

# **Waste Collection & Street-Sweeping Route Optimization using a 2-Stage Cluster Algorithm & Heuristic Approaches**

by

Tyler Parsons

A thesis submitted to the  
School of Graduate and Postdoctoral Studies in partial  
fulfillment of the requirements for the degree of

**Master of Applied Science in Mechanical Engineering**

Faculty of Engineering and Applied Sciences

University of Ontario Institute of Technology (Ontario Tech University)

Oshawa, Ontario, Canada

April 2023

© Tyler Parsons, 2023

## THESIS EXAMINATION INFORMATION

Submitted by: **Tyler Parsons**

### **Master of Applied Science in Mechanical Engineering**

Thesis title: Waste Collection & Street-Sweeping Route Optimization using a 2-Stage Cluster Algorithm & Heuristic Approaches
--

An oral defense of this thesis took place on April 12, 2023 in front of the following examining committee:

#### **Examining Committee:**

Chair of Examining Committee	Dr. Ghaus Rizvi
Research Supervisor	Dr. Jaho Seo
Examining Committee Member	Dr. Xianke Lin
Thesis Examiner	Dr. Meaghan Charest-Finn

The above committee determined that the thesis is acceptable in form and content and that a satisfactory knowledge of the field covered by the thesis was demonstrated by the candidate during an oral examination. A signed copy of the Certificate of Approval is available from the School of Graduate and Postdoctoral Studies.

## **ABSTRACT**

Waste collection and street-sweeping play a vital role in public health, safety, and overall cleanliness. Since these processes cannot be ignored, they should be done in an efficient manner. The following thesis proposes a novel 2-stage clustering approach, namely the Static and Dynamic Clustering, to divide a municipalities road network into several operational areas in which the routes can be assigned. A method of generating optimal routes within the respective operational areas is also developed so statistics can be used to quantify the improvements made using the proposed clustering methods. The proposed algorithms were used to optimize the waste collection and street-sweeping processes in The City of Oshawa. The results of this work show that the proposed clustering algorithms can generate operational areas that better distribute the workload and overall simulated statistics when compared to existing configurations. Additionally, the proposed techniques may be applied to other routing applications, and other areas of research involving optimizing data partitions using clustering methods, such as machine learning.

**Keywords:** waste collection; street-sweeping; route optimization; GIS; clustering

## AUTHOR'S DECLARATION

I hereby declare that this thesis consists of original work of which I have authored. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize the University of Ontario Institute of Technology (Ontario Tech University) to lend this thesis to other institutions or individuals for the purpose of scholarly research. I further authorize University of Ontario Institute of Technology (Ontario Tech University) to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research. I understand that my thesis will be made electronically available to the public.



*T Parsons*

---

Tyler Parsons

## STATEMENT OF CONTRIBUTIONS

Part of the work described in Chapter 3 has been accepted for publication as:

**T. Parsons, J. Seo, D. Livesey, “Waste Collection Route Optimization for the City of Oshawa,”** in *2022 International Conference on Smart Transportation and Future Mobility (CSTFM 2022)*.

Part of the work described in Chapter 3 has been published as:

**T. Parsons, J. Seo, D. Livesey, “Waste Collection Area Generation Using a 2 Stage Cluster Optimization Process and GIS Data,”** *IEEE Access*, vol. 11, pp. 11849-11859, 2023. doi: 10.1109/ACCESS.2023.3241626.

Part of the work described in Chapter 4 is being reviewed for publication as:

**T. Parsons, J. Seo, D. Livesey, “Municipal Street-Sweeping Area Generation with Route Optimization,”** *Intl. Trans. In Op. Res.*

Concepts based on the work described in Chapter 4 has been published as:

**T. Parsons, F. Hanafi Sheikhha, O. Ahmadi Khiyavi, J. Seo, W. Kim, S. Lee, “Optimal Path Generation with Obstacle Avoidance and Subfield Connection for an Autonomous Tractor,”** *Agriculture*, vol. 13 (1), 2023. doi: 10.3390/agriculture13010056.

## ACKNOWLEDGEMENTS

I would like to sincerely thank my supervisor, Dr. Jaho Seo for his continuous support and patience throughout my academic journey. The accomplishments I've made within the last 2 years of my study would simply not have been possible without his guidance, insight, and kindness.

I would also like to thank Dan Livesey, Michelle Whitbread, Julie MacIsaac, and Jen Plishewsky from the City of Oshawa for allowing me to collaborate with them through the TeachingCity Oshawa partnership. Their support and interest in my research allowed me to bridge the gap between theoretical concepts and ideas to real world applications in a way I could not have done alone.

My sincere thanks to the AVEC lab team at Ontario Tech. I am proud to be part of such a kind, brilliant group of researchers.

Finally, I would like to thank my friends and family for being by my side throughout this journey. I simply could not have done this without them.

## TABLE OF CONTENTS

<b>Thesis Examination Information</b> .....	ii
<b>Abstract</b> .....	iii
<b>Authors Declaration</b> .....	iv
<b>Statement of Contributions</b> .....	v
<b>Acknowledgements</b> .....	vi
<b>Table of Contents</b> .....	vii
<b>List of Tables</b> .....	x
<b>List of Figures</b> .....	xi
<b>List of Abbreviations and Symbols</b> .....	xiii
<b>Chapter 1. Introduction</b> .....	<b>1</b>
<b>1.1. Introduction</b> .....	1
<b>1.2. Scope and Objectives</b> .....	2
<b>1.3. Outline of Thesis</b> .....	3
<b>1.4. Problem Background &amp; Working Foundations</b> .....	4
<b>1.4.1. Waste Collection and Street-sweeping in The City of Oshawa</b> .....	4
<b>1.4.2. QGIS and GIS Data</b> .....	6
<b>1.4.3. Capacitated Arc Routing Problem (CARP)</b> .....	7
<b>1.4.4. Graph Traversal Algorithms</b> .....	8
<b>1.4.5. Optimization Algorithms</b> .....	12
<b>1.4.6. Clustering Algorithms</b> .....	16
<b>1.4.7. Parallel Computing</b> .....	19
<b>Chapter 2. Literature Review</b> .....	<b>20</b>
<b>2.1. Introduction</b> .....	20
<b>2.2. Routing Problems &amp; Models</b> .....	20
<b>2.2.1. Node Routing Problems</b> .....	21
<b>2.2.2. Edge Routing Problems</b> .....	21
<b>2.3. Clustering &amp; Zoning Approaches</b> .....	26
<b>2.4. Smart Routing Approaches</b> .....	27
<b>2.5. GIS-Based Approaches</b> .....	28
<b>2.6. Workload Balancing</b> .....	29
<b>2.7. Conclusions</b> .....	30

<b>Chapter 3. Waste Collection Route Optimization .....</b>	<b>33</b>
<b>3.1. Introduction.....</b>	<b>33</b>
<b>3.2. Data Preparation .....</b>	<b>34</b>
<b>3.2.1. Supplied &amp; Downloaded Data .....</b>	<b>34</b>
<b>3.2.2. Route Shapefiles.....</b>	<b>38</b>
<b>3.3. Waste Collection Route Optimization Algorithm .....</b>	<b>40</b>
<b>3.4. Dimensionless Objective Function .....</b>	<b>43</b>
<b>3.4.1. Waste Collection Route Objective Function Values .....</b>	<b>43</b>
<b>3.4.2. Waste Collection Area Objective Function Values.....</b>	<b>46</b>
<b>3.5. Workload Balance (Waste Collection Area Pairing) Algorithm .....</b>	<b>47</b>
<b>3.6. Waste Collection Area Generation Algorithms.....</b>	<b>51</b>
<b>3.6.1. GIS Data Preparation for the 2-Stage Clustering Algorithm.....</b>	<b>52</b>
<b>3.6.2. Static Clustering.....</b>	<b>53</b>
<b>3.6.3. Dynamic Clustering .....</b>	<b>54</b>
<b>3.7. Results &amp; Analysis.....</b>	<b>56</b>
<b>3.7.1. Current Configuration Analysis.....</b>	<b>56</b>
<b>3.7.2. Clustered Configuration Analysis .....</b>	<b>61</b>
<b>3.8. Conclusions.....</b>	<b>70</b>
<b>Chapter 4. Street-sweeping Route Optimization.....</b>	<b>72</b>
<b>4.1. Introduction.....</b>	<b>72</b>
<b>4.2. Data Preparation .....</b>	<b>73</b>
<b>4.3. Street-sweeping Route Optimization Algorithm .....</b>	<b>75</b>
<b>4.3.1. 3-Phase Augment Merge Algorithm.....</b>	<b>78</b>
<b>4.3.2. U-Turn Minimization Algorithm .....</b>	<b>81</b>
<b>4.3.3. U-Turn Removal Algorithm &amp; Redundant Edge Reduction.....</b>	<b>84</b>
<b>4.4. Street-sweeping Area Generation Algorithms.....</b>	<b>88</b>
<b>4.4.1. Static Clustering.....</b>	<b>89</b>
<b>4.4.2. Dynamic Clustering .....</b>	<b>92</b>
<b>4.5. Results &amp; Analysis.....</b>	<b>92</b>
<b>4.5.1. Current Configuration Analysis.....</b>	<b>92</b>
<b>4.5.2. Clustered Configuration Analysis .....</b>	<b>97</b>



4.6. Conclusions.....	104
<b>Chapter 5. Conclusion.....</b>	<b>106</b>
5.1. Conclusions.....	106
5.2. Recommendations and Future Work .....	108
<b>REFERENCES... ..</b>	<b>110</b>
<b>APPENDICIES... ..</b>	<b>117</b>
<b>Appendix A. Waste Collection.....</b>	<b>117</b>
1. Route Statistics (Existing – Garbage and Organics).....	117
2. Route Statistics (Clustered – Garbage and Organics) .....	120
<b>Appendix B. Street-sweeping.....</b>	<b>124</b>
1. RES Route Statistics (Current - Fall).....	124
2. AC Route Statistics (Current - Fall).....	127
3. RES Route Statistics (Clustered - Fall) .....	128
4. AC Route Statistics (Clustered - Fall) .....	130

## LIST OF TABLES

### CHAPTER 3

Table 3-1 Supplied Shapefiles and Their Useful Information .....	34
Table 3-2 Service Condition and Corresponding Simulation Variables .....	45
Table 3-3 Objective Function and Simulation Parameters .....	46
Table 3-4 Improvements Made by the Workload Balance Algorithm .....	61
Table 3-5 Static and Dynamic Cluster Optimization Parameters for Waste Collection ...	62
Table 3-6 Dwelling Deviation Improvements.....	64
Table 3-7 Static and Dynamic Cluster Optimization Parameters for the Vehicle Assignment.....	65
Table 3-8 Total City Statistics for the Current Arrangement and Clustered Arrangement for Week 1 Collection.....	67
Table 3-9 GA Parameters for the Workload Balance Algorithm.....	68
Table 3-10 Comparison of All Solutions for the Workload Balance Problem .....	70

### CHAPTER 3

Table 4-1 Street-sweeping Constraints .....	80
Table 4-2 Route Simulation Parameters .....	80
Table 4-3 Experimental Trip Rates for RES and AC Road Classes for Arbitrary Clusters .....	89
Table 4-4 Optimization Parameters for the Street-sweeping Route Optimization Algorithm.....	93
Table 4-5 Fall Statistics for the Current Sweeping Areas .....	95
Table 4-6 Spring/Summer Statistics for the Current Sweeping Areas .....	96
Table 4-7 Static and Dynamic Cluster Optimization Parameters for Street-sweeping .....	98
Table 4-8 Fall Statistics for the Clustered Sweeping Areas.....	100
Table 4-9 Spring/Summer Statistics for the Clustered Sweeping Areas .....	101
Table 4-10 Improvements Made Using the Clustered Configuration in Fall .....	103
Table 4-11 Improvements Made Using the Clustered Configuration in Spring/Summer .....	103

# LIST OF FIGURES

## CHAPTER 1

Figure 1-1 Waste Collection Area Configuration for The City of Oshawa [1] .....	5
Figure 1-2 Road Network GIS Data for The City of Oshawa.....	7
Figure 1-3 Stop Locations (Pink) for the Curbside Waste Collection.....	8
Figure 1-4 Dijkstra's Algorithm Example for a Simple Graph .....	9
Figure 1-5 A* Example for a Simple Graph .....	10
Figure 1-6 ACO Example .....	11
Figure 1-7 Hierholzer's Algorithm to Create an Eulerian Tour .....	12
Figure 1-8 GA Crossover Phase.....	14
Figure 1-9 GA Mutation Phase .....	14
Figure 1-10 DE Example .....	15
Figure 1-11 Clustering 2D Data .....	17

## CHAPTER 2

Figure 2-1 Directional Demand for the Curbside Waste Collection .....	22
Figure 2-2 Multiple Passes to Sweep Curbs on Either Side of the Road .....	25
Figure 2-3 IoT Waste Bin Fill Level Example.....	28
Figure 3-1 Monday Yellow - Route 8 PDF Map.....	36

## CHAPTER 3

Figure 3-2 Road Connectivity Information - Downloaded Open-Source Dataset .....	38
Figure 3-3 Side-By-Side Comparison of Route PDF Map and Shapefile .....	39
Figure 3-4 "route_block" Stop Location Filtering Algorithm – Visualized .....	40
Figure 3-5 Importance of Optimal Starting and Ending Node Connections .....	41
Figure 3-6 Example of a Safe U-Turn Location (a), and Unsafe U-turn Location (b).....	42
Figure 3-7 All Possible Collection Area Combinations for the Current Configuration....	48
Figure 3-8 Duplicates in Chromosome Structure After Crossover .....	49
Figure 3-9 Joining Dwelling Counts to Road Segments .....	52
Figure 3-10 Frame-by-Frame Animation Example.....	56
Figure 3-11 Python-Generated Statistic Table for the Purple Friday Area in Week 1 Collection .....	57
Figure 3-12 Workload Distribution for the Current Collection Area Configuration in Week 1 (a) and Week 2 (b) .....	57
Figure 3-13 Current Configuration (a) vs. Solution 1 Configuration (b) .....	58
Figure 3-14 Workload Distribution for Solution 1 Collection Area Configuration in Week 1 (a) and Week 2 (b) .....	59
Figure 3-15 Current Configuration (a) vs. Solution 2 Configuration (b) .....	59

Figure 3-16 Workload Distribution for Solution 2 Collection Area Configuration in Week 1 (a) and Week 2 (b) .....	60
Figure 3-17 Current Collection Area Configuration Dwelling Deviation .....	61
Figure 3-18 Clusters Produced by the Static Cluster Optimization Process (a), and the Evolutionary Improvement Trend (b).....	63
Figure 3-19 Clusters Produced by the Dynamic Cluster Optimization Process (a), and the Evolutionary Improvement Trend (b).....	64
Figure 3-20 Vehicle Assignment using Static and Dynamic Clustering in Cluster 6 .....	65
Figure 3-21 Current Configuration (a) vs. Clustered Configuration with Collection Weeks and Daily Pairs Assigned (b).....	69

## CHAPTER 4

Figure 4-1 RES (Red) and AC (Blue) Roads in The City of Oshawa .....	74
Figure 4-2 Route Area PDF (a) to Shapefile (b) Conversion .....	75
Figure 4-3 General Area of South Oshawa.....	76
Figure 4-4 Overall Street-sweeping Route Optimization Algorithm Flow .....	77
Figure 4-5 Cycle Permutation Example for U-Turn Minimization.....	82
Figure 4-6 FS-ACO for U-Turn Removal .....	85
Figure 4-7 FS-ACO First (a), and TR First (0.911 km Improvement) (b) .....	87
Figure 4-8 FS-ACO First (a) (0.18 km Improvement), and TR First (b) .....	87
Figure 4-9 Depot-to-Cluster-Center Issue (Pink Area) .....	90
Figure 4-10 Cluster Convex Hulls.....	91
Figure 4-11 Route Figure (a) and Sweeping Area Figure (b) for Fall.....	94
Figure 4-12 Normalized Total Statistic Plot for Routes in Existing Sweeping Area 1 in Fall .....	95
Figure 4-13 Normalized Total Statistic Plots for the Existing RES Areas in Fall (a) and Spring/Summer (b) .....	96
Figure 4-14 Normalized Total Statistic Plots for the Existing AC Areas in Fall (a) and Spring/Summer (b) .....	97
Figure 4-15 RES Clusters Produced by the Dynamic Cluster Optimization Process (a), and the Evolutionary Improvement Trend (b) .....	99
Figure 4-16 AC Clusters Produced by the Dynamic Cluster Optimization Process (a), and the Evolutionary Improvement Trend (b).....	99
Figure 4-17 Normalized Total Statistic Plots for the Clustered RES Areas in Fall (a) and Spring/Summer (b) .....	101
Figure 4-18 Normalized Total Statistic Plots for the Clustered AC Areas in Fall (a) and Spring/Summer (b) .....	102

## LIST OF ABBREVIATIONS AND SYMBOLS

RES	Residential
AC	Arterial & Collector
CBD	Central Business District
GIS	Geographic Information System
CARP	Capacitated Arc Routing Problem
VRP	Vehicle Routing Problem
ACO	Ant Colony Optimization
GA	Genetic Algorithm
DE	Differential Evolution
TS	Tabu Search
TR	Transitive Reduction
TSP	Travelling Salesman Problem
WK-Means	Weighted K-Means
2D	Two Dimensional
CPU	Computer
SA	Simulated Annealing
CVRP	Capacitated Vehicle Routing Problem
MDVRP	Multi-Depots Vehicle Routing Problem
PSO	Particle Swarm Optimization
VNS	Variable Neighborhood Search
CPP	Chinese Postman Problem
DCPP	Directed Chinese Postman Problem
DRCPP	Directed-Rural Chinese Postman Problem
WRCPP	Windy-Rural Chinese Postman Problem
GTSP	Generalized Travelling Salesman Problem
NN	Nearest Neighbor

RNN	Repetitive Nearest Neighbor
INN	Improved Nearest Neighbor
RINN	Repetitive Improved Nearest Neighbor
NLN	Loneliest Neighbor
PCARP	Periodic Capacitated Arc Routing Problem
MCCARPIF	Multi-Compartment Capacitated Arc Routing Problem with Intermediate Facilities
ALNS	Adaptive Large Neighborhood Search Algorithm
LARP	Location-Arc Routing Problem
IHAC	Improved Hierarchical Agglomerative Clustering
GCPP	Garbage Collection Path Planning
CluVRP	Clustered Vehicle Routing Problem
GPS	Global Positioning System
IoT	Internet of Things
ANN	Artificial Neural Network
SANS	Simulated Annealing and Neighborhood Search
FS-ACO	Forward-Searching Ant Colony Optimization

# Chapter 1. Introduction

## 1.1. Introduction

Route optimization is a broad, yet complicated area of research that spans over many disciplines, including waste collection, street-sweeping, snowplowing, mail delivery, and many others. Common to all areas of route optimization, some organizations rely upon humans to create service routes manually while using their own judgment. In smaller cases, this seems like an ideal approach. However, manually creating servicing routes on a large scale can prove to be a time consuming and inefficient method that yields results far from optimal. Additionally, developing areas typically undergo rapid expansion and population growth. As a result, complications with existing servicing routes arise, and a method of efficiently updating existing routes, or creating new routes can be beneficial.

In large municipalities, it is common to divide a large area into several smaller operational regions or zones. This is seen in applications such as curbside waste collection and street-sweeping and allows for better dispatching organization and workload distribution for the operations team. However, creating arbitrary areas without any knowledge of the service demand in these areas can lead to an unbalanced workload and statistics distribution, resulting in less efficient routes. For the case of waste collection, some stop locations may generate more waste (in the case of multi-dwelling households), or in street-sweeping, some streets may accumulate more debris (in the case of dense tree canopies overhanging the roads), and it is critical that these areas are treated differently since they have an influence on the expected number of trips due to the capacity constraint.

Additionally, servicing vehicles of any type have several operational constraints that must be considered. Vehicle constraints such as debris and fuel capacity, and operational

constraints such as shift length and u-turn avoidance all play an important role in route optimization. With respect to creating routes manually, these concepts are difficult (if not, impossible) to incorporate.

Currently, route optimization is a largely researched area with many heuristic approaches that have been proposed and validated. However, some municipalities have different vehicle and operational constraints which require the development of new heuristic methods to properly optimize servicing routes.

## **1.2. Scope and Objectives**

The primary scope of this research is to develop novel heuristic algorithms capable of optimizing waste collection and street-sweeping routes for The City of Oshawa. The route optimization methods are to be applied on a city-wide scale, making use of large datasets representing the city's complex road network. Due to the operational differences between waste collection and street-sweeping, different heuristics will be developed for each respective area.

The detailed objectives of the waste collection route optimization include:

- The development of a waste collection route optimization algorithm
- The development of a collection area pairing algorithm with workload balancing
- The development of a collection area generation algorithm with expected waste distribution balancing
- A detailed comparison of the existing collection area configuration vs. the generated collection area configuration

The detailed objectives of the street-sweeping route optimization include:



- The development of a street-sweeping route optimization algorithm capable of multi season (Spring/Summer/Fall) interpretation and unsafe u-turn elimination
- The development of a street-sweeping area generation algorithm with service distance distribution balancing
- A detailed comparison of the existing street-sweeping area configuration vs. the generated street-sweeping area configuration

### **1.3. Outline of Thesis**

**Chapter 1** introduces the research scope, objectives, thesis outline, and working foundations used to achieve the research objectives.

**Chapter 2** provides an in-depth literature review regarding route optimization models, route optimization algorithms, and clustering algorithms used in routing applications.

**Chapter 3** explains the developed heuristic used for waste collection route optimization in The City of Oshawa. The simulated results from the current collection area configuration were compared to the optimized collection area configuration.

**Chapter 4** explains the developed heuristic used for waste street-sweeping route optimization in The City of Oshawa. The simulated results from the current street-sweeping area configuration were compared to the optimized street-sweeping area configuration.

**Chapter 5** concludes the thesis by highlighting the contributions made to the route optimization research discipline and states the limitations of the proposed methods. Future research objectives and recommendations are also discussed.

## **1.4. Problem Background & Working Foundations**

The problem background, working foundations, algorithms, and methodologies presented in this thesis will be discussed in the following section. The operational principle of waste collection and street-sweeping in The City of Oshawa will be explained. Next, the data preparation and software used will be discussed. From there, the utilized graph traversal algorithms will be explored. Followed by the optimization algorithms selected. The chosen clustering algorithms will be explained. Finally, parallel computing methods and resources will be explored.

### **1.4.1. Waste Collection and Street-sweeping in The City of Oshawa**

This section will explain the operational principle behind the waste collection and street-sweeping operations in The City of Oshawa. These processes differ from municipality to municipality, so it is important to define the working procedure being examined in the following research.

#### **1.4.1.1. Waste Collection**

The City of Oshawa is divided into 10 different areas [1] (2 for each day of the working week) for curbside waste collection, and each of the 10 areas is divided into 11 individual routes. Each day of the week, a fleet of 11 vehicles (made up of the Labrie Expert Co-Mingle Split Side Loader and the Shu-Pak PK Split Side Loader) is dispatched to service the 2 collection areas corresponding to the current day of the week. The vehicles will service a route in each area. The collection week alternates between Week 1 and Week 2, meaning that one area will have only organics collected, while the other has garbage and organics collected. Each dwelling may have up to 4 bags of waste which can weigh up to 20 kg each, and 1 green bin that can weigh up to 20 kg [1], and there may be multiple

dwellings at a single stop (e.g., basement apartments). The collection areas can be seen in Figure 1-1.

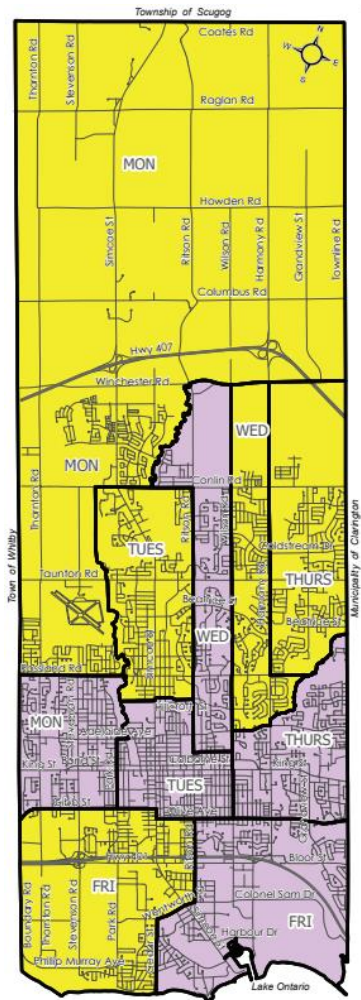


Figure 1-1 Waste Collection Area Configuration for The City of Oshawa [1]

#### 1.4.1.2. Street-sweeping

The City of Oshawa consists of 3 road classes, residential (RES), arterial and collector (AC), and central business district (CBD) for street-sweeping, each with their own set of standards. On AC roads, the travel speed and traffic volume are typically larger than RES roads, and RES roads contain direct connections (driveways) to most of the housing

in the city. For this study, only the RES and AC road classes will be considered as the CBD is a localized area that is treated differently than RES and AC roads. The AC roads can be divided into 5 sweeping areas, and RES can be divided into 12 different areas that are swept by Elgin Whirlwind mechanical street sweepers [2].

AC sweeping areas are made up of a combination of city owned roads and regional owned roads, where all AC roads are swept in Spring and Summer, but only the city owned roads are swept in the Fall. RES roads are swept all 3 seasons; however, 2 subclasses of RES roads exist, canopy and non-canopy. Canopy roads are defined as roads with street or boulevard trees that have a denser canopy overhead, contributing to more debris generation (via. leaves) in the Fall season. It should be noted that canopy roads will change over time as some trees may die, and eventually non-canopy roads may become canopy roads as small trees will grow. Non-canopy roads still accumulate leaves in the Fall, but canopy roads accumulate more and must be treated as such, possibly requiring frequent and multiple trips due to the increase in volume of material collected. All roads must be swept twice, once in each direction for 2-way roads, and twice in the same direction for 1-way roads.

#### **1.4.2. QGIS and GIS Data**

Geographic information system (GIS) data was supplied by The City of Oshawa, and QGIS [3] is used to view/manipulate the data. GIS integrates location data (map) with descriptive data (attributes) [4]. The supplied GIS data represents the road network and can be seen in Figure 1-2, where each line is a road segment. Each road segment has several useful attributes associated with it that otherwise cannot be seen visually, such as speed limit, road class, jurisdiction, etc.

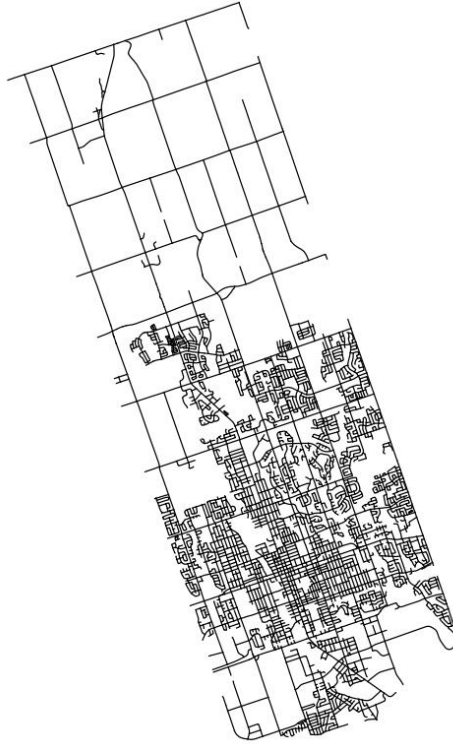


Figure 1-2 Road Network GIS Data for The City of Oshawa

### **1.4.3. Capacitated Arc Routing Problem (CARP)**

The capacitated arc routing problem (CARP) model is commonly used to optimize servicing routes consisting of roads with demand and cost [5]. This directly correlates to waste collection in the form of collecting household waste and organics, and street-sweeping in the form of debris collection. CARP can be used to model 1 or many servicing vehicles, which makes it a versatile approach for many applications. In most literature, the cost is typically the travel distance, and the demand is specific to the application. However, as the routing problem becomes increasingly complex, there may be several cost and demand variables.

CARP differs from the traditional vehicle routing problem (VRP) approach in the sense that the roads/edges themselves need to be serviced instead of servicing locations/nodes [6]. Although the curbside waste collection problem can technically be modelled as a VRP, it is assumed to be an edge routing problem because of the many servicing locations that exist on a road segment, as seen in Figure 1-3.

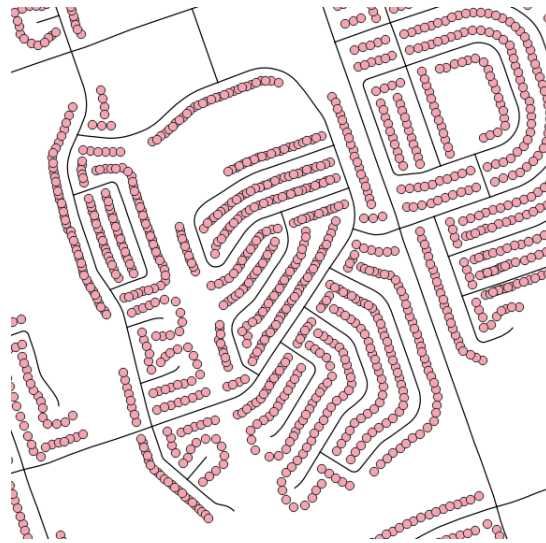


Figure 1-3 Stop Locations (Pink) for the Curbside Waste Collection

#### **1.4.4. Graph Traversal Algorithms**

This section will explain the graph traversal algorithms used in the following research. Specifically, the operation principles and importance of Dijkstra's algorithm, A\* algorithm, ant colony optimization (ACO), and Hierholzer's algorithm will be briefly discussed.

##### **1.4.4.1. Dijkstra's Algorithm**

Dijkstra's shortest path finding algorithm was proposed by E. W. Dijkstra in 1959 [7] and has been used by hundreds of researchers in many different path finding

applications. Dijkstra’s algorithm finds the shortest path from one node to all other nodes in a weighted graph. For this research, Dijkstra’s algorithm was used to connect pairs of nodes together in a road network such that all nodes have an in-degree equal to their out-degree. An example of Dijkstra’s algorithm can be seen in Figure 1-4.

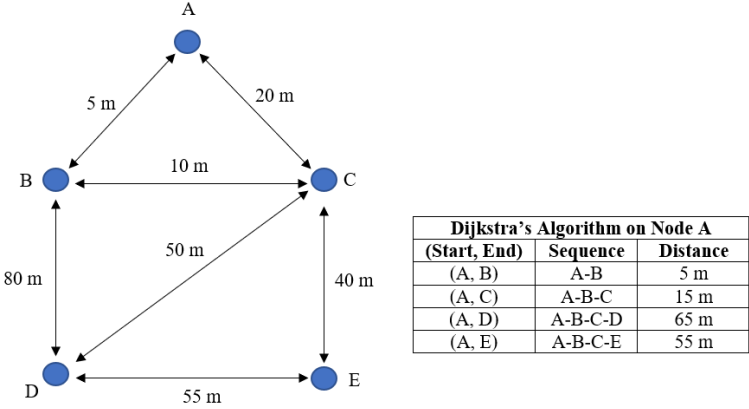


Figure 1-4 Dijkstra's Algorithm Example for a Simple Graph

**1.4.4.2. A\* Algorithm**

The A\* algorithm is similar to Dijkstra’s algorithm, however, A\* simply finds the shortest path between any 2 nodes in a strongly connected multigraph rather than 1 node and all other nodes [8]. A\* is a more informed version of Dijkstra’s algorithm because it consists of 2 cost functions, one to keep track of the current path distance (*gcost*), and the other to keep track of the Euclidian distance to the goal node (*hcost*). Each candidate node has a total cost that is equal to the sum of the *gcost* and *hcost*, thus providing insight as to which nodes are in the direction of the goal node, eventually exploiting the guaranteed shortest path without the need to explore all nodes in the network. For this research, A\* was used to compute the shortest path between road segments in an efficient manner within a genetic algorithm (GA). A simple example of the A\* algorithm can be seen in Figure 1-5.

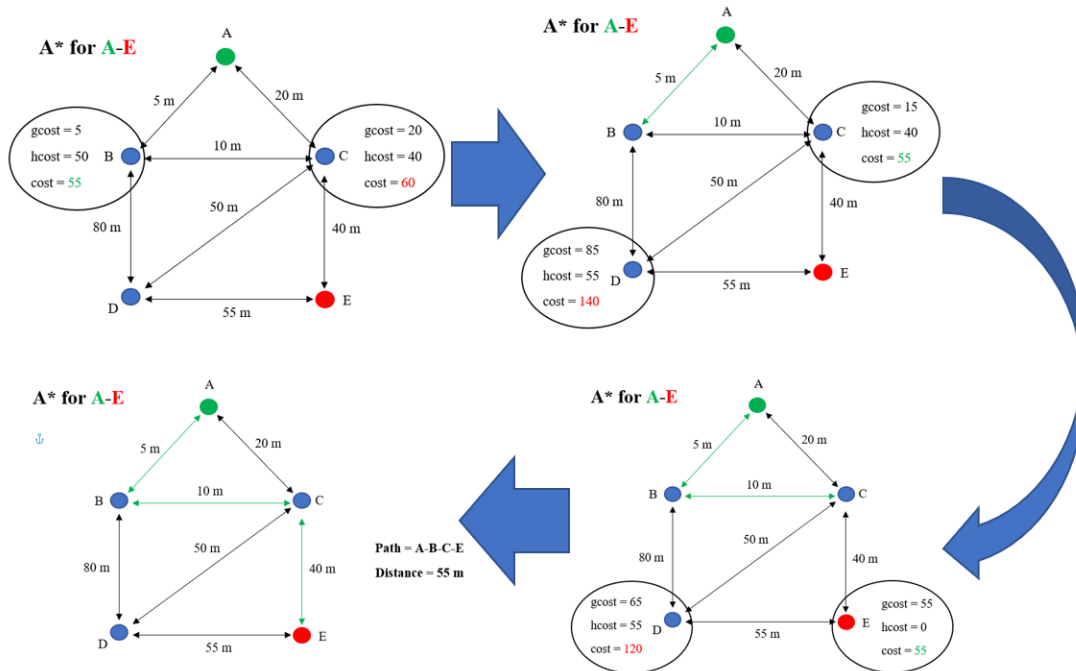


Figure 1-5 A\* Example for a Simple Graph

#### 1.4.4.3. Ant Colony Optimization (ACO)

Different versions of ACO exist, however they all share the same operational principle. ACO is a stochastic process developed for combinatorial optimization problems of many types [9]. ACO simulates a colony of ants searching for food, and the least expensive paths found by ants will be rewarded with greater pheromone levels, thus making them more desirable after each iteration. In accordance with nature, the pheromone levels slowly evaporate from all paths, exploiting the paths with the best fitness.

Each iteration simulates a colony of ants finding the shortest path to the goal location; a roulette style selection is used to choose the next edge to travel where the probability is proportional to the pheromone level of the corresponding edge. Once the ants reach the goal location, the algorithm stops and rewards the best path found by the colony.



ACO works best with a graph-like structure, which easily translates to a road network. For this research, the ACO algorithm was modified to work as a u-turn removal algorithm. A visual representation of the ACO algorithm can be seen in Figure 1-6.

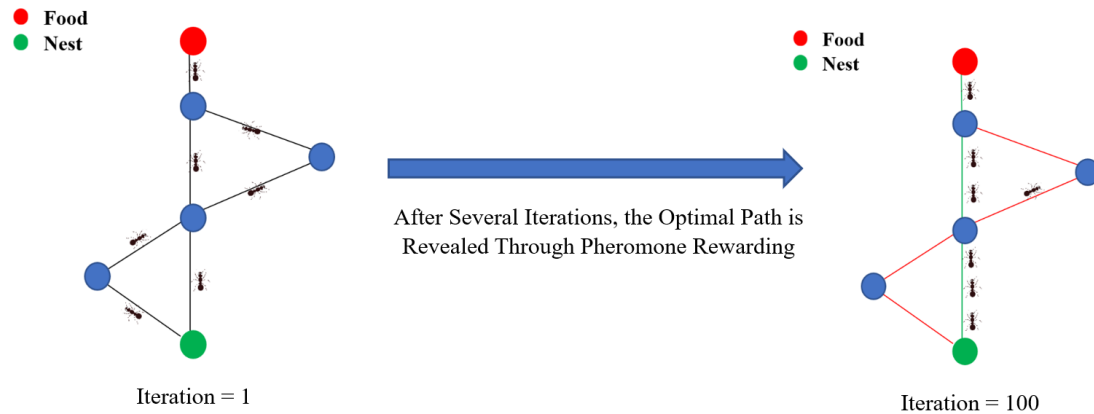


Figure 1-6 ACO Example

#### 1.4.4.4. Hierholzer's Algorithm

Hierholzer's algorithm is a graph traversal algorithm that is used to generate an Eulerian circuit that traverses each edge once in directed multigraphs with all nodes having an in-degree equal to the out-degree [10]. Additionally, Hierholzer's algorithm can create an Eulerian path in directed multigraphs only if the starting node has an in-degree equal to 1 more than the out-degree, and the ending node has an out-degree equal to 1 more than the in-degree.

Hierholzer's algorithm works by randomly selecting unvisited outgoing edges from the current node until a node is reached with no unvisited edges. When this occurs, the path is reversed until the next node is reached that has unvisited edges. While the path is being reversed, the edges are added to a separate list representing the Eulerian circuit/path. Once

all edges have been visited, the algorithm will stop, and the Eulerian circuit/path can be seen. For this research, Hierholzer’s algorithm was modified to minimize the number of u-turns in a servicing circuit/path. An example of Hierholzer’s algorithm being used to create an Eulerian tour can be seen in Figure 1-7.

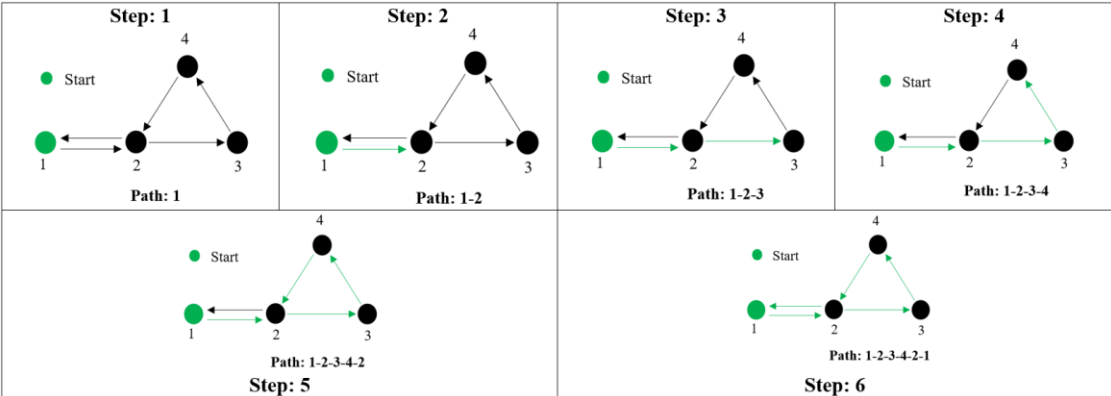


Figure 1-7 Hierholzer's Algorithm to Create an Eulerian Tour

**1.4.5. Optimization Algorithms**

This section will explain the optimization algorithms used in the following research. Specifically, the operation principles and importance of the GA, differential evolution (DE), tabu search (TS), and transitive reduction (TR) will be briefly discussed.

**1.4.5.1. Genetic Algorithm (GA)**

The GA is an evolutionary algorithm developed to replicate the natural process of evolution through survival of the fittest [11]. GAs are ideal for combinatorial optimization processes, but with proper encoding, they can be used for continuous optimization problems as well. To use a GA for the combinatorial case, the optimization problem should be formulated as a list of genes in a chromosome. For example, in the travelling salesman

problem (TSP), each gene in the chromosome would represent a city to visit, and the order of the genes in the chromosome would represent the order of which the cities will be visited.

The GA works by first creating an initial population of chromosomes with a random order or configuration of genes in accordance with the optimization problem. A fitness function (specific to the optimization problem) will be used to evaluate the fitness of each member in the population, where the best solutions are the ones with the lowest fitness score. Typically, a small sample of the best solutions (chromosomes) are copied to the next generation to ensure they do not get destroyed by the next phase, crossover, and mutation.

In the crossover phase, 2 parents will be randomly selected with probability proportional to their respective fitness, and their genes will be swapped at a randomly selected crossover point to create 2 children, an example of this can be seen in Figure 1-8. In the mutation phase, the genes of the children's chromosomes are iterated over and the ones that are selected for mutation are changed to a random variable in accordance with the optimization problem, an example of this can be seen in Figure 1-9. This process is repeated until a new population is created with the same size as the original one. Over several generations, the chromosomes will gradually improve fitness, and the best solution from the final generation is selected as the optimal solution. For this research, the GA was used to pair waste collection areas together with the objective of balancing the workload, and it is used as a part of the 3-phase augment merge algorithm to optimize the order of roads to be swept.

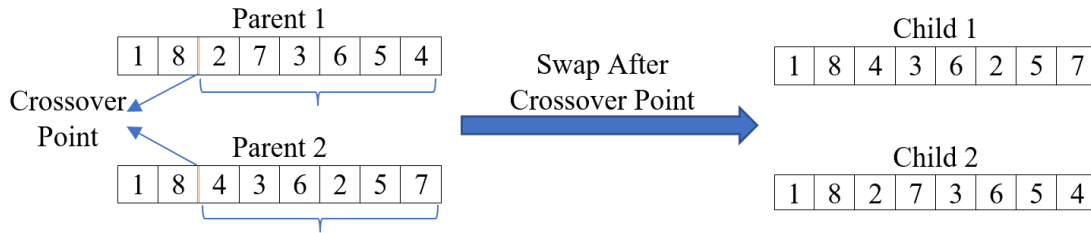


Figure 1-8 GA Crossover Phase

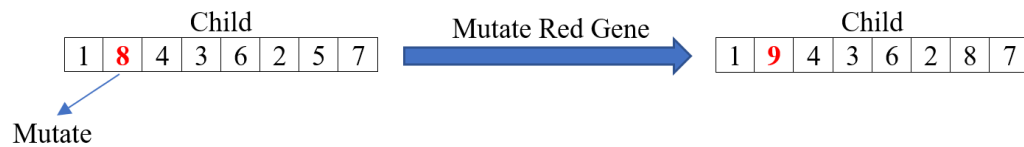


Figure 1-9 GA Mutation Phase

### 1.4.5.2. Differential Evolution (DE)

DE, like the GA, falls under the umbrella of evolutionary algorithms, but is more applicable to multi-dimensional continuous optimization problems [12]. DE also shares the concept of survival of the fittest through crossover and mutation, but this is done in a different manner due to the multi-dimensional nature of the problem.

DE works by first creating an initial population of chromosomes with a random order or configuration of genes in accordance with the optimization problem. The fitness of each member of the population is evaluated using an objective function specific to the optimization problem. Then the population is iterated over, the current member of the population is called the target vector, and 3 other chromosomes are randomly selected from the current population. A difference vector is created by subtracting 2 of the randomly selected chromosomes, and this is multiplied by a weight factor. This is called the weighted difference vector. The weighted difference vector is added to the 3<sup>rd</sup> randomly selected

chromosome, creating the noisy vector. The mutant vector is created by iterating through the genes of the target vector, using a mutation probability, the current gene will be selected from either the mutant vector or target vector. Once the mutant vector has been created, the fitness is compared to the target vector and the better solution is added to the next population. An example of the DE process can be seen in Figure 1-10. For this research, DE was used to tune the  $\beta$  values in the Static and Dynamic waste collection and street-sweeping area clustering.

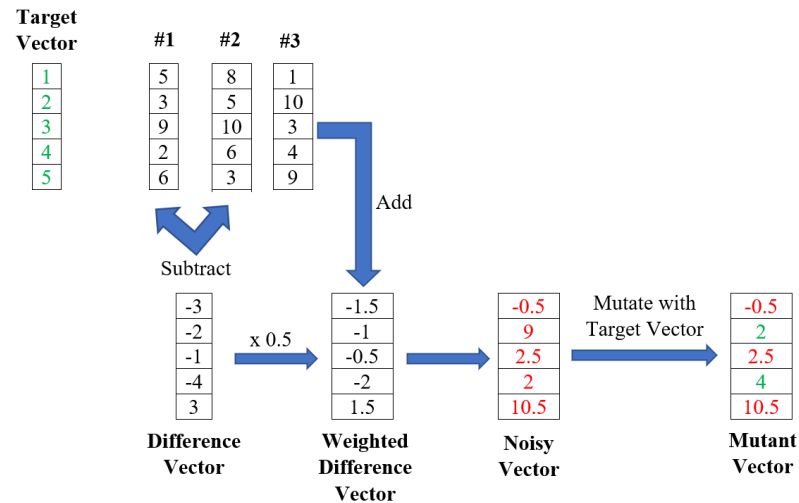


Figure 1-10 DE Example

### 1.4.5.3. Tabu Search (TS)

TS is a metaheuristic approach developed to overcome local optimality in combinatorial optimization problems by forbidding certain moves until a short time later [13]. TS can be used in many ways; however, all make use of the fixed length tabu list that stores recently evaluated solutions to prevent cycling.

To use TS, an initial solution must be formulated, and depending on the optimization problem, a change will be made in the initial solution to create a new solution.

If the new solution is not in the tabu list, it will be used as the current solution and added to the tabu list. However, if the solution is a part of the tabu list, another change will be made to the current solution until a solution that is not in the tabu list is found.

If the tabu list exceeds its defined length, the oldest solution will be removed from the list, thus allowing the solution to reappear again as a candidate. The fitness of each solution must be calculated, and the best-found solution must continuously update until the final iteration. For this research, TS was used to minimize u-turns in an Eulerian cycle/tour.

#### **1.4.5.4. Transitive Reduction (TR)**

A TR is defined as a subgraph of the original graph consisting of less edges. A TR must maintain the same strong connectivity as the original graph (all nodes reachable from each other) but it must also consist of the minimal number of edges to do so [14].

There are several ways to accomplish this. For removing redundancies in routing applications, an initial solution can be generated using an edge traversal algorithm, and cycles of the initial solution can be removed if they still yield a strongly connected subgraph. For this research, TR subgraphs were used to improve the solutions (reduce redundant deadhead travel) generated by the proposed heuristic.

#### **1.4.6. Clustering Algorithms**

This section will explain the clustering algorithms used in the following research. Specifically, the operation principles and importance of the weighted K-Means (WK-Means) algorithm, and the developed Static and Dynamic clustering approaches will be briefly discussed.

### 1.4.6.1. Weighted K-Means (WK-Means)

Clustering algorithms work by arranging a collection of instances into several clusters consisting of similar traits. One of the most commonly used clustering approaches is the K-Means algorithm, where the instances are clustered into  $k$  number of groups [15]. The K-Means algorithm works by selecting  $k$  number of instances to be the initial centers, and an equation is used to calculate the proximity of other instances surrounding each initial centers. The instances join the nearest cluster center, then the cluster center locations are updated. This process is repeated until the cluster centers converge, or the final iteration is reached. In the case of 2D data, a collection of points would be the instances, and the proximity equation would simply be the Euclidean distance. A visualization of clustering on 2D data can be seen in Figure 1-11.

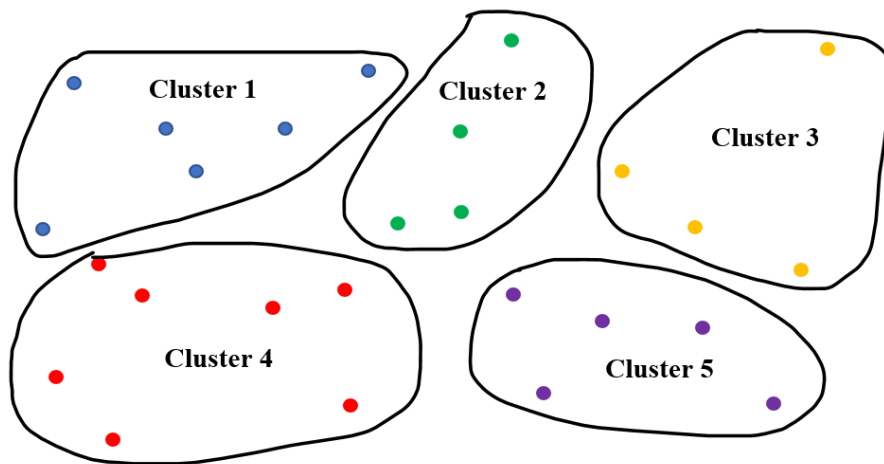


Figure 1-11 Clustering 2D Data

In many cases, K-Means would simply be enough to cluster data. However, a more informed version of the K-Means algorithm exists for weighted data called the WK-Means algorithm [16]. The WK-Means algorithm works like K-Means, except for the proximity

calculation which includes a variable to consider the respective cluster weights. Conceptually, this can be thought of as the gravitational pull (or push) that different clusters may have on the dataset. For this research, the WK-Means algorithm was used in the developed Static and Dynamic clustering algorithms.

#### **1.4.6.2. Static and Dynamic Clustering**

The Static and Dynamic clustering methods were developed to balance the weights of all clusters while still retaining spatial similarities of the dataset. By combining WK-Means and DE, the influence of the weights of each respective cluster can be optimized to yield clusters that are more balanced for the application. For optimal results, the Static cluster optimization should be used first, then the Dynamic cluster optimization should be used with the cluster centers and weights produced by the Static approach.

The Static approach uses DE to tune a single influence factor that will be applied to all clusters simultaneously in the WK-Means clustering algorithm. Over several generations, an optimal cluster configuration will be found with balanced weights. However, the clusters can still be improved by optimizing the influence factor of each cluster individually.

The Dynamic approach uses DE to tune each cluster's influence factor while keeping the cluster center locations and weights from the Static approach fixed. Each instance can be joined to the nearest weighted cluster center to generate a new cluster configuration; however, the cluster centers and weights do not update like they would in WK-Means. The distribution of cluster weights in the new cluster configuration will be used to evaluate the effectiveness of the influence factors. Using DE, each clusters influence factor will be tuned to yield the optimal cluster configuration over several



generations. For this research, the Static and Dynamic clustering algorithms were used to divide the City of Oshawa into several well-balanced servicing areas for each respective application.

#### **1.4.7. Parallel Computing**

Traditional programming workflows require tasks to be executed in a serial manner, one after another. However, as algorithms become increasingly complex and lengthy, serial methods are no longer ideal and require a clever workaround. With access to a large collection of computing resources, developers can modify their algorithms to allow for a parallel approach where applicable [17]. By carefully distributing tasks amongst several CPUs, they may be executed alongside one another, and thus do not need to wait for the previous task to finish before the next one can start. In Southern Ontario, SHARCNET grants access to a vast amount of remote computing resources to be used for parallel computing [18]. For this research, parallel computing methods were carefully integrated into the Static and Dynamic clustering methods and are also used to generate routes within different servicing areas.

## **Chapter 2. Literature Review**

### **2.1. Introduction**

In this Chapter, a detailed literature review is conducted to present the state-of-the art techniques used within the field of route optimization. Since waste collection and street-sweeping route optimization is a specialized area under the route optimization umbrella, other relevant routing applications (snowplow, mail delivery, etc.) will be explored which fall under the same scope as the research presented in this thesis.

Specifically, this chapter will cover routing problems and models, clustering and zoning approaches, smart routing approaches with real time data access, GIS-based routing approaches, and workload balancing.

### **2.2. Routing Problems & Models**

Routing problems can be classified into 2 main types of problems, the edge routing problem, and node routing problem. In node routing problems, there exist several points that must be visited, and the optimal visiting order must be found to minimize the travelling distance between points. In edge routing problems, there exists several edges that must be traversed and the optimal traversal order must be found to minimize the travelling distance between edges [19]. Due to the differences in the problem formulation, algorithms cannot be interchanged between edge routing and node routing problems [19].

Node routing problems are arguably easier to formulate, thus methods of converting edge routing problems to node routing problems have been proposed [20]. In this chapter, both edge routing and node routing applications will be explored.

### **2.2.1. Node Routing Problems**

The simplest version of the node routing problem is the TSP [21]. The TSP can be solved using a number of combinatorial optimization algorithms such as ACO, Simulated Annealing (SA), and the GA [22]. Expanding on the TSP, the capacitated vehicle routing problem (CVRP) is the most studied version of the node routing problem in which each node has a demand. Additionally, there may exist a fleet of homogeneous vehicles in the CVRP [23]. The CVRP is ideal for waste bin collection in the sense that each collection bin can be modelled as a node on a graph with a specific demand, and each edge traversed to get to the bin has a cost proportional to the distance travelled.

There also exists several other variations of the CVRP that are used within the waste collection route optimization scope. The multi-depot vehicle routing problem (MDVRP) in which multiple depots serve as candidate depot locations to start and end trips. Wang et al. solved the MDVRP using particle swarm optimization (PSO) [24], Kyung Hwan et al. and Mingozzi used exact methods to solve the MDVRP [25, 26], and Polacek et al. used a variable neighborhood search (VNS) to solve the MDVRP [27].

### **2.2.2. Edge Routing Problems**

The simplest version of the edge routing problem is the Chinese postman problem (CPP) [28]. There is an exact algorithm to solve the CPP as it is just an Eulerian tour that visits all edges at least once. As the CPP is designed for undirected graphs, it is not ideal to use this approach for routing within road networks that contain 1-way roads, in these cases the directed Chinese postman problem (DCPP) may be used [29]. However, many more useful versions of CPP exist [30], such as the directed-rural Chinese postman problem (DRCPP) [31] and the windy-rural Chinese postman problem (WRCPP) [32].

In the DRCPP, only a subset of edges in a directed graph needs to be serviced, and in the WRCPP, the subset of edges that need to be serviced have a demand that is dependent on the direction of travel. The DRCPP can directly translate to curbside waste collection and street-sweeping as specific routes are a subset of edges within the city's directed road network. Additionally, the WRCPP directly translates to curbside waste collection. Since the waste collection vehicles collect on the right-hand side, there may be more pickup points on one side of the road than the other, thus creating more (or less) of a demand depending on the direction of travel as seen in Figure 2-1.

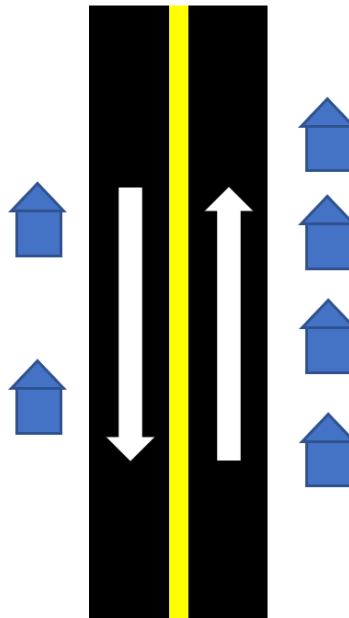


Figure 2-1 Directional Demand for the Curbside Waste Collection

Through transformative approaches proposed by Pearn et al., any form of the CPP can be formulated as the generalized travelling salesman problem (GTSP) [20], in which the nearest neighbor (NN), repetitive nearest neighbor (RNN), improved nearest neighbor

(INN), repetitive improved nearest neighbor (RINN), or loneliest neighbor (NLN) can be used to approximately solve the problem [30].

Additionally, any version of CPP that considers a demand on its edges can be considered as a CARP. There are several studies that make use of the CARP model in waste collection literature. Lacomme et al. model the CARP as a version of the TSP with starting and ending locations at either end of a serviceable road, thus making use of the GA as the appropriate solver [33]. Tirkolaee et al. developed a novel mathematical model for the periodic capacitated arc routing problem (PCARP) that was solved using SA [34]. Tirkolaee et al. optimized the number of waste collection vehicles and the route for each collection vehicle using their proposed model. In a separate article, Tirkolaee et al. used a multi-objective invasive weed optimization algorithm to solve for a multi-trip PCARP for the waste collection problem [35]. Mourão et al. developed a novel heuristic to solve the CARP in Lisbon [36]. They allowed a vehicle to make multiple trips to the dump to dispose of waste, and the depot location is different from the dump location [36]. Their model made use of the single-vehicle multi-trip principle in which the algorithm generates optimal solutions until a feasible solution is generated.

In some cases, municipalities separate different types of waste that may undergo different treatments at the disposal facility. For example, household waste may be disposed in a landfill, and organics waste may be used as compost for crops and cycled back into the environment. In these cases, municipalities may make use of the multi-compartment collection vehicle, as explored by Mofid-Nakhaee et al. [37]. They modelled their problem as a multi-compartment capacitated arc routing problem with intermediate facilities (MCCARPIF) in which 2 algorithms are developed to solve it; namely the adaptive large

neighborhood search algorithm (ALNS), and hybrid ALNS with whale optimization [37]. The purpose of this research was to confirm that using multi-compartment vehicles was beneficial compared to single-compartment vehicles making separate trips for different types of waste, and their case study in a district in Tehran proved this.

The location-arc routing problem (LARP) is also considered to be under the same umbrella as CARP. LARPs bridge the gap between facility location and route generation and distribution [38], and are especially useful in the case of developing cities without any facilities. As the LARP is capacitated, it is especially ideal for applications such as waste collection, street-sweeping, postal delivery, and road maintenance [39]. A good example of a LARP was presented by Yang, in which TS and the Augment-Merge heuristic approaches were combined to solve a LARP [40].

Edge routing optimization is also seen in several street-sweeping applications in literature. Bodin et al. developed the primitive “route first-cluster second” and “cluster first-route second” approaches for mechanical street-sweeping route optimization [41]. The “route first-cluster second” approach creates a giant tour of the road network that is divided into several feasible routes afterwards, and the “cluster first-route second” approach partitions the road network into several smaller areas in which the algorithm will generate tours in. Eglese et al. modelled their rural street-sweeping problem as the CCPP in which the deadhead travel is to be minimized [42]. Eglese et al. highlighted the importance of different road classes that require different frequencies of sweeping, something that is also seen in the street-sweeping problem in The City of Oshawa. Blazquez et al. proposed a unique method of graph construction for the street-sweeping optimization problem in which parallel edges are added to represent multiple passes needed to sweep roads with curbs on

each side of the road, even for 1-way and 2-way roads [43]. Blazquez et al. then used TSP to solve the street-sweeping routing problem [43]. This approach effectively models the real-world street-sweeping operations where multiple passes are needed to sweep against the curbs along the edge of the road. A physical representation of this can be seen in Figure 2-2.

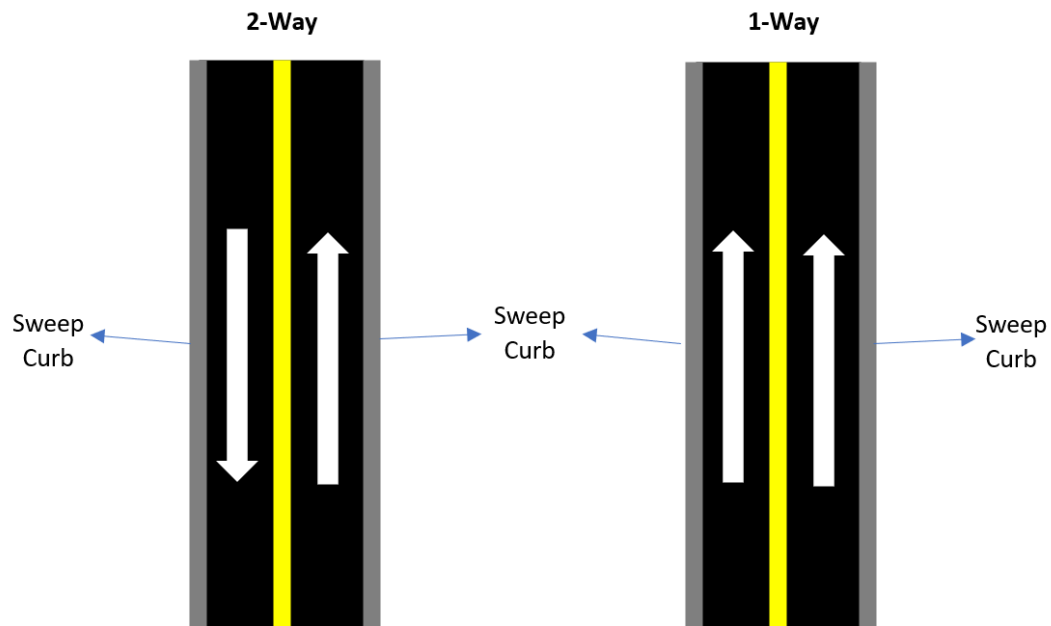


Figure 2-2 Multiple Passes to Sweep Curbs on Either Side of the Road

Outside the scope of waste collection and street-sweeping, edge routing is commonly used in snowplow route optimization problems. Rasul et al. were able to create an initial solution for the municipal snowplow route optimization problem using CPP, which was then improved using TS [44]. Similarly, Xu et al. used TS to improve the initial solution found by their k-trucks plowing algorithm [45]. In both cases mentioned, cycle permutation was used as the local search scheme to explore neighboring solutions [46].

### **2.3. Clustering & Zoning Approaches**

A majority of research in routing applications focuses on the optimization of routes themselves [47]. However, it is common for municipalities to divide a large area into several smaller areas for better productivity, dispatching, and organization of the operations team. This occurrence is seen in several real-world applications, such as waste collection, street-sweeping, and snowplowing. Little research has been done in this area, but the few related instances will be explored in this section.

Xin et al. highlighted the importance of a well-designed waste management system within rapidly expanding urban areas [48]. When a population grows, some waste facilities become redundant, and thus relocation is needed. Using Voronoi polygons and WK-Means clustering methods, Xin et al. were able to identify redundant waste collection centers, and appropriately relocate them [48]. Al-Refaie et al. conducted similar research where the location of several communal waste bins needs to be optimized [49]. However, Al-Refaie et al. not only optimized the collection-bin-to-depot cost, they also considered maximizing the demand collected by the waste collection vehicles to ensure the vehicles had adequate workload [49]. Wei et al. proposed an improved hierarchical agglomerative clustering (IHAC) algorithm that clusters collection points together such that each cluster is expected to fill the waste collection vehicle, then a garbage collection path planning (GCPP) algorithm is used to generate the servicing path that visits all collection points within the respective clusters [50]. Pop et al. developed a 2-level approach to solve the clustered vehicle routing problem (CluVRP) where an upper-level subproblem uses a GA to generate clusters of customers, and a lower-level subproblem aims to find the optimal route within each cluster using the TSP model [51].



Asides from waste management applications, Zheng et al. developed a novel grid based K-Means clustering method for traffic zone division of a city [52]. By carefully analyzing global positioning system (GPS) data from the city of Nanjing, the authors were able to effectively divide the city into several different traffic zones to accurately predict the flow of traffic. Zheng et al. overlaid the clustered GPS data on a grid map of Nanjing, and the grids were able to define hard boundaries for the clusters that a collection of points simply could not achieve [52]. Soor et al. modified K-Means to include connectivity constraints through a repeated application of the watershed transform [53]. With the addition of the watershed transformations, the developed clustering approach was verified to be ideal for road network applications. A case study shown using Mumbai's road network showed ideal locations for 16 emergency stations, as well as the relationship between cost and number of emergency stations, which decreased with the number of emergency stations (as expected) [53].

#### **2.4. Smart Routing Approaches**

As smart-technology becomes the norm, some researchers have explored the possibility of integrating smart-technology into routing scenarios via Internet of Things (IoT) applications. Waste management systems can benefit from smart-technology by providing intelligence to waste bins using IoT sensors [54]. By monitoring the fill level status of the waste collection bins in the network, empty bins can be ignored in the route optimization algorithms, thus reducing the deadhead travel.

In recent years, IoT applications in waste management systems are becoming increasingly popular amongst researchers [55]–[61]. However, the main idea across all the works is almost identical. Each collection bin is equipped with a sensor that is capable of

transmitting information through cloud applications regarding the fill level of the waste, and an optimization algorithm is used to generate the service route collecting from the appropriate bins. Typically speaking, the mentioned IoT applications are ideal for node routing problems since the bins would be located at points in the road network. An example of the smart-bin approach can be seen in Figure 2-3.

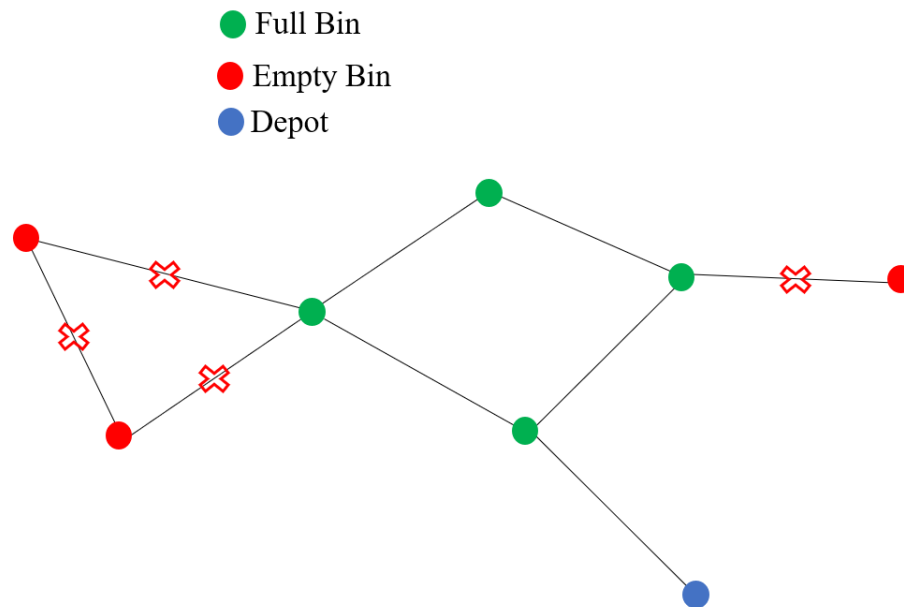


Figure 2-3 IoT Waste Bin Fill Level Example

## 2.5. GIS-Based Approaches

GIS data contains spatial data and numerical information in a single dataset, making it the ideal package for road network representation. Additionally, GIS applications (such as ArcGIS [62] and QGIS [3]) have built in tools that are capable of simple routing applications. For example, O'Connor [63], Malakahmad et al. [64], Abdallah et al. [65], Kallel et al. [66], and Chalkias [67] used The ArcGIS Network Analyst solve tool for their solid waste collection problems. The Network Analyst toolbox is used to maintain network

datasets, and optimize simple routing problems like the VRP and find the nearest facility [68]. Additionally, Apaydin et al. used the Route View Pro™ software integrated with GIS data (containing numerical pathways, demographic distribution, container distribution, and solid waste production) as the optimization tool [69].

Due to the limited scope of commercially available software and toolboxes, some researchers may choose to develop their own routing algorithms and still use the GIS data as it contains valuable information. This is the case with Vu et al., where GIS data was combined with an artificial neural network (ANN) to predict the waste generation rate of garbage and recycling streams in the future [70]. Vu et al. were able to recommend changes to the waste collection routes in Austin, Texas that coincide with the expected increase in waste generation rate predicted by the ANN [70]. Similarly, Ghose et al. determined the minimum cost path of the waste collection model using GIS road network data and bin locations for the Asansol Municipality Corporation of West Bengal State, India [71].

## **2.6. Workload Balancing**

In addition to generating optimal routes, the satisfaction of the operations team should be met. One way to achieve employee satisfaction and reduce fatigue is through a proper workload distribution [72]. Most of the research in workload balancing focuses on assembly lines in factories. However, a few instances that do consider the workload balance of waste collection routes will be discussed in this section.

Rabbani et al. formulated the workload balance of a fleet of collection vehicles by calculating the deviation of each vehicle's travel distance from the fleets total travel distance [73]. Jorge et al. developed the simulated annealing and neighborhood search (SANS) algorithm with workload concerns to penalize imbalanced solutions that do not

comply with the maximum shift duration [74]. Qiao et al. explored the workload balance of the disposal facilities by analyzing the expected workload for each day of the week from a fleet of vehicles optimized using the CVRP model [75]. Linfati et al. balanced the workload for each collection day by analyzing the average number of customers and daily containers delivered for the waste collection problem [76]. Ideally there should be a similar number of customers and containers delivered each day, which was noticed from their results [76]. Shih et al. quantified the workload distribution of the waste collection problem by calculating the difference between the maximum and minimum daily travel distance, where a lower difference correlates to a more balanced result [77].

Some approaches used in assembly line workload distribution can be transferable to route optimization. For example, Qian, Kim et al., and Zaplana et al. all used some form of the GA to balance workloads amongst line workers in assembly plants [78]–[80]. The idea of balancing the workload for line workers directly translates the routing problem with multiple routes or areas. Due to the flexible nature of the GA, some simple modifications can be made to make it suitable for the application of route balancing.

## **2.7. Conclusions**

Through the discussed literature review, state-of-the-art methods of route optimization have been explored in detail. Although some methods discussed were not directly applied to waste collection or street-sweeping route optimization, the respective applications shared enough similarity that the algorithms and methods can be interchanged with little modifications. However, there still exists gaps within the discussed literature for new heuristic approaches for the unique routing problems that will be explored in this thesis.

As seen in Section 2.2, several different models exist for the edge routing and node routing problems. The models discussed in the literature are more generalized in the sense that they work for different types of routing problems. Despite this, different problems require the use of different models. Routing problems that have not been seen in literature require new (and potentially complex) models to be developed. This paper aims to create a new model for the CARP with many constraints and service roads that have several demands (e.g., street dust/debris and water dispersion from the sweeping trucks to aid in collection), as well as a novel u-turn removal algorithm to make optimal routes feasible with respect to traffic operations.

As seen in Section 2.3, several different clustering and zoning approaches have been explored that are used to group road networks. In most cases, these methods are used on smaller areas rather than city-wide. From the conducted literature review, there were no existing studies that used a cluster-based approach on a city-wide scale for operational area division (waste collection areas, street-sweeping areas, snowplow areas, etc.), so a novel 2-stage clustering approach (Static and Dynamic clustering) was developed to fill this gap found in routing operations literature.

As seen in Section 2.5, different GIS-based approaches to the route optimization problem were discussed. Since the scope of this research was to develop the route optimization algorithms, GIS-based solvers (ArcGIS Network Analyst) were not used. However, a comprehensive set of GIS data was supplied by the City of Oshawa to model the complex road network. Having access to such a detailed library of information allows for more realistic simulation results. Looking at existing research, little to no papers consider speed limits of different roads, and instead assume a constant travel speed in their

optimization model. In the real world, vehicles must follow the speed limit of the respective roads within the network, and different roads typically have different speed limits. The following research makes use of the detailed GIS data supplied by The City of Oshawa to dynamically change the speed of the service vehicle in the optimization model with respect to the current roads speed limit.

As seen in Section 2.6, the workload balancing approaches in existing research was discussed. Several cases were examined that are applicable to the waste collection route optimization, however they were very primitive. For example, some studies quantified the workload balance strictly based on the distance travelled. In any type of route optimization problem, quantifying the workload balance by only considering one variable may be biased, thus more variables should be considered. The following research aims to fill this gap by defining a normalized objective function that consists of multiple variables to be considered in the workload balance amongst the operations team. Additionally, a problem specific method of pairing waste collection areas for each respective day of the week is proposed that makes use of the multi-variable workload balance objective function value.

The following two chapters will explore the methodology behind the developed heuristic approaches used in each respective application for The City of Oshawa. Each chapter contains several components specifically developed to bridge the gap in literature highlighted in this chapter.

## **Chapter 3. Waste Collection Route Optimization**

### **3.1. Introduction**

Research shows that waste collection costs can range from 50-90% of the municipal solid waste budget [81], thus it is critical that the waste collection process is done in an efficient manner. For optimal results, the cost, environmental impact, collection and travel time, and social aspects of the waste collection process should all be considered [75].

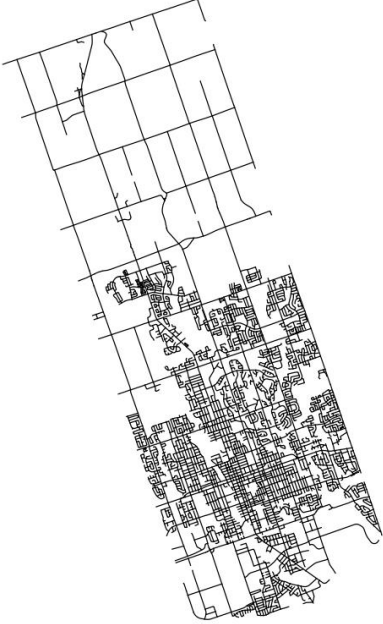
This chapter will cover the waste collection route optimization for The City of Oshawa in its entirety. Although comprehensive GIS data was supplied by the city, additional datasets and methods of data preparation were needed to complete the study, which will also be discussed. With a complete dataset, waste collection route optimization algorithms were developed, and the working principle and methods used to create the proposed algorithm will be explored. Using the proposed route optimization algorithm, realistic route simulations were used to calculate several statistics that can be combined into a single objective function to represent the fitness of each respective route. The dimensionless objective function values of all routes in each collection area were summed together to represent the fitness of each collection area, which was used to properly distribute the workload balance across each day of the working week through the proposed workload balance algorithm. Additionally, it was hypothesized that the existing waste collection areas can be improved with respect to the workload balance, so a novel 2-stage clustering algorithm was developed to generate (and optimize) waste collection areas without any previous knowledge of the existing configuration. Finally, a complete analysis of the current routing configuration was compared to the routing configuration generated by the proposed 2-stage clustering approach.

### 3.2. Data Preparation

#### 3.2.1. Supplied & Downloaded Data

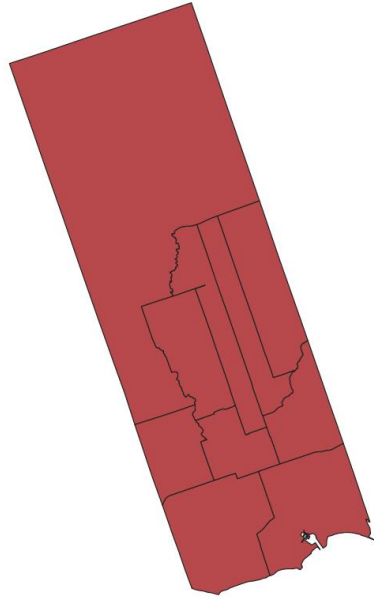
As mentioned earlier, a comprehensive set of GIS data was supplied by the city to conduct the following research. Specifically, several shapefiles were supplied that can be viewed and manipulated through GIS applications such as QGIS [3]. The supplied shapefiles include the “TeachingCity\_RouteOptimization CityStreetNetwork,” “TeachingCity\_RouteOptimization CollectionAreas,” “TeachingCity\_RouteOptimization WasteView,” and “route\_blocks.” Table 3-1 depicts a visual representation of the supplied shapefiles as well as a brief description of the useful information contained.

Table 3-1 Supplied Shapefiles and Their Useful Information

Shapefile Name and Visual Representation	Brief Description and Useful Information Contained
<p data-bbox="342 1020 794 1087">“TeachingCity_RouteOptimization CityStreetNetwork”</p> 	<p data-bbox="867 1236 1414 1377">A multiline type shapefile representing the road network within The City of Oshawa. Each multiline segment contains the following useful information.</p> <ul data-bbox="915 1423 1357 1646" style="list-style-type: none"> <li>• Unique Road ID</li> <li>• Street Name</li> <li>• Speed Limit</li> <li>• Traffic Operation (1-way or 2-way)</li> <li>• Street Length</li> </ul>

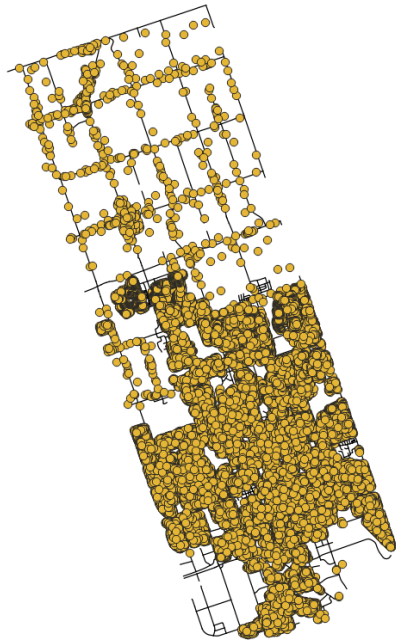


“TeachingCity\_RouteOptimization  
CollectionAreas”



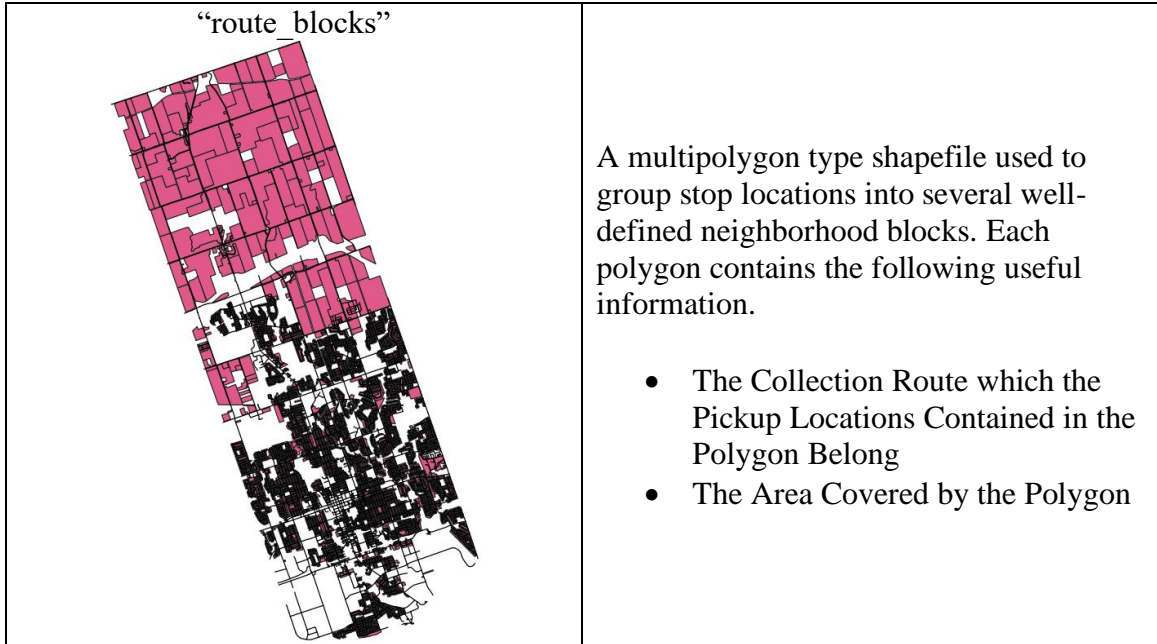
A multipolygon type shapefile defining the boundaries of the existing 10 collection areas within The City of Oshawa. Each area contains 11 servicing routes. This information is also seen in Figure 1-1.

“TeachingCity\_RouteOptimization  
WasteView”



A multipoint type shapefile depicting all stop locations for curbside waste collection operations. Each point contains the following useful information.

- The Number of Dwellings
- The Adjacent Street Name



In addition to this, a complete set of 111 route PDF maps used by the waste collection team were supplied showing the existing collection routes and pickup locations. An example of an existing route PDF map can be seen in Figure 3-1, where the orange highlighted locations show which side of the road collection is to be conducted.

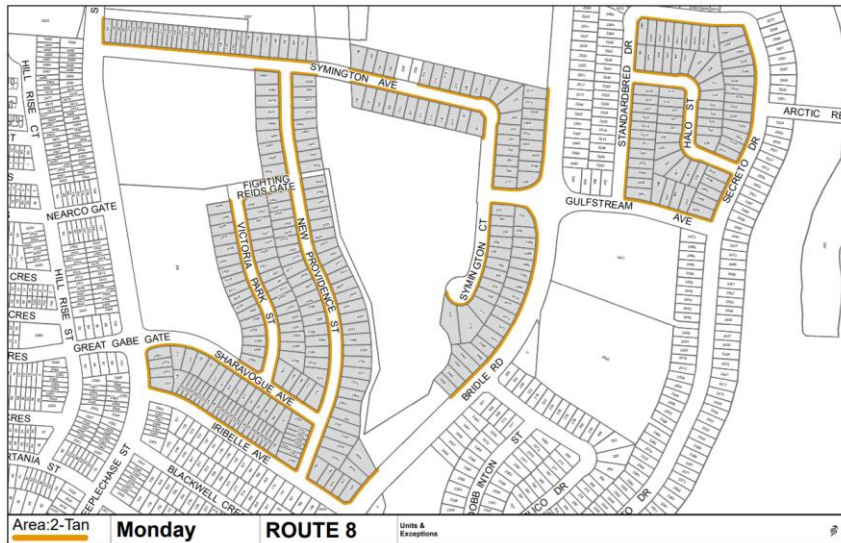


Figure 3-1 Monday Yellow - Route 8 PDF Map

As seen in Figure 3-1, the sequence of which to travel through the road network to collect the curbside waste is missing. This means that each week the collection sequence may be different from driver-to-driver, and thus a method of optimizing and standardizing the collection sequence can be beneficial from an operations point of view.

There was still information missing regarding the connectivity of the road network shapefile. Referencing graph theory, a graph consists of several nodes connected with edges, mathematically modelled as  $G = (V, E)$ , where  $V = \{v_0, v_1 \dots, v_n\}$  is the set of nodes and  $E = \{(v_i, v_j) | v_i, v_j \in V\}$  is the set of directed edges. In the “TeachingCity\_RouteOptimization CityStreetNetwork” shapefile, such intersection connectivity was missing to be modelled as a graph. So, separate datasets were used which can be found on Durham Region’s open source GIS database [82]. Specifically, a multipoint shapefile representing the intersections in the road network, and another multiline shapefile (like the “TeachingCity\_RouteOptimization CityStreetNetwork”) were downloaded. In the new road network shapefile, each road has an “F\_NODE” and “T\_NODE” attribute (as seen in Figure 3-2) that references the IDs of the intersections found at either end of the road. In the intersection shapefile, each point has a unique ID and pair of coordinates that localize the intersection.

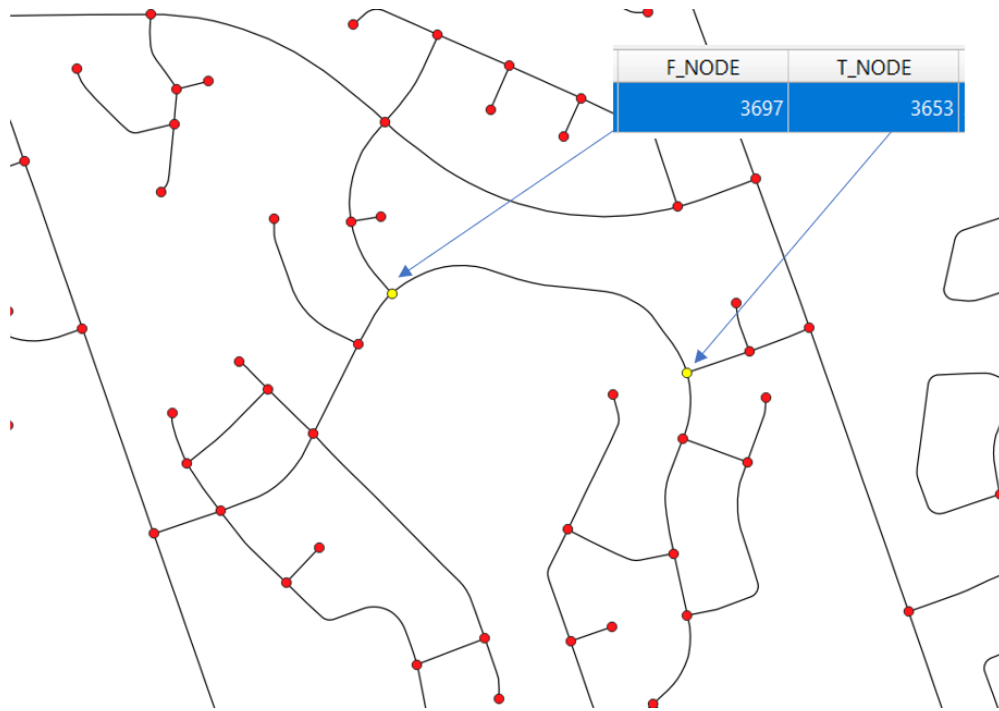


Figure 3-2 Road Connectivity Information - Downloaded Open-Source Dataset

Each of the mentioned datasets contains valuable information used to formulate the routing problem, so a hybrid dataset needed to be created. Using the “Join Attributes by Location” tool in QGIS [3], the useful attributes found in the “TeachingCity\_RouteOptimization CityStreetNetwork” discussed in Table 3-1 were added to the downloaded road network, thus creating a single multiline type shapefile for the road network with a complete set of useful attributes.

### 3.2.2. Route Shapefiles

To optimize the routes, they need to be isolated from the whole road network. To do so, the route PDF maps were used as a reference and the corresponding intersections and road segments were manually selected from the hybrid dataset. Additional roads were added to connect isolated parts of the route or to provide safe turnaround locations to

minimize u-turns in the route. At this point, each road represents a single direction of travel, so the route shapefile was copied and reversed, thus creating a single shapefile with bi-directional roads. There exist several 1-way roads in the City of Oshawa, in these cases, the roads are not reversed to preserve the intended travel direction. Also, an additional attribute is added at this point to distinguish between serviceable roads with stop locations, and deadhead roads. This is a simple binary variable, 1 for service roads, and 0 for deadhead roads. A side-by-side comparison between a route PDF and the corresponding route shapefile can be seen in Figure 3-3.

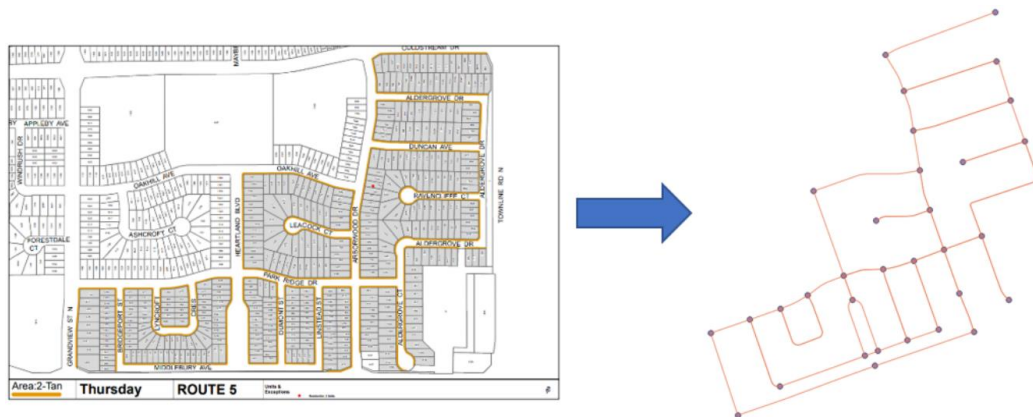


Figure 3-3 Side-By-Side Comparison of Route PDF Map and Shapefile

The stop locations and dwelling unit information is also needed in the route simulations. Using the “route\_blocks” shapefile, the corresponding stop locations were filtered for each respective route. A simple algorithm was developed that would use the polygons defined in the “route\_blocks” shapefile to filter out stop locations from the “TeachingCity\_RouteOptimization WasteView” shapefile for each route. The algorithm works by identifying which stop locations from the “TeachingCity\_RouteOptimization

WasteView” are contained by the boundary of the corresponding “route\_block” polygon. The stop location filtering algorithm can be visualized in Figure 3-4.

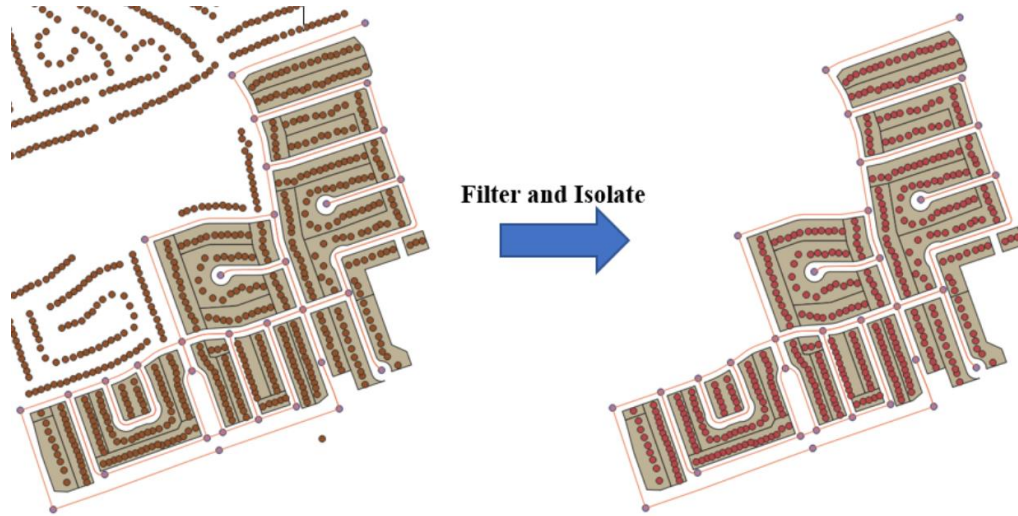


Figure 3-4 "route\_block" Stop Location Filtering Algorithm – Visualized

### 3.3. Waste Collection Route Optimization Algorithm

The objective of the proposed route optimization algorithm is to generate the shortest feasible Eulerian tour that services all required edges in each collection route. To do so, a combination of Dijkstra’s [7] algorithm and Hierholzer’s [10] algorithm is used. The proposed heuristic will only work if all nodes are reachable (strongly connected), thus proper route preparations are needed, as discussed in Section 3.2.2.

Eulerian tours are only possible if each node has as many incoming edges as outgoing edges, mathematically formulated as  $\sum_{v \in V} deg^+(v) - \sum_{v \in V} deg^-(v) = 0$ . If this condition is not met, a node is called unbalanced. So, each node in the graph is checked for this condition. Depending on the route, this condition may not always be true, which is where Dijkstra’s algorithm is used. In the case where nodes are unbalanced, the nodes with

too many incoming edges are denoted as Dijkstra's starting nodes, and nodes with too many outgoing edges are denoted as Dijkstra's ending nodes. A brute-force method was used to find the optimal pairing of Dijkstra's starting and ending nodes such that the minimum distance is needed to connect the starting and ending nodes together to create a balanced graph. In larger instances, a combinatorial optimization algorithm (like GA) can be applied to find the optimal pairing, but the individual routes are relatively small, so a brute force approach was used to identify all permutations of starting and ending nodes in a short amount of time. All edges used to balance the graph will be added back into the graph as deadhead edges, thus creating several instances of parallel edges. The significance of finding the optimal pairing of starting and ending nodes can be seen in Figure 3-5, as some pairs may add significantly more redundant travelling than others.

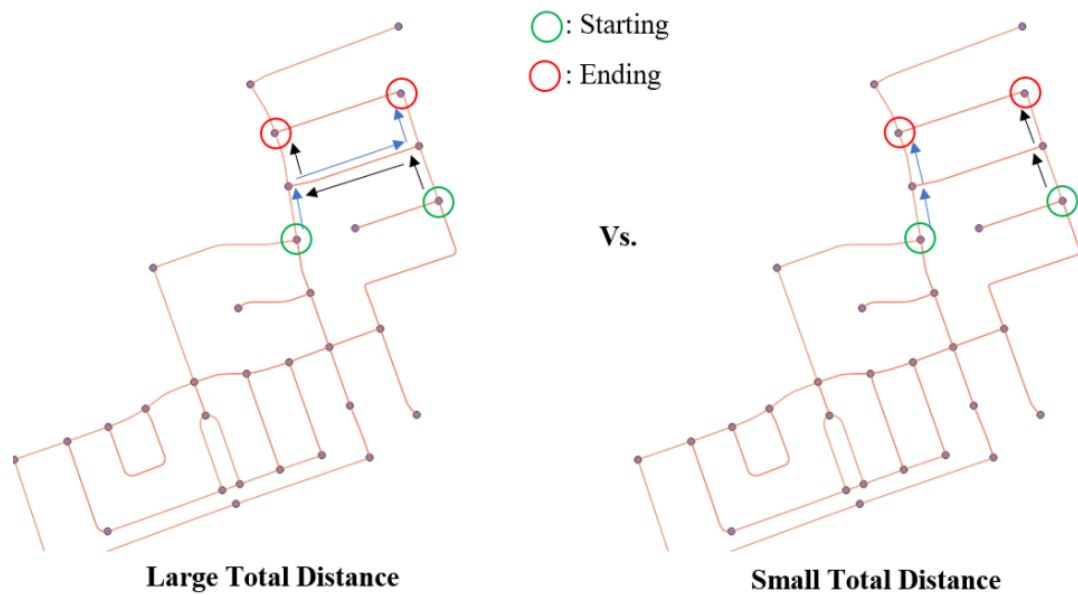


Figure 3-5 Importance of Optimal Starting and Ending Node Connections

With a balanced graph, an Eulerian tour can be generated using Hierholzer's algorithm. Hierholzer's algorithm was used instead of CPP due to its ability to work on directed

graphs, which is an important aspect of waste collection since the direction of travel is important for curbside waste collection (vehicles collect on the right-hand side). Additionally, the direction of travel should be consistent with the respective traffic operation of the road (1-way or 2-way). Hierholzer's algorithm works by choosing a start node and randomly selecting an outgoing edge from the current node, but this process may not be ideal for generating feasible routes with respect to generating unsafe u-turns at intersections where u-turns are not allowed, such as the one seen in Figure 3-6 (b). As Hierholzer's algorithm selects random outgoing edges, in the case of parallel edges, the previous node may be selected to travel to the next, thus creating a u-turn. A simple, yet effective modification is made to Hierholzer's algorithm by randomly selecting the next outgoing edge that *is not* the previous edge. In some cases, selecting the previous edge may be the only option (in the case of dead-ends and courts, as seen in Figure 3-6 (a)), but the cases with multiple outgoing edges benefit from this simple modification.

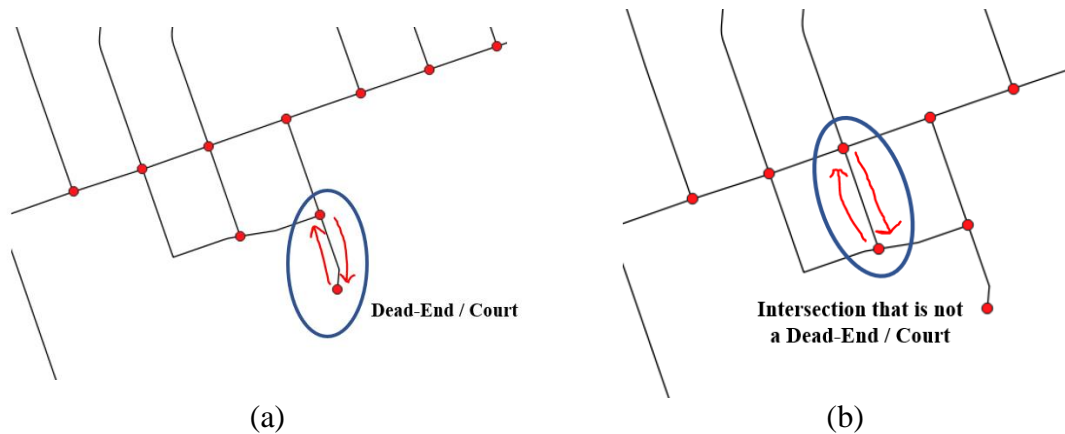


Figure 3-6 Example of a Safe U-Turn Location (a), and Unsafe U-turn Location (b)

The starting node also has an influence on the feasibility of a route, so all nodes in the input graph were used as candidate starting locations, the optimal starting location is the



one that yields the least u-turns. The pseudocode for the proposed waste collection route optimization algorithm can be seen as follows.

---

**Algorithm 1** Waste Collection Route Optimization Algorithm

---

**Input:** Route shapefile

**Output:** Optimal feasible Eulerian tour

---

**identify** unbalanced nodes in the input graph

**for** all pairs of Dijkstra's starting and ending nodes **do**

**calculate** the total distance needed to balance the graph

**update** the optimal pairing

**add** edges used in the optimal pair back into the original graph

**for** each node in the graph **do**

**run** modified Hierholzer's algorithm

**count** the number of u-turns

**update** the best solution

**return** optimal feasible Eulerian tour

---

### 3.4. Dimensionless Objective Function

#### 3.4.1. Waste Collection Route Objective Function Values

To quantify the fitness of the optimal tours generated by the waste collection route optimization algorithm, a dimensionless objective function was used that is composed of several statistics. Specifically, the travel distance, consumed fuel, collection time, and travel time will all be considered. The distance was simply calculated by summing the traversed edges in a route. Referencing the operation principle of The City of Oshawa's waste collection program, the collection week alternates between Week 1 and Week 2 where either garbage and organics, or only organics are collected. As a result, the collection time varies between Week 1 and Week 2 collection, thus requiring 2 separate fitness values. It is assumed that 10 s per dwelling unit are needed to collect for only organics, and 30 s

per dwelling unit are needed to collect garbage and organics. This assumption was supplied by the city as it is common for a 3:1 ratio of collection time between garbage and organics and only organics. The fuel consumption rate was also supplied by the city as 0.705 L/km for deadhead travelling, and 1.523 L/km for servicing. This makes sense because servicing requires continuous stop-and-go motion, thus a lot of acceleration and deceleration typically burns more fuel than travelling at a constant speed when the vehicle is not servicing. The travel time depends on the respective road's speed limit and the service status. If a road is simply being used as deadhead travel, the collection vehicle will travel 90% of the speed limit (because it is a considerably large truck), and the servicing collection speed is a function of the dwelling unit density in the route. It was assumed that the minimum average travel speed for a servicing road is 15 km/hr on the densest route, and 90% the speed limit on the most sparsely populated route. The formula relating the route density to servicing travel speed can be seen in Eq. (3-1), where  $s_c$  is the collection speed,  $d_r$  is the route density,  $s_l$  is the speed limit,  $s_m$  is the minimum assumed speed (15 km/hr),  $\min(\{d_R\})$  is the minimum calculated route density (13.15 DUs/km<sup>2</sup>), and  $\max(\{d_R\})$  is the maximum route density (5313.15 DUs/km<sup>2</sup>). The route density was calculated by simply dividing the total number of dwelling units by the area covered for each respective route. The collection speed is a negative linear relationship where the densest routes have the lowest average servicing speed due to more frequent stops.

$$s_c(d_r) = \frac{0.9s_l - s_m}{\min(\{d_R\}) - \max(\{d_R\})} * d_r + 0.9s_l \quad (3-1)$$

As mentioned earlier, the servicing roads and deadhead roads have different speed limits, this is also the case with travelling on a road that has already been serviced. The binary service variable of each edge needs to constantly be updated so the proper speed

limit and fuel consumption rate can be applied. If a road has already been serviced, it should not be serviced again, thus any additional travelling will be considered deadhead. Table 3-2 summarizes the different servicing conditions that are seen in a simulation, and the corresponding variables used.

Table 3-2 Service Condition and Corresponding Simulation Variables

Condition	Service Variable	Travel Speed	Fuel Consumption Coefficient
A road is serviceable and has not been serviced	1	Linear relationship dependent on route density (3-1)	1.523 L/km
A road is serviceable, but has already been serviced	0	$0.9 * s_l$	0.705 L/km
A road that is only included as deadhead to balance or connect the road network	0	$0.9 * s_l$	0.705 L/km

The dimensionless objective function to represent the fitness of each route can be seen in Eq. (3-2), where  $d_{ij}$  is the distance along edge  $(v_j, v_j)$ ,  $c_{ijf}d_{ij}$  is the fuel burnt on edge  $(v_j, v_j)$ ,  $d_{ij}/s_{ijs}$  is the travel time along edge  $(v_j, v_j)$ , and  $du_t$  is the time to collected from each dwelling unit  $t$ . As shown in Eq. (3-2), each component is normalized by dividing by the maximum of each respective component, thus allowing an unbiased comparison from route to route. It is also worth mentioning that all route simulations need to be completed to calculate the normalized objective function value because the maximum of each component is included in the objective function. Table 3-3 summarizes all the variables used to formulate the objective function value.

$$\begin{aligned}
f_1 = & \sum_{i=0}^n \sum_{j=0}^n \frac{d_{ij}}{\max(\{d_{IJ}\})} \\
& + \sum_{i=0}^n \sum_{j=0}^n \sum_{f=0}^F \frac{c_{ijf} d_{ij}}{\max(\{c_{IJF} d_{IJ}\})} \\
& + \sum_{i=0}^n \sum_{j=0}^n \sum_{s=0}^S \frac{(d_{ij}/s_{ijs})}{\max(\{d_{IJS} s_{IJS}\})} \\
& + \sum_{t=1}^T \frac{du_t}{\max(du_T)}
\end{aligned}$$

(3-2)

Table 3-3 Objective Function and Simulation Parameters

Parameter	Meaning	Value
$f_1$	Dimensionless objective function value	Calculated in Eq. (3-2)
$d_{ij}$	Road distance	Dependent on road (km)
$c_{ijf}$	Fuel consumption rate	0.705 L/km for deadhead 1.523 L/km for servicing
$s_{ijs}$	Travel speed	0.9 speed limit for deadhead Calculated in Eq. (3-1) for servicing (km/hr)
$du_t$	Collection time per dwelling unit	30 s for garbage and organics 10 s for only organics

### 3.4.2. Waste Collection Area Objective Function Values

As discussed earlier, each of the 10 collection areas consist of 11 individual routes. The methods in Section 3.4.1 were used to calculate the 2 objective function values (Week 1 and Week 2 collection) for all routes. The objective function values were expanded to represent the fitness of the collection areas themselves by summing the objective function values for all routes in each respective area. Similarly, each area consists of 2 fitness values for each collection week. Since 2 areas are collected for each day of the working week, the

collection area objective function values will be used to provide insight as to how these areas can be better paired.

### **3.5. Workload Balance (Waste Collection Area Pairing) Algorithm**

Referencing Figure 1-1, each day of the week has a purple and a yellow area to collect waste in; from a spatial perspective, the areas seem to be unbalanced (Monday yellow is much larger than other areas). However, simply looking at the area coverage of the collection areas is not enough to ensure a proper workload balance is distributed across the days of the week as many other variables need to be considered. Using the collection area objective function values, a GA was developed to find the optimal pairing of collection areas for The City of Oshawa. A brute force method may be applied, but it would take significantly longer to optimize because of the many possible area combinations as seen in Figure 3-7.

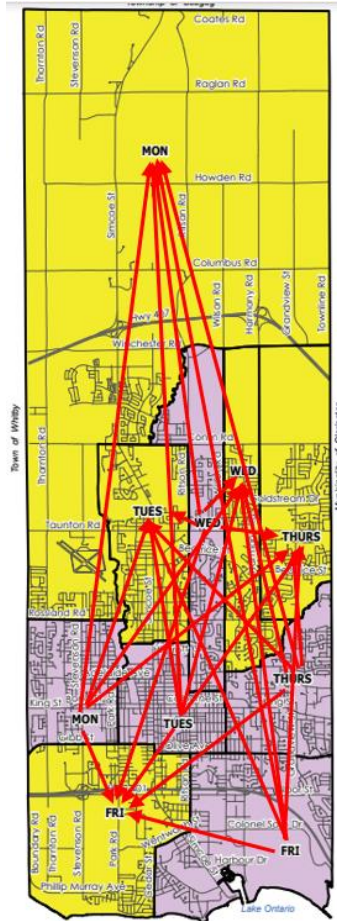


Figure 3-7 All Possible Collection Area Combinations for the Current Configuration

The proposed GA works very similarly to traditional GA with crossover and mutation operators, but some small modifications have been made to ensure that collection areas do not duplicate in the chromosome structure. In the traditional GA, the crossover operator works by selecting a random location to splice parent chromosomes together. However, if this was used for the collection area pairing problem, this would result in duplicated areas (visualized in Figure 3-8). The gene duplication issue is also seen in the traditional mutation operator as well.

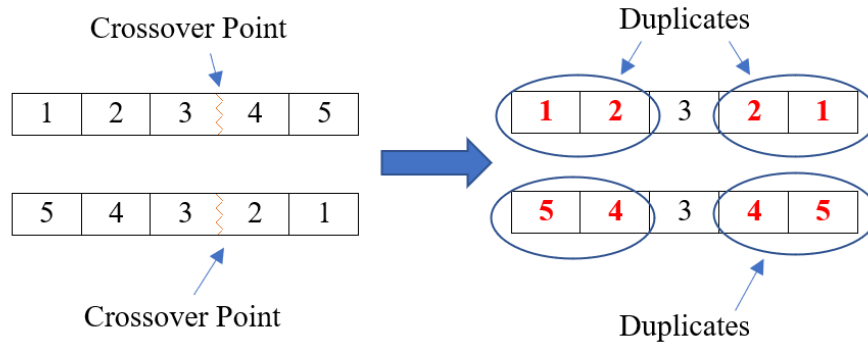


Figure 3-8 Duplicates in Chromosome Structure After Crossover

To combat the occurrence of duplication in the chromosomes, simple workarounds are used. In the case of the duplicates in crossover, one of the duplicated areas is randomly selected and removed from the chromosome structure, and one of the missing areas is inserted in the same location in the chromosome. This is repeated until all duplicated areas have been replaced.

When mutation happens in the original GA, a gene is randomly changed to another. In the case of the collection area problem, all collection areas are already in the chromosome structure, so simply changing the gene to another value is insufficient. Instead of changing genes in the mutation phase, a swapping operation was used [83]. If mutation should occur, the gene would simply be swapped with another, changing their location in the chromosome structure.

To calculate the fitness of the chromosome, an encoding method was applied. As each chromosome contains all collection areas in a unique sequence, every odd gene will be assigned Week 1 collection, and every even gene will be assigned Week 2 collection, thus making use of the 2 objective function values discussed in Section 3.4.2. The chromosome length was fixed at 10, and every 2 genes were paired together for each day of the week.

For example, the first 2 genes are assigned to Monday collection, where the first gene will be Week 1 collection, and the second gene will be Week 2 collection. The daily workload can be calculated by adding the objective function values for the 2 collection areas that make up each day of the week, resulting in 5 objective function values. The fitness of each chromosome was calculated as the standard deviation of the 5 objective function values for each day of the week, formulated in Eq. (3-3) where  $l$  is the workload of day  $i$ , and  $\mu_l$  is the average workload across all days  $D$ .

$$f_2 = \sqrt{\frac{1}{D} \sum_{i=1}^D (l_i - \mu_l)^2} \quad (3-3)$$

The workload balance algorithm can be seen as follows.

---

**Algorithm 2** Workload Balance Algorithm

---

**Input:** All collection areas Week 1 and Week 2 objective function value

**Output:** Optimal collection area pairs

---

**create** a population of random shuffled collection areas

**for** generation **in** generations **do**

*# Calculate fitness of all members*

**for** member **in** population **do**

**assign** Week 1 to even genes

**assign** Week 2 to odd genes

**sum** the objective function values for each pair

**calculate** the fitness

*# Create new population*

**add** the 2 best solutions to the new population

**while** the new population size  $\neq$  the old population size **do**

**roulette wheel** to select parents

**perform** crossover

**resolve** duplicate areas

**perform** swap mutation

**add** children **to** new population

**return** optimal collection area pairs

---



### 3.6. Waste Collection Area Generation Algorithms

It is speculated that the existing waste collection area boundaries for The City of Oshawa can be improved. To do so, a novel 2-stage clustering algorithm is developed that uses the W-K Means clustering algorithm and DE. To understand how the 2-stage cluster algorithm works, the operating principle of the WK-Means algorithm needs to be explored.

WK-Means is the weighted version K-Means, where the objective is to find the minimized cost partitioning of weighted points in a specified number of clusters [84]. With regards to the waste collection problem, the collection points would have a weight equivalent to the number of dwellings associated with the pickup location. In K-Means, the objective function is simply the minimum total distance of each point to its cluster center formulated in Eq. (3-4), where  $k$  is the number of clusters,  $n$  is the number of points to partition,  $x_{i,x}^{(j)}$  and  $x_{i,y}^{(j)}$  are the x and y locations of point  $x$  belonging to cluster  $j$ , and  $c_{j,x}$  and  $c_{j,y}$  are the x and y locations of cluster center  $j$ . In the case of WK-Means, each cluster has a weight proportional to the weight of points belonging to the respective cluster. The WK-Means algorithm can be formulated in Eq. (3-5) where  $w_j$  is the weight of cluster  $j$ ,  $\sum_{j=1}^k w_j = 1$ , and  $\beta$  is the cluster weight influence factor.

$$f_3 = \sum_{j=1}^k \sum_{i=1}^n \sqrt{\left(x_{i,x}^{(j)} - c_{j,x}\right)^2 + \left(x_{i,y}^{(j)} - c_{j,y}\right)^2} \quad (3-4)$$

$$f_4 = \sum_{j=1}^k \sum_{i=1}^n w_j^\beta \sqrt{\left(x_{i,x}^{(j)} - c_{j,x}\right)^2 + \left(x_{i,y}^{(j)} - c_{j,y}\right)^2} \quad (3-5)$$

Since the objective of the WK-Means algorithm is to find the minimum cost partitioning, it may not produce the best distribution of weights, which is why it was paired with advanced optimization methods.

### 3.6.1. GIS Data Preparation for the 2-Stage Clustering Algorithm

If the clustering was used on only the curbside collection locations, this may lead to some streets having houses that belong to several different collection areas. Ideally, all the houses on a single street should belong to the same collection area. Because of this, additional data preparation was needed on the GIS data.

All stop locations in the “TeachingCity\_RouteOptimization WasteView” shapefile contain an attribute that relates the stop location to the adjacent street. An algorithm was developed that iterates through all stop locations in the “TeachingCity\_RouteOptimization WasteView” shapefile and adds the corresponding pickup locations to the adjacent road segment with the same name. In doing so, a new attribute was created for each road segment that shows how many dwellings reside on each road, as seen in Figure 3-9.

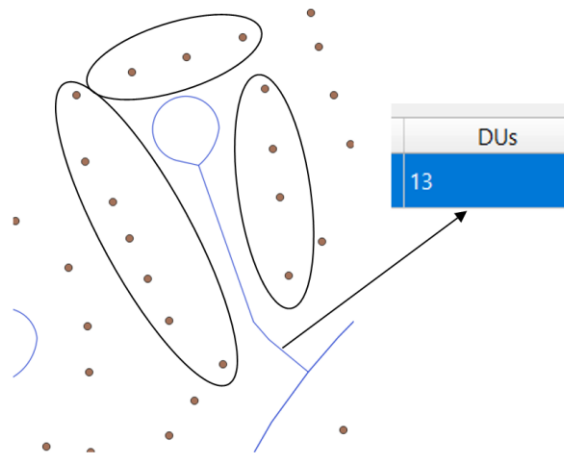


Figure 3-9 Joining Dwelling Counts to Road Segments

Since the clustering is going to be done on the roads instead of the collection locations, the roads needed to be converted into points. Each road segment is a multiline (multiple lines connected head-to-tail), so the points used to create the geometry can be

easily extracted. By taking the average  $x$  and  $y$  coordinate locations of the points formulating the multiline, the approximate road center location was found. Upon doing so, the roads have successfully been converted to weighted points that can be used for clustering.

### 3.6.2. Static Clustering

The first stage of the 2-stage clustering algorithm is the Static clustering stage. In this stage, a single  $\beta$  value will be optimized to yield the cluster configuration with the best distribution of weights. The objective function can be formulated in Eqs. (3-6) and (3-7), where  $w_j$  is the weight of cluster  $j$ ,  $\mu_w$  is the average cluster weight.

$$f_5 = \sqrt{\frac{1}{k} \sum_{j=1}^k (w_j - \mu_w)^2} \quad (3-6)$$

$$0 \leq \beta \leq 1 \quad (3-7)$$

DE was used to tune a single  $\beta$  value in the Static clustering process. For each  $\beta$  value, the WK-Means clustering algorithm was computed several times until the cluster centers converge or the maximum clustering iteration is reached. WK-Means involves stochastic processes, so multiple runs of WK-Means were executed for a single  $\beta$  value to ensure an ideal partition is found. Since  $\beta$  can be within the range of 0 to 1, some  $\beta$  values caused the cluster centers to shift indefinitely, so a maximum clustering iteration was defined to terminate the process in these cases. Through experimentation, it was found that the best distribution of weights occurs somewhere between the first and final iterations. As a result, the objective function was calculated at each iteration in the WK-Means clustering. For the Static cluster optimization, the DE algorithm was used as described in the literature.

Since the WK-Means algorithm was being executed several times for each member in the population, the computation time was significantly large for the serial approach. SHARCNET [18] was used to rent a single CPU for each member in the population, thus allowing WK-Means to be computed in parallel for the entire population. In doing so, an entire generation was computed in the same amount of time as a single member. The pseudocode for the Static cluster optimization process can be seen as follows.

---

**Algorithm 3** Static Cluster Optimization Algorithm

---

**Input:** Weighted road network points, number of clusters, maximum WK-Means iteration, and DE parameters

**Output:** Optimal cluster center locations and weights

---

**create** the population of random  $\beta$  values

**for** generation **in** generations **do**

*# Do this in parallel*

**for**  $\beta$  **in** population **do**

**run** WK-Means

**calculate** the weight deviation

*# Do this in parallel*

*# Create new population*

**for** member **in** population **do**

**select** 3 random members **of** the population

**create** mutant vector

**run** WK-Means **for** mutant vector

**calculate** weight deviation

**select** member **or** mutant vector

**add** selection **to** new population

**return** cluster center locations **and** cluster weights of optimal cluster configuration

---

### 3.6.3. Dynamic Clustering

The Dynamic cluster optimization process is very similar to the Static cluster optimization, except each cluster  $\beta$  value was tuned rather than a single  $\beta$  value for all clusters, this can be formulated in Eq. (3-8). Additionally, the cluster center locations and

weights remain fixed from the Static clustering, and rather than running the entire WK-Means algorithm, all points were simply joined to the nearest weighted cluster. The objective of this clustering is to minimize the deviation of cluster weights *in the new* configuration which does not include the cluster weights used to partition the points. This can be formulated in Eq. (3-9).

$$f_6 = \sum_{j=1}^k \sum_{i=1}^n w_j^{\beta_j} \sqrt{(x_{i,x}^{(j)} - c_{j,x})^2 + (x_{i,y}^{(j)} - c_{j,y})^2} \quad (3-8)$$

$$f_7 = \sqrt{\frac{1}{k} \sum_{j=1}^k (w_{j,new} - \mu_{w,new})^2} \quad (3-9)$$

The Dynamic Cluster optimization algorithm can be seen as follows.

---

**Algorithm 4** Dynamic Cluster Optimization Algorithm

---

**Input:** Weighted road network points, cluster center locations and weights from the Static optimization process, and DE parameters

**Output:** Optimal cluster configuration

---

**create** population of 10 random  $\beta$  values

**for** generation **in** generations **do**

*# Do this in parallel*

**for** member **in** the population **do**

**join** each point in the road network **to** the nearest weighted cluster center

**calculate** the new weight deviation

*# Do this in parallel*

*# Create new population*

**for** member **in** population **do**

**select** 3 random members of the population

**create** mutant vector

**join** each point in the road network to the nearest weighted cluster center

**calculate** the new weight deviation

**select** member **or** mutant vector

**add** selection **to** new population

**return** optimal cluster configuration

---

### 3.7. Results & Analysis

This section covers the results and analysis of the current waste collection configuration, the clustered waste collection configuration, and a comparison between them.

#### 3.7.1. Current Configuration Analysis

Using the methods discussed in this chapter, a complete analysis of the current routing configuration was conducted for The City of Oshawa. A complete set of route simulations were generated using the proposed waste collection route optimization algorithm, and accompanying statistics were used to quantify the fitness of the respective routes and collection areas. In total, 111 routes were simulated, and all route shapefiles and animations were created. The animations created can be used by the waste collection team as a recommendation for the best possible way to service a collection route. An example of a route simulation can be seen in Figure 3-10, where all roads start off grey, and as they get traversed, the colour updates according to the legend provided. Additionally, each collection area's statistics can be seen in python-generated statistic tables, as seen in Figure 3-11.

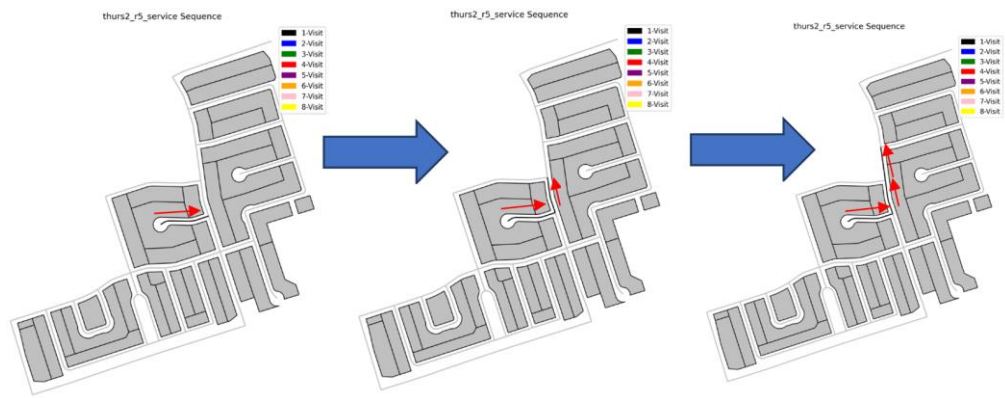


Figure 3-10 Frame-by-Frame Animation Example

### Fri. Week 1 Area Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
fri1_r1_service	13.12	18.01	0.4	4.24	24.6
fri1_r2_service	13.61	19.16	0.36	4.22	24.22
fri1_r3_service	12.06	17.3	0.34	4.98	23.23
fri1_r4_service	13.67	17.77	0.34	4.37	23.6
fri1_r5_service	8.87	12.89	0.25	4.37	17.79
fri1_r6_service	9.87	12.2	0.27	3.77	18.02
fri1_r7_service	9.26	12.65	0.25	3.92	17.61
fri1_r8_service	8.42	11.28	0.24	3.72	16.27
fri1_r9_service	11.36	15.25	0.3	3.87	20.4
fri1_r10_service	14.75	18.94	0.4	4.58	25.86
fri1_r11_service	12.47	17.02	0.33	4.31	22.6
<b>Total</b>	<b>127.49</b>	<b>172.47</b>	<b>3.48</b>	<b>46.35</b>	<b>234.22</b>

Figure 3-11 Python-Generated Statistic Table for the Purple Friday Area in Week 1 Collection

An analysis of the workload distribution across each day of the week for the current collection area configuration was used as the benchmark to quantify the improvements made through the proposed workload balance algorithm. By summing the objective function values for the areas covered each day, the deviation is calculated. Graphically, the current workload distribution can be seen in Figure 3-12. The workload deviation works out to be 103.69 and 102.84 for Week 1 and Week 2 collection, respectively.

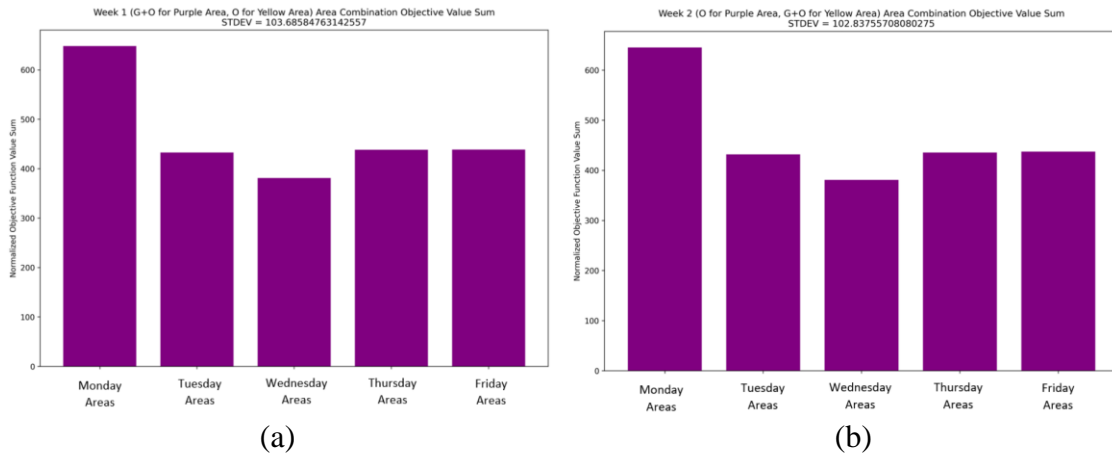


Figure 3-12 Workload Distribution for the Current Collection Area Configuration in Week 1 (a) and Week 2 (b)

Two other solutions are possible using the existing collection areas. The first one is where the colour assignment remains the same (purple areas remain purple, and yellow areas remain yellow) and different areas are paired together, and the second one is where all areas begin colourless and the workload balance algorithm chooses the colour and assigns the day of the week. A map of the first solution can be seen in Figure 3-13, and the corresponding workload distribution plots can be seen in Figure 3-14, where the workload deviation works out to be 83.80 and 85.69 for Week 1 and Week 2 collection, respectively.

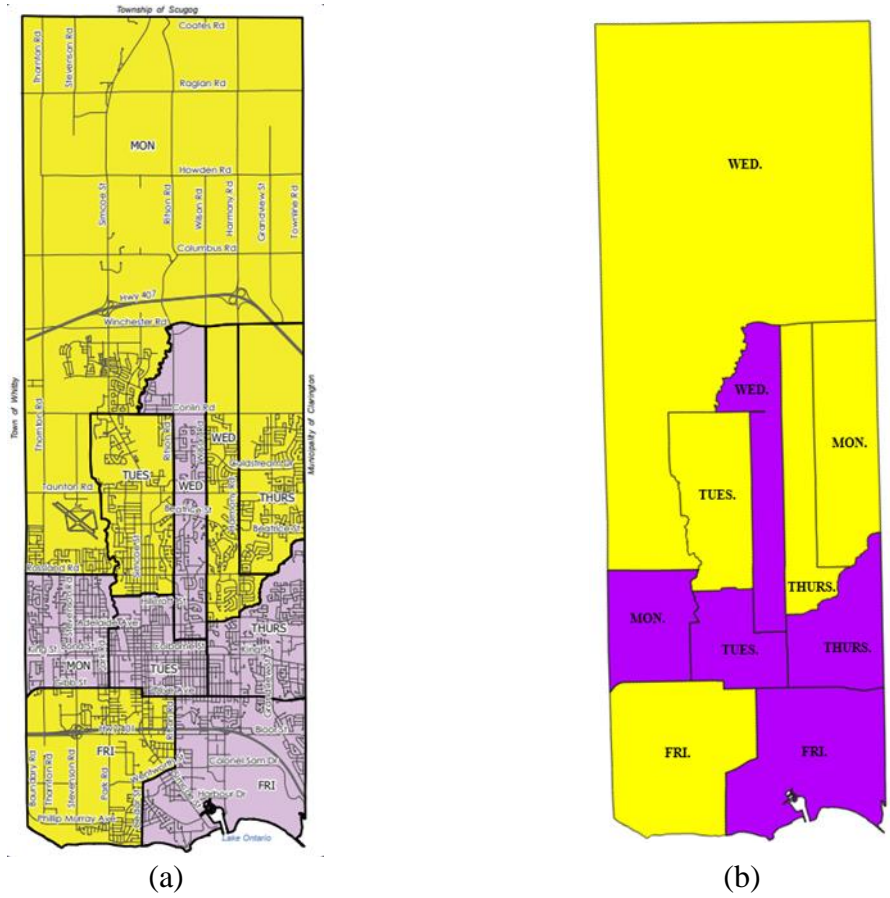


Figure 3-13 Current Configuration (a) vs. Solution 1 Configuration (b)



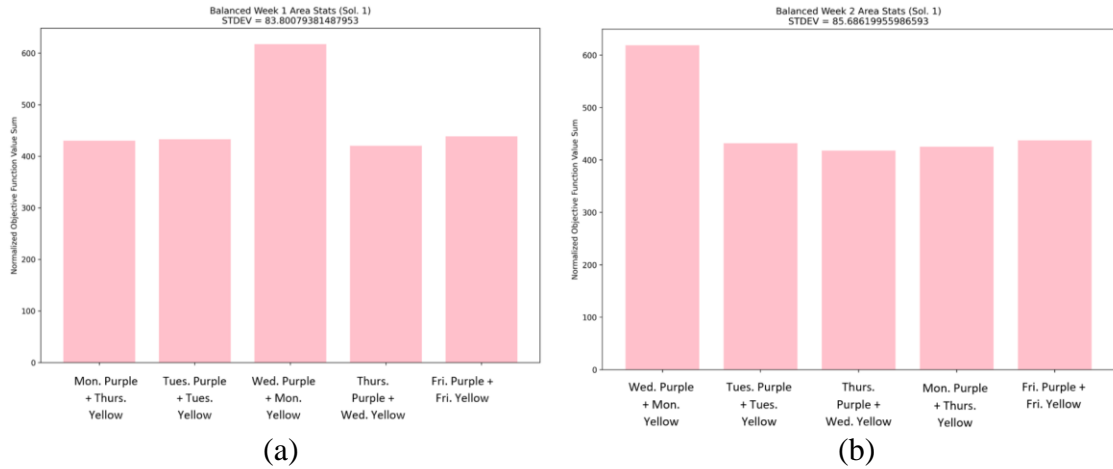


Figure 3-14 Workload Distribution for Solution 1 Collection Area Configuration in Week 1 (a) and Week 2 (b)

Finally, a map of the second solution can be seen in Figure 3-15, and the corresponding workload distribution plots can be seen in Figure 3-16, where the workload deviation works out to be 75.06 and 77.07 for Week 1 and Week 2 collection, respectively.

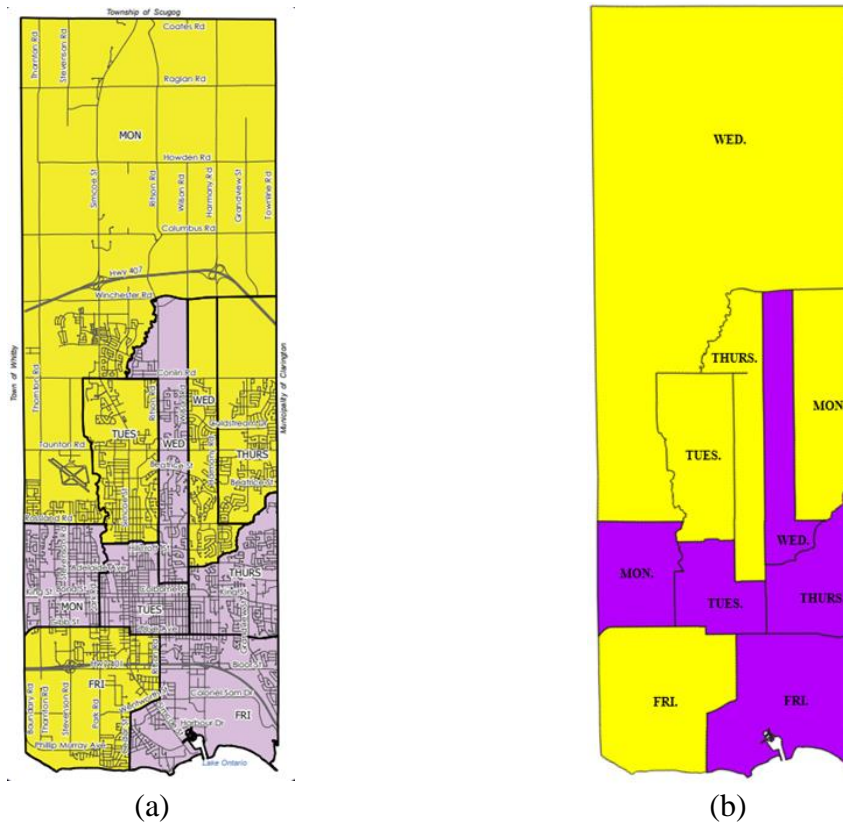


Figure 3-15 Current Configuration (a) vs. Solution 2 Configuration (b)

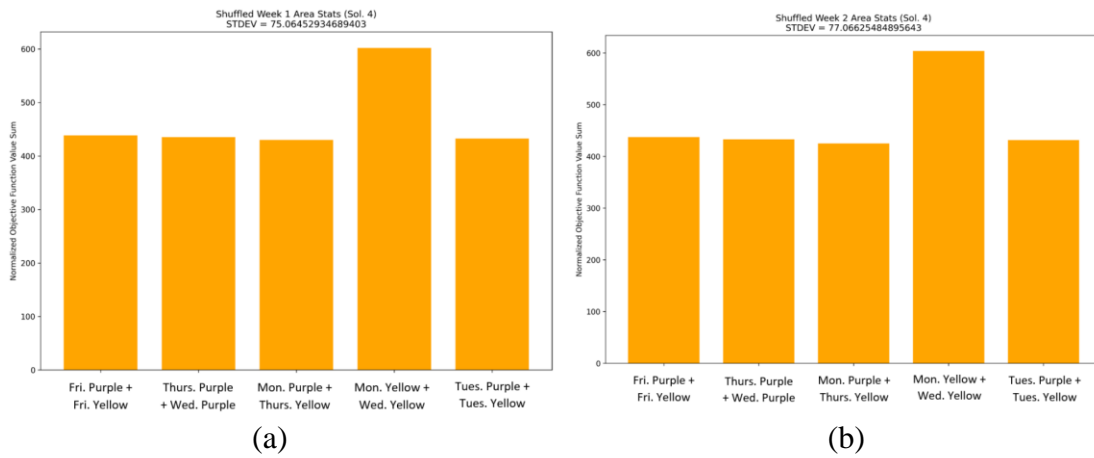


Figure 3-16 Workload Distribution for Solution 2 Collection Area Configuration in Week 1 (a) and Week 2 (b)

The improvements made through the workload balance algorithm can be summarized in Table 3-4. Analyzing the results, the optimal workload distribution is found in Solution 2 where the colours and days are assigned from scratch. However, when it comes to implementing these solutions in the real world, Solution 1 may be ideal as it may cause less confusion amongst residence since none of the collection weeks change (e.g., garbage and organics, and only organics still go out on the same week). In Solution 2, only 2 of the areas change the collection week, but this small change can yield much more improvement with respect to the workload balance across each day of the week. So ultimately, there is a tradeoff between the best workload distribution and customer satisfaction.

Table 3-4 Improvements Made by the Workload Balance Algorithm

Solution	Week 1 Deviation	Week 1 Improvement	Week 2 Deviation	Week 2 Improvement
Current	103.69		102.84	
Solution 1 (Same Colour Assignment)	83.80	19.18%	85.69	16.68%
Solution 2 (Colours and Pairs Assigned)	75.06	27.61%	77.07	25.06%

### 3.7.2. Clustered Configuration Analysis

This section will make use of the Static and Dynamic cluster optimization processes mentioned in Section 3.6 to generate new waste collection areas in the city. To quantify the improvements of the proposed clustering methods, an analysis of the current dwelling distribution was conducted. The deviation was calculated to be 376.60 dwellings and can be seen graphically in Figure 3-17.

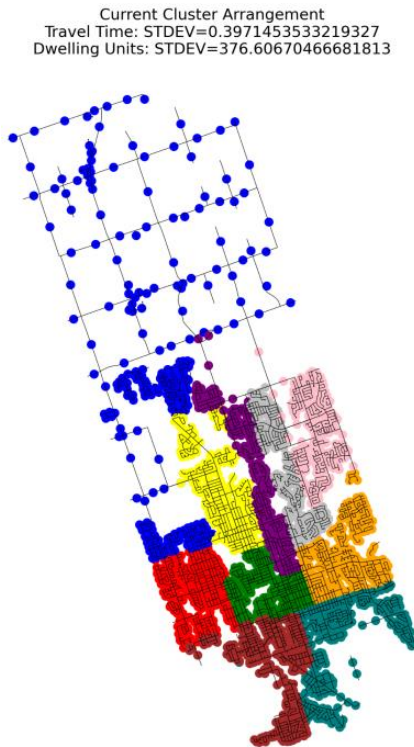


Figure 3-17 Current Collection Area Configuration Dwelling Deviation

For both the Static and Dynamic clustering, a population size of 32 was used. To make use of parallel computing, 32 CPUs were rented from SHARCNET, 1 for each member of the population. The remainder of the clustering and optimization parameters can be found in Table 3-5 for both optimization processes. As seen in Table 3-5, the computation time for the Dynamic clustering was far less than that of the Static clustering. This is because the Static clustering runs the WK-Means several times for each member of the population, and the Dynamic clustering simply joins the points to the nearest weighted cluster center that has already been computed by the Static clustering. So, the Dynamic clustering can manage to have a significantly larger number of generations while still yielding far less computation time.

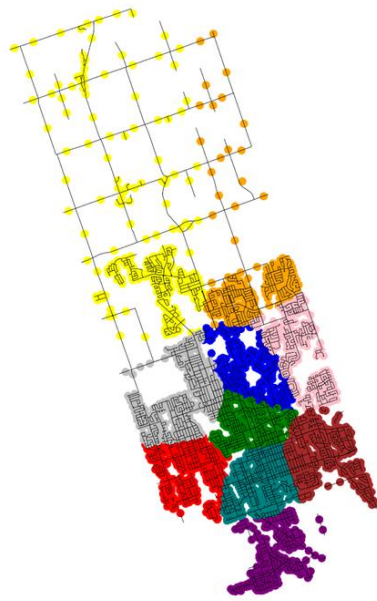
Table 3-5 Static and Dynamic Cluster Optimization Parameters for Waste Collection

Parameter	Value	
	Static Clustering	Dynamic Clustering
Number of Control Variables ( $\beta$ )	1	10
Population Size (Number of CPUs)	32	32
Crossover Rate	1	Linearly decreasing (0.95-0.50)
Mutation Rate	Linearly decreasing (0.8-0.05)	Linearly decreasing (0.9-0.05)
Generations	200	400
Clusters	10	10
WK-Means Runs	5	/
WK-Means Stop Iteration	600	/
Run Time	$\approx 7$ hr	$\approx 0.4$ hr

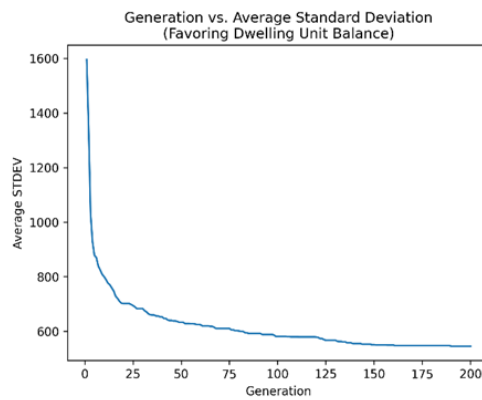
The output of the Static clustering algorithm can be found in Figure 3-18. From Figure 3-18 (a), a deviation of 400.26 dwellings is observed when  $\beta$  is 0.1763 for all clusters. Examining Figure 3-18 (b), an exponentially decreasing trend is found for the

average deviation of each generation, which is ideal in optimization processes. Although the Static cluster optimization did not yield any improvement compared to the existing configuration, the cluster center location and weights were used in the Dynamic cluster optimization algorithm to fine tune the clusters.

New Collection Area Assignment (k=10, beta=0.17633581609859733, runs=5)  
Generations=200  
Dwelling Units: STDEV=400.2594283711503



(a)

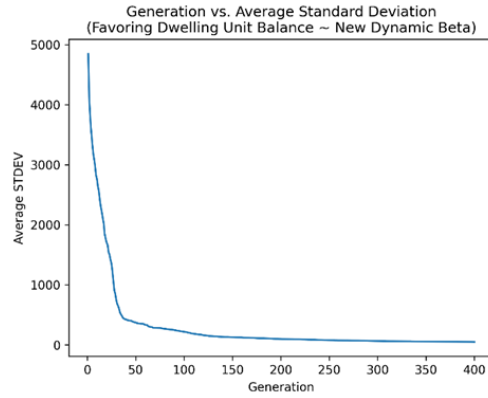
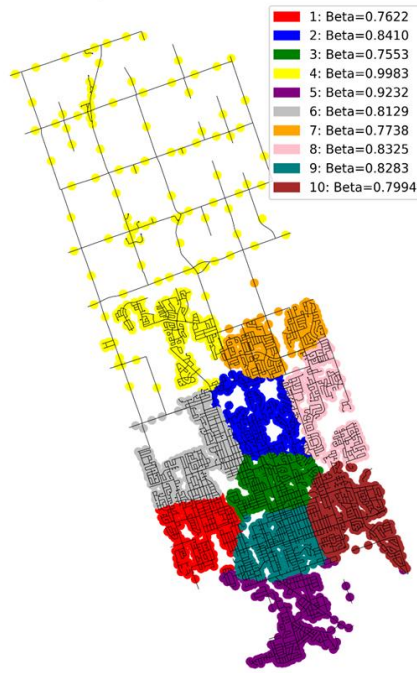


(b)

Figure 3-18 Clusters Produced by the Static Cluster Optimization Process (a), and the Evolutionary Improvement Trend (b)

The output of the Dynamic clustering process can be found in Figure 3-19. Examining Figure 3-19 (a), a dwelling deviation of 46.13 is seen, and the legend shows each cluster's respective  $\beta$  value that is proportional to the weights and locations. Like the Static process, an exponentially decreasing trend is shown in the average deviation for each generation in Figure 3-19 (b).

New Collection Area Assignment (k=10, beta=Dynamic, Generations=400)  
 Dwelling Units: STDEV=46.12819094653507



(a)

(b)

Figure 3-19 Clusters Produced by the Dynamic Cluster Optimization Process (a), and the Evolutionary Improvement Trend (b)

The improvements made using the proposed 2-stage cluster optimization process can be summarized in Table 3-6. Examining Table 3-6, the best approach was the Dynamic clustering method with a dwelling deviation improvement of 87.75% compared to the existing configuration.

Table 3-6 Dwelling Deviation Improvements

Configuration	Dwelling Deviation	Improvement
Current	376.61	/
Static Clustering	400.26	-6.28%
Dynamic Clustering	46.13	87.75%

To assign routes within the newly defined collection areas, the same clustering methods were applied on each area. First, some small manual modifications were made to the clusters to clearly define the collection area boundaries to prepare them for the vehicle

assignment clustering. The parameters used in the vehicle assignment clustering can be found in Table 3-7. An example of the vehicle assignment clustering can be seen in Figure 3-20 when it is applied to cluster 6 in Figure 3-19 (a).

Table 3-7 Static and Dynamic Cluster Optimization Parameters for the Vehicle Assignment

Parameter	Value	
	Static Clustering	Dynamic Clustering
Number of Control Variables ( $\beta$ )	1	11
Population Size (Number of CPUs)	32	32
Crossover Rate	1	Linearly decreasing (0.95-0.50)
Mutation Rate	Linearly decreasing (0.8-0.05)	Linearly decreasing (0.9-0.05)
Generations	100	250
Clusters	11	11
WK-Means Runs	5	/
WK-Means Stop Iteration	600	/
Run Time	$\approx 30$ min/area	$\approx 5$ min/area

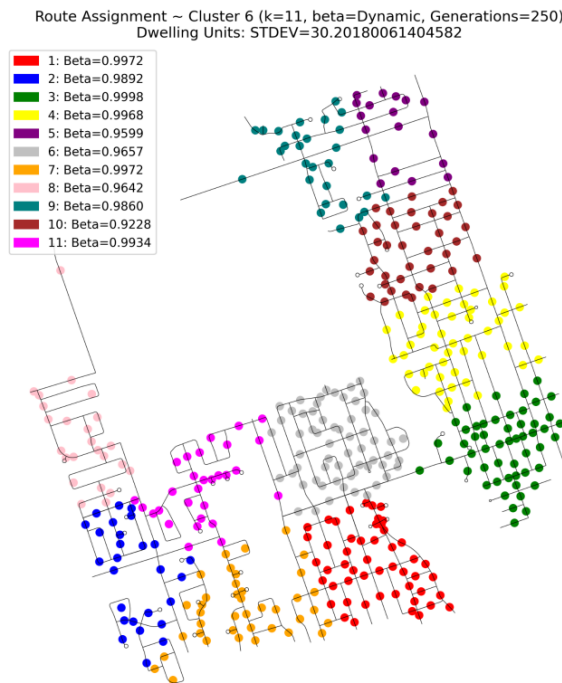


Figure 3-20 Vehicle Assignment using Static and Dynamic Clustering in Cluster 6

Still some manual modifications were necessary to resolve minor clustering issues seen in the vehicle assignment. Specifically, there were some outliers in adjacent neighborhoods that had crescents and road segments protruding into a neighboring cul-de-sac without having any direct connection via the road network. Upon the completion of careful manual modifications, the new stop densities in each respective route were calculated. Since the new route configuration differs from the existing route configuration, using the “route\_block” areas to filter out the pickup locations to calculate the route density in each respective route cannot be used. Instead, a new metric was used where the number of dwellings in the respective routes can be divided by the servicing distance, thus creating a DUs/km rate. When comparing results generated in the existing configuration to the areas created by the clustering methods, there were some discrepancies with regards to the travel time because of the new method used to approximate the route density. Additionally, more safe turn-around locations were added to avoid u-turns, which slightly increased the distance, travel time, and fuel consumed. The route simulations were completed for the new routes using the same methods discussed in Section 3.3, and animations with accompanying statistic tables were generated, like the ones in Figure 3-10 and Figure 3-11, respectively.

Since the clustered collection area boundaries are different from the existing ones, a side-by-side comparison of the statistics for each collection area is not an ideal way to compare the effectiveness of the proposed clustering methods. Instead, the total statistics for the whole city were compared, which can be summarized in Table 3-8.



Table 3-8 Total City Statistics for the Current Arrangement and Clustered Arrangement for Week 1 Collection

<b>Configuration</b>	<b>Total Travel Distance (km)</b>	<b>Total Fuel Consumed (L)</b>	<b>Total Travel Time (hr)</b>	<b>Total Collection Time (hr)</b>	<b>Total Objective Function Value</b>
Current	1461.67	1819.11	37.08	441.04	2482.92
Clustered	1481.09	1868.25	44.28	441.04	2594.99
Percentage Difference from the Current Configuration	1.32%	2.70%	19.42%	0%	4.51%

Examining the total statistics in Table 3-8, the largest percentage difference of 19.42% can be found in the travel time. As previously mentioned, a different method of calculating the average servicing speed for the routes was used, which would ultimately lead to this large difference. All other statistics differed by a maximum of 4.51%. The increase in each respective statistic can be justified though the inclusion of additional safe turn-around locations that were added to the routes to remove u-turns. The small difference between the clustered and current configuration leads to the conclusion that the proposed clustering methods can produce collection areas and assign servicing routes comparable to human judgment. This study validates the clustering methods as an appropriate means to cluster geographical data into several divisions with respect to balancing a specified workload, thus implying that it can be used in other routing applications such as street-sweeping and snowplowing.

The final analysis of the clustered configuration consists of assigning the collection week and daily pairs. For this, the proposed workload balance algorithm was used described in Section 3.5. Again, the objective of this process is to properly distribute the workload

across each day of the week. The parameters used in the GA optimization can be seen in Table 3-9.

Table 3-9 GA Parameters for the Workload Balance Algorithm

Parameter	Value
Length of Chromosome	10
Population Size	150
Crossover Rate	0.9
Mutation Rate	0.05
Generations	100
Runs	20
Run Time	< 1 min

The structure of the best-found chromosome from the workload balance algorithm was found as ['area\_7', 'area\_2', 'area\_1', 'area\_5', 'area\_9', 'area\_10', 'area\_3', 'area\_4', 'area\_6', 'area\_8']. As mentioned in Section 3.5, an encoding method was used to assign the collection week and daily pairs of collection areas in the chromosome. A side-by-side comparison of the existing configuration and the new configuration can be seen in Figure 3-21 with the area number in brackets. The workload deviation was calculated to be 64.25 and 64.23 for Week 1 and Week 2 collection, respectively. Examining Figure 3-21, the existing collection areas have a somewhat polygonal boundary, whereas the clustered configuration yields “blob-like” areas due to the unique  $\beta$  value optimized for each respective cluster.

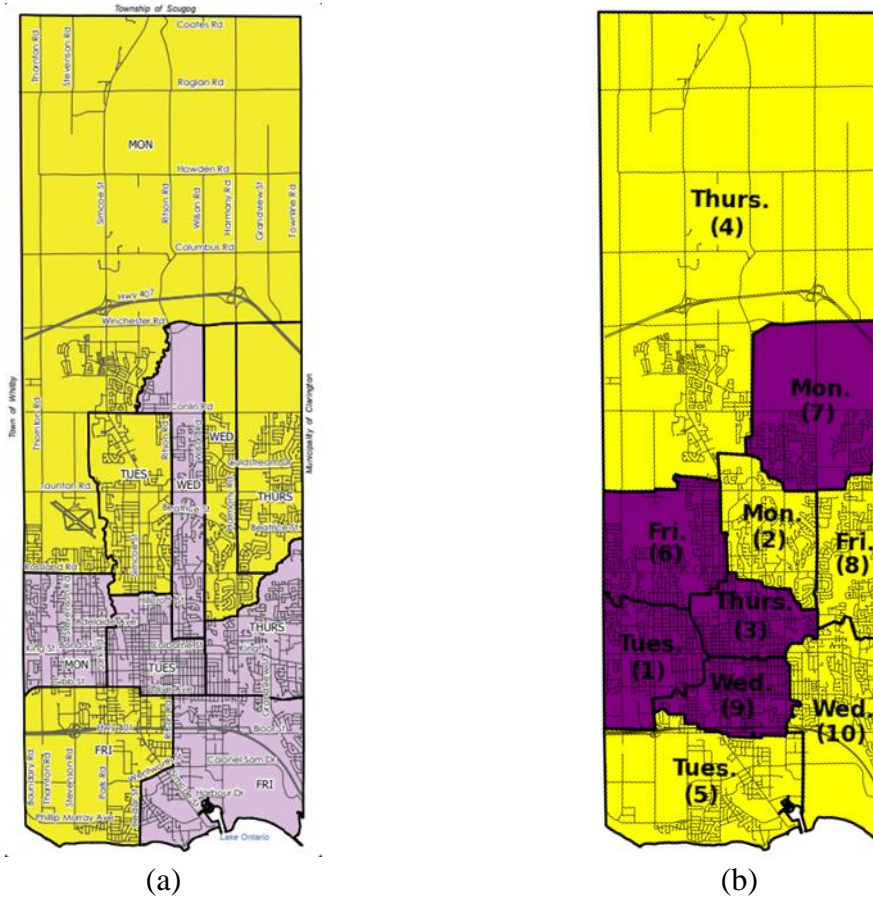


Figure 3-21 Current Configuration (a) vs. Clustered Configuration with Collection Weeks and Daily Pairs Assigned (b)

Table 3-10 summarizes the improvements of all possible collection area configurations with respect to the workload balance problem. The best solution is the clustered configuration with a Week 1 and Week 2 improvement of 38.04% and 37.54%, respectively. However, when it comes to implementing the proposed clustered collection areas in the real-world, some issues may arise. Examining the current configuration in Figure 3-21 (a), all daily pairs of collection areas are adjacent to each other, resulting in less travel distance from one area to the next. Figure 3-21 (b) yields an optimal workload balance at the expense of separating the daily collection area pairs. In the clustered configuration, the Thursday and Friday areas are not adjacent to each other, thus requiring

more deadhead travel between the areas. This presents a tradeoff in the optimal workload balance and the feasibility of the clustered collection area configuration.

Table 3-10 Comparison of All Solutions for the Workload Balance Problem

<b>Solution</b>	<b>Week 1 Deviation</b>	<b>Week 1 Improvement</b>	<b>Week 2 Deviation</b>	<b>Week 2 Improvement</b>
Current	103.69		102.84	
Solution 1 (Same Colour Assignment for Existing Areas)	83.80	19.18%	85.69	16.68%
Solution 2 (Colours and Pairs Assigned for Existing Areas)	75.06	27.61%	77.07	25.06%
Solution 3 (Colours and Pairs Assigned to Clustered Areas)	64.25	38.04%	64.23	37.54%

### 3.8. Conclusions

In this chapter, a complete analysis of the current routing configuration for The City of Oshawa was conducted. To do so, a combination of heuristic approaches were used to create simulations necessary for calculating route-specific statistics for Week 1 and Week collection. Then, a GA was used to provide recommendations as to how the existing collection areas can be assigned to minimize the workload distribution across each day of the week. Results show that simply changing the collection day pairs can dramatically improve the workload distribution for the waste collection team.

A novel 2-stage cluster optimization algorithm was also proposed with the objective of minimizing the deviation of weights across each cluster. The proposed algorithm was tested, and verified, in The City of Oshawa by minimizing the deviation of dwellings in each cluster. The results show that the proposed clustering methods yield a collection area configuration that can reduce the workload distribution more than any combination of

existing collection areas. Although the proposed algorithms were tested on the waste collection process, it can be applied to any other routing application in which proper partitioning of an area is needed.

Regarding the limitations of the proposed methods discussed in this chapter, all optimization algorithms used involve stochastic processes. This means that, each time the simulation is executed, a different set of results will present itself. Plus, it is not guaranteed that the global optimal solution can be found, which is why the heuristic methods are used to generate an acceptable solution. Also, the results in Table 3-8 show a slight increase in the overall statistics for the waste collection routes in the clustered configuration. This was justified by the inclusion of safe turn-around locations for the collection vehicles, but ideally a new solution should yield results that are improved with regards to the overall statistics *and* workload balance, thus leading to the conclusion that the proposed clustering algorithms can be improved to reduce the overall statistics. Chapter 4 aims to improve the clustering methods to reduce the overall statistics of the street-sweeping operations in The City of Oshawa, as well as ensure a proper workload distribution implemented across each sweeping area.

## Chapter 4. Street-sweeping Route Optimization

### 4.1. Introduction

Street-sweeping plays an important role in public health as particulate matter from the abrasion of the roads surface, break dust, and litter, make their way into stormwater drainage systems and back into the ecosystem [85]. The accumulation of debris in catch basins also requires cleaning at an additional cost. Limiting the amount of debris on the roads *before* it reaches catch basins can further reduce this cost. Additionally, street-sweeping can improve public safety in the Fall months where significant amounts of leaves accumulate on the roads surface making it hard to detect potholes, crosswalks, as well as reducing tire friction [86]. Similarly, roadway debris like oil, dirt, and sand in the Spring and Summer months can cause slippage when breaking. Mechanical street sweepers use a set of rotating brushes and suction nozzles to sweep and collect debris on the road [2], the debris is collected in the vehicle until it is deposited at a yard for testing and final disposal. It is also common to wet the pavement in this process to make smaller particles of debris easier to collect, thus each vehicle is equipped with a water tank and appropriate nozzles.

Since street-sweeping operations are costly, realistic simulations with accompanying statistics can be used as a theoretical guide for real-world application. In this chapter, the street-sweeping operations in The City of Oshawa will be simulated and theoretically optimized. The supplied GIS data was modified to incorporate important information regarding the street-sweeping operations, which will be discussed. A novel street-sweeping route optimization algorithm was developed consisting of several different heuristic approaches, namely the 3-Phase Augment Merge algorithm, A\*, GA, TS, Hierholzer's algorithm, TR, and ACO. The operating principle behind the proposed algorithm will be

discussed. The 2-stage cluster optimization process will also be used to define new sweeping areas for the city, however some modifications to the clustering approaches mentioned in Section 3.6 are applied that will be explored. Finally, a complete analysis of the current sweeping configuration will be compared to the clustered configuration to highlight the improvements made using the new sweeping areas. Additionally, the proposed algorithms will make use of seasonal conditions, meaning that different debris collection rates will be used for Spring, Summer, and Fall to reflect the Spring cleanup after winter operations, and the increased volume of leaves in the Fall months. As a result, 3 sets of solutions will be analyzed. The proposed algorithms will also incorporate several real-world constraints in the routing problem, such as the debris capacity, shift length, water capacity, fuel capacity, u-turn elimination, and different temporary debris storage facility and depot locations.

#### **4.2. Data Preparation**

The merged dataset discussed in Section 3.2.1 was used for the street-sweeping optimization problem, specifically merged attribute road network shapefile with intersection connectivity information and the downloaded intersection shapefile. A complete set of sweeping area PDFs were supplied by the city highlighting which roads belong to which sweeping area for the AC and RES road classes. The RES and AC road classes can be seen in Figure 4-1.



Figure 4-1 RES (Red) and AC (Blue) Roads in The City of Oshawa

The PDF maps also highlight which roads have a dense tree canopy overhanging the road. This information is important for the Fall months as there will be a higher debris generation rate on these roads, possibly requiring multiple passes. The sweeping area and canopy information were not included in the supplied GIS data, so it was added manually by referencing the supplied PDF maps. Specifically, 3 new attributes were added to the road network GIS file, “SWEEP\_CLAS,” “ROUTE\_NUM,” and “CANOPY.” The data used for the canopy attribute was collected in 2018, thus this data may be out of date with regards to tree growth. A simple data query can be used to isolate each route from the whole road network. A side-by-side comparison of the RES sweeping area 1 in PDF and shapefile format can be seen in Figure 4-2.



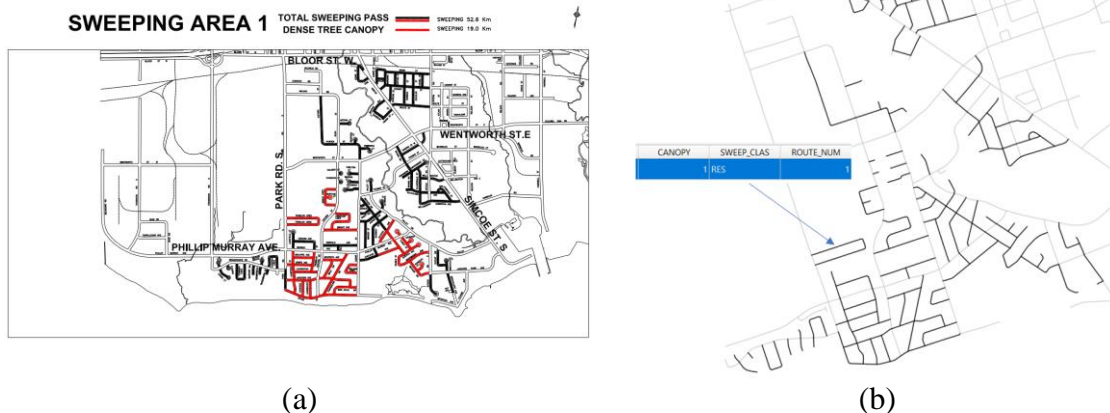


Figure 4-2 Route Area PDF (a) to Shapefile (b) Conversion

### 4.3. Street-sweeping Route Optimization Algorithm

In Section 3.2.2, the route shapefiles were created manually, meaning human judgment was used when selecting the roads needed to connect isolated parts of the route and the safe turnaround locations. Ideally, this should be done in a systematic way without human intervention. Since a route area can be considered a disconnected subgraph of the total road network, safe turnaround locations and optimal connections between the roads should be found using advanced optimization techniques. Also, entire sweeping areas cannot be swept in a single pass with respect to constraints (fuel, water, shift length, and vehicle debris capacity). So, a method of systematically dividing a sweeping area into several routes satisfying all constraints is needed.

To make the route optimization problem more realistic, each trip must begin at the depot, and end with a temporary debris storage facility-to-depot trip, thus requiring significant deadhead distances to travel to and from the depot and temporary debris storage facility locations in South Oshawa. The depot is the home base for equipment and vehicle storage, each shift begins and ends here. The temporary debris storage facility is modelled

as the waste management facility in the CARP. However, it is a location to temporarily store the street-sweeping debris before it is tested and properly disposed of. The general area of South Oshawa can be seen in Figure 4-3. South Oshawa contains both the temporary debris storage facility and depot locations which are approximately 1.5 km apart.



Figure 4-3 General Area of South Oshawa

For a disconnected subgraph  $g$  (representing a collection of serviceable roads) of  $G$  (the whole road network), there exists a minimum cost path from the depot that services every edge in  $g$  and ends with a temporary debris storage facility-to-depot trip. This can be formulated in Eqs. (4-1) to (4-3), where the traversal nodes in Eq. (4-2) include the starting and ending nodes of the edges in  $g$ ,  $d_{ij}$  is the shortest path from node  $i$  to  $j$ , and  $x_{ij}$  is a

binary indicator equal to 1 if node  $j$  is the next node to visit after  $i$  in the sequence. This can be modelled as a version of the TSP, where each edge has a starting and ending node related to the direction of travel.

$$f_8 = \sum_{i=1}^n \sum_{j \neq i, j=1}^n d_{ij} x_{ij} \quad (4-1)$$

$$x = \{x_1, \dots, x_n\} \quad (4-2)$$

$$x_1 = x_{depot} \quad (4-3)$$

$$x_{n-1} = x_{dump} \quad (4-4)$$

$$x_n = x_{depot} \quad (4-5)$$

However, as previously mentioned, a single tour cannot be used in each area due to the operational constraints of the routing problem, so several heuristic approaches are combined to solve the street-sweeping route optimization algorithm with multi-demand edges and many constraints. The overall flow of the proposed algorithm can be seen in Figure 4-4.

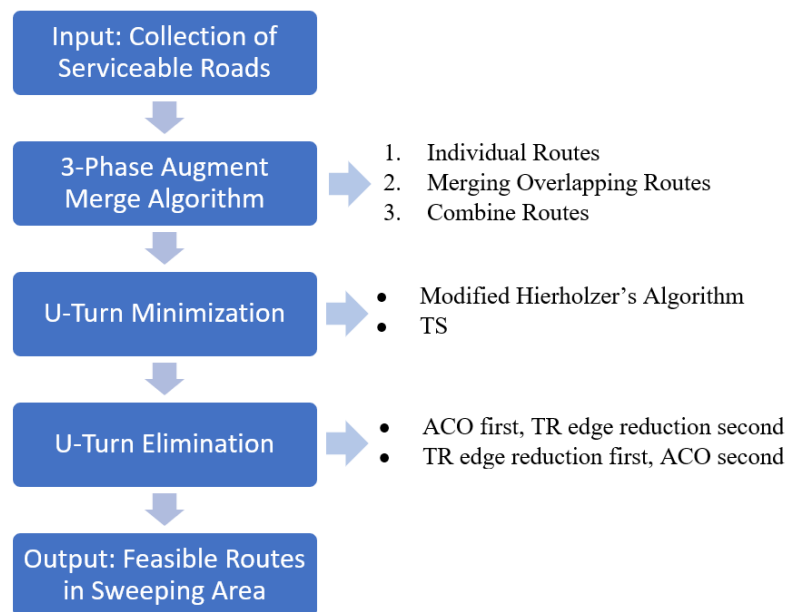


Figure 4-4 Overall Street-sweeping Route Optimization Algorithm Flow

### 4.3.1. 3-Phase Augment Merge Algorithm

Inspired by the work of Yang [40], a 3-phase augment merge algorithm was developed to find the optimal servicing routes within a given collection area while maintaining the operational constraints. The algorithm can be explained as follows.

---

**Algorithm 5** 3-Phase Augment Merge Algorithm

---

**Input:** Subgraph ( $g$ ) representing the serviceable edges, GA parameters

**Output:** Several feasible routes for a given sweeping area

---

*# Phase 1 – Individual Routes*

```
for edge in g do
    generate route servicing the edge
    add route to route list
```

sort route list (max to min distance)

*# Phase 2 – Merging Overlapping Routes*

```
for route in route list do
    for smaller route after route in route list do
        if route contains serviceable edge in smaller route do
            merge serviceable edge from smaller route into route
            delete smaller route
```

sort route list (max to min efficiency)

*# Phase 3 – Combine Routes*

```
while True do
    old route list == route list
    for route in route list do
        for less efficient route after route in route list do
            run GA to service all serviceable edges in route and less
            efficient route
            calculate statistics and efficiency improvement
            update max discovered efficiency improvement

            for max efficiency improvement route do
                if constraints satisfied and efficiency improvement > 0 do
                    add combined route to new route list
                    delete less efficient route from route list
                else do
                    add route to new route list

    if len(old route list) == len(new route list) do
        break
```

route list == new route list

**return** new route list

---

The efficiency can be formulated in Eq. (4-6), where  $d_{ij}^{(s)}$  is the distance of serviceable edge  $(v_j, v_j)$ , and  $d_{ij}^{(r)}$  is the distance of edge  $(v_j, v_j)$  in the total route. The efficiency improvement can be formulated in Eq. (4-7), where  $d_{ij}^{(s,m)}$  is the distance of serviceable edge  $(v_j, v_j)$  in the merged route,  $d_{ij}^{(r,m)}$  is the distance of edge  $(v_j, v_j)$  in the merged route,  $d_{ij}^{(r,1)}$  is the distance of edge  $(v_j, v_j)$  in the first individual route, and  $d_{ij}^{(r,2)}$  is the distance of edge  $(v_j, v_j)$  in the second individual route.

$$\eta = \frac{\sum_{i=1}^n \sum_{j \neq i, j=1}^n d_{ij}^{(s)}}{\sum_{i=1}^n \sum_{j \neq i, j=1}^n d_{ij}^{(r)}} \quad (4-6)$$

$$\eta_{improvement} = \frac{\sum_{i=1}^n \sum_{j \neq i, j=1}^n d_{ij}^{(s,m)}}{\sum_{i=1}^n \sum_{j \neq i, j=1}^n d_{ij}^{(r,m)}} - \frac{\sum_{i=1}^n \sum_{j \neq i, j=1}^n d_{ij}^{(s,m)}}{\sum_{i=1}^n \sum_{j \neq i, j=1}^n d_{ij}^{(r,1)} + \sum_{i=1}^n \sum_{j \neq i, j=1}^n d_{ij}^{(r,2)}} \quad (4-7)$$

The GA in the 3-phase augment merge algorithm computes the shortest path between servicing edges and does not pay any attention to the feasibility of the generated route with regards to u-turns. To account for the unsafe u-turns, a penalty distance should be added to the total route distance computed by the GA. Mentioned in Section 3.3, there are safe locations to perform u-turns, so these u-turns will not be penalized. A study was conducted where safe turnaround locations were found for 20 randomly selected intersections in the city. From the study, an average distance of 839.2 m was needed to detour and return to same intersection to resolve a u-turn. This distance was used as the penalty for unsafe u-turns in the GA. As a simplification, it was also assumed that the penalty travel speed was 50 km/hr (since this is the average speed limit in The City of Oshawa), so the shift length constraint will also be affected by the unsafe u-turn penalty. The constraints for the route

optimization can be found in Table 4-1, and the route simulation variables can be found in Table 4-2.

Table 4-1 Street-sweeping Constraints

<b>Constraint</b>	<b>Value</b>
Shift Length (Not Including Breaks + Misc. Activities)	6.5 hr
Fuel Capacity	189 L
Debris Capacity	6 m <sup>3</sup>
Water Capacity	1268 L

Table 4-2 Route Simulation Parameters

<b>Variable</b>	<b>Value</b>
Fuel Consumption Rate	0.55 L/km
Service Speed	5 km/hr
Water Dispersion Rate	14.61 L/km
RES Debris Collection Rate	0.14 m <sup>3</sup> /km for Spring 0.07 m <sup>3</sup> /km for Summer 0.35 m <sup>3</sup> /km for Fall (Normal Road) 0.44 m <sup>3</sup> /km for Fall (Dense Canopy Road)
AC Debris Collection Rate	0.175 m <sup>3</sup> /km for Spring 0.105 m <sup>3</sup> /km for Summer 0.35 m <sup>3</sup> /km for Fall

In Table 4-2, the fuel consumption rate and service travel speed were directly supplied by the city, but the water dispersion rate and debris collection rates were calculated using field data that was collected by the operations team. For the water dispersion rate, data was supplied to highlight how much water was used for a small sample of routes. From that, the water dispersion rate was calculated by finding the average water dispersion rate for the sample of routes provided. For Fall, a small selection of RES route data was supplied that revealed the cubic yardage of leaves collected. The average debris collection rate was calculated from the small sample of RES routes with respect to the servicing distance, and the same rate was applied to AC routes (since this data was missing). For the case of RES

canopy roads, it was assumed that they produce 25% more debris. In Spring and Summer, pile mass information was supplied, but it was not used since the debris constraint of the vehicle was in terms of volume. This information may have been used if the density of the pile debris was known (so it can be converted from mass to volume), but this information was missing. So, a scale was applied to the Fall debris rates to calculate the Spring and Summer debris rates. From conversations with the operations team, they agreed that the Fall seasons contribute the largest volume of debris, so the Spring and Summer debris rates will be a factor of the Fall debris rate. Using a scale of 1 to 5, it was assumed that Fall would be a 5, Spring RES is a 2, Spring AC is a 2.5, Summer RES is a 1, and Summer AC is a 1.5, resulting in the values shown in Table 4-2.

#### **4.3.2. U-Turn Minimization Algorithm**

Eventually the u-turns will be eliminated from the route sequence, but deadhead distance is increased because of finding safe turn-around locations. Before removing the u-turns, they should be minimized to reduce the amount of deadhead travel needed to resolve unsafe u-turns. So, a heuristic approach was developed that makes use of the modified Hierholzer's algorithm discussed in Section 3.3, and a TS algorithm like the one proposed by in [44] and [45].

A route can be modelled as a sequence of nodes in a specified traversal order. Unsafe u-turns can be identified by examining the previous and next node in the sequence; if the nodes are the same, a u-turn has occurred, physically seen as there-and-back motion between 2 nodes. Since some routes consist of more edges than others, 2 different methods of u-turn minimization were used, and the solution that yields the least u-turns was selected for the next process.

The first method is to use the modified Hierholzer’s algorithm from Section 3.3. However, the solution needs to be encoded into a graph-like structure for Hierholzer’s algorithm to be executed. The input solution consists of a series of nodes in a specified traversal order, where each pair of nodes relates to a directed edge. Knowing that, a graph structure can easily be created by iterating through each node in the solution and creating the weighted connection to the next node. Since the trip starts at the depot and ends with a temporary debris storage facility-to-depot trip, the depot will be the start node, and the constant temporary debris storage facility-to-depot sequence can be removed from the end of the solution and added back after optimization since it is constant. By removing this, an Eulerian tour cannot be completed, and instead an Eulerian path was created that starts at the depot and ends at the temporary debris storage facility. The modified Hierholzer’s algorithm works as described in Section 3.3 with the objective of creating a Eulerian path with minimal u-turns.

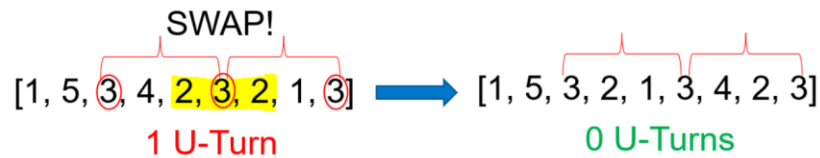


Figure 4-5 Cycle Permutation Example for U-Turn Minimization

The other method of u-turn minimization is the TS approach. Cycle permutations were applied to the node sequence with the objective of minimizing the unsafe u-turns, an example can be seen in Figure 4-5. For this method, a multi tabu list approach was developed where one tabu list stores the tabu solutions, and the other stores tabu nodes that have been selected to evaluate permutations. By using 2 lists, the chances of getting stuck



in locally optimal solutions is further reduced while promoting greater exploration of the solution space. The multi tabu list approach works by selecting a node to perform cycle permutations with that is not in the tabu nodes list, and the optimal permutation with the least u-turns that is not in the tabu solutions list is used as the next solution. The tabu solutions and nodes lists remain fixed in length, so the items that have been in the list the longest get removed first. The search is an iterative process, where the global best solution is constantly being updated until the final tabu iteration. The pseudo code for the proposed multi-tabu list search algorithm can be seen as follows.

---

**Algorithm 6** Multi List TS Algorithm

---

**Input:** Route and TS Parameters

**Output:** Optimal node traversal order to minimize u-turns

---

**for** iteration **in** iterations **do**

**randomly select** cycle node that **is not in** the tabu nodes list

**evaluate** fitness **of all** permutations of cycles **for** the selected node

    current route == best permutation that **is not in** the tabu solutions list

**add** current route **to** tabu solutions list

**add** cycle node **to** tabu nodes list

**update** tabu solutions list **if** len(tabu solutions list) > max length

**update** tabu nodes list **if** len(tabu nodes list) > max length

**update** best found solution

**return** best found solution

---

Both the modified Hierholzer's algorithm and multi-list TS algorithm were applied to the routes generated by the 3-phase augment merge algorithm. Through experimentation, it was found that the modified Hierholzer's algorithm and multi-list TS algorithm would generate different results for some routes, so the best solution found by either method was used in the next part of the algorithm.

### 4.3.3. U-Turn Removal Algorithm & Redundant Edge Reduction

After the u-turn minimization stage, there still may be remaining u-turns that could not be resolved. These u-turns should be removed as they pose a risk to public safety. To remove the unsafe u-turns, a forward searching ACO algorithm (FS-ACO) was developed. In ACO, a colony of ants can be simulated travelling on the weighted edges of a graph to find the shortest tour beginning at the nest location. After each iteration, the pheromone levels of the edges can be updated, all edges undergo a small percentage of pheromone evaporation, and the edges traversed in the best path are rewarded with an increase in pheromone proportional to the distance of the path.

Initially, the ants generate random paths, but as the iterations increase, the shortest path begins to reveal itself through the reward of pheromones. When ants select the next edge to traverse, an informed random decision is made where the edges with higher pheromones have a greater chance of being selected. This probability of selecting node  $j$  can be seen in Eq. (4-8) [9] where  $\tau_{ij}$  is the pheromone level on edge  $(v_i, v_j)$ ,  $\alpha$  is the pheromone influence factor,  $d_{ij}$  is the distance of edge  $(v_i, v_j)$ ,  $\beta$  is the distance influence factor, and  $S$  is the set of nodes adjacent to  $i$ .

$$p_{ij} = \frac{\tau_{ij}^{\alpha} (\frac{1}{d_{ij}})^{\beta}}{\sum_{k \in S} \tau_{ik}^{\alpha} (\frac{1}{d_{ik}})^{\beta}} \quad (4-8)$$

In the case of u-turn removal for the street-sweeping optimization problem, ACO was modified appropriately for the problem. First, the unsafe u-turns should be identified by the methods discussed in Section 4.3.2. As each u-turn is representative of there-and-back travel on an edge, this edge should be removed from the graph before running the FS-ACO algorithm to prevent more u-turns from being added by the ants. Additionally, the road

network is a large dataset, and if the ants were simply set out to find a seemingly random path in network to return to the start node, this will result in large computation time. To prevent this from happening, a radius was defined around the u-turn node to filter out a smaller portion of the road network. Finally, when the ants were simulated travelling along the edges of the graph, they used a novel forward-searching approach where the previously visited edge is ignored when selecting the next edge to travel. In some cases, the previously visited edge will be the only available edge to travel, like courts and physical dead ends. These are the only exceptions to the FS-ACO as they are safe turnaround locations. An example of the FS-ACO process can be seen in Figure 4-6.

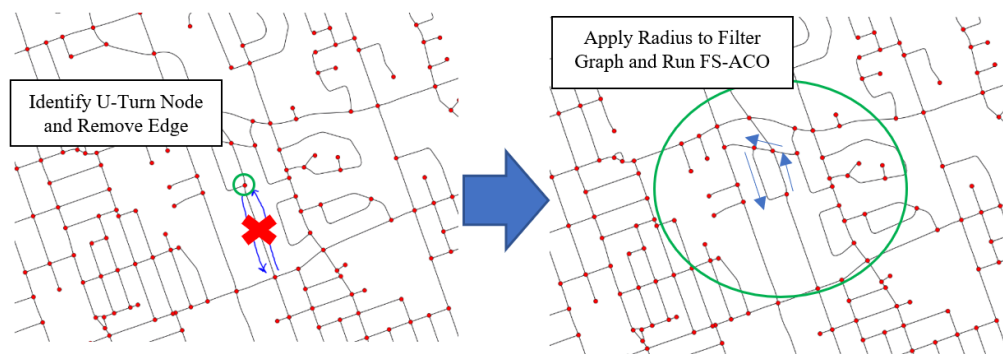


Figure 4-6 FS-ACO for U-Turn Removal

An extra measure is taken to further reduce the deadhead travelling. A method of redundant edge reduction was developed that is based on the concept of TR, first introduced by Aho et al. [14]. In literature, a TR is the minimum edge subgraph that provides a strong connection between all nodes in the original graph. This concept can be used in the routing problem as there should be a minimum number of edges added to service all required roads. Also, the service roads must remain in the TR as they are the main reason for the routing problem, so they will be constant between the original graph and the TR.

To remove redundant collections of edges, a simple heuristic was used. Like the TS u-turn minimization approach described in Section 4.3.2, the initial solution can be represented as a sequence of nodes in which the order corresponds to the visiting order. For each node in the solution, cycles can be identified in the total sequence. A cycle may be removed if the following 2 conditions are met.

1. Removing the cycle still yields a strongly connected solution
2. Removing the cycle does not remove any servicing connections

The order in which to do FS-ACO and TR differs route to route. When FS-ACO is used first, additional cycles can be created in the solution that can be removed by the TR afterwards. When TR is used first, the location of unsafe u-turn locations can be altered to a configuration that needs less total u-turn distance found by the FS-ACO. There is no way to distinguish which of these cases apply to a corresponding route, so FS-ACO first then TR, and TR first then FS-ACO will both be applied to the same solution. Since both orders of operation produce feasible routes, the optimal solution was selected as the one requiring the least distance. An example of when TR should be done first can be seen in Figure 4-7 with an improvement of 0.911 km when compared to FS-ACO first. An example of when FS-ACO should be done first can be seen in Figure 4-8 with an improvement of 0.18 km when compared to TR first. In both figures, the green roads represent roads added to resolve u-turns, and the red dashed roads represent roads that can be removed to create the TR.



(a)



(b)

Figure 4-7 FS-ACO First (a), and TR First (0.911 km Improvement) (b)



(a)



(b)

Figure 4-8 FS-ACO First (a) (0.18 km Improvement), and TR First (b)

The pseudo code for the whole sweeping area route optimization process can be seen as follows.

---

**Algorithm 7** Street-sweeping Route Optimization Algorithm

---

**Input:** Street-sweeping area, constraints, optimization parameters for GA, TS, and FS-ACO

**Output:** Several feasible routes servicing all roads in the sweeping area

---

*# Generate Initial Solution of Routes*

**run** 3-phase augment merge algorithm **on** sweeping area

**for** route **in** routes created by 3-phase augment merge algorithm **do**

*# Minimize U-Turns*

**run** modified Hierholzer's algorithm **on** route

**run** TS **on** route

**select** minimal u-turn route

*# Remove Unresolved U-Turns*

**run** FS-ACO + TR **on** minimal u-turn route to remove unresolved u-turns

**run** TR + FS-ACO **on** minimal u-turn route to remove unresolved u-turns

**update** route **with** no u-turn route

**return** feasible routes

---

#### 4.4. Street-sweeping Area Generation Algorithms

Like the waste collection problem, the street sweeper operations rely on pre-assigned street-sweeping areas in the city. Specifically, there are 12 RES sweeping areas, and 5 AC sweeping areas. It is speculated that a better arrangement of street-sweeping areas can be found using advanced clustering methods with respect to minimize the deviation of workload across each area, as well as reducing the overall statistics. Previous methods show that a proposed 2-stage clustering approach can be used to properly balance the workload across areas, but the methods should be improved to reduce the total statistics as well. In this section, the modifications made to the Static and Dynamic clustering approaches used in Section 3.6 will be discussed in detail.

#### 4.4.1. Static Clustering

As previously explained, the Static clustering method makes use of WK-Means and DE to fine tune a single  $\beta$  value used in the weighted distance calculation for all clusters. Previously, the weights were proportional to the dwellings in each cluster. For the street-sweeping problem, the weights will be proportional to the service distance, deadhead travel distance to-and-from the depot, and the approximate number of trips.

In Chapter 3, the depot and waste transfer facility locations were not considered as part of the study, however both the depot and temporary debris storage facility were considered for the street-sweeping problem, and thus this information needed to be incorporated into the clustering methods. First, arbitrary cluster configurations were made for RES and AC roads without considering the depot location and number of trips. Complete simulations using the proposed heuristics in Section 4.3 were made using the cluster configurations for all 3 seasons, and the trip rate was calculated. The trip rate was used as a rough approximation for the number of trips needed in each cluster that is proportional to the amount of servicing distance, it will be incorporated into the new clustering algorithm. The trip rate was calculated by finding the average number of trips needed in each area for all 3 seasons and dividing that by the serviceable distance in the respective areas. Since the topology of the RES roads are denser than AC, different trip rates were calculated. The expected trip rates for RES and AC road classes can be seen in Table 4-3. During clustering, the approximate number of trips will be rounded up to the next whole integer.

Table 4-3 Experimental Trip Rates for RES and AC Road Classes for Arbitrary Clusters

Road Class	Trip Rate (trips/km)	Trip Rate (km/trips)
RES	0.0940	10.64
AC	0.1171	8.54

Initially, the trips from the depot were calculated to the nearest intersection of each respective cluster center, but this was problematic for clusters that had center locations very near to the depot location. In these cases, the depot-to-cluster-center trips were very small, and thus did not accurately reflect the distance needed to travel from the depot to the cluster, causing the clusters near the depot to be very large as seen in Figure 4-9. So, another method of calculating the depot to cluster distance was needed.

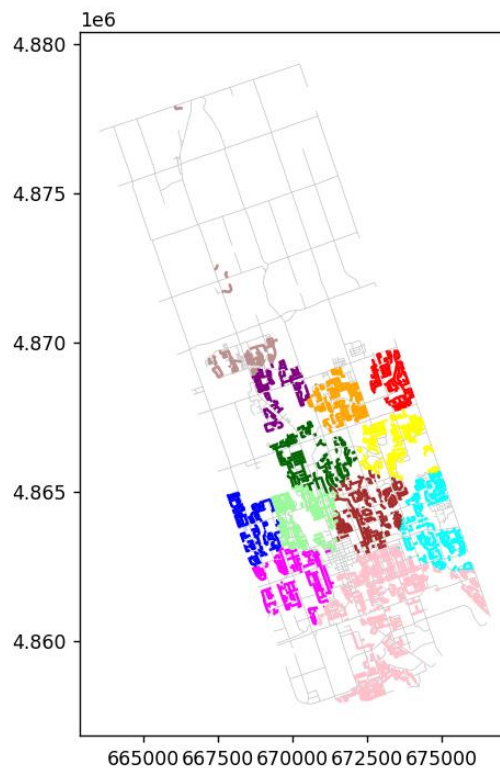


Figure 4-9 Depot-to-Cluster-Center Issue (Pink Area)

To resolve the large clusters near the depot location, a new method of trip distance estimation was used. Each road has a node on either end, so the clusters can be represented as a collection of the road end points. Using the Shapely library in Python, a convex hull was used to create the smallest convex polygon that encapsulates all road end points within



each cluster [87]. Instead of using the cluster center as the estimated trip destination, several critical points along the boundary of the convex hull can be used, thus making use of the entire cluster's area and not just the cluster center. The convex hull representation of clusters can be seen in Figure 4-10.

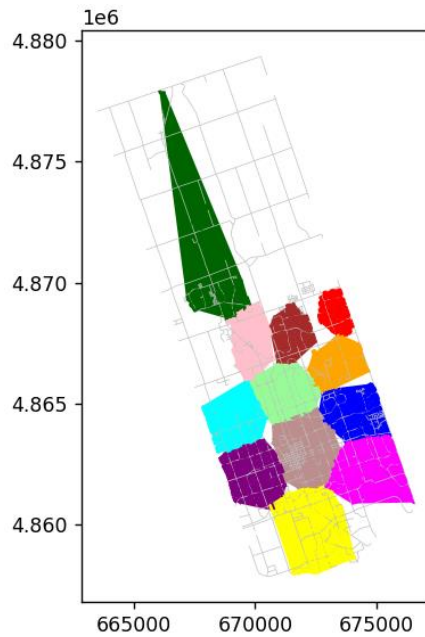


Figure 4-10 Cluster Convex Hulls

The mentioned Static clustering algorithm is used as explained in Section 3.6.2 with some small modifications to incorporate trip distance estimations. In each iteration in the WK-Means clustering, several critical points on the cluster's boundary will be selected, and the distance to and from the depot will be calculated using A\* in the road network. The total depot-to-cluster trip distances will be added to the clusters servicing distance, thus having an influence on the cluster weights in the WK-Means algorithm. By including the estimated number of depot trips in the clustering, sweeping areas further away from the depot should have less servicing distance to account for having to travel a considerably

longer distance to sweep. The opposite can be said for clusters near the depot as they would ideally have more servicing distance because less distance is needed to reach them.

#### **4.4.2. Dynamic Clustering**

The same Dynamic clustering methods used in Section 3.6.3 were used in the street-sweeping application. By using the cluster center locations and weights calculated in the Static clustering process (described in Section 4.4.1), the depot-to-cluster distances have already been included in the initial cluster weighting, so no additional modifications to the Dynamic method were required.

### **4.5. Results & Analysis**

In this section, a complete study of the existing street-sweeping configuration will be compared to the optimized street-sweeping configuration. The objective of this study is to quantify the improvements made using the proposed clustering algorithms with regards to the overall statistics and workload distribution.

#### **4.5.1. Current Configuration Analysis**

The street-sweeping route optimization algorithm discussed in Section 4.3 was used for all 12 RES areas, and all 5 AC areas to generate feasible street-sweeping routes for all 3 seasons. Additionally, a collection of routes was also generated for all canopy roads, but this was ignored for this study as it is constant between the existing and clustered configurations. All the generated routes were feasible with respect to the problems constraints shown in Table 4-1, and operational constraints (u-turns and proper traffic operation on 1-way and 2-way roads). The parameters used in the route optimization algorithm can be found in Table 4-4. Each value in Table 4-4 was selected through experimentation to yield an acceptable solution within an appropriate timeframe.

Additionally, some values in the optimization algorithms were selected to be within the range that is commonly seen for that value, for example the crossover rate (typically near 1) and mutation rate (typically near 0) in the GA.

Table 4-4 Optimization Parameters for the Street-sweeping Route Optimization Algorithm

Parameter	Value
<b>GA</b>	
Runs	5
Generations	150
Population Size	100
Crossover Probability	0.85
Mutation Probability	0.15
<b>TS</b>	
Iterations	100
Tabu Solutions List Length	30
Tabu Nodes List Length	(Number of Cycle Nodes)/2
<b>ACO</b>	
Iterations	100
Number of Ants	100
$\rho$ (Evaporation Rate)	0.25
$\alpha$ (Pheromone Influence)	0.5
$\beta$ (Edge Distance Influence)	0.5
Graph Filter Radius	750 m

Animations, like the one seen in Figure 3-10, were generated for all routes with the inclusion of the depot and temporary debris storage facility locations to show the entire street-sweeping route. Additionally, figures for all routes and route areas were generated depicting the respective service roads and deadhead roads required to complete the route. An example of a route figure and sweeping area figure can be seen in Figure 4-11 (a) and Figure 4-11 (b), respectively.



Figure 4-11 Route Figure (a) and Sweeping Area Figure (b) for Fall

Through experimentation, it was observed that the same routes can be used for Spring and Summer since the debris capacity constraint is not violated for either case, thus the routes can be interchangeable across these seasons. So, the results presented in this section will use the larger debris rate (Spring), and the total debris for Summer can be calculated using the appropriate debris scale that was previously discussed. However, Fall still needs its own set of routes since the debris collection rate is much larger, which requires more trips to the temporary debris storage facility. Accompanying statistic tables for each route and each route area were made, like the one seen in Figure 3-11 for the waste collection optimization. To better visualize the statistic distribution, normalized statistic distribution plots were made for each sweeping area and road class. By normalizing the statistics (dividing each statistic by the maximum respective statistic), they can all be visualized on the same plot. An example of a sweeping area statistic distribution plot can be seen in Figure 4-12 for RES sweeping area 1.

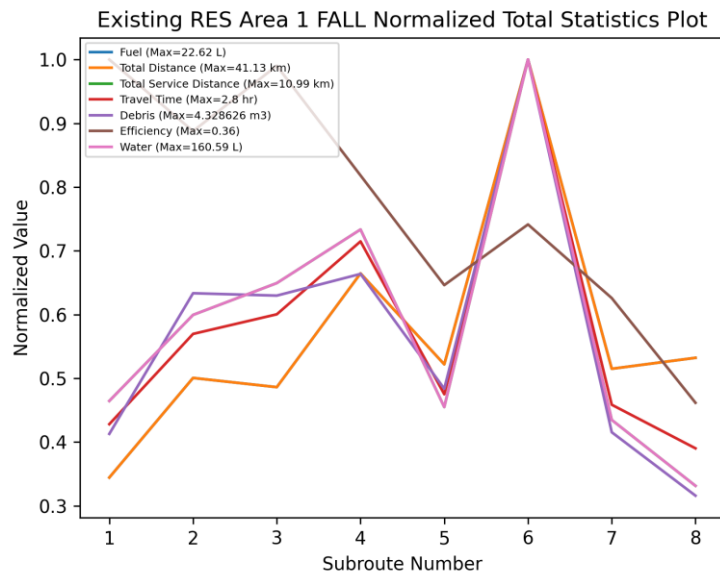


Figure 4-12 Normalized Total Statistic Plot for Routes in Existing Sweeping Area 1 in Fall

The statistical distribution plots generated for routes within the sweeping areas provide good insight on the micro scale, but the bigger picture is still missing. To analyze the street-sweeping problem on the macro scale, the total area statistics should be examined for each respective road class. Table 4-5 and Table 4-6 summarize the overall statistics for the current sweeping area configuration for Fall and Spring/Summer, respectively.

Table 4-5 Fall Statistics for the Current Sweeping Areas

Metric	Fuel (L)	Distance (km)	Time (hr)	Debris (m <sup>3</sup> )	Water (L)	Service Distance (km)	Efficiency (%)
<b>AC</b>							
<b>Total</b>	497.96	905.37	63.52	88.93	3709.18	253.88	28.04
<b>Average</b>	99.59	181.07	12.70	17.79	741.84	50.78	27.90
<b>Deviation</b>	13.61	24.75	2.88	4.86	202.20	13.84	5.57
<b>RES</b>							
<b>Total</b>	1500.14	2727.53	179.08	256.59	10122.75	692.86	25.40
<b>Average</b>	125.0	227.29	14.92	21.38	843.56	57.74	24.73
<b>Deviation</b>	62.75	114.10	7.53	10.90	426.25	29.18	6.43

Table 4-6 Spring/Summer Statistics for the Current Sweeping Areas

Metric	Fuel (L)	Distance (km)	Time (hr)	Debris (m <sup>3</sup> )	Water (L)	Service Distance (km)	Efficiency (%)
<b>AC</b>							
<b>Total</b>	692.96	1259.94	104.88	77.81	6496.10	444.63	35.29
<b>Average</b>	138.59	251.99	20.97	15.56	1299.22	88.93	35.92
<b>Deviation</b>	18.35	33.38	2.08	2.01	167.48	11.46	7.83
<b>RES</b>							
<b>Total</b>	1460.50	2655.46	177.77	97.00	10122.75	692.86	26.09
<b>Average</b>	121.71	221.29	14.81	8.08	843.56	57.74	25.63
<b>Deviation</b>	62.12	112.95	7.51	4.08	426.25	29.18	7.06

The statistics in Table 4-5 and Table 4-6 can be visualized in a normalized statistic distribution plot like Figure 4-12. To normalize the values, each statistic was divided by the respective statistic's maximum. In Figure 4-13 and Figure 4-14, the normalized statistics for RES and AC sweeping areas can be seen for Fall and Spring/Summer.

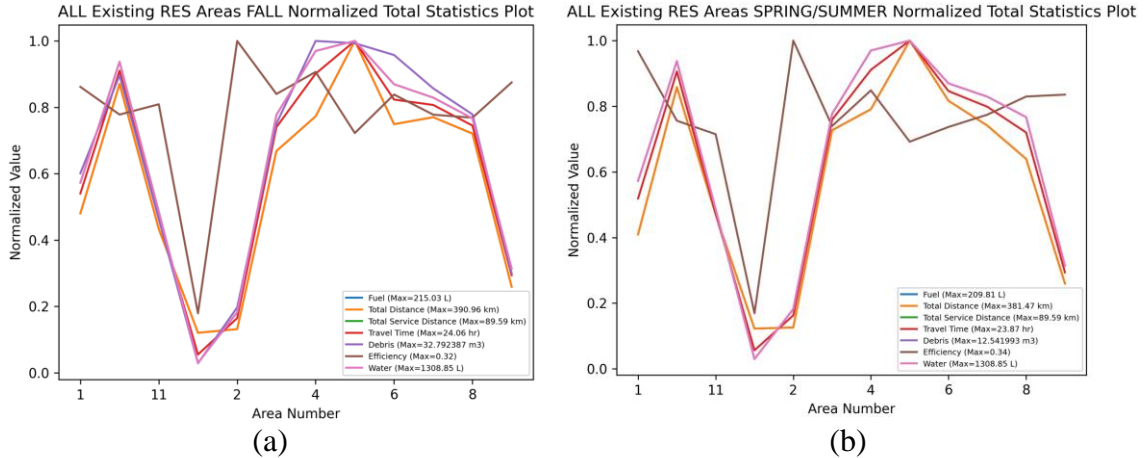


Figure 4-13 Normalized Total Statistic Plots for the Existing RES Areas in Fall (a) and Spring/Summer (b)

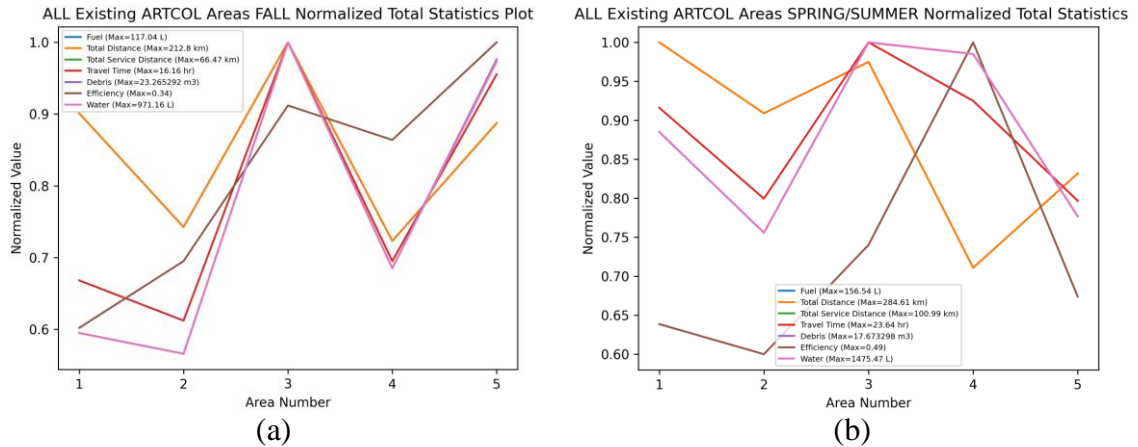


Figure 4-14 Normalized Total Statistic Plots for the Existing AC Areas in Fall (a) and Spring/Summer (b)

Figure 4-13 and Figure 4-14 highlight the statistic deviation seen in Table 4-5 and Table 4-6 for the respective road class and season, most notably the RES area 15 area for all seasons. Ideally, the deviation of statistics should be minimized in a properly assigned sweeping area configuration. This represents a properly balanced workload. Also, the overall statistics should be reduced, except for efficiency (calculated in Eq. (4-6)). The results from the current street-sweeping configuration will be used as a benchmark to quantify the improvements made by properly assigning the street-sweeping areas through the 2-stage cluster approach.

#### 4.5.2. Clustered Configuration Analysis

Using the clustering methods discussed in Section 4.4.1 and 4.4.2, the new street-sweeping areas can be assigned. As discussed, the street-sweeping Static and Dynamic clustering will differ from the waste collection approach since approximate trips to, and from, the depot will be included. Since the convex hull operation [87] explained in Section 4.4.1 was needed in each iteration of WK-Means clustering, the computation time will increase significantly, thus not as many WK-Means runs can be used. The parameters for

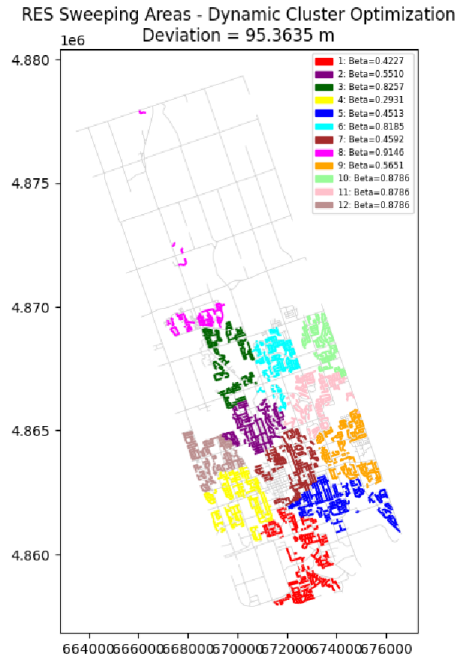
the Static and Dynamic clustering can be found in Table 4-7. In Table 4-7, the population size is dependent on the number of CPUs available. Since each CPU will be computing a complete WK-Means algorithm for the corresponding weight influence factor(s), the execution time is reduced by assigning each member of the population to a single CPU. In doing so, an entire population can be computed in parallel. If more CPUs are available, a larger population size may be used which may yield better solutions.

Table 4-7 Static and Dynamic Cluster Optimization Parameters for Street-sweeping

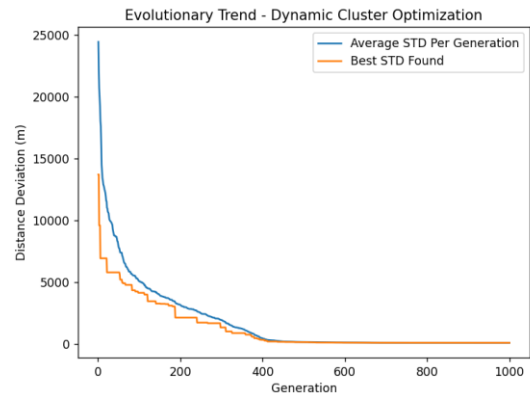
Parameter	Value	
	Static Clustering	Dynamic Clustering
Number of Control Variables ( $\beta$ )	1	10
Population Size (Number of CPUs)	32	32
Crossover Rate	1	0.75
Mutation Rate	Linearly decreasing (0.8-0.05)	Linearly decreasing (0.8-0.05)
Generations	100	1000
Clusters	12 (RES)	12 (RES)
	5 (AC)	5 (AC)
WK-Means Runs	1	/
WK-Means Stop Iteration	200	/
Run Time	$\approx$ 14 hr (RES)	$\approx$ 2 hr (RES)
	$\approx$ 10 hr (AC)	$\approx$ 0.5 hr (AC)

The final clusters produced by the Dynamic clustering for RES and AC road classes can be seen in Figure 4-15 and Figure 4-16, where the distance deviation is 95.36 m and 88.98 m, respectively. The evolutionary plots for RES and AC show an exponentially decreasing trend, which highlights the effectiveness of the proposed clustering techniques. However, in both cases the solutions converge early in the evolutionary process (about generation 500 for RES and 200 for AC), so the remaining time spent exploring the solution space was not needed.



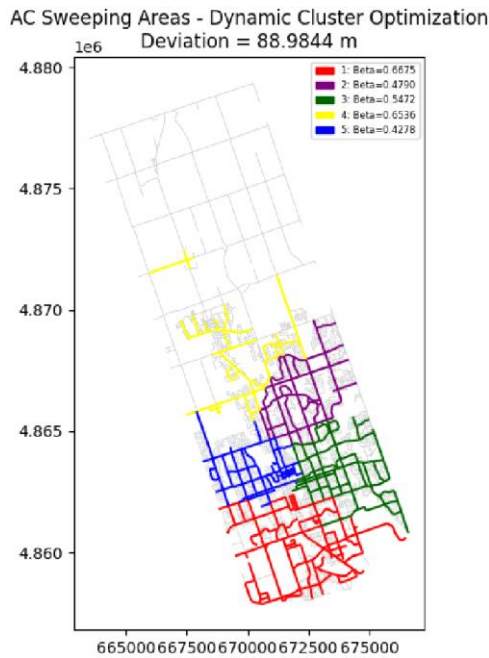


(a)

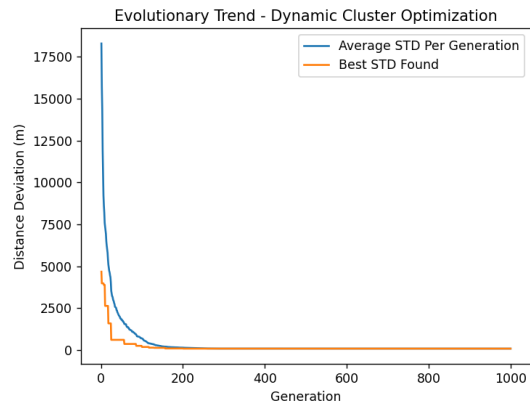


(b)

Figure 4-15 RES Clusters Produced by the Dynamic Cluster Optimization Process (a), and the Evolutionary Improvement Trend (b)



(a)



(b)

Figure 4-16 AC Clusters Produced by the Dynamic Cluster Optimization Process (a), and the Evolutionary Improvement Trend (b)

Each of the clustered areas were used as an input to the street-sweeping route optimization discussed in Section 4.3 to generate feasible route simulations. Like the current area configuration, route and area plots area made for the clustered configuration to distinguish between deadhead and serviceable roads. Additionally, animations and route specific statistic tables were made. As the clustered areas are different than the existing configuration, a side-by-side comparison of each area will not be sufficient to quantify the improvements made. Instead, the overall statistics for each road class were compared for the respective seasons in the clustered sweeping areas. Table 4-8 and Table 4-9 show the total statistics for the clustered configuration areas for Fall and Spring/Summer, respectively.

Table 4-8 Fall Statistics for the Clustered Sweeping Areas

<b>Metric</b>	<b>Fuel (L)</b>	<b>Distance (km)</b>	<b>Time (hr)</b>	<b>Debris (m<sup>3</sup>)</b>	<b>Water (L)</b>	<b>Service Distance (km)</b>	<b>Efficiency (%)</b>
<b>AC</b>							
<b>Total</b>	491.64	893.90	63.34	89.03	3713.15	254.15	28.43
<b>Average</b>	98.32	178.78	12.66	17.81	742.63	50.83	28.36
<b>Deviation</b>	8.46	15.38	2.58	4.75	198.32	13.57	6.78
<b>RES</b>							
<b>Total</b>	1456.5 1	2648.19	177.5 4	256.67	10126.19	693.10	26.17
<b>Average</b>	121.38	220.68	14.79	21.39	843.85	57.76	26.04
<b>Deviation</b>	25.26	45.93	3.35	5.39	197.40	13.51	2.09

Table 4-9 Spring/Summer Statistics for the Clustered Sweeping Areas

Metric	Fuel (L)	Distance (km)	Time (hr)	Debris (m <sup>3</sup> )	Water (L)	Service Distance (km)	Efficiency (%)
<b>AC</b>							
<b>Total</b>	659.17	1198.50	103.75	77.86	6500.07	444.91	37.12
<b>Average</b>	131.83	239.70	20.74	15.57	1300.01	88.98	36.78
<b>Deviation</b>	15.28	27.79	5.32	4.73	394.90	27.03	7.90
<b>RES</b>							
<b>Total</b>	1449.91	2636.19	177.28	97.03	10126.19	693.10	26.29
<b>Average</b>	120.83	219.68	14.77	8.09	843.85	57.76	26.40
<b>Deviation</b>	29.22	53.14	3.43	1.89	197.40	13.51	3.10

The statistics in Table 4-8 and Table 4-9 can be visualized in normalized statistic plots for the respective road classes. The distribution plots can be seen in Figure 4-17 and Figure 4-18 for the RES and AC areas, respectively.

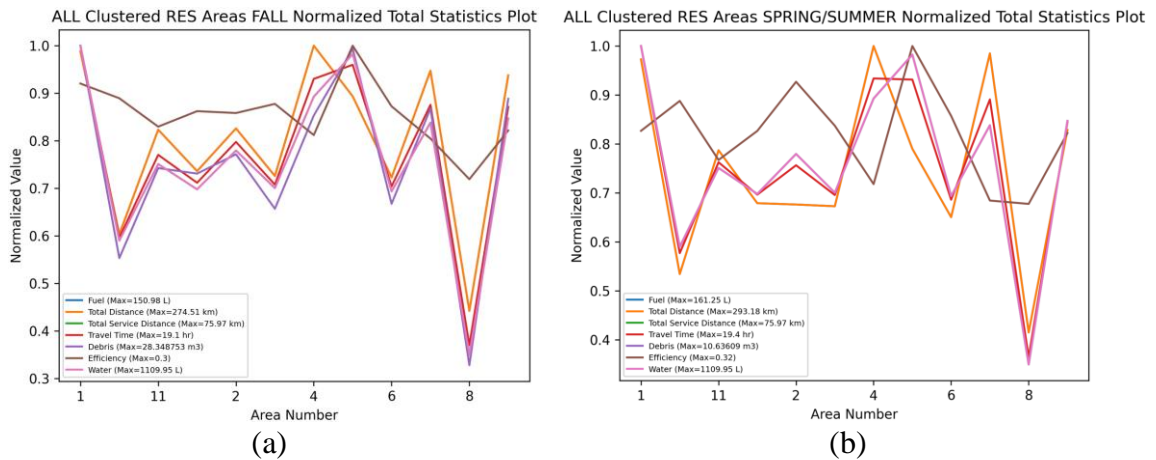


Figure 4-17 Normalized Total Statistic Plots for the Clustered RES Areas in Fall (a) and Spring/Summer (b)

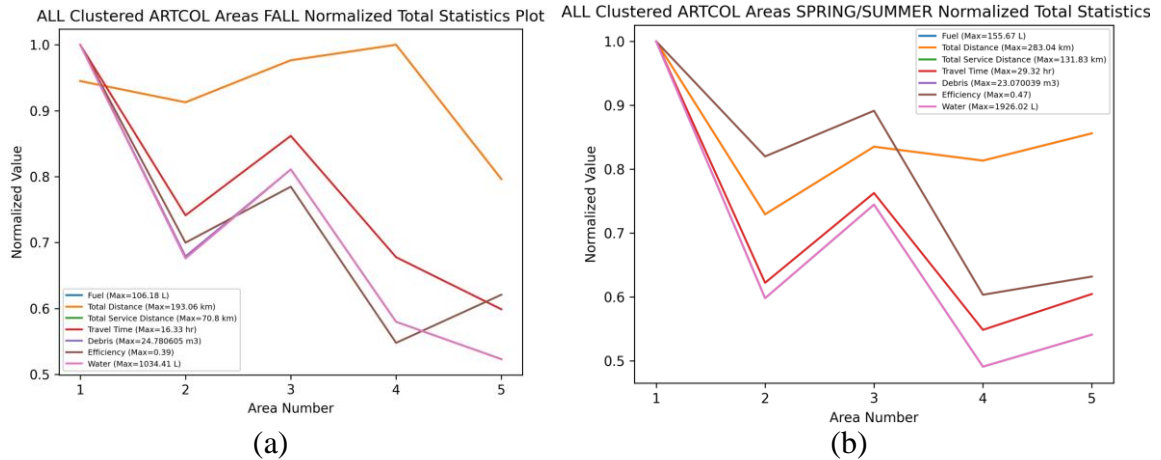


Figure 4-18 Normalized Total Statistic Plots for the Clustered AC Areas in Fall (a) and Spring/Summer (b)

Examining the plots in Figure 4-17, the normalized statistic values fluctuate around 0.8. Compared to Figure 4-13 (where the values fluctuate around 0.75), an increase in about 0.05 is seen. This reflects the improvement of the workload distribution, where ideally, all sweeping areas would have the exact same statistics (normalized value of 1). If all statistics had a normalized value of 1, this means that all statistics would be equal across each sweeping area. However, examining Figure 4-18 (a), a decreasing trend in the normalized values is seen from 1 to about 0.6. This is comparable to the existing AC configuration seen in Figure 4-14 (a). The same decreasing trend is seen in Spring/Summer (Figure 4-18 (b)), except the normalized statistics extend just below 0.6. Additionally, the normalized efficiency values (brown line in Figure 4-18 (a)) for AC roads are smaller in Fall than Spring/Summer. This is because clustering was performed on all AC roads, and in the Fall, the regional roads are not swept. As explained in Section 1.4.1, the AC road class is comprised of city owned, and regional owned roads. Since clustering was performed on all AC roads, the regional and city roads were included. Since the regional roads are swept in 2/3 sweeping seasons, they were included in the clustering. Ideally, different cluster

configurations should be used for AC roads in Spring/Summer and Fall because regional roads are not included in Fall. Having different sweeping areas for different seasons may cause confusion, specifically, when switching from one area configuration to another after a season ends. To standardize the sweeping areas, regional roads were included in the clustering as they are swept in most of the sweeping seasons. As a result, when the regional roads are removed from the AC clusters in Fall, the previously well-distributed clusters may become less efficient.

To complete the analysis of the clustered sweeping area configuration, a comparison of the total statics will be made. The improvements of the proposed clustered sweeping areas can be summarized in Table 4-10 and Table 4-11 for Fall and Spring/Summer, respectively.

Table 4-10 Improvements Made Using the Clustered Configuration in Fall

<b>Metric</b>	<b>Fuel (L)</b>	<b>Distance (km)</b>	<b>Time (hr)</b>	<b>Efficiency (%)</b>
<b>AC</b>				
<b>Total</b>	6.31	11.48	0.18	0.39
<b>Average</b>	1.26	2.30	0.04	0.45
<b>Deviation</b>	5.15	9.37	0.30	-1.21
<b>RES</b>				
<b>Total</b>	43.64	79.34	1.54	0.77
<b>Average</b>	3.64	6.612	0.13	1.30
<b>Deviation</b>	37.49	68.17	4.18	4.34

Table 4-11 Improvements Made Using the Clustered Configuration in Spring/Summer

<b>Metric</b>	<b>Fuel (L)</b>	<b>Distance (km)</b>	<b>Time (hr)</b>	<b>Efficiency (%)</b>
<b>AC</b>				
<b>Total</b>	33.79	61.44	1.14	1.83
<b>Average</b>	6.76	12.29	0.23	0.85
<b>Deviation</b>	3.07	5.58	-3.24	-0.07
<b>RES</b>				
<b>Total</b>	10.60	19.27	0.49	0.20
<b>Average</b>	0.88	1.61	0.04	0.77
<b>Deviation</b>	32.90	59.81	4.07	3.95

Examining Table 4-10 and Table 4-11, almost all the metrics have been improved. Only three metrics were not improved, the deviation of efficiency (service distance divided by total distance for each area) for AC areas in Fall and Spring/Summer, and deviation of travel time for AC areas in Spring/Summer. The least improved arrangement of sweeping areas is the AC configuration for Fall. As previously discussed, this was because residential roads are not swept in Fall but were included in the clustering process. Overall, the proposed clustering methods were able to improve almost every aspect of the existing sweeping area configuration.

#### **4.6. Conclusions**

In this chapter, a complete analysis of the existing street-sweeping area configuration was conducted in The City of Oshawa for RES and AC roads in all operating seasons. To do so, a novel street-sweeping route optimization algorithm was developed using a combination of several different heuristic methods. The proposed algorithm could divide a street-sweeping area into several feasible routes while satisfying multiple real-world constraints. Additionally, the FS-ACO algorithm was developed to eliminate any unsafe u-turns that could not be resolved using the modified Hierholzer's algorithm or TS.

The Static clustering approach was improved by considering the approximate number of trips needed to a cluster proportional to its servicing distance. Using A\*, the depot-to-cluster distance can be added to the cluster weight for several critical points along the cluster boundary. In doing so, clusters near the depot are larger than ones far from the depot; this reflects the travel distance required to reach the cluster. The new clustering methods were used to generate street-sweeping areas in The City of Oshawa, and the proposed street-sweeping route optimization algorithm was used to validate the new clustering methods.

The results in Section 4.5.2 show considerable improvements with respect to the overall statistics and statistic distribution.

## Chapter 5. Conclusion

### 5.1. Conclusions

Route optimization is a part of everyday life. Sometimes routing problems have simple solutions, such as finding the shortest path from point a to point b. But real-world applications in routing have many constraints that make a simple trip from point a to b less obvious. In this thesis, two real-world routing problems were theoretically optimized for The City of Oshawa, waste collection and street-sweeping. Waste collection and street-sweeping play an important role in the cleanliness, health, and overall safety of the public. However, these operations require a great deal of planning and resources for an efficient implementation.

Sometimes, a large problem needs to be separated into several smaller problems. Many routing applications make use of the macro and micro analysis, waste collection is no different. On a macro scale, a large road network can be separated into several smaller areas. On the micro scale, the smaller areas can be divided into several routes. To achieve such area division, a novel 2-stage clustering algorithm was developed, the Static and Dynamic clustering algorithm. The proposed clustering methods use DE to fine tune the weight influence factor of the WK-Means clustering algorithm such that the deviation of weights are minimized. The weights of the clusters are specific to the application. In waste collection, the weights are proportional to the number of dwellings within each area. In street-sweeping the weights are proportional to the amount of road to service.

The proposed 2-stage clustering algorithm was used in The City of Oshawa to generate new waste collection areas. To validate the performance of the clustering, the routes within each cluster were simulated to generate statistics. A waste collection route optimization



algorithm was developed using Dijkstra's algorithm and Hierholzer's algorithm to find the minimum distance path that services all roads in a route. Several statistics were generated, and a dimensionless objective function value was used to quantify the fitness of a route. The dimensionless objective function values for the routes in each respective area were summed to quantify the fitness of a collection area. Using a GA, a workload balancing algorithm was developed that can pair collection areas together for each day of the week with the objective of minimizing the deviation of the expected workload. The results highlighted that the clustered areas offer a better workload distribution than any combination of existing areas. However, the balanced workload is at the expense of a slight increase in the total statistics. Specifically, a small increase in distance, fuel, and travel time was seen. This can be justified since the added statistics relate to the inclusion of safe turn-around locations. However, the safe turn-around locations were added manually, and thus were not optimal, highlighted by the increased statistics. Leading to the conclusion that the proposed routing algorithm can be improved with regards to optimally selecting safe turn-around locations.

To achieve a proper workload balance and reduce the total statistics, the clustering methods needed to be improved. The street-sweeping operations within The City of Oshawa also divide the road network into several sweeping areas. The proposed clustering methods were improved by including the depot-to-cluster trip distance for the approximate number of trips needed. To divide the area into several feasible routes, a novel street-sweeping route optimization algorithm was developed. The algorithm makes use of the 3-phase augment merge algorithm, GA, Hierholzer's algorithm, TS, and TR. The proposed algorithm satisfies all vehicle constraints (fuel capacity, water capacity, and debris

capacity) and operational constraints (shift length, proper travel direction and removal of u-turns) and resolves all unsafe u-turns using the developed FS-ACO algorithm. An analysis of the current and clustered street-sweeping area configurations was conducted, and the results show that the clustered configuration was able to improve the workload distribution across all areas and reduce the total statistics. Specifically, the routes were able to improve the overall fuel, distance, travel time, and efficiency for RES and AC sweeping areas in all operating seasons. In all routes, the proposed u-turn removal algorithm was able to find the optimal turn-around location for all unsafe u-turns. The sweeping areas made using the clustering algorithm were improved with respect to the statistical balance when compared to the existing sweeping areas. Results show the deviation of fuel, distance, travel time, and efficiency were improved (with some small exceptions that were discussed).

In conclusion, the proposed 2-stage clustering approach for road network division offers a novel method to divide municipalities with the objective of minimizing the deviation of a specific attribute. For routing applications, the proposed clustering method can generate servicing areas that minimize the statistics deviation and total statistics.

## **5.2. Recommendations and Future Work**

The works discussed in this thesis can be expanded upon by implementing smart routing technology. For waste collection route optimization, smart routing technology can be used to estimate the approximate amount of curbside waste that will be collected on a service road. In doing so, the waste collection rate for each road can be unique, and appropriate algorithms will be able to incorporate this information during optimization. In street-sweeping, smart routing applications may include the use of drones to monitor the amount of debris on roads. If the debris on a road reaches a certain threshold, it will be included in

a route for sweeping. If the debris of a road is below a certain threshold, it will not be swept, thus increasing the efficiency of a route.

In the waste collection problem, seasonal specific variations were not considered. For example, tree collection around Christmas time, or yard waste collection in Spring/Summer. Also, the depot and waste storage facility were not included in this study. For the street-sweeping problem, an updated tree canopy information study should also be conducted. As mentioned, the data used for the tree canopy information was collected in 2018, thus more dense tree canopies may exist because of tree growth.

For both the waste collection and street-sweeping, all data presented in this report is based upon simulated data. Future work can include the possibility of implementing the proposed solutions and modifying them based upon real-world data.

Additionally, the performance of the proposed clustering algorithms can be improved. Examining Table 4-7, the computation time for the RES road class (2309 roads) was approximately 14 hr, thus any reductions to the computation time will be beneficial.

## REFERENCES

- [1] The City of Oshawa, “Waste collection calendar & information guide,” 2022, [Online]. Available: [https://www.oshawa.ca/en/home-property/resources/Documents/2022\\_2023\\_Waste\\_Calendar\\_FINAL\\_ACCESSIBLE.pdf](https://www.oshawa.ca/en/home-property/resources/Documents/2022_2023_Waste_Calendar_FINAL_ACCESSIBLE.pdf).
- [2] Elgin, “Elgin whirlwind,” 2017, [Online]. Available: [https://www.elginsweeper.com/hubfs/Elgin\\_August2020/Docs/WhirlwindBrochure.pdf](https://www.elginsweeper.com/hubfs/Elgin_August2020/Docs/WhirlwindBrochure.pdf).
- [3] QGIS, “QGIS,” 2022, [Online]. Available: <https://qgis.org/en/site/>.
- [4] Esri, “What is GIS?,” 2023, [Online]. Available: [https://www.esri.com/en-us/what-is-gis/overview#:~:text=A geographic information system \(GIS,what things are like there\)](https://www.esri.com/en-us/what-is-gis/overview#:~:text=A geographic information system (GIS,what things are like there)).
- [5] B. Golden and R. Wong, “Capacitated arc routing problems,” *Networks*, vol. 11, pp. 305–315, 1981.
- [6] G. B. Dantzig and J. H. Ramser, “The truck dispatching problem,” *Manage. Sci.*, vol. 6, no. 1, pp. 80–91, 1959.
- [7] E. W. Dijkstra, “A note on two problems in connexion with graphs,” *Numer. Math.*, vol. 271, pp. 269–271, 1959.
- [8] P. E. Hart, N. J. Nilsson, and B. Raphael, “Formal basis for the heuristic determination of minimum cost paths,” *IEEE Trans. Syst. Sci. Cybern.*, vol. 4, no. 2, pp. 100–107, 1968, doi: 10.1109/TSSC.1968.300136.
- [9] M. Dorigo and T. Stützle, “Ant Colony Optimization.” The MIT Press, Jun. 04, 2004, doi: 10.7551/mitpress/1290.001.0001.
- [10] C. Hierholzer and C. Wiener, “Ueber die möglichkeit, einen linienzug ohne wiederholung und ohne unterbrechung zu umfahren,” *Math. Ann.*, vol. 6, no. 1, pp. 30–32, 1873, doi: 10.1007/BF01442866.
- [11] J. H. Holland, “Genetic algorithms and the optimal allocation of trials,” *SIAM J. Comput.*, vol. 2, no. 2, pp. 88–106, 1973, doi: 10.7551/mitpress/1090.003.0008.
- [12] R. Storn and K. Price, “Differential evolution - A simple and efficient heuristic for global optimization over continuous spaces,” *J. Glob. Optim.*, vol. 11, no. 4, pp. 341–359, 1997, doi: 10.1023/A:1008202821328.
- [13] F. Glover, “Future paths for integer programming and links to artificial intelligence,” *Comput. Oper. Res.*, vol. 13, no. 5, pp. 533–549, 1986, doi: 10.1016/0305-0548(86)90048-1.
- [14] A. V. Aho, M. R. Garey, and J. D. Ullman, “The transitive reduction of a directed graph,” *SIAM J. Comput.*, vol. 1, no. 2, pp. 131–137, 1972, doi: 10.1137/0201008.
- [15] J. MacQueen, “Some methods for classification and analysis of multivariate

- observations,” *Proc. Fifth Berkeley Symp. Math. Stat. Probab.*, vol. 1, pp. 281–297, 1967, [Online]. Available: [https://digitalassets.lib.berkeley.edu/math/ucb/text/math\\_s5\\_v1\\_article-17.pdf](https://digitalassets.lib.berkeley.edu/math/ucb/text/math_s5_v1_article-17.pdf).
- [16] J. Z. Huang, M. K. Ng, H. Rong, and Z. Li, “Automated variable weighting in k-means type clustering,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 657–668, 2005, doi: 10.1109/TPAMI.2005.95.
- [17] H. De Sterck, R. Markel, and R. Knight, “A lightweight, scalable grid computing framework for parallel bioinformatics applications,” *Proc. - Int. Symp. High Perform. Comput. Syst. Appl.*, pp. 251–257, 2005, doi: 10.1109/HPCS.2005.7.
- [18] Sharcnet, “Cluster graham.sharcnet.ca,” 2022, [Online]. Available: <https://www.sharcnet.ca/my/systems/show/114>.
- [19] H. Chen, T. Cheng, and J. Shawe-Taylor, “A balanced route design for min-max multiple-depot rural postman problem (MMMDRPP): A police patrolling case,” *Int. J. Geogr. Inf. Sci.*, vol. 32, no. 1, pp. 169–190, 2018, doi: 10.1080/13658816.2017.1380201.
- [20] W. Pearn, A. Assad, and B. L. Golden, “Transforming arc routing into node routing problems,” *Comput. Oper. Res.*, vol. 14, no. 4, pp. 285–288, 1987, doi: 10.1016/0305-0548(87)90065-7.
- [21] M. M. Flood, “The travelling-salesman problem,” *Oper. Res.*, vol. 4, no. 1, pp. 61–75, 1956, doi: 10.1287/opre.4.1.61.
- [22] R. Nemani, N. Cherukuri, G. R. K. Rao, P. V. V. S. Srinivas, J. J. Pujari, and C. Prasad, “Algorithms and optimization techniques for solving TSP,” in *Proceedings of the 5th International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud), I-SMAC 2021*, 2021, pp. 809–814, doi: 10.1109/I-SMAC52330.2021.9640907.
- [23] S. Irnich, P. Toth, and D. Vigo, “Chapter 1: The family of vehicle routing problems,” in *Vehicle Routing*, 2014, pp. 1–33.
- [24] D. W. S. Wang, L. Wang, “Particle swarm optimization for multi-depot single vehicle routing problem,” in *2008 Chinese Control and Decision Conference*, 2008, pp. 4659–4661, doi: 10.1109/CCDC.2008.4598213.
- [25] B. K. L. K. Hwan Kang, Y. Hoon Lee, “An exact algorithm for multi depot and multi period vehicle scheduling problem,” in *Computational Science and Its Applications – ICCSA 2005*, 2005, pp. 350–359, doi: 10.1007/11424925\_38.
- [26] A. Mingozzi, “The multi-depot periodic vehicle routing problem,” in *Abstraction, Reformulation and Approximation*, 2005, pp. 347–350, doi: 10.1007/11527862\_27.
- [27] M. Polacek, R. F. Hartl, K. Doerner, and M. Reimann, “A variable neighborhood search for the multi depot vehicle routing problem with time windows,” *J. Heuristics*, vol. 10, no. 6, pp. 613–627, 2004, doi: 10.1007/s10732-005-5432-5.
- [28] K. Mei-Ko, “Graphic programming using odd or even points,” *Chinese Math*, vol.

- 1, pp. 237–277, 1962.
- [29] H. Thimbley, “The directed Chinese postman problem,” *Softw Pr. Exper.*, vol. 33, no. 11, pp. 1081–1096, 2003, doi: 10.1002/spe.540.
- [30] M. K. Gordenko and S. M. Avdoshin, “The variants of Chinese postman problems and way of solving through transformation into vehicle routing problems,” *Proc. Inst. Syst. Program. RAS*, vol. 30, no. 3, pp. 221–232, 2018, doi: 10.15514/ispras-2018-30(3)-16.
- [31] H. Robinson, “Graph theory techniques in model-based testing,” in *International Conference on Testing Computer Software*, 1999, vol. 1, no. March, pp. 1–10, [Online]. Available: [https://www.researchgate.net/profile/Harry-Robinson-5/publication/228767978\\_Graph\\_theory\\_techniques\\_in\\_model-based\\_testing/links/551ac30f0cf251c35b4f5054/Graph-theory-techniques-in-model-based-testing.pdf](https://www.researchgate.net/profile/Harry-Robinson-5/publication/228767978_Graph_theory_techniques_in_model-based_testing/links/551ac30f0cf251c35b4f5054/Graph-theory-techniques-in-model-based-testing.pdf).
- [32] J. Edmonds and E. L. Johnson, “Matching, Euler tours and the Chinese postman,” *Math. Program.*, vol. 5, no. 1, pp. 88–124, 1973, doi: 10.1007/BF01580113.
- [33] P. Lacomme, C. Prins, and W. Ramdane-Cherif, “A genetic algorithm for the capacitated arc routing problem and its extensions,” in *Applications of Evolutionary Computing. EvoWorkshops 2001. Lecture Notes in Computer Science*, vol. 2037, E. J. W. Boers, Ed. Berlin, Heidelberg: Springer, 2001, pp. 473–483.
- [34] E. B. Tirkolaei, I. Mahdavi, and M. Mehdi Seyyed Esfahani, “A robust periodic capacitated arc routing problem for urban waste collection considering drivers and crew’s working time,” *Waste Manag.*, vol. 76, pp. 138–146, 2018, doi: 10.1016/j.wasman.2018.03.015.
- [35] R. M. K. E. B. Tirkolaei, A. Goli, M. Pahlevan, “A robust bi-objective multi-trip periodic capacitated arc routing problem for urban waste collection using a multi-objective invasive weed optimization,” *Waste Manag. Res.*, vol. 37, no. 11, pp. 1089–1101, 2019, doi: 10.1177/0734242X19865340.
- [36] M. C. Mourão and L. Amado, “Heuristic method for a mixed capacitated arc routing problem: A refuse collection application,” *Eur. J. Oper. Res.*, vol. 160, no. 1, pp. 139–153, 2005, doi: 10.1016/j.ejor.2004.01.023.
- [37] E. Mofid-Nakhaei and F. Barzinpour, “A multi-compartment capacitated arc routing problem with intermediate facilities for solid waste collection using hybrid adaptive large neighborhood search and whale algorithm,” *Waste Manag. Res.*, vol. 37, no. 1, pp. 38–47, 2019, doi: 10.1177/0734242X18801186.
- [38] R. B. Lopes, F. Plastria, C. Ferreira, and B. S. Santos, “Location-arc routing problem: Heuristic approaches and test instances,” *Comput. Oper. Res.*, vol. 43, pp. 309–317, 2014, doi: 10.1016/j.cor.2013.10.003.
- [39] G. Ghiani and G. Laporte, “Location-arc routing problems,” *Opsearch*, vol. 38, no. 2, pp. 151–159, 2001, doi: 10.1007/BF03399222.

- [40] T. L. Yang, “A tabu search , augment-merge heuristic to solve the stochastic location arc routing problem,” University of Arkansas, 2016.
- [41] L. D. Bodin and J. K. Samuel, “A computer-assisted system for the routing and scheduling of street sweepers,” *Oper. Res.*, vol. 26, no. 4, pp. 525–537, 1978, doi: 10.1287/opre.26.4.525.
- [42] R. W. Eglese and H. Murdock, “Routeing road sweepers in a rural area,” *J. Oper. Res. Soc.*, vol. 42, no. 4, pp. 281–288, 1991, doi: 10.1057/jors.1991.66.
- [43] C. A. Blazquez, A. Beghelli, and V. P. Meneses, “A novel methodology for determining low-cost fine particulate matter street sweeping routes,” *J. Air Waste Manag. Assoc.*, vol. 62, no. 2, pp. 242–251, 2012, doi: 10.1080/10473289.2011.646048.
- [44] A. Rasul, J. Seo, S. Xu, T. J. Kwon, J. MacLean, and C. Brown, “Optimization of snowplow routes for real-world conditions,” *Sustain.*, vol. 14, no. 20, pp. 1–17, 2022, doi: 10.3390/su142013130.
- [45] S. Xu and T. J. Kwon, “Optimizing snowplow routes using all-new perspectives: Road users and winter road maintenance operators,” *Can. J. Civ. Eng.*, vol. 48, no. 8, pp. 959–968, 2021, doi: 10.1139/cjce-2019-0768.
- [46] B. Dussault, B. Golden, C. Groër, and E. Wasil, “Plowing with precedence: A variant of the windy postman problem,” *Comput. Oper. Res.*, vol. 40, no. 4, pp. 1047–1059, 2013, doi: 10.1016/j.cor.2012.10.013.
- [47] F. M. Assef, M. T. A. Steiner, and E. P. de Lima, “A review of clustering techniques for waste management,” *Heliyon*, vol. 8, no. 1, p. e08784, 2022, doi: 10.1016/j.heliyon.2022.e08784.
- [48] C. lin Xin, S. Liang, and F. wu Shen, “Reconfiguration of garbage collection system based on Voronoi graph theory: a simulation case of Beijing region,” *J. Comb. Optim.*, vol. 43, no. 5, pp. 953–973, 2020, doi: 10.1007/s10878-020-00614-z.
- [49] A. Al-Refaie, A. Al-Hawadi, and S. Fraij, “Optimization models for clustering of solid waste collection process,” *Eng. Optim.*, vol. 53, no. 12, pp. 2056–2069, 2021, doi: 10.1080/0305215X.2020.1843165.
- [50] Z. Wei, C. Liang, and H. Tang, “A cross-regional scheduling strategy of waste collection and transportation based on an improved hierarchical agglomerative clustering algorithm,” *Comput. Intell. Neurosci.*, vol. 2022, pp. 412611–412617, 2022, doi: 10.1155/2022/7412611.
- [51] P. C. Pop, L. Fuksz, A. H. Marc, and C. Sabo, “A novel two-level optimization approach for clustered vehicle routing problem,” *Comput. Ind. Eng.*, vol. 115, no. November 2017, pp. 304–318, 2018, doi: 10.1016/j.cie.2017.11.018.
- [52] Y. Zheng, G. Zhao, and J. Liu, “A novel grid based k-means cluster method for traffic zone division,” in *Cloud Computing and Big Data. CloudCom-Asia 2015. Lecture Notes in Computer Science*, vol. 9106, W. Qiang, X. Zheng, and C. HSU,

Eds. Springer, Cham, 2015, pp. 165–178.

- [53] S. Soor, A. Challa, S. Danda, B. S. D. Sagar, and L. Najman, “Iterated watersheds, a connected variation of K-Means for clustering GIS data,” *IEEE Trans. Emerg. Top. Comput.*, vol. 9, no. 2, pp. 626–636, 2021, doi: 10.1109/TETC.2019.2910147.
- [54] G. K. Shyam, S. S. Manvi, and P. Bharti, “Smart waste management using internet-of-things (IoT),” in *2017 2nd International Conference on Computing and Communications Technologies (ICCCCT)*, 2017, pp. 199–203, doi: 10.35940/ijitee.g5334.078919.
- [55] M. A. Hannan, M. Akhtar, R. A. Begum, H. Basri, A. Hussain, and E. Scavino, “Capacitated vehicle-routing problem model for scheduled solid waste collection and route optimization using PSO algorithm,” *Waste Manag.*, vol. 71, pp. 31–41, 2018, doi: 10.1016/j.wasman.2017.10.019.
- [56] T. R. P. Ramos, C. S. de Moraes, and A. P. Barbosa-Póvoa, “The smart waste collection routing problem: Alternative operational management approaches,” *Expert Syst. Appl.*, vol. 103, pp. 146–158, 2018, doi: 10.1016/j.eswa.2018.03.001.
- [57] J. M. Gutierrez, M. Jensen, M. Henius, and T. Riaz, “Smart waste collection system based on location intelligence,” *Procedia Comput. Sci.*, vol. 61, pp. 120–127, 2015, doi: 10.1016/j.procs.2015.09.170.
- [58] K. Pardini, J. J. P. C. Rodrigues, O. Diallo, A. K. Das, V. H. C. de Albuquerque, and S. A. Kozlov, “A smart waste management solution geared towards citizens,” *Sensors (Switzerland)*, vol. 20, no. 8, pp. 1–15, 2020, doi: 10.3390/s20082380.
- [59] M. Aazam, M. St-Hilaire, C. H. Lung, and I. Lambadaris, “Cloud-based smart waste management for smart cities,” in *2016 IEEE 21st International Workshop on Computer Aided Modelling and Design of Communication Links and Networks (CAMAD)*, 2016, pp. 188–193, doi: 10.1109/CAMAD.2016.7790356.
- [60] K. F. Haque, R. Zabin, K. Yelamarthi, P. Yanambaka, and A. Abdelgawad, “An IoT based efficient waste collection system with smart bins,” in *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, 2020, pp. 1–5, doi: 10.1109/WF-IoT48130.2020.9221251.
- [61] W. E. Chen, Y. H. Wang, P. C. Huang, Y. Y. Huang, and M. Y. Tsai, “A smart IoT system for waste management,” in *2018 1st International Cognitive Cities Conference (IC3)*, 2018, pp. 202–203, doi: 10.1109/IC3.2018.00-24.
- [62] ArcGIS, “ArcGIS Online,” 2023, [Online]. Available: <https://www.arcgis.com/index.html>.
- [63] D. L. O’Connor, “Solid waste collection vehicle route optimization for the City of Redlands, California,” University of Redlands, 2013.
- [64] A. Malakahmad, P. Md Bakri, M. R. Md Mokhtar, and N. Khalil, “Solid waste collection routes optimization via GIS techniques in Ipoh city, Malaysia,” *Procedia Eng.*, vol. 77, pp. 20–27, 2014, doi: 10.1016/j.proeng.2014.07.023.



- [65] M. Abdallah, M. Adghim, M. Maraqa, and E. Aldahab, "Simulation and optimization of dynamic waste collection routes," *Waste Manag. Res.*, vol. 37, no. 8, pp. 793–802, 2019, doi: 10.1177/0734242X19833152.
- [66] A. Kallel, M. M. Serbaji, and M. Zairi, "Using GIS-based tools for the optimization of solid waste collection and transport: Case study of Sfax City, Tunisia," *J. Eng. (United Kingdom)*, vol. 2016, pp. 1–7, 2016, doi: 10.1155/2016/4596849.
- [67] C. Chalkias and K. Lasaridi, "Optimizing municipal solid waste collection using GIS Optimizing municipal solid waste collection using GIS," in *Proceedings of Energy, Environment, Ecosystems, Development and Landscape Architecture*, 2015, pp. 45–50, [Online]. Available: [https://d1wqtxts1xzle7.cloudfront.net/46321509/Optimizing\\_municipal\\_solid\\_waste\\_collect20160607-6016-1avx3qm-libre.pdf?1465336962=&response-content-disposition=inline%3B+filename%3DOptimizing\\_municipal\\_solid\\_waste\\_collect.pdf&Expires=1677608973&Signature=](https://d1wqtxts1xzle7.cloudfront.net/46321509/Optimizing_municipal_solid_waste_collect20160607-6016-1avx3qm-libre.pdf?1465336962=&response-content-disposition=inline%3B+filename%3DOptimizing_municipal_solid_waste_collect.pdf&Expires=1677608973&Signature=).
- [68] Esri, "An overview of the Network Analyst toolbox," 2023, [Online]. Available: <https://pro.arcgis.com/en/pro-app/latest/tool-reference/network-analyst/an-overview-of-the-network-analyst-toolbox.htm>.
- [69] O. Apaydin and M. T. Gonullu, "Route optimization for solid waste collection: Onitsha (Nigeria) case study," *Glob. Nest J.*, vol. 9, no. 1, pp. 6–11, 2007, doi: 10.4314/jasem.v13i2.55299.
- [70] H. L. Vu, D. Bolingbroke, K. T. W. Ng, and B. Fallah, "Assessment of waste characteristics and their impact on GIS vehicle collection route optimization using ANN waste forecasts," *Waste Manag.*, vol. 88, no. 2019, pp. 118–130, 2019, doi: 10.1016/j.wasman.2019.03.037.
- [71] M. K. Ghose, A. K. Dikshit, and S. K. Sharma, "A GIS based transportation model for solid waste disposal - A case study on Asansol municipality," *Waste Manag.*, vol. 26, no. 11, pp. 1287–1293, 2006, doi: 10.1016/j.wasman.2005.09.022.
- [72] S. Levin, D. Aronsky, R. Hemphill, J. Han, J. Slagle, and D. J. France, "Shifting toward balance: Measuring the distribution of workload among emergency physician teams," *Ann. Emerg. Med.*, vol. 50, no. 4, pp. 419–423, 2007, doi: 10.1016/j.annemergmed.2007.04.007.
- [73] M. Rabbani, A. Nikoubin, and H. Farrokhi-Asl, "Using modified metaheuristic algorithms to solve a hazardous waste collection problem considering workload balancing and service time windows," *Soft Comput.*, vol. 25, no. 3, pp. 1885–1912, 2021, doi: 10.1007/s00500-020-05261-4.
- [74] D. Jorge, A. Pais Antunes, T. Rodrigues Pereira Ramos, and A. P. Barbosa-Póvoa, "A hybrid metaheuristic for smart waste collection problems with workload concerns," *Comput. Oper. Res.*, vol. 137, no. August 2021, 2022, doi: 10.1016/j.cor.2021.105518.
- [75] Q. Qiao, F. Tao, H. Wu, X. Yu, and M. Zhang, "Optimization of a capacitated

- vehicle routing problem for sustainable municipal solid waste collection management using the PSO-TS algorithm,” *Int. J. Environ. Res. Public Health*, vol. 17, no. 6, 2020, doi: 10.3390/ijerph17062163.
- [76] R. Linfati, G. Gatica, and J. W. Escobar, “A mathematical model for scheduling and assignment of customers in hospital waste collection routes,” *Appl. Sci.*, vol. 11, no. 22, 2021, doi: 10.3390/app112210557.
- [77] L. H. Shih and H. C. Chang, “A routing and scheduling system for infectious waste collection,” *Environ. Model. Assess.*, vol. 6, no. 4, pp. 261–269, 2001, doi: 10.1023/A:1013342102025.
- [78] X. Qian, “A Heuristic-Mixed Genetic Algorithm for Type II assembly line balancing with multiple workers in workstations,” *Math. Probl. Eng.*, vol. 2022, pp. 1–8, 2022, doi: 10.1155/2022/9954518.
- [79] Y. J. Kim, Y. K. Kim, and Y. Cho, “A heuristic-based genetic algorithm for workload smoothing in assembly lines,” *Comput. Oper. Res.*, vol. 25, no. 2, pp. 99–111, 1998, doi: 10.1016/S0305-0548(97)00046-4.
- [80] I. Zaplana *et al.*, “A novel strategy for balancing the workload of industrial lines based on a genetic algorithm,” in *2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2020, pp. 785–792, doi: 10.1109/ETFA46521.2020.9212038.
- [81] A. S. Aremu, “In - town tour optimization of conventional mode for municipal solid waste collection,” *Niger. J. Technol.*, vol. 32, no. 3, pp. 443–449, 2013.
- [82] Regional Municipality of Durham, “Open Data,” 2023, [Online]. Available: <https://opendata.durham.ca/>.
- [83] L. Davis, *Handbook of genetic algorithms*. New York: Van Nostrand Reinhold, 1991.
- [84] A. Subasi, “Chapter 7 - Clustering examples,” in *Practical Machine Learning for Data Analysis Using Python*, A. Subasi, Ed. Academic Press, 2020, pp. 465–511.
- [85] F. Amato, X. Querol, C. Johansson, C. Nagl, and A. Alastuey, “A review on the effectiveness of street sweeping, washing and dust suppressants as urban PM control methods,” *Sci. Total Environ.*, vol. 408, no. 16, pp. 3070–3084, 2010, doi: 10.1016/j.scitotenv.2010.04.025.
- [86] National Highway Safety Administration, “Four Tips for Driving on Wet Leaves,” 2019, [Online]. Available: <https://nationalhighwaysafetyadministration.com/en/four-tips-for-driving-on-wet-leaves>.
- [87] S. Gillies, “Shapely 2.0.1 documentation,” 2023, [Online]. Available: <https://shapely.readthedocs.io/en/stable/index.html>.

# APPENDICIES

## Appendix A. Waste Collection

### 1. Route Statistics (Existing – Garbage and Organics)

Mon. Week 1 Area Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
mon1_r1_service	13.12	17.07	0.41	5.4	25.8
mon1_r2_service	10.37	14.33	0.33	5.02	21.54
mon1_r3_service	11.64	14.66	0.34	4.79	22.16
mon1_r4_service	12.85	18.67	0.35	5.11	24.43
mon1_r5_service	13.43	19.24	0.36	5.15	25.15
mon1_r6_service	14.59	18.43	0.37	4.84	25.3
mon1_r7_service	15.1	21.12	0.39	5.0	27.02
mon1_r8_service	14.48	21.61	0.39	4.79	26.49
mon1_r9_service	15.39	20.14	0.39	4.05	25.79
mon1_r10_service	10.53	14.88	0.27	3.98	19.21
<b>Total</b>	<b>131.48</b>	<b>180.15</b>	<b>3.6</b>	<b>48.13</b>	<b>242.88</b>

Mon. Week 2 Area Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
mon2_r1_service	15.26	21.9	0.38	3.89	25.77
mon2_r2_service	13.05	17.26	0.32	3.86	22.08
mon2_r3_service	10.93	15.71	0.29	3.86	20.09
mon2_r4_service	33.62	40.85	0.6	4.0	44.79
mon2_r5_service	15.53	18.07	0.36	3.93	24.37
mon2_r6_service	15.98	18.09	0.42	4.17	26.29
mon2_r7_service	10.53	13.82	0.29	3.65	19.12
mon2_r8_service	7.29	9.47	0.21	2.79	13.68
mon2_r9_service	9.35	10.62	0.43	4.04	21.3
mon2_r10_service	10.17	13.73	0.28	3.57	18.7
mon2_r11_service	181.82	128.18	3.21	6.05	198.5
<b>Total</b>	<b>323.53</b>	<b>307.7</b>	<b>6.79</b>	<b>43.81</b>	<b>434.69</b>

Tues. Week 1 Area Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
tues1_r1_service	7.95	11.12	0.22	3.97	15.82
tues1_r2_service	8.41	12.81	0.25	4.0	17.29
tues1_r3_service	12.14	16.49	0.34	4.08	22.26
tues1_r4_service	8.16	12.42	0.27	4.09	17.63
tues1_r5_service	16.04	19.51	0.42	4.1	26.8
tues1_r6_service	8.29	11.14	0.29	4.5	18.25
tues1_r7_service	8.96	12.19	0.29	4.09	18.5
tues1_r8_service	11.75	14.85	0.31	4.39	21.21
tues1_r9_service	11.87	16.5	0.41	4.03	23.57
tues1_r10_service	7.62	11.18	0.29	4.37	17.77
tues1_r11_service	14.32	18.1	0.43	4.63	26.25
<b>Total</b>	<b>115.49</b>	<b>156.31</b>	<b>3.53</b>	<b>46.26</b>	<b>225.35</b>

### Tues. Week 2 Area Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
tues2_r1_service	11.35	15.77	0.3	3.65	20.19
tues2_r2_service	14.72	20.88	0.35	3.9	24.68
tues2_r3_service	9.37	12.49	0.27	4.02	18.23
tues2_r4_service	13.33	18.34	0.34	3.97	23.23
tues2_r5_service	13.97	17.5	0.32	4.96	23.74
tues2_r6_service	15.34	19.99	0.35	4.05	24.82
tues2_r7_service	13.13	16.57	0.33	4.08	22.45
tues2_r8_service	11.65	15.32	0.29	3.91	20.25
tues2_r9_service	12.83	18.57	0.31	4.21	22.58
tues2_r10_service	10.74	12.57	0.31	3.97	19.61
tues2_r11_service	9.09	13.15	0.25	4.01	17.67
<b>Total</b>	<b>135.52</b>	<b>181.15</b>	<b>3.43</b>	<b>44.72</b>	<b>237.46</b>

### Wed. Week 1 Area Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
wed1_r1_service	12.54	15.82	0.31	3.37	20.71
wed1_r2_service	9.04	10.37	0.28	3.63	17.28
wed1_r3_service	16.73	23.36	0.3	3.87	25.14
wed1_r4_service	6.33	8.75	0.19	4.16	13.87
wed1_r5_service	9.85	12.78	0.29	3.82	18.66
wed1_r6_service	13.78	15.88	0.34	4.11	22.68
wed1_r7_service	13.15	16.01	0.33	3.68	21.89
wed1_r8_service	10.04	15.0	0.28	3.7	19.0
wed1_r9_service	9.34	13.83	0.25	3.74	17.73
wed1_r10_service	8.86	13.12	0.23	3.63	16.75
wed1_r11_service	9.6	13.59	0.28	3.57	18.22
<b>Total</b>	<b>119.26</b>	<b>158.51</b>	<b>3.07</b>	<b>41.3</b>	<b>211.92</b>

### Wed. Week 2 Area Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
wed2_r1_service	19.79	23.69	0.37	3.78	28.02
wed2_r2_service	11.04	13.81	0.28	3.27	18.8
wed2_r3_service	6.95	9.96	0.22	3.46	14.57
wed2_r4_service	10.57	14.79	0.28	3.9	19.4
wed2_r5_service	9.36	12.21	0.27	3.86	17.88
wed2_r6_service	9.89	12.44	0.25	4.43	18.31
wed2_r7_service	7.19	10.07	0.19	3.59	14.16
wed2_r8_service	7.2	9.96	0.22	3.78	15.13
wed2_r9_service	9.52	11.61	0.26	3.41	17.03
wed2_r10_service	9.16	11.27	0.25	3.73	16.85
wed2_r11_service	8.68	12.09	0.23	3.62	16.43
<b>Total</b>	<b>109.37</b>	<b>141.89</b>	<b>2.83</b>	<b>40.84</b>	<b>196.59</b>

### Thurs. Week 1 Area Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
thurs1_r1_service	11.28	14.32	0.3	3.34	19.47
thurs1_r2_service	10.28	14.03	0.29	3.97	19.41
thurs1_r3_service	10.59	14.09	0.29	3.87	19.32
thurs1_r4_service	9.34	12.64	0.24	3.34	16.75
thurs1_r5_service	22.16	26.15	0.48	4.1	32.68
thurs1_r6_service	12.22	16.88	0.31	4.17	21.79
thurs1_r7_service	13.17	18.6	0.36	4.29	23.94
thurs1_r8_service	15.81	20.48	0.37	3.93	25.51
thurs1_r9_service	14.15	18.03	0.36	4.68	24.59
thurs1_r10_service	14.08	18.58	0.4	4.58	25.54
thurs1_r11_service	12.41	15.88	0.33	4.11	22.04
<b>Total</b>	<b>145.49</b>	<b>189.68</b>	<b>3.73</b>	<b>44.39</b>	<b>251.03</b>

### Thurs. Week 2 Area Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
thurs2_r1_service	26.61	32.56	0.45	3.87	35.61
thurs2_r2_service	10.58	14.12	0.28	3.32	18.62
thurs2_r3_service	6.63	9.82	0.18	3.12	13.12
thurs2_r4_service	8.21	11.75	0.22	3.7	16.0
thurs2_r5_service	9.65	12.62	0.27	3.7	17.92
thurs2_r6_service	7.81	10.56	0.2	3.22	14.41
thurs2_r7_service	8.97	12.02	0.25	3.94	17.24
thurs2_r8_service	11.26	14.28	0.27	3.92	19.38
thurs2_r9_service	8.5	12.04	0.25	4.04	17.05
thurs2_r10_service	15.31	19.13	0.33	3.75	23.9
thurs2_r11_service	11.83	16.02	0.32	3.87	21.15
<b>Total</b>	<b>125.35</b>	<b>164.92</b>	<b>3.02</b>	<b>40.44</b>	<b>214.39</b>

### Fri. Week 1 Area Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
fri1_r1_service	13.12	18.01	0.4	4.24	24.6
fri1_r2_service	13.61	19.16	0.36	4.22	24.22
fri1_r3_service	12.06	17.3	0.34	4.98	23.23
fri1_r4_service	13.67	17.77	0.34	4.37	23.6
fri1_r5_service	8.87	12.89	0.25	4.37	17.79
fri1_r6_service	9.87	12.2	0.27	3.77	18.02
fri1_r7_service	9.26	12.65	0.25	3.92	17.61
fri1_r8_service	8.42	11.28	0.24	3.72	16.27
fri1_r9_service	11.36	15.25	0.3	3.87	20.4
fri1_r10_service	14.75	18.94	0.4	4.58	25.86
fri1_r11_service	12.47	17.02	0.33	4.31	22.6
<b>Total</b>	<b>127.49</b>	<b>172.47</b>	<b>3.48</b>	<b>46.35</b>	<b>234.22</b>

### Fri. Week 2 Area Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
fri2_r1_service	10.18	13.79	0.29	4.02	19.27
fri2_r2_service	10.25	14.2	0.28	4.0	19.21
fri2_r3_service	9.71	11.68	0.28	4.03	18.22
fri2_r4_service	7.63	10.46	0.25	4.03	16.39
fri2_r5_service	12.88	15.83	0.34	4.45	22.75
fri2_r6_service	15.26	18.6	0.43	4.24	26.45
fri2_r7_service	18.52	24.05	0.5	4.57	31.54
fri2_r8_service	9.56	13.12	0.27	3.74	18.08
fri2_r9_service	9.14	12.22	0.27	4.02	18.04
fri2_r10_service	9.25	11.71	0.26	3.32	17.0
fri2_r11_service	16.3	20.66	0.42	4.38	27.44
<b>Total</b>	<b>128.69</b>	<b>166.33</b>	<b>3.6</b>	<b>44.8</b>	<b>234.39</b>

## 2. Route Statistics (Clustered – Garbage and Organics)

### area\_1 Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
route_1	13.14	20.01	0.41	3.82	24.41
route_2	8.55	12.22	0.29	3.48	17.06
route_3	6.62	8.68	0.3	3.87	15.43
route_4	14.15	19.23	0.39	3.54	23.98
route_5	12.56	17.86	0.45	4.94	25.14
route_6	14.28	17.64	0.4	4.18	24.3
route_7	16.28	21.66	0.47	4.57	28.24
route_8	13.26	15.77	0.38	3.76	22.44
route_9	8.48	11.68	0.31	3.88	17.45
route_10	10.92	15.55	0.43	5.35	23.7
route_11	7.14	10.11	0.31	3.74	16.21
<b>Total</b>	<b>125.37</b>	<b>170.4</b>	<b>4.14</b>	<b>45.13</b>	<b>238.35</b>

### area\_2 Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
route_1	16.07	23.34	0.46	4.18	28.12
route_2	15.89	20.92	0.47	4.14	27.42
route_3	7.38	11.23	0.32	3.96	17.09
route_4	8.89	11.64	0.36	4.25	19.14
route_5	12.14	14.77	0.34	4.04	21.03
route_6	12.25	15.16	0.35	4.1	21.39
route_7	7.97	9.51	0.3	4.58	16.96
route_8	8.59	11.86	0.32	4.18	18.2
route_9	8.78	13.07	0.37	4.1	19.49
route_10	10.96	15.19	0.36	4.22	21.17
route_11	13.85	19.35	0.41	4.29	25.0
<b>Total</b>	<b>122.78</b>	<b>166.05</b>	<b>4.06</b>	<b>46.03</b>	<b>235.01</b>

### area\_3 Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
route_1	10.32	13.11	0.34	4.13	19.62
route_2	12.78	18.65	0.48	4.77	25.96
route_3	12.68	16.68	0.35	3.46	21.57
route_4	11.46	16.14	0.46	4.92	24.39
route_5	9.37	12.82	0.29	3.28	17.33
route_6	7.66	10.78	0.27	3.37	15.53
route_7	9.27	13.09	0.34	4.27	19.37
route_8	9.14	13.17	0.33	3.68	18.64
route_9	10.39	13.8	0.31	3.09	18.41
route_10	9.57	13.31	0.3	3.38	17.88
route_11	11.0	13.07	0.4	4.57	21.6
<b>Total</b>	<b>113.65</b>	<b>154.61</b>	<b>3.87</b>	<b>42.91</b>	<b>220.29</b>

### area\_4 Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
route_1	10.27	15.35	0.35	3.9	20.41
route_2	30.47	34.46	0.68	3.86	42.56
route_3	117.68	82.96	2.3	3.93	131.37
route_4	43.98	31.0	0.87	3.93	51.52
route_5	8.06	11.48	0.31	3.73	17.13
route_6	27.46	32.72	0.61	4.48	39.88
route_7	11.99	15.82	0.36	4.22	21.79
route_8	13.66	18.48	0.4	4.17	24.32
route_9	15.18	19.72	0.41	3.63	25.03
route_10	9.97	13.73	0.34	3.89	19.45
route_11	21.67	28.43	0.45	3.69	31.68
<b>Total</b>	<b>310.37</b>	<b>304.16</b>	<b>7.07</b>	<b>43.44</b>	<b>425.14</b>

### area\_5 Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
route_1	15.1	19.72	0.43	4.11	25.84
route_2	11.81	16.13	0.38	4.07	22.02
route_3	13.37	17.68	0.41	3.91	23.79
route_4	8.21	11.31	0.32	3.96	17.52
route_5	6.63	8.32	0.35	4.28	16.66
route_6	13.41	17.93	0.37	3.88	23.16
route_7	17.1	20.91	0.44	4.21	27.4
route_8	10.9	13.38	0.34	4.17	20.06
route_9	11.67	14.86	0.36	3.88	21.16
route_10	11.78	15.17	0.35	3.87	21.06
route_11	12.69	16.31	0.38	4.09	22.6
<b>Total</b>	<b>132.66</b>	<b>171.72</b>	<b>4.12</b>	<b>44.4</b>	<b>241.26</b>

### area\_6 Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
route_1	15.31	21.96	0.46	4.47	27.66
route_2	11.18	15.09	0.33	3.79	20.31
route_3	16.7	22.95	0.47	4.17	28.43
route_4	16.24	22.06	0.47	4.73	28.54
route_5	8.55	11.22	0.31	3.51	17.05
route_6	17.74	25.59	0.51	4.45	30.85
route_7	17.63	21.66	0.47	4.18	28.56
route_8	12.22	17.64	0.38	3.75	22.54
route_9	11.09	15.65	0.35	3.85	20.86
route_10	12.61	17.2	0.41	4.92	24.16
route_11	11.22	15.01	0.34	3.97	20.64
<b>Total</b>	<b>150.48</b>	<b>206.03</b>	<b>4.5</b>	<b>45.78</b>	<b>269.58</b>

### area\_7 Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
route_1	10.39	14.1	0.33	3.86	19.62
route_2	16.99	21.66	0.46	3.77	27.62
route_3	18.73	25.79	0.5	4.06	30.91
route_4	8.79	12.5	0.32	3.61	17.84
route_5	11.37	16.12	0.34	4.09	21.2
route_6	12.59	15.68	0.36	3.93	21.81
route_7	13.66	18.29	0.39	4.21	24.01
route_8	9.8	13.07	0.35	4.04	19.62
route_9	12.29	15.27	0.35	3.98	21.42
route_10	9.12	13.67	0.5	5.13	23.41
route_11	6.23	8.58	0.31	4.04	15.73
<b>Total</b>	<b>129.97</b>	<b>174.73</b>	<b>4.22</b>	<b>44.72</b>	<b>243.2</b>

### area\_8 Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
route_1	11.64	15.79	0.36	3.88	21.39
route_2	7.3	9.98	0.29	3.55	15.79
route_3	9.46	14.0	0.37	4.38	20.37
route_4	13.21	17.89	0.4	3.89	23.52
route_5	6.54	9.01	0.28	3.79	15.19
route_6	13.07	17.73	0.38	3.75	22.87
route_7	13.32	16.48	0.37	3.67	22.3
route_8	13.2	15.25	0.36	3.97	21.98
route_9	9.86	13.05	0.29	3.44	17.83
route_10	14.97	18.25	0.39	4.25	24.65
route_11	10.3	14.41	0.33	3.86	19.75
<b>Total</b>	<b>122.87</b>	<b>161.84</b>	<b>3.82</b>	<b>42.43</b>	<b>225.65</b>



### area\_9 Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
route_1	14.68	19.5	0.45	4.26	26.2
route_2	10.72	14.97	0.37	4.42	21.45
route_3	7.69	10.68	0.32	3.97	17.22
route_4	12.35	16.15	0.37	3.87	21.98
route_5	9.79	13.17	0.32	3.9	18.9
route_6	9.85	13.32	0.36	4.11	19.87
route_7	12.12	16.23	0.39	4.64	23.1
route_8	14.31	18.56	0.44	4.79	25.85
route_9	9.7	11.7	0.29	3.43	17.36
route_10	12.82	16.85	0.38	4.17	22.98
route_11	10.79	13.61	0.33	3.93	19.61
<b>Total</b>	<b>124.82</b>	<b>164.74</b>	<b>4.03</b>	<b>45.49</b>	<b>234.51</b>

### area\_10 Stats (G+O)

	Distance (km)	Fuel Consumed (L)	Time Travelling (hr)	Time Collecting (hr)	Objective Function Value
route_1	10.14	13.99	0.36	4.07	20.19
route_2	7.47	11.38	0.37	3.89	18.09
route_3	11.44	14.83	0.35	3.94	20.88
route_4	16.85	21.13	0.42	3.77	26.6
route_5	11.43	15.1	0.35	4.08	21.04
route_6	20.63	28.59	0.59	3.84	34.27
route_7	12.28	16.9	0.37	3.97	22.37
route_8	22.74	26.74	0.58	4.87	35.46
route_9	12.58	16.97	0.38	4.16	22.79
route_10	11.51	13.62	0.32	3.85	19.82
route_11	11.06	14.71	0.34	3.99	20.5
<b>Total</b>	<b>148.13</b>	<b>193.97</b>	<b>4.44</b>	<b>44.42</b>	<b>262.01</b>

## Appendix B. Street-sweeping

### 1. RES Route Statistics (Current - Fall)

Existing RES Sweeping Area 1 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	7.790312975559102	14.164205410107465	5.105825564371379	1.2000980271831112	1.7870389475299828	74.59611149546586	36.04738435046912
2	11.324223403076168	20.58949709650214	6.588983469479024	1.596998246774357	2.7419893377000224	96.26504848908857	32.00167269067683
3	10.997421499070809	19.995311816492393	7.1376101654022355	1.6832167195992465	2.7258425302776463	104.2804845165267	35.696418395011186
4	15.03307789570145	27.332868901275372	8.062018384791603	2.0039261079957	2.8730955542710017	117.78608860180529	29.495690386220026
5	11.80696512662877	21.46720932114323	5.000489208078093	1.3318075284746806	2.093725083827759	73.05714733002095	23.293615547657936
6	22.621423957846385	41.12986174153889	10.991638435806204	2.8034831219538443	4.328625562445784	160.58783754712866	26.724228991767447
7	11.648812416046768	21.179658938266854	4.778267506034313	1.2855591031326834	1.7982178627349594	69.81048826316129	22.560644248152002
8	12.038468734474892	21.888124971772527	3.642326823810597	1.093834251634195	1.3674982430580624	53.21439489587283	16.640652538798243
<b>TOTAL</b>	<b>103.26070600840434</b>	<b>187.74673819709886</b>	<b>51.30715955777345</b>	<b>12.998923106747819</b>	<b>19.716033121845214</b>	<b>749.5976011390701</b>	<b>27.327856691662234</b>

Existing RES Sweeping Area 2 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	12.051836558073886	21.912430105588886	6.924516568913826	1.6790108050331698	2.7679771508461	101.16718707183098	31.600860952193933
2	16.25335010701112	29.55154564911112	9.39834745842853	2.28576017506199	3.692302976295668	137.3098563676409	31.803234829144117
<b>TOTAL</b>	<b>28.305186665085003</b>	<b>51.4639757547</b>	<b>16.322864027342355</b>	<b>3.96477098009516</b>	<b>6.460280127141768</b>	<b>238.47704343947186</b>	<b>31.71706769244631</b>

Existing RES Sweeping Area 3 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	16.033350599508402	29.15154654456073	7.985403846398222	2.0307497927548295	2.8360801457026823	116.66675019587797	27.39272797822861
2	21.465370239321356	39.02794588967511	10.671433163148858	2.701896245988706	3.735001607102098	155.9096385136047	27.343056161128885
3	19.1022435446849	34.73135189942709	10.85355006003489	2.668738959825486	3.9453687722615345	158.5703663771098	31.250007461454228
4	29.237563869363008	53.15920703520544	16.389435909928597	4.012722389633499	5.7363025684750095	239.44965864405663	30.830850992706605
5	18.13049439593434	32.96453526533516	7.631264621313853	2.0291320084116053	2.6709426174598487	111.49277611739534	23.149923273266154
6	26.540870632910803	48.25612842347416	11.616259191316821	3.0732024941613014	4.065690716960886	169.71354678513873	24.07209109147285
7	13.184367463530618	23.971577206419287	4.471801316410936	1.2868987779987175	1.6292729778792174	65.3330172327638	18.654597809331644
<b>TOTAL</b>	<b>143.69426074525342</b>	<b>261.262292264097</b>	<b>69.61914810855218</b>	<b>17.803340668774144</b>	<b>24.618659405841278</b>	<b>1017.1357538659471</b>	<b>26.647223946951232</b>

Existing RES Sweeping Area 4 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	20.679020443420722	37.59821898803767	14.892302247214063	3.427881248571361	5.47903731645839	217.5765358317974	39.60906300362852
2	10.075425257765572	18.318955014119222	6.577311435163169	1.5490551571530098	2.3020590023071086	96.0945200677339	35.904403008216065
3	13.240643629204946	24.073897507645356	6.772742159737271	1.7039671495685396	2.5879874811649155	98.94976295376152	28.13313530800898
4	17.19733313435179	31.26787842609414	9.712347463568243	2.3781497801011042	3.4951370011901135	141.89739644273197	31.061741162019324
5	14.97455370184635	27.226461276084265	8.53634612254133	2.0784013897509452	3.2470003361124293	124.71601685032876	31.353123845145674
6	12.668800474307254	23.034182680558636	6.035393402009395	1.540136859435991	2.538952384413794	88.17709760335725	26.201899523456508
7	20.026588709351923	36.41197947154896	10.980128288235836	2.7156116620130297	3.9844350735229472	160.41967429112557	30.155263316061465
8	18.930920941431598	34.41985625714833	10.375619742753829	2.560155599937365	4.044604600018153	151.5878044416335	30.14428551135804
9	10.809438803466001	19.653525097210903	4.7675253582404595	1.2412784726131267	2.061944800353949	69.6535454838931	24.25786384202921
10	9.780641071350052	17.782983766091014	3.893527495463205	1.0634095399012056	1.4496991617017352	56.88443670871743	21.894680592845557
11	11.208401865486044	20.378912482701903	3.0369722956973852	0.9564119489844962	1.1102427806923145	44.370165240138775	14.902523862719555
12	6.574352164975348	11.953367572682454	1.2996402590516654	0.4702145373274378	0.49128715282980223	18.98774418474483	10.872586751383679
<b>TOTAL</b>	<b>166.16612019695762</b>	<b>302.12021853992286</b>	<b>86.87985626967586</b>	<b>21.684673345357613</b>	<b>32.79238709076565</b>	<b>1269.314700099964</b>	<b>28.756717008065895</b>

Existing RES Sweeping Area 5 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	20.320666001106286	36.94666545655689	10.397831545639715	2.6301936750752017	3.8949528579386787	151.9123188817962	28.142814560263435
2	27.64805905603151	50.26919828369356	13.423023148720072	3.4407176774058006	4.83092082798433	196.11036820280037	26.702282127054055
3	21.91237385252843	39.8406797318698	12.02746839347688	2.959422084433505	4.465648485325681	175.72131322869723	30.18891362904066
4	21.831651679341988	39.693912144258114	8.192027415428786	2.276814656062788	2.9625040967815934	119.68552053941453	20.63799452585274
5	18.339065452884633	33.34375536888114	6.808595203719926	1.8906632268632344	2.449539579115875	99.4735759263481	20.41940125938607
6	33.756471901307606	61.37540345692284	12.141806529018519	3.441963480465931	4.301372489866259	177.3917933889605	19.782854116047336
7	29.389247463661427	53.43499538847523	9.305415062466823	2.758860388416489	3.318245912350081	135.95211406264028	17.414458436490854
8	29.97173544188167	54.49406443978482	13.847619018807771	3.607566394139921	5.139027323763598	202.3137138647814	25.411242786100484
9	11.861391724817143	21.566166772394784	3.441880971963423	1.0518216519608505	1.2046583401871982	50.28588100038562	15.959632549856352
<b>TOTAL</b>	<b>215.03066257356068</b>	<b>390.9648410428372</b>	<b>89.58566728924193</b>	<b>24.05802323482372</b>	<b>32.5668699133133</b>	<b>1308.8465990958243</b>	<b>22.913995808494253</b>

Existing RES Sweeping Area 6 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	15.88489566971509	28.881628490391087	9.729434303164346	2.320458515614951	3.9359873134460943	142.14703516923112	33.687277386042574
2	18.291930320940747	33.25805512898318	12.047241137804473	2.8304296052544258	4.843375024758146	176.01019302332327	36.2235286792394
3	13.213970014063143	24.02540002556937	7.864289387520142	1.8886590289518979	3.3632017730207333	114.89726795166926	32.73322974498015
4	13.757236446232696	25.01315717496854	7.123941103733836	1.774264445196592	3.073887677465102	104.08077952555132	28.4807753531529
5	21.832109208113188	39.69474401475129	11.063531605224915	2.788507274366582	4.582819640514449	161.63819675233597	27.8715277988316
6	24.764316708932764	45.02603037987774	10.396630920190026	2.7530727440863147	3.989588467616505	151.8947774397625	23.090267635132918
7	19.85309000198402	36.09652727633456	8.771692001784281	2.2857476447334086	3.231651142530754	128.1544201460682	24.300653452430968
8	19.205307561261826	34.91874102047601	7.489944955680357	2.0424745964147912	2.9859390985084184	109.42809580249005	21.4496420548018
9	14.294160312097416	25.989382385631643	3.4136678804714236	1.124616813234926	1.3961216831634933	49.87368773368748	13.134855726154873
<b>TOTAL</b>	<b>161.09701624334087</b>	<b>292.90366589698345</b>	<b>77.9003732955738</b>	<b>19.80823066785389</b>	<b>31.402571821023695</b>	<b>1138.124453848333</b>	<b>26.595902464044947</b>

Existing RES Sweeping Area 7 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	22.93558366013018	41.70106120023667	13.683592733659989	3.3051234632811735	5.411852980168951	199.9172898387724	32.81353601040332
2	22.221933213988375	40.403514934524324	9.7793092479658	2.56621575640463	3.774740108253864	142.8757081127804	24.204105172071298
3	16.407144885777743	29.831172519595874	6.799951191624856	1.8185929440571362	2.7747583981581605	99.34728690963912	22.794783500910057
4	24.2317156737482	44.0576648613604	10.645006973165318	2.8092615426951286	3.8103828874366985	155.52355187794538	24.16153240681904
5	27.292429997174207	49.62259999486219	12.58986998596827	3.2628310271572953	4.752989032224878	183.9380004949964	25.371242109989794
6	26.682107683748068	48.51292306136012	11.945635120156417	3.1277720543802863	4.326553678768874	174.52572910548525	24.623614423413198
7	25.834304242635515	46.97146225933734	8.83933946469494	2.5263765161234275	3.250514527498058	129.142749579193	18.818531592419795
<b>TOTAL</b>	<b>165.6052193572023</b>	<b>301.1003988312769</b>	<b>74.28270471723559</b>	<b>19.41617330333491</b>	<b>28.101791612509484</b>	<b>1085.2703159188118</b>	<b>24.67041060243174</b>

Existing RES Sweeping Area 8 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	19.770104974561676	35.94564540829399	9.844138358993249	2.4752307127862543	3.8699835259284066	143.8228614248913	27.386177789206815
2	28.178399888117383	51.2334543420316	14.362702031545128	3.5863800089308517	5.125785359633124	209.83907668087426	28.03383495413086
3	22.59254732621507	41.07735877493651	11.163033550909272	2.8238465753173294	3.907061742818243	163.09192017878442	27.175636126148483
4	29.151697963036604	53.00308720552106	12.21817226489708	3.256004056161309	4.327486588745251	178.5074967901464	23.05181246805634
5	15.47887643247042	28.143411695400765	6.286707430300062	1.6768477852925896	2.371723718828215	91.84879555668394	22.338114150273558
6	17.698761760241812	32.179566836803325	6.068456497315858	1.7191018312869357	2.3474226682979378	88.66014942578464	18.858104983487376
7	22.11815588105797	40.21482887465082	8.756356993416876	2.384672616058114	3.586636713168687	127.9303756738206	21.773950650667555
<b>TOTAL</b>	<b>154.98854422570093</b>	<b>281.7973531376381</b>	<b>68.69956712737752</b>	<b>17.922083585833384</b>	<b>25.536100317419862</b>	<b>1003.7006757309857</b>	<b>24.379067568396444</b>

Existing RES Sweeping Area 9 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	28.69186400821631	52.16702546948421	14.394685602122433	3.6298651808372866	5.038139960742852	210.30635664700876	27.593456733588912
2	27.02403967266873	49.13461758667041	13.718337209137585	3.4399809182698564	4.801418023198155	200.4249066255001	27.919902266338582
<b>TOTAL</b>	<b>55.71590368088504</b>	<b>101.30164305615462</b>	<b>28.113022811260016</b>	<b>7.069846099107143</b>	<b>9.839557983941006</b>	<b>410.73126327250884</b>	<b>27.751793517974928</b>

Existing RES Sweeping Area 10 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	29.873138959364695	54.314798107935765	16.06158772794679	3.971158210755937	5.621555704781376	234.65979670530265	29.57129233184074
2	31.73724136364059	57.70407520661926	14.203954387500294	3.7098270400540074	4.9713840356251024	207.51977360137926	24.615166843313265
3	37.99834860445876	69.08790655356127	15.965147066760272	4.276213207965802	5.5878014733661	233.25079864536772	23.108453943937487
4	16.15820128167574	29.378547784865	6.136986991952302	1.6852689369864746	2.1479454471833055	89.66137995242312	20.88934768625257
5	35.9696039085273	65.39927983368592	16.46803598341516	4.249755274835931	5.7638125941953025	240.59800571769551	25.18076043848542
6	35.44270511473563	64.44128202679202	15.154240562328704	4.008177596303902	5.303984196815048	221.40345461562248	23.516354867099317
<b>TOTAL</b>	<b>187.17923923240272</b>	<b>340.32588951345923</b>	<b>83.98995271990353</b>	<b>21.900400266902054</b>	<b>29.39648345196623</b>	<b>1227.0932092377907</b>	<b>24.6792722344978</b>

Existing RES Sweeping Area 11 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	29.802295509896506	54.18599183617553	15.20003043926902	3.7893382784485596	5.3200106537441565	222.07244471772051	28.05158662634502
2	30.45635666174203	55.375193930440105	12.706901693649773	3.3570982869183554	4.447415592777417	185.64783374422305	22.946920438078514
3	32.9306610665338	59.87392921187962	15.578600728839763	3.9789906394341314	5.452510255093915	227.60335668434895	26.019005156168713
<b>TOTAL</b>	<b>93.18931323817233</b>	<b>169.43511497849525</b>	<b>43.48553286175856</b>	<b>11.125427204801047</b>	<b>15.21993650161549</b>	<b>635.3236351102926</b>	<b>25.66500625757403</b>

Existing RES Sweeping Area 15 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	25.911361949509992	47.111567180927246	2.678859461108704	1.3255844361088625	0.9376008113880465	39.13813672679817	5.68620324350666
<b>TOTAL</b>	<b>25.911361949509992</b>	<b>47.111567180927246</b>	<b>2.678859461108704</b>	<b>1.3255844361088625</b>	<b>0.9376008113880465</b>	<b>39.13813672679817</b>	<b>5.68620324350666</b>

## 2. AC Route Statistics (Current - Fall)

Existing ARTCOL Sweeping Area 1 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	32.504003437532916	59.098188068241654	14.64618187779859	3.7671682022208124	5.126163657229506	213.98071723463744	24.782793443491705
2	34.705314581703846	63.10057196673421	13.789596272753009	3.692179112359717	4.826358695463549	201.46600154492143	21.853361773048118
3	16.863294304516522	30.66053099120956	5.499229124752203	1.5880772215498555	1.9247301936632715	80.3437375126297	17.935855023319103
4	21.402149194446288	38.91299853535687	5.613511782984376	1.7517815194066524	1.9647291240445315	82.01340714940171	14.42580113142364
<b>TOTAL</b>	<b>105.47476151819956</b>	<b>191.7722936694537</b>	<b>39.54851905828818</b>	<b>10.799206055537038</b>	<b>13.841981670400859</b>	<b>577.8038634415902</b>	<b>20.622644857372137</b>

Existing ARTCOL Sweeping Area 2 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	20.34232750790625	36.98605001437498	10.920583851172049	2.678434975508562	3.8222043479102186	159.5497300656237	29.526223662509675
2	20.217906707542657	36.75983037735029	9.256609216763675	2.3820905419169267	3.2398132258672856	135.23906065691727	25.18131645805191
3	26.529947603806665	48.236268370557546	10.107904889209047	2.7694233587637482	3.5377667112231665	147.67649043134415	20.954989327861668
4	19.828705740554224	36.05219225555313	7.342782547544976	2.0651546279480963	2.569973891640742	107.2780530196321	20.36709028814736
<b>TOTAL</b>	<b>86.9188875598098</b>	<b>158.03434101783597</b>	<b>37.62788050468975</b>	<b>9.895103504137333</b>	<b>13.169758176641414</b>	<b>549.7433341735173</b>	<b>23.809939195711276</b>

Existing ARTCOL Sweeping Area 3 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	14.470523066909928	26.310041939836225	10.892095030363578	2.479592889587024	3.812233260627251	159.1335083936118	41.39900291786224
2	18.251871736032488	33.18522133824088	13.123774336216256	3.019641619068016	4.59332101767569	191.7383430521195	39.54704475962954
3	4.583471201759438	8.333584003198972	2.9545200068725954	0.6908113301228221	1.034082002405409	43.16553730040862	35.453173637398486
4	15.598415131606366	28.360754784738845	9.180108231668612	2.190907656054412	3.213037881084013	134.1213812646784	32.36905470727635
5	16.35950363036843	29.744552055215333	9.616225494563558	2.3140672730300538	3.365678923097244	140.4930544755736	32.3293673298989
6	15.420610002470616	28.037472731764737	7.275605197824576	1.855535486794268	2.5464618192386004	106.296591940217	25.94957565337821
7	32.35662193960355	58.83022170837004	13.429935732870055	3.610315724443553	4.700477506504518	196.21136105723156	22.828293592779914
<b>TOTAL</b>	<b>117.04101670875082</b>	<b>212.801848561365</b>	<b>66.47226403037924</b>	<b>16.160890040985304</b>	<b>23.265292410632725</b>	<b>971.1597774838405</b>	<b>31.23669483125323</b>

Existing ARTCOL Sweeping Area 4 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	23.057975219111988	41.92359130747635	12.114142338109524	3.0051872805836406	4.239949818338333	176.98761955978006	28.895764795677643
2	30.259973261329517	55.018133202417296	16.768055420681126	4.101883210918057	5.868819397238394	244.9812896961512	30.47732528290218
3	31.332889238424027	56.96888952440736	16.66293185400861	4.126643732667203	5.832026148903014	243.44543438706592	29.24917791643044
<b>TOTAL</b>	<b>84.65083771886553</b>	<b>153.910614034301</b>	<b>45.54512961279926</b>	<b>11.233714224168901</b>	<b>15.940795364479744</b>	<b>665.4143436429972</b>	<b>29.591935487080136</b>

Existing ARTCOL Sweeping Area 5 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	11.61482041392389	21.117855298043445	8.361800231247658	1.9237388688594736	2.926630080936679	122.16590137852822	39.59587805312017
2	19.157022327019117	34.83094968548929	12.362689399745129	2.922893647516148	4.379358878631239	180.6188921302763	35.49340316981215
3	14.636029046683731	26.61096190306131	10.513114496812438	2.430317364107708	3.6795900738843526	153.59660279842967	39.50670605260239
4	18.29437851885735	33.26250639792244	10.180279987232707	2.500200443407441	3.584421674241503	148.73389061346984	30.6058715643526
5	25.336442883415334	46.066259788027864	15.410143744339639	3.70633721633357	5.393550310518872	225.14220010480219	33.45212703451252
6	14.831757472008869	26.966831767288863	7.857945933700486	1.9560819975116652	2.75028107679517	114.80459009136413	29.139299720155787
<b>TOTAL</b>	<b>103.8704506619083</b>	<b>188.85536483983321</b>	<b>64.68597379307806</b>	<b>15.439569537736007</b>	<b>22.713832095007813</b>	<b>945.0620771168703</b>	<b>34.25159451940259</b>

### 3. RES Route Statistics (Clustered - Fall)

Clustered RES Sweeping Area 1 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	11.687807885563585	21.2505597919338	8.01914133957748	1.8658299296695282	3.3773839812050057	117.15965497122696	37.736141626826004
2	8.32993593007137	15.14533805467522	5.032824354511084	1.204980909188296	1.7614885240788796	73.52956381940695	33.23018830178901
3	5.077542973585741	9.23189631561044	2.7971054506435906	0.6845720927283694	0.9789869077252563	40.86571063390284	30.298276269783237
4	8.906910778265734	16.19438323321043	4.999842739938726	1.222351791448907	1.7499449589785534	73.04770243050478	30.873931213912247
5	9.891215869306913	17.984028853285306	5.172403294224222	1.2928138867687633	1.910775992496592	75.56881212858957	28.761093170052003
6	11.878968787106299	21.598125067466015	5.809963299765333	1.483576917992337	2.1331228569646403	84.88356380957154	26.900313252269648
7	5.689979525866553	10.34541731975737	2.574051076590031	0.6697997059896096	0.9009178768065108	37.60688622898034	24.881075330563853
8	21.49203364710742	39.076424812922596	12.57697562976328	3.0550682931539748	4.89831587814335	183.74961395084154	32.18558424926343
9	8.659606782407495	15.744739604377278	3.7728189630424795	0.9918494038867277	1.4021474455795997	55.12088505005063	23.962409400493218
10	8.4485836193791	15.361061126143822	4.424741554434336	1.1008034383119016	1.7069183627065543	64.64547411028562	28.804921210186635
11	5.94275283573599	10.804991424679267	2.7193602031754054	0.709230947698252	0.9517760711113917	39.72985256839267	25.16762944359418
12	9.14190920170838	16.621653094015247	3.8829956822802614	1.0279170939534552	1.3590484887980916	56.73056691811463	23.36106800158381
13	9.170686797067042	16.673975994667366	3.536289778736407	0.9654073359856468	1.2377014225577423	51.66519366733891	21.208437506851247
14	12.051689986398134	21.912163611632966	4.757633822517721	1.295275860551997	1.9166500256918342	69.50903014698389	21.712295996146803
15	12.82872806636231	23.32496012065875	5.89592584916848	1.5320642104752578	2.0635740472089674	86.13947665635148	25.277324457015897
<b>TOTAL</b>	<b>149.19834513376972</b>	<b>271.2697184250359</b>	<b>75.97207303836703</b>	<b>19.101541543306226</b>	<b>28.34875284005297</b>	<b>1109.9519870905424</b>	<b>28.006101631782965</b>

Clustered RES Sweeping Area 2 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	24.158207861714207	43.92401429402584	13.603316068146178	3.3310219374335817	4.798172721131922	198.74444775561577	30.97011194951468
2	25.927057969007553	47.140105398195544	11.996260375449456	3.1179744329990067	4.220481740770936	175.26536408531646	25.44809833180529
3	20.826861141127058	37.86702025659468	10.303118750378022	2.6152927334611293	3.850221279931706	150.5285649430228	27.208686293671857
4	28.580262653370735	51.964113915219514	14.14911621230274	3.5998450457247793	5.399748616750326	206.7185878617429	27.228629810540607
5	25.22590858110522	45.865288329282194	9.179137201355838	2.5733203615424594	3.6055892761220516	134.10719451180873	20.013255199565634
<b>TOTAL</b>	<b>124.71829820632477</b>	<b>226.76054219331775</b>	<b>59.230948607632236</b>	<b>15.237454511160955</b>	<b>21.87421363470694</b>	<b>865.3641591575067</b>	<b>26.120482882395258</b>

Clustered RES Sweeping Area 3 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	23.45959104448084	42.65380189905608	12.55685599168348	3.107236336473829	4.3948995970892195	183.4556660384956	29.43900762093932
2	30.502827638798234	55.45968661599679	15.259084048584148	3.8417427161424227	5.340679417004452	222.93521794981447	27.513830278627683
3	29.284053944959037	53.24373444538004	14.135553981267174	3.5908394109631289	4.947443893443509	206.52044366631335	26.548765086656527
4	26.359853902200452	47.92700709490991	11.266370330688678	2.980313535469642	3.9432296157410374	164.60167053136158	23.50735214568661
<b>TOTAL</b>	<b>109.60632653043857</b>	<b>199.28423005534285</b>	<b>53.2178643522348</b>	<b>13.520131999049283</b>	<b>18.626252523278218</b>	<b>777.512998185985</b>	<b>26.704503581364392</b>

Clustered RES Sweeping Area 4 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	16.406481276924364	29.829965958044294	7.011178296926633	1.8733828751183341	2.453912403924321	102.4333149180981	23.503809246003907
2	19.634447493384627	35.69899544251751	11.233740027358493	2.7459126048012914	4.1323868403860295	164.1249417997077	31.467944372404126
3	27.47247933061123	49.949962419293165	12.21158006059954	3.20351149116572	4.322919488874654	178.4111846853593	24.44762612250298
4	22.154354985870544	40.28064542885552	9.689158924987249	2.562056591656181	3.401396561319693	141.55861189406363	24.05413026983501
5	24.69369785756508	44.89763246830011	11.769432511154609	3.0298515208310284	4.267689754869366	171.95140898796885	26.213926802185828
6	14.5557117415809	26.464930439238007	5.25234394142528	1.4740163491821863	1.8383203794988474	76.73674498422332	19.846430178550314
7	26.064085484087034	47.38924633470377	10.662471978495313	2.8781788210635484	3.770115529817333	155.7787156058165	22.499771157337534
<b>TOTAL</b>	<b>150.98125817002378</b>	<b>274.5113784909524</b>	<b>67.82990574094713</b>	<b>17.76691025381829</b>	<b>24.18674095869024</b>	<b>990.9949228752374</b>	<b>24.7093239317884</b>

Clustered RES Sweeping Area 5 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	9.209160786626352	16.743928702956996	6.488232800773568	1.498695204937791	2.2972154861752694	94.79308121930187	38.74976366584587
2	6.938246392578813	12.614993441052388	4.273046090142627	1.0185924559742474	1.4955661315499196	62.4292033769838	33.87275712912423
3	10.922317662376392	19.85875938613888	7.3213837867766065	1.7127592776233365	2.6330808836742396	106.96541712480618	36.867276774030636
4	18.819918254480307	34.218033189964196	11.225446274877353	2.713170296973301	3.9813043267174875	164.0037700759582	32.805644358804535
5	21.406064965849488	38.92011811972631	11.361553213472003	2.8236118106492683	4.501802110114426	165.99229244882602	29.191980298008147
6	19.948306329874864	36.269647872499746	10.312334675936981	2.583044037579365	4.222981481548213	150.66320961543923	28.43240913776824
7	14.661086653718574	26.65652118857924	8.275440218386493	2.0254408552168957	2.8964040764352723	120.90418159062669	31.044711948129354
8	10.74845209563397	19.54264017387995	6.5307083975991596	1.5599200946806286	2.6490580475734107	95.41364968892373	33.417738542450834
9	11.320599624524283	20.582908408225972	5.201785581885832	1.3471281288663086	2.0597857409944	75.99808735135201	25.272354512382396
10	10.942916947697368	19.896212632177033	3.6515760815741105	1.0519875216601282	1.4245890689607161	53.34952655179774	18.35312151654743
<b>TOTAL</b>	<b>134.9170697133604</b>	<b>245.3037631152007</b>	<b>74.64150712142472</b>	<b>18.33434968416127</b>	<b>28.161787353743357</b>	<b>1090.5124190440154</b>	<b>30.428194893354014</b>

Clustered RES Sweeping Area 6 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	27.208296843667025	49.4696306248492	13.133099390261698	3.3468615131387907	4.689691063873342	191.87458209172343	26.54780159944186
2	24.756996689490997	45.01272125361997	12.760192199260565	3.195038466355448	4.466067269741197	186.42640803119684	28.347968849438036
3	29.619048528568822	53.85281550648876	13.116153789617423	3.438701768696422	4.665717882896717	191.62700686631047	24.3555813797154
4	27.570163126346184	50.127569320629426	13.680703630096545	3.4687459169392305	5.099205101218306	199.8750800357105	27.291775395274172
<b>TOTAL</b>	<b>109.15450518807303</b>	<b>198.46273670558736</b>	<b>52.69014900923623</b>	<b>13.449347665129892</b>	<b>18.920681317729564</b>	<b>769.8030770249412</b>	<b>26.549139593595473</b>

Clustered RES Sweeping Area 7 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	19.273157275857574	35.04210413792284	10.235663314441794	2.54483575651418	3.9758500104391024	149.54304102399462	29.20961388093325
2	26.63437520331784	48.426136733305164	12.967970276727126	3.311260731085146	5.2428081612009905	189.46204574298324	26.77886602465313
3	21.278509095109694	38.688198354744905	9.281738428273494	2.449005492106471	3.60729476180646	135.6061984370758	23.991136374886626
4	26.66847120213403	48.488129458425476	12.00384140314581	3.143617716984328	4.8047913792223085	175.3761228996034	24.756247636729526
5	33.745123314389716	61.35476966252688	12.683685174333885	3.5447142072821016	4.696441017267857	185.30864039701802	20.672696261592503
6	15.448220696655822	28.087673993919662	6.511223484721043	1.7328914463457308	2.2789282196523657	95.12897511177442	23.18178246490106
<b>TOTAL</b>	<b>143.04785678746467</b>	<b>260.0870123408449</b>	<b>63.68412208164315</b>	<b>16.726325350317957</b>	<b>24.606113549589082</b>	<b>930.4250236128064</b>	<b>24.48569865464289</b>

Clustered RES Sweeping Area 8 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	27.85000384049466	50.636370619081276	11.943559368192354	3.139693308656546	4.180245778867324	174.49540236929033	23.586918300364264
2	38.93549783501834	70.79181424548798	14.627070586579222	3.930971442159167	5.119474705302723	213.70150126992252	20.662093128248227
<b>TOTAL</b>	<b>66.785501675513</b>	<b>121.42818486456926</b>	<b>26.570629954771576</b>	<b>7.0706647508157126</b>	<b>9.299720484170047</b>	<b>388.19690363921285</b>	<b>21.881764916774646</b>

Clustered RES Sweeping Area 9 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	20.078883141336973	36.50706025697625	12.082855939144615	2.8996190319833732	4.8872525146185914	176.53052527090287	33.09731283234635
2	22.739968351647043	41.345397002994694	11.310272203811342	2.8470415764067423	4.565557865277432	165.24307689768372	27.355577703104718
3	23.658640330586497	43.01570969197544	12.89870283241425	3.161461765071434	5.180723855045318	188.45004838157215	29.98602818546662
4	12.64283517148571	22.98697303906492	5.146585655147806	1.376994101614881	1.9504888699006409	75.19161642170944	22.38914034658459
5	11.537638465511813	20.977524482748755	4.493192650941029	1.2143407658536125	1.6325112023149704	65.64554463024844	21.419079523115737
6	23.86365699557441	43.38846726468072	9.551153228709586	2.577370026327882	3.5545700355784637	139.54234867144706	22.013115076052618
7	14.35650777556825	26.102741413739693	5.21926363066342	1.4570941257495846	2.062573452548419	76.25344167909927	19.995078487499708
8	12.66993080198085	23.03623782178338	3.648812432559285	1.1085398878818076	1.3493141205537977	53.30914963969116	15.839445923365655
<b>TOTAL</b>	<b>141.54806103568015</b>	<b>257.36011097396386</b>	<b>64.3508385759426</b>	<b>16.642461280889314</b>	<b>25.182991915837633</b>	<b>940.1657515923541</b>	<b>25.004200663522557</b>

Clustered RES Sweeping Area 10 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	31.38222113499181	57.05858388180328	16.89586060998901	4.183068910354612	5.913551213496157	246.8485235119395	29.611426468257164
2	30.838356186296064	56.069738520538216	14.782160405697503	3.7832084585062735	5.173756141994126	215.9673635272406	26.36388325635375
3	28.869750713160666	52.49045584211027	13.145910028300813	3.4115257561285204	4.601068509905283	192.06174551347488	25.044381530698303
<b>TOTAL</b>	<b>91.09032803444855</b>	<b>165.61877824445176</b>	<b>44.82393104398733</b>	<b>11.377803124989406</b>	<b>15.688375865395564</b>	<b>654.877632552655</b>	<b>27.06452222333607</b>

Clustered RES Sweeping Area 11 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	29.38295628382485	53.423556879681485	14.51145854809423	3.670408054707774	5.5178597344928635	212.01240938765667	27.16303330528214
2	26.657719638300982	48.468581160547224	13.290631723621793	3.3428364452154082	4.94411540369123	194.1761294821143	27.421128090376595
3	16.44294655788318	29.89626646887853	7.1016358608732055	1.856945450434203	2.4855725513056224	103.7548999273575	23.754256633569547
4	27.361246205354902	49.74772037337247	11.857175213758108	3.10722556605255	4.458936121977788	173.2332987300589	23.834610158548443
5	24.460096629400013	44.47290296254549	10.29976287357299	2.738483916997074	3.6466157205756207	150.47953558290126	23.15963696421463
<b>TOTAL</b>	<b>124.30496531476392</b>	<b>226.00902784502523</b>	<b>57.060664219920326</b>	<b>14.71589943385701</b>	<b>21.053099532043127</b>	<b>833.6563042530356</b>	<b>25.247072988184755</b>

Clustered RES Sweeping Area 12 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	23.913203875614858	43.47855250111785	14.083169624261675	3.4049167886138525	5.8658674645175495	205.755108210463	32.39107287184777
2	26.411361885358527	48.02065797337912	13.171939989154229	3.334657334086183	4.983098431306444	192.44204324154313	27.429736586400516
3	27.124700256388753	49.31763682979773	12.792513351309164	3.2925827146727014	5.0172221605559315	186.89862006262703	25.939023387227518
4	18.88014641943387	34.32753894442522	8.693006971133288	2.2483496177298865	3.3293003797843306	127.00483184825731	25.323711627585315
5	14.824293163855462	26.953260297918998	4.2866232632685275	1.3147211687162648	1.5263662917648464	62.627565876353195	15.903913722821459
<b>TOTAL</b>	<b>111.15370560065146</b>	<b>202.0976465466389</b>	<b>53.027253199126875</b>	<b>13.595227623818888</b>	<b>20.721854727929106</b>	<b>774.7281692392437</b>	<b>26.238431820080375</b>

## 4. AC Route Statistics (Clustered - Fall)

Clustered ARTCOL Sweeping Area 1 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	9.500356639911416	17.273375708929855	9.631446508552242	2.069121133091745	3.3710062779932852	140.71543348994828	55.758912854382324
2	14.171538053710176	25.76643282492759	11.865601343022876	2.646306759328836	4.152960470058005	173.35643562156417	46.05061718727152
3	18.578740057778276	33.77952737777869	13.384458784532768	3.068975509109418	4.684560574586468	195.5469428420237	39.62299008759227
4	14.776593985111724	26.86653451838495	11.03415605753488	2.5170181663698057	3.8619546201372073	161.20902000058462	41.070261778586044
5	4.583471201759438	8.333584003198972	2.9545200068725954	0.6908113301228221	1.034082002405409	43.16553730040862	35.453173637398486
6	22.064865512553215	40.11793729555132	14.025402901907158	3.31503185284948	4.908891015667506	204.91113639686358	34.960428794185376
7	16.65508717033115	30.281976673329336	7.9061437021584045	2.0214493226326815	2.7671502957554415	115.5087594885343	26.10841355386715
<b>TOTAL</b>	<b>100.3306526211554</b>	<b>182.4193684021007</b>	<b>70.80172930458092</b>	<b>16.328714073504788</b>	<b>24.780605256603323</b>	<b>1034.413265139927</b>	<b>38.812616184765595</b>

Clustered ARTCOL Sweeping Area 2 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	20.97105721912178	38.1291949438578	13.448553614606556	3.180156278162491	4.743864398827546	196.48336830940184	35.27101381083047
2	25.55235223353291	46.45882224278715	13.525866717351327	3.355023400148633	4.749600306078159	197.61291274050294	29.113666822346644
3	19.7672289340373	35.94041624370419	12.06406664211089	2.879119939263281	4.222423324738814	176.2560136412401	33.56685287200644
4	30.62037862429857	55.673415680542874	8.817608869363697	2.689768427160494	3.107486782987353	128.82526558140364	15.838095725901969
<b>TOTAL</b>	<b>96.91101701099056</b>	<b>176.201849110892</b>	<b>47.85609584343247</b>	<b>12.1040680447349</b>	<b>16.82337481263187</b>	<b>699.1775602725485</b>	<b>27.159814772042722</b>



Clustered ARTCOL Sweeping Area 3 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	18.62867255946104	33.8703137444746	11.27230558341905	2.7088945522171746	3.9453069541966665	164.6883845737523	33.28078289578273
2	23.181398972030603	42.14799813096472	14.870274905257695	3.5127187393164285	5.204596216840192	217.25471636581483	35.28109415553239
3	17.189163588115008	31.253024705663652	8.812898393139818	2.2004896563948466	3.0845144375989353	128.75644552377273	28.19854550443801
4	20.18338594446566	36.69706535357392	10.191605737168743	2.5616674402442854	3.5670620080090596	148.89935982003541	27.772263637359725
5	23.737812112514177	43.1596583863894	12.07883437015328	3.0272014778953578	4.227592029553648	176.4717701479395	27.9863993871703
<b>TOTAL</b>	<b>102.92043317658649</b>	<b>187.12806032106627</b>	<b>57.225918989138584</b>	<b>14.010971866068093</b>	<b>20.029071646198503</b>	<b>836.0706764313148</b>	<b>30.58115329734772</b>

Clustered ARTCOL Sweeping Area 4 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	28.197961697984137	51.269021269062065	12.196589384590496	3.1727694253176706	4.268806284606674	178.19217090886713	23.789393834109415
2	31.24751114075478	56.81365661955415	14.774098953515654	3.720616504583796	5.170934633730476	215.84958571086375	26.00448524630027
3	26.117286278916847	47.485975052576094	8.793163976182583	2.503671420444011	3.077607391663904	128.46812569202757	18.51739164342074
4	20.619681043608676	37.49032917019759	5.282780840042845	1.6693316027100418	1.8489732940149954	77.18142807302596	14.091049497218918
<b>TOTAL</b>	<b>106.18244016126445</b>	<b>193.0589821113899</b>	<b>41.04663315433158</b>	<b>11.066388953055519</b>	<b>14.36632160401605</b>	<b>599.6913103847844</b>	<b>21.261188008672274</b>

Clustered ARTCOL Sweeping Area 5 FALL Statistics

Subroute	Fuel (L)	Distance (km)	Serviceable Distance (km)	Travel Time (hr)	Debris (m3)	Water (L)	Efficiency (%)
1	22.295196888093614	40.53672161471566	11.900670365240352	2.9666681345853534	4.1652346278341215	173.86879403616157	29.357752406203875
2	17.370085720831856	31.581974037876105	7.25564624375194	1.9348926139117253	2.539476185313179	106.00499162121582	22.97401117184847
3	17.960537904285488	32.65552346233721	7.08861123751318	1.9384448456766503	2.4810139331296135	103.56461018006755	21.707235058377584
4	26.915691462918385	48.937620841669734	10.790384288142132	2.9366981257223483	3.776634500849744	157.64751444975647	22.049262106657753
<b>TOTAL</b>	<b>84.54151197612934</b>	<b>153.7118399565987</b>	<b>37.035312134647604</b>	<b>9.776703719896076</b>	<b>12.962359247126658</b>	<b>541.0859102872015</b>	<b>24.09398790952259</b>