Snow Clearing Vehicle Routing – The Postman Problem in a Hierarchical Network

by

Dong Qiang Liu

A Thesis
Presented to the University of Manitoba
in Partial Fulfillment of the Requirement for the Degree of
Master of Science
in
Industrial Engineering Program
Mechanical Engineering Department

Winnipeg, Manitoba

August 1988

ISBN 0-315-47937-X

SNOW CLEARING VEHICLE ROUTING - THE POSTMAN PROBLEM IN A HIERARCHICAL NETWORK

BY

DONG QIANG LIU

A thesis submitted to the Faculty of Graduate Studies of the University of Manitoba in partial fulfillment of the requirements of the degree of

MASTER OF SCIENCE

Acknowledgments

The author deeply acknowledges the guidance and encouragement he received from Dr. A. S. Alfa, Transport Institute and Department of Civil Engineering, over the course of this research. The author is also grateful for the financial support supplied by the Transport Institute. The assistance and cooperation from Mr. A. Hope, the City of Winnipeg, is greatly appreciated.

Abstract

The snow clearing vehicle routing problem (SCVRP) is a problem of finding an optimal route in a street system such that the streets of higher priority are covered before the streets of lower priority. This thesis defines the SCVRP as the postman problem in a hierarchical network, or the Hierarchical Chinese Postman Problem (HCPP). The HCPP is concerned with finding an optimal postman tour on a special network where the links are associated with priorities. Unlike most of the other link-covering vehicle routing problems in literature, the HCPP, or the SCVRP, has received relatively little attention in both the theoretical aspect and the practical aspect. This thesis studies the SCVRP and developes solution method to deal with it. The proposed method is based on decomposing the SCVRP into two subproblems: (1) the problem of finding an optimal postman tour for each link class (hierarchy), and (2) the problem of optimally composing the postman tour of each class into a total postman tour. Heuristic procedures are presented to solve these sub-problems.

The presented method to the SCVRP is tested with numerical and practical examples. It has been applied to solve the SCVRP in a district street system of the City of Winnipeg with three priority street classes. By using the proposed routing strategy, the total completion times for priority II and priority III streets, in theory, are only 14% and 26% of their maximum allowed completion times.

Contents

Chapter 1

Introduction

1.1 Background

Snow clearing and hauling operations from streets and sidewalks after a major snow fall in the City of Winnipeg constitutes a major activity during winter months. The operations often take a very long time to complete and are sometimes very expensive. This situation becomes more acute after a major snow storm which often leaves the city paralysed for a length of time and also reduces the available budget for handling operations for the rest of the winter. In Nov. 1986, the cost of snow clearing after a major snow storm was about \$6m, 37.5 % of the 1986-1987 fiscal year total snow removal expenditure, and took nearly one week to complete the operations on major streets.

In the City of Winnipeg, the engineering operations activities are decentralized under six districts. Each district is responsible for arranging its snow clearing operations. All districts, except the downtown district, contract out most of the snow clearing operations. Contractors are paid for the operations on the basis of the total length of the streets that they have to clear out and the amount of snow to be cleared. The cost to the contractor is dependent on these two factors and how efficiently the contractor's co-ordinater plans the routes of the vehicles for the operations. When a district carries out some or all its snow clearing operations it often needs to hire equipment and operators to carry out most of the operations. It pays for the equipment and the operators on an hourly rate basis. The cost to the district, and hence to the city, is dependent on the vehicle routing strategy. It is therefore imperative that a very efficient vehicle routing strategy is put in place that can cut the cost and length of time needed to carry out the snow clearing operations. The research work conducted in this thesis is motivated by such an idea.

1.2 Research Scope and Purpose

To efficiently organize and plan the snow clearing and hauling operations requires solving several problems simultaneously. Operational problems to deal with are not only the routing of vehicles for snow clearing but also the routing of the vehicles for snow hauling, organization of the crew, crew scheduling, use of the equipment, etc. We must state, however, that the finding of the solutions to all these problems is complex. In this thesis, the author will focus solely on the development of the model and the solution techniques to efficiently solve the problem of routing the vehicles for snow clearing operations. Other problems will not be treated in any detail in this thesis and will be left as extensions of the problem.

The author must emphasize that the work conducted in this thesis are mainly the theoretical research. The results obtained are suffered from certain inadvertent omissions. In the actual situation, there exist many factors that affect the vehicle route. For instance, the multi-clearing vehicle case, the geometrical variants of streets, the changing of weather conditions (e.g. thickness of snow), the location of depot, etc. In order to keep the problem in a reasonable size, we will ignore these factors in this thesis. We consider only the situation where the street network is ideal and only single vehicle is being used.

The remaining chapters of this thesis are organized as follows: Chapter 2 defines the snow clearing vehicle routing problem (SCVRP) and provides the literature review. The SCVRP is defined as a new extension of the Chinese Postman Problem (CPP) known as the Hierarchical Chinese Postman Problem (HCPP). Chapter 3 introduces the CPP and reviews the basic solution techniques. Chapter 4 presents the solution techniques to the SCVRP. A heuristic algorithm based on discomposing the SCVRP into sub-problems is introduced. Heuristic solution procedures used to solve the sub-problems are described. Chapter 5 presents an example of using the proposed method to solve the SCVRP on the practical street network system. Chapter 6 discusses the solution algorithm for the SCVRP. Finally, Chapter 7 presents conclusions and directions for future research.

Chapter 2

Problem Description and Literature Review

2.1 Problem Description

In the City of Winnipeg, snow clearing operations have the following basic requirements regarding the vehicle route:

—regional streets have the highest priority (Priority I) and they are required to be cleared first, followed by Collectors and the Bus routes (Priority II). The local residential streets with the lowerest priority (Priority III) are cleared next.

We have arbitrarily selected district #2 of the City of Winnipeg's street network as a case study in this thesis. The map of the district is shown in Figure 2.1. Notice that only part of the regional streets (Priority I) are included since the rest are scattered in a wide area of the city. To simplify the problem only the Priority II and Priority III street classes are considered in the discussions.

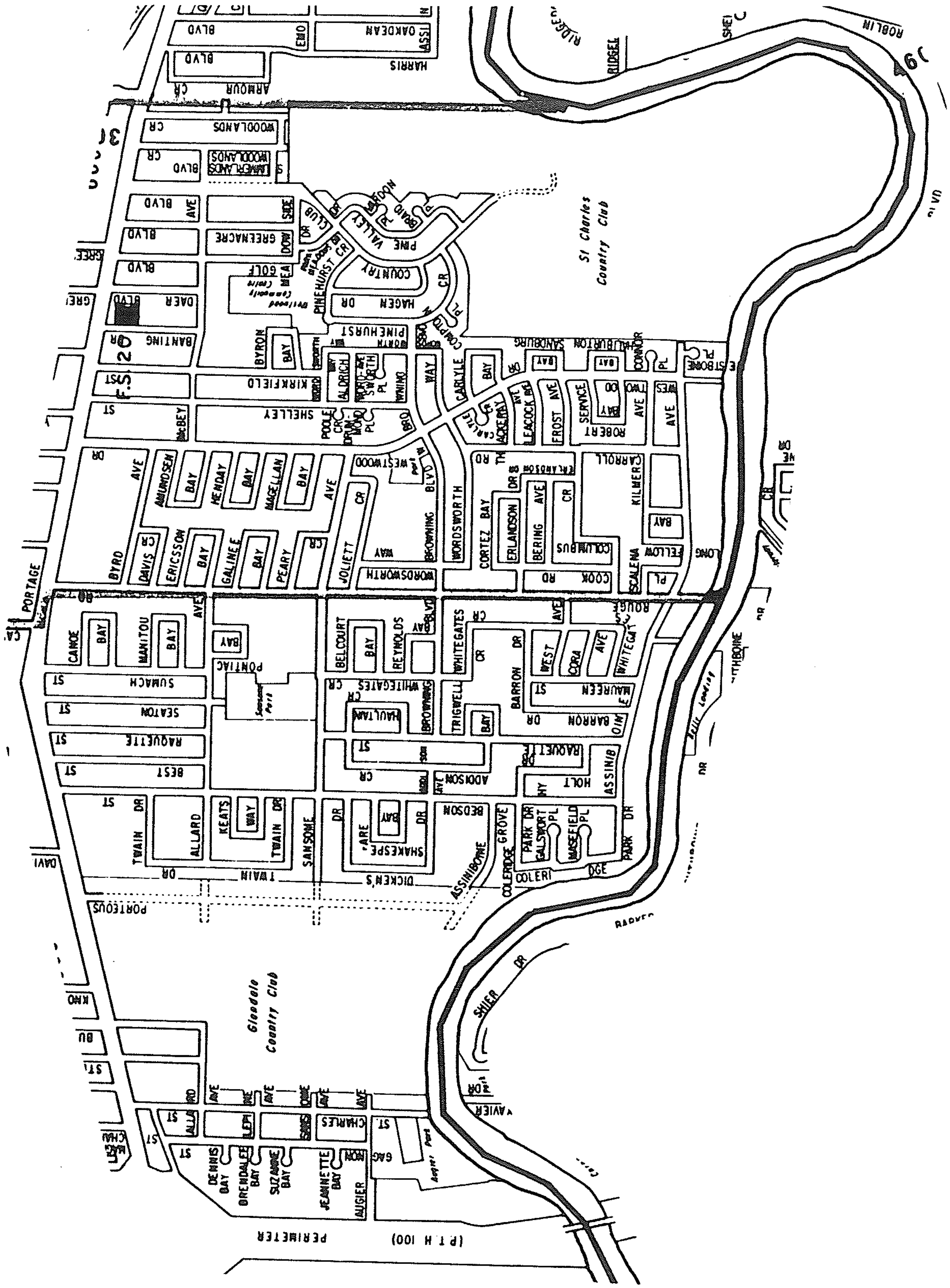


Figure 2.1: District #2 of the City of Winnipeg street network

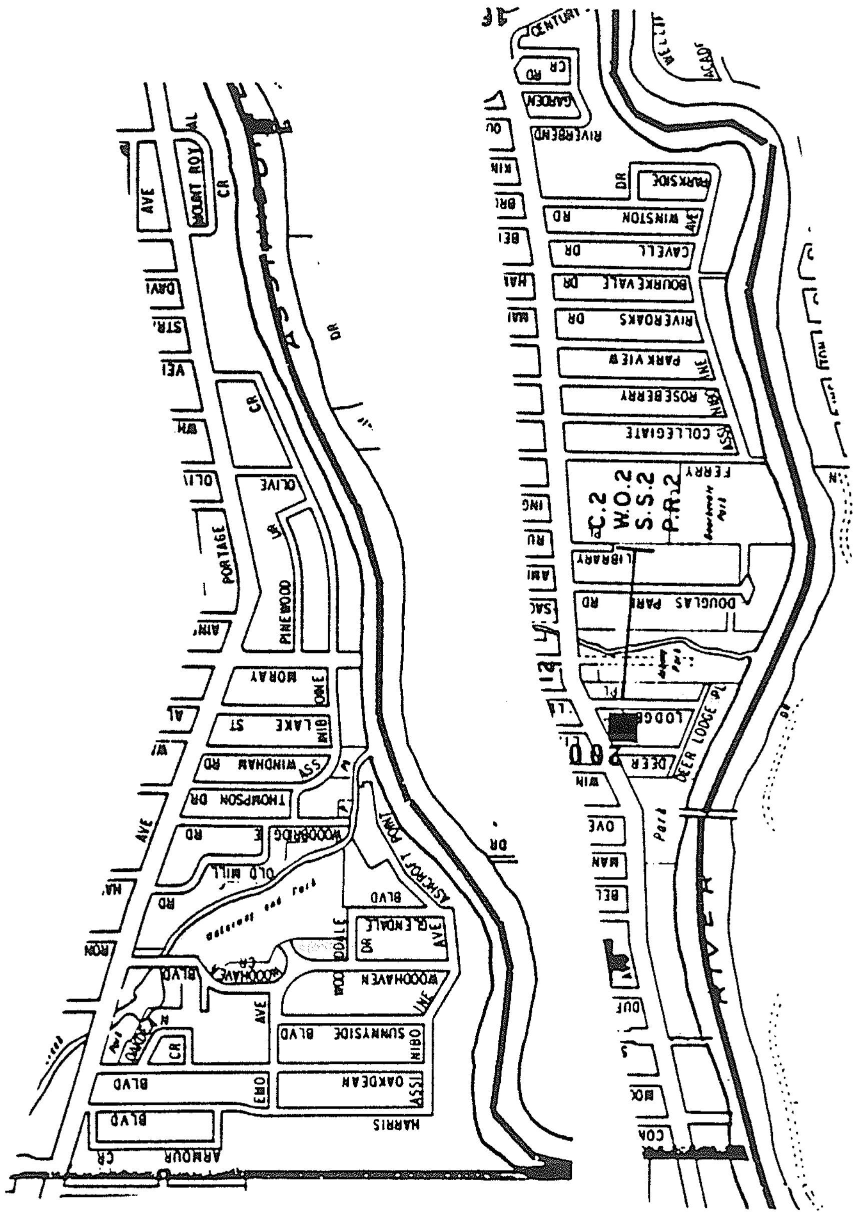


Figure 2.1: District #2 of the City of Winnipeg street network (continued)

The problem of routing the vehicles for snow clearing operations, or the SCVRP, can be described as a problem of finding an "optimal" vehicle route to cover each and every street in the network at least once, without violating the priority requirements described above. The "optimal" vehicle route is the one that contains the least deadheading length. Deadheading is referred to as the non-operational vehicle movement.

Figure 2.2 shows the corresponding graphical representation of the street network system shown in Figure 2.1. A directed network is used because the clearing operations are usually carried out one side at a time on a street in the traffic-flow direction. In such a network, a link represents one side of the street. We assume all streets have two sides to be cleared. The direction of the link indicates the traffic-flow direction on the real street. Hence each street block is represented by two links in either the same directions for a one-way street or two opposite directions for a two-way street. A node stands for an intersection of the street system where more than two streets meet. Associated with each link are two numbers representing the information on the corresponding real street block:

- the length of the link; and
- the priority level of the link.

We will call such a network the *Hierarchical Network*[1].

With the graphical representation, the SCVRP can be defined formally as the network optimization problem given below.

Figure 2.3: Graphical representation of the street network

Given a connected network $G = (N, A, C, P)$, where N is the node set, A is the link set, $C = [c_{ij}]$ is the cost matrix ($c_{ij} > 0$; $\forall ij \in N$) and $P = [p_{ij}]$ is the priority matrix ($p_{ij} = 1, 2, ..., h$, where h is the number of priority level in G). Let $A(1), A(2), \ldots, A(h)$ be the corresponding link subsets of priority level $1, 2, \ldots, h$ and $A(1) \cup A(2) \cup, \ldots, \cup A(h) = A$, $A(r) \cap A(s) = \phi(r, s = 1, 2, \ldots, h, r \neq s)$. Find the optimal cycle Ω such that

(1) each and every link of G is covered at least once; and

(2) $A(r)$ is covered completely before starting $A(s)$ if $r < s(r, s = 1, 2, \ldots, h)$.

It can be seen that the above problem is similar to the well-known *Chinese Postman Problem* (CPP) [33]. The CPP a link covering problem that is concerned with the finding of an optimal cycle–called the optimal postman tour– that traverses each of the link of a given network at least once. The basic CPP has been extended to the CPP in the un-directed network, CPP in a directed network, CPP in a mixed network, CPP in a capacitated network, and CPP in the rural network (See extensive literature review in next section). Our problem is significantly different in two aspects:

(1) links of certain hierarchy may not be all directly connected; and

(2) the tour (vehicle route) must cover each link hierarchy in the priority specified.

Hence we view the SCVRP as a new extention of CPP - or called the Hierarchical Chinese Postman Problem due to Alfa and Liu[1]. In [1], the HCPP was explored initially in theoretical aspect. To some extent, the presented research can be viewed as the extension of that work.

2.2 Literature Review

The *Chinese Postman Problem*, or CPP, is closely related to the *Euler tour.* An Euler tour is a closed tour that passes through each link of the given network exactly once. The CPP can be traced back to a famous mathematical problem known as the *Konigsberg Bridge Problem*, which was a problem of finding a closed tour that passes through each of the bridges exactly once without repeating any. Euler in 1736 studied such a problem as the network problem and proved that the sufficient and necessary condition for the existence of an Euler tour in a connected network is that it must be an even network. A network is even if all its nodes are of even degree.

Kwan[33] in 1962 addressed the problem of finding a tour that covers each link of the given network at least once with the minimum repeated length. When studying the routing of mail delivery, Kwan noticed that sometimes the postman has to repeat some streets in order to cover every street in the district. He was then interested in knowing which route the postman should choose so that each street is covered at least once and the repeated length, or the equivalent of deadheading, is minimized. Such a problem, as well as the other problems having such a property, were later called the Chinese

Postman Problem or the Postman Problem.

The basic CPP, since it was introduced by Kwan[33]), has been extensively studied by many other researchers. Some of these researchers are Edmonds[22], Bellman and Cooke[5], Stricker[46], Christofides[14], Edmonds and Johnson[23]), Minieka[35][36] and Sebo[44].

Historically, the basic CPP has been thought of as the postman problem in the un-directed network. Given a connected network $G = (N, A, C)$, where all links of A are un-directed, find a cycle that covers each link of A at least once. Here the postman can visit each link from any direction. Such a problem is often referred to as the *un-directed Chinese Postman Problem.* The un-directed CPP has been studied in Kwan[33], Bellman and Cooke[5], Edmonds and Johnson[23] and Minieka[35] and the optimal solution algorithms are also available.

Several other variants of CPP are summarized as follows:

The directed Chinese Postman Problem

Given a network $G = (N, A, C)$, where each link of A is directed and is visited by the postman only in a specific direction, find an optimal cycle that passes through each link of A at least once. Edmonds and Johnson[23] and Minieka[33] have discussed this kind of problem and presented efficient solution algorithm to solve it.

The mixed Chinese Postman Problem

Here the link set A of the given network $G = (N, A, C)$ consists of both directed links and undirected links. Find the optimal cycle that passes through each link of A at least once. The mixed network reflects the fact that a real street system may consist of some one-way streets and some two-way streets. Note that the street network for SCVRP can not be represented by a mixed network since a street will always have a direction associated with it. Discussions and heuristic/optimal solution algorithms for the mixed CPP have been provided in [23][36]. Minieka[36] also proposed a solution method to treat a special case of the mixed CPP where the cost of traversal of a link is dependent on the direction. Such a mixed CPP is called the *Windy Postman Problem*[36].

The un-directed, directed, and the mixed CPP are often referred to as the general CPP.

The Capacitated Chinese Postman Problem (CCPP)

Given a network $G = (N, A, C, Q)$ where each link $l_{ij} \in A$ is associated with a demand $q_{ij} > 0$, each vehicle(postman) has the capacity w, find cycles, each of which passes through the depot, that satisfies all demands at the optimal of travel. This problem has been studied by Christofides[15], Golden and Wong[27], Golden *et al*[26].

The Rural Chinese Postman Problem (RCPP)

Given a connected network $G = (N, A, C)$, let $R \in A$ be a link subset required to be covered by the postman, let $G(R)$ be the subnetwork of G generated from link subset R. Find an optimal cycle that passes through each link of $G(R)$ at least once. This problem has been addressed and studied by Orloff[39][40] and Lenstra and Kan[30]. The optimal solution method to the RCPP when the $G(R)$ is directly connected and a heuristic method when $G(R)$ is not directly connected have been developed in [39]. Tucker and Bodin[51] also tackled a similar problem and provided a different heuristic solution method when $G(R)$ is not directly connected

The Hierarchical Chinese Postman Problem (HCPP)

Given a network $G = (N, A, C, P)$ where A is divided into h subsets $A(1), A(2), \ldots, A(h)$, each subset is assigned to a different number $p, (p = 1, 2, ..., h)$ indicating the priority for being traversed by the postman. Find an optimal cycle such that each link of A is visited at least once and the subset $A(r)$ is covered earlier than $A(s)$ if $r < s$, $(r, s = 1, 2, \ldots, h)$. It can be seen that the HCPP is identical to the SCVRP. This problem was addressed initially in a paper by Alfa and Liu[1]. Alfa and Liu[1] have developed a heuristic solution algorithm to solve part of the problem. (See detailed discussion in chapter 4).

The solutions of the CPP and its extensions have been found widely applicable in solving the real life vehicle routing problems such as the police patrols, street sweepers, indefatigable tourists, repair crews, refuse collections, etc, which have the link-covering requirements. Successful application of the CPP solutions in vehicle routing have been reported in street sweeper vehicle routing [9][10][51], public sector vehicle routing[46], roadway maintenance vehicle routing[47], and electric meter vehicle routing[45]. It seems, however, that the existing solutions do not take into account the link priority, that is, the priority link-covering vehicle routing problem has not been dealt with.

Chapter 3

Introduction to the Chinese Postman Problem

In this chapter we will give the notations which are often used in the link-traversal in graph theory and the formal definition of the general Chinese Postman Problem (CPP). We will also discuss the solution techniques developed for solving the general CPP. This is because most of the existing methods for solving the extensions of the CPP are based on the solutions to the general CPP. Special attention will be given to the solution technique for solving the CPP in a directed network. Since its solution is adopted to the method for solving the SCVRP discussed in next chapter.

3.1 The Chinese Postman Problem (CPP)

Given a connected network $G = (N, A, C)$, N is the node set, A is the link set and $C = [c_{ij}]$ is the cost matrix. A *Walk* is a finite alternative sequence of nodes and links, beginning and ending with nodes, such that each link is incident with nodes preceding and following it. No link appears more than once. A *Closed walk* is a walk with the same beginning and ending node; otherwise it is an *Open walk*. A *Path* between node i and node j is an open walk with the beginning node i and ending node j, where no node is repeated more than once. An *Euler tour* is a closed walk such that it covers each and every link of a given network G exactly once. An *Euler network* is a network that contains an Euler tour. A *Postman tour* is a closed walk than covers each and every link of a given network G at least once. The *Chinese Postman Problem* is a problem of finding an optimal postman tour in the given connected network G.

3.2 Solution Techniques to the CPP

For the CPP on the connected network $G = (N, A, C)$, Stricker[46] has provided the following integer programming (IP) formulation:

$$\text{Minimize } \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{3.1}$$

$$\text{Subject to } \sum_{k=1}^{n} x_{ki} - \sum_{k=1}^{n} x_{ik} = 0 \; for \; i = 1, 2, \ldots, n \tag{3.2}$$

$$x_{ij} + x_{ji} > 1 \forall l_{ij} \in A \tag{3.3}$$

$$x_{ij} > 0 \text{ and integer} \tag{3.4}$$

where n is the number of the node in G; x_{ij} is the number of times the link (i, j) is repeated. Constraint (3.2) makes sure that each node becomes even after the repeated links are added to the original network. Constraint (3.3) guarantees that each link of the network is visited at least once.

The above integer programming formulation gives the framework of the general CPP. Unfortunately, such a model, in general, can not be solved by using the classical IP techniques because the size of the problem is too large (Bodin *et al*[11]). To solve the CPP, special purpose algorithms have been established to find the optimal or heuristic solutions. For example, Bellman and Cooke[5] have developed a dynamic programming algorithm which can find the optimal solution in the small network. Edmonds and Johnson[23] presented algorithms using the optimal matching or network flow techniques to solve the CPP optimally in the directed, un-directed network and mixed network in the special case. Minieka[36] developed an optimal algorithm to find the solution in the general mixed network. The solution methods suggested by Edmonds and Johnson[23] are highly recommended as the results of their computational efficiency[11]. In the following discussion, we will briefly look at the solution method for solving the CPP in the directed network suggested in [23]. Such a technique will be adopted in the method proposed in this thesis for solving the SCVRP.

The method suggested by Edmonds and Johnson[23] is based on the following theorem.

Theorem 1 *A network G is an Euler network if and only if G is connected and every node of G is even.*

Proof: (See Deo[20])

From the above theorem it can be seen that if every node of the given connected network G is even, it follows that the CPP is reduced to finding an Euler tour, which is known to exist on G;

On the other hand, if G contains some odd nodes, then any postman tour in G must contain a set of repeated links $\tilde{A}_{rep}$. If we add $\tilde{A}_{rep}$ artificially to the original network G to construct an augmented network G^*, then G^* must be even and the postman tour in G is equivalent to the Euler tour in G^*. Hence solving the CPP can be divided into two parts:

–determining the optimal repeated link set $\tilde{A}^*_{rep}$, constructing an even augmented network G^*, and

–finding the Euler tour on G^*.

The method assumes that the given network satisfies the following two conditions:

(1) The total network $G = (N, A, C)$ is connected;

(2) There is no proper subset, V, of nodes such that every link meeting one node in V and one node not in V is directed away from/to the node in V.

These two assumptions are the necessary and sufficient conditions for a postman tour to exist in a given network[23]. The solution method is summarized as a three-step algorithm given below.

Let us define

d^+ = number of links incident to a node

d^- = number of links incident out of a node

D^+_{odd} = set of odd nodes with the property that $d^+ > d^-$

D^-_{odd} = set of odd nodes with the property that $d^+ < d^-$

e_{ij} = length of the shortest path from node i ($i \in D^+_{odd}$) to node j ($j \in D^-_{odd}$)

Step 1 Determine the optimal set of links $\tilde{A}^*_{rep}$ by solving the following transportation model:

$$\text{Minimize } \sum_{i,j} e_{ij} y_{ij} \tag{3.5}$$

$$\text{Subject to } \sum_{j} y_{ij} = d^+(i) - d^-(i) \forall i \in D^+_{odd} \tag{3.6}$$

$$\sum_i y_{ij} = d^+(j) - d^-(j) \forall j \in D^-_{odd} \tag{3.7}$$

$$y_{ij} = 0 \text{ or } 1 \tag{3.8}$$

where y_{ij} is the decision variable that is equal to 1 if a shortest path from node i to node j needs to be repeated, and 0 otherwise. $\forall i \in D^+_{odd}, \forall j \in D^-_{odd}$. Constraints (3.6, 3.7) ensure that each node in D^+_{odd} is matched with the node(s) in D^-_{odd} and vice versa.

Step 2 Construct an even augmented network G^* as follows: If node i is matched with node j (i.e. $y_{ij} \neq 0$), then identify the shortest path with the traversal cost e_{ij} in G between i and j and add artificial links on such a path to the origin network G.

Step 3 Find an Euler tour Ω on G* by using any one of Euler tour finding algorithms developed, such as the *end-pairing, next-node, maze-searching*(Edmonds and Johnson[23]) or *cycle-building*(Deo[20]). For instance, the cycle-building algorithm finds an Euler tour in a given G network as follows:

Step (i) Let w be any node in G. All links are not included. Cycles $C = \phi, C' = \phi$.

Step (ii) Start from w and go through the links of G such that no link is traced more than once. Continue to do so until the tracing stops at w (since every node of G is even, we can exit from every node we enter). This forms a new cycle C'. If $C = \phi$ then let $C = C'$ and go to Step (iii). Otherwise go to Step (iv)

Step (iii) Back from w to the first node, say, v, on C with a non-included link incident to it. If no v exits, go to step (v); Otherwise rename v as w and go to step (ii)

Step (iv) Break the original cycle C from w and include C'. This form a new cycle C^{*} which includes more links of G. Rename C^{*} as C and go to Step (iii).

Step (v) C is an Euler tour on G. Terminate

Chapter 4

Proposed Solution Method for SCVRP

This chapter presents the solution method to the SCVRP. The SCVRP is viewed as the combination of two sub-problems: The problem of finding the optimal postman tour (sub-postman tour) on each of the hierarchical subnetworks [1], and the problem of composing all sub-postman tours into the optimal total postman tour. Heuristic solution procedures for solving these sub-problems are illustrated. An numerical example is presented.

4.1 Solution Method

The solution method used for solving the SCVRP presented below is the improved version of that proposed by Alfa and Liu[1]. It is based on decomposing the original problem into sub-problems, finding the solutions

[1]A hierarchical subnetwork is a subnetwork of the total hierarchical network formed from a set of links of certain hierarchy

to these sub-problems, and then composing them into a final solution to the original problem. The original problem is decomposed into the following two sub-problems:

(1) The problem of finding the optimal sub-postman tours $\omega(1), \omega(2), \ldots, \omega(h)$ on each of the subnetworks $G(1), G(2), \ldots, G(h)$ generated from link subsets $A(1), A(2), \ldots, A(h)$ of hierarchies $1, 2, \ldots, h$; and

(2) The problem of combining the sub-postman tours found in (1) into the optimal total postman tour Ω over the entire network G such that it covers each link of G in the priority specified.

The original problem is then re-defined over these two sub-problems and solving the original problem becomes the solutions to the two sub-problems.

The solution method is based on the following assumptions:

(1) The total hierarchical network $G = (N, A, C, P)$ must be connected. Although a hierarchical subnetwork $G(k)(k \in h)$ may not be directly connected, i.e. it consists of $m(h)$ number of components [2] $c_1(h), c_2(h) \ldots, c_{m(h)}(h)$, these components should be connected through links of other hierarchies.

(2) There is no proper subset, V, of nodes in the total network V as well as in each subnetwork $G(h)$, such that every link meeting one node not in

[2] A component is a connected subnetwork of the non-directly connected network

V and one node not in V is directed away from/to the node in V

These two assumptions are the necessary and sufficient conditions for constructing the final postman tour Ω. They ensure that each link of the network is 'accessable' from other links.

Initially, the solution method - which consists of two phases - constructs the hirarchical subnetworks $G(1), G(2), \ldots G(h)$. In Phase I, it determines the optimal sub-postman tours $\omega(1), \omega(2), \ldots, \omega(h)$. This is accomplished by the following: Apply the algorithm for solving the CPP in a directed network suggested by Edmonds and Johnson[23] to each subnetwork and then find the solution. Although such an algorithm requires that the given network must be connected, this is not true for some subnetworks. In order to find the sub-postman tour in a non-directly connected subnetwork, the proposed solution method employs a heuristic procedure developed in [1] to derive a solution. See detailed description in next section.

In Phase II, the method determines the optimal to compose the subpostman tours $\omega(1), \omega(2), \ldots \omega(h)$ into a total postman tour Ω. Since there is no method available in [1] to solve such a problem, a procedure based on the dynamic programming has been developed to find a solution, a detailed description of which can be found in the next section.

The complete solution method is summarized into a four-step algorithm below.

Initialization

STEP 0 Read in the total network $G(N, A, C, P)$.

STEP 1 Generate the corresponding hierarchical subnetworks, $G(1), G(2), \ldots, G(h)$, from hierarchical link subsets $A(1), A(2), \ldots, A(h)$.

Phase I

STEP 2 Arbitrarily select an un-processed subnetwork $G(k)$, go to **STEP 3**; If no such subnetwork exists, go to **STEP 4**.

STEP 3 Check if $G(k)$ is directly connected. If it is, then find the optimal sub-postman tour, $\omega(k)$, on $G(k)$ by applying the CPP algorithm in the directed network suggested by Edmonds and Johnson[23]; otherwise find such a tour by using the procedure for solving the CPP in a non-directly connected subnetwork described in section 4.2.1.

Phase II

STEP 4 Combine the sub-postman tours found in **STEP 3** into the total postman tour Ω over the entire network by using the subpostman tour composing procedure described in section 4.2.2.

The flow chart of the algorithm is shown in Figure 4.1

4.2 Optimization Procedures

This section presents the two special optimization procedures used in the proposed solution method for SCVRP introduced in the last section. The first procedure discussed is for finding the subpostman tour on a non-directly connected subnetwork. The second procedure presented is for combining the sub-postman tours found into the total postman tour that covers the links of the entire network in the priority order specified.

4.2.1 Solving the CPP in a non-directly connected subnetwork

When applying the general CPP algorithm to find a postman tour in a given network, one of the requirements for a solution is that the given network must be connected. Hence when applying the CPP to a subnetwork, $G(h)$, to find a sub-postman tour, one must also ensure that $G(h)$ is connected. The problem, then, is how to find the optimal postman tour, $\omega(h)$, in $G(h)$ if it consists of $m(h)$ components $c_1(h), c_2(h), \ldots, c_{m(h)}(h)$, which are connected only through links of the other hierarchies.

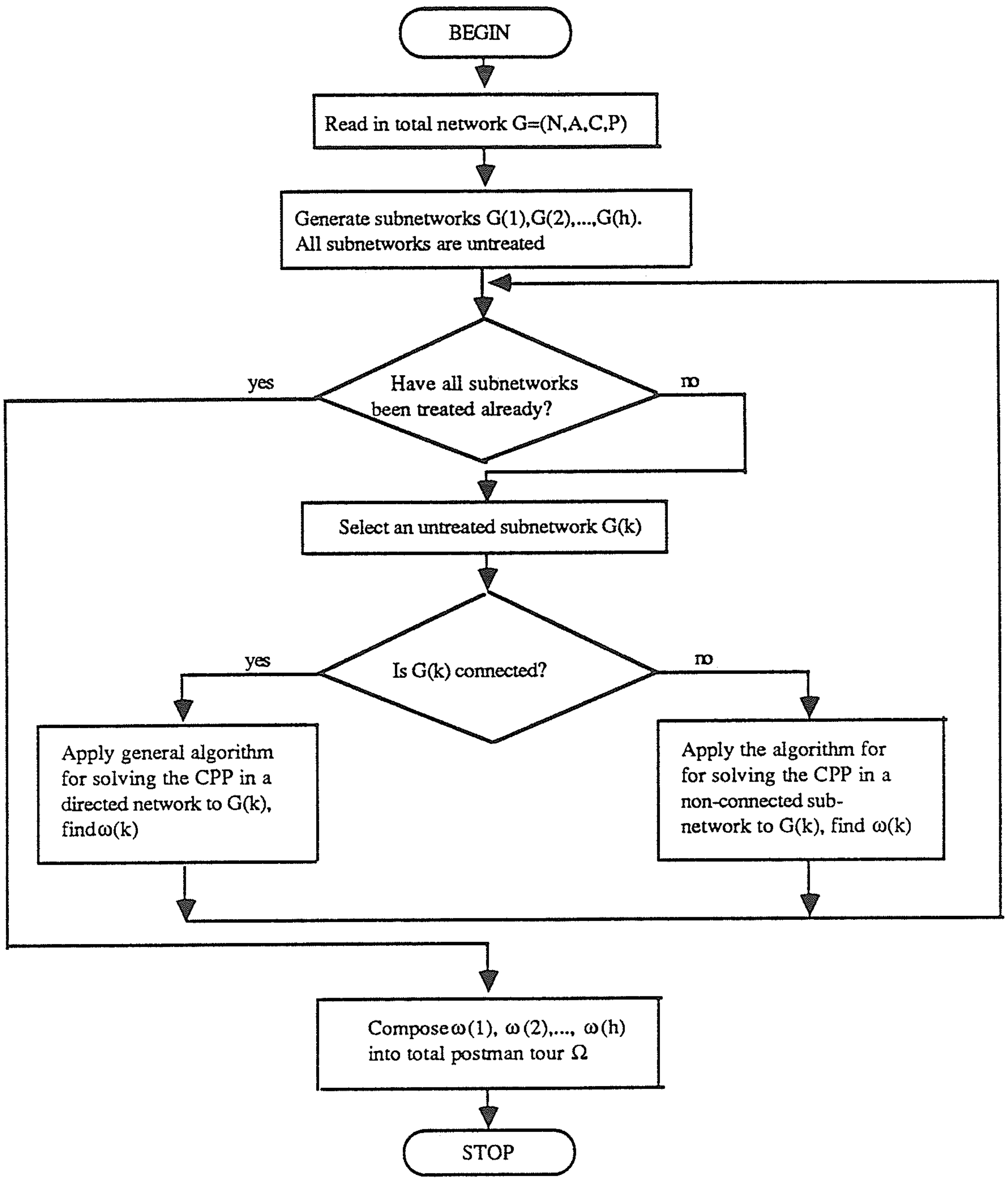


Figure 4.1: Flow chart of the algorithm for solving the SCVRP

In constructing a postman tour on such a network, obviously, two types of deadheading may be incurred. Firstly, the postman has to use a set of links from the other hierarchies to connect the components, because the tour is a continuous walk; Secondly, if the subnetwork contains some odd degree nodes, he must repeat some links in order to traverse each the links of $G(h)$ at least once. Hence solving the CPP in a non-directly connected subnetwork, in general, involves the following:

(1) determining a set of links, $\tilde{A}_{other}(h)$, from the other hierarchies such that by adding $\tilde{A}_{other}(h)$ to the original subnetwork it becomes connected and the additional cost is minimized;

(2) determining a set of links, $\tilde{A}_{rep}(h)$, from the links of the entire network such that by adding $\tilde{A}_{rep}(h)$ to the subnetwork generated from (1), the subnetwork becomes even and connected and the additional cost is minimized; and

(3) finding the Euler tour on the resulting subnetwork.

The most difficult parts are (1) and (2).

In the following we will present an algorithm developed in [1] to solve CPP in a non-directly connected subnetwork. The algorithm finds $\tilde{A}_{other}(h)$ and $\tilde{A}_{rep}(h)$ independently in different steps.

Algorithm

Step 1 Determining $\tilde{A}_{other}(h)$, constructing a connected subnetwork $G^{*}(h)$.

$\tilde{A}_{other}(h)$ can be found by solving the following integer programming problem:

$$\text{Minimize} \quad \sum_{w}\sum_{v} C_{w,v}(h) X_{w,v}(h) \tag{4.1}$$

$$\text{Subject to} \quad \sum_{v} X_{w,v}(h) = 1 \quad 1 \leq w \leq m(h) \tag{4.2}$$

$$\sum_{w} X_{w,v}(h) = 1 \quad 1 \leq v \leq m(h) \tag{4.3}$$

$$\sum_{w \in Q} \sum_{v \in \bar{Q}} X_{w,v}(h) \geq 1 \tag{4.4}$$

$$C_{w,v}(h) > 0; \ X_{w,v}(h) = 0, 1 \tag{4.5}$$

where

$m(h)$ = total number of non-directly connected components in$G(h)$

$C_{w,v}(h)$ = length of the shortest path from component w to v. It is assumed that $C_{w,v} \leq C_{w,r} + C_{r,v}$ $(w, v, r = 1, 2, \ldots, m(h), w \neq v, v \neq r)$

$X_{w,v}(h)$ = decision variable. It is equal to 1 if the shortest path links the component w and v, and equal to 0 otherwise

$Q, \bar{Q}$ = a partition of the integer set $\{1, 2, \ldots, m(h)\}$ such that $Q \cup \bar{Q} = \{1, 2, \ldots, m(h)\}$ and $Q \cap \bar{Q} = \{\phi\}$

Constraints (4.2) and (4.3) ensure that each component will be visited by the postman once and only once. Constraint (4.4) guarantees no subtour will be formed.

The above model is actually the typical integer programming form of the TSP[34]. Hence the connection problem is converted to the TSP and may be solved by utilizing existing techniques for the TSP, provided the shortest distance between each pair of components is known. In [1], $\tilde{A}_{other}(h)$ is determined through the following sub-steps:

Step 1 (i). Initially, let OPEN set $\Gamma = \{\phi\}$, CLOSE set $\Delta = \{\phi\}, I = \{\phi\}, \gamma = \{\phi\}$.

Step 1 (ii). Select arbitrarily two unconnected components in $G(h)$, assign all the nodes on one to set S, all nodes on another set to T.

Step 1 (iii). Let $I = S$, put I on Γ.

Step 1 (iv). Find, if there exist, all immediate successors of I, put them on Γ. Remove I from Γ, put I in Δ. Among all elements in Γ, find γ^* such that $P_{s,\gamma^*} = \min \{P_{s,\gamma}\}(\gamma^*, \gamma \in \Gamma, s \in S)$. $P_{s,\gamma}$ is the shortest path from $s(\forall s \in S)$ to $\gamma(\forall \gamma \in \Gamma)$. If more than one is available, then arbitrarily select one. Note that the immediate successors of I are a set of nodes in Δ which have direct connections to I.

Step 1 (v). If γ is NOT in T, then assign γ to I and go to **Step 1**(iv), otherwise $P_{s,\gamma}$ is the shortest path from S to T. Reverse S and T, find the shortest path from the opposite direction by repeating **Step 1**(iii) to **Step 1**(v).

Step 1 (vi) Repeat **Step 1**(ii) to **Step 1**(v) until the closest nodes and shortest paths between each pair of components are found. The length of the shortest path becomes the shortest distance between two components.

Step 1 (vii) Construct a complete graph [3] $\Phi(h) = (\check{N}, \check{A})$ where $\check{N}$ is a set of nodes denoting the components of $G^{*}(h)$ and $\check{A}$ is a set of links representing the shortest paths between the closest nodes of each pair of components.

Step 1 (vii) Find the optimal Hamilton circuit $H(h)$ on $\Phi(h)$ by using the existing TSP algorithm (e.g Little *et al*[31]). The branches of $H(h)$ indicate which pair of components should be connected.

Step 1 (ix) For each pair of components to be connected, join the closest nodes using the corresponding shortest path. This construct a connected augmented subnetwork $G^{*}(h)$

Step 2 Determine the optimal $\tilde{A}_{rep}(h)$ and construct an even augmented subnetwork $G^{**}(h)$ by using the algorithm for solving the CPP in a

[3]The term 'network' and 'graph' are used interchangeably throughout this thesis

directed network suggested in [23]. Note that such an algorithm is applied over the entire network G.

Step 3 Apply the Euler tour algorithm suggested in [23] to $G^{**}(h)$ to find the minimum postman tour $\omega(h)$.

We must state that the above algorithm is heuristic. It may or may not produce the optimal solution to the overall problem. This is because the procedure for constructing a connected subnetwork $G^{*}(h)$ (**Step 1**), and the procedure for constructing an even augmented subnetwork $G^{**}(h)$ (**Step 2**), are applied in the different steps in problem-solving process. The results of these steps are considered to be independent. However, this is not always true since adding a set of links to connect the components may change the degree of the connection nodes, depending on the nature of connection. In other words, it may affect the length of $\tilde{A}_{rep}(h)$ required to make the subnetwork even. For example, consider a component $c_r(h)$ consists of two even nodes v, v' and u, u' as shown in Figure 4.2 (a), suppose that $c_r(h)$ is connected in two different cases, (a) and (b). Then one can see that in (b) the odd nodes u, u' become even and thus no link requires to repeat; while in (c), on the other hand, the even nodes v, v' become odd and some links must be repeated. Hence the optimal solution to the overall problem is obtained if, and only if, after adding $\tilde{A}^{*}$ to $G(h)$, the total length of the repeated links required to construct an even augmented subnetwork is also the minimum. In section 6.1, we will discuss this problem in more detail and present a method

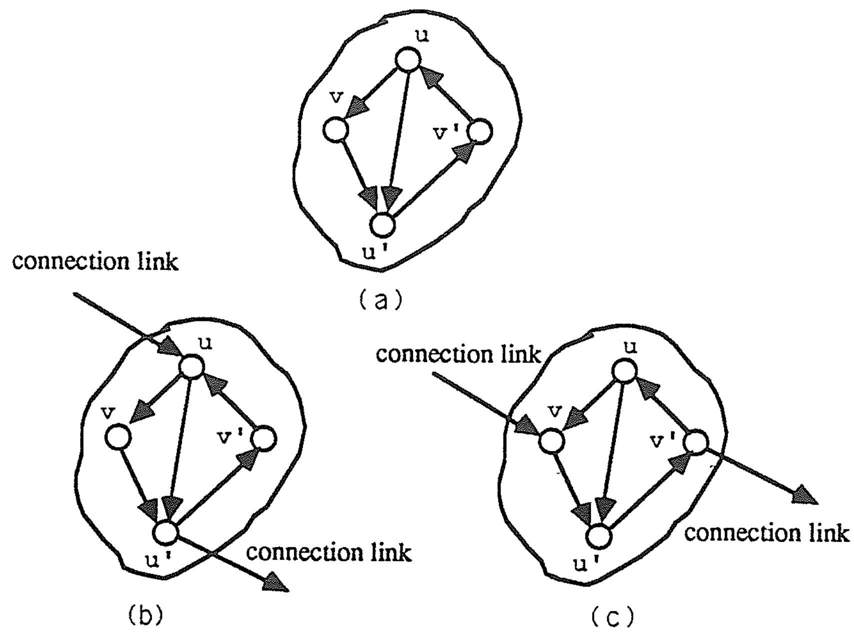


Figure 4.2: Adding a set of links to connect the components may change the degree of the connection nodes, depending on the nature of connection

based on *dynamic programming* which can produce a globally optimal solution.

4.2.2 Composing the sub-postman tours into the total postman tour

Upon finding out the sub-postman tours $\omega(1), \omega(2), \ldots, \omega(h)$ from each of hierarchical subnetworks, the next problem is how to compose them into a total postman tour Ω over the entire network.

Solving the subpostman tour composing problem is relatively easier. It can be easily verified that if the starting node, s_1, of the sub-postman tour $\omega(1)$

is the common node to $\omega(2), \omega(3), \ldots, \omega(h)$, then the problem is solved by selecting s_1 as the starting node for $\omega(2), \omega(3), \ldots, \omega(h)$. THis is because each subpostman tour is a closed walk. However, it is possible that a subpostman tour, $\omega(r)$, for example, may not have a common node to $\omega(1)$. To handle such a case, we select a node in $\omega(1)$ which is common to most of other subpostman tours as s_1 so that the additional cost to visit these subpostman tours are the minimum. For the subpostman tour $\omega(r)$, since we also wish to minimize the additional cost to visit it, we select a node, s_r, as the starting node of $\omega(r)$ such that the traversal cost from s_1 to s_r as well as from s_r to s_1 is the minimum. The shortest path algorithm can be used to find s_r.

We must state that the sub-postman tour composing procedure described in this section is, again, heuristic, even though there exists a node in $\omega(1)$ which is common to all the subpostman tours. Firstly, to form Ω by using the strategy of visiting all links of one sub-postman tour and then going on to the next, may not be the optimal. In fact, one need not cover all links of a subpostman tour and return to the starting node before going on to the next, provided that all the non-repeated links have already been covered. In such a case, leaving the subpostman tour at the ending node of the lastly-visited non-repeated link of the tour, rather than the starting node, one may obtain an improved solution. For example, consider the subpostman tour $\omega(r)$ having such a property shown in Figure 4.3, where s_r is the starting node of $\omega(r)$, s'_r is the ending node of the lastly-visited non-repeated link of $\omega(r)$,

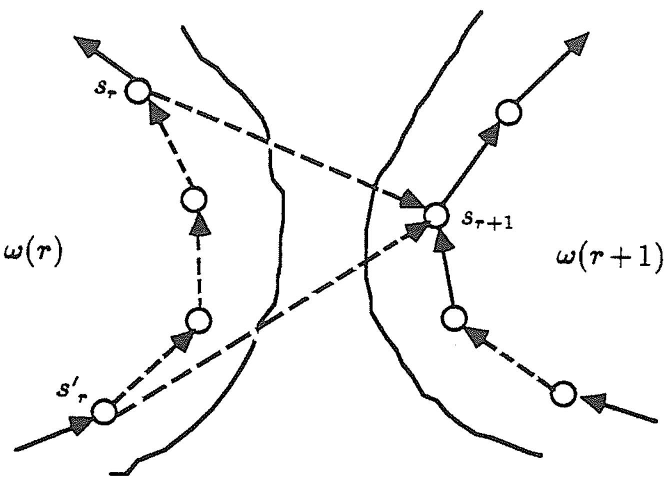


Figure 4.3: Leaving the sub-postman tour at the ending node of the lastly-visited non-repeated link rather than the starting node, one may yield an improved solution

and s_{r+1} is starting node on next subpostman tour $\omega(r+1)$. Obviously, one can leave $\omega(r)$ at node s'_r rather than node s_r if the following is true:

$$e_{s'_r, s_{r+1}} < p_{s'_r, s_r} + e_{s_r, s_{r+1}} \tag{4.6}$$

where

$e_{s'_r, s_{r+1}}$ = the length of shortest path from node s'_r to node s_{r+1}

$p_{s'_r, s_r}$ = the length of the lastly-visited segment of $\omega(r)$ starting from node s'_r and ending at node s_r.

$e_{s_r, s_{r+1}}$ = the length of shortest path from node s_r to node s_{r+1}

We will discuss such a problem in more detail in section 6.2.

The second situation for not guaranteeing the optimal solution to the sub-postman tour composing problem occurs when some sub-postman tours do not have a common node to $\omega(1)$. Although selecting a node, s_1, on $\omega(1)$ which is common to most of the other sub-postman tours can minimize the additional cost to visit these subpostman tours, such a strategy may not be optimal to a subpostman tour, $\omega(r)$, that has no common node to $\omega(1)$. Since we use the node, s_r, on $\omega(r)$ which is the closest to s_1, the position of s_1 determines s_r and hence the additional cost to visit $\omega(r)$. Therefore the best starting node s_1 may not be the one which is common to most other subpostman tours but the one that result in the minimum additional cost to visit $\omega(r)$ as well as to all the other subpostman tours.

4.3 Numerical Example

Consider a hypothetical network shown in Figure 4.4. It has two hierarchy of links: 1 and 2. The length and the hierarchy of each link are shown in the diagram. Suppose we wish to find the optimal postman tour that traverses each link of the network at least once and the links of hierarchy 1 are visited first and then the links of hierarchy 2.

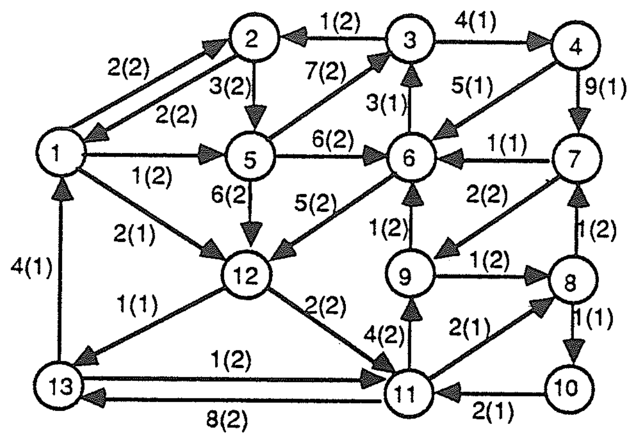


Figure 4.4: A hypothetical hierarchical network

STEP 0 : (Ignored)

STEP 1 : Generating subnetwork $G(1)$ and $G(2)$ from hierarchy 1 links and hierarchy 2 links (Figure 4.6)

STEP 2 : Choosing $G(1)$

STEP 3 : $G(1)$ is not directly connected. It has three components: $c_1(1), c_2(1)$ and $c_3(1)$. Applying algorithm for solving the CPP in a non-connected

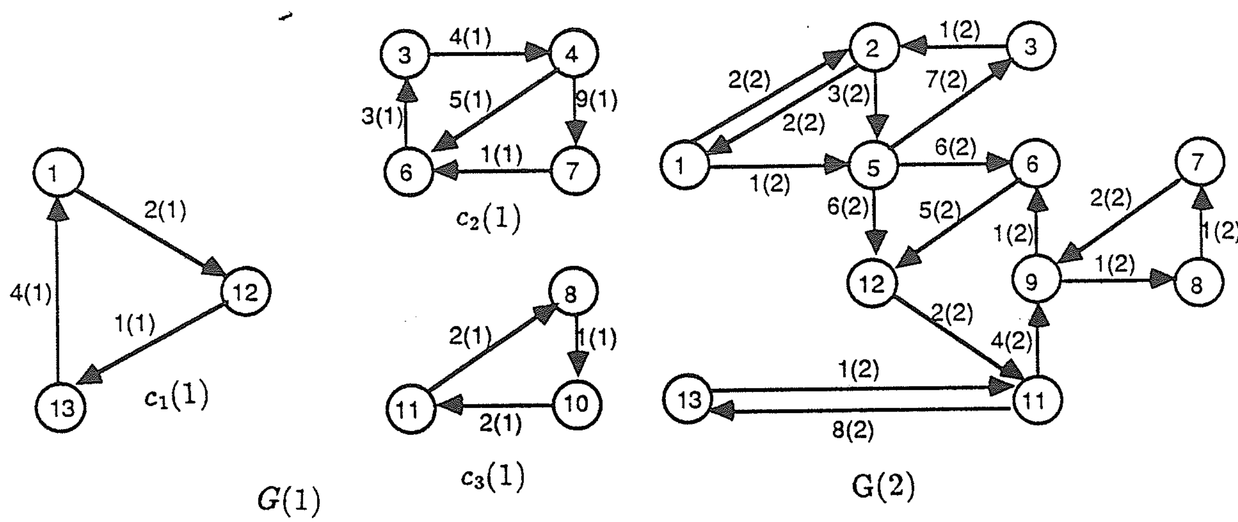


Figure 4.5: Hierarchical subnetwork G(1) and G(2)

subnetwork to find $\omega(1)$:

Step 1 : Determining $\tilde{A}_{other}(1)$ and construct a connected augmented subnetwork $G^*(1)$:

Step 1 (i): $\Gamma = \{\phi\}; \Delta = \{\phi\}; I = \{\phi\}; \gamma = \{\phi\}$

Step 1 (ii): choose components $c_2(1)$ and $c_1(1)$. $S = \{3, 4, 6, 7\}; T = \{1, 12, 13\}$

Step 1 (iii):$I = S = \{3, 4, 6, 7\}; \Gamma = \{3, 4, 6, 7\}$

Step 1 (iv): Immediate successors of I are 2(of 3), 12(of 6) and 9(of 7)

$\Gamma = \{2, 12, 9\}; \Delta = \{3, 4, 6, 7\}$

min.$\{P_{s,2} = 1, P_{s,12} = 5, P_{s,9} = 2\} = P_{s,2} = 1$

therefore $\gamma^* = 2$

Step 1 (v): γ^* is not in T, therefore $I = \{2\}$

Step 1 (iv): Immediate successors of I are 1,5

$\Gamma = \{12, 9, 1, 5\}; \Delta = \{3, 4, 6, 7, 2\}$

min.$\{P_{s,12} = 5, P_{s,9} = 2, P_{s,1} = 3, P_{s,5=4}\} = P_{s,9} = 2$

therefore $\gamma^* = 9$

Step 1 (v): γ^* is not in T, therefore $I = \{9\}$

Step 1 (iv): Immediate successors of I is 8 $\Gamma = \{12, 1, 5, 8\}; \Delta = \{3, 4, 6, 7, 2, 9\}$

min.$\{P_{s,12} = 5, P_{s,1} = 3, P_{s,8} = 3, P_{s,5=4}\} = P_{s,1} = 2$ or $P_{s,8} = 3$

Arbitrarily select 1. therefore $\gamma^* = 1$

Step 1 (v): $\gamma = 1$ is in $T = \{1, 12, 13\}$.

Therefore $C_{2,1} = 3(3 \rightarrow 2 \rightarrow 1)$

Step 1 (v)(vi) In a similar way, we find

$C_{1,2} = 5(13 \rightarrow 11 \rightarrow 8 \rightarrow 7)$ $\quad C_{2,3} = 3(7 \rightarrow 9 \rightarrow 8)$

$C_{3,2} = 1(8 \rightarrow 7)$ $\quad C_{1,3} = 1(13 \rightarrow 11)$

$C_{3,1} = 8(11 \rightarrow 13)$

Step 1 (vii): Using above information we construct a complete graph $\Phi(1)$ (Figure 4.6)

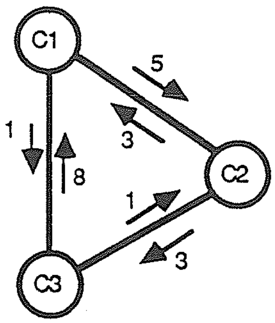


Figure 4.6: Complete graph $\Phi(1)$

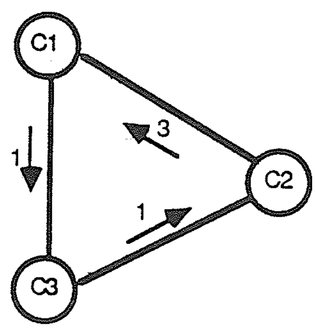


Figure 4.7: Optimal Hamilton circuit $H(1)$

Step 1 (viii) Applying the TSP algorithm (e.g. Little *et al*[31]) to $\Phi(1)$ we find the optimal optimal Hamilton circuit $H(1)$ (Figure 4.7)

Step 1 (ix) From solution found in **Step 1**[viii], we construct a connected augmented subnetwork $G^*(1)$ (Figure 4.8).

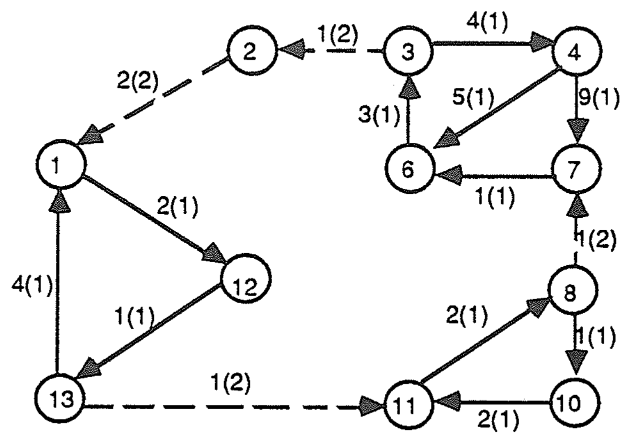


Figure 4.8: Connected augmented subnetwork $G^*(1)$

Step 2 Applying the general algorithm for solving the CPP in a directed network[23]to the entire network G we determine the optimal $\tilde{A}_{rep}$ and construct an even augmented subnetwork $G^{**}(1)$ (Figure 4.9) and find the optimal postman tour $\omega(1)$ (Figure 4.10).

STEP 2 Choosing $G(2)$

STEP 3 $G(2)$ is connected. By Applying the algorithm for solving the CPP in a directed network suggested in [23] to the entire network G we deter-

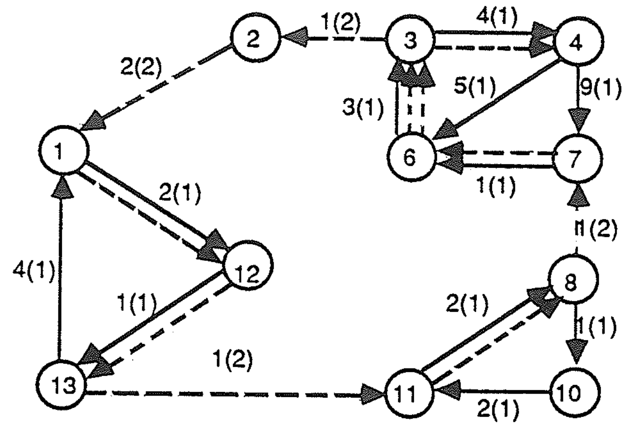


Figure 4.9: Even augmented subnetwork $G^{**}(1)$

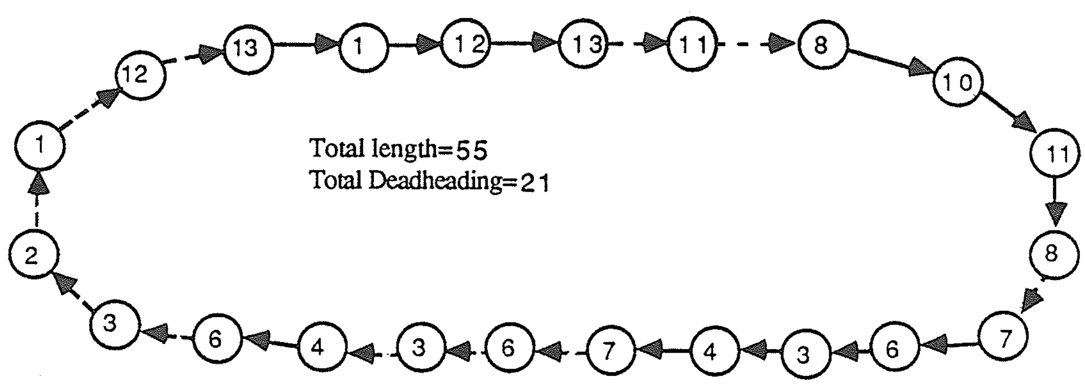


Figure 4.10: Sub-postman tour $\omega(1)$

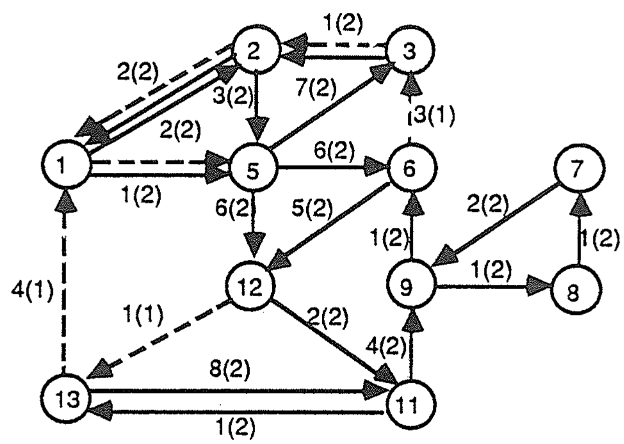


Figure 4.11: Even augmented subnetwork $G^*(2)$

mine the optimal $\bar{A}_{rep}$ and construct an even augmented subnetwork $G^*(2)$ (Figure 4.11), and find the optimal postman tour $\omega(2)$ (Figure 4.12)

STEP 2 All subnetworks have been treated

STEP 4 Composing the subpostman tour $\omega(1)$ and $\omega(2)$ as follows:

Set of nodes in $\omega(1) = \{1, 12, 13, 3, 4, 6, 7, 8, 10, 11\}$

Set of nodes in $\omega(2) = \{1, 2, 3, 5, 6, 12, 7, 8, 9, 11, 13\}$

Set of common nodes to $\omega(1)$ and $\omega(2) = \{1, 3, 6, 12, 7, 8, 11, 13\}$

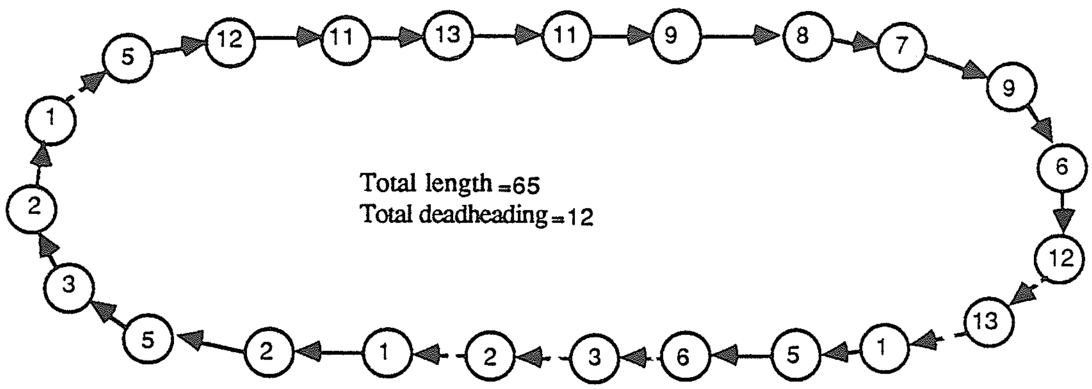


Figure 4.12: Sub-postman tour $\omega(2)$

Since $\omega(1)$ and $\omega(2)$ have nodes in common, from the previous discussions we know that if any one of the common nodes is selected as starting node s_1, then the optimal tour is the tour with $s_2 = s_1$. For example, let $s_1 = s_2 = 6$, we obtain the corresponding total postman tour as shown in Figure 4.13.

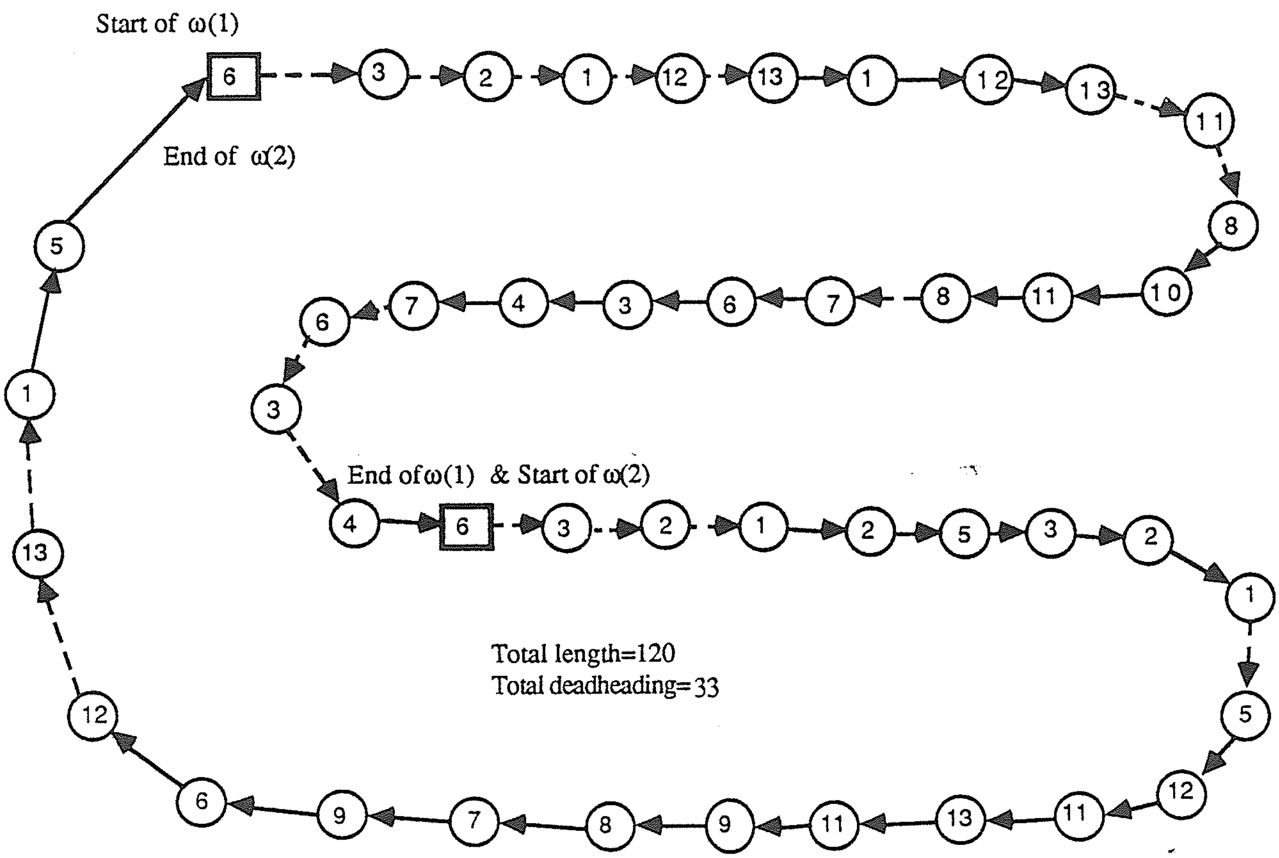


Figure 4.13: Total postman tour Ω

Chapter 5

Solving the District #2 of the City of Winnipeg Example

The method developed in this thesis has been applied to solve the snow clearing vehicle routing problem on Priority II streets (Collectors) and Priority II streets (Residential streets) of the street network system shown in Figure 2.1. The real data for the streets of the total network are listed in Appendix A, C and F. The Priority II class consists of 122 street blocks with 37,700 meters total length to be cleared. The Priority III class consists of 208 street blocks with 89,610 meters total cleared length. All the streets of Priority II and Priority III are two-way streets.

By using the routing strategy developed in this thesis, we obtained the actual vehicle routes for clearing out the streets of these two classes. For Priority II street class, the streets are not all directly connected. They constitute 5 components which are linked by streets of other classes (Appendix D). The actual vehicle route for clearing this street class is shown in Appendix E. It contains 8,322 meters deadheading length.

For the Priority III street class, the street blocks are not all directly connected either. They consists of 54 components (Appendix G). The actual vehicle route for clearing this street class is shown in Appendix H. It contains 18,742 meters deadheading length.

We will not provide the composed vehicle route that contains both the Priority II streets and Priority III streets here. Such a route can be easily obtained by selecting a common node to both the Priority II and Priority III streets as the starting node.

To this moment we have not yet used the vehicle routes obtained in this thesis to perform the actual clearing operations. In order to assess the routing strategy developed in this thesis, we use the following formula to estimate, in theory, the total time required to complete the operations on each priority street class:

$$T_{total}(h) = \frac{L_c(h)}{\bar{v}_c} + \frac{L_d(h)}{\bar{v}_d} \tag{5.1}$$

where

$T_{total}(h)$ = total time required to complete the operations on priority h streets

$L_c(h)$ = total cleared length for priority h streets.

$L_d(h)$ = total deadheading length for priority h streets

$\bar{v}_c$ = average clearing speed

$\bar{v}_d$ = average deadheading speed

From the City of Winnipeg, we obtained that the average clearing speed $\bar{v}_c$ is $5 - 7km/h$ and the average deadheading speed $\bar{v}_d$ is $20 - 30km/h$. Hence using the routing strategy developed in this thesis, the total time required to complete the operations on Priority II & III streets, in theory, are

$$T_{total}(1) = \frac{L_c(1)}{\bar{v}_c} + \frac{L_d(1)}{\bar{v}_d} = \frac{37.700}{6} + \frac{8.322}{25} = 6.56(hrs) \tag{5.2}$$

for the Priority II streets and

$$T_{total}(2) = \frac{L_c(2)}{\bar{v}_c} + \frac{L_d(2)}{\bar{v}_d} = \frac{89.61}{6} + \frac{18.742}{25} = 15.56(hrs) \tag{5.3}$$

for the Priority III streets.

In the City of Winnipeg, the maximum allowed completion times for Priority II streets and Priority III streets are $48hrs$ and $60hrs$, respectively. Hence by using the routing strategy developed in this thesis, the total completion time required, in theory, are only about 14% and 26% of the maximum allowed completion times for these two street classes.

The results are summarized in Table 5.1.

Table 5.1: Results of using the proposed strategy to solve the SCVRP on District #2 streets

Priority	Total Deadheading	Total Cleared Length	Completion time	
			Max. allowed	Estimated
I	N/A	117,416m	24hrs	N/A
II	8,322m	37,700m	48hrs	6.56hrs
III	18,742m	89,610m	60hrs	15.56hrs

Chapter 6

Discussion of the Solution Method

We have already presented a complete solution method to solve the SCVRP in chapter 4. Such a method, as we have pointed out, is only heuristic. This is because the methods for solving the CPP in the non-directly connected subnetwork and subpostman tour composing problem may or may not produce the optimal solutions, depending on the structure of the network. In this chapter, we will discuss these problems in more detail and explore the optimal solution method to solve them. We will give particular attention to the CPP in the non-directly connected subnetwork since such a problem is met very frequently in the SCVRP. The discussion will be divided into two sections. In Section 6.1, the solution method for solving the CPP in a non-directly connected subnetwork, that is, finding the optimal postman tour in a subnetwork consisting of a number of components which are connected only through links of other classes, will be discussed. We will first examine the heuristic solution methods existing in literature to solve such a problem and then present a method based on *dynamic programming* which can solve the problem optimally. All these methods will be tested in a small example network shown in Figure 4.4. The computational efficiency and solution quality are then compared. In Section 6.2, the solution method to the subpostman tour composing problem will be studied. We will present a method to obtain the optimal solution to such a problem.

6.1 Discussion of the Solution Methods to the CPP in A Non-connected Subnetwork

6.1.1 Heuristic methods

Apart from the effort made by Alfa and Liu[1] to solve the CPP in the non-directly connected subnetwork, Orloff[39], in the Rural Postman Problem (RPP) and Tucker and Bodin[51], in the street sweeping vehicle routing problem have also dealt with a similar problem. They both presented solution methods which are different from that suggested in [1] to solve such a problem. Unfortunately, their methods, are heuristic and can obtain the optimal solution only in special cases. The methods suggested in [39] and [51] are described below.

Orloff's method

The problem-solving scheme suggested by Orloff[39] is similar to that suggested in [1] in deriving a solution, that is, solve the problem through three major steps:

–a step for determining an optimal set of repeated links and constructing an even augmented subnetwork (making-even step);

–a step for determining an optimal set of connection links and forming a connected augmented subnetwork (connecting step); and

–a step for finding an Euler tour (finding-Euler-tour step).

Different from [1] where the connecting step is taken first and then followed by the making-even step, in [19], these two steps are executed in the reverse order. The method employed by Orloff[39] for determining the optimal set of repeated links and constructing an even augmented network is the same as that for solving the CPP in a connected network suggested in [23], except it is applied to the entire network. In other words, the links not belonging to the subnetwork may be used to make the odd nodes even. If the even augmented subnetwork is now connected, then the optimal solution to the problem can be found by applying the procedure for finding the Euler tour, since it satisfies the two conditions for an Euler tour to exist[23]. However, when the even augmented subnetwork still remains un-connected, then a set of links to connect the remaining un-connected components is determined by using a method based on a subtour elimination technique similar to that used to solve the TSP. The method is summarized in an algorithm displayed below.

Step 1 Construct an even augmented subnetwork $G^{*}(h)$ from the original subnetwork $G(h)$ using the algorithm for solving the CPP in the directed network suggested in [23]. Note that such an algorithm is applied to the entire network G.

Step 2 Check whether $G^{*}(h)$ is connected. If it is then go to **Step 4**; Otherwise go to **Step 3**.

Step 3 Construct an even, connected augmented subnetwork $G^{**}(h)$ as follows: When duplicating some shortest paths between odd-degree nodes on $G^{*}(h)$, let one of them cross a node of the even component, say $c_k(h)$, which remains un-connected after **Step 1**. Find the least cost way to do so by using the subtour elimination algorithm similar to that for solving the TSP [1]. If $c_k^{(h)}$ is far from the rest of $G^{*}(h)$, then use any node of $c_k(h)$ as the connection node.

Step 4 Find an Euler tour, $\omega(h)$, on $G^{*}(h)$ using the algorithm for the Euler tour (e.g. Edmonds and Johnson[23]'s algorithm).

[1]No detail solution method was provided in [35]

Example

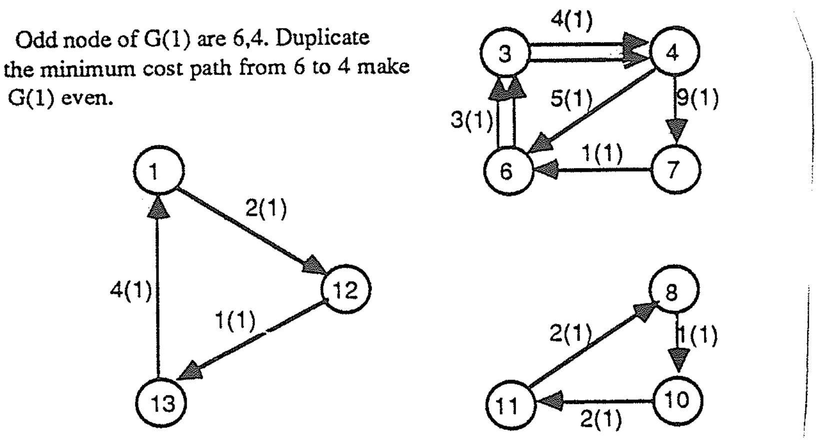
Consider the example network shown in Figure 4.5. We wish to find subpostman tour $\omega(1)$ on $G(1)$. By applying the above algorithm we construct a subpostman tour of G(1) as shown in Figure 6.1. The total deadheading is 19 units.

Shortcomings

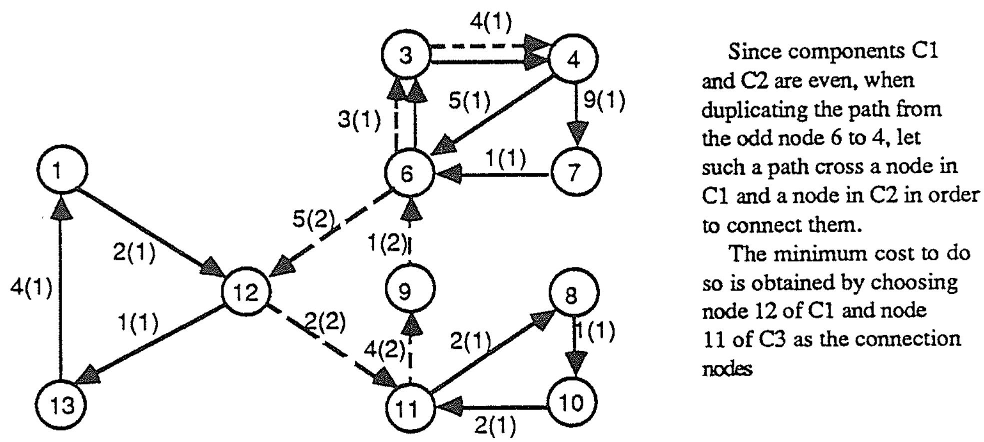
The shortcomings of Orloff[39]'s method can be identified as follows:

1. The solution quality obtained is dependent on the structure of $G(h)$. No optimal solution is guaranteed if $G^*(h)$ remains un-connected after **Step 1**. This is due to the fact that connecting an even component in other ways may yield an improved solution. See for example the dynamic programming method described in Section 6.1.2.

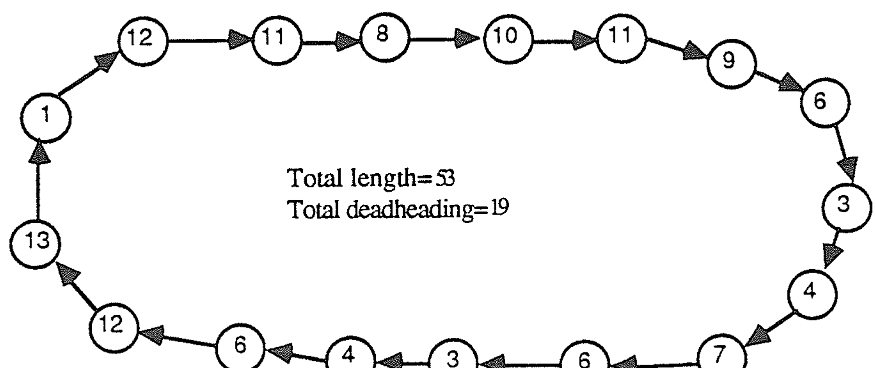
2. The computation becomes exceedingly difficult when more than one component of $G(h)$ is even. This is due to the fact that the possible ways to connect these components would be a very large number.



(a) Subnetwork G*(1)



(b) An even, connected augmented subnetwork G**(1)



(c) Sub-postman tour ω(1)

Figure 6.1: Subpostman tour $\omega(1)$ obtained by using Orloff's method

Tucker and Bodin's method

Tucker and Bodin[51]'s method to solve the CPP in the non-directly connected subnetwork is quite similar to [39], that is, it employs the same problem-solving scheme to construct a solution. However, it uses, in the connecting step, a different technique based on the *Minimum Spanning Tree* to find the optimal set of links to connect the remaining un-connected components. The method can be described as the following algorithm:

Step 1 Construct an even augmented subnetwork $G^{*}(h)$ from the origin subnetwork $G(h)$ using the algorithm for solving the CPP in the directed network suggested in [23]. Note that such an algorithm is applied to the entire network G.

Step 2 Check whether $G^{*}(h)$ is connected. If it is, then go to **Step 4**. Otherwise go to **Step 3**.

Step 3 Construct an even, connected augmented subnetwork $G^{**}(h)$ from $G^{*}(h)$.

Step 3 (i) Determine the closed nodes between each pair of components in $G^{*}(h)$ using a geographical estimate called the "Manhattan" estimate (See detailed description in [51]). Construct a complete graph $\Phi(h) = (\check{N}, \check{A})$ where $\check{N}$ is a set of nodes denoting the components of $G^{*}(h)$ and $\check{A}$ is a set of links representing the shortest distance between each pair of the components.

Step 3 (ii) Find the minimum spanning tree $T(h)$ on $\Phi(h)$. The branches of $T(h)$ indicate which pair of components should be connected.

Step 3 (iii) Using the solution found in (ii) construct an an even, connected augmented subnetwork $G^{**}(h)$ as follows: For each pair of components to be connected, identify the shortest paths between the closest nodes in *each* direction by using the shortest path algorithm and join the closest nodes using these paths.

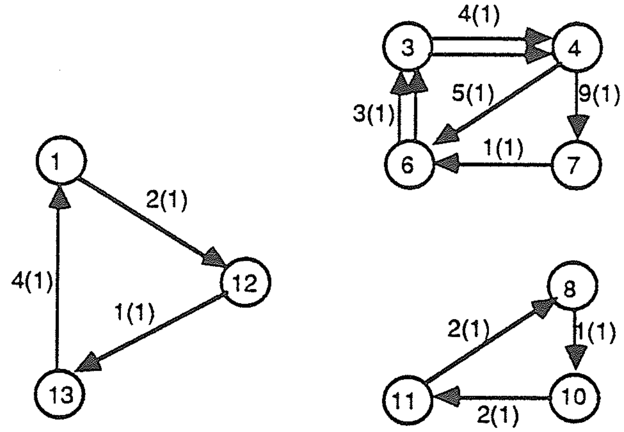
Step 4 Find an Euler tour $\omega(h)$ on $G^{*}(h)$ using the Euler tour finding algorithm (e.g. Edmonds and Johnson[23]'s algorithm).

Example

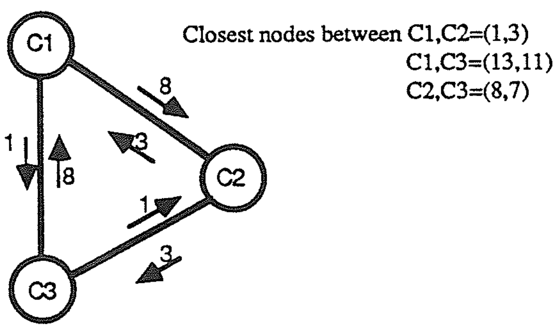
Using the above algorithm we find the solution (Figure 6.2(a), (b),(d)) to the same sample problem shown in Figure 4.4 The total deadheading is 23 units. Note that the method for determining the closest nodes and the shortest path between components developed in [1] is employed since the geographical estimate is not applicable in this case.

Shortcomings

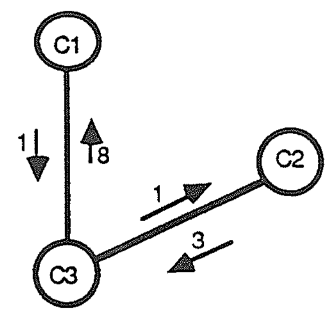
The shortcomings of Tucker and Bodin[50]'s method appear in the following two aspects:



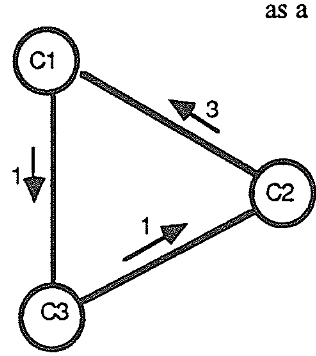
(a) Subnetwork G*(1)



(b) A complete graphΦ (1) for linking the components of G(1)

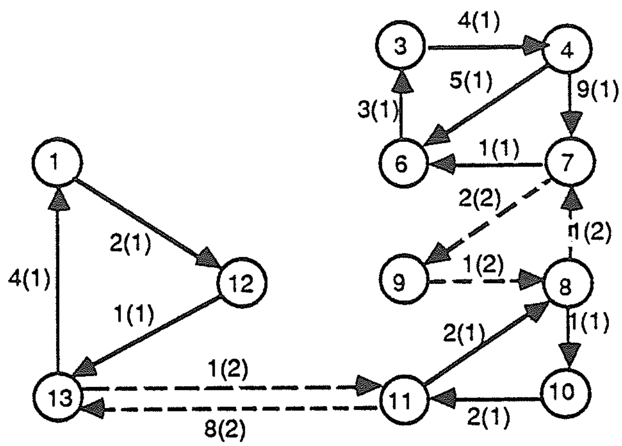


(c) The Minimum Spanning tree T(1). Note that two links connecting a pair of components is viewed as a tree branch.

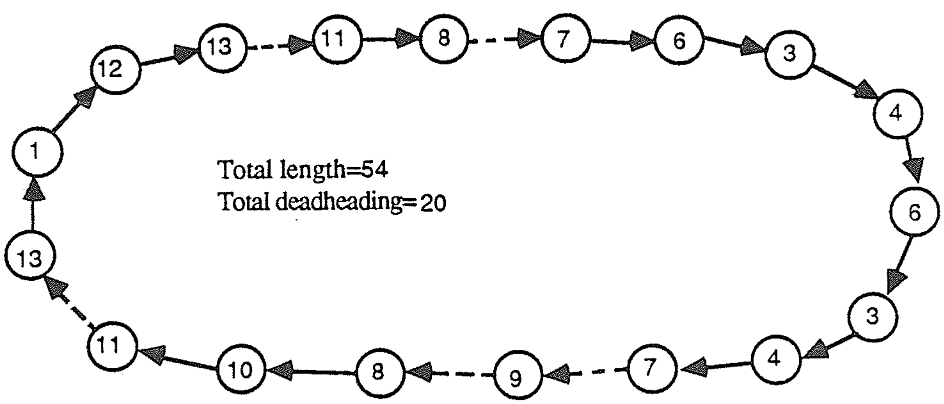


(d) A solution which is not a tree may yield an improved solution

Figure 6.2: Subpostman tour $\omega(1)$ obtained by using Tucker and Bodin's method



(e) An even, connected subnetwork G**(1)



(f) Subpostman tour ω(1)

Figure 6.2: continued

1. The method used to find the shortest distance between components may not be accurate enough and may become very inefficient when a component contains a large number of nodes.

2. Finding the optimal set of links to connect the components by using the Minimum Spanning Tree solution may not be optimal. This is because a solution which is not a tree (Figure 6.2 (c)) may yield an improved solution.

The approaches for solving the CPP in the non-directly connected subnetwork discussed so far are only heuristic and the solution quality is dependent upon the nature of the subnetwork. The main reason for not guaranteeing the optimal solution is that the optimization procedure is divided into the making-even step, the connecting step, and the finding-Euler tour step and they are taken independently. The nodes used as the connection nodes are restricted in a certain subset of nodes in the components; either the closest nodes between components (e.g. Alfa and Liu[1] and Bodin and Tucker[51]), or the odd nodes in the odd-components. However, the connection nodes, can be any nodes in a component. This can be proved as follows: Suppose in a subnetwork $G(h)$ consisting of $m(h)$ components $c_1(h), c_2(h), \ldots, c_{m(h)}(h)$, one wishes to construct an optimal postman tour $\omega(h)$. Since such a tour is a round trip we may start at any component in analyses. Without losing generality, let us select component $c_1(h)$, for example, to start with. Now the problem is how to select a node in $c_1(h)$ to start with the tour. Suppose we arbitrarily select the node, s_1, as the starting node. We can see that after we traverse each link of $c_1(h)$ at least once via an optimal tour we may end up with either s_1 when s_1 is even or another odd node s_1' when s_1 is odd. Let us denote $\sigma_{s_1,s_1'}(h)$ as the length of such an optimal tour and let $\eta_{s_1',s_1}(h)$ be the length of the optimal partial tour of $\omega(h)$ starting from s_1' that visits each link of the remaining components $c_{j_1}(h), c_{j_2}(h), \ldots c_{j_{m(h)-1}}(h)$ at least once and finally returns to s_1. Obviously, the length of the optimal tour $\omega(h)$ is determined by

$$| \omega(h) |= \min_{\forall s_1}[\sigma_{s_1,s_1'}(h) + \eta_{s_1',s_1}(h)] \tag{6.1}$$

Therefore, any node, not only the closest node between two components (i.e. Tucker and Bodin[51], Alfa and Liu[1]) or the odd nodes (i.e. Orloff[39]), can be used as the connection node, provided the rest of the postman tour to be constructed is optimal.

6.1.2 Optimal method – dynamic programming approach

This section presents an optimal solution method to solve the CPP in a non-directly connected subnetwork. The method is based on the dynamic programming approach which is similar to the one for solving the Travelling

Salesman Problem(TSP)[3].

Consider the CPP in a non-directly connected subnetwork $G(h)$ consisting of $m(h)$ components $c_1(h), c_2(h), \ldots, c_{m(h)}(h)$ as a multi- stage decision-making problem. Without losing generality, since the solution $\omega(h)$ is a round trip, it can be viewed as a closed walk which starts from a node of certain component, visits each of its links at least once, and then the next un-visited component and visits each of its links at least once, and so forth until it finishes visiting all the remaining un-visited components, and finally returns to the node of the beginning component. We may start at any component, say $c_1(h)$, in analyses. Suppose at some stage $k (k = 1, 2, \ldots, m(h))$, the optimal tour, $\omega(h)$, starting from the starting node, $s_1^{(1)}$, of component $c_1(h)$ one has reached the starting node, $s_i^{(k)}$, of the component $c_i(h)$. There are $(m(h) - k)$ components $c_{j_1}(h), c_{j_2}(h), \ldots, c_{j_{m(h)-k}}(h)$ remaining to be visited before returning to $s_1^{(1)}$. Then $\omega(h)$ is optimal if, and only if, the length of its partial tour starting from $s_i^{(k)}$ that visits each link of $c_i(h)$ at least once and then the remaining components $c_{j_1}(h), c_{j_2}(h), \ldots, c_{j_{m(h)-k}}(h)$ and finally returns to $s_1^{(1)}$ is optimal. Otherwise one may always obtain an improved tour by choosing the least-cost partial tour starting from $s_i^{(k)}$ that visits each link of $c_i(h)$ at least once and then the remaining components $c_{j_1}(h), c_{j_2}(h), \ldots, c_{j_{m(h)-k}}(h)$ and finally returns to $s_1^{(1)}$.

Let us define

k =stage. $k = 1, 2, \ldots, m(h)$. At stage 1, the postman from starting node $s_1^{(1)}$ begins to visit the links of $c_1(h)$.

$s_i^{(k)}$ = state variable on stage k. It is the starting node on component $c_i(h)(i = 1, 2, \ldots, m(h)$ and $i \neq 1$ if $k \neq 1)$, where the postman tour $\omega(h)$ begins to visit links of component c_i and then the remaining components $c_{j_1}(h), c_{j_2}(h), \ldots, c_{j_{m(h)-k}}(h)$ and finally returns to $s_1^{(1)}$. A set of decision variables $S_i^{(k)}$ is a set of nodes which are to be selected as the starting node on component $c_i(h)$

$u_{j_n}^{(k)}(s_i^{(k)})$ = decision variable on stage k. It is the selection of the starting node on the component $c_{j_n}(h)(j_n = j_1, j_2, \ldots, j_{m(h)-k})$, given that the node $s_i^{(k)}$ on component $c_i(h)$ has been used as the starting node on $\omega(h)$. A set of decision variables $U_{j_n}^{(k)}(s_i^{(k)})$ is a set of nodes which are to be selected as the starting nodes on component c_{j_n}

$f^{(k)}(c_i(h); c_{j_1}(h), c_{j_2}(h), \ldots, c_{j_{m(h)-k}}(h)) \mid_{s_i^{(k)}, s_1^{(1)}}$= length of the optimal partial tour of $\omega(h)$ at stage k. It starts from $s_i^{(k)}$ and visits each link of $c_i(h)$ at least once and then the $m(h) - k$ remaining components $c_{j_1}, c_{j_2}, \ldots, c_{j_{m(h)-1}}$ and finally returns to $s_1^{(1)}$

$d_{ij_n}(s_i^{(k)}, u_{i,j_n}^{(k)}(s_i^{(k)}))$ = length of the shortest path starting from node $s_i^{(k)}$ on $c_i(h)$ that passes through each link of $c_i(h)$ at least once and reaches $u_{j_n}^{(k)}(s_i^{(k)})$ of component c_{j_n}

With the above remarks the CPP in a non-directly connected subnetwork can be defined as the following typical dynamic programming problem:

$$\begin{aligned} f^{(k)}(c_i(h); c_{j_n}(h), c_{j_n}(h), \ldots, c_{j_{m(h)-k}}(h)) \mid_{s_i^{(k)}, s_1^{(1)}} = \\ \min_{1 \le k \le m(h)} \{ \min_{u_{j_n}^{(k)}(s_i^{(k)})} \{ d_{ij_n}(s_i^{(k)}, u_{j_n}^{(k)}(s_i^{(k)})) + f^{(k+1)}(c_i(h); c_{j_1}(h), c_{j_2}(h), \ldots, \\ c_{j_{n-1}}(h), c_{j_{n+1}}(h), \ldots, c_{j_{m(h)-k}}(h)) \mid_{u_{j_n}^{(k)}, s_1^{(1)}} \}\} \end{aligned} \tag{6.2}$$

Applying the above equation recursively we begin with

$$f^{(m(h))}(c_i(h)) \mid_{s_i^{(m(h))}, s_1^{(1)}} = d_{i1}(s_i^{(m(h))}, s_1^{(1)}) \ \forall i \neq 1 \tag{6.3}$$

and terminate with

$$f^{(1)}(c_1(h); c_{j_1}(h), c_{j_2}(h), \ldots, c_{j_{m-1}}(h)) \mid_{s_1^{(1)}, s_1^{(1)}} \tag{6.4}$$

The length of the optimal path is

$$\mid \omega(h) \mid = \min_{\forall s_1^{(1)}} \{ f^{(1)}(c_1(h); c_{j_1}(h), c_{j_2}(h), \ldots, c_{j_{m-1}}(h)) \mid_{s_1^{(1)}, s_1^{(1)}} \} \tag{6.5}$$

and the optimal tour is the one having such a length.

Example

Using the dynamic programming approach, we solve the CPP in $G(1)$ of the network shown in Figure 4.4. The steps are displayed below. (note that (h) is omitted if there is no ambiguity)

Number of component $m = 3 \;\; (c_1, c_2, c_3)$

Select c_1 as the starting component $S_1^{(1)} = \{s_1^{(1)}\} = \{1, 12, 13\}$

(1) $s_1^{(1)} = 1$

At stage $k = 3$

$m - k = 3 - 3 = 0$ No component left to be visited. After visiting all links of c_i, one returns to $s_1^{(1)}$

Calculate $f^{(3)}(c_i) \mid_{s_i^{(3)}, s_1^{(1)}}$

Select c_i:

$k \neq 1, \quad c_i \neq c_1, \;\; c_i = c_2 \;\; \text{or} \;\; c_i = c_3$

(i) $c_i = c_2 \quad S_2^{(3)} = \{s_2^{(3)}\} = \{3, 4, 6, 7\}$

$$f^{(3)}(c_2)\mid_{s_2^{(3)},s_1^{(1)}} = d_{21}(s_2^{(3)}, s_1^{(1)}) = \begin{cases} f^{(3)}(c_2)\mid_{s_2^{(3)}=3,s_1^{(1)}=1} = d_{21}(3,1) = 32 \\ f^{(3)}(c_2)\mid_{s_2^{(3)}=4,s_1^{(1)}=1} = d_{21}(4,1) = 28 \\ f^{(3)}(c_2)\mid_{s_2^{(3)}=6,s_1^{(1)}=1} = d_{21}(6,1) = 35 \\ f^{(3)}(c_2)\mid_{s_2^{(3)}=7,s_1^{(1)}=1} = d_{21}(7,1) = 36 \end{cases}$$

(ii) $c_i = c_3 \quad S_3^{(3)} = \{s_3^{(3)}\} = \{8, 10, 11\}$

$$f^{(3)}(c_3)\mid_{s_3^{(3)},s_1^{(1)}} = d_{31}(s_3^{(3)}, s_1^{(1)}) = \begin{cases} f^{(3)}(c_3)\mid_{s_3^{(3)}=8,s_1^{(1)}=1} = d_{31}(8,1) = 13 \\ f^{(3)}(c_3)\mid_{s_3^{(3)}=10,s_1^{(1)}=1} = d_{31}(10,1) = 16 \\ f^{(3)}(c_3)\mid_{s_3^{(3)}=11,s_1^{(1)}=1} = d_{31}(11,1) = 15 \end{cases}$$

<u>At stage $k = 2$</u>

$m - k = 3 - 2 = 1$ 1 component left to be visited.

Calculate $f^{(2)}(c_i; c_{j_1})\mid_{s_i^{(2)},s_1^{(1)}}$

Select $c_i,\ c_{j_1}$

$k \neq 1 \quad c_i \neq c_1 \quad c_{j_1} \neq c_1$ Components other than c_1 are c_2 and c_3

Therefore $c_i = c_2; c_{j_1} = c_3$ or $c_i = c_3; c_{j_1} = c_2$

From equation 6.2 we have

(i) $c_i = c_2; c_{j_1} = c_3 \quad S_2^{(2)} = \{s_2^{(2)}\} = \{3, 4, 6, 7\} \quad U_3^{(2)}(s_2^{(2)}) = \{u_3^{(2)}(s_2^{(2)})\} = \{8, 10, 11\}$

From equation (6.2) we have

$$f^{(2)}(c_2; c_3)\mid_{s_2^{(2)}, s_1^{(1)}} = \min_{\forall u_3^{(2)}(s_2^{(2)})} \{d_{23}(s_2^{(2)}, u_3^{(2)}(s_2^{(2)}) + f^{(3)}(c_3)\mid_{u_3^{(2)}(s_2^{(2)}), s_1^{(1)}}\}$$

For $s_2^{(2)} = 3$,

$$f^{(2)}(c_2; c_3)\mid_{s_2^{(2)}=3, s_1^{(1)}=1} = \min \left\{ \begin{array}{l} d_{23}(3,8) + f^{(3)}(c_3)\mid_{8,1} = 35 + 13 = 48 \\ d_{23}(3,10) + f^{(3)}(c_3)\mid_{10,1} = 36 + 16 = 42 \\ d_{23}(3,11) + f^{(3)}(c_3)\mid_{11,1} = 33 + 15 = 48 \end{array} \right\} = 48$$

For $s_2^{(2)} = 4, 6, 7$, in a similar way, we obtain:

$f^{(2)}(c_2; c_3)\mid_{s_2^{(2)}=4, s_1^{(1)}=1} = 44 \quad f^{(2)}(c_2; c_3)\mid_{s_2^{(2)}=6, s_1^{(1)}=1} = 51$

$f^{(2)}(c_2; c_3)\mid_{s_2^{(2)}=7, s_1^{(1)}=1} = 45$

(ii) $c_i = c_3; c_{j_1} = c_2 \quad S_3^{(2)} = \{s_3^{(2)}\} = \{8, 10, 11\} \quad U_2^{(2)}(s_3^{(2)}) = \{u_2^{(2)}(s_3^{(2)})\} = \{3, 4, 6, 7\}$

From equation (6.2) we have

$$f^{(2)}(c_3; c_2)\mid_{s_3^{(2)}, s_1^{(1)}} = \min_{\forall u_3^{(2)}(s_3^{(2)})} \{d_{32}(s_3^{(2)}, u_2^{(2)}(s_3^{(2)}) + f^{(3)}(c_3)\mid_{u_2^{(2)}(s_3^{(2)}), s_1^{(1)}}\}$$

For $s_3^{(2)} = 8$ we have:

$$f^{(2)}(c_3; c_2) \mid_{s_3^{(2)}=8, s_1^{(1)}=1} = \min \left\{ \begin{array}{l} d_{32}(8,3) + f^{(2)}(c_3) \mid_{3,1} = 10 + 32 = 42 \\ d_{32}(8,4) + f^{(2)}(c_3) \mid_{4,1} = 14 + 28 = 42 \\ d_{32}(8,6) + f^{(2)}(c_3) \mid_{6,1} = 7 + 35 = 42 \\ d_{32}(8,7) + f^{(2)}(c_3) \mid_{6,1} = 6 + 36 = 42 \end{array} \right\} = 42$$

In a similar way, for $s_3^{(2)} = 10, 11$ we obtain:

$f^{(2)}(c_3; c_2) \mid_{s_3^{(2)}=10, s_1^{(1)}=1} = 46$ $\quad f^{(2)}(c_2; c_3) \mid_{s_2^{(2)}=7, s_1^{(1)}=1} = 44$

At stage $k = 1$

$m - k = 3 - 1 = 2$ 2 component left to be visited. The next starting node is $s_i^{(1)}$

Calculate $f^{(2)}(c_i; c_{j_1}, c_{j_2}) \mid_{s_i^{(1)}, s_1^{(1)}}$

Select c_i, c_{j_1}, c_{j_2}

$k = 1$ $\quad c_i = c_1$ $\quad c_{j_1} \neq c_1$ since component other than c_1 are c_2 and c_3

therefore $c_{j_1} = c_2; c_{j_2} = c_3$ or $c_{j_1} = c_3; c_{j_1} = c_2$

The length of the optimal tour with the starting node $s_1^{(1)}$ is

$$f^{(1)}(c_1; c_{j_1}, c_{j_2}) \mid_{s_1^{(1)}=1, s_1^{(1)}=1} = \min \left\{ \begin{array}{l} \min\limits_{\forall u_2^{(1)}(s_1^{(1)})} \{ d_{12}(s_1^{(1)}, u_2^{(1)}(s_1^{(1)}) + f^{(2)}(c_2; c_3) \mid_{u_2^{(1)}(s_1^{(1)}), s_1^{(1)}=1} \} \\ \min\limits_{\forall u_3^{(1)}(s_1^{(1)})} \{ d_{13}(s_1^{(1)}, u_3^{(1)}(s_1^{(1)})) + f^{(2)}(c_3; c_2) \mid_{u_3^{(1)}(s_1^{(1)}), s_1^{(1)}=1} \} \end{array} \right\}$$

(i) $c_i = 1, c_{j_1} = c_2, c_{j_2} = c_3 \quad \{s_1^{(1)}\} = \{1\}, \{u_2^{(1)}(s_1^{(1)})\} = \{3, 4, 6, 7\}$

$$f^{(1)}(c_1; c_2, c_3)\mid_{s_1^{(1)}=1, s_1^{(1)}=1} = \min \left\{ \begin{array}{l} d_{12}(1,3) + f^{(2)}(c_2; c_3)\mid_{3,1} \\ d_{12}(1,4) + f^{(2)}(c_2; c_3)\mid_{4,1} \\ d_{12}(1,6) + f^{(2)}(c_2; c_3)\mid_{6,1} \\ d_{12}(1,7) + f^{(2)}(c_2; c_3)\mid_{7,1} \end{array} \right\}$$

$$= \min \left\{ \begin{array}{l} 15 + 48 = 63 \\ 19 + 44 = 63 \\ 14 + 51 = 65 \\ 14 + 45 = 59 \end{array} \right\} = 59$$

(ii) $c_i = 1, c_{j_1} = c_3, c_{j_2} = c_2 \quad \{s_1^{(1)}\} = \{1\}, \{u_3^{(1)}(s_1^{(1)})\} = \{8, 10, 11\}$

$$f^{(1)}(c_1; c_3, c_2)\mid_{s_1^{(1)}=1, s_1^{(1)}=1} = \min \left\{ \begin{array}{l} d_{12}(1,8) + f^{(2)}(c_3; c_2)\mid_{8,1} \\ d_{12}(1,10) + f^{(2)}(c_3; c_2)\mid_{10,1} \\ d_{12}(1,11) + f^{(2)}(c_3; c_2)\mid_{11,1} \end{array} \right\}$$

$$= \min \left\{ \begin{array}{l} 13 + 42 = 55 \\ 14 + 46 = 60 \\ 11 + 44 = 55 \end{array} \right\} = 55$$

Hence the length of the optimal tour $\omega(1)$ for $s_1^{(1)} = 1$ is

$$\min \left\{ \begin{array}{l} f^{(1)}(c_1; c_2, c_3) \mid_{s_1^{(1)}=1, s_1^{(1)}=1} = 59 \\ f^{(1)}(c_1; c_3, c_2) \mid_{s_1^{(1)}=1, s_1^{(1)}=1} = 55 \end{array} \right\} = f^{(1)}(c_1; c_3, c_2) \mid_{s_1^{(1)}=1, s_1^{(1)}=1} = 55$$

In a similar way, if we let

(2) $s_1^{(1)} = 12$ we yield

$$f^{(1)}(c_1; c_3, c_2) \mid_{s_1^{(1)}=12, s_1^{(1)}=12} = 52$$

(3) $s_1^{(1)} = 13$ we yield

$$f^{(1)}(c_1; c_3, c_2) \mid_{s_1^{(1)}=13, s_1^{(1)}=13} = 53$$

Hence the length of the optimal tour to $G(1)$ is

$$\min \left\{ \begin{array}{l} f^{(1)}(c_1; c_3, c_2) \mid_{s_1^{(1)}=1, s_1^{(1)}=1} = 55 \\ f^{(1)}(c_1; c_3, c_2) \mid_{s_1^{(1)}=12, s_1^{(1)}=12} = 52 \\ f^{(1)}(c_1; c_3, c_2) \mid_{s_1^{(1)}=13, s_1^{(1)}=13} = 53 \end{array} \right\} = f^{(1)}(c_1; c_3, c_2) \mid_{s_1^{(1)}=12, s_1^{(1)}=12} = 52$$

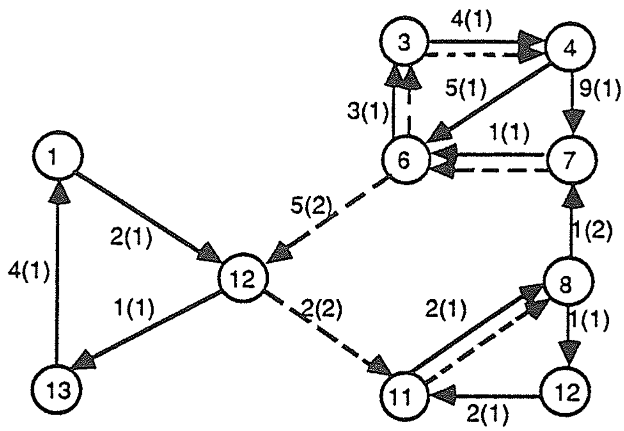


Figure 6.3: Corresponding augmented subnetwork of $G(1)$

The corresponding augmented network with above solution is shown in Figure 6.3 and the sub-postman tour is shown in Figure 6.4. The total deadheading is 52 units.

Computational feasibility

Although the dynamic programming approach presented above can produce the optimal solution to the CPP in a non-directly connected subnetwork, it is very difficult to implement on the computer due to the storage requirements, a problem that also encountered when using dynamic programming method to solve the TSP. In fact, the CPP in a non-directly connected sub-

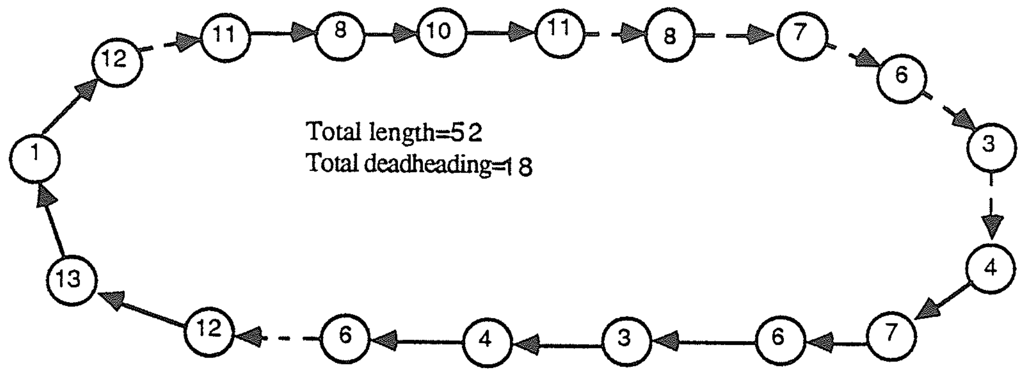


Figure 6.4: Sub-postman tour $\omega(1)$

network can be viewed as a special case of the TSP where one is required not only to visit each component (equivalent to the node in TSP) at least once but also to visit the links of each component at least once. If there is only one way to visit the links of each component, then the CPP in a non-directly connected subnetwork is exactly the same as the TSP.

In solving the TSP using the dynamic programming approach, to calculate $f^{(k)}$ on stage k, the number of values of $f^{(k+1)}$ that needs to be known is

$$g(n,k) = (n-1)!/(k-1)!(n-k-1)![3] \tag{6.6}$$

where n is the number of nodes in TSP. On a computer with 32k memory (such as the IBM 7094), the largest problem size reported to be solvable for TSP was $n = 15$[3].

Since we can actually visit the links of a component by starting from any of its nodes, the number of ways to visit the links of a component is usually more than one. Hence, using the dynamic programming approach, the problem size that can be handled in solving the CPP in a non-directly connected subnetwork, is less than the one that can be handled in solving the TSP.

In concluding this section we can see that the key to solving the CPP in the non-directly connected subnetwork is to determine the starting node on each un-visited component. This is due to the links to connect the components can also be used to make the subnetwork even at the same time. We must state that the solution to such a problem is very difficult. None of the solution methods (heuristic or optimal) discussed so far is superior to another, when taking into account the computational efficiency and the solution quality requirements. For the heuristic approaches, the computation is easy but the quality of the solution is dependent on the structure of the subnetwork. Orloff[37]'s method works well when the subnetwork contains only one even component. If it contains more than one such component, the solution approach becomes exceedingly inefficient, since the solution branches expand rapidly. Tucker and Bodin[50]'s method may produce a solution in a large scale network but such an approach requires a special method to determine the distance between components, hence it may not be suitable for theoretical research. Alfa and Liu[1]'s method, which is also heuristic, works fast and is able to solve the relatively large scale problem since algorithms for solving the general CPP in the directed network and the TSP are readily available.

The dynamic programming approach, on the other hand, is able to produce the optimal solution all the time from the theoretical point of view, but the large storage requirements make the practical implementation on the computer extremely difficult. Therefore such an approach needs to be more computationally feasible.

The evaluations of the solution methods for solving the CPP in the non-directly connected subnetwork are summarized in Table 6.1. In this thesis, since we intend to use the IBM PC to perform the vehicle routing, taking into account the solution quality and computational efficiency requirements, the proposed solution method adopts Alfa and Liu[1]'s method to solve the SCVRP.

6.2 Discussion of the Solution Method to the Sub-postman Tour Composing Problem

As we stated in Section 4.3.2 of chapter 4, the solution obtained by using the subpostman tour composing method is only near optimal, even though there exists a common node to all the subpostman tours. This is because sometimes one may not need to visit all links of a subpostman tour before

Table 6.1: Comparison of the different solution methods to the CPP in a non-directly connected subnetwork

Approach	Problem size handled	Solution quality	Special requirement	Proposed by
Subtour elimination	Small	Heuristic	****	Orloff[39]
Minimum Spanning Tree	Large	Heuristic	Co-ordinate of each node	Tucker and Bodin[51]
Travelling Salesman Problem	Large	Heuristic	****	Alfa and Liu[1]
Dynamic Programming	Small	Optimal	Large computer memory	this thesis

going on to the next, provided that all the links that are required to be covered have already been visited

If we observe the problem more closely, we can see that such a problem occurs only in the subpostman tour, $\omega(r)$, where its starting node, s_r, is an odd node in the original subnetwork $G(r)$. In such a case we need to find an optimal tour that starts from s_r and traverses each link of $G(r)$ at least once and ends at another node, s'_r. Such a tour is often called the *Open Postman tour* and its corresponding open walk that traverses each link of $G(r)$ exactly once is called the *Open Euler tour*. It can easily be proved that if a given network contains more than two odd nodes, there exists an Open Euler tour which begins at one odd node and ends at another(See Deo[20]). Hence we can see that wherever the starting node of a subpostman tour is odd in the original subnetwork, we need to identify an open postman tour and check the ending node of the tour to see if the condition in equation (4.6) is satisfied.

Chapter 7

Conclusions

The present investigation of the optimal vehicle routing design has revealed that the Chinese Postman Problem on a hierarchical network, or the hierarchical link-covering vehicle routing problem, has received relatively little attention in both the theoretical aspect and the practical aspect. Such a problem appears not only in snow clearing vehicle routing but also in many other real-life vehicle routing problems such as the street sweeping, garbage collection, police patrol, etc. This thesis developed the optimization methodology to solve such a problem in theory and apply the developed methodology to solve the real-life problem in practice. The Chinese Postman Problem in a hierarchical network, in practice, can be solved optimally using the general CPP solution and the dynamic programming method if the given network size is relatively small and the links of each hierarchy are all connected. If the links of some of the hierarchies are not directly connected, the problem becomes exceedingly complicated and may be solved only through the heuristic solution method.

In summary, the following aspects of the postman problem in a hierarchical network are tackled throughout this thesis:

- The modelling;
- The solution method to solve the problem. This includes
  - Optimal solution method when links of each hierarchy of the hierarchical network are connected;
  - Heuristic method when links of some hierarchies of the hierarchical network are not connected;
  - Exploration of optimal solution method when links of some hierarchies are not connected –introduce a dynamic programming method
- Practical application of the proposed solution method–solve the SCVRP.

Direction of future research

In theory, the following may be developed:

- A more accurate mathematical model formulation.
- A more efficient solution algorithm. This includes:
  - A more efficient algorithm to solve the CPP in a non-directly connected subnetwork optimally;

- A more efficient algorithm to solve the subpostman composing problem;
- A solution algorithm which does not require the decomposing of the original problem.
- An algorithm to deal with the problem in the multi-postmen case

In practice, the following may be considered to be the future improvements:

- The exact number of times for the vehicle to clear out a street, given the number of lanes, should be calculated more accurately. In this thesis, we assume that to clear out one side of street with r number of lanes, r vehicle movements are required. However, such a value may be varied. For example, the weather conditions such as the thickness of the snow fall, the variation of the road such as the yield turns, bus routes, changes of the lane number, etc.
- The width of the street should be considered in the vehicle route. This is because the number of snow plows needed during the operation are dependent on the width of the streets being cleared. Sometimes the operations are better completed on a group of streets having the same width before going on to the others.
- The division of the total vehicle route into segments in order to allow several clearing vehicles to carry out the operation.

- The snow hauling operations should be incorporated.

Bibliography

[1] Alfa, A.S. and D.Q. Liu, "Postman Routing Problem in a Hierarchical Network," 1988, *Forcoming in Engineering Optimization.*

[2] Barahona, F., Maynard, R., Rammal, R. and J. Uhry, "Morphology of Ground States of Two-dimensional Frustration Model," Journal of Physics A, Mathematical and General 15, 673-679,1982.

[3] Bellman, R. "Dynamic Programming Treatment of the Travelling Salesman Problem," J.ACM 9,61-63, 1962

[4] Bellman, R. *Dynamic Programming.* Princeton University Press, Princeton, New Jersey,1957.

[5] Bellman, R. and K. Cooke, "The Konigsberg Bridge Problem Generalized," J. Math. Analysis Applic., 25, 1, 1969.

[6] Bellmore, M. and G. Nemhauser, "The Travelling Salesman Problem: A Survey," Op. Res. 16, 538-558, 1968.

[7] Beltrami, E. and L. Bodin, "Network and Vehicle Routing for Municipal Waste Collection," Networks, Vol. 4(1), 65-94, 1974.

[8] Bennett, B. and D. Gazis, "School Bus Routing by Computer," Trans. Res. 6, 317-325, 1974.

[9] Bodin, L and S. J. Kursh, "A Computer-Assisted System for the Routing and Scheduling of Street Sweepers," Ops. Res., Vol. 26(4), 525-537, 1978.

[10] Bodin, L and S. J. Kursh, "A Detailed Description of A Computer System for The Routing and Scheduling of Street Sweepers," Comput. & Ops. Res. Vol. 6, 181-198, 1979.

[11] Bodin, L. *et al*, "Routing and Scheduling of Vehicles and Crews," Comput. & Ops. Res. Vol. 10(2), 63-211, 1983.

[12] Bodin, L. and L. Berman, "Routing and Scheduling of School Buses by Computer," Trans. Sci. Vol. 13(2), 113-129, 1979.

[13] Bodin, L. and B. Golden, Classification in Vehicle Routing and Scheduling," Networks, Vol. 11(2), 97-108, 1981.

[14] Christofides, N., "The Optimum Traversal of a Graph," OMEGA, Vol. 1, No. 6, 719-732, 1973.

[15] Christofides, N., "The Vehicle Routing Problem," Presented at the NATO Conference on Combinatorial Optimization, Paris, 1974.

[16] Christofides, N. and S. Eilon, "An Algorithm for the Vehicle Dispatching Problem," Opnl. Res. Q., 20, 309, 1969.

[17] Clarke, G and J. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points," Ops. Res. 12, 568-581, 1964.

[18] Dantzig, G. and J. Ramser, "The Truck Dispatching Problem," Management Science 5, 81-91,1959.

[19] DeArmon, A., "A Comparison of Heuristics for the Capacitated Chinese Postman Problem," Master Thesis, University of Maryland at College Park, 1981.

[20] Deo, N., *Graph Theory with Applications to Engineering and Computer Science,* Prentice-Hall, Inc. N.J. 1974.

[21] Dijkstra, E., "A note on Two Problems in Connection with Graphs," Num. Math., 1, 269, 1959

[22] Edmonds, J., "The Chinese Postman's Problem," ORSA Bull., 13, 73, 1965

[23] Edmonds, J. and E. Johnson, "Matching, Euler Tours and the Chinese Postman," Math. Prog. 4, 88-124, 1973.

[24] Floyd, R., "Algorithm 97, Shortest Path," Commun. ACM 5, 345, 1982.

[25] Golden, B., Magnanti, T. and H. Nguyen, "Implementing Vehicle Routing Algorithms," Networks 7, 113-148, 1977.

[26] Golden, B., DeArmon, J. and E. Baker, "Computational Experiments with Algorithms for a Class of Routing Problems," Management Science & Statistics Working Paper No. 81-033, University of Maryland at College Park, 1981, *forcoming in Comput. & Ops. Res.*

[27] Golden B. and R. Wong, "Capacitated Arc Routing Problems," Networks 11(3), 305-315, 1981.

[28] Korach, E., "Packings of T-cuts and Other Aspects of Dual Intergrality," Ph.D. thesis, University of Walterloo 1982.

[29] Kulkarni, R. and P. Bhave, "Integer Programming Formulations of Vehicle Routing Problems," EJOR, 20, 58-67, 1985.

[30] Lenstra, J. and A. Rinnooy Kan, "On General Routing Problems," Networks 6(3), 273-280, 1976.

[31] Little, J., Murty, K., Sweeney, D. and C Karel, "An Algorithm for the Travelling Salesman Problem," Ops. Res. 11(6), 972-989, 1963.

[32] Magnanti, T. "Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospects," Networks, 11(2), 179-214, 1981.

[33] Kwan, Mei-Ko, "Graphic Programming Using Odd or Even Points," Chinese Math., Vol. 1, 273, 1962.

[34] Miller, C., A. Tucker and R. Zemlin, "Integer Programming Formulation of Travelling Salesman Problems," JACM 7, 326-329, 1960.

[35] Minieka, E., *Optimization Algorithms for Networks and Graphs,* Marcel-Dekker, N.Y., 1978.

[36] Minieka, E., "The Chinese Postman Problem for Mixed Networks," Management Science, Vol. 25, No. 7, 634-648, 1979.

[37] Moon, J., "The Urban Postman," OMEGA 10, 334-337, 1982.

[38] Newton, R. and W. Thomas, "Developing a Computer Program for Bus Routing: Final Report," State University of New York Research Foundation, Albany, N.Y., 1970.

[39] Orloff, C., "A Fundamental Problem in Vehicle Routing," Networks 4, 35-64, 1974.

[40] Orloff, C., "On General Routing Problems: Comments," Networks 6(3), 281-284, 1976.

[41] Orloff, C., "Routing a Fleet of M-Vehicles to/from a Central facility," Networks 4, 147-162, 1974.

[42] Orloff, C., "Routing and Scheduling a Fleet of Vehicles to/from Central Facilities - The School Bus Problem," Ph.D. dissertation, Cornell University, 1972.

[43] Papadimitriou, C., "On the Complexity of Edge Traversing," J. Assoc. Comput. Mach., Vol. 23, 1976.

[44] Sebo, A., "Finding the t-join Structure of Graphs," Math. Prog. 36, 123-134,1986. bibitemkn:CPP11 Sebo, A., "The Chinese Postman Problem: Algorithms, Structure and Applications," Working paper MO/61 MTA SZTAKI, 1985, forcoming in Discrete Applied Mathematics.

[45] Stern, H. and M. Dror, "Routing Electric Meter Readers," Comput. & Ops. Res. Vol. 6(4), 209-223, 1979.

[46] Stricker, R., "Public Sector Vehicle Routing: The Chinese Postman Problem," Master Thesis at M.I.T., 1970.

[47] Teply, G. and S. Teply, "Computer Assisted Routing Design for Roadway maintenance Vehicles," paper presented at The 3rd Canadian Seminar on Systems Theory for Civil Engineer, Ecole Polytechnique de Montreal, 1987.

[48] Tracz, G., and M. Norman, "A computerized System for School Bus Routing," The Ontario Inst. for Studies in Education, Toronto, canada, 1970.

[49] Turner, W., Ghare, P. and L. Foulds, "Transportation Routing Problems - A Survey," AIIE Transactions, 6(4), 288-301, 1974.

[50] Vu, V. and W. Turner, "System Design for Rural Refuse Collection," AIIE Transactions, 8(1), 84-96, 1976.

[51] Tucker, A. and L. Bodin, "A Model for Municipal Street Sweeping Operations," in Case Study of Applied Mathematics, MAOA, Washington, 1976.

Appendix A
District #2: Priority I Streets

FN	TN	LTH	P	#LN	BLOCK NAME	FROM	TO
1	2	63	1	8	Portage	Riverbend	Garden
2	1	63	1	8	Portage	Garden	Riverbend
2	4	80	1	8	Portage	Graden	Riverbend
4	2	80	1	8	Portage	Riverbend	Graden
4	5	188	1	8	Portage	Riverbend	Winston
5	4	188	1	8	Portage	Winston	Riverbend
5	9	80	1	8	Portage	Winston	Cavell
9	5	80	1	8	Portage	Cavell	Winston
9	11	78	1	8	Portage	Cavell	Bourkevale
11	9	78	1	8	Portage	Bourkevale	Cavell
11	13	80	1	8	Portage	Bourlevale	Riveroaks
13	11	80	1	8	Portage	Riveroaks	Bourkevale
13	15	90	1	8	Portage	Riveroaks	Parkview
15	13	90	1	8	Portage	Parkview	Riveroaks
15	17	88	1	8	Portage	Parkview	Roseberry
17	15	88	1	8	Portage	Roseberry	Parkview
17	19	88	1	8	Portage	Roseberry	Colligiate
19	17	88	1	8	Portage	Colligiate	Roseberry
19	21	90	1	8	Portage	Colligiate	Ferry
21	19	90	1	8	Portage	Ferry	Colligiate
21	23	212	1	8	Portage	Ferry	Library
23	21	212	1	8	Portage	Library	Ferry
23	25	95	1	8	Portage	Library	Douglas
25	23	95	1	8	Portage	Douglas	Library
25	27	175	1	8	Portage	Douglas	Albany
27	25	175	1	8	Portage	Albany	Douglas
27	29	112	1	8	Portage	Albany	Deer Lodge
29	27	112	1	8	Portage	Deer Lodge	Albany
29	30	125	1	8	Poratge	Deer Lodge	Deer Lodge
30	29	125	1	8	Portage	Deer Lodge	Deer Lodge
30	32	1740	1	8	Portage	Deer Lodge	Assiniboine
32	30	1740	1	8	Portage	Assiniboine	Deer Lodge
32	33	230	1	8	Portage	Assiniboine	Olive
33	32	230	1	8	Portage	Olive	Assiniboine
33	36	450	1	8	Portage	Olive	Moray
36	33	450	1	8	Portage	Moray	Olive
36	39	125	1	8	Portage	Moray	Lake
39	36	125	1	8	Portage	Lake	Moray
39	41	100	1	8	Portage	Lake	Windham
41	39	100	1	8	Portage	Windham	Lake
41	43	87	1	8	Portage	Windham	Thompson
43	41	87	1	8	Portage	Thompson	Windham
43	45	85	1	8	Portage	Thompson	Woodbridge
45	43	85	1	8	Portage	Woodbridge	Thopmson
45	48	170	1	8	Portage	Woodbridge	Old Mill
48	45	170	1	8	Portage	Old Mill	Woodbridge
48	49	175	1	8	Portage	Old Mill	Woodhaven
49	48	175	1	8	Portage	Woodhaven	Old Mill
49	55	263	1	8	Portage	Woodhaven	Oakdean
55	49	262	1	8	Portage	Oakdean	Woodhaven
55	63	100	1	8	Portage	Oakdean	Harris
63	55	100	1	8	Portage	Harris	Oakdean
63	68	200	1	8	Portage	Harris	Woodland
68	63	200	1	8	Portage	Woodland	Harris
68	70	90	1	8	Portage	Woodland	Woodland
70	68	90	1	8	Portage	Woodland	Woodland
70	73	125	1	8	Portage	Woodland	Country Club
73	70	125	1	8	Portage	Country Club	Woodland
73	76	115	1	8	Portage	Country Club	Greenacre
76	73	115	1	8	Portage	Greenacre	Country Club
76	79	100	1	8	Portage	Greenacre	Golf
79	76	100	1	8	Portage	Golf	Greenacre
79	93	100	1	8	Portage	Golf	Daer
93	79	100	1	8	Portage	Daer	Golf

93	99	115	1	8	Portage	Daer	Banting
99	93	115	1	8	Portage	Banting	Daer
99	101	120	1	8	Portage	Banting	Kirkfield
101	99	120	1	8	Portage	Kirkfield	Banting
101	111	90	1	8	Portage	Kirkfield	Shelley
111	101	90	1	8	Portage	Shelley	Kirkfield
111	139	130	1	8	Portage	Shelley	Westwood
139	111	130	1	8	Portage	Westwood	Shelley
139	159	430	1	8	Portage	Westwood	Rouge
159	139	430	1	8	Portage	Rouge	Westwood
159	190	240	1	8	Portage	Rouge	Sumach
190	159	240	1	8	Portage	Sumach	Rouge
190	218	85	1	8	Portage	Sumach	Seaton
218	190	85	1	8	Portage	Seaton	Sumach
218	220	85	1	8	Portage	Seaton	Raquette
220	218	85	1	8	Portage	Raquette	Seaton
220	232	85	1	8	Portage	Raquette	Best
232	220	85	1	8	Portage	Best	Raquette
232	234	85	1	8	Portage	Best	Bedson
234	232	85	1	8	Portage	Bedson	Best
234	258	700	1	8	Portage	Bedson	Buchanan
258	260	200	1	8	Portage	Buchanan	Stewart
258	234	700	1	8	Portage	Buchanan	Bedson
260	258	200	1	8	Portage	Stewart	Buchanan
260	262	150	1	8	Portage	Steward	St Charles
262	260	150	1	8	Portage	St Charles	Steward
262	264	100	1	8	Portage	St Charles	Gagnon
264	262	100	1	8	Portage	Gagnon	St Charles

TOTAL CLEARED LENGTH: 67,960m

Appendix B
District #2: Vehicle Route For Priority I Street

FN	TN	LTH	P	#LN	BLOCK NAME	FROM	TO
1	2	63	1	8	Portage	Riverbend	Garden
2	4	80	1	8	Portage	Graden	Riverbend
4	5	188	1	8	Portage	Riverbend	Winston
5	9	80	1	8	Portage	Winston	Cavell
9	11	78	1	8	Portage	Cavell	Bourkevale
11	13	80	1	8	Portage	Bourlevale	Riveroaks
13	15	90	1	8	Portage	Riveroaks	Parkview
15	17	88	1	8	Portage	Parkview	Roseberry
17	19	88	1	8	Portage	Roseberry	Colligiate
19	21	90	1	8	Portage	Colligiate	Ferry
21	23	212	1	8	Portage	Ferry	Library
23	25	95	1	8	Portage	Library	Douglas
25	27	175	1	8	Portage	Douglas	Albany
27	29	112	1	8	Portage	Albany	Deer Lodge
29	30	125	1	8	Poratge	Deer Lodge	Deer Lodge
30	32	1740	1	8	Portage	Deer Lodge	Assiniboine
32	33	230	1	8	Portage	Assiniboine	Olive
33	36	450	1	8	Portage	Olive	Moray
36	39	125	1	8	Portage	Moray	Lake
39	41	100	1	8	Portage	Lake	Windham
41	43	87	1	8	Portage	Windham	Thompson
43	45	85	1	8	Portage	Thompson	Woodbridge
45	48	170	1	8	Portage	Woodbridge	Old Mill
48	49	175	1	8	Portage	Old Mill	Woodhaven
49	55	263	1	8	Portage	Woodhaven	Oakdean
55	63	100	1	8	Portage	Oakdean	Harris
63	68	200	1	8	Portage	Harris	Woodland
68	70	90	1	8	Portage	Woodland	Woodland
70	73	125	1	8	Portage	Woodland	Country Club
73	76	115	1	8	Portage	Country Club	Greenacre
76	79	100	1	8	Portage	Greenacre	Golf
79	93	100	1	8	Portage	Golf	Daer
93	99	115	1	8	Portage	Daer	Banting
99	101	120	1	8	Portage	Banting	Kirkfield
101	111	90	1	8	Portage	Kirkfield	Shelley
111	139	130	1	8	Portage	Shelley	Westwood
139	159	430	1	8	Portage	Westwood	Rouge
159	190	240	1	8	Portage	Rouge	Sumach
190	218	85	1	8	Portage	Sumach	Seaton
218	220	85	1	8	Portage	Seaton	Raquette
220	232	85	1	8	Portage	Raquette	Best
232	234	85	1	8	Portage	Best	Bedson
234	258	700	1	8	Portage	Bedson	Buchanan
258	234	700	1	8	Portage	Buchanan	Bedson
260	262	150	1	8	Portage	Steward	St Charles
262	264	100	1	8	Portage	St Charles	Gagnon
264	262	100	1	8	Portage	Gagnon	St Charles
262	260	150	1	8	Portage	St Charles	Steward
260	258	200	1	8	Portage	Stewart	Buchanan
258	260	200	1	8	Portage	Buchanan	Stewart
234	232	85	1	8	Portage	Bedson	Best
232	220	85	1	8	Portage	Best	Raquette
220	218	85	1	8	Portage	Raquette	Seaton
218	190	85	1	8	Portage	Seaton	Sumach
190	159	240	1	8	Portage	Sumach	Rouge
159	139	430	1	8	Portage	Rouge	Westwood
139	111	130	1	8	Portage	Westwood	Shelley
111	101	90	1	8	Portage	Shelley	Kirkfield
101	99	120	1	8	Portage	Kirkfield	Banting
99	93	115	1	8	Portage	Banting	Daer
93	79	100	1	8	Portage	Daer	Golf
79	76	100	1	8	Portage	Golf	Greenacre
76	73	115	1	8	Portage	Greenacre	Country Club

73	70	125	1	8	Portage	Country Club	Woodland
70	68	90	1	8	Portage	Woodland	Woodland
68	63	200	1	8	Portage	Woodland	Harris
63	55	100	1	8	Portage	Harris	Oakdean
55	49	262	1	8	Portage	Oakdean	Woodhaven
49	48	175	1	8	Portage	Woodhaven	Old Mill
48	45	170	1	8	Portage	Old Mill	Woodbridge
45	43	85	1	8	Portage	Woodbridge	Thopmson
43	41	87	1	8	Portage	Thompson	Windham
41	39	100	1	8	Portage	Windham	Lake
39	36	125	1	8	Portage	Lake	Moray
36	33	450	1	8	Portage	Moray	Olive
33	32	230	1	8	Portage	Olive	Assiniboine
32	30	1740	1	8	Portage	Assiniboine	Deer Lodge
30	29	125	1	8	Portage	Deer Lodge	Deer Lodge
29	27	112	1	8	Portage	Deer Lodge	Albany
27	25	175	1	8	Portage	Albany	Douglas
25	23	95	1	8	Portage	Douglas	Library
23	21	212	1	8	Portage	Library	Ferry
21	19	90	1	8	Portage	Ferry	Colligiate
19	17	88	1	8	Portage	Colligiate	Roseberry
17	15	88	1	8	Portage	Roseberry	Parkview
15	13	90	1	8	Portage	Parkview	Riveroaks
13	11	80	1	8	Portage	Riveroaks	Bourkevale
11	9	78	1	8	Portage	Bourkevale	Cavell
9	5	80	1	8	Portage	Cavell	Winston
5	4	188	1	8	Portage	Winston	Riverbend
4	2	80	1	8	Portage	Riverbend	Graden
2	1	63	1	8	Portage	Garden	Riverbend

TOTAL DEADHEADING: 0m

Appendix C
District #2: Priority II Streets

FN	TN	LTH	P	#LN	BLOCK NAME	FROM	TO
7	10	88	2	2	Assiniboine	Winston	Cavell
10	7	88	2	2	Assiniboine	Cavell	Winston
10	12	78	2	2	Assiniboine	Bourkevale	Cavell
12	10	78	2	2	Assiniboine	Bourkevale	Cavell
12	14	80	2	2	Assiniboine	Bourkevale	Riveroaks
14	12	80	2	2	Assiniboine	Riveroaks	Bourkevale
14	16	100	2	2	Assiniboine	Riveroaks	Parkview
16	14	100	2	2	Assiniboine	Parkview	Riveroaks
16	18	95	2	2	Assiniboine	Parkview	Roseberry
18	16	95	2	2	Assiniboine	Roseberry	Parkview
18	20	95	2	2	Assiniboine	Roseberry	Colligiate
20	18	95	2	2	Assiniboine	Colligiate	Roseberry
20	22	95	2	2	Assiniboine	Colligiate	Ferry
21	22	425	2	2	Ferry	Portage	Assiniboine
22	20	95	2	2	Assiniboine	Ferry	Colligiate
22	21	425	2	2	Ferry	Assiniboine	Portage
32	34	350	2	2	Assiniboine	Portage	Olive
34	32	350	2	2	Assiniboine	Olive	Portage
34	35	88	2	2	Assiniboine	Olive	Pinewood
35	34	88	2	2	Assiniboine	Pinewood	Olive
35	38	350	2	2	Assiniboine	Pinewood	Moray
38	35	350	2	2	Assiniboine	Moray	Pinewood
38	40	120	2	2	Assiniboine	Moray	Lake
40	38	120	2	2	Assiniboine	Lake	Moray
40	42	90	2	2	Assiniboine	Lake	Windham
42	40	90	2	2	Assiniboine	Windham	Lake
42	44	125	2	2	Assiniboine	Windham	Thompson
44	42	125	2	2	Assiniboine	Thompson	Windham
44	47	80	2	2	Assiniboine	Thompson	Woodbridge
47	44	80	2	2	Assiniboine	Woodbridge	Thompson
49	50	340	2	2	Woodhaven	Portage	Emo
50	49	340	2	2	Woodhaven	Emo	Portage
50	51	150	2	2	Woodhaven	Emo	Glendale
51	50	75	2	2	Woodhaven	Glendale	Emo
51	54	260	2	2	Woodhaven	Glendale	Assiniboine
54	51	260	2	2	Woodhaven	Assiniboine	Glendale
54	61	130	2	2	Assiniboine	Woodhaven	Sunnyside
61	54	130	2	2	Assiniboine	Sunnyside	Woodhaven
61	62	110	2	2	Assiniboine	Sunnyside	Oakdean
62	61	110	2	2	Assiniboine	Oakdean	Sunnyside
62	67	105	2	2	Assiniboine	Oakdean	Harris
63	65	310	2	2	Harris	Portage	Armour
64	66	145	2	2	Harris	Armour	Emo
65	63	310	2	2	Harris	Armour	Portage
66	64	145	2	2	Harris	Emo	Armour
66	67	360	2	2	Harris	Emo	Assiniboine
67	62	105	2	2	Assiniboine	Harris	Oakdean
67	66	360	2	2	Harris	Assiniboine	Emo
71	72	85	2	2	Mcbey	Woodland	Summerland
72	71	85	2	2	Mcbey	Summerlands	Woodland
72	74	30	2	2	Mcbey	Summerlands	Country Club
73	74	220	2	2	Country Club	Portage	Mcbey
74	72	30	2	2	Mcbey	Country Club	Summerlands
74	73	220	2	2	Country Club	Mcbey	Portage
74	75	290	2	2	Country Club	Mcbey	Meadowside
75	74	290	2	2	Country Club	Meadowside	Mcbey
74	77	105	2	2	Mcbey	Country Club	Greenacre
75	83	150	2	2	Country Club	Meadowside	Pine Valley
77	74	105	2	2	Mcbey	Greenacre	Country Club
77	80	95	2	2	Mcbey	Greenacre	Golf
80	77	95	2	2	Mcbey	Golf	Greenacre
80	94	95	2	2	Mcbey	Golf	Daer
83	75	150	2	2	Country Club	Park West	Meadowside
83	92	112	2	2	Country Club	Park West	Pinehurst

88	89	125	2	2	Country Club	Pine Valley	S. End
88	90	90	2	2	Country Club	Pine Valley	Pinehurst
89	88	125	2	2	Country Club	S. End	Pine Valley
90	88	90	2	2	Country Club	Pinehurst	Pine Valley
90	91	80	2	2	Country Club	Pinehurst	Hagen
91	90	80	2	2	Country Club	Hagen	Pinehurst
91	92	183	2	2	Country Club	Hagen	Pinehurst
92	83	112	2	2	Country Club	Pinehurst	Park West
92	91	183	2	2	Country Club	Pinehurst	Hagen
94	80	95	2	2	Mcbey	Daer	Golf
94	100	110	2	2	Mcbey	Daer	Banting
96	110	85	2	2	Browning	Pinehurst	Wordsworth
100	94	110	2	2	Mcbey	Banting	Daer
100	102	120	2	2	Mcbey	Banting	Kirkfield
102	100	120	2	2	Mcbey	Kirkfield	Banting
102	112	90	2	2	Mcbey	Kirkfield	Shelley
110	96	85	2	2	Browning	Wordsworth	Pinehurst
110	119	160	2	2	Browning	Wordsworth	Shelley
112	102	90	2	2	Mcbey	Shelley	Kirkfield
112	142	130	2	2	Mcbey	Shelley	Westwood
119	110	160	2	2	Browning	Shelley	Wordsworth
119	120	95	2	2	Browning	Shelley	Westwood
120	119	95	2	2	Browning	Westwood	Shelley
120	121	85	2	2	Westwood	Browning	Wordsworth
120	150	220	2	2	Westwood	Browning	Sansome
120	175	350	2	2	Browning	Westwood	Wordsworth
121	120	85	2	2	Westwood	Wordsworth	Browning
121	122	85	2	2	Westwood	Wordsworth	Carlyle Cr
122	121	85	2	2	Westwood	Carlyle Cr	Wordsworth
122	125	100	2	2	Westwood	Carlyle	Carlyle
124	126	90	2	2	Westwood	Carlyle	Sandburg
125	122	100	2	2	Westwood	Carlyle	Carlyle
126	124	90	2	2	Westwood	Sandburg	Carlyle
126	128	90	2	2	Westwood	Sandburg	Sandburg
128	126	90	2	2	Westwood	Sandburg	Sandburg
128	130	85	2	2	Westwood	Sandburg	Haliburton
130	128	85	2	2	Westwood	Haliburton	Sandburg
131	133	85	2	2	Westwood	Haliburton	Haliburton
133	131	85	2	2	Westwood	Haliburton	Haliburton
133	134	80	2	2	Westwood	Haliburton	Kilmer
134	133	80	2	2	Westwood	Kilmer	Haliburton
134	136	85	2	2	Westwood	Kilmer	Assiniboine
136	134	85	2	2	Westwood	Assiniboine	Kilmer
136	137	70	2	3	Assiniboine	Westwood	Westboine
136	158	220	2	3	Assiniboine	Westwood	Caroll
137	136	70	2	3	Assiniboine	Westboine	Westwood
139	140	212	2	2	Westwood	Portage	Byrd
140	139	212	2	2	Westwood	Byrd	Portage
140	141	90	2	2	Westwood	Byrd	Amundsen
141	140	90	2	2	Westwood	Admunsen	Byrd
141	142	25	2	2	Westwood	Admunsen	Mcbey
142	112	130	2	2	Mcbey	Westwood	Shelley
142	141	25	2	2	Westwood	Mcbey	Admunsen
142	143	50	2	2	Westwood	Mcbey	Admunsen
143	142	50	2	2	Westwood	Admunsen	Mcbey
143	144	85	2	2	Westwood	Admunsen	Henday
144	143	85	2	2	Westwood	Henday	Admunsen
144	145	85	2	2	Westwood	Henday	Henday
145	144	85	2	2	Westwood	Henday	Henday
145	147	85	2	2	Westwood	Henday	Magellan
147	145	85	2	2	Westwood	Magellan	Henday
147	148	85	2	2	Westwood	Magellan	Magellan
148	147	85	2	2	Westwood	Henday	Magellan
148	150	85	2	2	Westwood	Magellan	Sansome
150	120	220	2	2	Westwood	Sansome	Browning
150	148	85	2	2	Westwood	Sansome	Magellan
150	151	80	2	2	Sansome	Westwood	Joliett
151	150	80	2	2	Sansome	Joliett	Westwood
151	171	175	2	2	Sansome	Joliett	Peary
158	136	220	2	3	Assiniboine	Caroll	Westwood
158	186	112	2	3	Assiniboine	Caroll	Long Fellow
159	160	212	2	2	Rouge	Portage	Byrd
160	159	212	2	2	Rouge	Byrd	Portage
160	162	85	2	2	Rouge	Byrd	Davis
162	160	85	2	2	Rouge	Davis	Byrd
162	163	85	2	2	Rouge	Davis	Ericson

163	162	85	2	2	Rouge	Ericson	Davis
163	165	85	2	2	Rouge	Ericson	Ericson
165	163	85	2	2	Rouge	Ericson	Ericson
165	166	80	2	2	Rouge	Ericson	Galinee
165	199	110	2	2	Allard	Rouge	Pontiac
166	165	80	2	2	Rouge	Galinee	Ericson
166	167	85	2	2	Rouge	Galinee	Galinee
167	166	85	2	2	Rouge	Galinee	Galinee
167	169	85	2	2	Rouge	Galinee	Peary
169	167	85	2	2	Rouge	Peary	Galinee
169	170	85	2	2	Rouge	Peary	Sansome
170	169	85	2	2	Rouge	Sansome	Peary
170	171	170	2	2	Sansome	Rouge	Peary
170	172	85	2	2	Rouge	Sansome	Joliett
170	201	245	2	2	Sansome	Rouge	Whitegates
171	151	175	2	2	Sansome	Peary	Joliett
171	170	170	2	2	Sansome	Peary	Rouge
172	170	85	2	2	Rouge	Joliett	Sansome
172	174	265	2	2	Rouge	Joliett	Browning
174	172	265	2	2	Rouge	Browning	Joliett
174	176	235	2	2	Rouge	Browning	Erlandson
175	120	350	2	2	Browning	Wordsworth	Westwood
176	174	235	2	2	Rouge	Erlandson	Browning
176	208	137	2	2	Rouge	Erlandson	West
184	189	85	2	2	Rouge	Columbus	Scalena
184	208	133	2	2	Rouge	Columbus	West
186	158	112	2	3	Assiniboine	Long Fellow	Caroll
186	187	85	2	3	Assiniboine	Long Fellow	Long Fellow
187	186	85	2	3	Assiniboine	Long Fellow	Long Fellow
187	188	85	2	3	Assiniboine	Long Fellow	Scalena
188	187	85	2	3	Assiniboine	Scalena	Long Fellow
188	211	85	2	3	Assiniboine	Scalena	Rouge
189	184	85	2	2	Rouge	Scalena	Columbus
189	211	75	2	2	Rouge	Scalena	Assiniboine
197	198	45	2	2	Allard	Sumach	Pontiac
198	197	85	2	2	Allard	Pontiac	Sumach
198	199	80	2	2	Allard	Pontiac	Pontiac
197	219	85	2	2	Allard	Sumach	Seaton
199	165	110	2	2	Allard	Pontiac	Rouge
199	198	80	2	2	Allard	Pontiac	Pontiac
201	170	245	2	2	Sansome	Whitegates	Rouge
201	222	175	2	2	Sansome	Whitegates	Raquette
208	176	137	2	2	Rouge	West	Erlandson
208	184	133	2	2	Rouge	West	Columbus
211	188	85	2	3	Assiniboine	Rouge	Scalena
211	189	75	2	2	Rouge	Assiniboine	Scalena
211	212	275	2	3	Assiniboine	Rouge	Maureen
212	211	275	2	3	Assiniboine	Maureen	Rouge
212	217	85	2	3	Assiniboine	Maureen	Barron
217	212	85	2	3	Assiniboine	Barron	Maureen
217	231	90	2	2	Assinibione	Barron	Raquette
219	197	85	2	2	Allard	Seaton	Sumach
219	221	85	2	2	Allard	Seaton	Raquette
221	219	85	2	2	Allard	Raquette	Seaton
221	233	85	2	2	Allard	Raquette	Best
222	201	175	2	2	Sansome	Raquette	Whitegates
222	242	175	2	2	Sansome	Raquette	Bedson
231	217	90	2	2	Assinibione	Raquette	Barron
231	255	180	2	2	Assinibione	Raquette	Bedson
233	221	85	2	2	Allard	Best	Raquette
233	237	85	2	2	Allard	Best	Bedson
234	235	260	2	2	Bedson	Portage	Twain
235	234	260	2	2	Bedson	Twain	Portage
235	237	155	2	2	Bedson	Twain	Allard
237	233	85	2	2	Allard	Bedson	Best
237	235	155	2	2	Bedson	Allard	Twain
237	238	80	2	2	Bedson	Allard	Keats
238	237	80	2	2	Bedson	Keats	Allard
238	239	80	2	2	Bedson	Keats	Keats
239	238	80	2	2	Bedson	Keats	Keats
239	241	85	2	2	Bedson	Keats	Twain
241	239	85	2	2	Bedson	Twain	Keats
241	242	85	2	2	Bedson	Twain	Sansome
242	222	175	2	2	Sansome	Bedson	Raquette
242	241	85	2	2	Bedson	Sansome	Twain
242	243	85	2	2	Bedson	Sansome	Dicken's

243	242	85	2	2	Bedson	Dicken's	Sansome
243	244	85	2	2	Bedson	Dicken's	Shakespeare
244	243	85	2	2	Bedson	Shakespeare	Dicken's
244	245	80	2	2	Bedson	Shakespeare	Shakespeare
245	244	80	2	2	Bedson	Shakespeare	Shakespeare
245	247	85	2	2	Bedson	Shakespeare	Dicken's
247	245	85	2	2	Bedson	Dicken's	Shakespeare
247	249	185	2	2	Bedson	Dicken's	Assinibione Gr
249	247	185	2	2	Bedson	Assinibione Gr	Dicken's
249	250	80	2	2	Bedson	Assinibione Gr	Coleridge
250	249	80	2	2	Bedson	Coleridge	Assinibione Gr
250	255	295	2	2	Bedson	Holt	Assinibione
255	231	180	2	2	Assinibione	Bedson	Raquette
255	250	295	2	2	Bedson	Assinibione	Holt
259	261	100	2	2	Allard	Buchanan	Stewart
261	259	100	2	2	Allard	Steward	Allard
261	263	145	2	2	Allard	Steward	St Charles
262	263	210	2	2	St Charles	Portage	Allard
263	261	145	2	2	Allard	St Charles	Stewart
263	262	210	2	2	St Charles	Allard	Portage
263	265	95	2	2	Allard	St Charles	Gagnon
263	269	170	2	2	St Charles	Allard	Lepine
265	263	95	2	2	Allard	Gagnon	St Charles
269	263	170	2	2	St Charles	Lepine	Allard
269	274	165	2	2	St Charles	Lepine	Sansume
274	269	165	2	2	St Charles	Sansome	Lepine
274	280	165	2	2	St Charles	Sansome	Augier
280	274	165	2	2	St Charles	Augier	Sansome

TOTAL CLEARED LENGTH: 37,700m

Appendix D
District #2: Components of Priority II Streets

FN	TN	LTH	P	#LN	BLOCK NAME	FROM	TO
COMPONENT 1:							
7	10	88	2	2	Assiniboine	Winston	Cavell
10	12	78	2	2	Assiniboine	Bourkevale	Cavell
12	14	80	2	2	Assiniboine	Bourkevale	Riveroaks
14	16	100	2	2	Assiniboine	Riveroaks	Parkview
16	18	95	2	2	Assiniboine	Parkview	Roseberry
18	20	95	2	2	Assiniboine	Roseberry	Colligiate
20	22	95	2	2	Assiniboine	Colligiate	Ferry
22	21	425	2	2	Ferry	Assiniboine	Portage
21	22	425	2	2	Ferry	Portage	Assiniboine
22	20	95	2	2	Assiniboine	Ferry	Colligiate
20	18	95	2	2	Assiniboine	Colligiate	Roseberry
18	16	95	2	2	Assiniboine	Roseberry	Parkview
16	14	100	2	2	Assiniboine	Parkview	Riveroaks
.14	12	80	2	2	Assiniboine	Riveroaks	Bourkevale
12	10	78	2	2	Assiniboine	Bourkevale	Cavell
10	7	88	2	2	Assiniboine	Cavell	Winston
COMPONENT 2							
32	34	350	2	2	Assiniboine	Portage	Olive
34	35	88	2	2	Assiniboine	Olive	Pinewood
35	38	350	2	2	Assiniboine	Pinewood	Moray
38	40	120	2	2	Assiniboine	Moray	Lake
40	42	90	2	2	Assiniboine	Lake	Windham
42	44	125	2	2	Assiniboine	Windham	Thompson
44	47	80	2	2	Assiniboine	Thompson	Woodbridge
47	44	80	2	2	Assiniboine	Woodbridge	Thompson
44	42	125	2	2	Assiniboine	Thompson	Windham
42	40	90	2	2	Assiniboine	Windham	Lake
40	38	120	2	2	Assiniboine	Lake	Moray
38	35	350	2	2	Assiniboine	Moray	Pinewood
35	34	88	2	2	Assiniboine	Pinewood	Olive
34	32	350	2	2	Assiniboine	Olive	Portage
COMPONENT 3							
49	50	340	2	2	Woodhaven	Portage	Emo
50	51	150	2	2	Woodhaven	Emo	Glendale
51	54	260	2	2	Woodhaven	Glendale	Assiniboine
54	61	130	2	2	Assiniboine	Woodhaven	Sunnyside
61	62	110	2	2	Assiniboine	Sunnyside	Oakdean
62	67	105	2	2	Assiniboine	Oakdean	Harris
67	66	360	2	2	Harris	Assiniboine	Emo
66	64	145	2	2	Harris	Emo	Armour
65	63	310	2	2	Harris	Armour	Portage
63	65	310	2	2	Harris	Portage	Armour
64	66	145	2	2	Harris	Armour	Emo
66	67	360	2	2	Harris	Emo	Assiniboine
67	62	105	2	2	Assiniboine	Harris	Oakdean
62	61	110	2	2	Assiniboine	Oakdean	Sunnyside
61	54	130	2	2	Assiniboine	Sunnyside	Woodhaven
54	51	260	2	2	Woodhaven	Assiniboine	Glendale
51	50	75	2	2	Woodhaven	Glendale	Emo
50	49	340	2	2	Woodhaven	Emo	Portage
COMPONENT 4							
71	72	85	2	2	Mcbey	Woodland	Summerland
72	74	30	2	2	Mcbey	Summerlands	Country Club
74	75	290	2	2	Country Club	Mcbey	Meadowside
75	83	150	2	2	Country Club	Meadowside	Pine Valley

83	92	112	2	2	Country Club	Park West	Pinehurst
92	91	183	2	2	Country Club	Pinehurst	Hagen
.91	90	80	2	2	Country Club	Hagen	Pinehurst
90	88	90	2	2	Country Club	Pinehurst	Pine Valley
88	89	125	2	2	Country Club	Pine Valley	S. End
89	88	125	2	2	Country Club	S. End	Pine Valley
88	90	90	2	2	Country Club	Pine Valley	Pinehurst
90	91	80	2	2	Country Club	Pinehurst	Hagen
91	92	183	2	2	Country Club	Hagen	Pinehurst
92	83	112	2	2	Country Club	Pinehurst	Park West
83	75	150	2	2	Country Club	Park West	Meadowside
75	74	290	2	2	Country Club	Meadowside	Mcbey
74	77	105	2	2	Mcbey	Country Club	Greenacre
77	80	95	2	2	Mcbey	Greenacre	Golf
80	94	95	2	2	Mcbey	Golf	Daer
94	100	110	2	2	Mcbey	Daer	Banting
100	102	120	2	2	Mcbey	Banting	Kirkfield
102	112	90	2	2	Mcbey	Kirkfield	Shelley
112	142	130	2	2	Mcbey	Shelley	Westwood
142	143	50	2	2	Westwood	Mcbey	Admunsen
142	141	25	2	2	Westwood	Mcbey	Admunsen
141	140	90	2	2	Westwood	Admunsen	Byrd
140	139	212	2	2	Westwood	Byrd	Portage
139	140	212	2	2	Westwood	Portage	Byrd
140	141	90	2	2	Westwood	Byrd	Amundsen
141	142	25	2	2	Westwood	Admunsen	Mcbey
143	144	85	2	2	Westwood	Admunsen	Henday
144	145	85	2	2	Westwood	Henday	Henday
145	147	85	2	2	Westwood	Henday	Magellan
147	148	85	2	2	Westwood	Magellan	Magellan
148	150	85	2	2	Westwood	Magellan	Sansome
150	151	80	2	2	Sansome	Westwood	Joliett
151	171	175	2	2	Sansome	Joliett	Peary
171	170	170	2	2	Sansome	Peary	Rouge
170	201	245	2	2	Sansome	Rouge	Whitegates
201	222	175	2	2	Sansome	Whitegates	Raquette
222	242	175	2	2	Sansome	Raquette	Bedson
242	222	175	2	2	Sansome	Bedson	Raquette
222	201	175	2	2	Sansome	Raquette	Whitegates
201	170	245	2	2	Sansome	Whitegates	Rouge
170	172	85	2	2	Rouge	Sansome	Joliett
172	174	265	2	2	Rouge	Joliett	Browning
174	176	235	2	2	Rouge	Browning	Erlandson
176	208	137	2	2	Rouge	Erlandson	West
208	184	133	2	2	Rouge	West	Columbus
184	189	85	2	2	Rouge	Columbus	Scalena
189	211	75	2	2	Rouge	Scalena	Assiniboine
211	189	75	2	2	Rouge	Assiniboine	Scalena
189	184	85	2	2	Rouge	Scalena	Columbus
184	208	133	2	2	Rouge	Columbus	West
208	176	137	2	2	Rouge	West	Erlandson
176	174	235	2	2	Rouge	Erlandson	Browning
174	172	265	2	2	Rouge	Browning	Joliett
172	170	85	2	2	Rouge	Joliett	Sansome
170	169	85	2	2	Rouge	Sansome	Peary
169	167	85	2	2	Rouge	Peary	Galinee
167	166	85	2	2	Rouge	Galinee	Galinee
166	165	80	2	2	Rouge	Galinee	Ericson
165	199	110	2	2	Allard	Rouge	Pontiac
199	198	80	2	2	Allard	Pontiac	Pontiac
198	197	85	2	2	Allard	Pontiac	Sumach
197	219	85	2	2	Allard	Sumach	Seaton
219	221	85	2	2	Allard	Seaton	Raquette
221	233	85	2	2	Allard	Raquette	Best
233	237	85	2	2	Allard	Best	Bedson
237	238	80	2	2	Bedson	Allard	Keats
238	239	80	2	2	Bedson	Keats	Keats
239	241	85	2	2	Bedson	Keats	Twain
241	242	85	2	2	Bedson	Twain	Sansome
242	243	85	2	2	Bedson	Sansome	Dicken's
243	244	85	2	2	Bedson	Dicken's	Shakespeare
244	245	80	2	2	Bedson	Shakespeare	Shakespeare
245	247	85	2	2	Bedson	Shakespeare	Dicken's
247	249	185	2	2	Bedson	Dicken's	Assinibione Gr
249	250	80	2	2	Bedson	Assinibione Gr	Coleridge
250	255	295	2	2	Bedson	Holt	Assinibione

255	250	295	2	2	Bedson	Assinibione	Holt
250	249	80	2	2	Bedson	Coleridge	Assinibione Gr
249	247	185	2	2	Bedson	Assinibione Gr	Dicken's
247	245	85	2	2	Bedson	Dicken's	Shakespeare
245	244	80	2	2	Bedson	Shakespeare	Shakespeare
244	243	85	2	2	Bedson	Shakespeare	Dicken's
243	242	85	2	2	Bedson	Dicken's	Sansome
242	241	85	2	2	Bedson	Sansome	Twain
241	239	85	2	2	Bedson	Twain	Keats
239	238	80	2	2	Bedson	Keats	Keats
238	237	80	2	2	Bedson	Keats	Allard
237	235	155	2	2	Bedson	Allard	Twain
235	234	260	2	2	Bedson	Twain	Portage
234	235	260	2	2	Bedson	Portage	Twain
235	237	155	2	2	Bedson	Twain	Allard
237	233	85	2	2	Allard	Bedson	Best
233	221	85	2	2	Allard	Best	Raquette
221	219	85	2	2	Allard	Raquette	Seaton
219	197	85	2	2	Allard	Seaton	Sumach
197	198	45	2	2	Allard	Sumach	Pontiac
198	199	80	2	2	Allard	Pontiac	Pontiac
199	165	110	2	2	Allard	Pontiac	Rouge
165	163	85	2	2	Rouge	Ericson	Ericson
163	162	85	2	2	Rouge	Ericson	Davis
162	160	85	2	2	Rouge	Davis	Byrd
160	159	212	2	2	Rouge	Byrd	Portage
159	160	212	2	2	Rouge	Portage	Byrd
160	162	85	2	2	Rouge	Byrd	Davis
162	163	85	2	2	Rouge	Davis	Ericson
163	165	85	2	2	Rouge	Ericson	Ericson
165	166	80	2	2	Rouge	Ericson	Galinee
166	167	85	2	2	Rouge	Galinee	Galinee
167	169	85	2	2	Rouge	Galinee	Peary
169	170	85	2	2	Rouge	Peary	Sansome
170	171	170	2	2	Sansome	Rouge	Peary
171	151	175	2	2	Sansome	Peary	Joliett
151	150	80	2	2	Sansome	Joliett	Westwood
150	120	220	2	2	Westwood	Sansome	Browning
120	119	95	2	2	Browning	Westwood	Shelley
119	110	160	2	2	Browning	Shelley	Wordsworth
110	96	85	2	2	Browning	Wordsworth	Pinehurst
96	110	85	2	2	Browning	Pinehurst	Wordsworth
110	119	160	2	2	Browning	Wordsworth	Shelley
119	120	95	2	2	Browning	Shelley	Westwood
120	175	350	2	2	Browning	Westwood	Wordsworth
175	120	350	2	2	Browning	Wordsworth	Westwood
120	121	85	2	2	Westwood	Browning	Wordsworth
121	122	85	2	2	Westwood	Wordsworth	Carlyle Cr
122	125	100	2	2	Westwood	Carlyle	Carlyle
125	126	90	2	2	Westwood	Carlyle	Sandburg
126	128	90	2	2	Westwood	Sandburg	Sandburg
128	130	85	2	2	Westwood	Sandburg	Haliburton
131	133	85	2	2	Westwood	Haliburton	Haliburton
133	134	80	2	2	Westwood	Haliburton	Kilmer
134	136	85	2	2	Westwood	Kilmer	Assiniboine
136	137	70	2	3	Assiniboine	Westwood	Westboine
137	136	70	2	3	Assiniboine	Westboine	Westwood
136	158	220	2	3	Assiniboine	Westwood	Caroll
158	186	112	2	3	Assiniboine	Caroll	Long Fellow
186	187	85	2	3	Assiniboine	Long Fellow	Long Fellow
187	188	85	2	3	Assiniboine	Long Fellow	Scalena
188	211	85	2	3	Assiniboine	Scalena	Rouge
211	212	275	2	3	Assiniboine	Rouge	Maureen
212	217	85	2	3	Assiniboine	Maureen	Barron
217	231	90	2	2	Assinibione	Barron	Raquette
231	255	180	2	2	Assinibione	Raquette	Bedson
255	231	180	2	2	Assinibione	Bedson	Raquette
231	217	90	2	2	Assinibione	Raquette	Barron
217	212	85	2	3	Assiniboine	Barron	Maureen
212	211	275	2	3	Assiniboine	Maureen	Rouge
211	188	85	2	3	Assiniboine	Rouge	Scalena
188	187	85	2	3	Assiniboine	Scalena	Long Fellow
187	186	85	2	3	Assiniboine	Long Fellow	Long Fellow
186	158	112	2	3	Assiniboine	Long Fellow	Caroll
158	136	220	2	3	Assiniboine	Caroll	Westwood
136	134	85	2	2	Westwood	Assiniboine	Kilmer

134	133	80	2	2	Westwood	Kilmer	Haliburton
133	131	85	2	2	Westwood	Haliburton	Haliburton
130	128	85	2	2	Westwood	Haliburton	Sandburg
128	126	90	2	2	Westwood	Sandburg	Sandburg
126	125	90	2	2	Westwood	Sandburg	Carlyle
125	122	100	2	2	Westwood	Carlyle	Carlyle
122	121	85	2	2	Westwood	Carlyle Cr	Wordsworth
121	120	85	2	2	Westwood	Wordsworth	Browning
120	150	220	2	2	Westwood	Browning	Sansome
150	148	85	2	2	Westwood	Sansome	Magellan
148	147	85	2	2	Westwood	Henday	Magellan
147	145	85	2	2	Westwood	Magellan	Henday
145	144	85	2	2	Westwood	Henday	Henday
144	143	85	2	2	Westwood	Henday	Admunsen
143	142	50	2	2	Westwood	Admunsen	Mcbey
142	112	130	2	2	Mcbey	Westwood	Shelley
112	102	90	2	2	Mcbey	Shelley	Kirkfield
102	100	120	2	2	Mcbey	Kirkfield	Banting
100	94	110	2	2	Mcbey	Banting	Daer
94	80	95	2	2	Mcbey	Daer	Golf
80	77	95	2	2	Mcbey	Golf	Greenacre
77	74	105	2	2	Mcbey	Greenacre	Country Club
74	73	220	2	2	Country Club	Mcbey	Portage
73	74	220	2	2	Country Club	Portage	Mcbey
74	72	30	2	2	Mcbey	Country Club	Summerlands
72	71	85	2	2	Mcbey	Summerlands	Woodland

COMPONENT 5

259	261	100	2	2	Allard	Buchanan	Stewart
261	263	145	2	2	Allard	Steward	St Charles
263	265	95	2	2	Allard	St Charles	Gagnon
265	263	95	2	2	Allard	Gagnon	St Charles
263	262	210	2	2	St Charles	Allard	Portage
262	263	210	2	2	St Charles	Portage	Allard
263	269	170	2	2	St Charles	Allard	Lepine
269	274	165	2	2	St Charles	Lepine	Sansume
274	280	165	2	2	St Charles	Sansome	Augier
280	274	165	2	2	St Charles	Augier	Sansome
274	269	165	2	2	St Charles	Sansome	Lepine
269	263	170	2	2	St Charles	Lepine	Allard
263	261	145	2	2	Allard	St Charles	Stewart

Appendix E
District #2: Vehicle Route For Priority II Street

FN	TN	LTH	P	#LN	C/D	BLOCK NAME	FROM	TO
7	10	88	2	2	C	Assiniboine	Winston	Cavell
10	12	78	2	2	C	Assiniboine	Bourkevale	Cavell
12	14	80	2	2	C	Assiniboine	Bourkevale	Riveroaks
14	16	100	2	2	C	Assiniboine	Riveroaks	Parkview
16	18	95	2	2	C	Assiniboine	Parkview	Roseberry
18	20	95	2	2	C	Assiniboine	Roseberry	Colligiate
20	22	95	2	2	C	Assiniboine	Colligiate	Ferry
22	21	425	2	2	C	Ferry	Assiniboine	Portage
21	23	212	1	8	D	Portage	Ferry	Library
23	25	95	1	8	D	Portage	Library	Douglas
25	27	27	1	8	D	Portage	Douglas	Albany
27	29	112	1	8	D	Portage	Albany	Deer Lodge
29	30	125	1	8	D	Portage	Deer Lodge	Deer Lodge
30	32	1740	1	8	D	Portage	Deer Lodge	Assiniboine
32	34	350	2	2	C	Assiniboine	Portage	Olive
34	35	88	2	2	C	Assiniboine	Olive	Pinewood
35	38	350	2	2	C	Assiniboine	Pinewood	Moray
38	40	120	2	2	C	Assiniboine	Moray	Lake
40	42	90	2	2	C	Assiniboine	Lake	Windham
42	44	125	2	2	C	Assiniboine	Windham	Thompson
44	47	80	2	2	C	Assiniboine	Thompson	Woodbridge
47	46	180	3	2	C	Woodbridge	Assiniboine	Old Mill
46	45	210	3	2	C	Woodbridge	Oll Mill	Portage
45	48	170	1	8	D	Portage	Woodbridge	Old Mill
48	49	175	1	8	D	Portage	Old Mill	Woodhaven
49	50	340	2	2	C	Woodhaven	Portage	Emo
50	51	150	2	2	C	Woodhaven	Emo	Glendale
51	54	260	2	2	C	Woodhaven	Glendale	Assiniboine
54	61	130	2	2	C	Assiniboine	Woodhaven	Sunnyside
61	62	110	2	2	C	Assiniboine	Sunnyside	Oakdean
62	67	105	2	2	C	Assiniboine	Oakdean	Harris
67	66	360	2	2	C	Harris	Assiniboine	Emo
66	64	145	2	2	C	Harris	Emo	Armour
65	63	310	2	2	C	Harris	Armour	Portage
63	68	200	1	8	D	Portage	Harris	Woodland
68	70	90	1	8	D	Portage	Woodland	Woodland
70	73	125	1	8	D	Portage	Woodland	Country Club
73	74	220	2	2	C	Country Club	Portage	Mcbey
74	72	30	2	2	C	Mcbey	Country Club	Summerlands
72	71	85	2	2	C	Mcbey	Summerlands	Woodland
71	72	85	2	2	C	Mcbey	Woodland	Summerland
72	74	30	2	2	C	Mcbey	Summerlands	Country Club
74	75	290	2	2	C	Country Club	Mcbey	Meadowside
75	83	150	2	2	C	Country Club	Meadowside	Pine Valley
83	92	112	2	2	C	Country Club	Park West	Pinehurst
92	91	183	2	2	C	Country Club	Pinehurst	Hagen
91	90	80	2	2	C	Country Club	Hagen	Pinehurst
90	88	90	2	2	C	Country Club	Pinehurst	Pine Valley
88	89	125	2	2	C	Country Club	Pine Valley	S. End
89	88	125	2	2	C	Country Club	S. End	Pine Valley
88	90	90	2	2	C	Country Club	Pine Valley	Pinehurst
90	91	80	2	2	C	Country Club	Pinehurst	Hagen
91	92	183	2	2	C	Country Club	Hagen	Pinehurst
92	83	112	2	2	C	Country Club	Pinehurst	Park West
83	75	150	2	2	C	Country Club	Park West	Meadowside
75	74	290	2	2	C	Country Club	Meadowside	Mcbey
74	77	105	2	2	C	Mcbey	Country Club	Greenacre
77	80	95	2	2	C	Mcbey	Greenacre	Golf
80	94	95	2	2	C	Mcbey	Golf	Daer
94	100	110	2	2	C	Mcbey	Daer	Banting
100	102	120	2	2	C	Mcbey	Banting	Kirkfield
102	112	90	2	2	C	Mcbey	Kirkfield	Shelley
112	142	130	2	2	C	Mcbey	Shelley	Westwood
142	143	50	2	2	C	Westwood	Mcbey	Admunsen

142	141	25	2	2	C	Westwood	Mcbey	Admunsen
141	140	90	2	2	C	Westwood	Admunsen	Byrd
140	139	212	2	2	C	Westwood	Byrd	Portage
139	140	212	2	2	C	Westwood	Portage	Byrd
140	141	90	2	2	C	Westwood	Byrd	Amundsen
141	142	25	2	2	C	Westwood	Admunsen	Mcbey
143	144	85	2	2	C	Westwood	Admunsen	Henday
144	145	85	2	2	C	Westwood	Henday	Henday
145	147	85	2	2	C	Westwood	Henday	Magellan
147	148	85	2	2	C	Westwood	Magellan	Magellan
148	150	85	2	2	C	Westwood	Magellan	Sansome
150	151	80	2	2	C	Sansome	Westwood	Joliett
151	171	175	2	2	C	Sansome	Joliett	Peary
171	170	170	2	2	C	Sansome	Peary	Rouge
170	201	245	2	2	C	Sansome	Rouge	Whitegates
201	222	175	2	2	C	Sansome	Whitegates	Raquette
222	242	175	2	2	C	Sansome	Raquette	Bedson
242	222	175	2	2	C	Sansome	Bedson	Raquette
222	201	175	2	2	C	Sansome	Raquette	Whitegates
201	170	245	2	2	C	Sansome	Whitegates	Rouge
170	172	85	2	2	C	Rouge	Sansome	Joliett
172	174	265	2	2	C	Rouge	Joliett	Browning
174	176	235	2	2	C	Rouge	Browning	Erlandson
176	208	137	2	2	C	Rouge	Erlandson	West
208	184	133	2	2	C	Rouge	West	Columbus
184	189	85	2	2	C	Rouge	Columbus	Scalena
189	211	75	2	2	C	Rouge	Scalena	Assiniboine
211	189	75	2	2	C	Rouge	Assiniboine	Scalena
189	184	85	2	2	C	Rouge	Scalena	Columbus
184	208	133	2	2	C	Rouge	Columbus	West
208	176	137	2	2	C	Rouge	West	Erlandson
176	174	235	2	2	C	Rouge	Erlandson	Browning
174	172	265	2	2	C	Rouge	Browning	Joliett
172	170	85	2	2	C	Rouge	Joliett	Sansome
170	169	85	2	2	C	Rouge	Sansome	Peary
169	167	85	2	2	C	Rouge	Peary	Galinee
167	166	85	2	2	C	Rouge	Galinee	Galinee
166	165	80	2	2	C	Rouge	Galinee	Ericson
165	199	110	2	2	C	Allard	Rouge	Pontiac
199	198	80	2	2	C	Allard	Pontiac	Pontiac
198	197	85	2	2	C	Allard	Pontiac	Sumach
197	219	85	2	2	C	Allard	Sumach	Seaton
219	221	85	2	2	C	Allard	Seaton	Raquette
221	233	85	2	2	C	Allard	Raquette	Best
233	237	85	2	2	C	Allard	Best	Bedson
237	238	80	2	2	C	Bedson	Allard	Keats
238	239	80	2	2	C	Bedson	Keats	Keats
239	241	85	2	2	C	Bedson	Keats	Twain
241	242	85	2	2	C	Bedson	Twain	Sansome
242	243	85	2	2	C	Bedson	Sansome	Dicken's
243	244	85	2	2	C	Bedson	Dicken's	Shakespeare
244	245	80	2	2	C	Bedson	Shakespeare	Shakespeare
245	247	85	2	2	C	Bedson	Shakespeare	Dicken's
247	249	185	2	2	C	Bedson	Dicken's	Assinibione Gr
249	250	80	2	2	C	Bedson	Assinibione Gr	Coleridge
250	255	295	2	2	C	Bedson	Holt	Assinibione
255	250	295	2	2	C	Bedson	Assinibione	Holt
250	249	80	2	2	C	Bedson	Coleridge	Assinibione Gr
249	247	185	2	2	C	Bedson	Assinibione Gr	Dicken's
247	245	85	2	2	C	Bedson	Dicken's	Shakespeare
245	244	80	2	2	C	Bedson	Shakespeare	Shakespeare
244	243	85	2	2	C	Bedson	Shakespeare	Dicken's
243	242	85	2	2	C	Bedson	Dicken's	Sansome
242	241	85	2	2	C	Bedson	Sansome	Twain
241	239	85	2	2	C	Bedson	Twain	Keats
239	238	80	2	2	C	Bedson	Keats	Keats
238	237	80	2	2	C	Bedson	Keats	Allard
237	235	155	2	2	C	Bedson	Allard	Twain
235	234	260	2	2	C	Bedson	Twain	Portage
234	258	700	1	8	D	Portage	Bedson	Buchanan
259	261	100	2	2	C	Allard	Buchanan	Stewart
261	263	145	2	2	C	Allard	Steward	St Charles
263	265	95	2	2	C	Allard	St Charles	Gagnon
265	263	95	2	2	C	Allard	Gagnon	St Charles
263	262	210	2	2	C	St Charles	Allard	Portage
262	263	210	2	2	C	St Charles	Portage	Allard

263	269	170	2	2	C	St Charles	Allard	Lepine
269	274	165	2	2	C	St Charles	Lepine	Sansume
274	280	165	2	2	C	St Charles	Sansome	Augier
280	274	165	2	2	C	St Charles	Augier	Sansome
274	269	165	2	2	C	St Charles	Sansome	Lepine
269	263	170	2	2	C	St Charles	Lepine	Allard
263	261	145	2	2	C	Allard	St Charles	Stewart
261	259	100	2	2	C	Allard	Steward	Allard
258	234	700	1	8	D	Portage	Buchanna	Bedson
234	235	260	2	2	C	Bedson	Portage	Twain
235	237	155	2	2	C	Bedson	Twain	Allard
237	233	85	2	2	C	Allard	Bedson	Best
233	221	85	2	2	C	Allard	Best	Raquette
221	219	85	2	2	C	Allard	Raquette	Seaton
219	197	85	2	2	C	Allard	Seaton	Sumach
197	198	45	2	2	C	Allard	Sumach	Pontiac
198	199	80	2	2	C	Allard	Pontiac	Pontiac
199	165	110	2	2	C	Allard	Pontiac	Rouge
165	163	85	2	2	C	Rouge	Ericson	Ericson
163	162	85	2	2	C	Rouge	Ericson	Davis
162	160	85	2	2	C	Rouge	Davis	Byrd
160	159	212	2	2	C	Rouge	Byrd	Portage
159	160	212	2	2	C	Rouge	Portage	Byrd
160	162	85	2	2	C	Rouge	Byrd	Davis
162	163	85	2	2	C	Rouge	Davis	Ericson
163	165	85	2	2	C	Rouge	Ericson	Ericson
165	166	80	2	2	C	Rouge	Ericson	Galinee
166	167	85	2	2	C	Rouge	Galinee	Galinee
167	169	85	2	2	C	Rouge	Galinee	Peary.
169	170	85	2	2	C	Rouge	Peary	Sansome
170	171	170	2	2	C	Sansome	Rouge	Peary
171	151	175	2	2	C	Sansome	Peary	Joliett
151	150	80	2	2	C	Sansome	Joliett	Westwood
150	120	220	2	2	C	Westwood	Sansome	Browning
120	119	95	2	2	C	Browning	Westwood	Shelley
119	110	160	2	2	C	Browning	Shelley	Wordsworth
110	96	85	2	2	C	Browning	Wordsworth	Pinehurst
96	110	85	2	2	C	Browning	Pinehurst	Wordsworth
110	119	160	2	2	C	Browning	Wordsworth	Shelley
119	120	95	2	2	C	Browning	Shelley	Westwood
120	175	350	2	2	C	Browning	Westwood	Wordsworth
175	120	350	2	2	C	Browning	Wordsworth	Westwood
120	121	85	2	2	C	Westwood	Browning	Wordsworth
121	122	85	2	2	C	Westwood	Wordsworth	Carlyle Cr
122	125	100	2	2	C	Westwood	Carlyle	Carlyle
125	126	90	2	2	C	Westwood	Carlyle	Sandburg
126	128	90	2	2	C	Westwood	Sandburg	Sandburg
128	130	85	2	2	C	Westwood	Sandburg	Haliburton
131	133	85	2	2	C	Westwood	Haliburton	Haliburton
133	134	80	2	2	C	Westwood	Haliburton	Kilmer
134	136	85	2	2	C	Westwood	Kilmer	Assiniboine
136	137	70	2	3	C	Assiniboine	Westwood	Westboine
137	136	70	2	3	C	Assiniboine	Westboine	Westwood
136	158	220	2	3	C	Assiniboine	Westwood	Caroll
158	186	112	2	3	C	Assiniboine	Caroll	Long Fellow
186	187	85	2	3	C	Assiniboine	Long Fellow	Long Fellow
187	188	85	2	3	C	Assiniboine	Long Fellow	Scalena
188	211	85	2	3	C	Assiniboine	Scalena	Rouge
211	212	275	2	3	C	Assiniboine	Rouge	Maureen
212	217	85	2	3	C	Assiniboine	Maureen	Barron
217	231	90	2	2	C	Assinibione	Barron	Raquette
231	255	180	2	2	C	Assinibione	Raquette	Bedson
255	231	180	2	2	C	Assinibione	Bedson	Raquette
231	217	90	2	2	C	Assinibione	Raquette	Barron
217	212	85	2	3	C	Assiniboine	Barron	Maureen
212	211	275	2	3	C	Assiniboine	Maureen	Rouge
211	188	85	2	3	C	Assiniboine	Rouge	Scalena
188	187	85	2	3	C	Assiniboine	Scalena	Long Fellow
187	186	85	2	3	C	Assiniboine	Long Fellow	Long Fellow
186	158	112	2	3	C	Assiniboine	Long Fellow	Caroll
158	136	220	2	3	C	Assiniboine	Caroll	Westwood
136	134	85	2	2	C	Westwood	Assiniboine	Kilmer
134	133	80	2	2	C	Westwood	Kilmer	Haliburton
133	131	85	2	2	C	Westwood	Haliburton	Haliburton
130	128	85	2	2	C	Westwood	Haliburton	Sandburg
128	126	90	2	2	C	Westwood	Sandburg	Sandburg

126	125	90	2	2	C	Westwood	Sandburg	Carlyle
125	122	100	2	2	C	Westwood	Carlyle	Carlyle
122	121	85	2	2	C	Westwood	Carlyle Cr	Wordsworth
121	120	85	2	2	C	Westwood	Wordsworth	Browning
120	150	220	2	2	C	Westwood	Browning	Sansome
150	148	85	2	2	C	Westwood	Sansome	Magellan
148	147	85	2	2	C	Westwood	Henday	Magellan
147	145	85	2	2	C	Westwood	Magellan	Henday
145	144	85	2	2	C	Westwood	Henday	Henday
144	143	85	2	2	C	Westwood	Henday	Admunsen
143	142	50	2	2	C	Westwood	Admunsen	Mcbey
142	112	130	2	2	C	Mcbey	Westwood	Shelley
112	102	90	2	2	C	Mcbey	Shelley	Kirkfield
102	100	120	2	2	C	Mcbey	Kirkfield	Banting
100	94	110	2	2	C	Mcbey	Banting	Daer
94	80	95	2	2	C	Mcbey	Daer	Golf
80	77	95	2	2	C	Mcbey	Golf	Greenacre
77	74	105	2	2	C	Mcbey	Greenacre	Country Club
74	73	220	2	2	C	Country Club	Mcbey	Portage
73	70	125	1	8	D	Portage	Country Club	Woodland
70	68	90	1	8	D	Portage	Woodland	Woodland
68	63	200	1	8	D	Portage	Woodland	Harris
63	65	310	2	2	C	Harris	Portage	Armour
64	66	145	2	2	C	Harris	Armour	Emo
66	67	360	2	2	C	Harris	Emo	Assiniboine
67	62	105	2	2	C	Assiniboine	Harris	Oakdean
62	61	110	2	2	C	Assiniboine	Oakdean	Sunnyside
61	54	130	2	2	C	Assiniboine	Sunnyside	Woodhaven
54	51	260	2	2	C	Woodhaven	Assiniboine	Glendale
51	50	75	2	2	C	Woodhaven	Glendale	Emo
50	49	340	2	2	C	Woodhaven	Emo	Portage
49	48	175	1	8	D	Portage	Woodhaven	Old Mill
48	45	170	1	8	D	Portage	Old Mill	Woodbridge
45	46	210	3	2	C	Woodbridge	Portage	Old Mill
46	47	180	3	2	C	Woodbridge	Old Mill	Assiniboine
47	44	80	2	2	C	Assiniboine	Woodbridge	Thompson
44	42	125	2	2	C	Assiniboine	Thompson	Windham
42	40	90	2	2	C	Assiniboine	Windham	Lake
40	38	120	2	2	C	Assiniboine	Lake	Moray
38	35	350	2	2	C	Assiniboine	Moray	Pinewood
35	34	88	2	2	C	Assiniboine	Pinewood	Olive
34	32	350	2	2	C	Assiniboine	Olive	Portage
32	30	1740	1	8	D	Portage	Assiniboine	Deer Lodge
30	29	125	1	8	D	Portage	Deer Lodge	Deer Lodge
29	27	112	1	8	D	Portage	Deer Lodge	Albany
27	25	27	1	8	D	Portage	Albany	Douglas
25	23	95	1	8	D	Portage	Douglas	Library
23	21	212	1	8	D	Portage	Library	Ferry
21	22	425	2	2	C	Ferry	Portage	Assiniboine
22	20	95	2	2	C	Assiniboine	Ferry	Colligiate
20	18	95	2	2	C	Assiniboine	Colligiate	Roseberry
18	16	95	2	2	C	Assiniboine	Roseberry	Parkview
16	14	100	2	2	C	Assiniboine	Parkview	Riveroaks
14	12	80	2	2	C	Assiniboine	Riveroaks	Bourkevale
12	10	78	2	2	C	Assiniboine	Bourkevale	Cavell
10	7	88	2	2	C	Assiniboine	Cavell	Winston

* C - CLEAR OUT D - DEADHEADING

TOTAL DEADHEADING: 8,322 M

Appendix F
District #2: Priority III Streets

FN	TN	LTH	P	#LN	BLOCK NAME	FROM	TO
1	3	237	3	2	Riverbend	Portage	Garden
2	3	130	3	2	Garden	Portage	Riverbend
3	1	237	3	2	Riverbend	Graden	Portage
3	2	130	3	2	Graden	Riverbend	Portage
3	4	243	3	2	Riverbend	Graden	Portage
4	3	243	3	2	Riverbend	Portage	Graden
5	6	225	3	2	Winston	Portage	Parkside
6	5	225	3	2	Winston	Parkside	Portage
6	7	175	3	2	Winston	Parkside	Assiniboine
6	8	270	3	2	Parkside	Winston	Assiniboine
7	6	175	3	2	Winston	Assiniboine	Parkside
8	6	270	3	2	Parkside	Assiniboine	Winston
9	10	370	3	2	Cavell	Portage	Assiniboine
10	9	370	3	2	Cavell	Assiniboine	Portage
11	12	340	3	2	Bourkevale	Portage	Assiniboine
12	11	340	3	2	Bourkevale	Assiniboine	Portage
13	14	350	3	2	Riveroaks	Portage	Assiniboine
14	13	350	3	2	Riveroaks	Assiniboine	Portage
15	16	370	3	2	Parkview	Portage	Assiniboine
16	15	370	3	2	Parkview	Assiniboine	Portage
17	18	390	3	2	Roseberry	Portage	Assiniboine
18	17	390	3	2	Roseberry	Assiniboine	Portage
19	20	370	3	4	Colligiate	Portage	Assiniboine
20	19	370	3	4	Colligiate	Assiniboine	Portage
23	24	120	3	2	Library	Portage	S. End
24	23	120	3	2	Library	S. End	Portage
25	26	430	3	2	Douglas	Portage	S. End
26	25	430	3	2	Douglas	S. End	Portage
27	28	150	3	2	Albany	Portage	Deer Lodge
28	27	150	3	2	Albany	S. End	Portage
29	31	670	3	2	Deer Lodge	Portage	Portage
31	29	670	3	2	Deer Lodge	Portage	Portage
33	34	150	3	2	Olive	Assiniboine	Portage
34	33	150	3	2	Olive	Assiniboine	Portage
35	37	420	3	2	Pinewood	Assiniboine	Moray
36	37	150	3	2	Moray	Portage	Pinewood
37	35	420	3	2	Pinewood	Moray	Lake
37	36	150	3	2	Moray	Pinewood	Portage
37	38	130	3	2	Moray	Pinewood	Assiniboine
38	37	130	3	2	Moray	Assiniboine	Pinewood
39	40	280	3	2	Lake	Portage	Assiniboine
40	39	280	3	2	Lake	Assiniboine	Portage
41	42	310	3	2	Windham	Portage	Assiniboine
42	41	310	3	2	Windham	Assiniboine	Portage
43	44	260	3	2	Thompson	Portage	Assiniboine
44	43	260	3	2	Thompson	Assiniboine	Portage
45	46	210	3	2	Woodbridge	Portage	Old Mill
46	45	210	3	2	Woodbridge	Old Mill	Portage
46	47	180	3	2	Woodbridge	Old Mill	Assiniboine
46	48	390	3	2	Old Mill	Woodbridge	Portage
47	46	180	3	2	Woodbridge	Assiniboine	Old Mill
48	46	390	3	2	Old Mill	Portage	Woodbridge
50	60	150	3	2	Emo	Woodhaven	Sunnyside
51	53	220	3	2	Glendale	Woodhaven	Assiniboine
52	53	370	3	2	Ashcroft	E. End	Assiniboine
53	51	220	3	2	Glendale	Assiniboine	Woodhaven
53	52	370	3	2	Ashcroft	Assiniboine	E. End
53	54	75	3	2	Assiniboine	Glendale	Woodhaven
54	53	125	3	2	Assiniboine	Woodhaven	Glendale
55	56	110	3	2	Oakdean	Portage	Oakdean Cr
56	55	110	3	2	Oakdean	Oakdean Cr	Portage
56	57	105	3	2	Oakdean	Oakdean Cr	Oakdean Cr
56	58	220	3	2	Oakdean Cr	Oakdean	Oakdean
57	56	105	3	2	Oakdean	Oakdean Cr	Oakdean Cr

57	59	200	3	2	Oakdean	Oakdean Cr	Emo
58	56	220	3	2	Oakdean Cr	Oakdean	Oakdean
59	57	220	3	2	Oakdean	Emo	Portage
59	60	110	3	2	Emo	Oakdean	Sunnyside
59	62	360	3	2	Oakdean	Emo	Assiniboine
59	66	105	3	2	Emo	Oakdean	Harris
60	59	110	3	2	Emo	Sunnyside	Oakdean
60	50	150	3	2	Emo	Sunnyside	Woodhaven
60	61	360	3	2	Sunnyside	Emo	Assiniboine
61	60	360	3	2	Sunnyside	Assiniboine	Emo
62	59	360	3	2	Oakdean	Assiniboine	Emo
63	64	460	3	2	Armour	Portage	Harris
64	63	460	3	2	Armour	Harris	Portage
66	59	105	3	2	Emo	Harris	Oakdean
68	69	760	3	2	Woodland	Portage	Summerlands
69	68	760	3	2	Woodland	Summerlands	Portage
69	71	212	3	2	Woodland	Summerlands	Mcbey
69	72	330	3	2	Summerlands	Woodland	Mcbey
70	71	200	3	2	Woodland	Portage	Mcbey
71	69	212	3	2	Woodland	Mcbey	Summerlands
71	70	200	3	2	Woodland	Mcbey	Portage
72	69	330	3	2	Summerlands	Mcbey	Woodland
75	78	100	3	2	Meadowside	Country Club	Greenacre
76	77	239	3	2	Greenacre	Portage	Mcbey
77	76	237	3	2	Greenacre	Mcbey	Portage
77	78	190	3	2	Greenacre	Mcbey	Meadowside
78	75	100	3	2	Meadowside	Greenacre	Country Club
78	77	290	3	2	Greenacre	Meadowside	Mcbey
78	82	55	3	2	Meadowside	Greenacre	Park West
79	80	250	3	2	Golf	Portage	Mcbey
80	79	250	3	2	Golf	Mcbey	Portage
80	81	290	3	2	Golf	Mcbey	Meadowside
81	80	290	3	2	Golf	Meadowside	Mcbey
81	82	37	3	2	Meadowside	Golf	Greenacre
82	78	55	3	2	Meadowside	Park West	Greenacre
82	81	37	3	2	Meadowside	Greenacre	Golf
82	83	330	3	2	Park West	Meadowside	Country Club
83	82	330	3	2	Park West	Country Club	Meadowside
83	84	133	3	2	Pine Valley	Country Club	Vardon
84	83	133	3	2	Pine Valley	Vardon	Country Club
84	85	140	3	2	Vardon	Pine Valley	E. End
84	87	204	3	2	Pine Valley	Vardon	Braid
85	84	140	3	2	Vardon	E. End	Pine Valley
86	87	60	3	2	Braid	E. End	Pine Valley
87	84	204	3	2	Pine Valley	Braid	Vardon
87	86	60	3	2	Braid	Pine Valley	E. End
87	88	133	3	2	Pine Valley	Braid	Country Club
88	87	133	3	2	Pine Valley	Country Club	Braid
90	97	225	3	2	Pinehurst	Country Club	Compton
91	95	360	3	2	Hagen	Country Club	Pinehurst
92	95	150	3	2	Pinehurst	Country Club	Hagen
93	94	260	3	2	Daer	Portage	Mcbey
94	93	260	3	2	Daer	Mcbey	Portage
95	91	360	3	2	Hagen	Pinehurst	Country Club
95	92	150	3	2	Pinehurst	Hagen	Country Club
95	96	337	3	2	Pinehurst	Hagen	Browning
96	95	337	3	2	Pinehurst	Browning	Hagen
96	97	115	3	2	Pinehurst	Browning	Compton
97	90	225	3	2	Pinehurst	Compton	Country Club
97	96	115	3	2	Pinehurst	Compton	Browning
97	98	60	3	2	Compton	Pinehurst	W. End
98	97	60	3	2	Compton	W. End	Pinehurst
99	100	280	3	2	Banting	Portage	Mcbey
100	99	280	3	2	Banting	Mcbey	Portage
101	102	285	3	2	Kirkfield	Portage	Mcbey
102	101	285	3	2	Kirkfield	Mcbey	Portage
102	103	230	3	2	Kirkfield	Mcbey	Byron
103	102	230	3	2	Kirkfield	Byron	Mcbey
103	104	88	3	2	Kirkfield	Byron	Byron
103	105	320	3	2	Byron	Kirkfield	Kirkfield
104	103	88	3	2	Kirkfield	Byron	Byron
104	106	75	3	2	Kirkfield	Byron	Wordsworth
105	103	320	3	2	Byron	Kirkfield	Kirkfield
106	104	75	3	2	Kirkfield	Wordsworth	Byron
106	107	145	3	2	Wordsworth	Kirkfield	Aldrich
106	113	90	3	2	Wordsworth	Kirkfield	Shelley

107	106	145	3	2	Wordsworth	Aldrich	Kirkfield
107	109	85	3	2	Wordsworth	Aldrich	Wordsworth Pl
107	116	106	3	2	Aldrich	Wordsworth	Shelley
108	109	80	3	2	Wordsworth Pl	W. End	Wordsworth
109	107	85	3	2	Wordsworth	Wordsworth Pl	Aldrich
109	108	80	3	2	Wordsworth Pl	Wordsworth	W. End
109	110	88	3	2	Wordsworth	Wordsworth Pl	Browning
110	109	88	3	2	Wordsworth	Browning	Wordsworth Pl
110	121	280	3	2	Wordsworth	Browning	Westwood
111	112	305	3	2	Shelley	Portage	Mcbey
112	111	90	3	2	Shelley	Mcbey	Portage
112	113	395	3	2	Shelley	Mcbey	Wordsworth
113	106	90	3	2	Wordsworth	Shelley	Kirkfield
113	112	395	3	2	Shelley	Wordsworth	Mcbey
113	115	50	3	2	Shelley	Wordsworth	Poole
114	115	40	3	2	Poole	W. End	Shelley
115	113	50	3	2	Shelley	Poole	Wordsworth
115	114	40	3	2	Poole	Shelley	W End
115	116	38	3	2	Shelley	Poole	Aldrich
116	107	160	3	2	Aldrich	Shelley	Wordsworth
116	115	38	3	2	Shelley	Aldrich	Poole
116	118	55	3	2	Shelley	Aldrich	Drummond
117	118	40	3	2	Drummond	W End	Shelley
118	116	55	3	2	Shelley	Drummond	Aldrich
118	117	40	3	2	Drummond	Shelley	W End
118	119	125	3	2	Shelley	Drummond	Browning
119	118	125	3	2	Shelley	Browning	Drummond
121	110	280	3	2	Wordsworth	Westwood	Browning
121	152	112	3	2	Wordsworth	Westwood	Carroll
122	123	60	3	2	Carlyle Cr	Westwood	W End
122	124	330	3	2	Carlyle	Westwood	Westwood
123	122	60	3	2	Carlyle Cr	W End	Westwood
124	122	330	3	2	Carlyle	Westwood	Westwood
124	153	180	3	2	Thackeray	Westwood	Caroll
126	129	210	3	2	Sandburg	Westwood	Westwood
126	154	310	3	2	Leacock	Westwood	Caroll
127	174	130	3	2	Browning	Reynolds	Rouge
127	205	200	3	2	Reynolds	Browning	Whitegates
127	206	112	3	2	Browning	Reynolds	Whitegates
128	155	210	3	2	Frost	Westwood	Caroll
129	126	210	3	2	Sandburg	Westwood	Westwood
130	133	340	3	2	Robert Serv.	Westwood	Westwood
132	133	210	3	2	Haliburton	Westwood	Westwood
133	130	340	3	2	Robert Serv.	Westwood	Westwood
133	132	210	3	2	Haliburton	Westwood	Westwood
134	135	70	3	2	Connor	Westwood	E End
134	157	215	3	2	Kilmer	Westwood	Caroll
135	134	70	3	2	Connor	E End	Westwood
137	138	50	3	2	Westboine	Assiniboine	S End
138	137	50	3	2	Westboine	S End	Assiniboine
140	161	250	3	2	Byrd	Westwood	Davis
141	143	410	3	2	Amundsen	Westwood	Westwood
143	141	410	3	2	Amundsen	Westwood	Westwood
144	146	410	3	2	Henday	Westwood	Westwood
146	144	410	3	2	Henday	Westwood	Westwood
147	149	410	3	2	Magellan	Westwood	Westwood
149	147	410	3	2	Magellan	Westwood	Westwood
151	173	335	3	2	Joliett	Sansome	Wordsworth
152	121	112	3	2	Wordsworth	Caroll	Westwood
152	153	125	3	2	Caroll	Wordsworth	Thackeray
152	175	425	3	2	Wordsworth	Caroll	Browning
153	124	180	3	2	Thackeray	Caroll	Westwood
153	152	125	3	2	Caroll	Thackeray	Wordsworth
153	154	80	3	2	Caroll	Thackeray	Leacock
154	126	210	3	2	Leacock	Caroll	Westwood
154	153	80	3	2	Caroll	Leacock	Thackeray
154	155	85	3	2	Caroll	Leacock	Frost
155	128	210	3	2	Frost	Caroll	Westwood
155	154	85	3	2	Caroll	Frost	Leacock
155	156	35	3	2	Caroll	Frost	Columbus
156	155	35	3	2	Caroll	Columbus	Frost
156	157	205	3	2	Caroll	Columbus	Kilmer
156	181	85	3	2	Columbus	Caroll	Erlandson
157	134	215	3	2	Kilmer	Caroll	Westwood
157	156	205	3	2	Caroll	Kilmer	Columbus
157	158	110	3	2	Caroll	Kilmer	Assiniboine

157	185	180	3	2	Kilmer	Caroll	Long Fellow
158	157	110	3	2	Caroll	Assiniboine	Kilmer
160	161	160	3	2	Byrd	Rouge	Davis
161	140	250	3	2	Byrd	Davis	Westwood
161	160	160	3	2	Byrd	Davis	Rouge
161	162	250	3	2	Davis	Byrd	Rouge
162	161	250	3	2	Davis	Rouge	Byrd
164	165	420	3	2	Ericson	Rouge	Rouge
165	164	420	3	2	Ericson	Rouge	Rouge
166	168	420	3	2	Galinee	Rouge	Rouge
168	166	420	3	2	Galinee	Rouge	Rouge
169	171	250	3	2	Peary	Rouge	Sansome
171	169	250	3	2	Peary	Sansome	Rouge
172	173	85	3	2	Joliett	Rouge	Wordsworth
173	151	175	3	2	Joliett	Wordsworth	Sansome
173	172	85	3	2	Joliett	Wordsworth	Rouge
173	175	250	3	2	Wordsworth	Joliett	Browning
174	175	80	3	2	Browning	Rouge	Wordsworth
174	127	130	3	2	Browning	Rouge	Reynolds
175	152	425	3	2	Wordsworth	Browning	Caroll
175	173	250	3	2	Wordsworth	Browning	Joliett
175	174	80	3	2	Browning	Wordsworth	Rouge
176	177	80	3	2	Erlandson	Rouge	Cortez
177	176	80	3	2	Erlandson	Portage	Rouge
177	178	210	3	2	Erlandson	Cortez	Cortez
177	179	370	3	2	Cortez	Erlandson	Erlandson
177	182	75	3	2	Cook	Erlandson	Bering
178	177	210	3	2	Erlandson	Cortez	Cortez
178	180	125	3	2	Erlandson	Cortez	Bering
179	177	370	3	2	Cortez	Erlandson	Erlandson
180	178	125	3	2	Erlandson	Bering	Cortez
180	181	82	3	2	Erlandson	Bering	Columbus
180	182	260	3	2	Bering	Erlandson	Cook
181	156	85	3	2	Columbus	Erlandson	Caroll
181	180	82	3	2	Erlandson	Columbus	Bering
181	183	375	3	2	Columbus	Erlandson	Cook
182	177	75	3	2	Cook	Bering	Erlandson
182	180	260	3	2	Bering	Cook	Erlandson
182	183	200	3	2	Cook	Bering	Columbus
183	181	375	3	2	Columbus	Cook	Erlandson
183	182	200	3	2	Cook	Columbus	Bering
183	184	80	3	2	Columbus	Cook	Rouge
184	183	80	3	2	Columbus	Rouge	Cook
185	157	180	3	2	Kilmer	Long Fellow	Caroll
185	186	115	3	2	Long Fellow	Kilmer	Assiniboine
185	187	205	3	2	Long Fellow	Kilmer	Assiniboine
186	185	115	3	2	Long Fellow	Assiniboine	Kilmer
187	185	205	3	2	Long Fellow	Assiniboine	Kilmer
188	189	180	3	2	Scalena	Assiniboine	Rouge
189	188	180	3	2	Scalena	Rouge	Assiniboine
190	191	125	3	2	Sumach	Portage	Canoe
191	190	125	3	2	Sumach	Portage	Canoe
191	192	78	3	2	Sumach	Canoe	Canoe
191	193	370	3	2	Canoe	Sumach	Sumach
192	191	78	3	2	Sumach	Canoe	Canoe
192	194	125	3	2	Sumach	Canoe	Manitou
193	191	370	3	2	Canoe	Sumach	Sumach
194	192	125	3	2	Sumach	Manitou	Canoe
194	195	75	3	2	Sumach	Manitou	Manitou
194	196	380	3	2	Manitou	Sumach	Sumach
195	194	75	3	2	Sumach	Manitou	Manitou
195	197	80	3	2	Sumach	Manitou	Allard
196	194	380	3	2	Manitou	Sumach	Sumach
197	195	80	3	2	Sumach	Allard	Manitou
198	200	110	3	2	Pontiac	Allard	Allard
200	198	110	3	2	Pontiac	Allard	Allard
201	202	85	3	2	Whitegates	Sansome	Belcourt
202	201	85	3	2	Whitegates	Belcourt	Sansome
202	203	85	3	2	Whitegates	Belcourt	Belcourt
202	204	380	3	2	Belcourt	Whitegates	Whitegates
203	202	85	3	2	Whitegates	Belcourt	Belcourt
203	205	85	3	2	Whitegates	Belcourt	Reynolds
204	202	380	3	2	Belcourt	Whitegate	Whitegate
205	203	85	3	2	Whitegates	Reynolds	Belcourt
205	206	85	3	2	Whitegates	Reynolds	Browning
205	127	200	3	2	Reynolds	Whitegates	Browning

206	127	112	3	2	Browning	Whitegates	Reynolds
206	205	85	3	2	Whitegates	Browning	Reynolds
206	207	410	3	2	Whitegates	Browning	Barron
206	226	175	3	2	Browning	Whitegates	Raquette
207	206	410	3	2	Whitegates	Barron	Browning
207	209	112	3	2	Whitegates	Barron	West
207	216	160	3	2	Barron	Whitegates	Maureen
208	209	90	3	2	West	Rouge	Whitegates
209	207	112	3	2	Whitegates	West	Barron
209	208	90	3	2	West	Whitegates	Rouge
209	210	75	3	2	Whitegates	West	Cora
209	215	140	3	2	West	Whitegates	Maureen
210	209	75	3	2	Whitegates	Cora	West
210	213	235	3	2	Whitegates	Cora	Maureen
210	214	150	3	2	Cora	Whitegates	Maureen
212	213	85	3	2	Maureen	Assiniboine	Whitegates
213	210	135	3	2	Whitegates	Maureen	Cora
213	214	78	3	2	Maureen	Whitrgates	Cora
213	212	85	3	2	Maureen	Whitegates	Assiniboine
214	210	150	3	2	Cora	Maureen	Whitegates
214	213	78	3	2	Maureen	Cora	Whitegates
214	215	80	3	2	Maureen	Cora	West
215	209	140	3	2	West	Maureen	Whitegates
215	214	80	3	2	Maureen	West	Cora
215	216	80	3	2	Maureen	West	Barron
216	207	160	3	2	Barron	Maureen	Whitegates
216	215	80	3	2	Maureen	Barron	West
216	217	395	3	2	Barron	Maureen	Assiniboine
217	216	395	3	2	Barron	Assiniboine	Maureen
218	219	480	3	2	Seaton	Portage	Allard
219	218	480	3	2	Seaton	Allard	Portage
220	221	470	3	2	Raquette	Portage	Allard
221	220	470	3	2	Raquette	Allard	Portage
222	223	85	3	2	Raquette	Sansome	Haultain
223	222	85	3	2	Raquette	Haultain	Sansome
223	224	160	3	2	Raquette	Haultain	Haultain
223	225	330	3	2	Haultain	Raquette	Raquette
223	248	320	3	2	Addison Cr	Raquette	Addison
224	223	160	3	2	Raquette	Haultain	Haultain
224	226	95	3	2	Raquette	Haultain	Browning
225	223	330	3	2	Haultain	Raquette	Raquette
226	206	75	3	2	Browning	Raquette	Whitegates
226	224	95	3	2	Raquette	Browning	Haultain
226	227	95	3	2	Raquette	Browning	Trigwell
227	226	95	3	2	Raquette	Trigwell	Browning
227	228	55	3	2	Raquette	Trigwell	Addsion Cr
227	229	250	3	2	Trigwell	Raquette	Raquette
228	227	55	3	2	Raquette	Addsion Cr	Trigwell
228	229	30	3	2	Raquette	Addsion Cr	Trigwell
228	248	245	3	2	Addsion Cr	Raquette	Addsion
229	227	250	3	2	Trigwell	Raquette	Raquette
229	228	30	3	2	Raquette	Trigwell	Addsion Cr
229	230	260	3	2	Raquette	Trigwell	Holt
230	229	260	3	2	Raquette	Holt	Trigwell
230	231	110	3	2	Raquette	Holt	Assinibione
230	250	360	3	2	Holt	Raquette	Bedson
231	230	110	3	2	Raquette	Assinibione	Holt
232	233	450	3	2	Best	Portage	Allard
233	232	450	3	2	Best	Allard	Portage
235	236	355	3	2	Twain	Bedson	Allard
236	235	355	3	2	Twain	Allard	Bedson
236	237	200	3	2	Allard	Twain	Bedson
236	241	495	3	2	Twain	Allard	Bedson
237	236	200	3	2	Allard	Bedson	Twain
238	240	340	3	2	Keats	Bedson	Bedson
240	238	340	3	2	Keats	Bedson	Bedson
241	236	495	3	2	Twain	Bedson	Allard
242	256	240	3	2	Sansome	Bedson	W End
243	247	640	3	2	Dicken's	Bedson	Bedson
244	246	320	3	2	Shakespeare	Bedson	Bedson
246	244	320	3	2	Shakespeare	Bedson	Bedson
247	243	640	3	2	Dicken's	Bedson	Bedson
247	248	80	3	2	Addsion	Bedson	Addsion Cr
248	223	320	3	2	Addsion Cr	Addsion	Raquette
248	228	245	3	2	Addsion Cr	Addsion	Raquette
248	247	80	3	2	Addsion	Addsion Cr	Bedson

249	257	260	3	2	Assinibione Gr	Bedson	W End
250	230	360	3	2	Holt	Bedson	Raquette
250	251	315	3	2	Coleridge	Bedson	Galsworthy
251	250	315	3	2	Coleridge	Galsworthy	Bedson
251	252	110	3	2	Galsworthy	Coleridge	E End
251	253	85	3	2	Coleridge	Galsworthy	Masefield
252	251	110	3	2	Galsworthy	E End	Coleridge
253	251	85	3	2	Coleridge	Masefield	Galsworthy
253	254	80	3	2	Masefield	Coleridge	E End
253	255	230	3	2	Coleridge	Masefield	Assinibione
254	253	80	3	2	Masefield	E End	Coleridge
255	253	236	3	2	Coleridge	Assinibione	Masefield
256	242	240	3	2	Sansome	W End	Bedson
257	249	260	3	2	Assinibione Gr	W End	Bedson
258	259	260	3	2	Buchanan	Allard	Portage
259	258	760	3	2	Buchanan	Allard	Portage
260	261	238	3	2	Stewart	Portage	Allard
261	260	238	3	2	Stewart	Allard	Portage
264	265	100	3	2	Gagnon	Portage	Allard
265	264	100	3	2	Gagnon	Allard	Portage
265	268	75	3	2	Gagnon	Allard	Dennis
266	268	60	3	2	Dennis	W End	Gagnon
267	271	60	3	2	Brendalee	W End	Gagnon
268	265	75	3	2	Gagnon	Dennis	Allard
268	266	60	3	2	Dennis	Gagnon	W End
268	271	85	3	2	Gagnon	Dennis	Brendalee
269	270	50	3	2	Lepine	St Charles	E End
269	271	100	3	2	Lepine	St Charles	Gagnon
270	269	50	3	2	Lepine	E End	St Charles
271	268	85	3	2	Gagnon	Brendalee	Dennis
271	267	60	3	2	Brendalee	Gagnon	W End
271	269	100	3	2	Lepine	Gagnon	St Charles
271	273	80	3	2	Gagnon	Lepine	Suzanne
272	273	60	3	2	Suzanne	W End	Gagnon
273	271	80	3	2	Gagnon	Suzanne	Lepine
273	272	60	3	2	Suzanne	Gagnon	W End
273	276	70	3	2	Gagnon	Suzanne	Sansome
274	275	50	3	2	Sansome	St Charles	E End
274	276	100	3	2	Sansome	St Charles	Gagnon
275	274	50	3	2	Sansome	E End	St Charles
276	273	70	3	2	Gagnon	Sansome	Suzanne
276	274	100	3	2	Sansome	Gagnon	St Charles
276	278	70	3	2	Gagnon	Sansome	Jeannette
277	278	70	3	2	Jeanette	W End	Gagnon
278	276	70	3	2	Gagnon	Jeanette	Sansome
278	277	70	3	2	Jeanette	Gagnon	W End
278	281	85	3	2	Gagnon	Jeanette	Augier
279	280	100	3	2	Augier	E End	St Charles
280	279	100	3	2	Augier	St Charles	E End

TOTAL CLEARED LENGTH: 89,610m

Appendix G
District #2: Components of Priority III Streets

FN	TN	LTH	P	#LN	BLOCK NAME	FROM	TO
COMPONENT 1							
1	3	237	3	2	Riverbend	Portage	Garden
3	2	130	3	2	Graden	Riverbend	Portage
2	3	130	3	2	Garden	Portage	Riverbend
3	4	243	3	2	Riverbend	Graden	Portage
4	3	243	3	2	Riverbend	Portage	Graden
3	1	237	3	2	Riverbend	Graden	Portage
COMPONENT 2							
5	6	225	3	2	Winston	Portage	Parkside
6	8	270	3	2	Parkside	Winston	Assiniboine
8	6	270	3	2	Parkside	Assiniboine	Winston
6	7	175	3	2	Winston	Parkside	Assiniboine
7	6	175	3	2	Winston	Assiniboine	Parkside
6	5	225	3	2	Winston	Parkside	Portage
COMPONENT 3							
9	10	370	3	2	Cavell	Portage	Assiniboine
10	9	370	3	2	Cavell	Assiniboine	Portage
COMPONENT 4							
11	12	340	3	2	Bourkevale	Portage	Assiniboine
12	11	340	3	2	Bourkevale	Assiniboine	Portage
COMPONENT 5							
13	14	350	3	2	Riveroaks	Portage	Assiniboine
14	13	350	3	2	Riveroaks	Assiniboine	Portage
COMPONENT 6							
15	16	370	3	2	Parkview	Portage	Assiniboine
16	15	370	3	2	Parkview	Assiniboine	Portage
COMPONENT 7							
17	18	390	3	2	Roseberry	Portage	Assiniboine
18	17	390	3	2	Roseberry	Assiniboine	Portage
COMPONENT 8							
19	20	370	3	4	Colligiate	Portage	Assiniboine
20	19	370	3	4	Colligiate	Assiniboine	Portage
COMPONENT 9							
23	24	120	3	2	Library	Portage	S. End
24	23	120	3	2	Library	S. End	Portage
COMPONENT 10							
25	26	430	3	2	Douglas	Portage	S. End
26	25	430	3	2	Douglas	S. End	Portage
COMPONENT 11							
27	28	150	3	2	Albany	Portage	Deer Lodge
28	27	150	3	2	Albany	S. End	Portage

COMPONENT 12

29	31	670	3	2	Deer Lodge	Portage	Portage
31	29	670	3	2	Deer Lodge	Portage	Portage

COMPONENT 13

33	34	150	3	2	Olive	Assiniboine	Portage
34	33	150	3	2	Olive	Assiniboine	Portage

COMPONENT 14

35	37	420	3	2	Pinewood	Assiniboine	Moray
37	38	130	3	2	Moray	Pinewood	Assiniboine
38	37	130	3	2	Moray	Assiniboine	Pinewood
37	36	150	3	2	Moray	Pinewood	Portage
36	37	150	3	2	Moray	Portage	Pinewood
37	35	420	3	2	Pinewood	Moray	Lake

COMPONENT 15

39	40	280	3	2	Lake	Portage	Assiniboine
40	39	280	3	2	Lake	Assiniboine	Portage

COMPONENT 16

41	42	310	3	2	Windham	Portage	Assiniboine
42	41	310	3	2	Windham	Assiniboine	Portage

COMPONENT 17

43	44	260	3	2	Thompson	Portage	Assiniboine
44	43	260	3	2	Thompson	Assiniboine	Portage

COMPONENT 18

45	46	210	3	2	Woodbridge	Portage	Old Mill
46	47	180	3	2	Woodbridge	Old Mill	Assiniboine
47	46	180	3	2	Woodbridge	Assiniboine	Old Mill
46	48	390	3	2	Old Mill	Woodbridge	Portage
48	46	390	3	2	Old Mill	Portage	Woodbridge
46	45	210	3	2	Woodbridge	Old Mill	Portage

COMPONENT 19

50	60	150	3	2	Emo	Woodhaven	Sunnyside
60	59	110	3	2	Emo	Sunnyside	Oakdean
59	57	220	3	2	Oakdean	Emo	Portage
57	56	105	3	2	Oakdean	Oakdean Cr	Oakdean Cr
56	55	110	3	2	Oakdean	Oakdean Cr	Portage
55	56	110	3	2	Oakdean	Portage	Oakdean Cr
56	58	220	3	2	Oakdean Cr	Oakdean	Oakdean
58	56	220	3	2	Oakdean Cr	Oakdean	Oakdean
56	57	105	3	2	Oakdean	Oakdean Cr	Oakdean Cr
57	59	200	3	2	Oakdean	Oakdean Cr	Emo
59	62	360	3	2	Oakdean	Emo	Assiniboine
59	66	105	3	2	Emo	Oakdean	Harris
66	59	105	3	2	Emo	Harris	Oakdean
62	59	360	3	2	Oakdean	Assiniboine	Emo
59	60	110	3	2	Emo	Oakdean	Sunnyside
60	61	360	3	2	Sunnyside	Emo	Assiniboine
61	60	360	3	2	Sunnyside	Assiniboine	Emo
60	50	150	3	2	Emo	Sunnyside	Woodhaven

COMPONENT 20

51	53	220	3	2	Glendale	Woodhaven	Assiniboine
53	54	125	3	2	Assiniboine	Glendale	Woodhaven
53	52	370	3	2	Ashcroft	Assiniboine	E. End
52	53	370	3	2	Ashcroft	E. End	Assiniboine
54	53	125	3	2	Assiniboine	Woodhaven	Glendale
53	51	220	3	2	Glendale	Assiniboine	Woodhaven

COMPONENT 21

63	64	460	3	2	Armour	Portage	Harris
64	63	460	3	2	Armour	Harris	Portage

COMPONENT 22

69	71	212	3	2	Woodland	Summerlands	Mcbey
71	70	200	3	2	Woodland	Mcbey	Portage
70	71	200	3	2	Woodland	Portage	Mcbey
71	69	212	3	2	Woodland	Mcbey	Summerlands
69	68	760	3	2	Woodland	Summerlands	Portage
68	69	760	3	2	Woodland	Portage	Summerlands
69	72	330	3	2	Summerlands	Woodland	Mcbey
72	69	330	3	2	Summerlands	Mcbey	Woodland

COMPONENT 23

75	78	100	3	2	Meadowside	Country Club	Greenacre
78	82	55	3	2	Meadowside	Greenacre	Park West
82	81	37	3	2	Meadowside	Greenacre	Golf
81	80	290	3	2	Golf	Meadowside	Mcbey
80	79	250	3	2	Golf	Mcbey	Portage
79	80	250	3	2	Golf	Portage	Mcbey
80	81	290	3	2	Golf	Mcbey	Meadowside
81	82	37	3	2	Meadowside	Golf	Greenacre
82	83	330	3	2	Park West	Meadowside	Country Club
83	84	133	3	2	Pine Valley	Country Club	Vardon
84	85	140	3	2	Vardon	Pine Valley	E. End
85	84	140	3	2	Vardon	E. End	Pine Valley
84	87	204	3	2	Pine Valley	Vardon	Braid
87	88	133	3	2	Pine Valley	Braid	Country Club
87	86	60	3	2	Braid	Pine Valley	E. End
86	87	60	3	2	Braid	E. End	Pine Valley
88	87	133	3	2	Pine Valley	Country Club	Braid
87	84	204	3	2	Pine Valley	Braid	Vardon
84	83	133	3	2	Pine Valley	Vardon	Country Club
83	82	330	3	2	Park West	Country Club	Meadowside
82	78	55	3	2	Meadowside	Park West	Greenacre
78	77	290	3	2	Greenacre	Meadowside	Mcbey
77	76	237	3	2	Greenacre	Mcbey	Portage
76	77	239	3	2	Greenacre	Portage	Mcbey
77	78	190	3	2	Greenacre	Mcbey	Meadowside
78	75	100	3	2	Meadowside	Greenacre	Country Club

COMPONENT 24

90	97	225	3	2	Pinehurst	Country Club	Compton
97	98	60	3	2	Compton	Pinehurst	W. End
98	97	60	3	2	Compton	W. End	Pinehurst
97	96	115	3	2	Pinehurst	Compton	Browning
96	97	115	3	2	Pinehurst	Browning	Compton
96	95	337	3	2	Pinehurst	Browning	Hagen
95	96	337	3	2	Pinehurst	Hagen	Browning
95	91	360	3	2	Hagen	Pinehurst	Country Club
91	95	360	3	2	Hagen	Country Club	Pinehurst
95	92	150	3	2	Pinehurst	Hagen	Country Club
92	95	150	3	2	Pinehurst	Country Club	Hagen
97	90	225	3	2	Pinehurst	Compton	Country Club

COMPONENT 25

93	94	260	3	2	Daer	Portage	Mcbey
94	93	260	3	2	Daer	Mcbey	Portage

COMPONENT 26

99	100	280	3	2	Banting	Portage	Mcbey
100	99	280	3	2	Banting	Mcbey	Portage

COMPONENT 27

101	102	285	3	2	Kirkfield	Portage	Mcbey
102	103	230	3	2	Kirkfield	Mcbey	Byron
103	104	88	3	2	Kirkfield	Byron	Byron
104	106	75	3	2	Kirkfield	Byron	Wordsworth
106	107	145	3	2	Wordsworth	Kirkfield	Aldrich
107	109	85	3	2	Wordsworth	Aldrich	Wordsworth Pl

109	110	88	3	2	Wordsworth	Wordsworth Pl	Browning
110	121	280	3	2	Wordsworth	Browning	Westwood
121	152	112	3	2	Wordsworth	Westwood	Carroll
152	153	125	3	2	Caroll	Wordsworth	Thackeray
153	154	80	3	2	Caroll	Thackeray	Leacock
154	126	210	3	2	Leacock	Caroll	Westwood
126	129	210	3	2	Sandburg	Westwood	Westwood
129	126	210	3	2	Sandburg	Westwood	Westwood
126	154	310	3	2	Leacock	Westwood	Caroll
154	155	85	3	2	Caroll	Leacock	Frost
155	156	35	3	2	Caroll	Frost	Columbus
156	181	85	3	2	Columbus	Caroll	Erlandson
181	180	82	3	2	Erlandson	Columbus	Bering
180	182	260	3	2	Bering	Erlandson	Cook
182	180	260	3	2	Bering	Cook	Erlandson
180	178	125	3	2	Erlandson	Bering	Cortez
178	177	210	3	2	Erlandson	Cortez	Cortez
177	179	370	3	2	Cortez	Erlandson	Erlandson
179	177	370	3	2	Cortez	Erlandson	Erlandson
177	182	75	3	2	Cook	Erlandson	Bering
182	183	200	3	2	Cook	Bering	Columbus
183	182	200	3	2	Cook	Columbus	Bering
182	177	75	3	2	Cook	Bering	Erlandson
177	176	80	3	2	Erlandson	Portage	Rouge
176	177	80	3	2	Erlandson	Rouge	Cortez
177	178	210	3	2	Erlandson	Cortez	Cortez
178	180	125	3	2	Erlandson	Cortez	Bering
180	181	82	3	2	Erlandson	Bering	Columbus
181	183	375	3	2	Columbus	Erlandson	Cook
183	184	80	3	2	Columbus	Cook	Rouge
184	183	80	3	2	Columbus	Rouge	Cook
183	181	375	3	2	Columbus	Cook	Erlandson
181	156	85	3	2	Columbus	Erlandson	Caroll
156	157	205	3	2	Caroll	Columbus	Kilmer
157	158	110	3	2	Caroll	Kilmer	Assiniboine
158	157	110	3	2	Caroll	Assiniboine	Kilmer
157	134	215	3	2	Kilmer	Caroll	Westwood
134	157	215	3	2	Kilmer	Westwood	Caroll
157	185	180	3	2	Kilmer	Caroll	Long Fellow
185	187	205	3	2	Long Fellow	Kilmer	Assiniboine
187	185	205	3	2	Long Fellow	Assiniboine	Kilmer
185	186	115	3	2	Long Fellow	Kilmer	Assiniboine
186	185	115	3	2	Long Fellow	Assiniboine	Kilmer
185	157	180	3	2	Kilmer	Long Fellow	Caroll
157	156	205	3	2	Caroll	Kilmer	Columbus
156	155	35	3	2	Caroll	Columbus	Frost
155	128	210	3	2	Frost	Caroll	Westwood
128	155	210	3	2	Frost	Westwood	Caroll
155	154	85	3	2	Caroll	Frost	Leacock
154	153	80	3	2	Caroll	Leacock	Thackeray
153	124	180	3	2	Thackeray	Caroll	Westwood
124	122	330	3	2	Carlyle	Westwood	Westwood
122	123	60	3	2	Carlyle Cr	Westwood	W End
123	122	60	3	2	Carlyle Cr	W End	Westwood
122	124	330	3	2	Carlyle	Westwood	Westwood
124	153	180	3	2	Thackeray	Westwood	Caroll
153	152	125	3	2	Caroll	Thackeray	Wordsworth
152	175	425	3	2	Wordsworth	Caroll	Browning
175	174	80	3	2	Browning	Wordsworth	Rouge
174	127	130	3	2	Browning	Rouge	Reynolds
127	174	130	3	2	Browning	Reynolds	Rouge
174	175	80	3	2	Browning	Rouge	Wordsworth
175	173	250	3	2	Wordsworth	Browning	Joliett
173	151	335	2	3	Joliett	Wordsworth	Sansome
151	173	335	3	2	Joliett	Sansome	Wordsworth
173	172	85	3	2	Joliett	Wordsworth	Rouge
172	173	85	3	2	Joliett	Rouge	Wordsworth
173	175	250	3	2	Wordsworth	Joliett	Browning
175	152	425	3	2	Wordsworth	Browning	Caroll
152	121	112	3	2	Wordsworth	Caroll	Westwood
121	110	280	3	2	Wordsworth	Westwood	Browning
110	109	88	3	2	Wordsworth	Browning	Wordsworth Pl
109	108	80	3	2	Wordsworth Pl	Wordsworth	W. End
108	109	80	3	2	Wordsworth Pl	W. End	Wordsworth
109	107	85	3	2	Wordsworth	Wordsworth Pl	Aldrich
107	116	106	3	2	Aldrich	Wordsworth	Shelley

116	107	160	3	2	Aldrich	Shelley	Wordsworth
107	106	145	3	2	Wordsworth	Aldrich	Kirkfield
106	113	90	3	2	Wordsworth	Kirkfield	Shelley
113	115	50	3	2	Shelley	Wordsworth	Poole
115	116	38	3	2	Shelley	Poole	Aldrich
116	118	55	3	2	Shelley	Aldrich	Drummond
118	119	125	3	2	Shelley	Drummond	Browning
119	118	125	3	2	Shelley	Browning	Drummond
118	117	40	3	2	Drummond	Shelley	W End
117	118	40	3	2	Drummond	W End	Shelley
118	116	55	3	2	Shelley	Drummond	Aldrich
116	115	38	3	2	Shelley	Aldrich	Poole
115	114	40	3	2	Poole	Shelley	W End
114	115	40	3	2	Poole	W. End	Shelley
115	113	50	3	2	Shelley	Poole	Wordsworth
113	112	395	3	2	Shelley	Wordsworth	Mcbey
112	111	90	3	2	Shelley	Mcbey	Portage
111	112	305	3	2	Shelley	Portage	Mcbey
112	113	395	3	2	Shelley	Mcbey	Wordsworth
113	106	90	3	2	Wordsworth	Shelley	Kirkfield
106	104	75	3	2	Kirkfield	Wordsworth	Byron
104	103	88	3	2	Kirkfield	Byron	Byron
103	105	320	3	2	Byron	Kirkfield	Kirkfield
105	103	320	3	2	Byron	Kirkfield	Kirkfield
103	102	230	3	2	Kirkfield	Byron	Mcbey
102	101	285	3	2	Kirkfield	Mcbey	Portage

COMPONENT 28

127	205	200	3	2	Reynolds	Browning	Whitegates
205	203	85	3	2	Whitegates	Reynolds	Belcourt
203	202	85	3	2	Whitegates	Belcourt	Belcourt
202	204	380	3	2	Belcourt	Whitegates	Whitegates
204	202	380	3	2	Belcourt	Whitegate	Whitegate
202	201	85	3	2	Whitegates	Belcourt	Sansome
201	202	85	3	2	Whitegates	Sansome	Belcourt
202	203	85	3	2	Whitegates	Belcourt	Belcourt
203	205	85	3	2	Whitegates	Belcourt	Reynolds
205	206	85	3	2	Whitegates	Reynolds	Browning
206	226	175	3	2	Browning	Whitegates	Raquette
226	224	95	3	2	Raquette	Browning	Haultain
224	223	160	3	2	Raquette	Haultain	Haultain
223	225	330	3	2	Haultain	Raquette	Raquette
225	223	330	3	2	Haultain	Raquette	Raquette
223	222	85	3	2	Raquette	Haultain	Sansome
222	223	85	3	2	Raquette	Sansome	Haultain
223	224	160	3	2	Raquette	Haultain	Haultain
224	226	95	3	2	Raquette	Haultain	Browning
226	227	95	3	2	Raquette	Browning	Trigwell
227	228	55	3	2	Raquette	Trigwell	Addsion Cr
228	229	30	3	2	Raquette	Addsion Cr	Trigwell
229	230	260	3	2	Raquette	Trigwell	Holt
230	250	360	3	2	Holt	Raquette	Bedson
250	251	315	3	2	Coleridge	Bedson	Galsworthy
251	252	110	3	2	Galsworthy	Coleridge	E End
252	251	110	3	2	Galsworthy	E End	Coleridge
251	253	85	3	2	Coleridge	Galsworthy	Masefield
253	254	80	3	2	Masefield	Coleridge	E End
254	253	80	3	2	Masefield	E End	Coleridge
253	255	230	3	2	Coleridge	Masefield	Assinibione
255	253	236	3	2	Coleridge	Assinibione	Masefield
253	251	85	3	2	Coleridge	Masefield	Galsworthy
251	250	315	3	2	Coleridge	Galsworthy	Bedson
250	230	360	3	2	Holt	Bedson	Raquette
230	231	110	3	2	Raquette	Holt	Assinibione
231	230	110	3	2	Raquette	Assinibione	Holt
230	229	260	3	2	Raquette	Holt	Trigwell
229	228	30	3	2	Raquette	Trigwell	Addsion Cr
228	248	245	3	2	Addsion Cr	Raquette	Addsion
248	223	320	3	2	Addsion Cr	Addsion	Raquette
223	248	320	3	2	Addison Cr	Raquette	Addison
248	228	245	3	2	Addsion Cr	Addsion	Raquette
228	227	55	3	2	Raquette	Addsion Cr	Trigwell
227	229	250	3	2	Trigwell	Raquette	Raquette
229	227	250	3	2	Trigwell	Raquette	Raquette
227	226	95	3	2	Raquette	Trigwell	Browning

226	206	75	3	2	Browning	Raquette	Whitegates
206	207	410	3	2	Whitegates	Browning	Barron
207	216	160	3	2	Barron	Whitegates	Maureen
216	207	160	3	2	Barron	Maureen	Whitegates
207	209	112	3	2	Whitegates	Barron	West
209	210	75	3	2	Whitegates	West	Cora
210	213	235	3	2	Whitegates	Cora	Maureen
213	212	85	3	2	Maureen	Whitegates	Assiniboine
212	213	85	3	2	Maureen	Assiniboine	Whitegates
213	214	78	3	2	Maureen	Whitrgates	Cora
214	215	80	3	2	Maureen	Cora	West
215	216	80	3	2	Maureen	West	Barron
216	217	395	3	2	Barron	Maureen	Assiniboine
217	216	395	3	2	Barron	Assiniboine	Maureen
216	215	80	3	2	Maureen	Barron	West
215	214	80	3	2	Maureen	West	Cora
214	210	150	3	2	Cora	Maureen	Whitegates
210	214	150	3	2	Cora	Whitegates	Maureen
214	213	78	3	2	Maureen	Cora	Whitegates
213	210	135	3	2	Whitegates	Maureen	Cora
210	209	75	3	2	Whitegates	Cora	West
209	215	140	3	2	West	Whitegates	Maureen
215	209	140	3	2	West	Maureen	Whitegates
209	208	90	3	2	West	Whitegates	Rouge
208	209	90	3	2	West	Rouge	Whitegates
209	207	112	3	2	Whitegates	West	Barron
207	206	410	3	2	Whitegates	Barron	Browning
206	205	85	3	2	Whitegates	Browning	Reynolds
205	127	200	3	2	Reynolds	Whitegates	Browning
127	206	112	3	2	Browning	Reynolds	Whitegates
206	127	112	3	2	Browning	Whitegates	Reynolds

COMPONENT 29

132	133	210	3	2	Haliburton	Westwood	Westwood
133	130	340	3	2	Robert Serv.	Westwood	Westwood
130	133	340	3	2	Robert Serv.	Westwood	Westwood
133	132	210	3	2	Haliburton	Westwood	Westwood

COMPONENT 30

134	135	70	3	2	Connor	Westwood	E End
135	134	70	3	2	Connor	E End	Westwood

COMPONENT 31

137	138	50	3	2	Westboine	Assiniboine	S End
138	137	50	3	2	Westboine	S End	Assiniboine

COMPONENT 32

140	161	250	3	2	Byrd	Westwood	Davis
161	162	250	3	2	Davis	Byrd	Rouge
162	161	250	3	2	Davis	Rouge	Byrd
161	160	160	3	2	Byrd	Davis	Rouge
160	161	160	3	2	Byrd	Rouge	Davis
161	140	250	3	2	Byrd	Davis	Westwood

COMPONENT 33

141	143	410	3	2	Admunsen	Westwood	Westwood
143	141	410	3	2	Admunsen	Westwood	Westwood

COMPONENT 34

144	146	410	3	2	Henday	Westwood	Westwood
146	144	410	3	2	Henday	Westwood	Westwood

COMPONENT 35

147	149	410	3	2	Magellan	Westwood	Westwood
149	147	410	3	2	Magellan	Westwood	Westwood

COMPONENT 36

164	165	420	3	2	Ericson	Rouge	Rouge

165	164	420	3	2	Ericson	Rouge	Rouge

COMPONENT 37

166	168	420	3	2	Galinee	Rouge	Rouge
168	166	420	3	2	Galinee	Rouge	Rouge

COMPONENT 38

169	171	250	3	2	Peary	Rouge	Sansome
171	169	250	3	2	Peary	Sansome	Rouge

COMPONENT 39

188	189	180	3	2	Scalena	Assiniboine	Rouge
189	188	180	3	2	Scalena	Rouge	Assiniboine

COMPONENT 40

190	191	125	3	2	Sumach	Portage	Canoe
191	192	78	3	2	Sumach	Canoe	Canoe
192	194	125	3	2	Sumach	Canoe	Manitou
194	195	75	3	2	Sumach	Manitou	Manitou
195	197	80	3	2	Sumach	Manitou	Allard
197	195	80	3	2	Sumach	Allard	Manitou
195	194	75	3	2	Sumach	Manitou	Manitou
194	196	380	3	2	Manitou	Sumach	Sumach
196	194	380	3	2	Manitou	Sumach	Sumach
194	192	125	3	2	Sumach	Manitou	Canoe
192	191	78	3	2	Sumach	Canoe	Canoe
191	193	370	3	2	Canoe	Sumach	Sumach
193	191	370	3	2	Canoe	Sumach	Sumach
191	190	125	3	2	Sumach	Portage	Canoe

COMPONENT 41

198	200	110	3	2	Pontiac	Allard	Allard
200	198	110	3	2	Pontiac	Allard	Allard

COMPONENT 42

243	247	640	3	2	Dicken's	Bedson	Bedson
247	248	80	3	2	Addsion	Bedson	Addsion Cr
248	247	80	3	2	Addsion	Addsion Cr	Bedson
247	243	640	3	2	Dicken's	Bedson	Bedson

COMPONENT 43

256	242	240	3	2	Sansome	W End	Bedson
242	256	240	3	2	Sansome	Bedson	W End

COMPONENT 44

218	219	480	3	2	Seaton	Portage	Allard
219	218	480	3	2	Seaton	Allard	Portage

COMPONENT 45

220	221	470	3	2	Raquette	Portage	Allard
221	220	470	3	2	Raquette	Allard	Portage

COMPONENT 46

232	233	450	3	2	Best	Portage	Allard
233	232	450	3	2	Best	Allard	Portage

COMPONENT 47

235	236	355	3	2	Twain	Bedson	Allard
236	241	495	3	2	Twain	Allard	Bedson
241	236	495	3	2	Twain	Bedson	Allard
236	237	200	3	2	Allard	Twain	Bedson
237	236	200	3	2	Allard	Bedson	Twain
236	235	355	3	2	Twain	Allard	Bedson

COMPONENT 48

238	240	340	3	2	Keats	Bedson	Bedson
240	238	340	3	2	Keats	Bedson	Bedson

COMPONENT 49

244	246	320	3	2	Shakespeare	Bedson	Bedson
246	244	320	3	2	Shakespeare	Bedson	Bedson

COMPONENT 50

249	257	260	3	2	Assinibione Gr	Bedson	W End
257	249	260	3	2	Assinibione Gr	W End	Bedson

COMPONENT 51

258	259	260	3	2	Buchanan	Allard	Portage
259	258	760	3	2	Buchanan	Allard	Portage

COMPONENT 52

260	261	238	3	2	Stewart	Portage	Allard
261	260	238	3	2	Steward	Allard	Portage

COMPONENT 53

264	265	100	3	2	Gagnon	Portage	Allard
265	268	75	3	2	Gagnon	Allard	Dennis
268	266	60	3	2	Dennis	Gagnon	W End
266	268	60	3	2	Dennis	W End	Gagnon
268	271	85	3	2	Gagnon	Dennis	Brendalee
271	267	60	3	2	Brendalee	Gagnon	W End
267	271	60	3	2	Brendalee	W End	Gagnon
271	269	100	3	2	Lepine	Gagnon	St Charles
269	270	50	3	2	Lepine	St Charles	E End
270	269	50	3	2	Lepine	E End	St Charles
269	271	100	3	2	Lepine	St Charles	Gagnon
271	273	80	3	2	Gagnon	Lepine	Suzanne
273	272	60	3	2	Suzanne	Gagnon	W End
272	273	60	3	2	Suzanne	W End	Gagnon
273	276	70	3	2	Gagnon	Suzanne	Sansome
276	274	100	3	2	Sansome	Gagnon	St Charles
274	275	50	3	2	Sansome	St Charles	E End
275	274	50	3	2	Sansome	E End	St Charles
274	276	100	3	2	Sansome	St Charles	Gagnon
276	278	70	3	2	Gagnon	Sansome	Jeannette
278	277	70	3	2	Jeanette	Gagnon	W End
277	278	70	3	2	Jeanette	W End	Gagnon
278	281	85	3	2	Gagnon	Jeanette	Augier
281	278	85	3	2	Gaganon	Augier	Jeanette
278	276	70	3	2	Gagnon	Jeanette	Sansome
276	273	70	3	2	Gagnon	Sansome	Suzanne
273	271	80	3	2	Gagnon	Suzanne	Lepine
271	268	85	3	2	Gagnon	Brendalee	Dennis
268	265	75	3	2	Gagnon	Dennis	Allard
265	264	100	3	2	Gagnon	Allard	Portage

COMPONENT 54

279	280	100	3	2	Augier	E End	St Charles
280	279	100	3	2	Augier	St Charles	E End

Appendix H
District #2: Vehicle Route For Priority III Street

FN	TN	LTH	P	#LN	C/D*	BLOCK NAME	FROM	TO	
1	3	237	3	2	C	Riverbend	Portage	Garden	
3	2	130	3	2	C	Graden	Riverbend	Portage	
2	3	130	3	2	C	Garden	Portage	Riverbend	
3	4	243	3	2	C	Riverbend	Graden	Portage	
4	5	188	1	8	D	Portage	Riverbend	Winston	
5	9	80	1	8	D	Portage	Winston	Cavell	
9	10	370	3	2	C	Cavell	Portage	Assiniboine	
10	9	370	3	2	C	Cavell	Assiniboine	Portage	
9	11	78	1	8	D	Portage	Vavell	Brourkevale	
11	13	80	1	8	D	Portage	Bourkevale	Riveroaks	
13	14	350	3	2	C	Riveroaks	Portage	Assiniboine	
14	13	350	3	2	C	Riveroaks	Assiniboine	Portage	
13	15	90	1	8	D	Portage	Riveroaks	Parkview	
15	16	370	3	2	C	Parkview	Portage	Assiniboine	
16	15	370	3	2	C	Parkview	Assiniboine	Portage	
15	17	88	1	8	D	Portage	Parkview	Roseberry	
17	18	390	3	2	C	Roseberry	Portage	Assiniboine	
18	17	390	3	2	C	Roseberry	Assiniboine	Portage	
17	19	88	1	8	D	Portage	Roseberry	Colligiate	
19	20	370	3	4	C	Colligiate	Portage	Assiniboine	
20	19	370	3	4	C	Colligiate	Assiniboine	Portage	
19	21	90	1	8	D	Portage	Coligiate	Ferry	
21	23	212	1	8	D	Portage	Ferry	Library	
23	24	120	3	2	C	Library	Portage	S. End	
24	23	120	3	2	C	Library	S. End	Portage	
23	25	95	1	8	D	Portage	Library	Douglas	
25	26	430	3	2	C	Douglas	Portage	S. End	
26	25	430	3	2	C	Douglas	S. End	Portage	
25	27	175	1	3	D	Portage	Douglas	Albany	
27	28	150	3	2	C	Albany	Portage	Deer Lodge	
28	27	150	3	2	C	Albany	S. End	Portage	
27	29	112	1	8	D	Portage	Albany	Deer lodge	
29	31	670	3	2	C	Deer Lodge	Portage	Portage	(30=31)
30	32	1740	1	8	D	Portage	Deer Lodge	Assiniboine	
32	33	230	1	8	D	Portage	Assiniboine	Olive	
33	34	150	3	2	C	Olive	Assiniboine	Portage	
34	35	88	2	2	D	Assiniboine	Olive	Pinewood	
35	37	420	3	2	C	Pinewood	Assiniboine	Moray	
37	38	130	3	2	C	Moray	Pinewood	Assiniboine	
38	37	130	3	2	C	Moray	Assiniboine	Pinewood	
37	36	150	3	2	C	Moray	Pinewood	Portage	
36	39	125	1	8	D	Portage	Moray	Lake	
39	40	280	3	2	C	Lake	Portage	Assiniboine	
40	39	280	3	2	C	Lake	Assiniboine	Portage	
39	41	100	1	8	D	Portage	Lake	Windham	
41	42	310	3	2	C	Windham	Portage	Assiniboine	
42	41	310	3	2	C	Windham	Assiniboine	Portage	
41	43	87	1	8	D	Portage	Windham	Thompson	
43	44	260	3	2	C	Thompson	Portage	Assiniboine	
44	43	260	3	2	C	Thompson	Assiniboine	Portage	
43	45	85	1	8	D	Portage	Thompson	Woodbridge	
45	46	210	3	2	C	Woodbridge	Portage	Old Mill	
46	47	180	3	2	C	Woodbridge	Old Mill	Assiniboine	
47	46	180	3	2	C	Woodbridge	Assiniboine	Old Mill	
46	48	390	3	2	C	Old Mill	Woodbridge	Portage	
48	49	175	1	8	D	Portage	Old Mill	Woodhaven	
49	50	340	2	2	D	Woodhaven	Portage	Emo	
50	60	150	3	2	C	Emo	Woodhaven	Sunnyside	
60	59	110	3	2	C	Emo	Sunnyside	Oakdean	
59	57	220	3	2	C	Oakdean	Emo	Portage	
57	56	105	3	2	C	Oakdean	Oakdean Cr	Oakdean Cr	
56	55	110	3	2	C	Oakdean	Oakdean Cr	Portage	
55	63	100	1	8	D	Portage	Oakdean	Harris	
63	64	460	3	2	C	Armour	Portage	Harris	

64	63	460	3	2	C	Armour	Harris	Portage
63	68	200	1	8	D	Portage	Harris	Woodland
68	70	90	1	8	D	Portage	Woodland	Woodland
70	73	125	1	8	D	Portage	Woodland	Country Club
73	76	115	1	8	D	Portage	Country Club	Greenacre
76	77	239	3	2	C	Greenacre	Portage	Mcbey
77	78	190	3	2	C	Greenacre	Mcbey	Meadowside
78	75	100	3	2	C	Meadowside	Greenacre	Country Club
75	78	100	3	2	C	Meadowside	Country Club	Greenacre
78	82	55	3	2	C	Meadowside	Greenacre	Park West
82	81	37	3	2	C	Meadowside	Greenacre	Golf
81	80	290	3	2	C	Golf	Meadowside	Mcbey
80	79	250	3	2	C	Golf	Mcbey	Portage
79	93	100	1	8	D	Portage	Golf	Daer
93	99	115	1	8	D	Portage	Daer	Banting
99	101	120	1	8	D	Portage	Banting	Kirkfield
101	102	285	3	2	C	Kirkfield	Portage	Mcbey
102	103	230	3	2	C	Kirkfield	Mcbey	Byron
103	104	88	3	2	C	Kirkfield	Byron	Byron
104	106	75	3	2	C	Kirkfield	Byron	Wordsworth
106	107	145	3	2	C	Wordsworth	Kirkfield	Aldrich
107	109	85	3	2	C	Wordsworth	Aldrich	Wordsworth Pl
109	110	88	3	2	C	Wordsworth	Wordsworth Pl	Browning
110	96	85	2	2	D	Browning	Wordsworth	Pinehurst
96	97	115	3	2	C	Pinehurst	Browning	Compton
96	95	337	3	2	C	Pinehurst	Browning	Hagen
95	96	337	3	2	C	Pinehurst	Hagen	Browning
95	91	360	3	2	C	Hagen	Pinehurst	Country Club
91	95	360	3	2	C	Hagen	Country Club	Pinehurst
95	92	150	3	2	C	Pinehurst	Hagen	Country Club
92	95	150	3	2	C	Pinehurst	Country Club	Hagen
97	90	225	3	2	C	Pinehurst	Compton	Country Club
90	97	225	3	2	C	Pinehurst	Country Club	Compton
97	98	60	3	2	C	Compton	Pinehurst	W. End
98	97	60	3	2	C	Compton	W. End	Pinehurst
97	96	115	3	2	C	Pinehurst	Compton	Browning
96	110	85	2	2	D	Browning	Pinehurst	Wordsworth
110	121	280	3	2	C	Wordsworth	Browning	Westwood
121	152	112	3	2	C	Wordsworth	Westwood	Carroll
152	153	125	3	2	C	Caroll	Wordsworth	Thackeray
153	154	80	3	2	C	Caroll	Thackeray	Leacock
154	126	210	3	2	C	Leacock	Caroll	Westwood
126	129	210	3	2	C	Sandburg	Westwood	Westwood
129	126	210	3	2	C	Sandburg	Westwood	Westwood
126	154	310	3	2	C	Leacock	Westwood	Caroll
154	155	85	3	2	C	Caroll	Leacock	Frost
155	156	35	3	2	C	Caroll	Frost	Columbus
156	181	85	3	2	C	Columbus	Caroll	Erlandson
181	180	82	3	2	C	Erlandson	Columbus	Bering
180	182	260	3	2	C	Bering	Erlandson	Cook
182	180	260	3	2	C	Bering	Cook	Erlandson
180	178	125	3	2	C	Erlandson	Bering	Cortez
178	177	210	3	2	C	Erlandson	Cortez	Cortez
177	179	370	3	2	C	Cortez	Erlandson	Erlandson
179	177	370	3	2	C	Cortez	Erlandson	Erlandson
177	182	75	3	2	C	Cook	Erlandson	Bering
182	183	200	3	2	C	Cook	Bering	Columbus
183	182	200	3	2	C	Cook	Columbus	Bering
182	177	75	3	2	C	Cook	Bering	Erlandson
177	176	80	3	2	C	Erlandson	Portage	Rouge
176	177	80	3	2	C	Erlandson	Rouge	Cortez
177	178	210	3	2	C	Erlandson	Cortez	Cortez
178	180	125	3	2	C	Erlandson	Cortez	Bering
180	181	82	3	2	C	Erlandson	Bering	Columbus
181	183	375	3	2	C	Columbus	Erlandson	Cook
183	184	80	3	2	C	Columbus	Cook	Rouge
184	189	85	2	2	D	Rouge	Columbus	Scalena
189	188	180	3	2	C	Scalena	Rouge	Assiniboine
188	189	180	3	2	C	Scalena	Assiniboine	Rouge
189	184	85	2	2	D	Rouge	Scalena	Columbus
184	183	80	3	2	C	Columbus	Rouge	Cook
183	181	375	3	2	C	Columbus	Cook	Erlandson
181	156	85	3	2	C	Columbus	Erlandson	Caroll
156	157	205	3	2	C	Caroll	Columbus	Kilmer
157	158	110	3	2	C	Caroll	Kilmer	Assiniboine
158	157	110	3	2	C	Caroll	Assiniboine	Kilmer

157	134	215	3	2	C	Kilmer	Caroll	Westwood	
134	136	85	2	2	D	Westwood	Kilmer	Assiniboine	
136	137	70	2	3	D	Assiniboine	Westwood	Westboine	
137	138	50	3	2	C	Westboine	Assiniboine	S End	
138	137	50	3	2	C	Westboine	S End	Assiniboine	
137	136	70	2	3	D	Assiniboine	Westboine	Westwood	
136	134	85	2	2	D	Westwood	Assiniboine	Kilmer	
134	135	70	3	2	C	Connor	Westwood	E End	
135	134	70	3	2	C	Connor	E End	Westwood	
134	157	215	3	2	C	Kilmer	Westwood	Caroll	
157	185	180	3	2	C	Kilmer	Caroll	Long Fellow	
185	187	205	3	2	C	Long Fellow	Kilmer	Assiniboine	
187	185	205	3	2	C	Long Fellow	Assiniboine	Kilmer	
185	186	115	3	2	C	Long Fellow	Kilmer	Assiniboine	
186	185	115	3	2	C	Long Fellow	Assiniboine	Kilmer	
185	157	180	3	2	C	Kilmer	Long Fellow	Caroll	
157	156	205	3	2	C	Caroll	Kilmer	Columbus	
156	155	35	3	2	C	Caroll	Columbus	Frost	
155	128	210	3	2	C	Frost	Caroll	Westwood	
128	130	85	2	2	D	Westwood	Sandburg	Haliburton	(130=132)
130	133	340	3	2	C	Robert Serv.	Westwood	Westwood	
133	132	210	3	2	C	Haliburton	Westwood	Westwood	
132	133	210	3	2	C	Haliburton	Westwood	Westwood	
133	130	340	3	2	C	Robert Serv.	Westwood	Westwood	
130	128	85	2	2	D	Westwood	Haliburton	Sandburg	
128	155	210	3	2	C	Frost	Westwood	Caroll	
155	154	85	3	2	C	Caroll	Frost	Leacock	
154	153	80	3	2	C	Caroll	Leacock	Thackeray	
153	124	180	3	2	C	Thackeray	Caroll	Westwood	
124	122	330	3	2	C	Carlyle	Westwood	Westwood	
122	123	60	3	2	C	Carlyle Cr	Westwood	W End	
123	122	60	3	2	C	Carlyle Cr	W End	Westwood	
122	124	330	3	2	C	Carlyle	Westwood	Westwood	
124	153	180	3	2	C	Thackeray	Westwood	Caroll	
153	152	125	3	2	C	Caroll	Thackeray	Wordsworth	
152	175	425	3	2	C	Wordsworth	Caroll	Browning	
175	174	80	3	2	C	Browning	Wordsworth	Rouge	
174	127	130	3	2	C	Browning	Rouge	Reynolds	
127	205	200	3	2	C	Reynolds	Browning	Whitegates	
205	203	85	3	2	C	Whitegates	Reynolds	Belcourt	
203	202	85	3	2	C	Whitegates	Belcourt	Belcourt	
202	204	380	3	2	C	Belcourt	Whitegates	Whitegates	
204	202	380	3	2	C	Belcourt	Whitegate	Whitegate	
202	201	85	3	2	C	Whitegates	Belcourt	Sansome	
201	202	85	3	2	C	Whitegates	Sansome	Belcourt	
202	203	85	3	2	C	Whitegates	Belcourt	Belcourt	
203	205	85	3	2	C	Whitegates	Belcourt	Reynolds	
205	206	85	3	2	C	Whitegates	Reynolds	Browning	
206	226	175	3	2	C	Browning	Whitegates	Raquette	
226	224	95	3	2	C	Raquette	Browning	Haultain	
224	223	160	3	2	C	Raquette	Haultain	Haultain	
223	225	330	3	2	C	Haultain	Raquette	Raquette	
225	223	330	3	2	C	Haultain	Raquette	Raquette	
223	222	85	3	2	C	Raquette	Haultain	Sansome	
222	223	85	3	2	C	Raquette	Sansome	Haultain	
223	224	160	3	2	C	Raquette	Haultain	Haultain	
224	226	95	3	2	C	Raquette	Haultain	Browning	
226	227	95	3	2	C	Raquette	Browning	Trigwell	
227	228	55	3	2	C	Raquette	Trigwell	Addsion Cr	
228	229	30	3	2	C	Raquette	Addsion Cr	Trigwell	
229	230	260	3	2	C	Raquette	Trigwell	Holt	
230	231	110	3	2	C	Raquette	Holt	Assinibione	
231	230	110	3	2	C	Raquette	Assinibione	Holt	
230	250	360	3	2	C	Holt	Raquette	Bedson	
250	251	315	3	2	C	Coleridge	Bedson	Galsworthy	
251	252	110	3	2	C	Galsworthy	Coleridge	E End	
252	251	110	3	2	C	Galsworthy	E End	Coleridge	
251	253	85	3	2	C	Coleridge	Galsworthy	Masefield	
253	254	80	3	2	C	Masefield	Coleridge	E End	
254	253	80	3	2	C	Masefield	E End	Coleridge	
253	255	230	3	2	C	Coleridge	Masefield	Assinibione	
255	253	236	3	2	C	Coleridge	Assinibione	Masefield	
253	251	85	3	2	C	Coleridge	Masefield	Galsworthy	
251	250	315	3	2	C	Coleridge	Galsworthy	Bedson	
250	249	80	2	2	D	Bedson	Coleridge	Assiniboine Gr	
249	257	260	3	2	C	Assinibione Gr	Bedson	W End	

257	249	260	3	2	C	Assinibione Gr	W End	Bedson
249	250	80	2	2	D	Bedson	Assiniboine Gr	Coleridge
250	230	360	3	2	C	Holt	Bedson	Raquette
230	229	260	3	2	C	Raquette	Holt	Trigwell
229	228	30	3	2	C	Raquette	Trigwell	Addsion Cr
228	248	245	3	2	C	Addsion Cr	Raquette	Addsion
248	247	80	3	2	C	Addsion	Addsion Cr	Bedson
247	243	640	3	2	C	Dicken's	Bedson	Bedson
243	244	85	2	2	D	Bedson	Dicken's	Shakespeare
244	246	320	3	2	C	Shakespeare	Bedson	Bedson
246	244	320	3	2	C	Shakespeare	Bedson	Bedson
244	243	85	2	2	D	Bedson	Shakespeare	Dicken's
243	247	640	3	2	C	Dicken's	Bedson	Bedson
247	248	80	3	2	C	Addsion	Bedson	Addsion Cr
248	223	320	3	2	C	Addsion Cr	Addsion	Raquette
223	248	320	3	2	C	Addison Cr	Raquette	Addison
248	228	245	3	2	C	Addsion Cr	Addsion	Raquette
228	227	55	3	2	C	Raquette	Addsion Cr	Trigwell
227	229	250	3	2	C	Trigwell	Raquette	Raquette
229	227	250	3	2	C	Trigwell	Raquette	Raquette
227	226	95	3	2	C	Raquette	Trigwell	Browning
226	206	75	3	2	C	Browning	Raquette	Whitegates
206	207	410	3	2	C	Whitegates	Browning	Barron
207	216	160	3	2	C	Barron	Whitegates	Maureen
216	207	160	3	2	C	Barron	Maureen	Whitegates
207	209	112	3	2	C	Whitegates	Barron	West
209	210	75	3	2	C	Whitegates	West	Cora
210	213	235	3	2	C	Whitegates	Cora	Maureen
213	212	85	3	2	C	Maureen	Whitegates	Assiniboine
212	213	85	3	2	C	Maureen	Assiniboine	Whitegates
213	214	78	3	2	C	Maureen	Whitrgates	Cora
214	215	80	3	2	C	Maureen	Cora	West
215	216	80	3	2	C	Maureen	West	Barron
216	217	395	3	2	C	Barron	Maureen	Assiniboine
217	216	395	3	2	C	Barron	Assiniboine	Maureen
216	215	80	3	2	C	Maureen	Barron	West
215	214	80	3	2	C	Maureen	West	Cora
214	210	150	3	2	C	Cora	Maureen	Whitegates
210	214	150	3	2	C	Cora	Whitegates	Maureen
214	213	78	3	2	C	Maureen	Cora	Whitegates
213	210	135	3	2	C	Whitegates	Maureen	Cora
210	209	75	3	2	C	Whitegates	Cora	West
209	215	140	3	2	C	West	Whitegates	Maureen
215	209	140	3	2	C	West	Maureen	Whitegates
209	208	90	3	2	C	West	Whitegates	Rouge
208	209	90	3	2	C	West	Rouge	Whitegates
209	207	112	3	2	C	Whitegates	West	Barron
207	206	410	3	2	C	Whitegates	Barron	Browning
206	205	85	3	2	C	Whitegates	Browning	Reynolds
205	127	200	3	2	C	Reynolds	Whitegates	Browning
127	206	112	3	2	C	Browning	Reynolds	Whitegates
206	127	112	3	2	C	Browning	Whitegates	Reynolds
127	174	130	3	2	C	Browning	Reynolds	Rouge
174	175	80	3	2	C	Browning	Rouge	Wordsworth
175	173	250	3	2	C	Wordsworth	Browning	Joliett
173	151	335	2	3	D	Joliett	Wordsworth	Sansome
151	173	335	3	2	C	Joliett	Sansome	Wordsworth
173	172	85	3	2	C	Joliett	Wordsworth	Rouge
172	173	85	3	2	C	Joliett	Rouge	Wordsworth
173	175	250	3	2	C	Wordsworth	Joliett	Browning
175	152	425	3	2	C	Wordsworth	Browning	Caroll
152	121	112	3	2	C	Wordsworth	Caroll	Westwood
121	110	280	3	2	C	Wordsworth	Westwood	Browning
110	109	88	3	2	C	Wordsworth	Browning	Wordsworth Pl
109	108	80	3	2	C	Wordsworth Pl	Wordsworth	W. End
108	109	80	3	2	C	Wordsworth Pl	W. End	Wordsworth
109	107	85	3	2	C	Wordsworth	Wordsworth Pl	Aldrich
107	116	106	3	2	C	Aldrich	Wordsworth	Shelley
116	107	160	3	2	C	Aldrich	Shelley	Wordsworth
107	106	145	3	2	C	Wordsworth	Aldrich	Kirkfield
106	113	90	3	2	C	Wordsworth	Kirkfield	Shelley
113	112	395	3	2	C	Shelley	Wordsworth	Mcbey
112	111	305	3	2	C	Shelley	Mcbey	Portage
111	139	130	1	8	D	Portage	Shelley	Westwood
139	159	430	1	8	D	Portage	Westwood	Rouge
159	190	240	1	8	D	Portage	Rouge	Sumach

190	191	125	3	2	C	Sumach	Portage	Canoe	
191	192	78	3	2	C	Sumach	Canoe	Canoe	
192	194	125	3	2	C	Sumach	Canoe	Manitou	
194	195	75	3	2	C	Sumach	Manitou	Manitou	
195	197	80	3	2	C	Sumach	Manitou	Allard	
197	198	45	2	2	D	Allard	Sumach	Pontiac	
198	200	110	3	2	C	Pontiac	Allard	Allard	(199=200)
199	165	110	2	2	D	Allard	Pontiac	Rouge	
165	164	420	3	2	C	Ericson	Rouge	Rouge	
163	162	85	2	2	D	Rouge	Ericson	Davis	(163=164)
162	161	250	3	2	C	Davis	Rouge	Byrd	
161	160	160	3	2	C	Byrd	Davis	Rouge	
160	161	160	3	2	C	Byrd	Rouge	Davis	
161	140	250	3	2	C	Byrd	Davis	Westwood	
140	141	90	2	2	D	Westwood	Byrd	Amundsen	
141	143	410	3	2	C	Amundsen	Westwood	Westwood	
143	144	85	2	2	D	Westwood	Amundsen	Henday	
144	146	410	3	2	C	Henday	Westwood	Westwood	
145	147	85	2	2	D	Westwood	Henday	Magellan	(145=146)
147	149	410	3	2	C	Magellan	Westwood	Westwood	
149	147	410	3	2	C	Magellan	Westwood	Westwood	
147	145	85	2	2	D	Westwood	Magellan	Henday	
146	144	410	3	2	C	Henday	Westwood	Westwood	
144	143	85	2	2	D	Westwood	Henday	Amundsen	
143	141	410	3	2	C	Amundsen	Westwood	Westwood	
141	140	90	2	2	D	Westwood	Amundsen	Byrd	
140	161	250	3	2	C	Byrd	Westwood	Davis	
161	162	250	3	2	C	Davis	Byrd	Rouge	
162	163	85	2	2	D	Rouge	Davis	Ericson	
164	165	420	3	2	C	Ericson	Rouge	Rouge	
165	166	80	2	2	D	Rouge	Ericson	Galinee	
166	168	420	3	2	C	Galinee	Rouge	Rouge	
167	169	85	2	2	D	Rouge	Galinee	Peary	(167=168)
169	171	250	3	2	C	Peary	Rouge	Sansome	
171	169	250	3	2	C	Peary	Sansome	Rouge	
168	166	420	3	2	C	Galinee	Rouge	Rouge	
166	165	80	2	2	D	Rouge	Galinee	Ericson	
165	199	110	2	2	D	Allard	Rouge	Pontiac	
200	198	110	3	2	C	Pontiac	Allard	Allard	
198	197	85	2	2	D	Allard	Pontiac	Sumach	
197	195	80	3	2	C	Sumach	Allard	Manitou	
195	194	75	3	2	C	Sumach	Manitou	Manitou	
194	196	380	3	2	C	Manitou	Sumach	Sumach	
196	194	380	3	2	C	Manitou	Sumach	Sumach	
194	192	125	3	2	C	Sumach	Manitou	Canoe	
192	191	78	3	2	C	Sumach	Canoe	Canoe	
191	193	370	3	2	C	Canoe	Sumach	Sumach	
193	191	370	3	2	C	Canoe	Sumach	Sumach	
191	190	125	3	2	C	Sumach	Portage	Canoe	
190	218	85	1	8	D	Portage	Sumach	Seaton	
218	219	480	3	2	C	Seaton	Portage	Allard	
219	218	480	3	2	C	Seaton	Allard	Portage	
218	220	85	1	8	D	Portage	Seaton	Raquette	
220	221	470	3	2	C	Raquette	Portage	Allard	
221	220	470	3	2	C	Raquette	Allard	Portage	
220	232	85	1	8	D	Portage	Raquettee	Best	
232	234	85	1	8	D	Portage	Best	Bedson	
234	258	700	1	8	D	Portage	Bedson	Buchanan	
258	259	260	3	2	C	Buchanan	Allard	Portage	
259	261	100	2	2	D	Allard	Buchanan	Stewart	
261	263	145	2	2	D	Allrad	Stewart	St Charles	
263	265	95	2	2	D	Allard	St Charles	Gagnon	
265	268	75	3	2	C	Gagnon	Allard	Dennis	
268	266	60	3	2	C	Dennis	Gagnon	W End	
266	268	60	3	2	C	Dennis	W End	Gagnon	
268	271	85	3	2	C	Gagnon	Dennis	Brendalee	
271	267	60	3	2	C	Brendalee	Gagnon	W End	
267	271	60	3	2	C	Brendalee	W End	Gagnon	
271	269	100	3	2	C	Lepine	Gagnon	St Charles	
269	270	50	3	2	C	Lepine	St Charles	E End	
270	269	50	3	2	C	Lepine	E End	St Charles	
269	271	100	3	2	C	Lepine	St Charles	Gagnon	
271	273	80	3	2	C	Gagnon	Lepine	Suzanne	
273	272	60	3	2	C	Suzanne	Gagnon	W End	
272	273	60	3	2	C	Suzanne	W End	Gagnon	
273	276	70	3	2	C	Gagnon	Suzanne	Sansome	

276	274	100	3	2	C	Sansome	Gagnon	St Charles
274	275	50	3	2	C	Sansome	St Charles	E End
275	274	50	3	2	C	Sansome	E End	St Charles
274	280	165	2	2	D	St Charles	Sansome	Augier
280	279	100	3	2	C	Augier	St Charles	E End
279	280	100	3	2	C	Augier	E End	St Charles
280	274	165	2	2	D	St Charles	Augier	Sansome
274	276	100	3	2	C	Sansome	St Charles	Gagnon
276	278	70	3	2	C	Gagnon	Sansome	Jeannette
278	277	70	3	2	C	Jeanette	Gagnon	W End
277	278	70	3	2	C	Jeanette	W End	Gagnon
278	281	85	3	2	C	Gagnon	Jeanette	Augier
281	278	85	3	2	C	Gaganon	Augier	Jeanette
278	276	70	3	2	C	Gagnon	Jeanette	Sansome
276	273	70	3	2	C	Gagnon	Sansome	Suzanne
273	271	80	3	2	C	Gagnon	Suzanne	Lepine
271	268	85	3	2	C	Gagnon	Brendalee	Dennis
268	265	75	3	2	C	Gagnon	Dennis	Allard
265	264	100	3	2	C	Gagnon	Allard	Portage
264	265	100	3	2	C	Gagnon	Portage	Allard
265	263	95	2	2	D	Allard	Gagnon	St Charles
263	261	145	2	2	D	Allard	St Charles	Stewart
261	260	238	3	2	C	Stewart	Allard	Portage
260	261	238	3	2	C	Stewart	Portage	Allard
261	259	100	2	2	D	Allard	Stewart	Buchanan
259	258	760	3	2	C	Buchanan	Allard	Portage
258	234	700	1	8	D	Portage	Buchanan	Bedson
234	232	85	1	8	D	Portage	Bedson	Best
232	233	450	3	2	C	Best	Portage	Allard
233	237	85	2	2	D	Allard	Best	Bedson
237	236	200	3	2	C	Allard	Bedson	Twain
236	235	355	3	2	C	Twain	Allard	Bedson
235	236	355	3	2	C	Twain	Bedson	Allard
236	241	495	3	2	C	Twain	Allard	Bedson
241	242	85	2	2	D	Bedson	Twain	Sansome
242	256	240	3	2	C	Sansome	Bedson	W End
256	242	240	3	2	C	Sansome	W End	Bedson
242	241	85	2	2	D	Bedson	Sansome	Twain
241	236	495	3	2	C	Twain	Bedson	Allard
236	237	200	3	2	C	Allard	Twain	Bedson
237	238	80	2	2	D	Bedson	Allard	Keats
238	240	340	3	2	C	Keats	Bedson	Bedson
240	238	340	3	2	C	Keats	Bedson	Bedson
238	237	80	2	2	D	Bedson	Keats	Allard
237	233	85	2	2	D	Allard	Bedson	Best
233	232	450	3	2	C	Best	Allard	Portage
232	220	85	1	8	D	Portage	Best	Raquette
220	218	85	1	8	D	Portage	Raquette	Seaton
218	190	85	1	8	D	Portage	Seaton	Sumach
190	159	140	1	8	D	Portage	Sumach	Rouge
159	139	430	1	8	D	Portage	Rouge	Westwood
139	111	130	1	8	D	Portage	Westwood	Shelley
111	112	305	3	2	C	Shelley	Portage	Mcbey
112	113	395	3	2	C	Shelley	Mcbey	Wordsworth
113	115	50	3	2	C	Shelley	Wordsworth	Poole
115	116	38	3	2	C	Shelley	Poole	Aldrich
116	118	55	3	2	C	Shelley	Aldrich	Drummond
118	119	125	3	2	C	Shelley	Drummond	Browning
119	118	125	3	2	C	Shelley	Browning	Drummond
118	117	40	3	2	C	Drummond	Shelley	W End
117	118	40	3	2	C	Drummond	W End	Shelley
118	116	55	3	2	C	Shelley	Drummond	Aldrich
116	115	38	3	2	C	Shelley	Aldrich	Poole
115	114	40	3	2	C	Poole	Shelley	W End
114	115	40	3	2	C	Poole	W. End	Shelley
115	113	50	3	2	C	Shelley	Poole	Wordsworth
113	106	90	3	2	C	Wordsworth	Shelley	Kirkfield
106	104	75	3	2	C	Kirkfield	Wordsworth	Byron
104	103	88	3	2	C	Kirkfield	Byron	Byron
103	105	320	3	2	C	Byron	Kirkfield	Kirkfield
105	103	320	3	2	C	Byron	Kirkfield	Kirkfield
103	102	230	3	2	C	Kirkfield	Byron	Mcbey
102	101	285	3	2	C	Kirkfield	Mcbey	Portage
101	99	120	1	8	D	Portage	Kirkfield	Banting
99	100	280	3	2	C	Banting	Portage	Mcbey
100	99	280	3	2	C	Banting	Mcbey	Portage

99	93	115	1	8	D	Portage	Banting	Daer
93	94	260	3	2	C	Daer	Portage	Mcbey
94	93	260	3	2	C	Daer	Mcbey	Portage
93	79	100	1	8	D	Portage	Daer	Golf
79	80	250	3	2	C	Golf	Portage	Mcbey
80	81	290	3	2	C	Golf	Mcbey	Meadowside
81	82	37	3	2	C	Meadowside	Golf	Greenacre
82	83	330	3	2	C	Park West	Meadowside	Country Club
83	84	133	3	2	C	Pine Valley	Country Club	Vardon
84	85	140	3	2	C	Vardon	Pine Valley	E. End
85	84	140	3	2	C	Vardon	E. End	Pine Valley
84	87	204	3	2	C	Pine Valley	Vardon	Braid
87	88	133	3	2	C	Pine Valley	Braid	Country Club
87	86	60	3	2	C	Braid	Pine Valley	E. End
86	87	60	3	2	C	Braid	E. End	Pine Valley
88	87	133	3	2	C	Pine Valley	Country Club	Braid
87	84	204	3	2	C	Pine Valley	Braid	Vardon
84	83	133	3	2	C	Pine Valley	Vardon	Country Club
83	82	330	3	2	C	Park West	Country Club	Meadowside
82	78	55	3	2	C	Meadowside	Park West	Greenacre
78	77	290	3	2	C	Greenacre	Meadowside	Mcbey
77	76	237	3	2	C	Greenacre	Mcbey	Portage
76	73	115	1	8	D	Portage	Greenacre	Country Club
73	70	125	1	8	D	Portage	Country Club	Woodland
70	71	200	3	2	C	Woodland	Portage	Mcbey
71	69	212	3	2	C	Woodland	Mcbey	Summerlands
69	68	760	3	2	C	Woodland	Summerlands	Portage
68	69	760	3	2	C	Woodland	Portage	Summerlands
69	72	330	3	2	C	Summerlands	Woodland	Mcbey
72	69	330	3	2	C	Summerlands	Mcbey	Woodland
69	71	212	3	2	C	Woodland	Summerlands	Mcbey
71	70	200	3	2	C	Woodland	Mcbey	Portage
70	68	90	1	8	D	Portage	Woodland	Woodland
68	63	200	1	8	D	Portage	Woodland	Harris
63	55	100	1	8	D	Portage	Harris	Oakdean
55	56	110	3	2	C	Oakdean	Portage	Oakdean Cr
56	58	220	3	2	C	Oakdean Cr	Oakdean	Oakdean
58	56	220	3	2	C	Oakdean Cr	Oakdean	Oakdean
56	57	105	3	2	C	Oakdean	Oakdean Cr	Oakdean Cr
57	59	200	3	2	C	Oakdean	Oakdean Cr	Emo
59	62	360	3	2	C	Oakdean	Emo	Assiniboine
59	66	105	3	2	C	Emo	Oakdean	Harris
66	59	105	3	2	C	Emo	Harris	Oakdean
62	59	360	3	2	C	Oakdean	Assiniboine	Emo
59	60	110	3	2	C	Emo	Oakdean	Sunnyside
60	61	360	3	2	C	Sunnyside	Emo	Assiniboine
61	54	130	2	2	D	Assiniboine	Sunnyside	Woodhaven
54	53	125	3	2	C	Assiniboine	Woodhaven	Glendale
53	52	370	3	2	C	Ashcroft	Assiniboine	E. End
52	53	370	3	2	C	Ashcroft	E. End	Assiniboine
53	51	220	3	2	C	Glendale	Assiniboine	Woodhaven
51	53	220	3	2	C	Glendale	Woodhaven	Assiniboine
53	54	125	3	2	C	Assiniboine	Glendale	Woodhaven
54	61	130	2	2	D	Assiniboine	Woodhaven	Sunnyside
61	60	360	3	2	C	Sunnyside	Assiniboine	Emo
60	50	150	3	2	C	Emo	Sunnyside	Woodhaven
50	49	340	2	2	D	Woodhaven	Emo	Portage
49	48	175	1	8	D	Portage	Woodhaven	Old Mill
48	46	390	3	2	C	Old Mill	Portage	Woodbridge
46	45	210	3	2	C	Woodbridge	Old Mill	Portage
45	43	85	1	8	D	Portage	Woodbridge	Thompson
43	41	87	1	8	D	Portage	Thompson	Windham
41	39	100	1	8	D	Portage	Windham	Lake
39	36	125	1	8	D	Portage	Lake	Moray
36	37	150	3	2	C	Moray	Portage	Pinewood
37	35	420	3	2	C	Pinewood	Moray	Lake
35	34	88	2	2	D	Assiniboine	Pinewood	Olive
34	33	150	3	2	C	Olive	Assiniboine	Portage
33	32	230	1	8	D	Portage	Olive	Assiniboine
32	30	1740	1	8	D	Portage	Assiniboine	Deer Lodge
31	29	670	3	2	C	Deer Lodge	Portage	Portage
29	27	112	1	8	D	Portage	Deer Lodge	Albany
27	25	175	1	8	D	Portage	Albany	Douglas
25	23	95	1	8	D	Portage	Douglas	Library
23	21	212	1	8	D	Portage	Library	Ferry
21	19	90	1	8	D	Portage	Ferry	Colligiate

19	17	88	1	8	D	Portage	Colligiate	Roseberry
17	15	88	1	8	D	Portage	Roseberry	Parkview
15	13	90	1	8	D	Portage	Parkview	Riveroaks
13	11	80	1	8	D	Portage	Riveroaks	Bourkevale
11	12	340	3	2	C	Bourkevale	Portage	Assiniboine
12	11	340	3	2	C	Bourkevale	Assiniboine	Portage
11	9	78	1	8	D	Portage	Bourkevale	Cavell
9	5	80	1	8	D	Portage	Cavell	Winston
5	6	225	3	2	C	Winston	Portage	Parkside
6	8	270	3	2	C	Parkside	Winston	Assiniboine
8	6	270	3	2	C	Parkside	Assiniboine	Winston
6	7	175	3	2	C	Winston	Parkside	Assiniboine
7	6	175	3	2	C	Winston	Assiniboine	Parkside
6	5	225	3	2	C	Winston	Parkside	Portage
5	4	188	1	8	D	Portage	Windston	Riverbend
4	3	243	3	2	C	Riverbend	Portage	Graden
3	1	237	3	2	C	Riverbend	Graden	Portage

*C - CLEAR OUT D - DEADHEADING

TOTAL DEADHEADING: 18,742 M

Appendix I
List of the Computer Program

```
PROGRAM SCVRPGM(INPUT, OUTPUT, NAMEDATA, LENGDATA, PRIODATA);

(***************************************************************)
(*   SCVRPRGM is a computer program designed to solve the snow*)
(* clearing vehicle routing problem (SCVRP) in a hierarchical *)
(* street network system. It contains the following main      *)
(* procedures:                                                 *)
(*                                                             *)
(*     1. BULDNTWK - a procedure that builds the sub-networks  *)
(*        G(1), G(2), ..., G(PL)  (PL = # of priorities)       *)
(*                                                             *)
(*     2. MAKECONN - a procedure for constructing the connected*)
(*        sub-networks G(1)*, G(2)*, ..., G(PL)*               *)
(*                                                             *)
(*     3. MAKEEVEN - a procedure for constructing the even     *)
(*        sub-networks G(1)**, G(2)**, ..., G(PL)**            *)
(*                                                             *)
(*     4. POSTTOUR - a procedure for finding the Euler tours on*)
(*        the sub-networks G(1)**, G(2)**, ..., G(PL)**        *)
(*                                                             *)
(*                                                             *)
(*   INPUT :                                                   *)
(*             N - number of nodes in the total network        *)
 (*           PL - number of priority levels                   *)
(*                                                             *)
(*        NETWORK - N x N matrix contains the network data     *)
(*            It has six fields for each street block:         *)
(*                 1. Name                                     *)
(*                 2. From-Name                                *)
(*                 3. To-Name                                  *)
(*                 4. Number of lanes                          *)
(*                 5. Length                                   *)
(*                 6. Priority                                *)
(*                                                             *)
(*   OUTPUT :                                                  *)
(*             Optimal routes for each of priority levels      *)
(*                                                             *)
(***************************************************************)


CONST
  N = 200; (* Number of nodes in network *)
  PL = 3; (* Number of priority levels in network *)
  MAXNUM = 99999;

TYPE
   ARRN  = ARRAY[1..N] OF INTEGER;
   ARRNN = ARRAY[1..N,1..N] OF INTEGER;
   BOOLARRN = ARRAY[1..N] OF BOOLEAN;
   BOOLARRNN = ARRAY[1..N,1..N] OF BOOLEAN;


   TRANSPDATATYPE = RECORD
                    NODE : 0..N;
                    AMOUNT : 0..MAXNUM;
                    END;

   NODETYPE = ARRAY[1..N] OF TRANSPDATATYPE;


   EDGE = RECORD
          LENGTH : 0..MAXNUM;
          PRIO : 0..PL;
          END;
```

```
    EDGEPOINTER = @PEDGE;
          PEDGE = RECORD
                  FNODE : 0..N;
                  TNODE : 0..N;
                  NAME : STRING(51);
                  LENGTH : 0..MAXNUM;
                  PRIO : 0..PL;
                  PREV, NEXT : EDGEPOINTER;
                  INIT, FINAL : INTEGER;
                  CLEARED : BOOLEAN;
                  END;

    ITEMPOINTER = @ITEM;
           ITEM = RECORD
                  ROWPOSI, COLPOSI : INTEGER;
                  VAL : INTEGER;
                  NEXT : ITEMPOINTER;
                  END;

    NAMEPOINTER = @PNAME;
          PNAME = RECORD
                  ROWPOSI, COLPOSI : INTEGER;
                  EDGENAME : STRING(51);
                  NEXT : NAMEPOINTER;
                  END;

    NODEPOINTER = @PNODE;
          PNODE = RECORD
                  NODE : 0..N;
                  NEXT : NODEPOINTER;
                  END;

VAR
    I, J, K, P, L: INTEGER;
    EDGEMAXNUMO : ARRAY[1..N,1..N] OF EDGE;
    DIST, PRED : ARRN;
    LAST : BOOLARRN;
    OK, PATH : BOOLEAN;
    WGHTTAB, PRIOTAB, SHRTTAB: ITEMPOINTER;
    NAMETAB : NAMEPOINTER;
    CURRENT, NEWEDGE: EDGEPOINTER;
    HEAD, TAIL, SUBNTWK : ARRAY [1..PL] OF EDGEPOINTER;
    NAMEDATA, LENGDATA, PRIODATA, NTWKDATA : TEXT;


PROCEDURE NEWTAIL ( NEWEDGE : EDGEPOINTER; P : INTEGER );

(* Inserts an edge at the tail of the list *)

BEGIN
  NEWEDGE@.PREV := TAIL[P];
  NEWEDGE@.NEXT := NIL;
  TAIL[P]@.NEXT := NEWEDGE;
  TAIL[P] := NEWEDGE;
END; (* Newtail *)


FUNCTION RV ( LST : ITEMPOINTER; I, J: INTEGER ): INTEGER;

(* This procedure finds real value in position (i,j) of a data table *)

VAR
    TABLE : ITEMPOINTER;
    FOUND : BOOLEAN;

BEGIN
TABLE := LST;
FOUND:= FALSE;
WHILE (TABLE <> NIL) AND (NOT FOUND) DO
  IF (TABLE@.ROWPOSI = I) AND (TABLE@.COLPOSI = J) THEN
    BEGIN
    RV := TABLE@.VAL;
    FOUND:= TRUE;
    END
  ELSE
```

```
    TABLE := TABLE@.NEXT
END;

FUNCTION RN ( LST : NAMEPOINTER; I, J: INTEGER ): STRING(51);

(* This function finds edge name in position (i,j) of a data table *)

VAR
   TABLE : NAMEPOINTER;
   FOUND : BOOLEAN;

BEGIN
TABLE := LST;
FOUND:= FALSE;
WHILE (TABLE <> NIL) AND (NOT FOUND) DO
  IF (TABLE@.ROWPOSI = I) AND (TABLE@.COLPOSI = J) THEN
    BEGIN
    RN := TABLE@.EDGENAME;
    FOUND:= TRUE;
    END
  ELSE
    TABLE := TABLE@.NEXT
END;

PROCEDURE SHOWLIST ( NTWK: EDGEPOINTER );

  (* Displays the list in order as entered so far *)

BEGIN
  CURRENT := NTWK;
  WRITELN;
  REPEAT
    WITH CURRENT@ DO
      BEGIN
      WRITELN('EDGE (',FNODE:2,',',TNODE:2,') L = ',LENGTH:2);
      END;
    CURRENT:= CURRENT@.NEXT;
  UNTIL CURRENT = NIL;
  WRITELN;
END; (* Showlist *)


PROCEDURE SHORTESTPATH( N,MAXNUM,S,T     :INTEGER;
                        VAR PATH      :BOOLEAN;
                        VAR FINAL     :BOOLARRN;
                        VAR DIST,PRED: ARRN);

(***************************************************************)
(*   This procedure finds the shortest path from one node to  *)
(* another in the network                                      *)
(***************************************************************)


VAR   U,V,Y,RECENT,NEWLABEL,TEMP:INTEGER;


BEGIN
  FOR V:= 1 TO N DO
    BEGIN
      DIST[V]:= MAXNUM; (*INF = WEIGHT OF A NON-EXISTENT EDGE*)
      FINAL[V]:= FALSE;
      PRED[V]  := -1
    END;
  DIST[S]:= 0; FINAL[S]:= TRUE;
  PATH:= TRUE; RECENT:= S;
    (*INITIALIZATION OVER*)

  WHILE NOT FINAL[T] DO
  BEGIN
    FOR V:=1 TO N DO
      IF (RV(WGHTTAB,RECENT,V) < MAXNUM) AND (NOT FINAL[V]) THEN
      BEGIN
        NEWLABEL:= DIST[RECENT] + RV(WGHTTAB,RECENT,V);
        IF NEWLABEL < DIST[V] THEN
        BEGIN
```

```
          DIST[V]:= NEWLABEL;
          PRED[V]:= RECENT
        END
      END;
    TEMP:= MAXNUM;
    FOR U:= 1 TO N DO  (*FIND SMALLEST LABELED NODE*)
      IF (NOT FINAL[U]) AND (DIST[U] < TEMP) THEN
      BEGIN
        Y:=U;
        TEMP:= DIST[U]
      END;
    IF TEMP < MAXNUM THEN
    BEGIN
      FINAL[Y]:= TRUE;
      RECENT:= Y
    END
    ELSE
    BEGIN
      PATH:=FALSE;
      FINAL[T]:= TRUE
    END
  END (*WHILE*)
END; (*SHORTESTPATH*)

                (************************************)
                (*                                  *)
                (*    MAIN PROCEDURE 1 BULDNTWK     *)
                (*                                  *)
                (************************************)


PROCEDURE BULDNTWK;

(************************************************************)
(* This procedure reads in the network data and builds sub-  *)
(* network for each priority level                          *)
(************************************************************)

VAR
  I, J, K, L, P : INTEGER;
  PRE, POST : EDGEPOINTER;
  CURRITEM : ITEMPOINTER;
  CURRNAME : NAMEPOINTER;
  NM : STRING(51);

BEGIN (* BULDNTWK *)
  (* Read in edge name data from NAMEDATA *)
  RESET(NAMEDATA);
  NAMETAB := NIL;
  FOR I:= 1 TO N DO
    FOR J:= 1 TO N DO
      BEGIN
      READLN(NAMEDATA, NM);
      NEW(CURRNAME);
      WITH CURRNAME@ DO
         BEGIN
         EDGENAME := NM; ROWPOSI := I; COLPOSI := J;
         END;
      CURRNAME@.NEXT := NAMETAB;
      NAMETAB := CURRNAME
      END;
  (* Read in edge length data from LENGDATA *)
  RESET(LENGDATA);
  WGHTTAB := NIL; (* Initialize length table *)
  FOR I:= 1 TO N DO
    FOR J:= 1 TO N DO
      BEGIN
      READLN(LENGDATA, L);
      NEW(CURRITEM);
      WITH CURRITEM@ DO
         BEGIN
         VAL := L; ROWPOSI := I; COLPOSI := J;
         END;
      CURRITEM@.NEXT := WGHTTAB;
```

```
      WGHTTAB := CURRITEM
      END;
  (* Read in edge priority data from PRIODATA *)
  RESET(PRIODATA);
  PRIOTAB := NIL; (* Initialize priority table *)
  FOR I:= 1 TO N DO
    FOR J:= 1 TO N DO
      BEGIN
      READLN(PRIODATA, P);
      NEW(CURRITEM);
      WITH CURRITEM@ DO
         BEGIN
         VAL := P; ROWPOSI := I; COLPOSI := J;
         END;
      CURRITEM@.NEXT := PRIOTAB;
      PRIOTAB := CURRITEM
      END;
  FOR I:=1 TO N DO
    BEGIN
    FOR J:=1 TO N DO
      WRITE(RV(WGHTTAB,I,J):5,' ');
    WRITELN;
    END;
  WRITELN;
  (* Builds shortest distance matrix *)
  SHRTTAB := NIL;
  FOR I:= 1 TO N DO
    FOR J:= 1 TO N DO
      BEGIN
      SHORTESTPATH (N,MAXNUM,I,J,PATH,LAST,DIST,PRED);
      NEW(CURRITEM);
      WITH CURRITEM@ DO
         BEGIN
         VAL := DIST[J]; ROWPOSI := I; COLPOSI := J;
         END;
      CURRITEM@.NEXT := SHRTTAB;
      SHRTTAB := CURRITEM
      END;
  (* Build subnetwork for each priority level *)
  FOR K:=1 TO PL DO
    BEGIN
    HEAD[K]:= NIL;
    TAIL[K]:= NIL;
     (* First edge *)
    FOR I:=1 TO N DO
      FOR J:=1 TO N DO
        BEGIN
        IF ( RV(WGHTTAB,I,J) < 10000 ) AND ( RV(PRIOTAB,I,J) = K ) THEN
          BEGIN
          NEW (CURRENT);
          WITH CURRENT@ DO
            BEGIN
            FNODE:= I; TNODE:= J;
            NAME:=RN(NAMETAB,I,J); LENGTH:= RV(WGHTTAB,I,J);
            PRIO:= RV(PRIOTAB,I,J); INIT:= 0; FINAL:= 0; CLEARED:= TRUE;
            END;
          IF (HEAD[K] = NIL) AND (TAIL[K] = NIL) THEN
            BEGIN
            CURRENT@.PREV := NIL; CURRENT@.NEXT := NIL;
            HEAD[K]:=CURRENT; TAIL[K]:=CURRENT
            END
          ELSE (* append to an existing list *)
            NEWTAIL (CURRENT,K);
          END
        END;
    SUBNTWK[K] := HEAD[K];
    END;
END; (* BULDNTWK *)



PROCEDURE INITIALIZE ( SUBNTWK: EDGEPOINTER );

BEGIN
WHILE SUBNTWK <> NIL DO
```

```
  BEGIN
  WITH SUBNTWK@ DO
    BEGIN
    INIT := 0;
    FINAL := 0;
    END;
  SUBNTWK := SUBNTWK@.NEXT;
  END;
END;



PROCEDURE DUPEDGES (VAR SUBNTWK : EDGEPOINTER;
                        DUP : BOOLARRNN;
                        P : INTEGER);

(*************************************************************)
(*   This procedure duplicates the eges of a given subnetwork *)
(* to form a augmented subnetwork                            *)
(*************************************************************)

VAR
  I, J, WORKINGNODE : INTEGER;
  PATH : BOOLEAN;
  LAST : BOOLARRN;
  DIST : ARRN;
  PRED : ARRN;
  CURRENT, NEWEDGE: EDGEPOINTER;


BEGIN
FOR I:= 1 TO N DO
  FOR J:= 1 TO N DO
    IF DUP[I,J] THEN
      BEGIN
      SHORTESTPATH (N,MAXNUM,I,J,PATH,LAST,DIST,PRED);
      WRITE('DUPLICTAING SHORTEST PATH FROM ');
      WRITELN(I:2,' TO ',J:2,': ',DIST[J]:2);
      WORKINGNODE:= J;
      WHILE WORKINGNODE <> I DO
        BEGIN
        WRITE('DUPLICATING EDGE ',WORKINGNODE:2,',');
        WRITELN(PRED[WORKINGNODE]:2);
        NEW(NEWEDGE);
        WITH NEWEDGE@ DO
          BEGIN
          INIT:= 0; FINAL:= 0;
          FNODE:= PRED[WORKINGNODE]; TNODE:= WORKINGNODE;
          NAME:= RN(NAMETAB,PRED[WORKINGNODE],WORKINGNODE);
          LENGTH:= RV(WGHTTAB, PRED[WORKINGNODE], WORKINGNODE);
          PRIO := RV(PRIOTAB, PRED[WORKINGNODE], WORKINGNODE);
          CLEARED := FALSE;
          END;
        NEWTAIL(NEWEDGE,P);
        WORKINGNODE:= PRED[WORKINGNODE];
        END;
      END;
WRITELN
END; (*Dupledge*)



              (***********************************)
              (*                                 *)
              (*    MAIN PROCEDURE 2 MAKECONN    *)
              (*                                 *)
              (***********************************)



PROCEDURE MAKECONN ( SUBNTWK : EDGEPOINTER;
                     P : INTEGER );
```

```
(**************************************************************)
(*   This procedure first checks the given sub-network to see *)
(* if it is connected. If it is not, it then uses the TSP ap- *)
(* proach to connect the components                           *)
(*                                                            *)
(*  The procedure contains the following main sub-procedures: *)
(*                                                            *)
(*    2.1. IDTCOMPT - it identifies the components as well as *)
(*         the shortest paths between each pair of them       *)
(*                                                            *)
(*    2.2. TSPCONN - it determines the best strategy to con-  *)
(*         nect the components                                *)
(*                                                            *)
(**************************************************************)


TYPE
   CONNDATATYPE = RECORD
                  DIST : 0..MAXNUM;
                  CONN : BOOLEAN;
                  FN : 0..N;
                  TN : 0..N;
                  END;
   COMPTCONNDATATYPE = ARRAY[1..N,1..N] OF CONNDATATYPE;

VAR
  I, J, NOC : INTEGER;
  X : ARRNN;
  ROUTE : ARRN;
  TOTALCOST : INTEGER;
  COMPT : COMPTCONNDATATYPE;
  OK : BOOLEAN;
  DUP : BOOLARRNN;



PROCEDURE IDNTCOMPT ( SUBNTWK : EDGEPOINTER;
                      VAR COMPTCONN : COMPTCONNDATATYPE;
                      VAR NOC : INTEGER );


VAR
  CURRENT : EDGEPOINTER;
  WCOMPTA, WCOMPTB, WCOMPTC, NEWNODE   : NODEPOINTER;
  COMPT : ARRAY[1..N] OF NODEPOINTER;
  I, J, K, M, Q, FN, TN, TOTALCOST : INTEGER;
  FOUND, CYCLECOMPLETED, ALLCOMPTEDGETRACED, NWNODEFOUND : BOOLEAN;
  NEXTNODE, NUMOFCOMPT : INTEGER;
  REPLY : CHAR;


PROCEDURE FNWNODE ( NTWK: EDGEPOINTER;
                    EXAMEDNODE: INTEGER;
                    VAR FOUND : BOOLEAN );
VAR
  CURRENT : EDGEPOINTER;

BEGIN (* FNWNODE *)
  CURRENT:= NTWK;
  FOUND := FALSE;
  WHILE (CURRENT <> NIL) AND (NOT FOUND) DO
    IF (CURRENT@.FNODE = EXAMEDNODE) AND (CURRENT@.INIT = 0) THEN
      FOUND := TRUE
    ELSE
      CURRENT:= CURRENT@.NEXT;
END; (* FWNODE *)


PROCEDURE CCDIST ( SCOMPT, TCOMPT : NODEPOINTER;
                          VAR TOTALCOST: INTEGER;
                          VAR FN, TN : INTEGER );
(* This procedure finds the shortest distance from component *)
(* to component                                              *)
```

```
VAR
  WCOMPT : NODEPOINTER;
  SPOINT, EPOINT, MINVAL, TEMPCOST, TEMPFN, TEMPTN : INTEGER;


PROCEDURE PCDIST ( MINVAL : INTEGER;
                             SPOINT : INTEGER;
                             WCOMPT : NODEPOINTER;
                             VAR TEMPCOST : INTEGER;
                             VAR TEMPFN, TEMPTN : INTEGER );
(* This procedure finds the shortest distance from point to *)
(* component                                                *)

BEGIN (* PCDIST *)
WHILE WCOMPT <> NIL DO
  BEGIN
  EPOINT := WCOMPT@.NODE;
  IF RV(SHRTTAB, SPOINT, EPOINT) < MINVAL THEN
     BEGIN
     MINVAL := RV(SHRTTAB, SPOINT, EPOINT);
     TEMPFN := SPOINT;
     TEMPTN := EPOINT;
     END;
  WCOMPT := WCOMPT@.NEXT;
  END;
TEMPCOST := MINVAL;
END; (* PCDIST *)


BEGIN (* CCDIST *)
MINVAL := MAXNUM;
WHILE SCOMPT <> NIL DO
  BEGIN
  SPOINT := SCOMPT@.NODE;
  WCOMPT := TCOMPT;
  PCDIST (MINVAL, SPOINT, WCOMPT, TEMPCOST, TEMPFN, TEMPTN);
  IF TEMPCOST < MINVAL THEN
    BEGIN
    MINVAL := TEMPCOST;
    FN := TEMPFN;
    TN := TEMPTN;
    END;
  SCOMPT := SCOMPT@.NEXT
  END;
TOTALCOST := MINVAL;
END; (* CCDIST  *)



BEGIN (* IDNTCOMPT *)
  CURRENT:= SUBNTWK;
  NUMOFCOMPT := 0;
  WHILE CURRENT <> NIL DO (* Find an un-treated edge *)
    WITH CURRENT@ DO
    IF (INIT = 0) AND (FINAL = 0) THEN
      BEGIN
      NUMOFCOMPT := NUMOFCOMPT + 1;
      INIT := 1;
      NEXTNODE := TNODE;
      CURRENT := CURRENT@.NEXT;
      IF CURRENT <> NIL THEN
        BEGIN
        ALLCOMPTEDGETRACED := FALSE;
        I := 2; J := 1;
        END
      ELSE
        ALLCOMPTEDGETRACED := TRUE;
      REPEAT (* until all edges of a component are traced *)

        CURRENT := SUBNTWK;
        CYCLECOMPLETED := FALSE;
        REPEAT (* unitl a new cycle is completed *)
          WITH CURRENT@ DO
          IF (INIT = 0) AND (FNODE = NEXTNODE ) THEN
            BEGIN
```

```
          INIT := I; I:= I + 1; NEXTNODE := TNODE;
          CURRENT := SUBNTWK; (* Start from beginning *)
          END
        ELSE
          CURRENT := CURRENT@.NEXT;
        IF CURRENT = NIL THEN
          CYCLECOMPLETED := TRUE;
      UNTIL CYCLECOMPLETED; (* A new cycle is completed *)
      CURRENT:= SUBNTWK;
      I:= I - 1;
      NWNODEFOUND:= FALSE;

      REPEAT
        WITH CURRENT@ DO
        IF (INIT = I) AND (FINAL = 0) THEN
          BEGIN
          FINAL:= J; INIT:= -1;
          NWNODEFOUND := FALSE;
          NEXTNODE:= FNODE;
          J:= J + 1;
          FNWNODE ( SUBNTWK, NEXTNODE, NWNODEFOUND);
          IF NOT NWNODEFOUND THEN
            BEGIN
            I:= I - 1;
            CURRENT:= SUBNTWK;
            END;
          END
        ELSE
          CURRENT:= CURRENT@.NEXT;
      UNTIL ( I = 0 ) OR ( NWNODEFOUND );

      IF NWNODEFOUND THEN
        CURRENT := SUBNTWK;
      IF I = 0 THEN (* All links have been traced *)
        ALLCOMPTEDGETRACED := TRUE;
    UNTIL ALLCOMPTEDGETRACED;
    WRITELN('** COMPONENT ',NUMOFCOMPT:2);
    COMPT[NUMOFCOMPT] := NIL;
    J := J - 1; Q := J;
    CURRENT := SUBNTWK;
    REPEAT
      WITH CURRENT@ DO
      BEGIN
      IF FINAL = J  THEN
        BEGIN
        FINAL := 0;
        NEW(NEWNODE);
        NEWNODE@.NODE := TNODE;
        NEWNODE@.NEXT := COMPT[NUMOFCOMPT];
        COMPT[NUMOFCOMPT] := NEWNODE;
        WRITELN ('(',FNODE:2,',',TNODE:2,') ',LENGTH:2,',',PRIO:2);
        J := J - 1;
        CURRENT := SUBNTWK;
        END
      ELSE
        CURRENT := CURRENT@.NEXT;
      END;
    UNTIL J = 0;
    WRITELN;
    END
  ELSE
    CURRENT := CURRENT@.NEXT;
NOC := NUMOFCOMPT;
FOR I:= 1 TO NUMOFCOMPT DO
  FOR J:= 1 TO NUMOFCOMPT DO
    IF I <> J THEN
       BEGIN
       WCOMPTB := COMPT[I];
       WCOMPTC := COMPT[J];
       CCDIST ( WCOMPTB, WCOMPTC, TOTALCOST, FN, TN );
       COMPTCONN[I,J].DIST := TOTALCOST;
       COMPTCONN[I,J].FN := FN;
       COMPTCONN[I,J].TN := TN;
       END;
END; (* IDNTCOPMT *)
```

```
PROCEDURE  TSPCONN ( N, MAXNUM : INTEGER;
                     VAR W : COMPTCONNDATATYPE;
                     VAR ROUTE : ARRN;
                     VAR TWEIGHT : INTEGER);


VAR
  BACKPTR, BEST, COL, FWDPTR, ROW : ARRN;
  I, INDEX : INTEGER;


PROCEDURE EXPLORE ( EDGES, COST : INTEGER;
                   VAR ROW, COL: ARRN);

Var
  AVOID, C, COLROWVAL, FIRST, I, J,
     LAST, LOWERBOUND, MOST, R, SIZE: INTEGER;
  NEWCOL, NEWROW, ROWRED, COLRED : ARRN;


FUNCTION MINIMIZE ( I, J : INTEGER ): INTEGER;
  BEGIN
  IF I <= J THEN MINIMIZE:= I
  ELSE MINIMIZE:= J
  END;


FUNCTION REDUCE( VAR ROW, COL, ROWRED, COLRED: ARRN ): INTEGER;

VAR
   I, J, TEMP, RVALUE : INTEGER;

BEGIN
  RVALUE:= 0;
  FOR I:= 1 TO SIZE DO
    BEGIN
    TEMP:= MAXNUM;
    FOR J:= 1 TO SIZE DO
      TEMP:= MINIMIZE ( TEMP,W[ROW[I], COL[J]].DIST );
    IF TEMP > 0 THEN
    BEGIN
      FOR J:= 1 TO SIZE DO
        IF W[ROW[I],COL[J]].DIST < MAXNUM THEN
           W[ROW[I],COL[J]].DIST := W[ROW[I], COL[J]].DIST - TEMP;
      RVALUE := RVALUE + TEMP;
    END;
    ROWRED[I]  := TEMP;
    END;
  FOR J  := 1 TO SIZE DO
    BEGIN
    TEMP  := MAXNUM;
    FOR I  := 1 TO SIZE DO
      TEMP:= MINIMIZE ( TEMP, W[ROW[I],COL[J]].DIST );
    IF TEMP > 0 THEN
    BEGIN
      FOR I:= 1 TO SIZE DO
        IF W[ROW[I],COL[J]].DIST < MAXNUM THEN
            W[ROW[I],COL[J]].DIST := W[ROW[I],COL[J]].DIST - TEMP;
      RVALUE:= RVALUE + TEMP;
    END;
    COLRED[J]:= TEMP;
    END;
  REDUCE:= RVALUE;
END; (*Procedure REDUCE*)


PROCEDURE BESTEDGE ( VAR R, C, MOST : INTEGER );

VAR
   I, J, K, ZEROES, MINCOLELT, MINROWELT : INTEGER;

BEGIN (* BESTEDGE *)
   MOST:= -MAXNUM;
   FOR I:= 1 TO SIZE DO
```

```
      FOR J:= 1 TO SIZE DO
          IF W[ROW[I],COL[J]].DIST = 0 THEN
          BEGIN
            MINROWELT:= MAXNUM;
            ZEROES:= 0;
            FOR K:= 1 TO SIZE DO
                BEGIN
                  IF W[ROW[I],COL[K]].DIST = 0 THEN ZEROES:= ZEROES + 1
                  ELSE
                    MINROWELT:= MINIMIZE(MINROWELT, W[ROW[I],COL[K]].DIST);
                END;
            IF ZEROES > 1 THEN MINROWELT:= 0;
            MINCOLELT:= MAXNUM;
            ZEROES:= 0;
            FOR K:= 1 TO SIZE DO
                BEGIN
                  IF W[ROW[K],COL[J]].DIST = 0 THEN ZEROES:= ZEROES+1
                  ELSE
                    MINCOLELT:= MINIMIZE(MINCOLELT, W[ROW[K],COL[J]].DIST);
                END;
            IF ZEROES > 1 THEN MINCOLELT:= 0;
            IF ( MINROWELT + MINCOLELT ) > MOST THEN
                BEGIN
                MOST:= MINROWELT + MINCOLELT;
                R:= I;
                C:= J;
                END;
          END;
END; (* BESTEDGE *)


BEGIN  (* EXPLORE *)
SIZE:= N - EDGES;  (*Number of rows,cols left in matrix*)
COST:= COST + REDUCE ( ROW, COL, ROWRED, COLRED);
IF COST < TWEIGHT THEN
  BEGIN
    IF EDGES = ( N - 2 ) THEN
       BEGIN (*Last two edges are forced*)
         FOR I:= 1 TO N DO BEST[I]:= FWDPTR[I];
           BEGIN
             IF W[ROW[1],COL[1]].DIST = MAXNUM THEN AVOID:= 1
             ELSE AVOID:= 2;
           END;
         BEST[ROW[1]]:= COL[3 - AVOID];
         BEST[ROW[2]]:= COL[AVOID];
         TWEIGHT:= COST;
       END
    ELSE
       BEGIN
         BESTEDGE(R, C, MOST);
         LOWERBOUND:= COST + MOST;
         FWDPTR[ROW[R]]:= COL[C];
         BACKPTR[COL[C]]:= ROW[R];
         LAST:= COL[C];
         WHILE FWDPTR[LAST] <> 0 DO LAST:= FWDPTR[LAST];
         FIRST:= ROW[R];
         WHILE BACKPTR[FIRST] <> 0 DO FIRST:= BACKPTR[FIRST];
         COLROWVAL:= W[LAST,FIRST].DIST;
         W[LAST,FIRST].DIST := MAXNUM;
         FOR I:= 1 TO R - 1 DO NEWROW[I]:= ROW[I];  (*Remove Row*)
         FOR I:= R TO SIZE - 1 DO NEWROW[I]:= ROW[I+1];
         FOR I:= 1 TO C - 1 DO NEWCOL[I]:= COL[I];  (*Remove Col*)
         FOR I:= C TO SIZE - 1 DO NEWCOL[I]:= COL[I+1];
         EXPLORE ( EDGES + 1, COST, NEWROW, NEWCOL );
         W[LAST,FIRST].DIST := COLROWVAL;  (*Restore previous value*)
         BACKPTR[COL[C]]:= 0;
         FWDPTR[ROW[R]]:= 0;
         IF LOWERBOUND < TWEIGHT THEN
           BEGIN
             W[ROW[R],COL[C]].DIST := MAXNUM;
             EXPLORE ( EDGES, COST, ROW, COL);
             W[ROW[R],COL[C]].DIST := 0;
           END;
         END;
  END;
  FOR I:= 1 TO SIZE DO
```

```
    FOR J:= 1 TO SIZE DO
      W[ROW[I],COL[J]].DIST := W[ROW[I], COL[J]].DIST +
                   ROWRED[I] + COLRED[J];
END; (* Explore *)


BEGIN (*TSPCONNECT*)
  FOR I:= 1 TO N DO
  BEGIN
    ROW[I]:= I;
    COL[I]:= I;
    FWDPTR[I]:= 0;
    BACKPTR[I]:= 0;
  END;
  TWEIGHT:= MAXNUM;
  EXPLORE( 0, 0, ROW, COL);
  INDEX:= 1;
  FOR I:= 1 TO N DO
    BEGIN
      ROUTE[I] := INDEX;
      INDEX:= BEST[INDEX];
    END;
END; (* TSPCONNECT *)


BEGIN (* MAKECONN *)
  FOR I:= 1 TO N DO
    BEGIN
    ROUTE[I] := 0;
    FOR J:= 1 TO N DO
      BEGIN
      DUP[I,J] := FALSE; X[I,J] := 0;
      COMPT[I,J].DIST := MAXNUM;
      COMPT[I,J].CONN := FALSE;
      COMPT[I,J].FN := 0;
      COMPT[I,J].TN := 0;
      END;
    END;
  IDNTCOMPT (SUBNTWK, COMPT, NOC);
  IF NOC > 1 THEN
    BEGIN (* Searching for the best connection strategy *)
    OK := FALSE;
    FOR I:= 1 TO NOC DO
      FOR J:= 1 TO NOC DO
        IF I <> J THEN
          WITH COMPT[I,J] DO
            BEGIN
            WRITE('SHORTEST DISTANCE FROM COMPT ',I:2,' TO COMPT ');
            WRITELN(J:2,' IS: ',DIST:3,' (',FN:2,',',TN:2,')');
            END;
    WRITELN;
    TSPCONN ( NOC, MAXNUM, COMPT, ROUTE, TOTALCOST );
    WRITE ('THE BEST CONNECTION STRATEGY IS: [');
    FOR I:=1 TO ( NOC - 1 ) DO
      BEGIN
      COMPT[ROUTE[I],ROUTE[I+1]].CONN := TRUE;
      WRITE(ROUTE[I]:2,',');
      END;
    COMPT[ROUTE[NOC],ROUTE[1]].CONN := TRUE;
    WRITE (ROUTE[NOC]:2,',', ROUTE[1]:2);
    WRITELN ('] WITH TOTAL COST : ', TOTALCOST:2 );
    WRITELN;
    FOR I:= 1 TO NOC DO
      FOR J:= 1 TO NOC DO
        IF COMPT[I,J].CONN THEN
          DUP[COMPT[I,J].FN, COMPT[I,J].TN] := TRUE;
    END (* Searching for best connection startegy *)
  ELSE
    OK := TRUE;
IF NOT OK THEN DUPEDGES (SUBNTWK, DUP, P);
END; (* MAKECONN *)


            (*******************        ************)
```

```
          (*                                   *)
          (*    MAIN PROCEDURE 3 MAKEEVEN      *)
          (*                                   *)
          (*************************************)

PROCEDURE MAKEVEN ( SUBNTWK : EDGEPOINTER;
                    P : INTEGER );

(***************************************************************)
(*  This procedure checks the given sub-network if it is even.*)
(* If it is not, the procedure then usese max-flow min-cost   *)
(* approach to determine the best strategy to make the sub-   *)
(* network even                                               *)
(*                                                            *)
(*  The procedure contain the following main sub-procedure:   *)
(*                                                            *)
(*    3.1. TRANSPORT - it determines the best set of edges to *)
(*         be duplicated to make the sub-network even         *)
(***************************************************************)

TYPE
  DEGREE = RECORD
           INDEG: INTEGER;
           OUTDEG : INTEGER;
           END;

VAR
  A, B : NODETYPE;
  C, X : ARRNN;
  KO : INTEGER;
  OUTDATAFILE : TEXT;
  NODEDEG : ARRAY [1..N] OF DEGREE;
  ABSDEG : INTEGER;
  CURRENT : EDGEPOINTER;
  I, J : INTEGER;
  INCOUNT, OUTCOUNT : INTEGER;
  DUP : BOOLARRNN;
  OK : BOOLEAN;


PROCEDURE TRANSPORT(
   M,MAXNUM   :INTEGER;
   VAR  A    :NODETYPE;
   VAR  B    :NODETYPE;
   VAR  C,X :ARRNN;
   VAR  KO   :INTEGER);

(***************************************************************)
(*  This procedure uses the TRANSPORTATION algorithm to de-    *)
(* termine the best strategy to duplicate the edges of a non- *)
(* even subnetwork to make it even                             *)
(***************************************************************)

VAR
  I,J,SF,R,RA   :INTEGER;
  LAB,LAB1,LAB2:BOOLEAN;
  U,W,EPS       :ARRN;
  V,K,DEL       :ARRN;


BEGIN (* TRANSPORT *)
   FOR I:=1 TO M DO U[I]:=0;
   FOR J:=1 TO M DO
   BEGIN
     R:=MAXNUM;
     FOR I:=1 TO M DO
     BEGIN
       X[I,J]:=0;
       SF:=C[I,J];
       IF SF < R THEN R:=SF
     END;
     V[J]:=R
   END;
   LAB1:= FALSE;
```

```
REPEAT
  (* INILIALIZATION OF ROW AND COLUMN LABELS *)
  FOR I:=1 TO M DO
  BEGIN
    W[I]:=0;
    EPS[I]:=A[I].AMOUNT
  END;
  FOR J:=1 TO M DO
  BEGIN
    K[J]:=0;
    DEL[J]:=0;
  END;
  REPEAT  (*UNTIL LAB*)
     (* PROCESS OF LABELLING ROWS AND COLUMNS*)
     LAB:=TRUE; (*LAB BECOMES FALSE WHEN A COLUMN IS LABELED*)
     LAB2:=TRUE; (*LAB2 IS FALSE IF BREAKTHROUGH*)
     I:=0;
     REPEAT (*UNTIL I = M OR LAB2 = FALSE*)
       I:=I+1;
       SF:=EPS[I]; EPS[I]:=-SF;
       IF SF > 0 THEN
       BEGIN (*ROW I BECOMES LABELED *)
         RA:=U[I];
         J:=0;
         REPEAT
           J:=J+1;
           IF (DEL[J] = 0) AND (V[J]-RA = C[I,J]) THEN
           BEGIN  (*ELEMENT I,J IS ADMISSIBLE*)
             K[J]:=I; (*COLUMN J CAN BE LABELED*)
             DEL[J]:=SF;
             LAB:=FALSE;
             IF B[J].AMOUNT > 0 THEN
             BEGIN (*BREAKTHROUGH*)
               LAB:= TRUE; LAB2:= FALSE;
               SF:=ABS(DEL[J]);
               R:=B[J].AMOUNT;
               IF R < SF THEN SF:=R;
               B[J].AMOUNT:=R - SF;
               REPEAT
                 I:=K[J];
                 X[I,J]:=X[I,J]+SF;
                 J:=W[I];
                 IF J <> 0 THEN
                   X[I,J]:=X[I,J]-SF
               UNTIL J = 0;
               A[I].AMOUNT:=A[I].AMOUNT - SF;
               J:=0;
               REPEAT
                 J:=J+1;
                 LAB1:=B[J].AMOUNT <= 0
               UNTIL (J = N) OR NOT LAB1;
               IF LAB1 THEN
               BEGIN
                 SF:=0;
                 FOR I:=1 TO M DO
                   FOR J:=1 TO M DO
                   BEGIN
                     R:=X[I,J];
                     IF R > 0 THEN SF:=SF+R*C[I,J]
                   END;
                 KO:=SF
               END (* LAB *)
             END (*BREAKTHROUGH*)
           END (* LABELING COLUMN J *)
         UNTIL (J = M) OR NOT LAB2
       END (* SF > 0 *)
     UNTIL (I = M) OR NOT LAB2;
     IF NOT LAB THEN
     BEGIN (* LABELING ROWS FROM COLUMNS *)
       LAB:=TRUE;
       FOR J:=1 TO M DO
       BEGIN
         SF:=DEL[J];
         IF SF > 0 THEN
         BEGIN
           FOR I:=1 TO M DO
```

```
                  IF EPS[I] = 0 THEN
                  BEGIN
                    R:=X[I,J];
                    IF R > 0 THEN
                    BEGIN
                      W[I]:= J;
                      IF R <= SF THEN EPS[I]:=R
                      ELSE EPS[I]:=SF;
                      LAB:=FALSE;
                    END
                  END; (*I, EPS[I]:=0*)
                DEL[J]:=-SF
              END (*SF > 0*)
            END (* J *)
          END (*NOT LAB*)
        UNTIL LAB; (*END OF LABELING*)
        IF LAB2 THEN
        BEGIN
          R:=MAXNUM;
          FOR I:=1 TO M DO
            IF EPS[I] <> 0 THEN
            BEGIN
              RA:=U[I];
              FOR J:=1 TO M DO
                IF DEL[J] = 0 THEN
                BEGIN
                  SF:=C[I,J] + RA - V[J];
                  IF R > SF THEN R:=SF
                END
            END; (* I, EPS[I] <> 0*)
          FOR I:=1 TO M DO
            IF EPS[I] = 0 THEN U[I]:=U[I]+R;
          FOR J:=1 TO M DO
            IF DEL[J] = 0 THEN V[J]:=V[J] + R;
        END (* LAB2 *)
      UNTIL LAB1
END; (* TRANSORT *)


BEGIN (* MAKEVEN *)
FOR I := 1 TO N DO
  BEGIN
  A[I].AMOUNT := 0; A[I].NODE := 0; B[I].AMOUNT := 0; B[I].NODE := 0;
  FOR J:=1 TO N DO
    BEGIN
    C[I,J] := MAXNUM; X[I,J]:= 0; DUP[I,J] := FALSE;
    END;
  END;
FOR I:= 1 TO N DO
  BEGIN
  INCOUNT := 0; OUTCOUNT := 0;
  CURRENT := SUBNTWK;
  WHILE CURRENT <> NIL DO
    BEGIN
    IF CURRENT@.FNODE = I THEN OUTCOUNT := OUTCOUNT + 1;
    IF CURRENT@.TNODE = I THEN  INCOUNT := INCOUNT + 1;
    CURRENT := CURRENT@.NEXT;
    END;
  NODEDEG[I].INDEG := INCOUNT;
  NODEDEG[I].OUTDEG := OUTCOUNT;
  END;

WRITELN;
OK := TRUE;
FOR I:= 1 TO N DO
  IF NODEDEG[I].INDEG > NODEDEG[I].OUTDEG THEN
    BEGIN
    OK := FALSE;
    ABSDEG := ABS(NODEDEG[I].INDEG - NODEDEG[I].OUTDEG);
    WRITELN ('ODD NDOE: ',I:2,'(', ABSDEG:2, ')');
    A[I].NODE := I; A[I].AMOUNT:= ABSDEG;
    END;

FOR I:= 1 TO N DO
  IF NODEDEG[I].INDEG < NODEDEG[I].OUTDEG THEN
    BEGIN
```

```
    ABSDEG := ABS(NODEDEG[I].INDEG - NODEDEG[I].OUTDEG);
    B[I].NODE := I; B[I].AMOUNT:= ABSDEG;
    WRITELN ('ODD NDOE: ',I:2,'(', ABSDEG:2, ')');
    END;

FOR I:= 1 TO N DO
  IF NODEDEG[I].INDEG > NODEDEG[I].OUTDEG THEN
    BEGIN
    FOR J:= 1 TO N DO
      IF NODEDEG[J].INDEG < NODEDEG[J].OUTDEG THEN
        C[I,J] := RV(SHRTTAB,I,J);
    END;

WRITELN;
TRANSPORT (N, MAXNUM, A, B, C, X, KO);
FOR I:= 1 TO N DO
  FOR J:= 1 TO N DO
    IF X[I,J] <> 0 THEN
       DUP[I,J] := TRUE;
IF NOT OK THEN DUPEDGES (SUBNTWK, DUP, P);
END; (* CHECKEVENESS *)



               (*************************************)
               (*                                   *)
               (*     MAIN PROCEDURE 4 POSTTOUR     *)
               (*                                   *)
               (*************************************)


PROCEDURE POSTTOUR( SUBNTWK : EDGEPOINTER;
                           P : INTEGER );

(*************************************************************)
(* This proceudre finds an Euler tour in a given connected and*)
(* even sub-network                                          *)
(*************************************************************)

VAR
  CURRENT : EDGEPOINTER;
  I, J, K, M, FN, TN : INTEGER;
  FOUND, ALLTRACED, NWNODEFOUND : BOOLEAN;
  NEXTNODE, CLENGTH, DLENGTH : INTEGER;


PROCEDURE FNWNODE ( NTWK: EDGEPOINTER;
                    EXAMEDNODE: INTEGER;
                    VAR FOUND : BOOLEAN );
VAR
  CURRENT : EDGEPOINTER;

BEGIN (*FNWNODE*)
  CURRENT:= NTWK;
  FOUND := FALSE;
  WHILE (CURRENT <> NIL) AND (NOT FOUND) DO
    IF (CURRENT@.FNODE = EXAMEDNODE) AND (CURRENT@.INIT = 0) THEN
      FOUND := TRUE
    ELSE
      CURRENT:= CURRENT@.NEXT;
END; (*FWNODE*)


BEGIN  (* POSTTOUR *)
  CLENGTH:=0; DLENGTH:=0;
  CURRENT:= SUBNTWK;
  WITH CURRENT@ DO
    BEGIN
    INIT := 1;
    NEXTNODE := TNODE;
    END;
  I:=2; J:=1;
  ALLTRACED := FALSE;
```

```
  REPEAT { until all rest edges are traced }
    CURRENT := SUBNTWK;
    REPEAT (* unitl a new cycle is completed *)
      IF (CURRENT@.INIT = 0) AND (CURRENT@.FNODE = NEXTNODE) THEN
         BEGIN
         CURRENT@.INIT := I; I:= I + 1;
         NEXTNODE := CURRENT@.TNODE;
         CURRENT := SUBNTWK; (* Starts from the beginning again *)
         END
      ELSE
         CURRENT := CURRENT@.NEXT;
    UNTIL CURRENT = NIL; (* A new cycle is completed *)
    CURRENT:= SUBNTWK;
    I:= I - 1;
    NWNODEFOUND:= FALSE;
    (* Back tracking *)
    REPEAT (* until a new working node is found *)
      WITH CURRENT@ DO
        IF (INIT = I) AND (FINAL = 0) THEN
          BEGIN (* Perm. label an edge *)
          FINAL:= J; INIT:= -1;
          NWNODEFOUND := TRUE;
          NEXTNODE:= FNODE;
          J:= J + 1;
          FNWNODE ( SUBNTWK, NEXTNODE, NWNODEFOUND);
          IF NOT NWNODEFOUND THEN
            BEGIN
            I:= I - 1;
            CURRENT:= SUBNTWK
            END
          END
        ELSE
          CURRENT:= CURRENT@.NEXT;
    UNTIL ( I = 0 ) OR ( NWNODEFOUND );
    IF NWNODEFOUND THEN
      CURRENT := SUBNTWK;
    IF I = 0 THEN (* All edges have been traced *)
      ALLTRACED := TRUE;
  UNTIL ALLTRACED; (* All egdes are traced *)
       (* output the Euler tour *)
  J := J - 1;
  CURRENT := SUBNTWK;
  REPEAT
   ,WITH CURRENT@ DO
      IF FINAL = J  THEN
        BEGIN (* outputing a matched edge *)
        FINAL := 0;
        WRITE(NAME:51,' (',FNODE:2,',',TNODE:2,') ');
        WRITELN(LENGTH:4,',',PRIO:2);
        IF CLEARED THEN
          CLENGTH := CLENGTH + LENGTH
        ELSE
          DLENGTH := DLENGTH + LENGTH;
        J := J - 1;
        CURRENT := SUBNTWK;
        END (* outputing *)
      ELSE
        CURRENT := CURRENT@.NEXT;
  UNTIL J = 0;
  WRITELN;
  WRITELN('TOTAL LENGTH: ',(CLENGTH + DLENGTH):5);
  WRITELN('TOTAL CLEARED LENGTH: ',CLENGTH:5);
  WRITELN('TOTAL DEADHEADING: ',DLENGTH:5);
  WRITELN;
END; (* POSTTOUR *)


BEGIN (* Main program SCVRPGM *)
  BULDNTWK;
  FOR P:= 1 TO PL DO
    BEGIN
    CURRENT := HEAD[P];
    WRITELN('SUBNETWORK G(',P:1,')');
    SHOWLIST (SUBNTWK[P]);
    MAKECONN (SUBNTWK[P], P);
    WRITELN('SUBNETWORK G(',P:1,')*');
```

```
    SHOWLIST (SUBNTWK[P]);
    MAKEVEN (SUBNTWK[P], P);
    WRITELN('SUBNETWORK G(',P:1,')**');
    SHOWLIST (SUBNTWK[P]);
    INITIALIZE (SUBNTWK[P]);
    WRITELN;
    WRITELN('**** POSTMAN TOUR FOR PRIORITY ',P:1,' ****');
    POSTTOUR (SUBNTWK[P], P);
    WRITELN;
    END;
END. (* Main program SCVRPGM *)
```