ZoneBadger: A Heuristic Model for Service Territory Design at Turf Badger's Stevens Point Branch

Adam Bruce

Department of Data Science, University of Wisconsin – Green Bay

DS785: Capstone

Abstract

Service-focused businesses dominate in mature world economies (Frei, 2008). As such, these businesses have sought competitive advantages by developing services customers want at affordable prices. Therefore, improving profitability, efficiency, and delivery through operations management is integral to their success. For Turf Badger, the opportunity to apply data science toward improving technician-specific operations was thus viewed valuably. Here, data analytics related to clustering and Genetic Algorithms were used to produce a heuristic approach for solving a combinatorial optimization-based Green Field Planning Territory Design Problem (Sharifi, 2015). The solution addresses the strategical assignment of technicians based on balance, contiguity, and compactness. Data on 2,282 active, recurring customers for Turf Badger's Stevens Point office was used to develop the final solution. Ultimately, the best solution here minimized the sum of standard deviation and radial compactness objective function across six service zones by adjusting centroid locations via a Genetic Algorithm. Initial zone centroids were provided to the optimizer via a stochastic custom machine learning K-Means Clustering derived starter solution. The custom starter method utilized Manhattan "travel" Distance as opposed to Euclidean Distance as its assignment mechanism. Ultimately, customer zone assignments were extracted from the best model of the heuristic, named ZoneBadger, and technician income and service capacity objectives were checked. Finally, the zones were integrated with Turf Badger's in-house routing software, FieldRoutes. The results streamline current technician operations and provide the potential for future business benefits like designing high-quality job descriptions, targeted marketing campaigns, and sales forecasting for each zone.

Keywords: Heuristic, Clustering, Genetic Algorithm, Green Field Planning Territory Design Problem, Radial Compactness, Objective Function, Manhattan Distance

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Background

In the service industry, determining how to utilize resources efficiently has long been considered paramount to sustained success (Wright & Mechling, 2002). Therefore, the need for data analytics to optimize resource deployment persists indefinitely throughout the business life cycle. For the mid-sized service-orientated business Turf Badger, operations management identified the allocation of their technician resource as a key area in need of such optimization. As an S corporation based in Wisconsin focusing on lawn, mosquito, and pest services, technicians directly link service offerings and their customer base. Consequently, ensuring they are assigned routes based on potential income, work volume, and geographical travel considerations provided an opportunity to improve worker and customer experiences.

Operating ten branches across four states in the US, the strategy for management is to expand a technician resource optimization solution from a starter branch pilot project to every location in time. Ultimately, a branch founded in 2022 in Stevens Point, Wisconsin, was chosen as the location for the initial project here because it was relatively young and had a smaller active customer base than other established branches. Such young, smaller locations reduce the risk of failure for pilot projects and allow for more effortless solution refinement before scaling (Ashkenas & Matta, 2021).

In the end, the project aim was to create zones that satisfied three principles common to territory design problems: balancedness (being similar in size according to some measure), contiguity (being able to travel to all customers in the territory without leaving it), and compactness (being somewhat round, undistorted, and without holes) (Moreno et al., 2020). Achieving balancedness was primarily addressed through equalizing income. Meanwhile, contiguity was addressed through the Manhattan Distance geographical clustering component. Finally, compactness was addressed utilizing a measure of tightness from points to a common center known as radial compactness (Bacao et al., 2005). Collectively, the newly designed zones will streamline operations, improve consistency of service delivery, and be exploitable for future business projects.

Meeting this vision required collaboration throughout the entirety of the data analytics process. Specifically, it was necessary to understand both the expected number of zones to be deployed and the value of technician income through upper management's lens, which routinely handles these concerns. Furthermore, clear communication of how the problem would be addressed through data analytics intervention was necessary for developing trust in the end product. This collaboration aided in understanding the business problem through positive social capital between the researcher and management, which significantly increased the chances of success for the project (Hagen & Hess, 2021).

Business Problem

Turf Badger's technician service areas needed improvement because their internal software, FieldRoutes, relied solely on non-overlapping time availability to route technicians daily for servicing customers. This has caused inefficiencies in resource deployment due to overlapping routes, unchecked workloads, unbalanced technician incomes, and customer dissatisfaction from inconsistent service. In addition, the combinatorial complexity and time consumption of creating such zones by hand or exhaustive search for a large customer base, some 2,200 plus for just this branch, further exacerbated the problem. Therefore, data analytics intervention was necessary to identify and implement a means for solving the problem.

Project Objectives

Three objectives existed for this project. The first was to define optimal "balanced" service zones for technician income that meet a workload maximum. By collaborating with management, it was determined that six zones, each with a maximum perspective number of services per month of 260, would be optimal. However, most importantly, the yearly salary for the six zones needed to be as close as possible to the mean available amount of $24,999.

The next objective was to create a heuristic approximate optimization algorithm based on balancedness, contiguity, and compactness. Such concepts are integral to Green Field Planning Territory Design Problems and can be achieved using methodology employed for this problem and in other similar areas like clustering (Duque et al., 2012). Ultimately, achieving this objective required consideration of numerous techniques from subject areas like metaheuristics and machine learning. For achieving balance, the metaheuristic Genetic Algorithm (hereafter GA) was used to minimize a fitness function, the standard deviation of technician income differences between the six zones, as a primary objective function. Sharifi and Murphy (2017) showed that standard deviation measures variance from the mean, equating to similarity between groups when optimizing with GA (Sharifi & Murphy, 2017). Thus, the smaller the standard deviation, the more balance in income gained.

To achieve contiguity, the partitional distance component of clustering was employed (Xu & Tian, 2015). The geographical data was forcefully partitioned into contiguous shapes by enforcing assignment based on the similarity measure of Manhattan Distance from a point (customer) to its closest cluster centroid. Manhattan distance averages about 25% larger than Euclidean “straight line” Distance, which has been shown to equate to the difference of traveling on perpendicular city and rural roads (Smith, 1979). Thus, efficient and realistic geographical travel considerations were produced using this distance measure.

To ensure optimal compactness, an additive term for radial compactness was built into the standard deviation GA fitness function. Radial compactness is defined as the sum of the distances from each point in a cluster to its centroid (Bacao et al., 2005). Thus, the sum of Euclidean "circular" Distance for each point of a cluster are summed for all clusters to essentially add a penalty onto the objective fitness function. Thus, less round, tight clusters had higher radial compactness and worse fitness values. This created a trade-off between balancing income and ensuring compactness to create the most optimal final zones according to this primary objective.

The final objective was to produce visualizations for the optimized customer assignments from the final model and adapt the results into FieldRoutes for use by branch management. The models and visualizations were produced utilizing Python (version 3.11.0) and included supporting visuals for meeting the constraints. Adaption into FieldRoutes was accomplished by utilizing its "Visual Grouping" capabilities (Chaney, 2021). To achieve these objectives, data was procured utilizing the reporting tool from FieldRoutes for active, recurring customers as of 1/1/2025. By working with Director of Operations Tricia Gruber, data entry errors and management test cases were identified, fixed, or removed. The final dataset, after cleaning, included ten columns of information on 2,282 customers.

Project Significance

Collectively, the product for this resource optimization project offers several significant use cases for Turf Badger. First, it will allow the business to provide consistent service for current and future customers by having the same technicians repeatedly servicing customers in their assigned zone(s). Service consistency is a key to successful service businesses; thus, this use case has high value (Frei, 2008). In the case of hiring new technicians for any zone(s), it allows for high-quality job descriptions with specific, measurable employee tasks to be created. Failure to attract high-quality job candidates is attributed to 14% of business failures, making this use case valuable (Morelli, 2023).

Furthermore, the project allows Turf Badger to route zones efficiently. Accounting for geospatial distance metrics reduces fuel consumption and time between tasks. This will increase profitability and improve customer experience, which are invaluable to businesses. Next, the balancing of the project end-product allows Turf Badger to promote equity and fairness in income for salaried service positions without pouring resources into costly audits. Salary equity is critical to talent retention, high morale, and churn reduction. As such, it influences employees' decisions to stay with companies (Nagele-Piazza, 2020).

Finally, optimizing service territories makes evaluating technician performance easier. Under disproportionate, non-optimal conditions, disparities in efficiency and customer satisfaction with technicians cannot be judged fairly (Hess & Samuels, 1971). Thus, data analytics optimization techniques allow Turf Badger to examine these metrics more objectively. Collectively, these use cases provide clear evidence of the impact and value of utilizing data science to solve this service industry problem.

Definition of Terms

The term Green Field Planning Territory Design Project is utilized frequently in this project. Defined by Sharifi (2015), the term refers to a territory design problem in which neither the optimal location of (service) representatives nor an existing sales territory structure exists (Sharifi, 2015). These problems require decision-makers (operations management here) to decide the number of territories needed. Then, the best assignment of representatives (service technicians here) is found through developing heuristics to build acceptable territory structures. This approach was central to designing the data analytics methods for addressing Turf Badger's problem here, and it is thus a common descriptor throughout the project.

Assumptions

In the present study, it is assumed that a general knowledge of the processes behind GAs is known. Namely, the concepts of population generation, mutation, crossover, elitism, and parenting are assumed to be understood. These five concepts, based around the ideas of evolution by natural selection, were central to the development of the GA model employed in the final heuristic of this study. While the concepts are addressed to some degree in the *Design Phase* sub-section of Chapter 3, they are not reviewed in depth. Thus, in the case of unfamiliarity, it is recommended to review them in Oliver Kramer's (2017) book *Genetic Algorithm Essentials* (Kramer, 2017).

Limitations

Three limitations exist for this study. The first involves choosing a metaheuristic for building the optimization model. GA was chosen among the four feasible choices for solving the problem because previous work has shown GA performs well when adapted with the central clustering methodology employed for this study (Maulik & Bandyopadhyay, 2000). Also, GAs adapted well to this study's custom-built objective minimization function for standard deviation and radial compactness.

Another potential limitation of the study is that the optimal solution may have occurred at a local minimum. However, GAs escape local minima by utilizing concepts of natural selection (Gupta & Ghafir, 2012). By iteratively tuning model parameters, an exploration of diverse populations was performed. Then, by making minor adjustments to the parameters from the solution that minimized the objective fitness function, the solution space was exploited to increase the likelihood of reaching the global minimum. These processes allowed for adequate exploration of the solution space and aided in dealing with escaping potential local minima. Thus, it is highly probable that the GA metaheuristic is at or near the optimal solution for this problem.

Finally, the methodology employed to solve the problem differs from the most common approach in the literature. Known as the Capacitated Clustering approach, the problem is defined and solved with a combined metaheuristic and integer programming methodology (Shieh & May, 2001). However, this methodology is mathematically and computationally complex. Furthermore, it is time-inefficient for large problems due to the employment of branch and bound methods (Maulik and Bandyopadhyay, 2000). Thus, due to the time limitations of this research, the alternative clustering methodology used here was chosen. Ultimately, it is unclear which methodology produces the most optimal solution based on the current literature, but the present research provides the opportunity for result comparisons in the future.

Conclusion

Undeniably, the original state of technician routing for Turf Badger's Stevens Point branch failed to optimize resource use. This made the prospective for intervention with data analytics very valuable to upper management. Not only are the business use cases of this project numerous, but they also have the potential to be used at scale for optimizing all ten branches in the future. Collectively, a heuristic methodology was developed to solve the optimization problem with combined GA and clustering techniques. These methodologies were chosen for their abilities to solve the problem in time and memory-efficient ways.

Overall, the project aimed to design zones based on balance, contiguity, and compactness. Balance was achieved by assigning customers to zones through the minimization of standard deviation for income between clusters in the GA objective function. Meanwhile, contiguity was achieved by utilizing the Manhattan Distance similarity component of clustering algorithms. Finally, compactness was achieved by adding a term for radial compactness into the objective function. Moving forward, a literature review of the applications for heuristic algorithms in territorial design-specific problems is performed.

Chapter 2: Literature Review

Introduction

The heuristic solutions to solving territorial design problems are vast. Despite being unique to the research problem, each generally applies the same principles in one form or another. They form larger grouping units from smaller ones, like customers, zip codes, or counties, and then adjust the groupings to meet pre-determined management planning criteria(s). The earliest algorithms coincide with the rise of commercial computers in the mid-to-late 1960s. These algorithms focused heavily on finding approximately exact solutions to problems with Linear/Integer Programming (hereafter LI Programming). Still, the large and complex search spaces proved difficult and exhaustive to explore.

Eventually, the Capacitated Clustering Approach was proven to be NP-Complete in a way that increased exponentially for time and memory complexity (Garey & Johnson, 1979). This led to a significant shift towards more time-relaxed, hybrid heuristic approaches to finding reasonably optimal solutions using emerging metaheuristic techniques in the 1990s. The following review explains the methodologies employed in solving territory design and similar problems chronologically. The studies for the review were sourced predominantly from academic publishing companies, like Elsevier, JSTOR, and Research Gate, after searching for leads with Google Scholar.

LI Programming Beginning

One of the earliest territory design algorithms was based on concepts employed in political districting (Hess & Samuels, 1971). The heuristic, coined GEOLINE, calculated a matrix of squared distances between manually inputted trial territory centers as a cost function before solving a Linear Programming problem. Researchers noted that management's decision of an activity measure value for achieving balancedness was critical to the success of this early heuristic. This was because the activity measure was utilized as the constraint for the model, and thus, the resulting territories were heavily influenced by its value(s).

At the core, the linear programming behind this heuristic involved creating an objective function that was subject to constraints for decision variables $X_1, X_2, \ldots X_n$. In a simple two-dimensional example, the constraints overlap to form a feasible boundary region through which the objective function passes. The point(s) at which the objective function is maximized or minimized at the intersection with this boundary provides the optimal solution values for the decision variables of the problem (Hillier & Lieberman, 2015). In this case, the decision variables represented the proportion of the j_{th} geographical unit assigned to the i_{th} territory.

In practice, GEOLINE was utilized by CIBA Pharmaceutical Company to realign their existing sales territories for selling medications to physicians. Utilizing doctor count as the activity measure, 40 territories were created from 5,000 zip code decision variables. Across three output solutions, the doctor count in every territory never deviated greater than 10% from the mean. Additionally, GEOLINE was used by the IBM World Trade Organization at the time to successfully realign its typewriter repairmen service territories based on workload. Overall, this foundational heuristic would pave the way for several future design improvements, which will be discussed briefly before the rise of metaheuristic procedures.

LI Adjustments

Many subsequent studies were quick to add onto Hess and Samuels's work. One proposed an improved territory design heuristic where the decision variables included the assignment of salespeople to subareas, the number of trips a salesmen should make to their assigned areas, and the amount of time the salespeople should spend in their subareas (Lodish, 1975). This design's unique contribution was implementing binary integer programming for assignment with decision variables. Ultimately, when used to redesign four territories for a large industrial products firm, the model improved sales by $86,000 and profit by $17,000.

Another heuristic was later proposed for mean workload equalizing with geographic considerations (Segal & Weinberger, 1977). The approach, designed to create telephone installation and repair service zones, was the first to deem standard deviation as an appropriate objective function for minimizing differences in activity (workload) from the mean across clusters. In a trial run for designing new territories for a Pennsylvania-based company, a decrease in absent rate from 30 days in 1974 to only 9 days in 1975 over the same five-month period was observed. The researchers described this result as a significant improvement in workforce morale.

The last heuristic to discuss proved that territorial design problems were definable as Capacitated Clustering Problems (Mulvey & Beck, 1984). A significant improvement offered in this heuristic was its use of clustering methodology to adjust assignments. Essentially, the approach utilized a K-Medoids Clustering algorithm by randomly assigning units as centers and then iteratively adjusting the centers to the median units of each cluster after completing Euclidean Distance-based assignment. This process was repeated until either no improvement in the objective function or no changes in the set of ρ median centroids occurred post-assignment.

Perhaps the most important contribution the researchers was their identification of the territorial design problem as being related to the Facility Location and Traveling Salesmen Problems (Mulvey & Beck, 1984). At the time, these problem types were heavily studied because of their business value. As a result, new techniques emerged to handle their complex nature. Among these techniques were metaheuristics, with Simulated Annealing being first described and then applied successfully to a 6,000-city Traveling Salesman Problem in 1983 (Kirkpatrick et al., 1983). With the link made, it was inevitable that territory design heuristics would be improved with these emerging metaheuristic methods.

Metaheuristic Application

Since clustering with metaheuristic GA is central to this study, the following sections emphasize problems related to GAs use in combinatorial optimization of clusters and territory design. Despite the popularity of territory design problems throughout the 1980s and 1990s, the first use of GA to aid in solving them did not occur until 2005. In the interim, several clustering adaptations that avoid LI Programming with GA were developed. One of the first, called GA-Clustering, was developed to minimize within-cluster spread and provide reasonable solutions regardless of initial starting centers (Maulik & Bandyopadhyay, 2000).

To achieve these goals, the authors of the GA-Clustering heuristic integrated components of the K-Means Clustering algorithm with the GA metaheuristic. An important note from Maulik & Bandyopadhyay's development phase was that the branch and bound clustering technique commonly employed in LI Programming solutions was purposefully avoided because "…the number of nodes to be searched becomes huge when the data set becomes large" (p. 1456). Such a reality justifies the avoidance of LI Programming in the heuristic methodology specific to this study.

A key component of GA-Clustering was chromosomes being encoded as floating-point potential cluster center values. Thus, for a GA population of size P, each of P chromosomes in an N-dimensional space with K cluster centers would have length $N*K$. The points to be clustered are assigned to each of K cluster centers based on the shortest Euclidean Distance, and the mean cartesian values of each cluster become the new center. Then, the objective function for each chromosome is calculated as one divided by the sum of the absolute Euclidean Distance of each point to its respective cluster center. Therefore, cluster centers that increase the similarity of groups are viewed as the most fit by GA-Clustering.

After each iteration, GA-Clustering would employ the GA evolution and natural selection-based methods of mutation, crossover, and elitism to form new trial centers for clustering randomly. This allowed for the exploration of the search space and the exploitation of reasonable solutions. In practice, GA-clustering was compared against traditional K-Means Clustering through application to the well-known iris dataset for creating clusters for three species. Utilizing five runs for populations of size 100 with GA-Clustering and up to 1000 iterations for K-Means, results showed GA-Clustering was able to achieve the global minimum of 97.10077 in all five iterations while K-Means failed to achieve this value even once.

Overall, GA clustering serves as the first application of the GA metaheuristic for a problem type comparable to territorial design. GA-Clustering avoidance of the complexities behind LI Programming makes it an enticing methodology for application to this study. The final heuristic here employs several ideas put forth from the GA-Clustering approach. However, some variations in the design were produced based on findings from two other territorial design-specific problems that will be discussed next.

GA Variants

GA was first applied specifically for territory design with the aim of equalizing populations (Bacao et al., 2005). The heuristic involved adjusting cluster centers within the cartesian study region utilizing GA mutation and crossover operators. Uniquely, the objective fitness function combined population balance with compactness by incorporating an additive penalty for compactness based on the sum of Euclidean “circular” Distances between centroids and their assigned units. This term, called radial compactness, would be built into the heuristic of this current study to address the compactness objective.

In practice, the heuristic was utilized for a political districting territorial design problem in Portugal (Bacao et al., 2005). Ultimately, the goal was to develop 93 contiguous electoral districts for the Lisbon region where the population was within $\pm$ 25% of the mean. Compared to a simulated annealing algorithm, ZDES, that achieved a minimum combined fitness of 9,642.6, the GA heuristic found a more optimal solution at a fitness of 6,482.04. The researchers attributed this result to the improved approach of letting centers be located anywhere within the search space and utilizing GA to explore it. They also noted that the results indicated that GA heuristics provide great potential for improving solutions to territory design problems.

In a final approach, the potential of GA was further exploited by using it to develop a heuristic solution to a Green Field Planning Territory Design Problem for an agricultural machinery manufacturer in Germany (Sharifi & Murphy, 2017). Tasked with creating 27 sales territories for 402 administrative districts, the researchers began by clustering the districts based on distance to the nearest salesman. Next, an initial GA minimized the objective fitness function of the sum of travel times within districts through reassignment. This initial solution was then used as a starting point for the heuristic to build from.

The core GA heuristic component of the approach worked with an objective fitness function based on standard deviation (Sharifi & Murphy, 2017). This approach meant that the lower the fitness, or in other words, the lower the standard deviation, the more equality/balance gained between territories. Ultimately, the GA decreased the standard deviation in workload from the starter solution at 6000 down to 50 by reassigning units to different territories through mutation and crossover. As for contiguity and compactness, it was noted that the initial clustering solution guaranteed the satisfaction of these conditions.

Conclusion

Based on the literature, many heuristic designs are reasonably good at solving territorial design problems. Temporally, it was shown that the methodology has evolved from strictly LI Programming-based heuristics to notedly less complicated combined clustering and metaheuristic approaches. Here, the combined approaches that specifically utilized Genetic Algorithms were highlighted. However, the studies where these approaches were applied were not always for territorial design. This was done to highlight the adaptability of combined approaches from any combinatorial problem to the territorial design problem. Ultimately, this study's final heuristic design is based on a fusion of the final three reviewed sources from the modern metaheuristic era and is outlined in the proceeding chapter.

Chapter 3: Data Science Applications

Introduction

Based on the literature, two options existed for designing this project's heuristic. That is, it would either employ LI Programming or clustering methodologies combined with GA for optimization. Initially, LI Programming was attempted, but several setbacks led to the clustering approach being utilized instead. The biggest issue with the LI Programming approach was its reliance on hard constraints. When solving combinatorial territorial design problems, objectives are coded as hard constraints or made part of the optimization objective function (Bacao et al., 2005).

With hard constraints, the complexity of the algorithm design increases significantly. While pseudo-code is available, it is complicated to follow and even more complicated to adapt to new problems. Additionally, hard constraints require branch and bound methods that significantly increase the time needed to reach solutions. For example, one branch and bound LI Programming approach took over three and a half hours to find a single solution for a dataset with 812 observations (Shieh & May, 2001). Alternatively, pseudo-code for clustering-based approaches is easily adaptable, more conceptually understandable, and requires less time to reach solutions for large datasets.

The reason for this is directly related to the approach of building "constraints" into the GA objective fitness function for clustering heuristics. When designing the objective fitness function, the aims of balancedness, contiguity, and compactness can be met through the minimization approach of GA. For example, building standard deviation to minimize differences in technician income from the required mean accomplishes meeting the constraint without hard coding it as a bounded region. While this means solutions are not guaranteed to meet the constraints exactly as in LI Programming, it typically produces "good enough" solutions. In this way, the clustering approach can be viewed as one that implements quasi-hard constraints (Bacao et al., 2005). Such a reality rationally made the clustering-based heuristic design approach best suited for this project.

Overall, the final design was like that of Maulik and Bandyopadhyay's heuristic in that it encoded chromosomes as floating-point potential cluster center values of length $N*K$. Further, mutation, crossover, and elitism were performed to randomly form new trial centers before cluster assignment based on the nearest center distance. Additionally, the final heuristic here was like that of Bacao et al. in that the collective fitness function combined a balance component with a Manhattan Distance version of their radial compactness additive penalty. Finally, it was like that of Sharifi and Murphy's heuristic in that an initial clustering solution was utilized as a starting point for center adjustment, and the income balance for the objective fitness function was standard deviation. Upon completion, the final heuristic was named ZoneBadger.

Data Overview

Data on active, recurring customer accounts for Turf Badger's Stevens Point branch were utilized to create the territorial zones in this study. The data was collected by generating a .csv file report with Turf Badger's FieldRoutes company software system and was current as of 01/01/2025. The original dataset, called 'customers.csv,' contained 2,311 rows representing customers with twelve descriptor columns in: 'Customer ID,' 'Last Name,' 'First Name,' 'Sold Date,' 'Company Name,' 'Address,' 'City,' 'Latitude,' 'Longitude,' 'Subscription Contract Value,' 'Subscription,' and 'Subscription Category.' The initial generated report was approved by Director of Operations Tricia Gruber. While no aggregation procedures were performed on the data, extensive cleaning and processing measures were required.

Data Cleaning & Processing

Cleaning and processing are essential for gaining management's trust and confidence in the results of data analytics projects (Sweary, 2019). Achieving a cleaned and processed dataset required utilizing functions in Python's Pandas and Numpy packages. Initial cleaning for the 'customers.csv' dataset began with changing the names of columns to simplify operating with them. All white spaces were filled with underscores to connect words, and the columns 'Customer ID,' 'First Name,' 'Last Name,' and 'Subscription Contract Value' were changed to 'Cust_ID,' 'First,' 'Last,' and 'Contract_Value' respectively. Next, missing data points were identified for substitution or deletion.

Exploration of the 'First' and 'Last' columns with null data indexing resulted in the identification of several rows containing missing data for all other columns. Management identified these rows as system test cases, as they contained only the names of employees, and thus, they were removed. The only other column with missing data, found with the same indexing approach, was 'Company_Name.' Only three observations existed in this column, namely Hilton Garden Inn, Bridge Street Partners, LLP, and Lazy Meadows Homes LLC. In these cases, the 'First' and 'Last' columns contained the names of company owners. Since most rows were non-business customers, the 'First' column for these three cases was replaced with the 'Company_Name' value, and the 'Last' column was replaced with a blank string. Afterward, the 'Company_Name' column was dropped from the dataset altogether.

For simplification, the 'First' and 'Last' name columns were combined into a single column, called 'Customer,' with string concatenation, and the original two columns were dropped. Next, data entry errors needing to be addressed were found based on extracting management-defined information to meet project aims. For example, management requested a list of all unique subscriptions for review. After obtaining such a list, it was observed that the 'Subscription' column needed refinement.

Management identified entries for 'X Flea & Tick' and 'Lawn - FERT TEMP' subscriptions as sales representative data entry errors. The company no longer serviced these subscriptions, so they were changed to their new classifications of 'Flea & Tick' and 'Fertilization,' respectively. Lastly, rows for customers with the subscription types of 'bed bug,' 'WL - Foundation Exclusion,' 'Flea & Tick,' 'WL - Wildlife/Full Exclusion Services,' and 'WL - Rodent/Wildlife Trapping' were dropped because these were scheduled single service contracts for the future that management did not want influencing zone totals.

The column for 'Address' was also dropped from the dataset to protect sensitive customer information. Unfortunately, because the ZoneBadger heuristic required 'Latitude' and 'Longitude' information on customers for clustering, these two sensitive columns could not be removed. After cleaning and processing, the final dataset contained nine columns and 2,282 rows. This dataset, aptly named 'SP_customers_clean,' would be the source for implementing the ZoneBager heuristic to solve the problem. However, prior to implementation, feature engineering and exploratory data analysis would need to be performed.

Feature Engineering

Several new features were built into the dataset with feature engineering techniques to help meet income and service balance objectives. Initially, the aim of service balance required modification of the ‘Sold_Date’ column. The owner of the Stevens Point branch, James Bruce, provided a list of total services per year for each of the eighteen remaining unique customer service subscriptions outlined in Table 1 below. This made it possible to calculate the months each subscription would be serviced based on the month the customer was sold their subscription.

Table 1

Subscription Types with Services Per Year at Turf Badger's Stevens Point Branch

Subscription Name	Services Per Year
Pest – Quarterly Pest Control	4
Mosquito – Monthly	5; May Through September Only
Pest – Tri-Annual Pest Control	3
Fertilization	5
Pest – Rodent Bait Box	4
Pest – One Time Pest Control	1
Pest – Bi-Monthly	6
Lawn – Grub Preventative	1
Lawn – Fungicide	1
Pest – Bi-Annual	2
Pest – Eave & Overhang Treatment	4
Commercial Pest	12
Mosquito – Tri-Weekly	8
Lawn – Aeration	1
Pest – German Cockroach	12
Commercial RBB/Trapping	12
Lawn – Grub Control	1
Pest – Organic Badger Service Plan	6

For example, a 'Pest - Quarterly Pest Control' subscription with four services per year sold to a customer in January would be serviced once in January, April, July, and October, respectively. Thus, only the subscription purchase month was needed from the 'Sold_Date' column containing the year, month, and day of sale. Therefore, this column was split into 'Start_Month,' 'Start_Day,' and 'Start_Year,' and then the original 'Sold_Date' column, 'Start_Day,' and 'Start Year' were then all dropped from the dataset.

The next step of the service feature engineering phase was adding a column for each month of the year in string format (i.e., '01' for January, '02' for February, and so on…) to the dataset. A column for each customer's summed row count of monthly services also needed to be added. To achieve this, a custom-built function called 'monthly_service_count' was built. This function first took in the dataset and converted all the 'Subscription' and 'Start_Month' column values for each row into tuple pairs in a list called 'sub_start.' Then, a for loop through 'sub_start' was combined with inner and outer conditional statements to match the 'Subscription' value at index 0 with mappings for service totals throughout the year based on the 'Start_Month' value at index 1.

For example, the first customer in the dataset had a 'Subscription' of type 'Pest - Quarterly Pest Control' and a 'Start_Month' value of "11". Their tuple index 0 matched their subscription type value in the outer for loop. Then, their index at 1 matched with the inner conditional statement for month "11". This resulted in a dictionary of service months and total services in the month being created (i.e. {"02": 1, "05": 1, "08": 1, "11":1} for the customer example based on Table 1). Each dictionary was then appended to a list returned by the function called 'Services_Per_Month' such that one dictionary for each row (customer) was in the list.

Notedly, management said the subscription 'Mosquito-Tri Weekly' was eight services from May to September regardless of the start month. Thus, all customers with this subscription had a final dictionary of {"05": 2, "06": 1, "07": 2, "08": 1, "09": 2}. At this point, a string column for each month with all zeros for every row was added to the dataset. Then, an outer for loop through the dataset rows was utilized to pull the dictionary out of 'Services_Per_Month' that matched the index for the row. An inner for loop then moved through the key/value pairs of the matching dictionary and replaced the column values of the original dataset with the service value for each key.

As an example, the index for the first customer was 0, and the matching dictionary at index 0 in 'Services_Per_Month' had key/value pairs of "02": 1, "05": 1, "08": 1, and "11": 1. Thus, the value of row 0 column "02" was changed from zero to one and so on. Once all service counts were filled, the last step was to add a column called 'Yearly_Service_Total,' with the row sums for services each customer required per year to the dataset.

The final feature engineering process centered on the technician income balancing objective. Management determined that if a technician serviced the entire duration of a customer contract, they would take home 14% of the total contract value as income. Thus, by multiplying the 'Contract_Value' column by 0.14 (14%), a new column of the total technician income for a contract, called 'Tech_Takehome,' was engineered for the dataset. After completing feature engineering, the final dataset used for exploratory data analysis contained 23 columns and 2,282 rows.

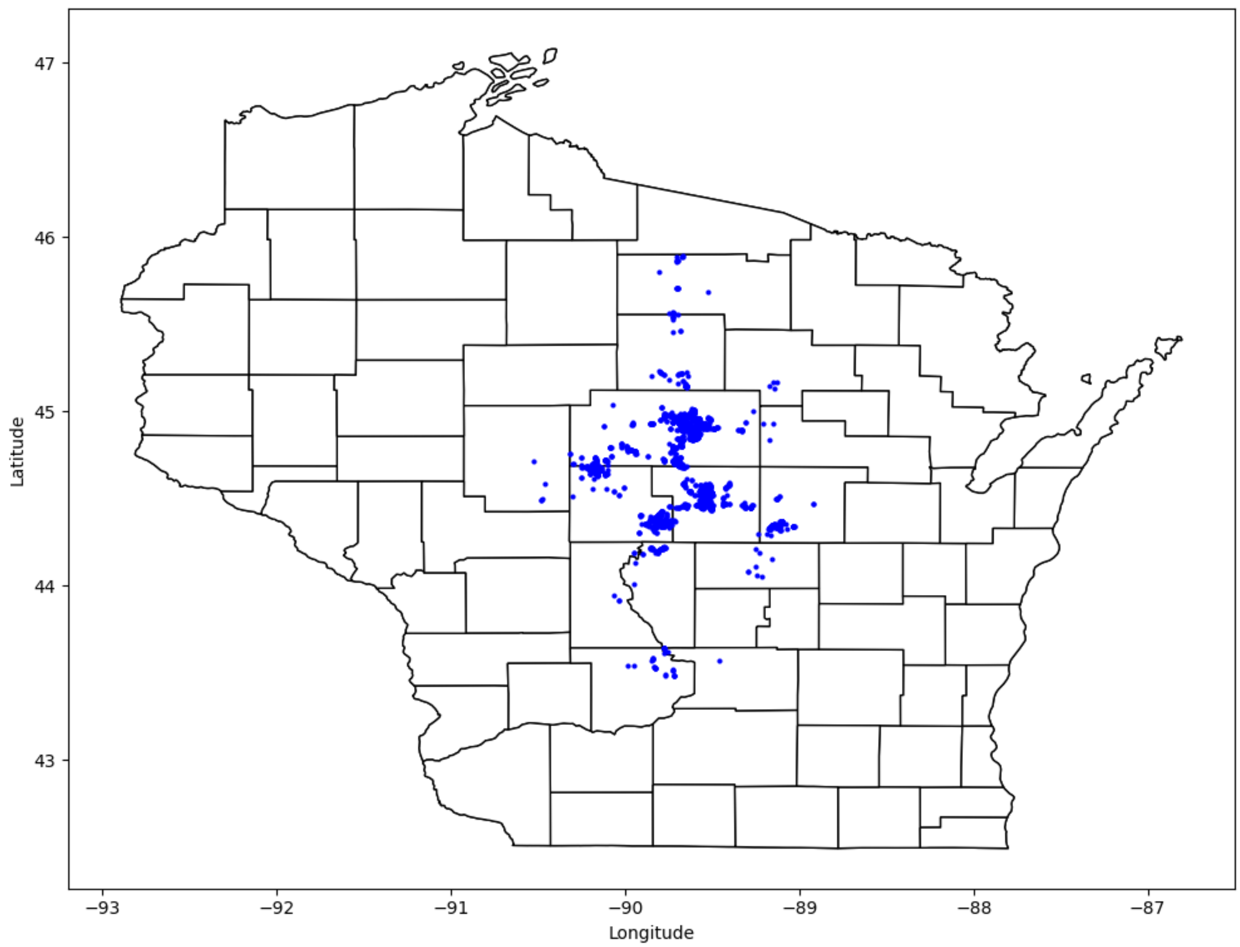
Exploratory Data Analysis

Since their introduction in the 1970s by statistician John Tukey, the techniques of exploratory data analysis (hereafter EDA) have become a principal component in data analytics studies. Collectively, EDA helps identify patterns, relationships, and basic mathematical tendencies for variables of interest. Typically, EDA is performed to show how these variables comply with underlying assumptions of modeling techniques to justify the correctness of utilizing the model. However, this study used two methods, clustering, and GA, with no underlying assumptions.

Therefore, EDA was instead used here to produce a starter Manhattan Distance K-Means solution for later comparison against the optimal ZoneBadger heuristic to measure performance in relation to the income and radial compactness objective fitness function. Before creating the starter solution, it was necessary to build a visual for the customer locations on a map of Wisconsin. To construct the map, state and county files available in Python's geopandas package were utilized (Ruhl & Solumine, n.d.). The final visualization can be observed in Figure 1 below.

Figure 1

Mapping Stevens Point Branch Customers to their Locations in Wisconsin

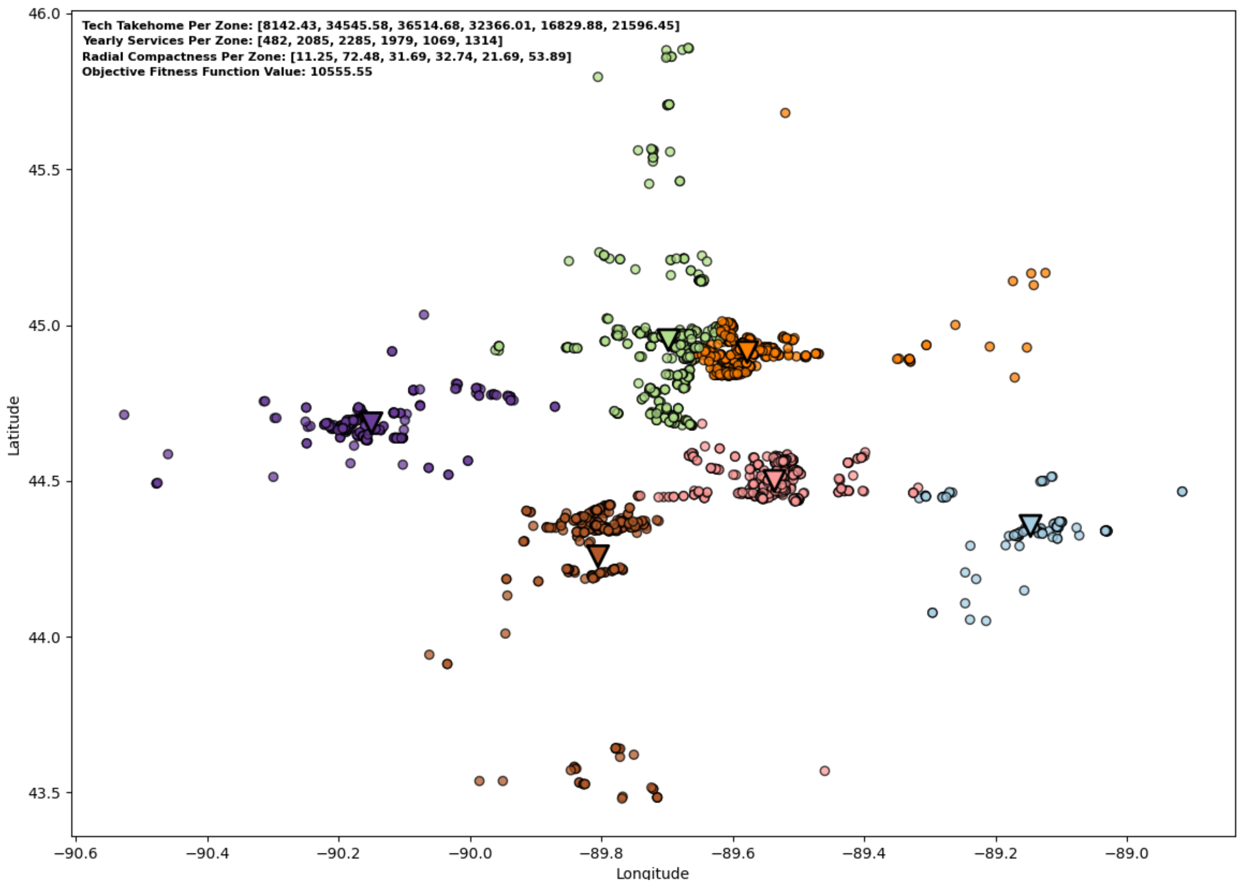


The mathematical procedure behind the traditional clustering process of K-Means can be found in Equation 1A of Appendix A. This approach was chosen to produce a starter solution because the distance partitioning component ensured contiguity. However, traditional K-Means assigns each of *n* points to a cluster center, C_k, based on the shortest Euclidean Distance from the point to each cluster center. In the K-Means approach for this study, Manhattan Distance was instead employed to generate a “travel distance” starter solution as in the final ZoneBadger heuristic. An explanation of the Manhattan Distance equation for this project can be found in Equation 2A of Appendix A.

Employing the Manhattan K-Means approach required utilizing a custom function called ‘k_means_custom.’ For the function, a maximum number of iterations of 100 was set, and after running, centroid labels and cluster assignments were extracted. By utilizing the cluster assignments, radial compactness for each cluster was calculated. The formula for this calculation can be found in Equation 3A of Appendix A. Additionally, the assignments were used to group and calculate the technician income and service totals for the clusters. With these values, the objective fitness function value for the starter solution was found at 10,555.55 based on Equation 4A in Appendix A. Figure 2 below shows a map of the starter solution with color-coded zone assignments for the customer location points.

Figure 2

Custom Manhattan K-Means Clustering Customer Zone Assignments Starter Solution with Centroids Denoted as Triangles



Through EDA, understandings of the data's shape, mapping abilities, and performance in an initial Manhattan Distance K-Means starter solution were gained. Ultimately, while the starter solution had a very low sum of radial compactness value, due to the distance-based assignment of K-Means, at 223.74, it did not account for technician income balance and thus had a high standard deviation in the values between zones at 10,331.80. Further, two of the six zones in this solution did not meet the minimum income constraint of 20,000, and none were considerably close to the goal of 24,999.17 USD. On this note, the largest percent deviation from the mean for the solution was 37.44% for the zone with an income value of 36,514.68. Thus, while the starter solution from the EDA process helped gauge the performance of traditional, non-metaheuristic optimized clustering approaches for the problem, it was far from achieving the project objectives. Therefore, it would serve solely as a comparison for the ZoneBadger solution, the development process of which is highlighted next.

Heuristic Modeling

Collectively, the processes behind designing, developing, and evaluating the ZoneBadger heuristic spanned four months (December 2024 – March 2025). Throughout this time, many trial-and-error tests were performed. The processes would improve through iteration and ultimately lead to an accepted solution. Here, the inner workings of the heuristic are first addressed in the design phase before moving into the development procedure of hyperparameter tuning and ending with the decision of stopping criterion.

Design Phase

Most of the methodology behind ZoneBadger was custom-built, but the framework was based on an example video for creating a Capacitated K-Means Clustering heuristic (Boddapu, 2021). This initial heuristic framework worked to equalize the count of individuals assigned to K clusters. Essentially, the nearest Euclidean Distance assignment was performed alongside the minimization of an objective fitness function value for the sum of squared differences between the mean number of individuals and the current count for all clusters. Python's geneticalgorithm package was used to achieve minimization.

Working from this framework, the first step was to define a function, called 'cluster_process_stdev_radialcompactness,' which performed the heuristic procedure. The function was paired with the geneticalgorithm package to generate random populations, P, of individuals, each with 2*K centroid adjustment genes in the range of $[-1.5, 1.5]$. This range was chosen because the difference between the maximum and minimum longitude/latitude values was 1.61 and 2.41, respectively. Thus, minimal adjustments ensured poor, time-wasting solutions for centroids far outside the range of customer locations were avoided. Every member of the generated population next had their adjustment genes combined with the corresponding centroids from the initial Manhattan K-Means Starter Solution to form *P*K* plausible new cluster centroid positions.

From this point, the clustering component of the ZoneBadger design process was implemented to ensure the contiguity objective of the project was met. For each individual in the population, the customer longitude/latitude location points were assigned a Manhattan Distance value (See Equation 2A in Appendix A) from the point to each of the *K* centroid longitude/latitude points of the current individual. The customer units were then assigned to the *Kth* centroid based on the shortest Manhattan Distance value. In the event of a minimal distance tie, the customer was assigned to the centroid whose Manhattan Distance value was calculated first for the point.

The post-clustering assignment design phase was centered around meeting the compactness and balancedness objectives for the project. First, the customers were grouped into clusters based on their assignment. The 'Tech_Takehome' variable was then summated for all customers within each cluster. Similarly, the customers were grouped, and the sum of their radial compactness values was calculated based on Equation 3A in Appendix A. The result was an array of *K* technician income and radial compactness values for the current population of individuals. These arrays were then provided to the GA to calculate an objective fitness function value (hereafter OFFV) based on Equation 4A of Appendix A.

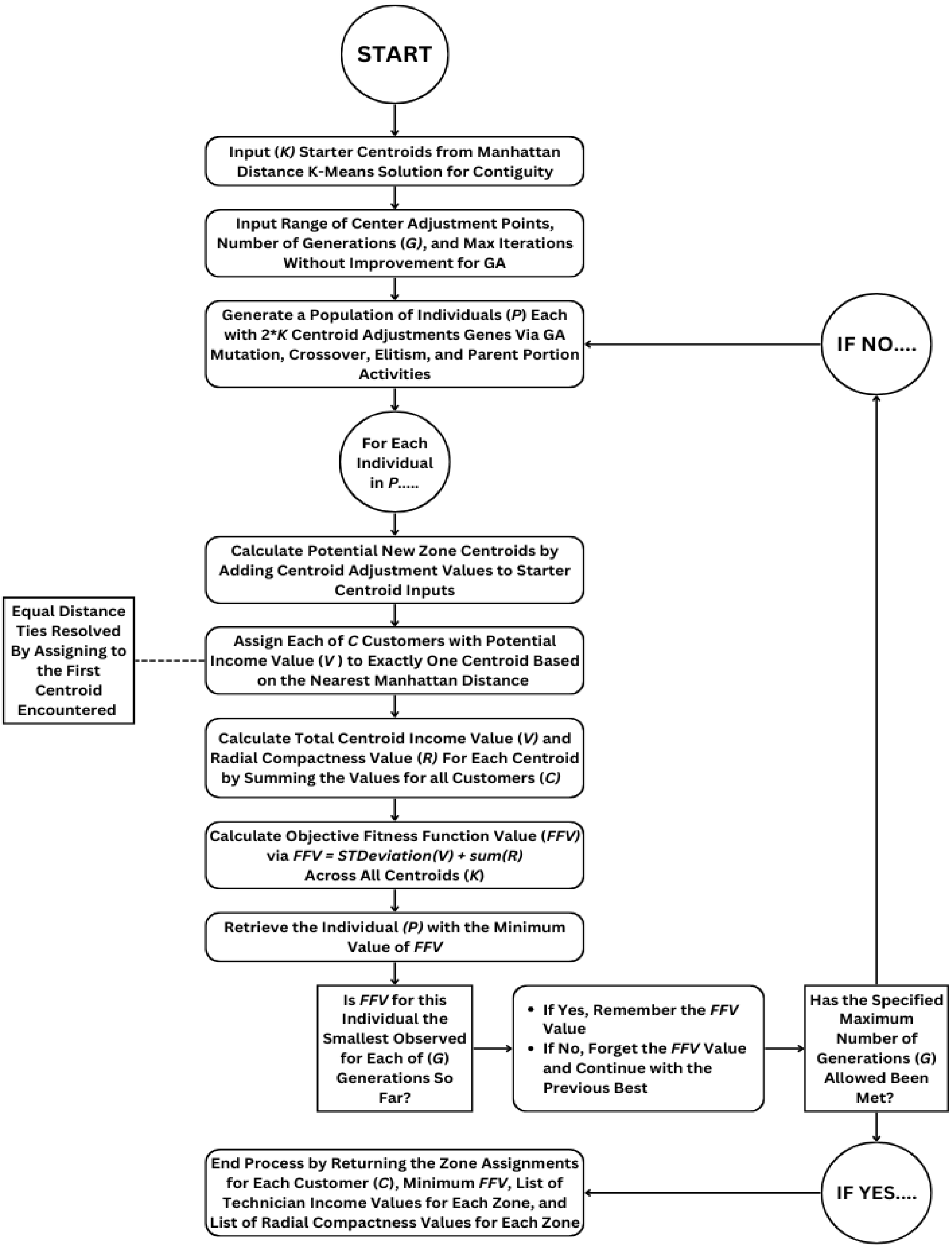
Because the GA operated to minimize the OFFV value, members of the population with small standard deviations (At $K = 6$, $\mu = 24{,}999.17$) in 'Tech_Takehome' and radial compactness values survived to produce offspring in subsequent generations. Offspring population production was achieved through GA implementation of mutation, crossover, elitism, and parent portioning (For an example, see Solgi, n.d.).

Across each generation, the lowest OFFV (highest evolutionary fitness) was compared against the best-known OFFV thus far. If the minimum OFFV for any subsequent population was lower than the best-known value to that point, the previous best value was forgotten, and the GA remembered the new one. Alternatively, if the minimum OFFV for any subsequent value was greater than the lowest-known value, the previous low was remembered, and the GA forgot the new population low. Based on this improvement approach, the solution became more balanced and more compact as the generations continued. Overall, a maximum number of generations of 500 was allowed, with any run that did not improve in OFFV after 100 generations also being terminated, and the chromosomal centroid adjustment values and best OFFV returned.

Overall, the design phase centered on implementing the components necessary to achieve contiguity, compactness, and balance. A flow chart of the final design for the ZoneBadger heuristic can be observed in Figure 3 below. The next step was utilizing the heuristic to reach a solution. To do this, the solution space was explored and then exploited thoroughly. Such exploration and exploitation were performed via parameter development utilizing systematic testing of parameter values through grid search as in machine learning model development (Kramer, 2017).

Figure 3

Flowchart of the Iterative Processes Behind the ZoneBadger Heuristic



Parameter Development

Finding the optimal solution for any problem that employs GA requires parameter tuning to explore and exploit the data space. This is because no optimal parameters adaptable to all problems exist (Kramer, 2017). Initially, the aim was to explore the entirety of the solution space until the functional area that minimized the OFFV was found. Once in this space, the area was exploited with minimal adjustments to the parameters to see if the OFFV could be improved. As such, exploration deals with finding the location of the optimal solution, while exploration deals with capitalizing on the optimal location. Here, parameter tuning was utilized to develop solution models for the Green Field Planning Territory Design Problem.

Of the seven parameters for the ZoneBadger heuristic, one was predominantly geared toward exploration. As the mutation value was increased, more centroid adjustment gene randomness was introduced in each generation. Random centroid gene generation meant testing of many centroid longitude/latitude locations for clustering the customer units was performed. Alternatively, two of the seven parameters were focal to exploitation. Namely, as the values for parent portion and elitism increased, fewer alternatives to the centroid adjustment genes were considered.

With the parent portion (also known as parenting), higher values resulted in more of the previous population being passed to subsequent generations. For example, a parent portion of 0.2 in a population of 100 meant that a random 20% of the individual chromosomes (20 individuals) were passed to the next generation. Thus, in this case, the subsequent generation would only produce 80 new individuals. As for elitism, higher values ensured the passing of the fittest individuals from a population. With an elitism value of 0.05, the top 5% of the population based on minimized OFFV would be passed to the next generation. Of the remaining four parameters, one could promote either exploration or exploitation.

Utilizing uniform crossover meant two parent chromosomes were compared gene by gene. If a randomly generated number was less than the chosen crossover threshold value, then the gene from parent two was carried on in the offspring. Alternatively, if the value was greater than the threshold, the gene from parent one survived. Thus, the ability of crossover to promote exploration or exploitation depended entirely on the current state of individual genes within the population. In cases where the mutation was high, and parent portion/elitism was low, crossover promoted exploration by randomly forming new individuals from a highly diverse gene pool. Alternatively, when the mutation was low, and parent portion/elitism values were high, crossover promoted exploitation by creating small changes to the genes from the near-optimal population.

The remaining parameters included the maximum number of iterations, the maximum number of iterations without improvement in the OFFV allowed, and the initial population size. Higher values for the initial population greatly influenced the number of trial solutions tested. As for the iteration parameters, these predominantly controlled the time the algorithm spent searching for solutions. Higher values did increase the likelihood of finding better solutions but also increased run time. Thus, a tradeoff had to be determined to find acceptable solutions in a reasonable amount of time.

Collectively, the parameters of the ZoneBadger heuristic served as the key to solving Turf Badger's problem because they controlled the GA's search for the minimum OFFV. However, tuning evolution-based metaheuristic parameters for optimization is known to be a laborious task that requires a significant amount of time (Tatsis & Parsopoulos, 2016). Thus, to reduce this task's time and resource complexity, models were developed to choose the best combination of parameters through a grid search of plausible "best values." Ultimately, the "best values" for the parameters were those that resulted in the minimization of the OFFV for the problem. This made it necessary to determine a criterion for optimal minimization that terminated subsequent search.

Stopping Criterion

The stopping criterion had to be determined to decide when the parameter tuning process had reached optimal OFFV minimization. A literature search did not uncover a clear guideline for this choice. Therefore, the decision had to be self-determined. Ultimately, model grid searches were performed until the OFFV value improvement was less than 20% from the previous best result. In the initial search, heuristic models with grids of parameters aiming to explore the solution space were developed. As the search continued, the parameter adjustments were fine-tuned to exploit this space. In addition to the overarching stopping criterion, developing a criterion for terminating any single heuristic model run was necessary.

The search for an optimal OFFV solution during any single run of the ZoneBadger heuristic was controlled by GA parameters in a maximum number of iterations and iterations without improvement. While similar, the parameters controlled two distinct mechanisms of the GA. The maximum number of iterations was the maximum number of populations that any single run was allowed to generate. Alternatively, iterations without improvement were time-saving because they allowed the run to terminate early when OFFV was no longer improving in subsequent generations. Choosing these values required consideration of a solution quality versus time tradeoff.

The greater the values of the maximum number of iterations and iterations without improvement, the longer the model's run time. This was because more populations were generated with increases in these parameters. In ZoneBadger, these parameters aimed to allow no more than two to three hours for the completion of any grid search through plausible heuristic models. This was because the researcher viewed the time range as reasonable. With a preset maximum of nine models in any single grid search, it was found that a maximum number of iterations value of 500 and iterations without improvement value of 50 achieved this aim while still finding quality solutions.

Conclusion

Data cleaning, feature engineering, exploratory data analysis, and heuristic modeling were essential for achieving the first two objectives of this project. The first three collectively allowed for accurately defining optimal "balanced" service zones for technician income that met the maximum workload allowed. Meanwhile, the heuristic modeling process centered around previous literature created a heuristic approximate optimization algorithm based on balancedness, contiguity, and compactness. The proceeding chapter presents the analytics results, visualizations, and FieldRoutes implementation of the ZoneBadger optimal model.

Chapter 4: Discussion of Analysis

Introduction

Turf Badger's need for service territories designed to improve resource deployment through balancedness, contiguity, and compactness was addressed by applying ZoneBadger to their active, recurring customer base. Ultimately, eighteen heuristic model iterations across three plausible parameter grids were tested to achieve the optimal solution per the procedures outlined in Chapter 3. Here, the grid search process and its results for selecting the optimal heuristic model are first presented. Then, visualizations for relating the model to the management-defined income and workload objectives are described. Finally, the FieldRoutes implementation of this model is shown and discussed.

Results

Grid Search & Optimal Model

Geared towards the exploration of the longitudinal/latitudinal customer location space, the initial parameter grid search required the development of nine ZoneBadger models. Both elitist and non-elitist GAs were implemented in this search. The population sizes were minimal at 10, 50, and 100 to avoid excess exploration of poor solution spaces. The models with their parameters and resulting OFFV output for the starter run are in Table 2 below. Ultimately, the best OFFV occurred in model number eight with a value of 1588.47. It was observed that non-elitist GAs performed poorly in this run and were thus not employed in future runs. Additionally, it was found that the best model had an OFFV of more than 60% less than the second-best model, indicating the optimal solution space was being narrowed in on.

Table 2

Initial Grid Search ZoneBadger Model Parameters with OFFV Outputs

Model Number	Objective Function	Population	Elitism	Crossover Probability	Mutation Probability	Parent Portion	Max Iterations	Selected?
M1	13390.7	10	No	0.3	0.05	0.1	500	No
M2	4343.47	10	Yes: 0.02	0.4	0.15	0.2	500	No
M3	10139	10	Yes: 0.04	0.5	0.25	0.3	500	No
M4	21440.3	50	No	0.3	0.05	0.1	500	No
M5	4193.21	50	Yes: 0.02	0.4	0.15	0.2	500	No
M6	5466.07	50	Yes: 0.04	0.5	0.25	0.3	500	No
M7	13814.7	100	No	0.3	0.05	0.1	500	No
M8	1588.47	100	Yes: 0.02	0.4	0.15	0.2	500	Yes
M9	2638.78	100	Yes: 0.04	0.5	0.25	0.3	500	No

For the second grid search, the parameters were adjusted to promote exploitation of the space found in model number eight. This meant the population, parent portion, and elitist parameters were all increased. Further, the mutation parameter was decreased from the previous best for increased exploitation. The crossover values were also increased, but this was done to introduce more variability in the gene selection process. In contrast with the nine models of the first search, only six models were tested here because population increases of only 200 and 250 were trialed due to vastly increased times to reach solutions past 250. The models with their parameters and resulting OFFV outputs for the second run are in Table 3 below. Among the models, the best was number fourteen, with an OFFV value of 540.70. This was a 34.04% reduction in OFFV from model eight in the first run, which triggered another grid search for exploitation.

Table 3

Grid Search ZoneBadger Model Parameters with OFFV Outputs for Iteration Two

Model Number	Objective Function	Population	Elitism	Crossover Probability	Mutation Probability	Parent Portion	Max Iterations	Selected?
M10	1255.95	200	Yes 0.02	0.4	0.15	0.2	500	No
M11	2423.07	200	Yes: 0.04	0.5	0.13	0.3	500	No
M12	2204.23	200	Yes: 0.06	0.6	0.11	0.4	500	No
M13	1389.73	250	Yes 0.02	0.4	0.15	0.2	500	No
M14	540.7	250	Yes: 0.04	0.5	0.13	0.3	500	Yes: OPTIMAL MODEL!
M15	1377.06	250	Yes: 0.06	0.6	0.11	0.4	500	No

The third run was the final iteration of the parameter and solution search process. The optimal space in model fourteen was exploited following the same adjustment procedures from the second iteration. However, the adjustments were minimal, as the range of new test solutions was significantly narrowed. Further, only three new models were tested in this iteration because the population was held constant at 250. The models with their parameters and resulting OFFV outputs for the third run are in Table 4 below. The best was model sixteen, which produced an OFFV value of 1299.68. This result worsened the optimal solution by 41.60%.

Table 4

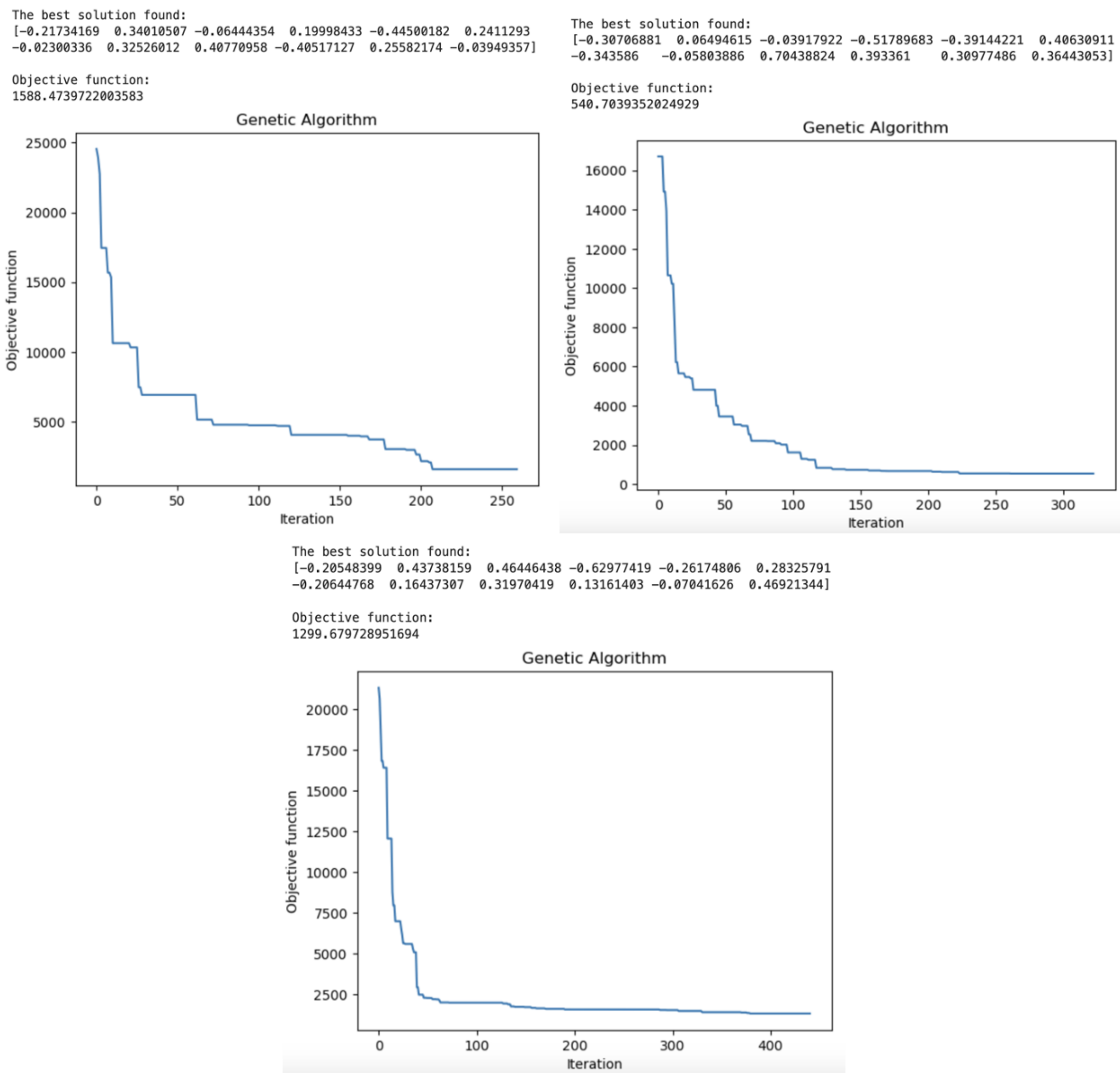
Grid Search ZoneBadger Model Parameters with OFFV Outputs for Iteration Three

Model	OFFV	Population	Elitism	Crossover Probability	Mutation Probability	Parent Portion	Max Iterations	Selected?
M16	1299.68	250	Yes 0.045	0.53	0.127	0.33	500	No
M17	2222.27	250	Yes: 0.05	0.56	0.124	0.36	500	No
M18	3482.67	250	Yes: 0.055	0.59	0.121	0.39	500	No

In iteration three, the lack of a solution that improved OFFV by at least 20% meant that the stopping criterion for subsequent grid search was met. Figure 4 below displays the GA output for the best models from the three iterations with their centroid adjustment genes from the fittest population member. This provides a visualization of the GA process for arriving at the optimal solution. Ultimately, the parameters and OFFV value from ZoneBadger model fourteen would be utilized to create result visualizations and implement a FieldRoutes solution to the business problem.

Figure 4

ZoneBadger GA Output with Best Centroid Adjustment Genes for Models 8 (Top Left), 14 (Top Right), and 16 (Bottom)

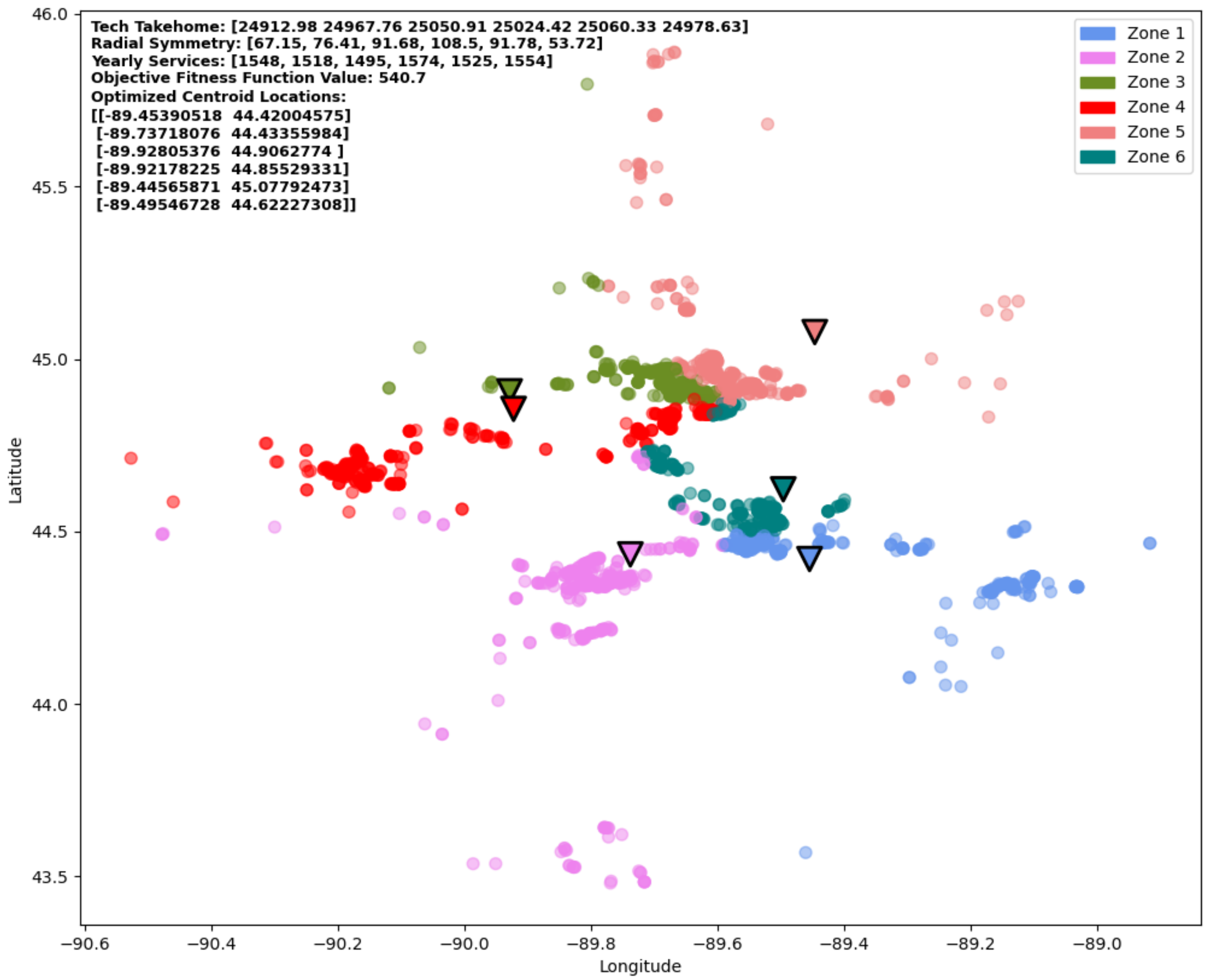


Result Visualizations

It was observed that model number fourteen minimized the OFFV for customer unit centroid assignment with Turf Badger's data among eighteen trialed parameter solutions. Thus, it served as the source for visuals aiming to prove ZoneBadger's solution generation process sufficiently worthy of implementation by meeting management objectives for balancedness, contiguity, and compactness. Figure 5 below was created for comparison against the initial non-optimized Manhattan Distance K-Means starter solution (see Figure 2). The solution was plotted with customer units color assigned for each zone, the optimal centroids plotted, and technician income, yearly service, radial symmetry, OFFV, and longitude/latitude coordinate values shown. This visual provided an initial insight into the ability of ZoneBadger to optimize the balance of technician income and workload while meeting contiguity and compactness in a way the original solution could not.

Figure 5

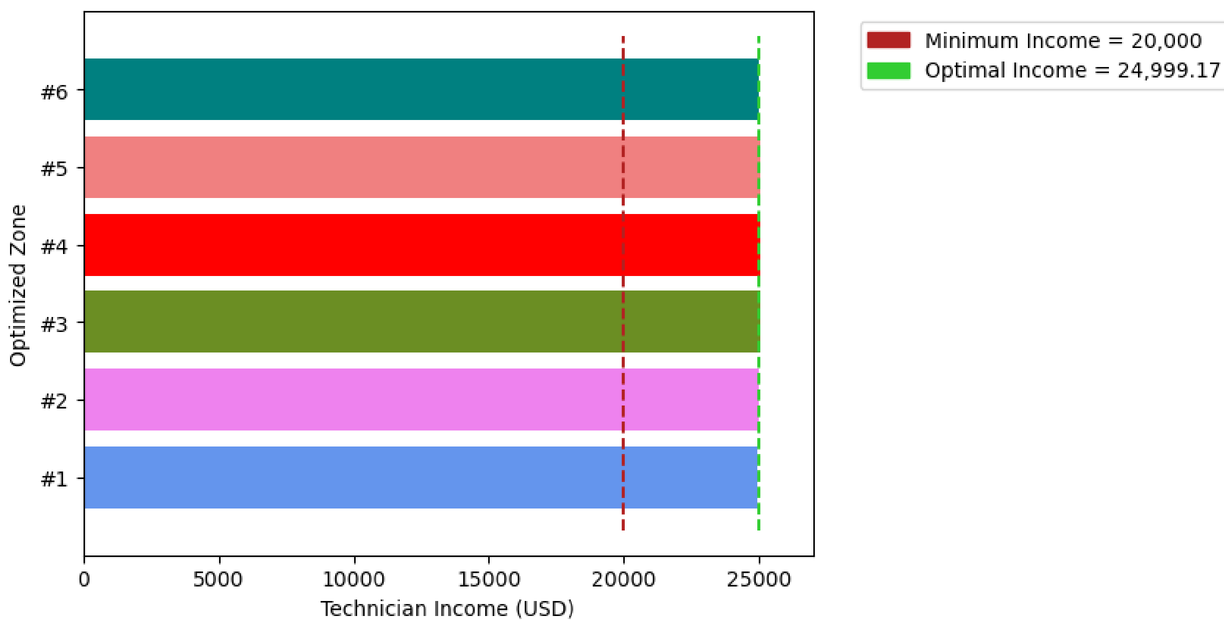
ZoneBadger Optimized Assignment of Stevens Point Branch Customers with Centroid Location, Technician Income, Yearly Services, Radial Symmetry, and OFFV Zone Values



After mapping, the ZoneBadger optimal model solution was subject to analysis for its performance in meeting the balanced zone objectives set forth by management. The first assessment broke down the OFFV subcomponent of technician income standard deviation from the model. Overall, the value between zones was low at 51.47 USD. Figure 6 below shows that the quasi-hard income constraint in the solution met management zone income aims as the final value for each was observably near the mean goal of 24,999.17. For example, at most the income value deviated from the goal by only 0.17% in zone 1. This was a reduction of 34.27% from the same deviation value in the initial non-optimized solution. Further, all values were well above the allowed minimum of 20,000 USD. Therefore, the ZoneBadger heuristic solution was sufficient for achieving the core balancedness objective.

Figure 6

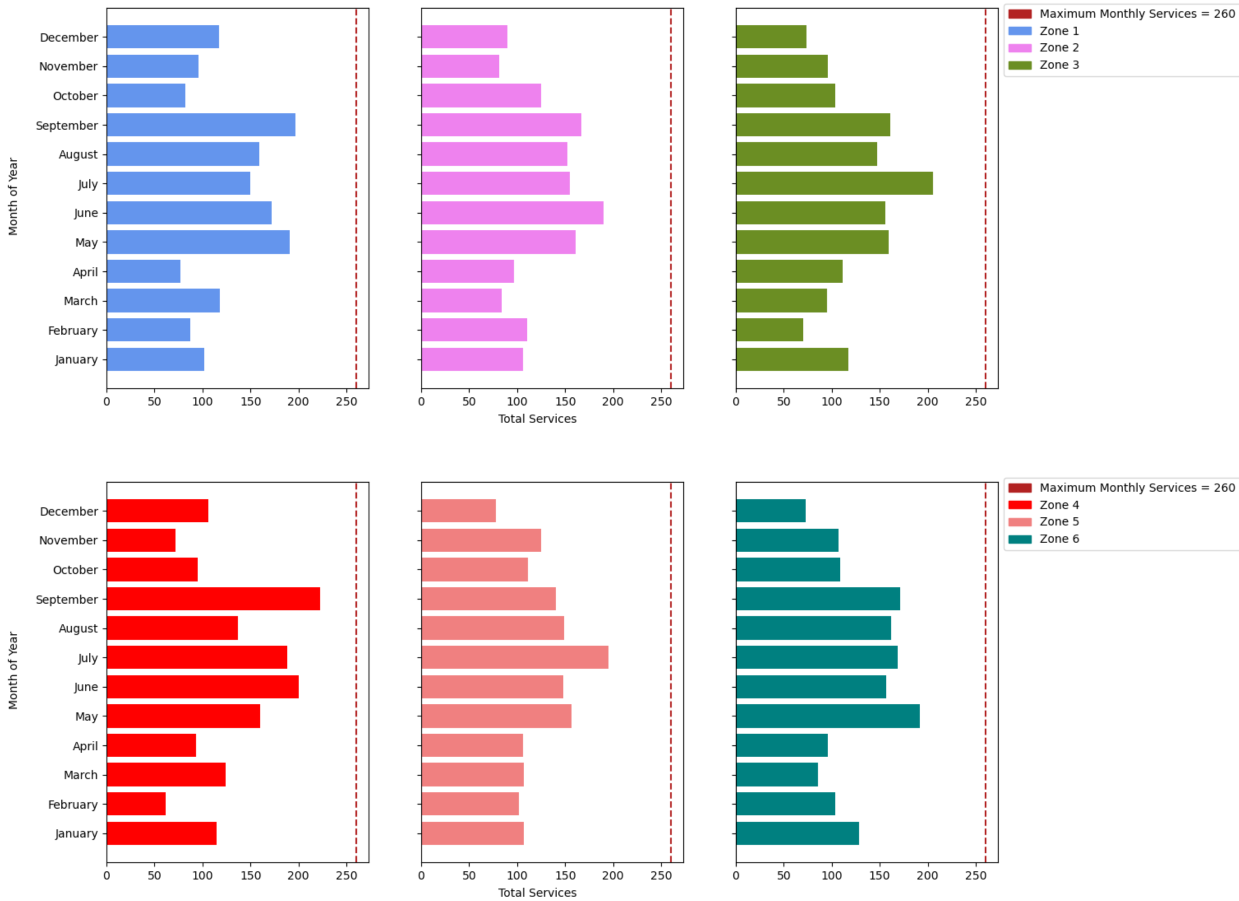
ZoneBadger's Optimal Model Income Values Meet the Primary Managerial Objective for Technician Income Balance Between Zones at Greater than 20,000 USD in Each



As for the technician workload aim of no more than 260 services per month, Figure 7 below shows that ZoneBadger's optimal model also met this requirement. It was observed that the service requirements in any single month only topped 200 twice, with a maximum of 222 services occurring in September for zone four. There were also twenty months when service totals were below 100 in the solution. These findings indicate that the ZoneBadger solution leaves room for increased service demands from new customers contracted into zones. Such a reality proved that the heuristic was sufficient for meeting the workload aims.

Figure 7

ZoneBadger's Optimal Model Workload Values Meet the Primary Managerial Objective for Balance Between Zones at No More Than 260 Services in Any Single Month



The final efficacy visualization for ZoneBadger utilized geographic mapping capabilities to show each zone's adherence to the contiguity and compactness aims. Figure 8 below shows the geographic overview of customer assignments in Wisconsin. Ultimately, some zones had geographic features, such as rivers, isolating individual customers from most of the others in the group. However, these cases did not violate the requirements as continuous lines could still be drawn to include only customers in their specific zones. Figure 9 below depicts two such cases on the geographic map. Such occurrences are addressed as limitations in the concluding chapter of this report, and plausible methods for addressing them are also highlighted. Nevertheless, ZoneBadger was sufficient in meeting the managerial aims of contiguity and compactness.

Figure 8

Color-Coded Wisconsin Geographic Overview for the ZoneBadger Solution

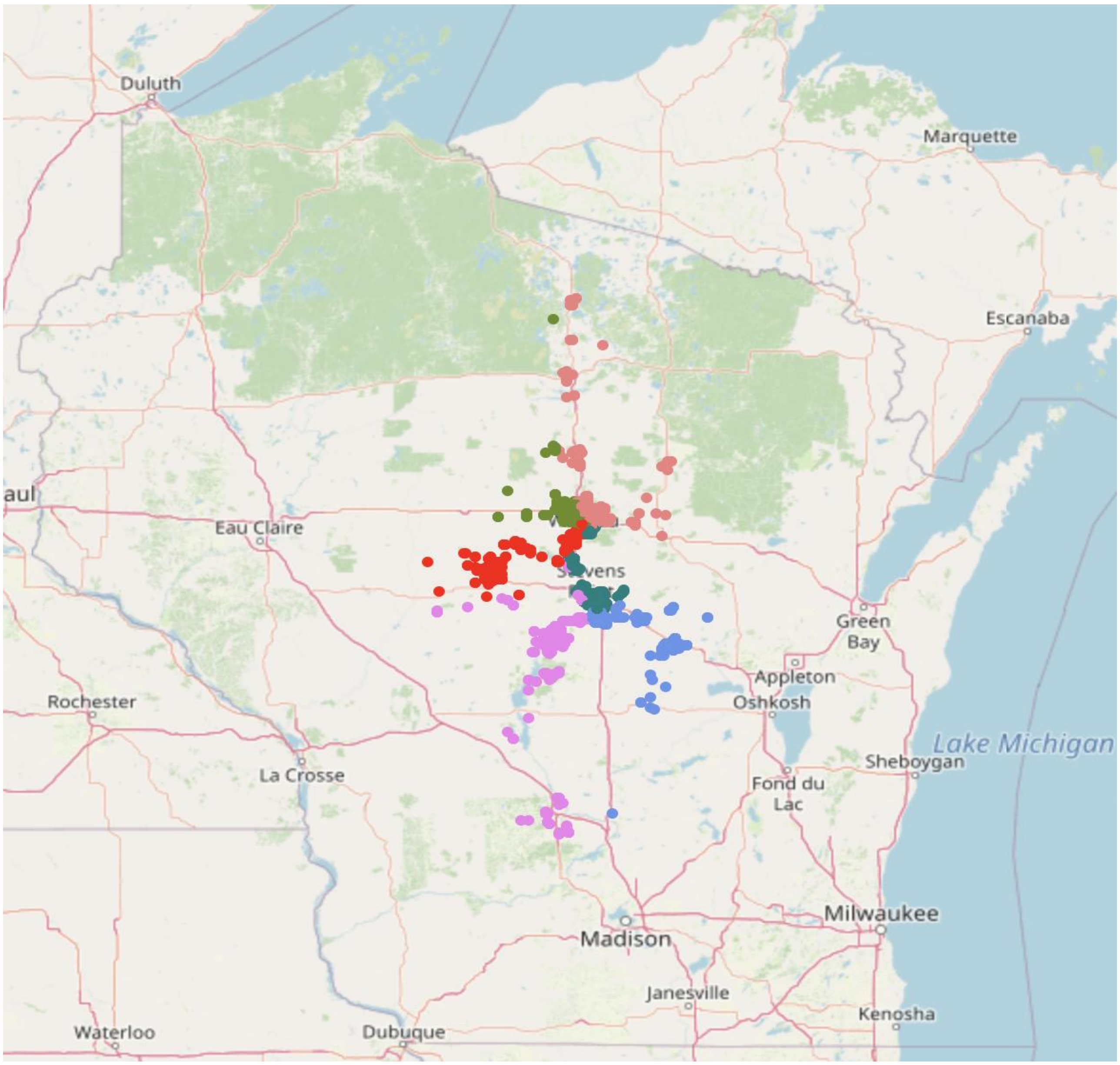
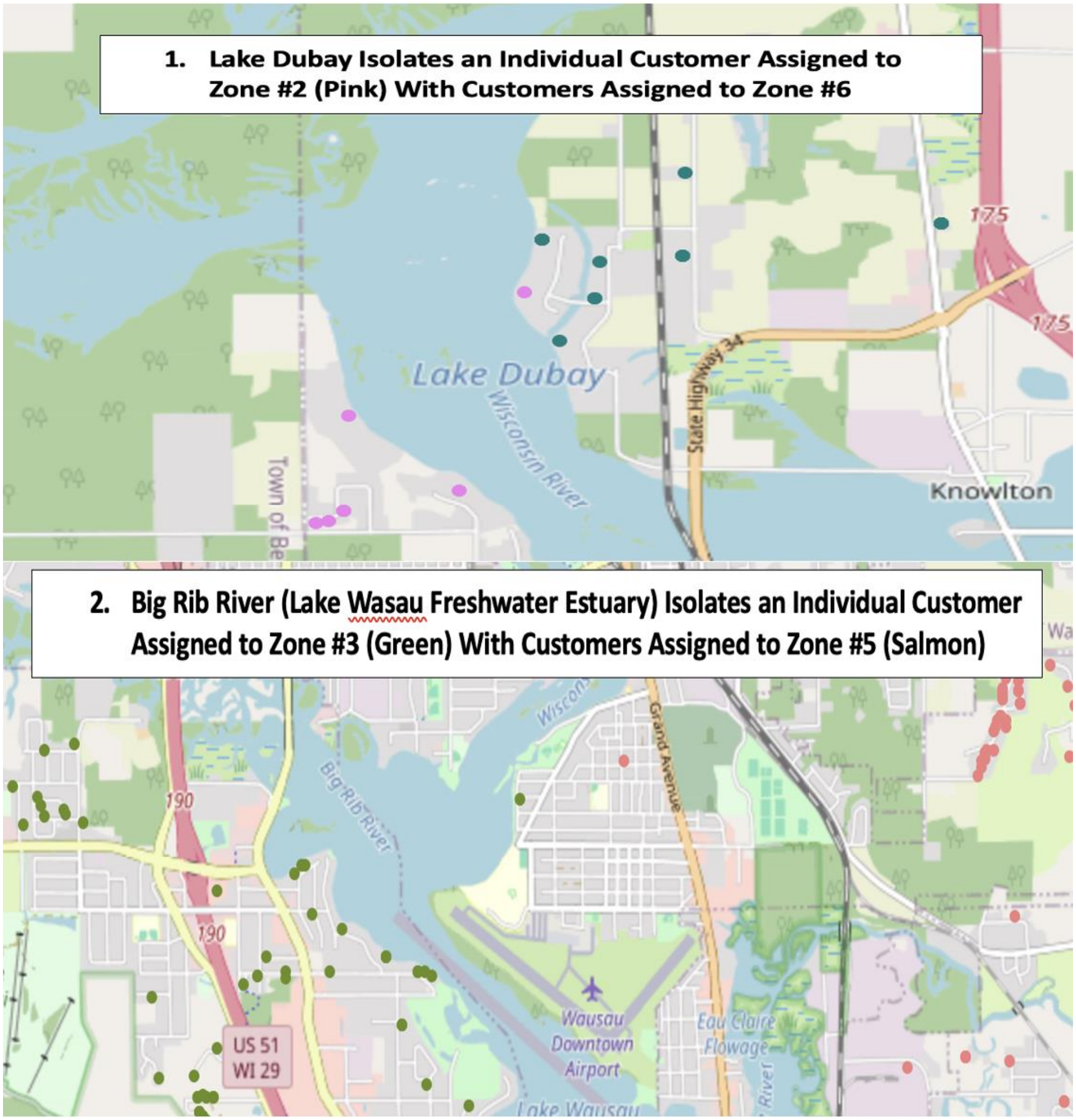


Figure 9

Two Zoomed-In Cases of Tight Junctions Between Zones Where Geographic Features Isolate Individual Customers from the Majority of Customers Assigned to the Group



FieldRoutes Implementation

Undeniably, the resulting analysis of the ZoneBadge heuristic optimal model proved its worth in solving the Green Field Planning Territory Design Problem for Turf Badger. With the solution in hand, data deployment was the final step in the analytics process. Known as "The Last Mile" of analytics endeavors, data deployment is the point where projects turn from technology to people (Dykes, 2019). In this project, getting operations management that deals with scheduling technicians for the branch to utilize the solution was the technology to people challenge. Ultimately, it was decided that the Visual Grouping tool from Turf Badger's operations suite, FieldRoutes, would be the tool to get the job done.

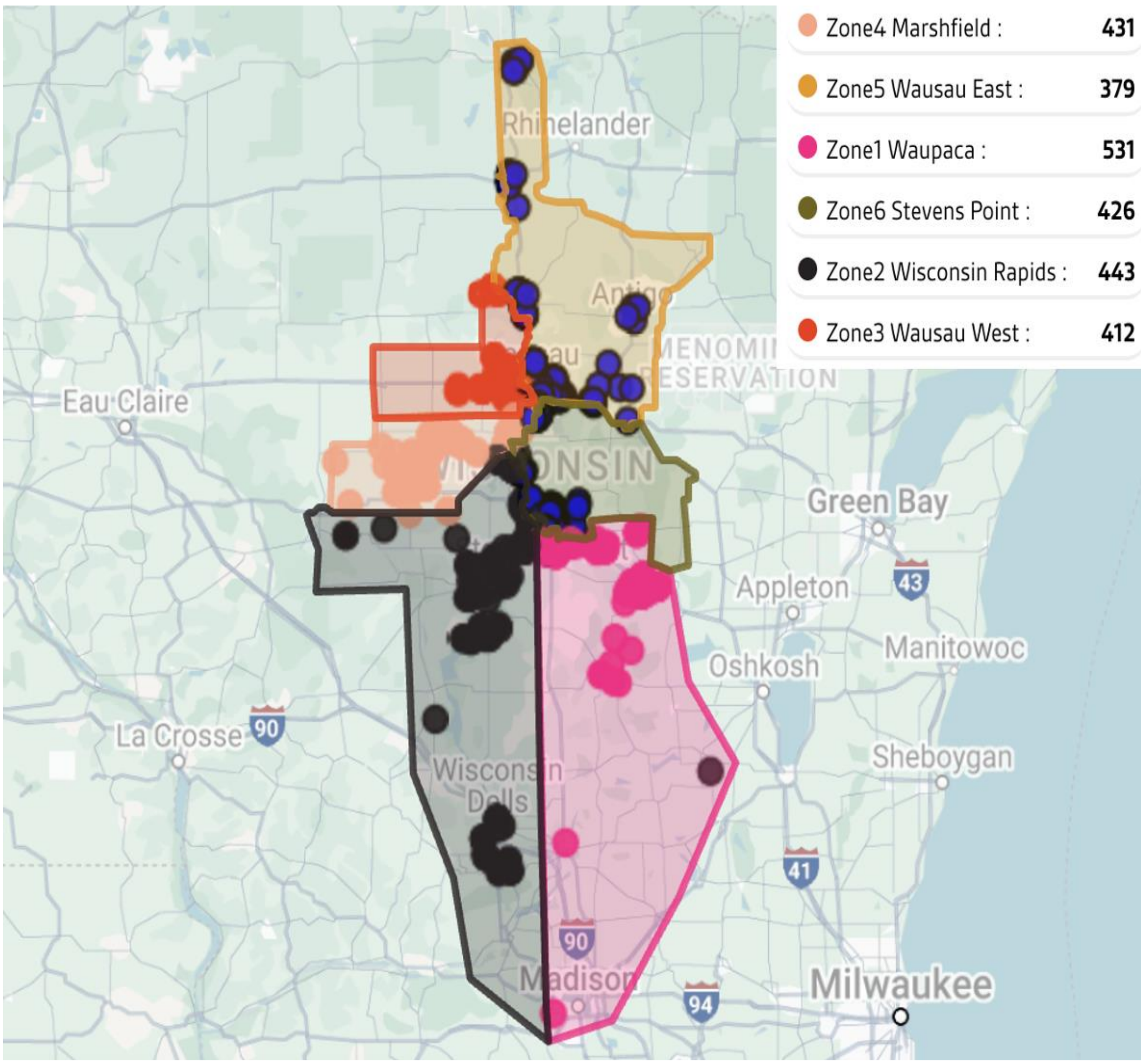
The Visual Grouping application allowed customers within a range of service dates to be mapped geographically, and they could then be assigned to hand-drawn zones for routing. Overall, the zone-design process was complicated because, while the final zones were continuous and compact, they lacked concrete borders. Thus, the border drawing process employed a degree of subjectivity. To combat this, the interactive geographic Python HTML solution (see Figure 9 above) was utilized to zoom in and out on individual customer points when drawing boundaries.

Figure 10 below depicts the Visual Grouping results from this process with the zone names, color coding scheme, and total customer counts shown in the legend. Notedly, the legend order and zone color coding were not manually adjustable within the FielRoutes Visual Grouping tool. Thus, the orders and colors do not match the manually coded solutions. The FieldRoutes zones were named based on the Wisconsin city (or location within the city) where most of their assigned customer location units were located. Lastly, the total customer count was higher at

2,622 than the ZoneBadger dataset at 2,282 because the branch gained 340 customers since the data was current on 01/01/2025 by the implementation date of 03/16/2025.

Figure 10

Zoomed-Out View of the Final Implementation with FieldRoutes Visual Grouping Tool for Hand-Drawn Zones Based on ZoneBadger's Best Model



The zone-design process of Figure 10 required a great degree of care because of the lack of boundaries from the ZoneBadger solution. Ultimately, the zooming process allowed for zone boundaries to be drawn down to the individual street customer location level, introducing the highest accuracy possible at junction areas. Figure 11 depicts the Python geographical map utilized to draw the subsequent FieldRoutes implementation between Zone1 Waupaca and Zone2 Stevens Point. The junction point occurred for customer units near Plover, Wisconsin.

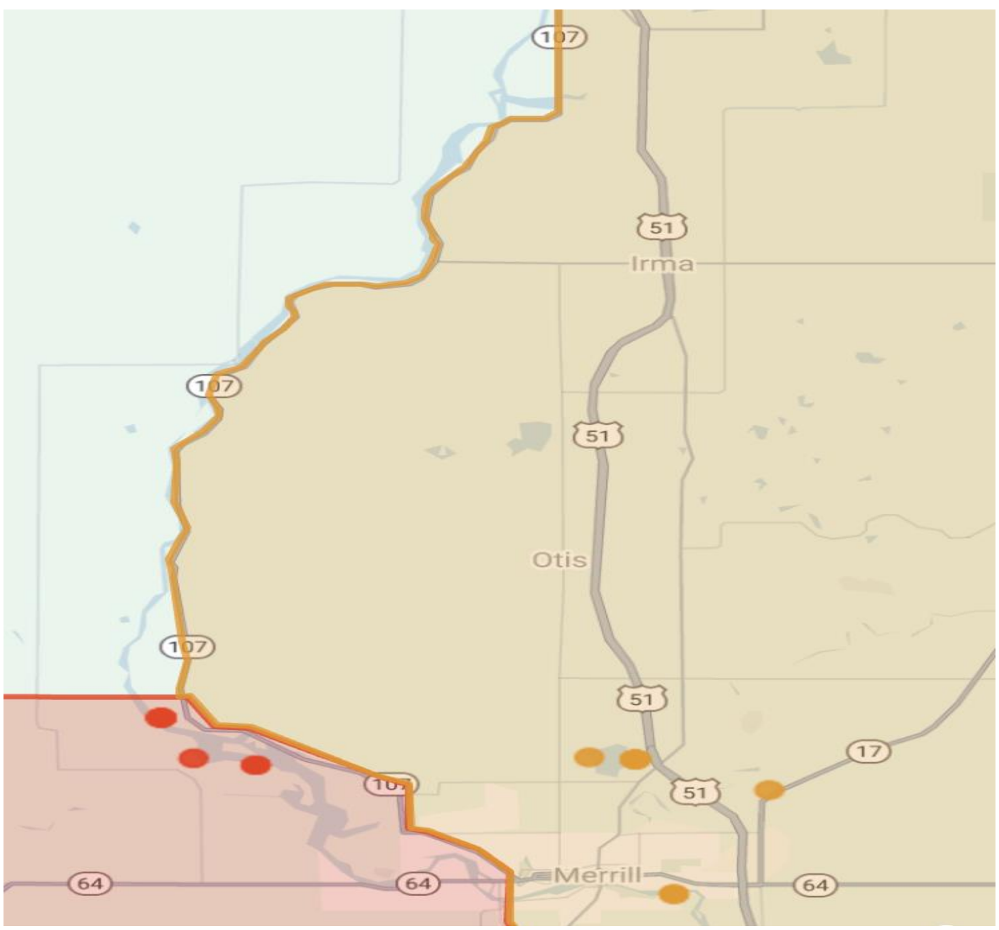
Figure 11

Zoom-In of the Python ZoneBadger HTML Interactive Geographic Map (Top) Utilized to Create the Hand-Drawn Visual Grouping Implementation for the Junction Between Zone1 Waupaca and Zone2 Stevens Point in FieldRoutes (Bottom) Near Plover, Wisconsin

As for the outer boundaries where junctions did not occur, it was decided that natural and artificial obstacles would be used to draw the edges based on proximity to customer units in the zone. Figure 12 shows an example of this process where Wisconsin State Highway 107 was utilized as the boundary edge for the northern extension of Zone5 Wausau East. Such major natural and artificial boundaries did not exist in all outer boundary design cases. Thus, these borders were subjectively drawn. However, the zones can be expanded manually as new customers are acquired outside their boundaries. As such, these subjective borders can easily be adjusted in the future.

Figure 12

The Outer Boundary of Zone5 Wausau East Follows the Man-Made Wisconsin State Highway 107 as it Extends Towards Northern Customer Units Near Merrill, Wisconsin



Overall, the implementation process of the ZoneBadger final model into FieldRoutes was accomplished with an emphasis on accurately drawing inner zone junctions. Such an emphasis ensured the solution was integrated with care so that the analytic balancedness, contiguity, and compactness findings held true for operations management when utilizing the tool for technician scheduling. However, variability in the final income and workload measures for these FieldRoutes zones was unavoidable because the active customer base changed from the data extraction date to the implementation date. Nevertheless, the implementation allows future scheduling to promote more equity in technician income and workload compared to the currently employed scheduling process. As such, the Visual Grouping results met this project's third and final objective.

Conclusion

Here, the grid search process for parameter and model selection, the visualizations relating to income and workload management defined objectives, and the FieldRoutes implementation of the final ZoneBadger model were discussed. Overall, the discussion pertaining these analytics processes to specific project objectives while validating the efficacy of the ZoneBadger heuristic. Moving forward, a project summary, implications for the results, and future steps for management are presented as a concluding chapter.

Chapter 5: Project Summary

Introduction

This project utilized a combined clustering and genetic algorithm heuristic modeling approach to solve a Green Field Planning Territory Design Problem for Turf Badger's Stevens Point branch. The aim was to create zones that satisfied three common territory design principles: balancedness, contiguity, and compactness (Moreno et al., 2020). Predefined zone objectives for technician workload and income were determined through collaboration with operations management. Ultimately, the newly designed zones will streamline operations, improve consistency of service delivery, and be exploitable for future business projects. Here, the project's success in meeting the three primary objectives, the project implications for the industry, and potential next steps are discussed.

Summary of Project

Three core objectives existed for the project. The first was to define optimal "balanced" service zones for technician income that meet a workload maximum. This objective was achieved in the planning stage by utilizing analytics research alongside collaboration with the Director of Operations, Tricia Gruber. It was decided that income needed to be as close to the average possible amount available for six zones, which was $24,999.17, and that it also needed to be at least $20,000 for each zone. The mean value was suggested because it represents an average central tendency measure. In other words, it was the "balance point" for the income values. Meanwhile, the minimum was determined by management to meet a threshold determined to be desirable for an increased likelihood of a successful hiring process.

For service loads, it was determined that technicians needed to perform no more than 260 services in any month throughout the year for the zones. Ultimately, this value was the result of management analytics research. It was found by taking the sum of services in 2024 for the branch's highest performing service technician, Chandler Huseby, at 4,440 and dividing it by twelve. This value, 370, was then reduced by 30% and rounded to the nearest tens at 260 to leave room for customer growth within zones to avoid overwhelming technicians.

The second objective for the project was to create a heuristic approximate optimization algorithm based on balancedness, contiguity, and compactness to meet the management-defined constraints from objective one. Consideration of multiple analytics approaches capable of implementing constraints under such conditions was thus required. An initial hard constraint combined LI Programming and GA heuristic approach, based on the literature for Capacitated Clustering Problems, proved too difficult to implement. However, the alternative combined Clustering and GA approach with quasi-hard constraints was successfully implemented.

Overall, the heuristic, ZoneBadger, relied on Manhattan Distance-based partitional clustering to ensure continuity and utilized an OFFV function built to equalize income by minimizing the standard deviation between zones with contiguity implemented through an additive radial compactness penalty term. ZoneBadger's final solution was proven sufficient for meeting management zone aims and thus achieved the second project objective.

The final objective was to produce visualizations for the optimized customer assignments from the final ZoneBadger model and implement the solution into FieldRoutes for use by branch management. The visualization component for this objective was achieved through the creation of standard and geographic maps with color-coded customer zone assignment legends. Further, bar graphs of zone incomes and workloads were created to show that zones met the required values for each constraint.

The implementation process was achieved by utilizing a Python interactive geographic HTML map to draw the zones in FieldRoutes with its Visual Grouping tool. This map was used to zoom in and out on individual customer points so that zone junctions were drawn nearly precisely to match the ZoneBadger solution. Then, natural and artificial boundaries were utilized, when possible, to define the outer edges of the zones. This approach produced a final mapping in Turf Badger's FieldRoutes software that can be used by branch management to schedule technicians going forward.

This project completed all three primary objectives. Achieving this required extensive collaboration and research of existing literature to fine-tune the data analytics process. Next, opportunities and implications of the results in terms of business use cases are discussed to provide a breakdown of the value gained from performing this data analytics project.

Implications of Results

Undoubtedly, this project provides Turf Badger with the potential for significant competitive advantages in the pest control/lawn care service industry. Firstly, the ZoneBadger solution allows the business to provide consistent service for current and future customers by having the same technicians repeatedly service customers in their assigned zone(s). Service consistency reduces churn and increases customer loyalty, which are measures of positive customer satisfaction (Pulido et al., 2014). Thus, ZoneBadger provides significant value through the advantage it gives the company in this realm.

Next, Turf Badger's Stevens Point branch will be able to utilize the ZoneBadger solution in its hiring process. When hiring new technicians, specific, measurable work expectations will be provided based on the zone the employee is being hired for. This allows Turf Badger to increase their workforce quality, as high-quality descriptions attract employees that align with the company's mission, values, and culture (Morelli, 2023). Optimizing service territories also makes evaluating technician performance easier. Under disproportionate, non-optimal conditions, disparities in efficiency and customer satisfaction with technicians cannot be judged fairly (Hess & Samuels, 1971). Thus, the ZoneBadger implementation allows Turf Badger to examine their service specialists more objectively.

Further, the solution allows Turf Badger to route zones efficiently by accounting for geospatial distance metrics between customer units. Continuity in zone travel reduces fuel consumption and time between tasks. This not only increases profitability and improves customer experience but also provides the opportunity to allocate the savings to alternative programs like pesticide product procurement. Additionally, balancing income and workload with ZoneBadger allows Turf Badger to promote equity and fairness in salaried service positions without pouring resources into costly audits. Salary equity promotes talent retention and high morale while contributing to churn reduction. As such, it influences employees' decisions to stay with companies and provides a competitive advantage in the industry (Nagele-Piazza, 2020).

Perhaps most importantly, the solution for the Steven's Point branch provides a blueprint for the expansion of ZoneBadger to the remaining nine branches. The primary goal set by management for this pilot project was to develop an approach that could be adapted to all branches in time. Such adaptation will require defining income and workload aims that are measurable specifically for the other branches before simply following the procedures outlined here to reach an "optimal" solution. This process will require a significant investment of employee time and require them to have data analytics knowledge, but if achieved, the results will revolutionize the business's efficiency and profitability. These use cases provide evidence of the impact and value of utilizing data science to solve service industry problems.

Next Steps

Three subsequent steps exist for Turf Badger with the conclusion of this project. First, they must capitalize on the value found in the project results by linking their active technician scheduling to the ZoneBadger solution implemented in the Visual Grouping tool. The solution met the objectives set forth by management. Thus, reluctance to utilize it for scheduling can be attributed to a lack of trust in analytics-driven decision-making (Mahony, 2021). However, the procedures, methodology, and results were demonstrated and articulated in detail here, with management and ownership being collaborators throughout the process. Therefore, the possibility for such a lack of trust was diminished to the fullest extent possible, and the decision to utilize the findings should be made.

Secondly, after some time, the results implementation should be subject to review to ensure the potential value explained here is being realized. Such a review could include report analyses of costs, such as fuel spending through travel time, data analytics analyses of zone incomes and workloads, and traditional non-analytics-based surveys to gauge how service technicians feel about the new system. Though ultimately a subjective decision, a review after the conclusion of the first “peak season” in which ZoneBadger is utilized is recommended.

If the review process shows the solution performing as expected, Turf Badger must expand the project to more branches. The procedures and methods for the ZoneBadger heuristic are explained in detail in the Jupyter Notebooks provided to the company. For easy reproducibility, links to a GitHub repository containing them can also be found in Appendix B of this report. This means they can expand the project to the nine remaining branches, as management desires, if they so choose.

While these three action steps for the future are attainable, they are not without limitations. For instance, expanding the ZoneBadger heuristic to additional branches would require employing at least one analytics professional with knowledge in the subject area of this project. However, the company currently does not employ an analytics specialist and does not have active plans to onboard this role in the future. Therefore, they are encouraged to consider creating this position to realize their aims.

Further, in Figure 9 of Chapter 4, it was noted that the FieldRoutes implementation of the ZoneBadger solution had cases where customer units were isolated from their assigned zones by natural obstacles like lakes or rivers. Such cases would increase drive times significantly, especially close crossing points do not exist, which would decrease the resource optimization of the solution. Thus, they should be addressed to improve the solution. It is recommended that identified cases should be reassigned to the nearest cluster that would avoid isolation completely. Alternatively, another additive penalty term could be considered for the ZoneBadger model where points are assigned higher values for cluster assignment if they are in isolation from the considered center. In either case, a tradeoff would likely exist where resource allocation would be improved and income/workload balancedness would be reduced.

Also in Chapter 4, it was noted that the implemented FieldRoutes solution included 2,622 total customers as opposed to the 2,282 from the final dataset for ZoneBadger. Thus, net growth occurred in customer volume for the branch within three months from data procurement to results. As such, some zones could have lost customers, and others could have gained them during this time. This highlights that businesses are not stagnant like the data sources often used in analytics. Therefore, the methodology here could be improved if the data were analyzed in real-time, with constant updates throughout the analytics process. This would guarantee a balance in incomes and workloads in a way that is impossible with the current project procedures.

Conclusion

Data analytics were employed in this client-based project to solve a Green Field Planning Territorial Design Problem for Turf Badger. Overall, heuristic techniques were utilized to develop an optimization model called ZoneBadger based on combined clustering and genetic algorithm methodologies. The optimal ZoneBadger model was found by performing iterative grid searches for parameter adjustments based on exploration and exploitation of the solution space. This model, which achieved a minimal OFFV of 540.70, was based on a combined standard deviation and radial compactness objective fitness function.

Ultimately, customers were assigned to zones based on the model's output before visualizations were utilized to evaluate the solution's efficacy in meeting management-defined balance, contiguity, and compactness aims. After the solution was proven adequate in meeting each of these aims, it was implemented in the company’s FieldRoutes scheduling software through the Visual Groupings tool. Finally, the use cases, business value, and future steps were highlighted to gain management’s trust in the results and provide a path forward once the model is actively deployed.

Sources

Ashkenas, R., and Matta, N. (2021, January 8th). How to Scale a Successful Pilot Project. *Harvard Business Review.* https://hbr.org/2021/01/how-to-scale-a-successful-pilot-project

Bacao, F., Lobo, V., and Painho, M. (2005). Applying Genetic Algorithms to Zone Design. *Soft Computing, 9*(5), 341-348. 10.1007/s00500-004-0413-4

Boddapu, H. (2021, January 4th). Capacitated K Means Clustering on Python || Machine Learning Tutorial by Hemanth Boddapu. YouTube. https://www.youtube.com/watch?v=8Z7ElWS2BKg

Chaney, W. (2021, January 19th). Make the Most out of PestRoutes in 2021. *FieldRoutes.* https://www.fieldroutes.com/blog/make-the-most-out-of-pestroutes-in-2021

Duque, J. C., Anselin, L., and Rey, S. J. (2012). The Max P-Regions Problem. *Journal of Regional Science, 52*(3), 397-419. https://doi.org/10.1111/j.1467-9787.2011.00743.x

Dykes, B. (2019, December 18th). Two Keys to Conquering The Last Mile in Analytics. *Forbes.* https://www.forbes.com/sites/brentdykes/2019/12/18/two-keys-to-conquering-the-last-mile-in-analytics/

Frei, F. X. (2008, April). The Four Things a Service Business Must Get Right. *Harvard Business Review.* https://hbr.org/2008/04/the-four-things-a-service-business-must-get-right

Garey, M. R., and Johnson, D. S. (1979). *Computers and Intractability: A guide to the Theory of NP-Completeness.* Bell Telephone Laboratories, Incorporated.

Gupta, D., and Ghafir, S. (2012). An Overview of Methods Maintaining Diversity in Genetic Algorithms. *International Journal of Emerging Technology and Advanced Engineering, 2*(5), 56-60. https://api.semanticscholar.org/CorpusID:1414600

Hagen, J., and Hess, T. (2021). Collaboration for Big Data Analytics: Investigating the (Troubled) Relationship between Data Science Experts and Functional Managers. *Proceedings of the 54th Hawaii International Conference on System Sciences,* 254-263. 10.24251/HICSS.2021.030

Hess, S. W., and Samuels, S. A. (1971). Experiences with a Sales Districting Model: Criteria and Implementation. *Management Science, 18*(4), 41-54. https://doi.org/10.1287/mnsc.18.4.P41

Hillier, F. S., and Lieberman, G. J. (2015). *Introduction to Operations Research Tenth Edition.* McGraw-Hill Education.

Kirkpatrick, S., Gelatt, C. D., and Vecchi, M. P. (1983). Optimization by Simulated Annealing. *Science, 220*(4598), 671-680. 10.1126/science.220.4598.671

Kramer, O. (2017). *Genetic Algorithm Essentials.* Springer International Publishing. https://www.elibrary.umegabuana.ac.id/index.php?p=fstream-pdf&fid=261&bid=697

Lodish, L. (1975). Sales Territory Alignment to Maximize Profit. *Journal of Marketing Research, 12*(1), 30-36. https://doi.org/10.2307/3150655

Mahony, C. (2021, July 26th). Three Steps to Building Trust in Data-Driven Decision- Making. *Forbes.*

https://www.forbes.com/councils/forbestechcouncil/2021/07/26/three-steps-to-building-trust-in-data-driven-decision-making/

Maulik, U., and Bandyopadhyay, S. (2000). Genetic Algorithm-Based Clustering Technique. *Pattern Recognition,* (33), 1455-1465. https://doi.org/10.1016/S0031-3203(99)00137-5

Morelli, N. (2023, January 4th). Why Quality Job Descriptions Still Matter in Today's World of Work. *Forbes.* https://www.forbes.com/councils/forbeshumanresourcescouncil/2023/01/04/why-quality-job-descriptions-still-matter-in-todays-world-of-work/

Moreno, S., Pereira, J., and Yushimito, W. (2020). A Hybrid K-Means and Integer Programming Method for Commercial Territory Design: A Case Study in Meat Distribution. *Annals of Operations Research,* 286, 87-117. https://doi.org/10.1007/s10479-017-2742-6

Mulvey, J. M., and Beck, M. P. (1984). Solving Capacitated Clustering Problems. *European Journal of Operations Research, 18*(3), 339-348. https://doi.org/10.1016/0377-2217(84)90155-3

Nagele-Piazza, L. (2020, February 21st). The Importance of Pay Equity. *SHRM.* https://www.shrm.org/topics-tools/news/hr-magazine/importance-pay-equity

Nesmachnow, S. (2014). An Overview of Metaheuristics: Accurate and Efficient Methods for Optimization. *International Journal of Metaheuristics, 3*(4), 320-347. 10.1504/IJMHEUR.2014.068914

Pulido, A., Stone, D., and Strevel, J. (2014, March 1st). The Three Cs of Customer Satisfaction: Consistency, Consistency, Consistency. *McKinsey & Company.*

https://www.mckinsey.com/industries/retail/our-insights/the-three-cs-of-customer-satisfaction-consistency-consistency-consistency#/

Ruhl, K., and Solimine, P. (n.d.). *Mapping in Python*. QuantEcon. https://datascience.quantecon.org/tools/maps.html

Segal, M., and Weinberger, D. B. (1977). Turfing. *Operations Research, 25*(3), 367-386. https://doi.org/10.1287/opre.25.3.367

Sharifi, S. (2015). *A Decision Support System for Sales Territory Planning Using the Genetic Algorithm.* [Unpublished master's thesis, Technische Universität München (Technical University of Munich)]. https://cartographymaster.eu/wp-content/theses/2015_Shahin_Thesis.pdf

Sharifi, S., and Murphy, C. E. (2017). Balanced Allocation of Multi-criteria Geographic Areas by a Genetic Algorithm. *International Cartographic Conference,* 417-433. 10.1007/978-3-319-57336-6_29

Shieh, HM., and May, MD. (2001). Solving the Capacitated Clustering Problem with Genetic Algorithms. *Journal of the Chinese Institute of Industrial Engineers, 18*(3), 1-12. https://doi.org/10.1080/10170660109509453

Smith, S. A. (1979). Estimating Service Territory Size. *Management Science, 25*(4), 301-311. https://doi.org/10.1287/mnsc.25.4.301

Solgi, R. M. (n.d.). *Geneticalgorithm.* Pypi. https://pypi.org/project/geneticalgorithm/

Sweary, R. (2019, April 12th). The Death of Dirty Data: The Importance of Keeping Your Database Clean. *Forbes.*

https://www.forbes.com/councils/forbestechcouncil/2019/04/12/the-death-of-dirty-data-the-importance-of-keeping-your-database-clean/

Tatis, V. A., and Parsopoulos, K. E. (2016). Grid Search for Operator and Parameter Control in Differential Evolution. *Proceedings of the 9th Hellenic Conference on Artificial Intelligence,* 1-7. https://doi.org/10.1145/2903220.2903238

Wright, C. M., and Mechling, G. (2002). The Importance of Operations Management Problems in Service Organizations. *The International Journal of Management Science,* 30, 77-87. 10.1016/S0305-0483(01)00058-5

Xu, D., and Tian, Y. (2015). A Comprehensive Survey of Clustering Algorithms. *Annals of Data Science, 2*(2), 165-193. https://doi.org/10.1007/s40745-015-0040-1

Appendix A: Project Equations with Explanations

Traditional K-Means:

The K-Means clustering algorithm functions by updating the centers for each of *K* clusters until a convergence criterion for minimizing the within-cluster sum-of-squares is met (Xu & Tian, 2015). In this approach, if a set of *n* points, or in this case *n* customer locations, of the form $\{X_1, X_2, \ldots, X_n\}$ is represented by a set *S* and the *K* clusters are represented by $Z_1, Z_2, \ldots, Z_k$ then Equation 1A below applies.

Equation 1A

The Traditional Process of Clustering

$$Z_i \neq 0 \; for \; i = 1, \ldots, K$$

$$Z_i \cap Z_j = \emptyset \; for \; i = 1, \ldots, K, j = 1, \ldots, K, and \; i \neq j$$

$$\bigcup_{i=1}^{K} Z_i = S$$

Manhattan Distance

For a customer in the cartesian plane represented as an (*x, y*) pair, where x is the longitude and y is the latitude for the customers location, the Manhattan Distance, *D,* between the customer and a centroid point (c_x, c_y) in the same plane is represented in Equation 2A below.

Equation 2A

The Manhattan Distance Equation between Customer and Centroid Points

$$D = |x - c_x| + |y - c_y|$$

Radial Compactness

Bacao et al. (2005) describe radial compactness as a penalty term that forces zones to be as “circular” as possible (Bacao et al., 2005). For this project, radial compactness can be considered as the sum of the sum of Manhattan Distances between the centroid point and all customer location points assigned to the centroid cluster for all zones. Mathematically, the equation for radial compactness among all *K* clusters, where D_{ij} represents the Manhattan Distance between the *ith* customer location unit and the *jth* zone center, can be found in Equation 3A below.

Equation 3A

Mathematical Formulation for Radial Compactness

for a solution of K clusters with j zone centers,

The radial compactness is given by:

$$\sum_{j} \sum_{i= \in Zj} D_{ij} = \left|x_{ij} - c_{xj}\right| + \left|y_{ij} - c_{yj}\right|$$

In the equation, x_{ij} represents the longitudinal value for the *ith* customer in the *jth* zone, y_{ij} represents the latitudinal value for the *ith* customer in the *jth* zone, c_{xj} represents the longitudinal (x) value for the *jth* zone, and c_{yj} represents the latitudinal (y) value for the *jth* zone. These values combine to give customer points (x_{ij}, y_{ij}), and centroid points (c_{xj}, c_{yj}).

Objective Fitness Function

Whether in the initial Manhattan Distance K-Means starter solution or an individual chromosome of the ZoneBadger heuristic, the objective fitness function value (OFFV) of any solution to the Green Field Planning Territorial Design Problem for Turf Badger can be calculated according to Equation 4A below.

Equation 4A

Calculating the Objective Fitness Function Value for any Territorial Design Solution

for a solution of K clusters with j zone centers:

$$OFFV = \sigma\left[\sum_{i= \in Zj} T_{ij}\right] + \sum\left[\sum_{i= \in Zj} D_{ij}\right]$$

In the equation, sigma (σ) represents the standard deviation in the sum of technician incomes for each of the *K* zones, T_{ij} represents the technician income for the *ith* customer assigned to the *jth* zone center, and D_{ij} represents the radial compactness for the *ith* customer assigned to the *jth* zone center. In non-mathematical notation, the equation can be read as the standard deviation of the total technician income across *K* zones plus the sum of the total radial compactness value for each of the *K* zones.

As an example, for the starter K-Means solution with standard deviations of [8142.43, 34545.58, 36514.68, 32366.01, 16829.88, 21596.45] and radial compactness values of [11.25, 72.48, 31.69, 32.74, 21.69, 53.89] for each zone respectively, the *OFFV* value would be computed as follows:

- $OFFV = \sigma$[8142.43, 34545.58, 36514.68, 32366.01, 16829.88, 21596.45] + sum[11.25, 72.48, 31.69, 32.74, 21.69, 53.89]
- *OFFV* = [10331.80] + [223.74]
- *OFFV* = 10555.55

Appendix B: Project Code

Jupyter Notebooks accessed via the Anaconda interface were chosen to implement the solution for this project in the Python coding language. The notebooks with complete code and commenting can be found in the GitHub repositories below.

1. Initial Data Cleaning Notebook:

   https://github.com/BruceTree1017/DS785_CAPSTONE/blob/36e1c2b2c8c06c6687a23fcd1eefb5b30ec9593f/BRUCEA_DS785Capstone_Code_P1_Cleaning.ipynb

2. ZoneBadger Heuristic Notebook:

   https://github.com/BruceTree1017/DS785_CAPSTONE/blob/36e1c2b2c8c06c6687a23fcd1eefb5b30ec9593f/BRUCEA_P2_GeneticAlgorithm.ipynb