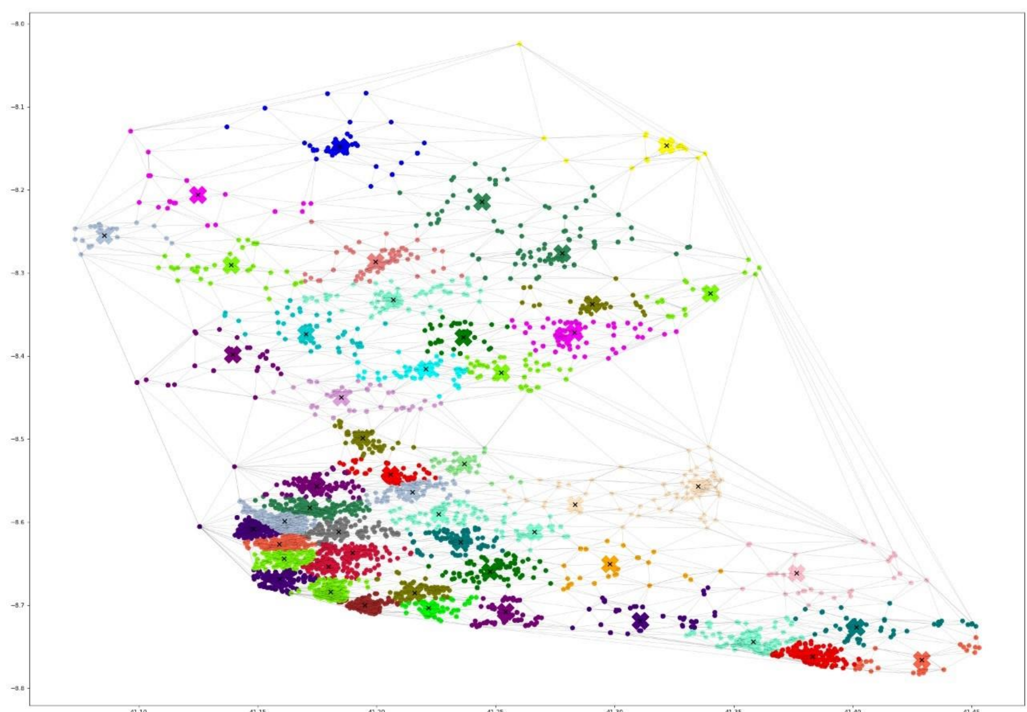

Figure 9: Sectorization with service centres - defining new locations for service centres to function

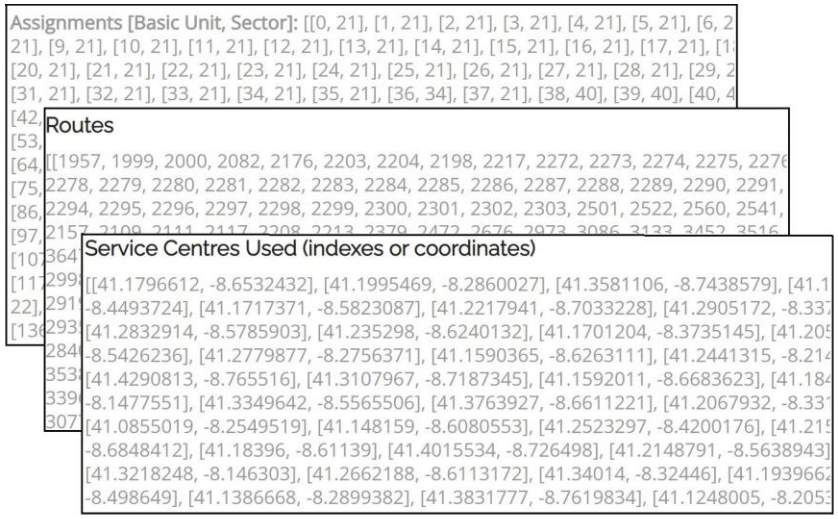

Figure 10: The details about the result provided by the D3S

8. Conclusions

This work describes the development of a web-based Decision Support System for Sectorization called D3S. The system gives solutions to sectorization problems relative to several applications. D3S provides four services, each named after four specific characterisations of sectorization. These are (i) **basic sectorization problems** which do not have defined service centres and do not plan to define them, (ii) **sectorization problems with service centres** which consider defined service centres or plan to define them, (iii) **resectorization problems** which aim to redesign a given solution considering changes in the instance, (iv) **dynamic sectorization problems** which provide adapted solutions dynamically in several time moments by changing the quantity of the basic units by a given probability. Both basic sectorization and sectorization problems with service centres can be subject to resectorization or dynamic sectorization problems. Sectorization follows an approach that adapts solution methods such as GA with AHP and NSGA-II.

There are several specifications within each problem in terms of objectives and restrictions. D3S is designed as a generic system that allows the most common objectives (i.e., balance regarding quantity or distance, compactness regarding the furthest element or all elements, and contiguity) and restrictions (i.e., capacity, boundaries, neighbours and similarity). In addition, D3S also provides routing, which minimises the travel time for the provided sectorization.

D3S is developed in Python as a web-based system with a user-friendly outline. Two surveys lead the user's problem to the relevant solver. The first survey is very small and intends to lead the user to the appropriate service. The second survey is more detailed, specific to each service, and tries to identify the basic characterisation of the problem as well as the solution type, objectives and restrictions. Question about the solution type defines the solution method and acts as the main filter to lead the problem to the proper problem solver. If the user asks for multiple solutions, NSGA-II based solvers are employed. Likewise, GA-based problem solvers are applied only if the user asks for a single solution. The other questions help to define the additional characteristics and lead the problem to the most relevant solver. After these surveys, the user is led to the data submission page, where detailed instructions about the data submission are given. When data is submitted, the results appear on the user-specific results page. The user can also observe the submission details, including the service, survey answers, and computational time.

A case study is presented for a parcel delivery company located in the district of Porto. The case is seen as sectorization with service centres, as it is planned to define service centres to serve customers. The related problem was quite extensive, with 7956 basic units/customers. It was divided into 50 sectors considering three objectives: balance (regarding quantity), compactness (regarding furthest distance) and compactness. Under equal weights for each objective, a single solution is asked from the D3S. Since it is important to serve all the customers, routing is also asked from D3S due to its relevance. D3S provided a single solution following GA. The fitness value, visualisation of the solution, assignment of each basic unit to the sectors, routing and the coordinates of the defined service centres to serve in each sector are delivered by the D3S.

D3S is a unique and modern decision support tool that utilises metaheuristic methods based on EA to solve various sectorization problems under different scenarios. Thus, due to its comprehensive and flexible design, the D3S will help researchers and decision-makers in their decision-making process and contribute to the sectorization literature.

References

Agustín-Blas, L.E., Salcedo-Sanz, S., Ortiz-García, E.G., Portilla-Figueras, A., Pérez-Bellido, A.M., 2009. A hybrid grouping genetic algorithm for assigning´ students to preferred laboratory groups. Expert Systems with Applications 36, 7234–7241.

Andria, J., Di Tollo, G., Pesenti, R., 2015. Detection of local tourism systems by threshold accepting. Computational Management Science 12, 559–575.

Bacao, F., Lobo, V., Painho, M., 2005. Applying genetic algorithms to zone design. Soft Computing 9, 341–348.

Begur, S.V., Miller, D.M., Weaver, J.R., 1997. An integrated spatial DSS for scheduling and routing home-health-care nurses. Interfaces 27, 35–48.

Benzarti, E., Sahin, E., Dallery, Y., 2013. Operations management applied to home care services: Analysis of the districting problem. Decision Support Systems 55, 587–598.

Bergey, P.K., Ragsdale, C.T., Hoskote, M., 2003. A decision support system for the electrical power districting problem. Decision Support Systems 36, 1–17.

Bouzarth, E.L., Forrester, R., Hutson, K.R., Reddoch, L., 2018. Assigning students to schools to minimize both transportation costs and socioeconomic variation between schools. Socio-Economic Planning Sciences 64, 1–8.

Bozkaya, B., Erkut, E., Haight, D., Laporte, G., 2011. Designing new electoral districts for the city of Edmonton. Interfaces 41, 534–547.

Bozkaya, B., Erkut, E., Laporte, G., 2003. A tabu search heuristic and adaptive memory procedure for political districting. European Journal of Operational Research 144, 12–26.

Camacho-Collados, M., Liberatore, F., 2015. A decision support system for predictive police patrolling. Decision support systems 75, 25–37.

Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: Nsga-II. IEEE transactions on evolutionary computation 6, 182–197.

Eveborn, P., Flisberg, P., Rönnqvist, M., 2006. Laps care—an operational system for staff planning of home care. European journal of operational research 171, 962–976.

Farughi, H., Tavana, M., Mostafayi, S., Santos Arteaga, F.J., 2020. A novel optimization model for designing compact, balanced, and contiguous healthcare districts. Journal of the Operational Research Society 71, 1740–1759.

Ferland, J.A., Guénette, G., 1990. Decision support system for the school districting problem. Operations Research 38, 15–21.

Goldberg, D.E., Holland, J.H., 1988. Genetic algorithms and machine learning.

Hajiali, M., Teimoury, E., Rabiee, M., Delen, D., 2022. An interactive decision support system for real-time ambulance relocation with priority guidelines. Decision Support Systems 155, 113712.

Holsapple, C.W., 2008. Dss architecture and types, in Handbook on Decision Support Systems 1. Springer, pp. 163–189.

Jahuira, C., Cuadros-Vargas, E., 2003. Solving the tsp by mixing gas with minimal spanning tree. Sociedad Peruana de Computacion, II-3 , 123–133.

Kalcsics, J., Nickel, S., Schröder, M., 2005. Towards a unified territorial design approach—applications, algorithms and GIS integration. Top 13, 1–56.

Kim, M.J., 2018. Multiobjective spanning tree based optimization model to political redistricting. Spatial Information Research 26, 317–325.

Konak, A., 2012. Network design problem with relays: A genetic algorithm with a path-based crossover and a set covering formulation. European Journal of Operational Research 218, 829–837.

Labelle, A., Langevin, A., Campbell, J.F., 2002. Sector design for snow removal and disposal in urban areas. Socio-Economic Planning Sciences 36, 183–202.

Lei, H., Wang, R., Laporte, G., 2016. Solving a multi-objective dynamic stochastic districting and routing problem with a co-evolutionary algorithm. Computers & Operations Research 67, 12–24.

Martinho, A., Alves, E., Rodrigues, A.M., Ferreira, J.S., 2017. Multicriteria location-routing problems with sectorization, in: Congress of APDIO, the Portuguese Operational Research Society, Springer. pp. 215–234.

Moynihan, G.P., Raj, P.S., Sterling, J.U., Nichols, W.G., 1995. Decision support system for strategic logistics planning. Computers in industry 26, 75–84.

Noorian, S.S., Murphy, C.E., 2017. Balanced allocation of multi-criteria geographic areas by a genetic algorithm, in: International Cartographic Conference, Springer. pp. 417–433.

Ozturk, E., Rocha, P., Sousa, F., Lima, M., Rodrigues, A.M., Ferreira, J.S., Nunes,A.C., Lopes, C., Oliveira, C., 2022. An application of preference-inspired coevolutionary algorithm to sectorization, in: International Conference Innovation in Engineering, Springer. pp. 257–268.

Ozturk, E.G., Rodrigues, A.M., Ferreira, J.S., 2023. A new matrix form genetic encoding for balanced, compact and connected sectorisation through NSGA-II. International Journal of Multicriteria Decision Making 9, 250–279.

Pan, Q., Castro-Gama, M.E., Jonoski, A., Popescu, I., 2016. Decision support system for daily and long term operations of the system of milan, italy. Procedia engineering 154, 58–61.

Ramachandra, T., George, V., Vamsee, K.S., Purnima, G., 2006. Decision support system for regional electricity planning. Energy Education Science and Technology 17, 7.

Rest, K.D., Hirsch, P., 2022. Insights and decision support for home health care services in times of disasters. Central European journal of operations research 30, 133–157.

Rodrigues, A.M., Ferreira, J.S., 2015. Sectors and routes in solid waste collection, in: Operational Research. Springer, pp. 353–375.

Saaty, T.L., 1977. A scaling method for priorities in hierarchical structures. Journal of mathematical psychology 15, 234–281.

Saaty, T.L., 1980. The analytical hierarchy process: Planning, priority setting, resource allocation. McGrawrHill International Book Co., London, England .

Saaty, T.L., 2008. Decision making with the analytic hierarchy process. International journal of services sciences 1, 83–98.

Xie, S., Ouyang, Y., 2016. Railroad caller districting with reliability, contiguity, balance, and compactness considerations. Transportation Research Part C: Emerging Technologies 73, 65–76.

Yanık, S., Kalcsics, J., Nickel, S., Bozkaya, B., 2019. A multi-period multi-criteria districting problem applied to primary care scheme with gradual assignment. International Transactions in Operational Research 26, 1676–1697.

Zhang, K., Yan, H., Zeng, H., Xin, K., Tao, T., 2019. A practical multi-objective optimization sectorization method for water distribution network. Science of The Total Environment 656, 1401–1412.

Zou, X., Cheng, P., An, B., Song, J., 2016. Sectorization and configuration transition in airspace design. Mathematical Problems in Engineering 2016.