A Mathematical Model for Winter Maintenance Operations Management

by

Mathieu R. Trudel

A thesis
presented to the University of Waterloo
in fulfilment of the
thesis requirement for the degree of
Master of Applied Science
in
Civil Engineering

Waterloo, Ontario, Canada, 2005

Author's Declaration for Electronic Submission of a Thesis

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Scheduling of winter maintenance operations such as plowing or salting is a difficult and complex problem. Proper selection and timing of such operations is critical to their effectiveness, however scheduling decisions must often be made with strict time and resource limitations imposed upon them. A decision support system which analyses current road conditions and makes scheduling suggestions based on them would be a valuable step toward improving the quality of treatment, while simultaneously reducing the burden of scheduling on maintenance managers.

This thesis proposes a real-time scheduling model based on an Operations Research framework that can be used by maintenance managers to develop and evaluate alternative resources allocation plans for winter road maintenance operations. The scheduling model is implemented as an Integer Linear Program and is solved using off-the-shelf software packages. The scheduling model takes into account a wide range of road and weather condition factors such as road network topology, road class, weather forecasts, and contractual service levels, and produces a vehicle dispatch schedule that is optimal with respect to operating costs and quality of service. A number of heuristics are also explored to aid in efficient approximations to this problem.

Acknowledgements

I would like to thank my supervisor, Dr. Liping Fu, for all of his help, guidance, and patience throughout my Masters program. I would also like to thank my fellow students in the Transportation group for being a great bunch of people to work and learn with. Most of all, I would like to thank my family, whose patience, support, and love made this work possible.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Introduction

The degree of complexity in winter maintenance operations may surprise the typical citizen. Where one typically sees only the passing of a plow or spreader vehicle, there is a tremendous amount of planning, careful dispatching, patrol yard management, maintenance, and continual process review that underlies this seemingly low-tech event. A substantial body of research exists that seeks to improve the efficiency and effectiveness of winter road management. Much of this improvement is due to operational changes at the treatment level: new blade and spreading technology, the advent of new salt chemistries and application techniques, and other such advances. However, there is still a great deal of progress to be made in the planning and control of winter maintenance.

Winter maintenance operations consume the majority of road maintenance budgets in most regions that experience a substantial winter season. In Canada alone, it is estimated that the cost of keeping roads clear of snow and ice throughout the winter months exceeds $CDN 1 billion dollars annually[2]. From an environmental perspective, winter maintenance operations result in the spreading of over 5 million tonnes of road salt and other chemicals annually. Environmental damage caused by the runoff of these chemicals from the roadway is significant[23], and can be expensive if not impossible to rehabilitate.

Although modern research into alternate de-icing and anti-icing strategies greatly mitigates this damage by reducing the amount of salt applied to achieve a given level of service, most research in the winter maintenance field focuses on operational concerns such as alternative chemical agents, application rates, and pavement condition factors. Beyond providing improved weather data and vehicle instrumentation for supervisors, little has been done to provide decision support at a supervisory level, where great benefit can be expected.

1.2 Winter Maintenance Management Background

Winter maintenance operations can be broken into three main categories depending on the task being undertaken. These categories are plowing, chemical freeze-point suppressant application ('salting'), and abrasive application ('sanding'). Plowing is used to remove snow from a roadway and deposit it alongside the roadway. Salting is used to treat snow and ice on a roadway through the application of chemicals that lower the freezing point of water and hence cause snow and ice on the roadway to either melt away or reduce their adhesion to the road surface. Sanding operations are undertaken when the temperature is too low for salt or other chemicals to work effectively; sanding is intended to provide increased friction on the roadway thus mitigating the effect of snow and ice until such time as they can be removed by other means.

1.3 Current State of Maintenance Practise

Current practise in the winter maintenance varies between urban and rural settings, and between jurisdictions. In a typical Ontario rural setting, responsibility for clearing various roadways falls upon the jurisdiction that owns the road. In practise, this creates a patchwork pattern of patrol regions, where provincially contracted private operators (and in some cases Provincial Ministry of Transportation managed vehicles) service provincial roads (typically major highways or arterials), and secondary county or municipal roads are serviced by the municipalities concerned. There is typically no explicit or formal co-operation between these governmental bodies, and thus many possible efficiencies are lost. Typically, a rural planning region consists of a subset of the roads within one or two rural counties. This roughly corresponds to ten to twenty roads, and approximately twice as many intersections of these roads.

In urban settings, a city is typically divided into discrete zones which loosely follow the neighbourhood breakdown of the city in question. Major arterials are often included in the planning region regardless of their jurisdiction, as their maintenance is commonly contracted to the municipal maintenance operators by the provincial body that owns them[26]. Although treatment of major routes and the individual neighbourhoods are typically scheduled by the dispatcher, the specific path and order that the constituent streets of a neighbourhood are treated internally is often left to the discretion of the vehicle operator. In short, a manager typically directs a plow operator to service a particular neighbourhood, but leaves it up to the plow operator to decide the best order to treat the roads in that neighbourhood.

1.3.1 Plowing Operational Policies in the Field

As a preliminary, we note that all plowing operations discussed in this work refer to standard displacement style plow vehicles as used by virtually all road maintenance operations in North America. More esoteric treatments such as rotary plows, brooms and blowers are not discussed or considered in this work, though the mathematical analysis would be unchanged procedurally.

Plowing is a relatively straightforward treatment to model, since its effect on snow depth is isolated entirely in the instant that service is applied. Notwithstanding a small amount of debris snow from the plowing operation, we expect that the majority of snow will be cleared to the side of the roadway as a result of plowing. We also expect that the effect of plowing on snow accumulation at any time other than the instant of treatment is zero. In other words, plowing does not affect snow depth in any way other than the actual snow removed during the plowing operation. Note that this does not include effects relating to changes in albedo or other physical factors as a result of the plowing operation; the direct effects of plowing are isolated in the instant of treatment and all other indirect effects proceed from that state. More detail can be found in Section 2.2.1 of this chapter.

Plowing operations in the field are generally driven by the need to meet level of service (LOS) requirements on managed roadways. In most planning regions a given level of service is established for different road types that specifies what maintenance operations must be underway on a roadway by the time a given snow depth on that roadway has been reached. As an example, the Region of Waterloo, Ontario, Canada, uses the LOS criteria outlined in Table 1.1. The Region of Waterloo determines road class as a function of Average Annual Daily Traffic (AADT) and the posted speed limit of the road, with Class 1 roads being the busiest and fastest arterials, and Class 6 roads being extremely low traffic tertiary roads. For exact details, please refer to [6].

Table 1.1: Level of Service criteria for the Region of Waterloo, Ontario, Canada

Class	Snow Depth	Service Time
1	2.5 cm	4 h
2	5 cm	6 h
3	8 cm	8 h
4	8 cm	16 h
5	10 cm	24 h
6	20 cm	72 h

The values in Table 1.1 are used to derive a policy as follows:

1. When maintenance personnel become aware that snow depth has reached the depth listed in Table 1.1 for the given road class, service should be dispatched to the road in question as soon as practicable.

2. After snowfall has ceased, all roads are to be cleared of snow to a depth less than that corresponding to the class of the road as listed in Table 1.1 within the time period listed in that same table.

Note that the above policy does not specify the exact treatment method to be undertaken. However, common practise in maintenance operations (at least with regards to reactive counter-measures such as those required once snow is already on the ground) is to utilise combination plow / spreaders, which plow away the majority of the snow, and deposit chemical agents on the road immediately after plowing, in order to melt away the remaining snow and ice[16, Module 2]. Thus, we can regard any dispatch indicated by the LOS policy above to mean a plowing operation at a minimum, and most likely a combined plow / spreading operation as well.

This LOS-driven plan (or close derivatives of it) are typical of most planning regions studied. Thus, all plowing operations (indeed, all winter maintenance operations) can be viewed as an indirect consequence of having to meet specified LOS requirements.

There are several drawbacks to this type of approach. Since all maintenance operations are undertaken as a consequence of LOS requirements being violated, they are by definition reactive in nature. Thus, adapting an LOS based approach to tasks such as anti-icing that are proactive by nature may be problematic, and is at best an awkward fit. Additionally, LOS based approaches do not explicitly describe what should be done when there is no hope of honouring LOS requirements for all roads (e. g., during a heavy snowstorm that overwhelms the resources available to treat it). In this case, there is typically no explicit treatment plan beyond a best effort based plan, in which resources are continually reassigned as soon as they become available in order to bring conditions back to nominal levels.

1.3.2 Chemical Agent Use Policies in the Field

Chemical Agent usage in winter maintenance generally fall into one of two categories: reactive countermeasures to combat snow and ice already on the roadway, and proactive anti-icing measures, which are applied in advance of precipitation to minimise the impact of the weather event. As these two categories have vastly different purposes and structure, they will be reviewed separately.

Reactive Snow and Ice Removal

Reactive snow and ice removal refers to the process of removing snow and ice that has already fallen on roadways. In cases where any substantial amount of snow has fallen (more than approximately 1 cm), chemical removal is usually paired with conventional plowing by dispersing salt or other chemical agents from the back of a plow vehicle. Chemical application in this capacity is intended to remove whatever snow and ice remains on the roadway after plowing, as well as to prevent ice formation on the roadway from residual moisture. In the case where plowing does not result in bare pavement (as a result of ice or packed snow underlying the removed snow) chemical agents are designed to melt through the remaining material, forming a layer of brine between the remaining material and the roadway. This brine layer makes removal of the residual material and future precipitation easier by breaking the snow/ice-pavement bond, allowing for the material to be dispersed by subsequent plowing or dispersion by vehicular traffic.

Table 1.2: Practical minimum working temperatures for various freeze point suppressant chemicals[16, 23]

Chemical	Practical min. working temperature (°C)	Cost to treat per 2 lane km (1991 $CDN)	Environmental impact
Calcium Chloride ($CaCl_2$)	-31.6	-	low
Sodium Chloride (NaCl)	-9.4	$4.55	moderate
Magnesium Chloride ($MgCl_2$)	-15	$327.00	moderate
Calcium Magnesium Acetate (CMA)	-6	$214.50	moderate
Urea	-10	$58.24	severe

Sodium chloride is by far the most used chemical for this purpose in North America, although there several other chemicals in common use. The use of various chemicals is determined by their effectiveness at various temperatures and their relative cost. These various properties are outlined in Table 1.2.

As previously mentioned, chemical application usually immediately follows plowing, with the particular choice of chemical and application rate dictated by the prevailing weather. Several guides exist to determine which chemical and rate should be used for a given situation, and at least one such guide is typically used as a standard best practise by a planning region. This information is typically presented in a tabular format, where combinations of factors including temperature, precipitation type, and existing road condition are correlated to suggested treatment operations for plowing and salting operations. Some excellent examples of these tables are the Ontario Ministry of Transportation Recommended Treatments table[23], and the US Federal Highway Administration Anti-Icing Program[10]. The tables outlined in these sources are not reproduced here, as they are quite expansive in size.

Proactive Anti-Icing

Proactive anti-icing refers to the act of applying chemical agents to roadway in advance of a weather event, in an effort to mitigate its effect. Specifically anti-icing seeks to minimise the probability of ice formation on a roadway, and to reduce the strength of the snow/ice-pavement bond, which often necessitates heavy salting and repeat plowing of roadways if left untreated. The timing of application of anti-icing chemicals is extremely important to their effectiveness. Before a storm, if chemicals are applied too late they will not reach their maximum effectiveness by the time the storm hits. Likewise, applying chemicals too early may cause them to be dispersed off the roadway by traffic thus going to waste. Temperature, both current and forecast, is also critical in determining which chemical to apply, how soon and how much to apply[16].

Pre-wet salt has become very popular in recent years as it can enlarge the time window during which anti-icing treatments can be applied. Pre-wetting of salt involves mixing the salt to be applied on a roadway with a small amount of water before application. It is intended to hasten the start of chemical action on the roadway, to increase the adhesion of the chemical to the roadway, and to decrease the likelihood of it being dissipated off the roadway by traffic splatter or road splatter (traffic splatter is the effect of vehicular traffic pushing or blowing granular salt off the roadway, and road splatter is the effect of granular salt bouncing off the roadway as it being deposited on the roadway from the spreader vehicle). Standard application rates for pre-wet salt are much lower than for conventional dry application; the Ontario Ministry of Transportation recommends as little as 50 kg of pre-wet salt to be applied per 2 lane km of roadway in light snow, whereas the previous dry chemical minimum rate for such conditions was 130 kg per 2 lane km[18]. Note that pre-wet application of salt is extremely common in proactive anti-icing strategies, and somewhat less so in conventional post-plowing operations (due to the fact that there is often a residual amount of snow on the roadway after a plowing event).

Regardless of the particular chemical and method being used in an anti-icing capacity, the optimal time and type of treatment is a function of many variables including current and forecast temperature and humidity, anticipated precipitation type, road classification, traffic levels, and several other factors. This area is undergoing extensive research by several organisations, and some working systems exist for determining optimal treatment types for a given weather event. The main goal of much of the research in this area is to develop a practical working model of this relationship, which a maintenance operator can use to easily select a treatment option given current weather and road conditions. The Ontario Ministry of Transportation's DART database[21] and the Swedish National Road Administration's Expert System Based Road Salting Prototype[3] each provide an automated, computerised 'black-box' system whereby operators can input the relevant forecast and condition data (in some cases, this data can be obtained automatically), and a treatment suggestion tailored to the specific conditions indicated will be output. Both of these systems, as well as most others in this area, are merely lookup tables of existing data, and as such are not models in the formal sense of the word. This and other shortcomings with these systems are addressed in Section 2.2.2.

1.4 Objectives and Scope of Work

1.4.1 Research Problems

From a supervisory and planning perspective, the modelling of resource allocation is the primary task; an operations manager has a fleet of plows and spreaders of a certain fixed size, and wishes to allocate these vehicles to sections of road to maximise their effectiveness relative to some specific measurement. This task is made all the more difficult by ever-changing weather conditions, traffic levels, and environmental concerns.

We term this task the Winter Road Maintenance Scheduling Problem (WRMSP), and formally define it as the construction of an optimal service vehicle dispatch schedule subject to a variety of constraints on fleet size, route structure, and a variety of other constraints discussed in detail in Chapter 3.

There are also more strategic tasks that must be addressed in winter maintenance, such as year to year funding requests, fleet size and composition choices, and other longer term concerns. Though we do not approach such problems directly in this work, our results can be seen to provide insight into and quantification of the impacts of such choices. However, direct modelling of such problems is left as future work.

The intent of this research is to develop a model for winter maintenance operations which is aware of both the complex factors underlying road condition estimation, and also of the inherent network based nature of road topology. The model is expected to have applications both as a research framework, and as a decision support tool in operational settings.

1.4.2 Thesis Outline

This thesis aims to provide automated solutions to these tasks, primarily in a scheduling and condition prediction capacity. Specifically, we define and demonstrate a model of winter road maintenance operations based on an Operations Research (OR) foundation, and demonstrate various approaches to solving such a model. By taking a mathematically formal approach to this model, we can apply a wealth of existing knowledge in the OR field toward solving the problem.

This model should be solvable using standard mathematical software packages. The model should be flexible with respect to optimisation criteria, input requirements, and road network structure. The model should also be extensible to enable future researchers to use this work as a framework. The ability to mathematically represent factors such as service level guarantees, service cost, fleet sizes, and other parameters is also required. The model must also represent real world operations with reasonable fidelity.

It must be made clear at this point that the intent of this research is not to propose alternate approaches to winter maintenance other than those already suggested. Rather, our intent is to develop a supervisory and scheduling model within which the operations already described can be scheduled in an optimal manner. Accordingly, this model does not directly address lower level operational aspects of treatment, such as recommending treatment rates, plowing speeds, or other such concerns.

In terms of practical deployment concerns, the model should be easily deployable in a production setting. Specifically, an operator-friendly front-end should hide the mathematical complexity of the model from the users. However, the flexibility for researchers to easily alter various parameters of the underlying model should also be present. While the eventual goal of this research is to produce a model which can be used by field practitioners, the model as will be presented here is still very much intended for a researcher audience for the present time.

As with any model, however, an accurate representation of the real-world parameters that the model is supposed to represent is critical. The majority of the preliminary work in this research is dedicated to building a complete picture of winter maintenance operations as they exist today, and particularly of components that will serve as inputs to our model. The latter part of this research is dedicated to the construction and solution of a model that represents these operations. A graphic overview of the inputs and outputs of our model is shown in Figure 1.1.

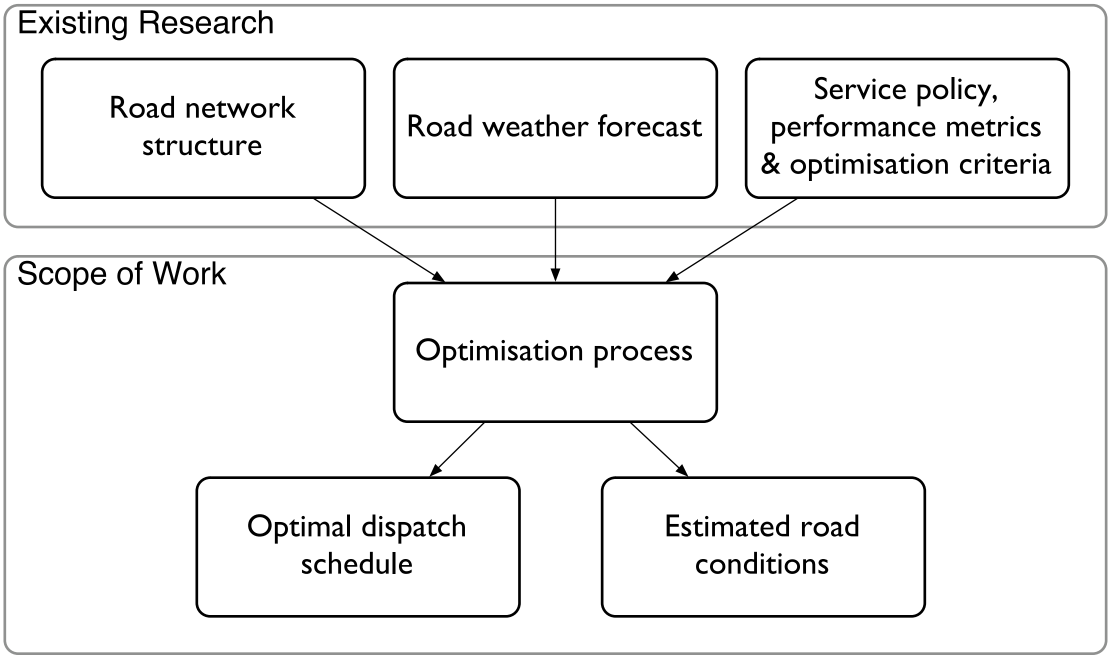

Figure 1.1: Overview of program requirements and outputs

We begin with a comprehensive review of the existing state of practise and technology in winter maintenance, as well as reviewing current and recent relevant research initiatives.

Chapter 2

Existing Research

The work undertaken in this research builds on a number of disparate topics within the broad fields of winter road maintenance and scheduling. As such, research into existing results breaks into several logical segments along the divisions in the 'Existing Research' box of Figure 1.1, all of which are largely distinct from one another. We consider these segments in order, beginning with an overview of current weather forecasting systems.

2.1 Weather Forecasting

Road Weather Information Systems (RWIS) is a general term for automated weather and condition gathering equipment designed with an emphasis on providing data useful for transportation related activities and planning. The technology is deployed as stations that are similar in form and function to conventional automated meteorological stations, although with a set of sensors geared more toward transportation specific measurements than general meteorological monitoring. For example, a typical RWIS station will be located adjacent to a roadway, often in a location that possesses a set of micro-climatic concerns not addressed by conventional weather forecasts, such as along an exposed ridge, at a bridge crossing, or other unique location. In addition to standard meteorological measurement instruments, an RWIS station will typically include sensors to monitor pavement surface temperature and cover, sub-surface temperature, and incoming solar radiation measurements. This data is accessible in real-time by planning and operations personnel, and can be used to assess weather impact on operations and traffic on an ongoing basis. A primary goal of RWIS deployments is to enable automated management systems such as the one discussed in this work. As a result, RWIS data plays a critical role in the data flows of the models described in this work.

The interest in RWIS provided data has increased dramatically in recent years, primarily in response to the increased use of computers and automated technologies in the maintenance field. A survey of winter maintenance operations undertaken in 2000 by the SHRP[25] indicates that there were 134 reported RWIS towers operating in Canada, in 8 of Canada's 10 provinces. As of the 2004 winter season, the number of RWIS stations in Ontario alone reached 112[17] (up from 67 stations as reported in the 2000 SHRP survey). All of the reported RWIS stations use pavement temperature sensors in addition to a conventional set of meteorological instruments. This level of coverage provides a great amount of data to be exploited for winter maintenance operations.

It should be noted that RWIS stations have the greatest impact (and therefore greatest market penetration to date) in regions where weather historically plays a significant role in the management of road systems. In general, the number of RWIS stations tends to be higher in more Northern regions of North America, although there are current deployments of some size in virtually all regions of the continent. As we concern ourselves only with snow-experiencing regions in this work, we therefore assume a reasonable level of coverage from RWIS stations exists in the planning region we seek to model.

2.1.1 RWIS Data Analysis and Integration

Data from RWIS stations expresses an observation at a given, fixed point. In extrapolating this data to cover a wider geographical area, inevitable trade-offs and errors will occur. An analogous condition arises when extrapolating data in a temporal dimension. Unfortunately, these errors are an intrinsic part of the meteorological field, and are not endemic to RWIS data specifically. A robust forecast model will effectively estimate and account for these inevitable errors. The FHWA funded Maintenance Decision Support System (MDSS)[4] has developed an extensive framework to integrate and analyse diverse meteorological data sources, with a specific emphasis toward winter road weather prediction and analysis.

The Weather Prediction Component of the MDSS

The Maintenance Decision Support System (MDSS) is conceptually described in Figure 2.1. The portion of the MDSS relevant to weather forecasting is contained within the Road Weather Forecast System (RWFS) component. Meteorological data and models are obtained from several sources (in the MDSS field test implementation in Iowa, USA, these data sources included NOAA's FSL laboratory, state DOT RWIS networks, and the US National Weather Service). The acquisition of this data is done by the 'Data Ingest' component, and is obtained from the relevant authority via standard Internet protocols such as FTP or the Local Data Manager (LDM) data distribution system[7].

This input information is used to create a set of forecasts at various locations and times in the future, using a set of different prediction algorithms. Each of these algorithms is encapsulated within a Forecast Module (FM), which operates on a given set of input data. The output of each of these modules is a time series weather forecast for a given location. A consensus of the forecast module outputs is built by the Integrator component, which outputs a confidence weighted average of the Forecast Modules. The output of the Integrator component is then fed through a post-process module, which ensures that the data meets some basic sanity measures, such as ensuring that percentage values are in the range of 0 to 100. The post processor also performs some interpolations on the forecast data to fill in gaps in the forecast that are required to be present by the consumers of the final forecast data.

Accurate weather forecasts are critical to the success of any winter road maintenance program, and form a necessary input for our research. The current MDSS project is the culmination of three years of work by 6 major US research labs, and represents the current state of the art in road weather forecasting. The software that comprises the MDSS is freely available (with the exception of one forecast module for which a license is required) and has been designed to be readily portable to different computer operating systems and data sets. Recognising the benefit to be obtained by a maintenance operator by having accurate weather predictions, and that accurate weather prediction is a strict pre-requisite of deploying the scheduling system discussed in this work, it is reasonable to expect that a maintenance operator interested in deploying this scheduling system will have a weather prediction system of the scope and quality similar to that of the MDSS.

2.2 Treatment Effectiveness Models

Having seen how maintenance operations are carried out in practise and how weather information is gathered, we turn to analysing how effective these treatment operations are in given weather situations.

Surveying the field of existing research in the area of treatment effectiveness, many models of effectiveness exist that are inapplicable for a variety of reasons. Typically, there are gaps in the study methodology used (for example, Keyser[11] noted the existence of time varying measures of road friction as a result of various chemical applications, but did not control for temperature or other critical variables).

Any treatment effectiveness model must quantify effectiveness with respect to a specific metric. For example, when considering plowing operations, the average depth of snow on a section of road would be a suitable candidate metric, as it reflects the main purpose and effect of plowing. The selection of which metric to use to evaluate a given maintenance operation on the network is critical; selecting a metric that does not capture the actual effect of the operation introduces both a smaller measurement range for the effect of the treatment, but also has the potential of allowing secondary causes into the effect measurement which may not be captured by the model. For example, if we were to measure the effectiveness of salting operations by the observed pavement temperature, secondary causes such as weather changes or diurnal variation could potentially skew our observations of treatment effectiveness. In addition, altering pavement temperature is not the primary use of chemical application, and so we will not have a particularly accurate metric to measure treatment effectiveness. For this reason, it is imperative to select a performance metric that is directly linked to the treatment being performed.

We undertake a survey of existing treatment effectiveness models for plowing in order to obtain a soundly researched gauge for the effectiveness of a given plowing schedule.

2.2.1 Plowing Effectiveness Models

Plowing effectiveness models seek to relate the operation of plowing with both the immediate and ongoing changes to road condition as a result of the plowing operation. The immediate effects on road condition are obvious – as a plow passes an area of a road, snow is removed down to a certain depth determined by road and plow geometry, and the composition of the snow and ice overlying the roadway. The ongoing effects are a direct result of this change in snow depth; if pavement (particularly darker asphalt) is uncovered as a result of plowing, then solar heating and environmental exposure of this surface will affect pavement temperature, and subsequently the trend and rate of snow depth on the roadway.

A number of studies have been undertaken to improve the physical composition and geometry of plow blades in an effort to improve their effectiveness and reduce their maintenance costs. One of the largest recent studies in this area was undertaken as part of the US Strategic Highway Research Project[19], which evaluated potential improvements in the classic displacement plow blade design. Unfortunately, this study, while exhaustive in the areas it was focused on, does not provide quantitative evidence of the amount of snow left behind on a roadway for a given set of parameters.

Regarding the ongoing effects of plowing on roadway conditions, we turn to a first-principle energy balance model of snow covered surface temperatures, as described in[9]. The model described in this work, termed SNTHERM, is a energy balance model that predicts temperatures at arbitrary depths within a gradient of snow, pavement, and soil. This model is produced from a physicist's point of view, and as such its internal details are far beyond the scope of this review. The algorithm itself is implemented in FORTRAN, and is suitable for use as a 'black-box' predictor of pavement temperature given a suitable set of conditional parameters. SNTHERM is used as the thermal prediction model for surface and subsurface temperatures in the MDSS project, which is described in greater detail in Section 2.5.2 on page 21.

2.2.2 Chemical Agent Effectiveness Research

Chemical agent treatment refers to methods of winter road treatment that involve the application of a free-point suppressant or other chemical to the roadway. Removal of snow and ice is thus affected through a chemical process. Chemical agent effectiveness models seek to model the effects of such treatment on an ongoing basis. Such a model is extremely critical in evaluating the effectiveness of various de-icing and anti-icing policies.

As discussed previously, anti-icing strategies involve the application of chemicals to a roadway in advance of the storm, in order to prevent an ice-pavement bond from forming. However, due to the chemical properties of the agents, and the fact that the applied chemicals are left sitting inactive on a roadway before the precipitation begins, anti-icing strategies are considerably more complicated to model than simple plowing. The relationship between the time of treatment and the change in condition due to treatment varies according to a number of external parameters, and is not constrained to an instantaneous change; where plowing has a single, instant and immediate effect on the road condition, chemical treatment has an varying, ongoing, and possibly delayed effect. These properties make it vastly more difficult to represent within our framework.

In terms of quantifying the effect of treatment on the road condition, we require a model that takes as input any number of situational factors such as vehicle or equipment type, chemical agent applied, pavement temperature and time of application, and provides an estimate of the road condition as a result of treatment for a given window into the future. This output is used as a metric to compare the effectiveness of two different treatment strategies, and may represent any convenient quantity or measure, depending on the parameters we are attempting to optimise. For example, the MTO's De-icing/Anti-icing Response Treatment (DART) database uses the percentage of bare pavement exposed along a given transverse section of road as the measure of a treatment's effectiveness[21]. Later refinements to the DART database considered the change in the average coefficient of friction along a section of road as the representative metric of a treatment's effectiveness. Again, the exact measure of effectiveness used is largely unimportant, so long as it accurately and reliably reflects the desired impact of treatment.

Several sources come close to filling our needs with respect to a comprehensive chemical effectiveness model, most notably the Chemical Concentration Model component of the MDSS system[5], the data tables used by the Aurora Project's DART database[21], and a research report undertaken as part of the Strategic Highway Research Program in 1994[1]. A brief overview of these sources follows.

The MDSS Chemical Concentration Model

The concentration of a freeze-suppressant chemical on a roadway is an essential factor in its effectiveness. After application, the concentration of a chemical on the roadway is affected by several factors, most notably: runoff, evaporation, vehicle dispersion (where vehicular traffic disperses the material off the roadway), and continued snowfall (which serves to dilute the chemical already on the roadway). The modelling of these various effects over time will determine the amount of chemical left on a roadway in the future, and thus how effective the initial treatment will still be. Compounding this problem, however, is the fact that previous treatment, or even the effectiveness of the current treatment in previous time intervals, changes the amount of precipitation on the roadway, and thus also its temperature and environmental exposure. Thus, evaluating the chemical concentration on a roadway over time is considerably more complicated than may it may seem at first, as many factors are at play in determining the concentration of the chemical on the roadway at any given time, including precipitation, liquid melt and runoff, vehicular dispersion, and other factors. The MDSS Chemical Concentration Model[5] seeks to overcome this difficulty by approaching the problem in a time-wise iterative order, considering each individual time interval in increasing order, and using results from each interval as inputs into subsequent interval predictions. The required inputs of this model (other than initial calibration values) are modest; only initial application amounts, precipitation depth, pavement temperature (as provided by the SNTERM model) and predicted traffic levels are required to produce an estimate of chemical concentration.

The iterative nature of this model produces several useful data points at each time interval, most notably pavement temperature and precipitation depth. Due to the integrated approach of the MDSS model (as described in Section 2.5.2 on page 21), these predictions are automatically merged with predicted and actual salting and plowing events as part of the model's structure, and are automatically determined in a flexible manner according to local policy.

The DART Database

The DART program was a project undertaken by the MTO to construct a decision support system for winter road maintenance. The goal of the program was to "develop a database that can be used to quantify and compare the effectiveness of snow removal using alternative chemicals and methods under specified environmental conditions"[21, p. 2]. The database records a number of different chemical agents, applied over a range of environmental conditions and closely monitored to evaluate their effectiveness. As part of the monitoring of each application, a record was made of the change in pavement snow coverage at 30, 60 and 120 minutes after each application. The overall melting rate, describing the rate of change in the percentage of road width covered in snow, was also determined, and is expressed in metres of pavement width per hour.

The DART database therefore correlates a large set of chemical agent and environmental condition combinations with expected melting rates they produce. This information can be used to derive a lookup table of operational guidelines, however, its structure is inherently query/response driven, and it does not immediately provide a mathematically expressible prediction function. As such, it is of limited use in construction of our model, though it could be used to validate treatment suggestions produced by our model by comparing them against the set of historical data which comprises the DART database.

The SHRP Anti-Icing Report

The SHRP initiative produced a large number of reports in the early to mid 1990s that re-evaluated many long-held practises and beliefs from elementary principles. Among these reports, a number concerned themselves with evaluating chemical anti-icing treatment methods, and one in particular entitled 'Anti-Icing Study: Controlled Chemical Treatments'[1] proves relevant to this work. This study examined the minimum application levels of various chemicals required to maintain service levels under a variety of meteorological conditions. The study was done in a controlled environment (an unused airport runway) and was undertaken using a relatively strict and contrived process in the name of study control. As such, the study succeeded in its mandate of carrying out an independent and controlled study of anti-icing chemical effectiveness, but does not provide any directly usable effectiveness model for our work. The data produced by this study could potentially be used to derive an effectiveness model such as the one we require, however this is left as future work.

2.2.3 Remarks on Existing Models

As has been seen in this section, the majority of the research to date in the field of chemical treatment falls short of our requirements, typically either because they answer the wrong questions (many studies exist which suggest the best course of action to take in a given situation, but not the specific time-varying effects a given course of action will have), or because they are somehow otherwise inapplicable to our work (for example, most data sheets produced by chemical suppliers consider only the supplier's own chemicals in their studies, and so an objective comparison between two chemicals is not readily available for many scenarios).

Unfortunately, most models existing in the literature present a shortcoming in that they fill a slightly different need than ours. In particular, where models simply recommend a given course of action over another for a given situation, we need a stronger result in that we require a time-varying impact estimate for a given treatment action in a given situation. The small number of such predictive models is identified as a primary missing piece in the overall real-time road maintenance management puzzle.

Notwithstanding this lack of predictive results, microscopic and/or reactive treatment effectiveness models can serve a vital role in the construction of future predictive models. In particular, since we will demonstrate that our optimisation model is largely impartial to the exact treatment model used, future advances in this area can easily be integrated with the framework.

2.3 Vehicle Route Design

The layout and design of service routes within a given planning region depends on many external factors that are not easily quantifiable. Among these factors are the following:

- Regional or district boundaries may necessitate special considerations that are not easily quantifiable. Policy or political concerns may dictate that neighbourhoods be treated in a specific order, or that individual neighbourhoods be treated independently of one another.
- Public transit is often afforded special treatment with respect to winter road maintenance. In most municipalities, public transit routes are given priority for winter maintenance operations in order to ensure the safety and expediency of public transit vehicles.
- Regions with a large number of routes used for commercial transportation may give priority to such routes, particularly in areas where high commercial volumes are carried or where a large amount of on/off-loading occurs. Delays caused either directly by weather conditions or by a weather related accident may have a large negative effect in such areas, and so effective treatment of such routes is preferred.

It is common practise for maintenance operations within an urban setting to be divided into neighbourhood sized sections, each of which is planned for and managed as a discrete unit. For example, the City of Waterloo divides the city of Waterloo (population approximately 100,000) into 17 service zones, each roughly corresponding to a neighbourhood, and each approximately 2-3 km sq[26]. Each of these service zones is divided into 3 subzones, which are used to vary the order in which individual streets are treated in subsequent storms. For example, in one storm the subzones of a particular zone may be treated in the order A, B, C, while in the next storm the subzones may follow the order C, B, A. This ensures that any particular street within a neighbourhood is never the first or last street serviced during every storm event. Current practise within the city is to dispatch service vehicles at the subzone level only. Once dispatched to a subzone, the vehicle operator is allotted a rough time interval to service the subzone and decides on their own in what order to treat the residential streets that comprise the subzone. The arterials which connect the secondary residential areas are commonly serviced by a mix of the service vehicles transiting these arterials on their way to residential neighbourhoods, and dedicated service vehicles which travel pre-defined routes through the arterial network.

Rural settings tend to have fewer roads, but individual roads are both longer and more prone to snow drifting than those in an urban setting[22]. In addition, jurisdictional concerns often dictate that a given maintenance authority is responsible for only a subset of the roads in a given region (for example, provincial maintenance operators are only responsible for the maintenance of provincial highways and not county or township roads). This creates a patchwork effect of responsibilities and jurisdictions, and complicates modelling of the network, as certain roads can be used for movement of service vehicles, but are not required to be serviced. Thus, the allowance of dead-heading and non-service legs within service routes is a critical concern in effectively modelling rural maintenance activities.

2.4 Vehicle and Crew Assignment

Once a dispatch schedule is constructed that honours the fleet size restrictions placed on the given problem instance, we are still faced with the problem of assigning specific vehicles and operators to each dispatch event. Although we know with certainty that such an assignment must exist (since we have a dispatch schedule that never requires more than the possible number of vehicles on the road at a given time), we do not know what specific assignment we are to use. Abstractly, this assignment task reduces directly to the *assignment problem* in Operations Research. Techniques for solving the assignment problem, subject to a number of different criteria, are well known in the OR community. See [8, sec. 4.4] for more details on the assignment problem and how it is formulated and solved.

We will not cover details of specific vehicle or operator assignment here. The end result of our research is a simple deployment schedule formulated without concern or respect to individual vehicles, other than to ensure that fleet size restrictions are always honoured.

2.5 Existing Decision Support Systems

Having examined many of the composite factors of a winter maintenance program, we conclude with an overview of existing decision support systems, each of which takes all or some of the components discussed previously, and combine these into a consolidated system geared toward a specific support or management purpose. We review two principal decision support systems here, the Ontario Ministry of Transportation's DART Program[20], and the Maintenance Decision Support System[4] funded by the US Federal Highway Administration (FHWA).

2.5.1 The DART Decision Support System

The DART Program is a project undertaken by the Ontario Ministry of Transportation, as part of the Aurora Program[24]. The DART Program uses a computerised database of recommended treatment plans for a number of given weather conditions, as established by FHWA guidelines. Users of the system input the current weather conditions and other operational parameters, and the program presents them with a recommended treatment plan based on this information[20]. In this sense, the DART Program is nothing more than an automated lookup table, and does not provide any high level integration with route scheduling considerations, adaptive anti-icing strategies, or other advanced technologies. As such, its suitability to this research is marginal.

2.5.2 The MDSS Decision Support System

The structure of the MDSS is detailed in Figure 2.1 on page 24. We have seen how the structure of the Road Weather Forecast System works in Section 2.1.1, and so we now turn to the structure of the remainder of the MDSS system, namely of the Road Condition and Treatment Module (RCTM). This module is responsible for taking forecast information, road network information, and local maintenance policy information and forming a prediction of road conditions based on a candidate road treatment. It accomplishes this through the following process, which is run iteratively for each road segment in the network:

1. Forecast data presented to the RCTM at point (1) on Figure 2.1 is combined with chemical concentration data obtained as described in Section 2.2.2. This information is used to build a 'mobility index' for each road segment for a given treatment scenario (initially, this treatment scenario is empty).

2. If the mobility index values for the given treatment scenario violate the local treatment policy (as determined by the Rules of Practice Module), a treatment is indicated at the correct time and road location, and this information is passed to the Chemical Concentration Model at point (2) on Figure 2.1, as well as being indicated to the user as a suggested treatment.

3. The Chemical Concentration Model constructs an updated estimate of chemical concentrations as a result of the newly suggested treatment, and passes this information back to the Road Temperature and Snow Depth Model. This process begins anew, starting just after the time of treatment indicated in this step.

The MDSS presents as output to the user a view of current and forecast weather information, as well as a list of suggested treatments and their anticipated effects. From an implementation point of view, the MDSS is constructed in a modular, extensible way. The MDSS is divided into a server component, which runs on Linux and is comprised of numerous modules that communicate through well defined means, and a client component written in portable Java that presents forecast data and suggested treatment data in a graphical interface to the user. The majority of MDSS is available under an Open Source license (certain components of MDSS were developed under funding models that do not allow for free distribution of their source code, although binary packages are available).

There are several shortcomings of the MDSS, which are outlined here. Foremost among them is the absence of a method to deal with the interconnected nature of roadway networks, and to account for routed treatment events in the proposal of a treatment event. Treatment suggestions are given on a road by road basis, and co-ordinating such events between road segments is not facilitated by the MDSS. Additionally, treatments are suggested in a greedy manner, meaning that the possible effects of moving the time of treatment up or back to free up resources later is not taken into account. In short, the MDSS does not have a particularly high-level view of a operational system, and does not take higher level constraints such as fleet size, route topology, and alternative treatment goals into account. That being said, the MDSS serves as an excellent framework for future work.

It is important to note that MDSS is currently a prototype system, and is not intended for real-world deployments at the present time. The Aurora Program[24] is beginning an initiative to bring the MDSS into production as of 2004, however no results are yet available.

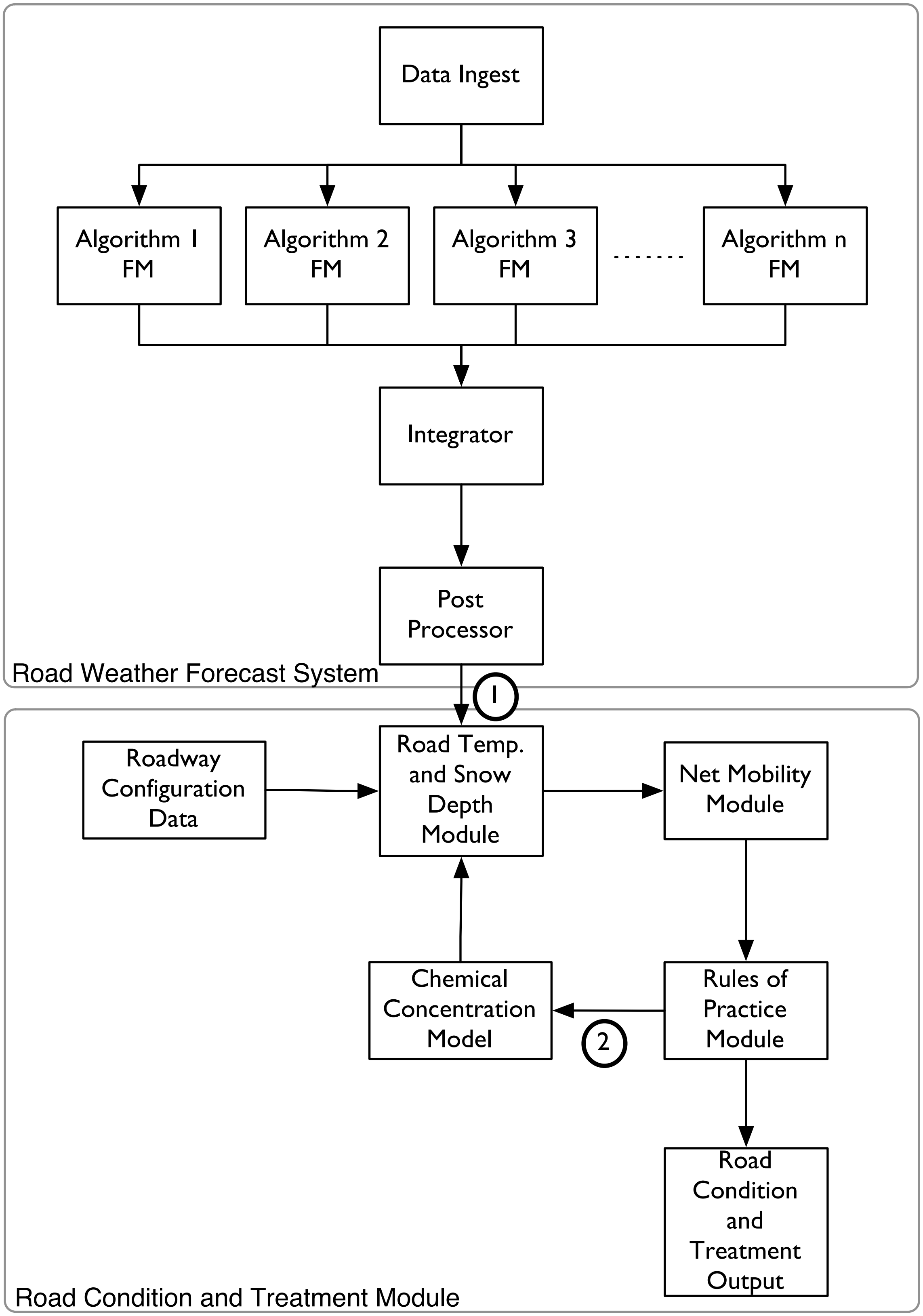

Figure 2.1: Structure of the MDSS

Chapter 3

A Mathematical Approach

3.1 Maintenance Planning as an Optimisation Model

The WRMSP seeks to find an optimal schedule for the deployment of winter road maintenance vehicles subject to a number of constraints, which model level of service requirements, fleet size restrictions, and route choices. Whereas existing practise is based on an almost entirely empirical and subjective approach, the approach described in this section approaches the WRMSP as a formal optimisation model, and as such brings to bear on the problem a wealth of mathematical insight.

In casting the WRMSP in a mathematically rigorous light, it is necessary to formalise some of the parameters and expectations of the problem and to restrict others. In particular, the approach described in this section assumes a single patrol yard, a fixed fleet size, and a set of pre-determined routes each of which begins and ends at the patrol yard. The model also considers treatment types such as plowing, salting, or combined plowing and salting as separately scheduled entities. The WRMSP does not consider multiple treatment events in its formulation, and so any discussion to treatment events within this discussion should be taken to mean a single and unchanging type of treatment.

It is also important to note that the research presented in this chapter directly considers only the operation of plowing, and does not present a model for other treatment types. The primary reason for this is to enable a clearer presentation of the model structure, although lack of adequate existing research in this area is a secondary reason. Thus, although the material in the previous chapter may suggest the presence of a chemical treatment based model in this research, the development of such a model is left as future work.

3.2 Input Data

We proceed to build the model up from its composite parts. As most of the data we will be managing is in the form of matrices, we first define the sets that will form the indices of these matrices.

Road Segments We decompose the road network into a set of road segments. Road segments represent a section of road that functions as a single element with respect to routing. Where indices are made with respect to road segments, they take the symbol i, and are indexed from 0 to $n-1$, where n is the total number of road segments in the model. Whether road segments are considered to be unidirectional or bidirectional depends on the particular method of treatment along the road segment. If the treatment being considered affects the entire width of the road segment in a single pass regardless of direction (such as plowing a narrow secondary rural road), then the physical road segment can be modelled as a single bidirectional segment. Conversely, if the treatment being considered must be applied to each lane, then multiple unidirectional road segments should be used to model the physical road segment. Roadways with multiple lanes are modelled in a similar fashion, with each lane requiring separate treatment being modelled as a distinct road segment in the model.

Routes Routes are ordered lists of road segments, and represent the path that a service vehicle may take through a road network. It is assumed that all routes are cyclic, and all originate from a single, central point. Where indices are made with respect to time periods, they take the symbol k, and are indexed from 0 to $m-1$, where m is the total number of routes considered in the model.

Time Periods The time period modelled by this program is broken into a number of discrete time periods. These time periods are the atomic unit of time for scheduling purposes. Where indices are made with respect to time periods, they take the root symbol t, and are indexed from 0 to $T-1$, where T is the total number of time periods considered in the model. Where instantaneous events are discretised to a specific time period, they are assumed to occur at the beginning of the time interval.

Next, we enumerate the input values for the problem. These are values that are fixed for a given solution of the model, and represent factors that do not or cannot be varied within a given run of the model.

$\boldsymbol{l}_i$ Represents the duration of time needed to traverse segment i, expressed in time periods (rounded up to the nearest integer). Note that while this approach is convenient for modelling purposes, a more conventional approach is to denote the length as a simple time interval. However, the conversion between the two forms is straightforward.

$\boldsymbol{d}_k$ Represents the length of route k, expressed in time periods required to traverse it (rounded up to the nearest integer). This value can be computed directly from $\boldsymbol{l}_i$ by summing over all road segments that are part of a given route k. As will be seen shortly, the value of $\boldsymbol{d}_k$ actually represents the monetary cost of servicing route k. Although it would be more realistic to measure this cost directly in monetary terms, this work assumes that the monetary cost varies directly with the amount of time required to service the route, hence the formulation given here.

$\boldsymbol{\lambda}_i$ Represents a weight factor ascribing an 'importance' value to road segment i. Its value is dimensionless and serves a completely ordinal capacity (that is, the actual values of $\boldsymbol{\lambda}_i$ do not matter, merely their relationship to one another). This allows the model to consider various classes of roads, and to differentiate between them for the purposes of establishing service priorities.

$\boldsymbol{\phi}_i$ Represents a level of service threshold for road segment i that must be maintained at all times. While the formulation in this paper uses a common threshold value for all time periods, this is not required by the model. Thus, the threshold could easily be made to vary by time of day as well as by road segment by indexing into a fixed value matrix of threshold values.

$\boldsymbol{w}_i$ Represents the absolute weather condition on road segment i at the beginning of time period 0.

$W_{i,t}$ Represents the relative change in road condition due to weather for road segment i during time interval t. These values are expected to be obtained from a weather model that black boxes these predictions. Note that these values represent changes for each time period relative to the previous time period, and are not absolute.

$P_{k,t}$ Represents a table with integer elements denoting the number of vehicles that have already been dispatched to depart on route k at the beginning of time t. These vehicles are not available for scheduling (their deployment is fixed), but they nonetheless count toward fleet size restrictions.

b Represents an integer bound on the size of the available fleet of vehicles. This represents the maximum number of vehicles which can be in service at a given time.

$R_{k,i}$ Represents a table with integer elements denoting the order of road segments in a specific route. The entries of this matrix take on the ordinal value of the road segment i's position in route k. Road segments that do not appear in a route are denoted by the value 0. For example, if a route denoted by A contained the road segments $5, 3, 6, 1$ in that order, then the row corresponding to route A would read $4, 0, 2, 0, 1, 3, \ldots$.

3.3 Decision Variables, Intermediate Variables and Objective Function

We must also define the variables we are intending to optimise. We define two sets of objective variables, as follows.

$X_{k,t}$ Represents a binary decision variable matrix. Its entry values are set to 1 if and only if a service vehicle is to depart on route k at the beginning of time t, and 0 otherwise.

$Y_{i,t}$ Represents the number of service vehicles on a road segment i during time t. These are intermediate variables with values depending on the objective values in $X_{k,t}$. The mapping from $X_{k,t}$ to $Y_{i,t}$ is covered in Section 3.4.

$Y'_{i,t}$ Represents the number of service events on a road segment i during time t. The difference between these values and those of $Y_{i,t}$ is subtle but critical; the values in $Y_{i,t}$ describe the number of vehicles at each location and time, whereas the values in $Y'_{i,t}$ describe when treatment events are assumed to take place. The need for such a distinction becomes clear when one considers the example of a long road segment. A treatment vehicle may spend several time periods traversing such a long route, which will be reflected in the values of $Y_{i,t}$. However, it is not correct to consider the road segment as being actively treated for all time periods a service vehicle is on it, since this will result in the section at the beginning of the road segment to still be treated long after the service vehicle has passed, and the section at the end of the road segment to be treated before the vehicle has passed. This makes it necessary to select a single time period to use as an approximation of when the segment as a whole is treated. As with $Y_{i,t}$, these are intermediate variables with values depending on the objective values in $X_{k,t}$. The derivation of $Y'_{i,t}$ is covered in Section 3.4.2.

$C_{i,t}$ Represents the condition of road segment i at the end of time t. These are intermediate variables with values depending on the condition level in the previous time interval $C_{i,t-1}$, any weather related changes $W_{i,t}$, and any changes due to service events $Y'_{i,t}$. The specific formulation of $C_{i,t}$ changes depending on the type of treatment being modelled, since the effect of treatment events $Y'_{i,t}$ are different for every type of treatment. This relationship is discussed further in Section 3.4.3.

As in most practical OR models, there are a number of possible choices for the objective function. For our purposes, we choose a weighted objective function that seeks to minimise treatment cost subject to level of service constraints (by minimising the number of nonzero values of X), and adverse road conditions subject to fleet size constraints (by minimising the sum of condition values in C). This balance of the two objective functions is accomplished by the weighting factor β.

We can now state our objective function as:

$$\text{minimise} \qquad \sum_{i=0}^{n-1}\sum_{t=0}^{T-1} \boldsymbol{\lambda}_i C_{i,t} \quad + \quad \beta \sum_{t=0}^{T-1}\sum_{k=0}^{m-1} \boldsymbol{d}_k X_{k,t} \tag{3.1}$$

where $\boldsymbol{d}_k$ represents the cost of servicing route k. In our implementation, $\boldsymbol{d}_k$ is equal to the sum of the time required to service each of the road segments that comprise route k.

Note that the length of a road segment i does not contribute to the summation of road conditions. In other words, a short road segment is assumed to contribute the same cost to the objective function as a long road segment with an equivalent condition value, even though the overall lane distance effected may be very different. This may be overcome by adding a multiplicative factor of $\boldsymbol{l}_i$ alongside the $\boldsymbol{\lambda}_i$ factor present in the above function.

It should also be mentioned that if values for the direct monetary cost of servicing a given route or road segment are available, the values of $\boldsymbol{d}_k$ in the second summation of Equation 3.1 could be replaced with such values, allowing for a direct expression of the cost of providing services based on real-world values instead of basing such costs on the length of time required to service the route in question.

The values of the two summations in Equation 3.1 represent different units of measure; the first equation represents the overall impact of the storm as the sum of the condition values of the road segments, while the second summation represents the monetary cost of providing service. Establishing a relationship between these summations is required to weigh them against one another, and this is accomplished through the value of β. The value of β can be thought of both as a weighting factor used to establish priority between the two summations, and also as a unit conversion factor to bring the two summations into a common measurement unit, most likely monetary cost. The task of establishing this relationship is substantial, as it requires ascribing a monetary cost to road condition, which may be a product of several factors such as the economic impact of road conditions, the monetary value associated with increased accident risk, and other factors. Moreover, as the unit value of the entries in C depends on the effectiveness measure used, this analysis would have to be carried out for each different type of treatment electiveness measure. Such work also depends largely on the policy and monetary constraints of the planning region being considered. As such, further investigation of this comparison is left as future work.

3.4 Deriving Location Information from Departure Information

We have already seen that one of our main objective variables is a departure schedule. We must now relate this departure schedule to actual treatment events on the roads that comprise the network. We have seen this requirement earlier, embodied in the value of Y', though we have not seen how this value is to be obtained. Indeed, the derivation of this function is the central and most novel part of this work. No known pedigree for this formulation exists in the literature, although its usefulness spreads far beyond winter road maintenance.

The principal difficulty in determining the effect of a deployment schedule from a central depot is in correlating the effects of a given route departure to the treatment times of roads in the network. A given road segment may be part of many routes, and will typically only be reached after a known delay from the departure time along the route. We need to construct a function that can take as input a set of route departures, determine all of the effects of those departures, and output how many service vehicles will be on each road segment in each time interval. Figure 3.1 illustrates this transformation.

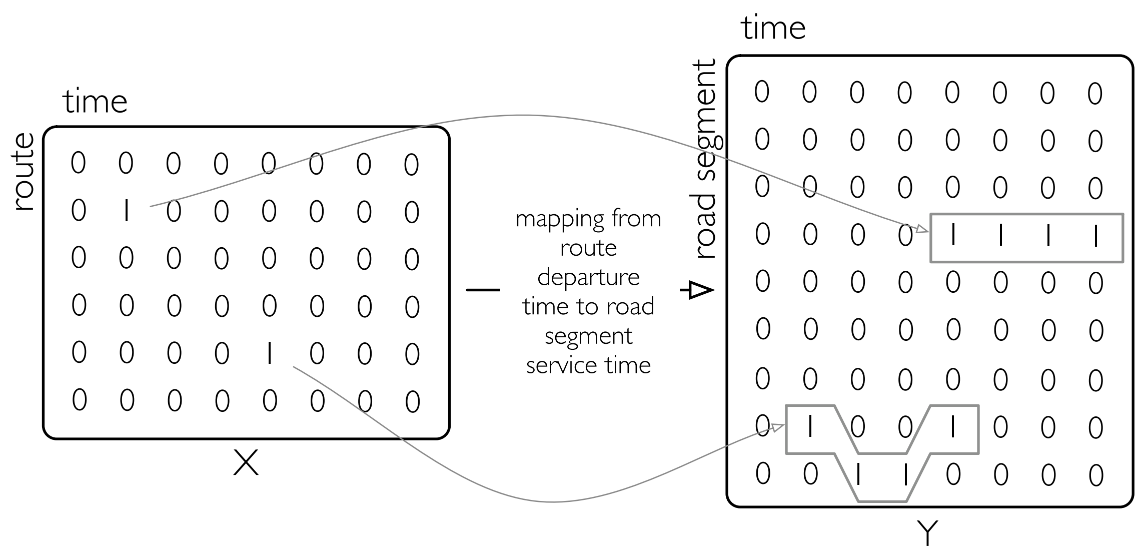

Figure 3.1: Mapping from route departures to service times

To accomplish this translation, we use a 4-dimensional binary matrix Q, which we derive directly from $\boldsymbol{l}$ and R. Q contains entries of the form $Q_{k,i,t,t'}$, where k,i and t all range over their standard domains, and t' ranges over the same domain as t. It is easiest to understand the use of Q by considering the entries as 2-dimensional 'slices' formed by fixing k and i values. For a fixed k and i, we have an array slice $Q_{k,i}$ that is indexed by t and t'. Entries of this array are set to 1 if and only if a departure of a vehicle on route k at time t' will result in a vehicle on road segment i at time t. For example, if we consider a road segment i that takes 3 time intervals to traverse, and that is reached on route k at the start of the second time interval after a vehicle departs on route k, then $Q_{k,i}$ looks like Figure 3.2.

As noted, Q is obtained directly from $\boldsymbol{l}$ and R. In the sample implementation of the WRMSP, this is accomplished by defining Q as a derived variable, using the following function to define its values automatically. Formally, the function used to define the entries in Q is as follows:

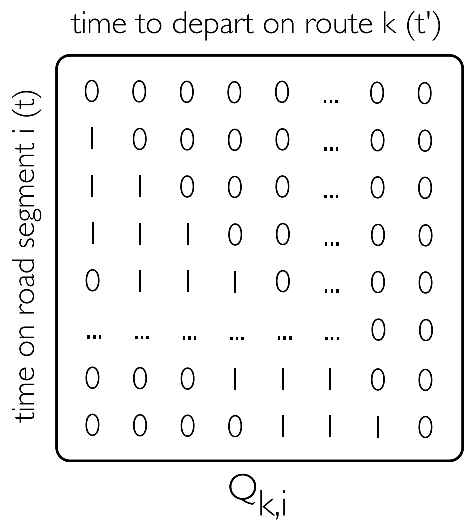

Figure 3.2: Sample slice of Q

$$Q_{k,i,t,t'} = \begin{cases} 1 & \text{if } (t \geq t' + \sum\limits_{\forall i' : R_{k,i'} > 0 \text{ and } R_{k,i'} \leq R_{k,i} - 1} \boldsymbol{l}_{i'}) \text{ and } (t < t' + \sum\limits_{\forall i' : R_{k,i'} > 0 \text{ and } R_{k,i'} \leq R_{k,i}} \boldsymbol{l}_{i'}) \\ 0 & \text{otherwise} \end{cases} \tag{3.2}$$

3.4.1 Using Q to Relate X and Y

Having Q at hand, we need to use it to relate X to Y, as that is the main purpose behind Q's existence. Recall that X is a 2-dimensional matrix of values $X_{k,t}$, where a value is set to 1 if and only if a vehicle departs on route k at time t. Thus, a single row of X represents all of the departures of vehicles on the corresponding route, laid out in time increasing order. Notice also that the '1' entries in a row of a slice of Q represent the time intervals at which a vehicle would have had to have left the service yard on the corresponding route to be on the corresponding road segment at the corresponding time. Thus, the dot product of these two vectors is exactly the number of vehicles from route k that will be on road segment i at time t. Thus, for a fixed k and i, the matrix product

$$Z_{k,i} = Q_{k,i}(X_k^T + P_k^T) \tag{3.3}$$

where Z is a 3-dimensional matrix whose entries $Z_{k,i,t}$ represent the number of vehicles on road segment i at time t due to route k. Note that the T superscript above refers to the transpose matrix operation, and not to the number of time periods in the model. Note also that the inclusion of P here allows us to include pre-determined vehicle departures into the program. A simple summation of the form

$$Y_{i,t} = \sum_{k=0}^{m-1} Z_{k,i,t} \tag{3.4}$$

produces the values of Y as a function of X, P, $\boldsymbol{l}$, and R, as required.

3.4.2 Treatment Time Modelling

The values of Y are crucial to derive and maintain fleet sizing restrictions. However, to correctly model the treatment of a road segment, we need to assume a single time of treatment for an entire road segment. This is due to the fact that a single service vehicle can only be at one place on a road segment at a time. Thus, if we consider the road as being 'treated' at all times that a vehicle is on it, we end up having the first physical section of a road segment being considered as 'treated' when the service vehicle is at the other end of the road segment. Likewise, we may get ahead of actual fact when considering the time of treatment for the very last physical sections of a road segment. Thus, we have to select a single point in time, and use that as an approximation of when the road segment as a whole is treated. Notice that if we use Y directly to model treatment times, we do not satisfy this condition, and so an alternate approach is needed.

For this purpose, we derive a table Y' which we will use for treatment time decisions. This table is based directly on S, a modified version of Q in which only one entry per column in a slice of S is set to 1. This means that in the resultant table Y', for every vehicle service event on a road segment, only one entry of Y' is set to 1, and the corresponding time interval is used as the representative treatment time for this service event on this road segment. Specifically, entries in Y' are set to 1 if and only if they represent the first time interval during which a vehicle services a road segment.

As an alternative explanation of the distinction between Y and Y', note that entries in Y' represent the time interval in which a vehicle is on a road segment as the discrete moment when this interval begins. While it may appear that this is a shortfall of this model, the analysis indicates quite the opposite. If one regards the treatment of a segment to take place in a discrete instant (the instant when in actuality treatment is being applied to the very beginning of the segment), then the treatment of the end of the segment will take place a known and constant time after this instant. Moreover, when considering road conditions between subsequent treatments of a segment, the interval between these discrete treatments will correspond exactly to the interval between treatment times for any subsection of the road segment. In other words, if 3.5 hours elapse between treatment events (which we correlate with the moment of treatment of the beginning of the road segment) of a road segment which is 25 minutes long, then exactly 3.5 hours will elapse between treatment events at the very end of the road segment as well, shifted 25 minutes after the treatment events on the beginning of the road segment.

The derivation of S proceeds from that of Q as follows:

$$S_{k,i,t,t'} = \begin{cases} 1 & \text{if } Q_{k,i,t,t'} = 1 \text{ and } \sum\limits_{u=0}^{t} Q_{k,i,u,t'} = 1 \\ 0 & \text{otherwise} \end{cases} \tag{3.5}$$

From S, we proceed to define Y' (and thus Z') as follows:

$$Z'_{k,i} = S_{k,i}(X_k^T + P_k^T) \tag{3.6}$$

similar to Z, Z' is a 3-dimensional matrix. However, in contrast to Z, the entries $Z'_{k,i,t}$ represent the number of vehicles that are considered to be treating road segment i at time t due to route k. Again, note that the T superscript above refers to the transpose matrix operation, and not to the number of time periods in the model. A simple summation of the form:

$$Y'_{i,t} = \sum_{k=0}^{m-1} Z'_{k,i,t} \tag{3.7}$$

produces the required values of Y'.

We note that it is also possible for the values of Y' to be set to any interval whose corresponding entry in Y is set to 1, making it possible for the instant of treatment to be any point at which a vehicle is on a road segment, not just the first such interval as is described above. Thus, treatment intervals could be considered to take the value of the first, last, middle or any other time interval within a road segment traversal. Although this possibility is not used in our model, it is nonetheless noted.

3.4.3 Treatment Effectiveness Modelling

Once we have derived a number of instantaneous times of treatment for each road segment from the departure schedule, we must then determine how these treatment events affect actual road conditions. The effectiveness of a treatment typically depends on several factors both external and internal to this ILP. An expression of how a given service affects a road network depends on both the type of service and possibly on the weather's effect on the treatment (for example, in the case of road salting, pavement temperature is a major factor in the effectiveness of salting operations). To avoid overly complicating the model structure, we assume that the effectiveness of a treatment event can be modelled linearly as a function of existing variables within the ILP. Although this research considers only plowing operations, we briefly describe how an effectiveness model for salting operations would be constructed.

Plowing Assume that a plowing operation removes all accumulated snow up to a certain maximum depth δ. Then the change in road condition due to treatment is exactly $-\delta$. As the values of $Y'_{i,t}$ give us the number of treatment events on each road segment for a given time interval, a simple multiplication of their values by δ will provide the desired result. Establishing a lower bound of zero on the road condition ensures that we will never have a negative amount of snow predicted for any road segment (that is, plowing can only ever take the road down to bare pavement).

Now that we have determined how to quantify the effectiveness of a plowing event, we can establish a relationship between the values of $Y'_{i,t}$ and their effect on road condition. As this has a direct impact on the formulation of the intermediate variables $C_{i,t}$, we have not been able to describe them until now. The values of C can be stated recursively as follows:

$$C_{i,t} = \begin{cases} \max(0, \boldsymbol{w}_i + W_{i,0} - \delta Y'_{i,0}) & \text{if } t = 0 \\ \max(0, C_{i,t-1} + W_{i,t} - \delta Y'_{i,t}) & \text{otherwise} \end{cases} \tag{3.8}$$

where δ represents the maximum amount of snow that can be removed by a single service event. Note that this formulation is specific to plowing operations, as it depends on the relation between treatment events and their effect on road condition, which varies for different types of treatment.

Salting Salting depends on both the application time and temperature. A linear estimate for future effectiveness given application times and temperatures should be able to be formulated from models in the literature, however, this work is not undertaken here. A re-statement of the WRMSP to model salting operations would involve restating the effect of a treatment event as spanning multiple time intervals, rather than the immediate and single period δ function used for plowing. A model would have to be developed describing how, if a treatment is delivered at time t, the road condition changes at time $t+1, t+2, \ldots$. Then, the effect of a treatment event would not be isolated solely in the instant when the treatment event takes place (as is the case with plowing), but for a given interval after the treatment.

3.5 Additional Constraints

3.5.1 Condition Recursion

We saw in Equation 3.8 that road conditions can be modelled recursively. To utilise this recursion in the context of a linear program, we must cast it as part of the linear program. We do this by introducing the following constraints:

$$C_{i,0} = \boldsymbol{w}_i + W_{i,0} - \delta Y'_{i,0} \qquad \forall i \in \{0, \ldots, n-1\} \tag{3.9}$$

$$C_{i,t} = C_{i,t-1} + W_{i,t} - \delta Y'_{i,t} \qquad \forall i,t \in \{\{0, \ldots, n-1\} \times \{0, \ldots, T-1\}\} \tag{3.10}$$

$$C_{i,t} \in \mathbb{R}^+ \qquad \forall i,t \in \{\{0, \ldots, n-1\} \times \{0, \ldots, T-1\}\} \tag{3.11}$$

The non-negative constraint on values in $C_{i,t}$ ensures that we do not ever have a ‘negative’ amount of snow on a road segment.

3.5.2 Level of Service Constraints

In our planning model, we require the condition of managed roads to not exceed a given threshold. Such a threshold is defined as an input to this program as $\boldsymbol{\phi}_i$. We already have in Equations 3.9 and 3.10 a recurrence relation for condition values for each combination of time interval and road segment, so constraining these values to lie below the level of service threshold is a simple matter of adding the following constraint into the program:

$$C_{i,t} \leq \boldsymbol{\phi}_i \qquad \forall i, t \in \{\{0, \ldots, n-1\} \times \{0, \ldots, T-1\}\} \tag{3.12}$$

We note that for rural settings where we may not have jurisdictional responsibility for certain road segments in a network, but which we may nonetheless use for deadheading travel, we can set their Level of Service values to be ∞, causing their condition to be disregarded in the solution of the model.

We also note that the insistence that Level of Service values are respected in any candidate solution may cause the algorithm to not find any feasible solutions if none exist that can guarantee service levels are respected. In other words, with this constraint in place, the algorithm may fail to find any candidate solutions if weather events are severe enough to preclude service levels being guaranteed on all road segments at all times. Thus, if we wish to use the WRMSP to find a solution that makes a 'best-effort' toward treating a storm event even though service levels may be violated some of the time, we can simply remove these constraints from the problem. This becomes particularly important when using heuristics to solve the WRMSP.

3.5.3 Fleet Size Constraints

A fleet size constraint is formed by ensuring that no more than b vehicles are in use during any time period. We have already derived a table Y whose values represent the number of vehicles on a given road segment at a given time, so taking a column sum of its elements will yield the total number of vehicles on the road at a given time. Thus, the following constraint suffices:

$$\sum_{i=0}^{n-1} Y_{i,t} \leq b \qquad \forall t \in \{0, \ldots, T-1\} \tag{3.13}$$

Note that this constraint does not restrict a vehicle to only undertake a single departure within the simulation period, but rather that there will never be more than b vehicles in service at any given time. A given service vehicle is free to undertake an arbitrary number of departures subject to the constraints defined within the WRMSP.

3.6 The Combined ILP

Having described all the constituent constraints and functions that form the WRMSP, we now express the WRMSP as a consolidated Integer Linear Program, as follows:

minimise

$$\sum_{i=0}^{n-1}\sum_{t=0}^{T-1}\boldsymbol{\lambda}_i C_{i,t} \quad + \quad \beta\sum_{t=0}^{T-1}\sum_{k=0}^{m-1}\boldsymbol{d}_k X_{k,t} \tag{3.1}$$

subject to

$$C_{i,0} = \boldsymbol{w}_i + W_{i,0} - \delta Y'_{i,0} \qquad \forall i \in \{0,\ldots,n-1\} \tag{3.9}$$

$$C_{i,t} = C_{i,t-1} + W_{i,t} - \delta Y'_{i,t} \qquad \forall i,t \in \{\{0,\ldots,n-1\} \times \{0,\ldots,T-1\}\} \tag{3.10}$$

$$C_{i,t} \leq \boldsymbol{\phi}_i \qquad \forall i,t \in \{\{0,\ldots,n-1\} \times \{0,\ldots,T-1\}\} \tag{3.12}$$

$$\sum_{i=0}^{n-1} Y_{i,t} \leq b \qquad \forall t \in \{0,\ldots,T-1\} \tag{3.13}$$

$$Z_{k,i} = Q_{k,i}(X_k^T + P_k^T) \qquad \forall i,k \in \{\{0,\ldots,n-1\} \times \{0,\ldots,m-1\}\} \tag{3.3}$$

$$Y_{i,t} = \sum_{k=0}^{m-1} Z_{k,i,t} \qquad \forall i,t \in \{\{0,\ldots,n-1\} \times \{0,\ldots,T-1\}\} \tag{3.4}$$

$$Z'_{k,i} = S_{k,i}(X_k^T + P_k^T) \qquad \forall i,k \in \{\{0,\ldots,n-1\} \times \{0,\ldots,m-1\}\} \tag{3.6}$$

$$Y'_{i,t} = \sum_{k=0}^{m-1} Z'_{k,i,t} \qquad \forall i,t \in \{\{0,\ldots,n-1\} \times \{0,\ldots,T-1\}\} \tag{3.7}$$

$$C_{i,t} \in \mathbb{R}^+ \qquad \forall i,t \in \{\{0,\ldots,n-1\} \times \{0,\ldots,T-1\}\} \tag{3.11}$$

$$X_{k,t} \in \{0,1\} \qquad \forall t,k \in \{\{0,\ldots,T-1\} \times \{0,\ldots,m-1\}\}$$

$$Y_{i,t} \in \mathbb{Z}^+ \qquad \forall i,t \in \{\{0,\ldots,n-1\} \times \{0,\ldots,T-1\}\}$$

$$Y'_{i,t} \in \mathbb{Z}^+ \qquad \forall i,t \in \{\{0,\ldots,n-1\} \times \{0,\ldots,T-1\}\}$$

$$Z_{k,i,t} \in \mathbb{Z}^+ \qquad \forall i,t,k \in \{\{0,\ldots,n-1\} \times \{0,\ldots,T-1\} \times \{0,\ldots,m-1\}\}$$

$$Z'_{k,i,t} \in \mathbb{Z}^+ \qquad \forall i,t,k \in \{\{0,\ldots,n-1\} \times \{0,\ldots,T-1\} \times \{0,\ldots,m-1\}\}$$

with Q and S being pre-processed from R and $\boldsymbol{l}$ as previously discussed.

Note that the above expression of the model simplifies presentation at the cost of brevity (and arguably tractability) by expanding the derivation of Y and Y' into several lines. In a concrete implementation of the model, the derivation of Y and Y' would proceed directly from X, Q and S in place where Y and Y' are referenced. It should, therefore, not be taken that the variables Y, Y', Z, and Z' are decision variables which negatively contribute to the computational complexity of the model; their existence here is for clarity of presentation only.

A summary version of the above ILP and a descriptive list of variables is included in Appendix A.

We now proceed to illustrate the WRMSP on a simple example before describing the implementation details of the model.

3.7 Example

The notation-heavy presentation of the previous section begs a simple example to clarify the reasoning and methodology behind this program. The following example runs through an iteration of the program to serve this purpose.

Consider a maintenance region as illustrated in Figure 3.3, where the patrol yard is located in the large central vertex. The 8 road segments are covered by three routes. Route 1 follows the path 4,2,1,3. Route 2 follows the path 3,7,6,5,4, and Route 3 follows the path 8,6,7,1,2,4. Assume that the service being undertaken is simple plowing (which we assume takes at most 50 cm of snow off the roadway in a single pass), and that there are two vehicles available for dispatch. All roads have the same level of service threshold of 15 cm and the same importance value (normalised to 1). Plows remove a maximum of 50 cm in a single pass, and time periods are 10 minutes in length.

The snowfall forecast is assumed to be 6 cm per hour (or 1 cm per time period) for road segments 1,3,5,7, and 2 cm per unit time for road segments 2,4,6,8 at all time periods within the planning horizon. Initial conditions have all roads at bare pavement. Assume a planning horizon of 100 minutes (so $T = 10$).

We wish to derive a deployment schedule using the data given. The algorithm will output this schedule in the form of the decision matrix, X. Entries in X that are set to 1 indicate that

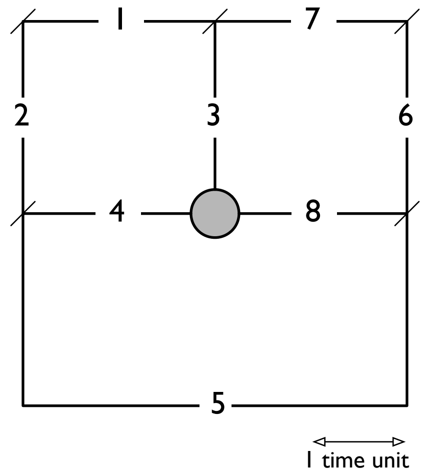

Figure 3.3: Example network graph

a vehicle should be dispatched on the road and at the time indicated by the index of the entry. Initially, the decision matrix is set to all 0, which corresponds to a deployment schedule in which no vehicles are indicated to depart at any time.

Should there be no feasible solution to this initial problem, there are insufficient resources to satisfy the level of service requirements. In this case, the only course of action is to add more service vehicles, or raise the contractual service levels until a feasible solution is found.

Note that the value of the scaling factor β has not yet been established. Recall that the value of β controls how the storm impact is weighted against the cost of providing service when determining an optimal solution. Although a formal calibration of the value of β is no undertaken, an informal sensitivity analysis indicates that a β value of 0.5 provides a dispatch schedule which elects to treat substantial snowfall, but does not dispatch vehicles where small (less than approximately 7 cm of snow). This would seem to be a reasonable treatment policy, even though the value of β which produced it was derived empirically.

We run the example with a β value of 0.5, chosen to provide a mix of weighting between road condition and cost of service. Numerically, this gives us the following parameters:

$$\boldsymbol{w} = \{0,0,0,0,0,0,0,0\} \qquad \boldsymbol{\phi} = \{15,15,15,15,15,15,15,15\}$$

$$\boldsymbol{\lambda} = \{1,1,1,1,1,1,1,1\} \qquad b = 2 \quad \delta = 50 \quad \beta = 0.5$$

$$R = \begin{Bmatrix} 3 & 2 & 4 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 5 & 4 & 3 & 2 & 0 \\ 4 & 5 & 0 & 6 & 0 & 2 & 3 & 1 \end{Bmatrix} \qquad \boldsymbol{l} = \{2,2,2,2,8,2,2,2\}$$

$$W_{i,t} = \begin{Bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 & 2 \end{Bmatrix}$$

The above parameters are taken to be invariant for a given invocation of the algorithm, so we can describe the problem's ILP numerically as follows:

minimise

$$\sum_{i=0}^{7}\sum_{t=0}^{9} \boldsymbol{\lambda}_i C_{i,t} \quad + \quad 0.5\sum_{t=0}^{9}\sum_{k=0}^{2} \boldsymbol{d}_k X_{k,t}$$

subject to

$$C_{i,0} = 0 + W_{i,0} - \delta Y'_{i,0} \qquad \forall i \in \{0,\dots,7\}$$

$$C_{i,t} = C_{i,t-1} + W_{i,t} - \delta Y'_{i,t} \qquad \forall i,t \in \{\{0,\dots,7\} \times \{0,\dots,9\}\}$$

$$C_{i,t} \leq 10 \qquad \forall i,t \in \{\{0,\dots,7\} \times \{0,\dots,9\}\}$$

$$\sum_{i=0}^{7} Y_{i,t} \leq 2 \qquad \forall t \in \{0,\dots,9\}$$

$$C_{i,t} \geq 0 \qquad \forall i,t \in \{\{0,\dots,7\} \times \{0,\dots,9\}\}$$

$$C_{i,t} \in \mathbb{R}^+ \qquad \forall i,t \in \{\{0,\dots,7\} \times \{0,\dots,9\}\}$$

$$X_{k,t} \in \{0,1\} \qquad \forall t,k \in \{\{0,\dots,9\} \times \{0,\dots,2\}\}$$

This program is solved with a standard ILP solver, which derives the required solution (details are provided in Appendix B). The solution of the example is as follows:

$$X = \begin{Bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{Bmatrix} \qquad C = \begin{Bmatrix} 0 & 1 & 2 & 3 & 4 & 0 & 1 & 2 & 3 & 0 \\ 2 & 4 & 6 & 0 & 2 & 4 & 6 & 8 & 10 & 12 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 0 & 1 & 2 \\ 2 & 0 & 2 & 4 & 6 & 8 & 10 & 12 & 14 & 0 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 4 & 6 & 8 & 10 & 0 & 2 & 4 & 6 & 8 \\ 1 & 2 & 3 & 4 & 5 & 6 & 7 & 0 & 1 & 2 \\ 2 & 4 & 6 & 0 & 2 & 4 & 6 & 8 & 10 & 12 \end{Bmatrix}$$

This result indicates a departure of a service vehicle along route 1 at times $t = 2$ and $t = 9$, and a third departure along route 3 at time $t = 3$. This dispatch plan will minimise the overall snow accumulation on the road network, resulting in the snow depths indicated in C, where the rows correspond to the numbered road segments, and the columns the time time intervals. The overall sum of entries in C is 349, which is demonstrated to be minimal for all possible dispatch schedules over the given route choices and time interval.

The final decision matrix X represents the selected dispatch schedule that should be followed, *given the current state and forecasts available.* It is expected that this program be solved repeatedly, as new information becomes available. Therefore, the solution above represents only the best possible schedule that can be derived given the information on hand at the time of its formulation. In this case, the objective function in the above equation sought to minimise the overall weighted impact of weather on the roads of the maintenance region for the time period covered by the planning horizon.

Having seen the structure of the problem's ILP formulation, we now undertake a study of the mathematical tools available to solve it, and analyse the trade-offs which must be made in terms of forecast horizon, road and route set sizes, and problem tractability.

Chapter 4

Solution Techniques and Computational Analysis

4.1 Introduction

Being an Integer Linear Program, the WRMSP suffers the same fate as any other non-trivial ILP, namely the difficulty of finding an exact solution to the problem in a reasonable amount of time. There are inevitable trade-offs to be made between the size and level of detail in the problem, and the time required to solve it. In offline applications of the WRMSP (such as when using it to compare the effects of different treatment policies or routing choices) long solution times may be acceptable, but for most real-time, online applications, a certain amount of sacrifice must be made in terms of the solution's scope and detail. The example in the previous chapter modelled the planning region for a period of 100 minutes, and included only 3 routes. In a realistic scenario, such a planning horizon may well be too short, and such a small number of routes will almost certainly be too small. Those familiar with Integer Programming concepts will notice a problem with expanding the scope of the problem; Integer Programming is an NP-complete problem, which means that the only general method to find an optimal solution is to iterate through *all* possible solutions (such an approach is termed a solution by *complete enumeration*). Thus, since all the entries in X are binary-valued, there are 2^{tk} possible solutions to an instance of the WRMSP. As a corollary, adding even a single entry in X makes the problem twice as hard to solve in the worst case. Adding even a single route, or extending the planning horizon by a small amount, may make the problem intractable. Thus, we are forced to make sacrifices when dealing with large scale problems; either forego the search for an exact solution, and satisfy ourselves with an approximation, or reduce the scope of the solution we find.

For the sake of completeness, and recognising that computational power and solution methods will only improve in the future, we elect to tackle larger problems in their full form with approximations rather than exact solutions to a smaller problem. Thus, we consider the WRMSP in a context where complete computational tractability is sacrificed in favour of the clarity and robustness of the model. The obvious initial choice of using a Branch and Bound algorithm allows us to find incremental solution approximations, and to derive an (often grossly overstated) approximation bound against the best possible solution. Such approximations are obtained as a consequence of the structure of the Branch and Bound approach, reviewed here.

4.2 Overview of Branch and Bound Solution Techniques

We consider without loss of generality all program instances to be maximisations. Branch and Bound works by considering the Linear Program formed by removing the integer or binary restrictions on decision variables in the original ILP (such a formulation is known as the *linear relaxation* of the original ILP). Branch and Bound is one of the most commonly used methods for solving ILP problems, and is a very well studied and widely used technique. The objective value of the linear relaxation forms an upper bound on the solution of the corresponding ILP, since any integer valued solution is obviously also real valued (since $\mathbb{Z} \subset \mathbb{R}$). If all decision variables in the linear relaxation are integer valued, then we have an optimal solution to the problem that satisfies the integer restrictions on the decision variables, and so we are done. Typically however, we are not so fortunate, and are forced to search for the optimal solution by considering (at least indirectly) all possible solutions to the original ILP.

Each step of a Branch and Bound solution involves fixing a subset of the Integer or Binary decision variables in an ILP, and considering the resultant problem. By fixing a subset of variables to an invariant value, we obtain a smaller problem (termed a *partial solution*) that we can analyse further. The *completions* of a partial solution are the set of possible solutions of the original ILP that agree with the partial solution in all the fixed decision variables.

The Branch and Bound method approaches an ILP by considering the tree formed by all possible combinations of integer values in the decision variables. Each leaf node of the tree corresponds to a completely determined set of decision values, and the internal nodes up to the root indicate partial solutions that have variables unfixed. Figure 4.1 illustrates an example with 3 binary decision variables. The unfixed variables in the internal nodes are indicated by asterisks. Notice that the set of completions that correspond to the root node is the entire set of possible solutions to the ILP since all variables are unfixed, and that the set of completions for each of the leaf nodes is exactly the possible solution of the node, since all variables are fixed.

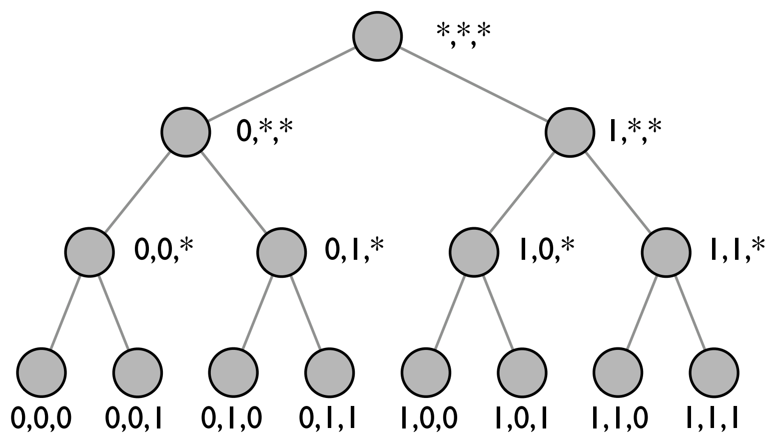

Figure 4.1: Branch and Bound search tree on 3 variables

We traverse the tree in pre-order fashion, maintaining an initially empty *incumbent solution* that represents the best solution seen thus far in our traversal. We denote the objective value of the incumbent solution by z_{inc}, and set it to $-\infty$ initially. At each node, we attempt to solve the linear relaxation of the corresponding partial solution, and denote the resultant objective value (if one exists) by z_{cur}. We will have one of four results after attempting to solve each linear relaxation:

1. The partial solution is infeasible. If the partial solution is infeasible as a Linear Program, then all completions of the partial solution are also infeasible. If any completions were feasible, it would imply a feasible solution to the partial solution. Thus, we can remove the entire subtree rooted at the current node from consideration, as we have no hope of finding any further feasible solutions in it. This is termed *Terminating by Infeasibility.*

2. The partial solution is feasible, with $z_{cur} \leq z_{inc}$. In this case, the best possible solution we could obtain from any partial solution in the subtree of the current node is less than the value of the incumbent solution, so there is no point in searching for them. Thus, we can remove the entire subtree rooted at the current node from consideration. This is termed *Terminating by Bounding*.

3. The partial solution is feasible, with $z_{cur} > z_{inc}$ and all decision variables in their required domains in the optimal solution. In this case, we know what the best possible solution is in the completions of the current partial solution, and we know that all decision variables are in whichever domain they are required to be in (typically either $\mathbb{Z}$ or $\{0, 1\}$). Thus, we cannot hope to find a better solution in the subtree that satisfies the domain requirements on the decision variables, so we can replace the incumbent solution with this optimal solution, and remove the current node's subtree from further consideration. This is termed *Termination by Solving*.

4. The partial solution is feasible, with $z_{cur} > z_{inc}$ and some decision variables not in their required domains in the optimal solutions. In this case, we determine that there exists a better solution within the completions of the current node than the existing incumbent solution, but we do not know for certain whether this solution meets all domain requirements on the decision variables. Therefore, we cannot remove the subtree from further consideration, as it may well hold a better solution to the ILP than the incumbent solution. Therefore, we *branch* at the current node, and create two further subtrees, each formed by fixing a single binary decision variable as either 0 or 1. We then recursively run the Branch and Bound algorithm on these subtrees. Note that we assume that all variables are $\{0, 1\}$ valued; in the case of integer variables, we would have to branch once for each possible value of the variable. This makes $\{0, 1\}$ decision problems much easier to solve than general integer problems.

By following the above procedure, a traversal of the solution tree is completed which yields the optimal solution of the partial solution represented by the root node. If traversal of the tree is completed without finding an incumbent solution, then the original ILP is infeasible. Otherwise, the incumbent solution remaining after the tree is traversed is the optimal solution to the original ILP.

It should be noted that the above discussion is a slight simplification of real-world Branch and Bound implementations. Specifically, the pre-order traversal indicated above is often replaced with a more robust semantic such as considering a sorted list of candidate searches ordered by the objective value of their partial solutions. Also, the decision of which variable to branch on is typically more complicated than selecting free variables in order of their index. Selecting the variable whose value is closest to being in its required domain in the node's linear relaxation solution is a typical strategy for this purpose.

It is clear that a complete enumeration of possible solutions to an ILP is solved in $\Theta(2^n)$ (that is exponential) time. The worst case running time of the Branch and Bound algorithm is no better, being $O(2^n)$ in the worst case. In reality, however, Branch and Bound provides useful results by examining the ranges of the solution tree most likely to contain good solutions first, and quickly pruning large sections which are not found to be promising. In the case of computationally large problems then, Branch and Bound remains largely intractable. However, the tree search can be terminated at any point and provided that an incumbent solution already exists, an approximate solution can be derived. An approximation bound can be obtained by comparing this incumbent solution to the optimal solution of the linear relaxation of the original ILP, above which no valid solution of the problem can hope to rise[1]. In most cases, such an early termination is exactly what happens; rarely are ILPs of a small enough size to undertake a complete evaluation of the tree.

4.2.1 Example Branch and Bound Execution

We proceed to illustrate the execution of a Branch and Bound solution of a small example problem. This problem is meant for illustration in this context only, and is specifically not an instance of the WRMSP.

[1] We can derive a stronger approximation bound than this through the use of parent bounds, but such techniques are beyond the scope of this discussion.

Consider the following ILP:

$$
\begin{aligned}
&\text{minimise} \\
&\quad x + 2y + 3z \\
&\text{subject to} \\
&\quad x + y \geq 0.5 \\
&\quad x, y, z \in \{0, 1\}
\end{aligned}
$$

We assume a search tree as shown in Figure 4.2, and a pre-order traversal. A trace of the algorithm execution follows:

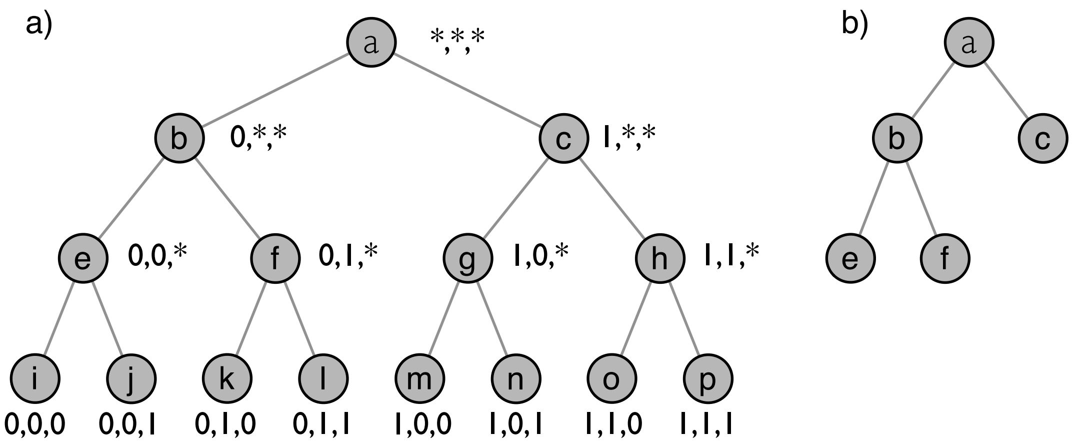

Figure 4.2: a) Search tree at full expansion, b) Search tree at algorithm termination

LP solution at a A solution of the LP relaxation specified at **a** yields an optimal solution of $0.5, 0, 0$, and an objective value of 0.5. We therefore know that no integer solution can yield a better objective value than 0.5, however, the optimal integer solution may well be worse than that. As the LP relaxation is feasible, better than our incumbent solution, but not integer, we must *branch* and examine both children trees of **a**.

LP solution at b By fixing the value of x to be 0, we obtain a two variable LP relaxation, with an optimal solution of $0, 0.5, 0$, and an objective value of 1. The solution is not integer, and the objective value is better than our (empty) incumbent integer solution, so we must *branch* again.

LP solution at e By fixing the values of x and y to be 0, we create an infeasible LP relaxation (there is no choice of z that can satisfy the LP relaxation). Thus, we *terminate by infeasibility*, and can therefore exclude the entire subtree of **e** from consideration.

LP solution at f By fixing the values of x and y to be 0 and 1 respectively, we arrive at a single variable LP relaxation with an optimal solution of $0, 1, 0$ and an objective value of 2. This solution is both feasible and integer, and it is also better than the incumbent solution. We therefore designate this solution as our new incumbent solution, and *terminate by solving* at this node, since no child solution can rise above the LP relaxation assigned at **f**.

LP solution at c With x fixed at 1, we solve the two variable LP relaxation and obtain an optimal solution of $1, 0, 0$, with an objective value of 1. Again, this solution is both feasible and integer, and is also better than the incumbent solution. We therefore designate this solution as our new incumbent solution, and *terminate by solving* at this node.

At this point, we have exhausted all nodes in the search tree, and have therefore considered (either directly or indirectly) all possible solutions x, y, z to the ILP. The optimal solution to the problem is therefore $1, 0, 0$, with an objective value of 1. As the size of this problem is very small, we can easily verify the optimality of this solution manually.

Solving this problem via Branch and Bound required conducting a total of five LP relaxations, an improvement over the eight that would be required for a complete enumeration of all possible solutions to the ILP. This improvement is not guaranteed, however, and so the worst case running time for Branch and Bound remains $O(2^n)$, which although an improvement in most real world cases over the $\Theta(2^n)$ running time of complete enumeration, is still typically intractable for large n.

4.3 Branch and Bound Applied to the WRMSP

We now consider the Branch and Bound algorithm as it relates to the WRMSP problem, beginning with an analysis of factors that affect the runtime of Branch and Bound approaches to solving the WRMSP.

4.3.1 Impact of Prediction Horizon and Size of Candidate Route Set on Branch and Bound Tractability

The following discussion concerns a solution approach that uses only a naive, general purpose Branch and Bound algorithm, and in particular does not use purpose built algorithms, nor any exploitation of the problem structure.

As with any ILP, the input size dominating the computational complexity of the WRMSP model is in the number of integer valued decision variables present. Given no restrictions on the number of such variables that can take on a certain value (in other words, without limiting the number of such decision variables which could take on the value '1', for example), the complexity involved in finding an exact solution to the ILP grows exponentially with the number of integer decision variables, with the base of the exponential growth being the number of possible discrete values the variables may take on. In the case of the WRMSP, the integer valued decision variables are exactly the values of the $X_{t,k}$ matrix. The worst case complexity of Branch and Bound remains the same as complete enumeration ($O(2^{tk})$), however, we can usually expect a decrease in typical runtimes of a substantial amount, as seen in Section 4.5, later in this chapter.

Note that although the condition values for each road segment are considered as decision variables, they are real valued in the model and so contribute only to the complexity in optimising the systems described by the partial solutions encountered during a traversal of the decision search tree. As such, their contribution to the model's complexity disappears asymptotically under the complexity of a complete search tree traversal.

4.3.2 Impact of Fleet Size Restrictions on Branch and Bound Tractability

Typically, the number of possible departure time and route combinations for a problem instance is much greater than the actual number of such departures that can be indicated in a solution. Stated mathematically, any feasible solution X to the WRMSP is expected to be sparse. This result is due to the fleet size constraint in place within the model, which requires that no feasible solution allow more service vehicles to be on the road at any time than exist in the fleet. On the surface, this may seem to suggest that only as many non-zero entries in X as there are vehicles in the fleet, but this is not the case. The fleet size constraint only restricts the number of concurrent vehicles on the road at any one time, not the total number of dispatches made throughout the prediction period. If the prediction window is long and most routes of a short duration, it may well be the case that a single vehicle will make several dispatches within the prediction window, and will thus contribute several non-zero entries to the solution described in X. However, results obtained by running the WRMSP with small fleet sizes produce drastically protracted runtimes, hinting at the efficiencies gained within the Branch and Bound algorithm as a result of reduced fleet size constraints. For details of these efficiencies, see Section 4.5 later in this chapter. We note here that optimal solutions of the WRMSP with a single vehicle to be scheduled can be obtained efficiently using Branch and Bound (within approximately 10 seconds on modest hardware on a typical WRMSP instance). This observation will provide a critical foundation to approximation heuristics described shortly. Nonetheless, for realistic combinations of route set sizes, planning horizons, and fleet restrictions, we are forced to abandon conventional Branch and Bound approaches and focus on approximation heuristics.

4.4 Heuristic Approaches

Approximation heuristics remain the only way to solve many otherwise intractable computational problems. As the WRMSP has been demonstrated to be generally computationally intractable using Branch and Bound techniques, we turn to approximation algorithms as an alternate solution approach. We first outline three heuristic approaches, and then establish estimates of their approximation quality.

4.4.1 Add-One-At-A-Time Heuristic

We have seen that the WRMSP is readily tractable for small fleet sizes, and in particular that a fleet size of 1 leads to very rapid solutions. Combining this with the observation that previously dispatched vehicles can be integrated into the simulation model through the use of the $P_{k,t}$ parameter, we can derive an iterative solution method which follows the following structure:

This algorithm is greedy in nature. Initially, the most effective single vehicle deployment schedule is derived (note that this schedule may include more than one deployment, but will never violate the fleet size constraint). These deployments are then forced in a second instance of the model, and the next most effective deployment is determined. This process continues until

Algorithm 1 Add-One-At-A-Time Approximation Heuristic for the WRMSP

$P = 0$

$b = 1$

for $i = 1, \ldots, fleetsize$ **do**

  $X = solveWRMSP()$

  $P = P + X$

end for

return P

a solution violates the actual fleet size constraint. For each problem instance, the fleet size is constrained at 1, which ensures that only a single vehicle will be considered for scheduling.

In order for this heuristic to be applied, two minor changes must be made to the WRMSP:

1. Level of service constraints are removed from the WRMSP formulation, as discussed in Section 3.5.2. This is to ensure that the WRMSP will still find a feasible solution to the ILP instance, even though a single vehicle may have no hope of maintaining overall level of service guarantees. Note that this may preclude the satisfaction of level of service constraints in the overall solution, although the WRMSP will still seek to minimise storm impact.

2. The fleet size constraint functions are reformulated to enforce fleet size restrictions only on the number of service vehicles that can be controlled through the value of X, as opposed to the sum of values in X and those in P. This allows us to keep the fleet size b as 1 throughout all iterations of the model, and ensures the efficient solution of each iteration without violating fleet size constraints. Formally, we change the equality constraint:

$$Z_{k,i} = Q_{k,i}(X_k^T + P_k^T) \qquad \forall i, k \in \{\{0, \ldots, n-1\} \times \{0, \ldots, m-1\}\} \tag{3.3}$$

in the formulation of the WRMSP to read:

$$Z_{k,i} = Q_{k,i} X_k^T \qquad \forall i, k \in \{\{0, \ldots, n-1\} \times \{0, \ldots, m-1\}\}$$

At the completion of Algorithm 1, the value of P will contain a dispatch schedule that is guaranteed by construction to respect the fleet size constraint (and indeed, may not even be tight against this bound). Empirical analysis of the approximation quality of this algorithm is undertaken in Section 4.5. One of the expected benefits of this approach is that the most effective treatments are those that are prescribed in the first iterations of the algorithm – later iterations serve to find departures which, although presumably of positive benefit, provide less benefit than those already found (in short, the marginal benefit of each subsequent departure recommendation decreases monotonically with each iteration).

In addition, the nature of the algorithm ensures that all solutions are derived deterministically. That is, there is no stochastic aspect to this algorithm; given identical inputs, two runs of this algorithm will produce identical results. This repeatability may be beneficial in that it does not preclude stochastic analysis (input weather information can still be varied across multiple trials to model the variable nature of weather forecasts), but does allow for more predictable analysis and research.

4.4.2 Add-n-At-A-Time Heuristic

This heuristic is very similar in structure to the Add-One-At-A-Time heuristic, and is included here primarily in an attempt to overcome the greedy nature of the Add-One-At-A-Time heuristic. In this heuristic, we solve the WRMSP in consecutive iterations, adding a small, constant n vehicles into the predetermined solution at every iteration, stopping when we have reached the fleet size constraint of the overall problem.

The primary drawback of this approach is that the optimal solution search time increases exponentially with every additional vehicle to schedule. Thus, a solution for n vehicles will take a^n for some fixed a, where the value of a in the growth function depends on the size of the problem inputs. That being said, problem instances may remain tractable for small values of n if the solution time for each iteration is small enough. The growth of the overall solution time is a function of both the value of n, and the total number of iterations required to reach the fleet size constraint. For example, if the fleet size is given by k, then the total number of iterations required is $\lceil k/n \rceil$. The overall running time is thus $O(\lceil k/n \rceil 2^n)$. As this running time also holds true for the case where $n = 1$ (the Add-One-At-A-Time heuristic described previously), it is obvious that the Add-n-At-A-Time heuristic will require more time to execute in virtually all scenarios than the Add-One-At-A-Time heuristic.

4.4.3 Remove-And-Reinsert-Entries Heuristic

This heuristic is an improvement step that can be added to the previously discussed heuristics. The procedure is largely unchanged from those discussed above, with the addition of an extra step every fixed number of iterations which removes a previously inserted dispatch from the overall solution. The intention of this algorithm is to reverse greedy decisions which may have been bad in hindsight. A variety of choices exist for the mechanism for selecting which previous vehicle to remove, however, we will utilise two approaches in this work:

FIFO Selection: A FIFO queue of vehicles is maintained based on the insertion order of the original heuristic, and vehicles are removed from the front of this queue. In other words, the oldest remaining vehicle entry in the overall solution is removed at each removal step. Having the removed vehicle added back into the solution on the next iteration of the original heuristic serves to validate the original choice, as it shows the original choice to still be optimal in the overall solution at hand.

Random Selection: A vehicle is randomly chosen to be removed from the set of vehicles in the current overall solution. This makes the algorithm non-deterministic in nature. Again, having the removed vehicle added back into the solution serves to validate its original choice.

We will run simulations with every third iteration being replaced by a deletion step for the Add-One-At-A-Time heuristic described previously, using both the FIFO and random selection models to determine which vehicle to remove. When deleting a vehicle, we actually delete all entries specified by a given iteration of the algorithm (in other words, if the algorithm specified three departures for a single vehicle in a given iteration, then all three departures would be removed by the deletion step). This is to ensure that fleet size restrictions are easily kept track of.

It should be noted that this improvement step can be applied to an existing heuristic solution at any time, and specifically can be used on a complete solution to attempt to improve its final result. In this sense, this heuristic can be used as an independent improvement heuristic in the general case.

4.5 Computational Analysis

4.5.1 Case Descriptions

We utilise two road models for this analysis. The first model is taken from the street network of Waterloo, Ontario, and is illustrated in Figure 4.3. The corresponding abstract network is illustrated in Figure 4.4. This model is meant to illustrate the typical approach to urban maintenance operations, which is to dispatch vehicles at the neighbourhood level and leave the decision of how to traverse the streets within a neighbourhood up to the vehicle operator.

The neighbourhood / connecting arterial structure of this network is modelled as a connected graph where each neighbourhood is represented by a vertex. Edges between the vertices represent the travel time along arterials to travel from neighbourhood to neighbourhood, and loop edges at each vertex represent the time required to treat the neighbourhood streets[27]. In this way, the entire road network is considered traversed only when all edges of its representative graph are traversed.

The second model seeks to emulate the structure commonly found in rural environments. In particular, the graph is more sparse in structure, consisting of primary highways with high levels of service, and secondary roads of lower service levels. There are no loop edges, as the road network is simple enough to model completely. In addition, road segments present in certain routes may be used for dead-heading purposes, and are not serviced during their traversal due to jurisdictional concerns, as discussed in Section 2.3.

The rural road network used is illustrated in Figure 4.5, and in loosely based on the road network in north Charlottenburgh Township, Ontario. We consider treatment from the point of view of the township/county operator, and so while the provincial highways in the network are available for transit, they do not require treatment. To ensure the WRMSP correctly models our obligations in this regard, weather impact on provincial highway segments of the network is set to zero for all time periods (to ensure that we are not penalised for weather impact that we cannot treat).

The weather forecasts in both cases are assumed to be blanket snowfalls, with uniform variation across all road segments forecast in both cases. Simulation windows are 3 hours into the future in both cases, divided into eighteen 10 minute intervals. Storm intensity is forecast to slowly ramp up over the first 60 minutes of simulation, reaching a constant peak for 30 minutes, and then tapering off in the last 90 minutes of simulation. As in the example in Section 3.7, a β value of 0.5 is chosen empirically as a balanced trade-off between effective treatment of major snowfall and cost of treatment.

The exact palette of routes available to choose from in each model were chosen to guarantee coverage of all road segments, and to provide duplicate route choices for most segments. In total, there were 11 routes considered for both the urban and the rural case. All roads were given the same treatment priority in both cases.

4.5.2 Simulation Settings and Results

Each of the heuristics was run as described above for fleet sizes from 1 to 12 vehicles (with the exception of the exact solutions, which were both intractable for any more than 2 vehicles). In addition to the heuristics described above, the following results are included:

LP Relaxation The LP relaxation of any ILP forms an absolute lower bound on the objective value of the ILP, since any solution constrained to the integers is also a solution in the reals (formally, since $\mathbb{Z} \subset \mathbb{R}$). Although the LP relaxation objective values are included for comparison purposes, it is likely that the lower bound they describe is quite loose, and so little weight should be ascribed to these values

Abbreviated Branch and Bound search The Branch and Bound algorithm was allowed to run on an otherwise complete form of the WRMSP for a fixed time period (10 minutes in this case), and the optimal objective value obtained at the conclusion of this interval was taken as the result.

To ensure a fair comparison between algorithms, level of service constraints were removed from all simulations, as described in Section 4.4.1.

Details of the testing environment, including the hardware and software used, are described in Appendix F.

As an example, the solution determined by the first 5 iterations of the Add-One-At-A-Time heuristic for the rural case is shown in Figure 4.6. Recall that since each iteration of the Add-One-At-A-Time heuristic adds another vehicle into the plan, the solution set after 5 iterations must be valid for a fleet of at most 5 vehicles (and indeed, the illustrated example meets this bound tightly).

Results are described in Figure 4.8 for the urban case, and Figure 4.7 for the rural case.

In the included Figures, the x axis describes the number of vehicles deployed. In the exact, abbreviated Branch and Bound, and LP relaxation solutions, this corresponds to the explicit value of b. In the Add-One-At-A-Time, Add-Two-At-A-Time, and removal heuristics, this corresponds to the fleet size as determined by the overall algorithm, and not the sub-iterations of the WRMSP. The y axis describes the optimal objective value determined by the algorithm. In the implementation of the WRMSP used in this simulation, this is a weighted sum of the overall snow accumulation on all road segments over time and the cost of providing treatment. A β value of 0.5 was used for the weight.

Runtimes for the ILP solver component of the various heuristics are described in Figure 4.10 for the urban case, and Figure 4.9 for the rural case.

In the included runtime Figures, the x axis describes the number of vehicles deployed. As in the objective value plots, in the exact, abbreviated Branch and Bound, and LP relaxation solutions, this corresponds to the fleet size as specified in the WRMSP. In the Add-One-At-A-Time, Add-Two-At-A-Time, and removal heuristics, this corresponds to the fleet size as determined by the overall algorithm, and not the sub-iterations of the WRMSP. The y axis describes the number of seconds the ILP solver component of each required to run to completion (in the case of the abbreviated Branch and Bound, the value is the lesser of the time to complete and 10 minutes). Execution time for the data manipulation components of each algorithm are not included, but would not materially affect the results even if they were.

4.5.3 Analysis of Simulation Results

As can be seen from the included figures, the results obtained from all heuristics correlate very closely with one another (with the exception of the LP relaxation, which is included here solely as an absolute lower bound on the solution, and is not proposed as a solution heuristic per se).

Highlights of specific approaches are described in the following sections.

Add-One-At-A-Time Heuristic

The Add-One-At-A-Time heuristic performed quite well in simulation. While it rarely was the best performing of all the heuristics, it typically came within a close bound of the best performing algorithm (deviating at most 2.5% from the best heuristic solution in the urban case, and 17.4% in the rural case).

The major benefit to this heuristic is the approximately linear scaling of its runtime. Because each additional vehicle added to the existing solution corresponds to a solution of the WRMSP with a b of 1, every additional vehicle takes approximately the same marginal amount of time to solve for. This is a major benefit for applications where real-time operation is paramount.

In the urban model the route entries selected by this heuristic spanned all route choices, and in the rural model, nine of the eleven route choices were selected. This suggests that there is indeed a benefit to providing a large number of routes to choose from, particularly in light of the fact that the heuristic solved each new vehicle addition to optimality.

It should also be noted that marginal benefit provided by the first 5 plows comprises more than half of the overall improvement in the objective value. While it is true that we can expect that the earlier vehicle additions will provide a larger share of the overall benefit, this observation illustrates that the cost effectiveness of deploying more than 5 plows in this area may be questionable, an observation borne out by actual practise where no more than 4 or 5 plows would ever be in this service area for the simulated storm event. Decisions not to deploy later vehicles would be produced by larger values of β in the objective function.

Add-Two-At-A-Time Heuristic

The Add-Two-At-A-Time heuristic performed extremely well in the urban case, where it produced the optimal heuristic objective value in 5 out of 6 instances (because this heuristic adds vehicles in pairs, it is only included in comparisons for even numbered fleet sizes). However, the runtime of this heuristic leaves much to be desired, with each 2 vehicle addition taking well over 2000 seconds to compute. This deficiency makes this algorithm all but unusable for real-time applications, although it is still very useful for finding optimal heuristic solutions in test or simulation cases.

This heuristic did not perform as well in the rural case as in the urban case, most likely due to its greedy nature (as will be described in the 4.5.3 section below).

Removal Heuristics

Recall that the intent of the removal steps in this heuristic were meant to overcome the greedy nature of the Add-n-At-A-Time heuristics. By removing previously indicated solutions and then resolving the resultant problem, we can hope to minimise the impact of early solutions that may be suboptimal after subsequent iterations.

A crucial observation is that this algorithm will never produce results worse than the corresponding solution given by the Add-One-At-A-Time heuristic, provided that the two solutions were identical in the step preceding the removal. Proving this observation is straightforward; if the WRMSP search after the removal step solves to optimality (which it does), then it is guaranteed to consider the configuration described by the Add-One-At-A-Time heuristic, and so will find a solution at least as optimal as that one. This observation does not hold true for the algorithm as a whole, however. It may well be that the algorithm decides on an alternate treatment plan after a removal step which proves sub-optimal in the overall problem. Indeed, this is exactly what happens with the Random Remove heuristic in the urban case; a bad choice at step 8 of the algorithm causes the Add-One-At-A-Time heuristic to eclipse it in overall performance.

In the urban case, the Random Removal heuristic provided slightly better results than the FIFO Removal heuristic, while in the rural case the opposite was true, with the FIFO proving a much better algorithm than the random heuristic. Given the stochastic nature of the random algorithm (all other heuristics considered here are entirely deterministic), its usefulness may also be marginalised depending on the application.

Overall, the removal heuristics provide a good balance of runtime efficiency and solution quality. They generally provide superior results to the Add-One-At-A-Time heuristic, while maintaining a runtime asymptotically equivalent to the Add-One-At-A-Time heuristic.

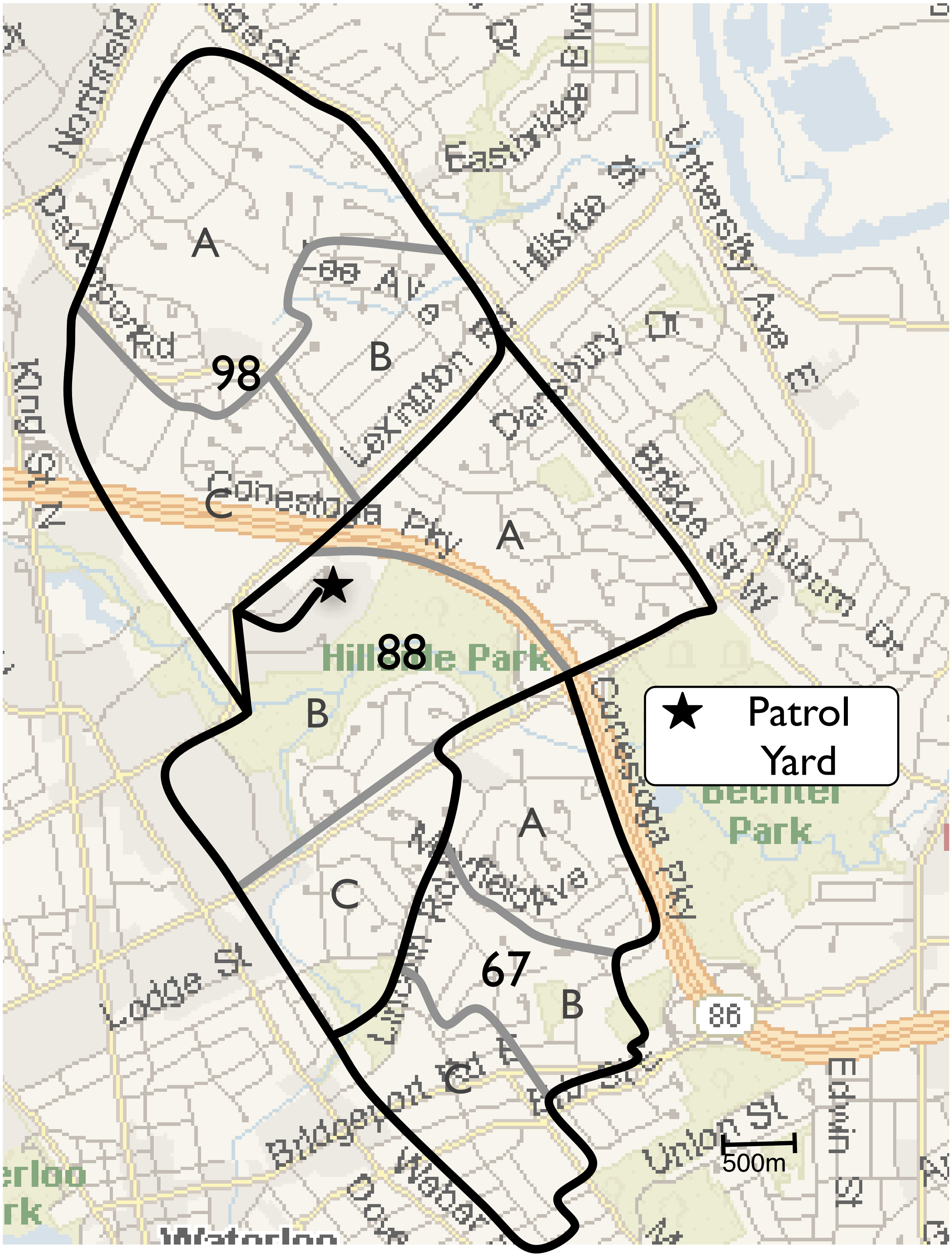

Figure 4.3: Map of portion of the City of Waterloo

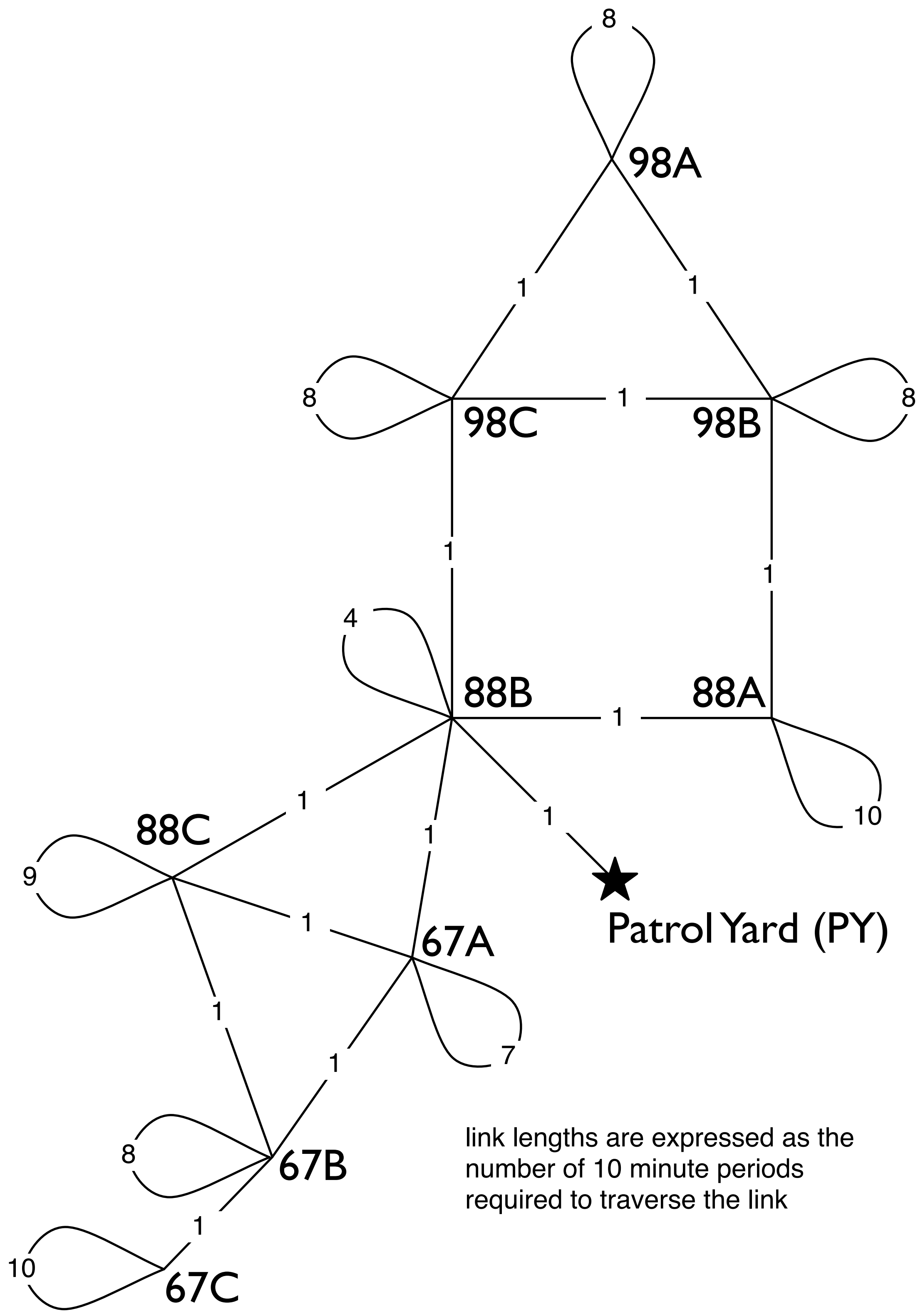

Figure 4.4: Network model of portion of the City of Waterloo

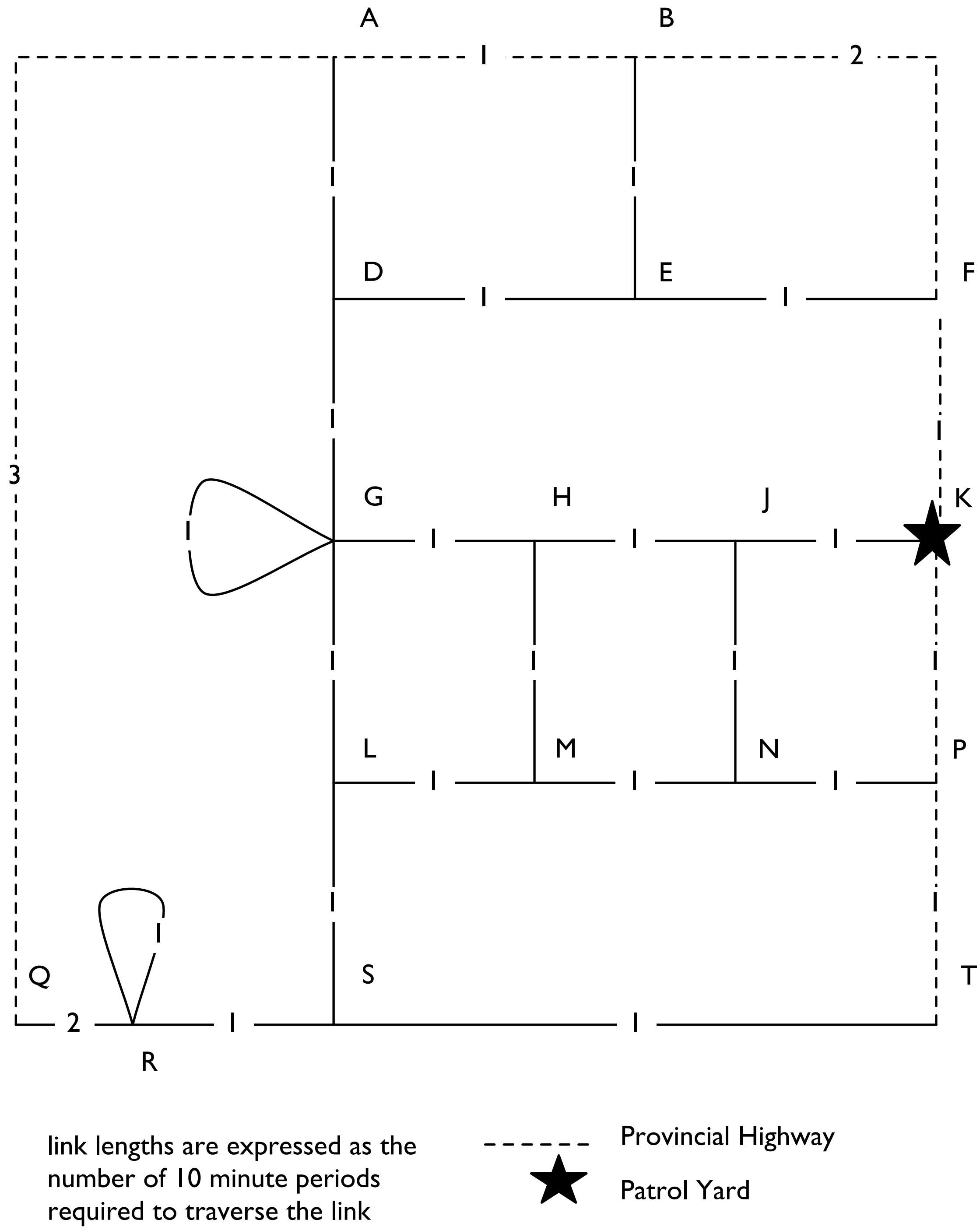

Figure 4.5: Example rural road network

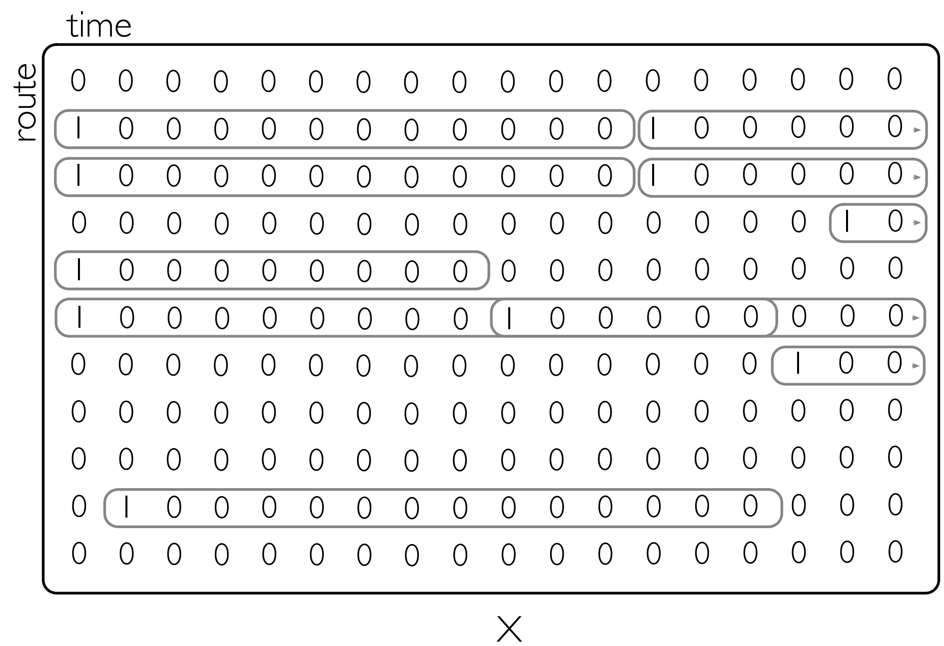

Figure 4.6: Sample solution schedule for rural case, $b = 5$

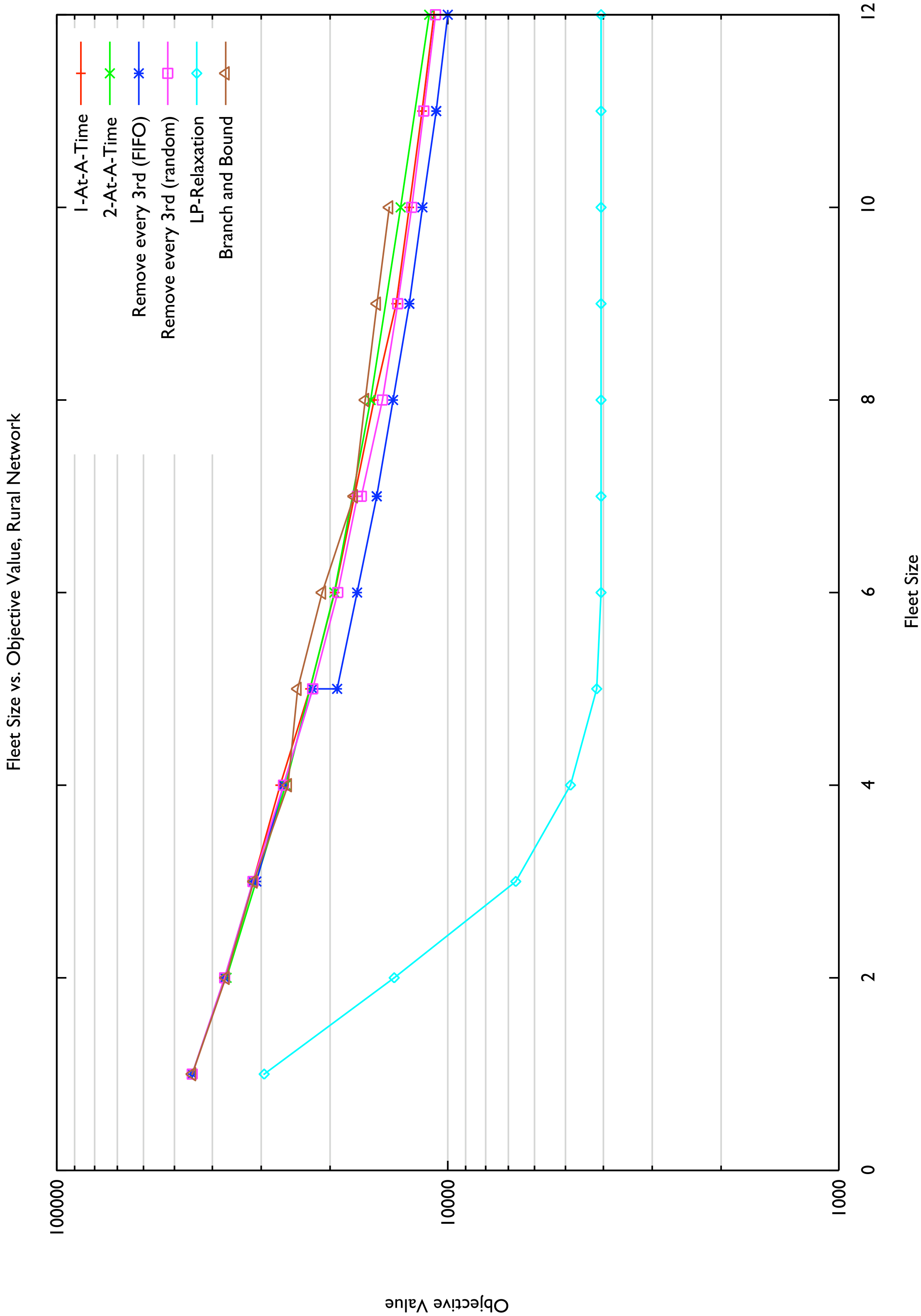

Figure 4.7: Objective value vs. fleet size for various heuristics in an rural network

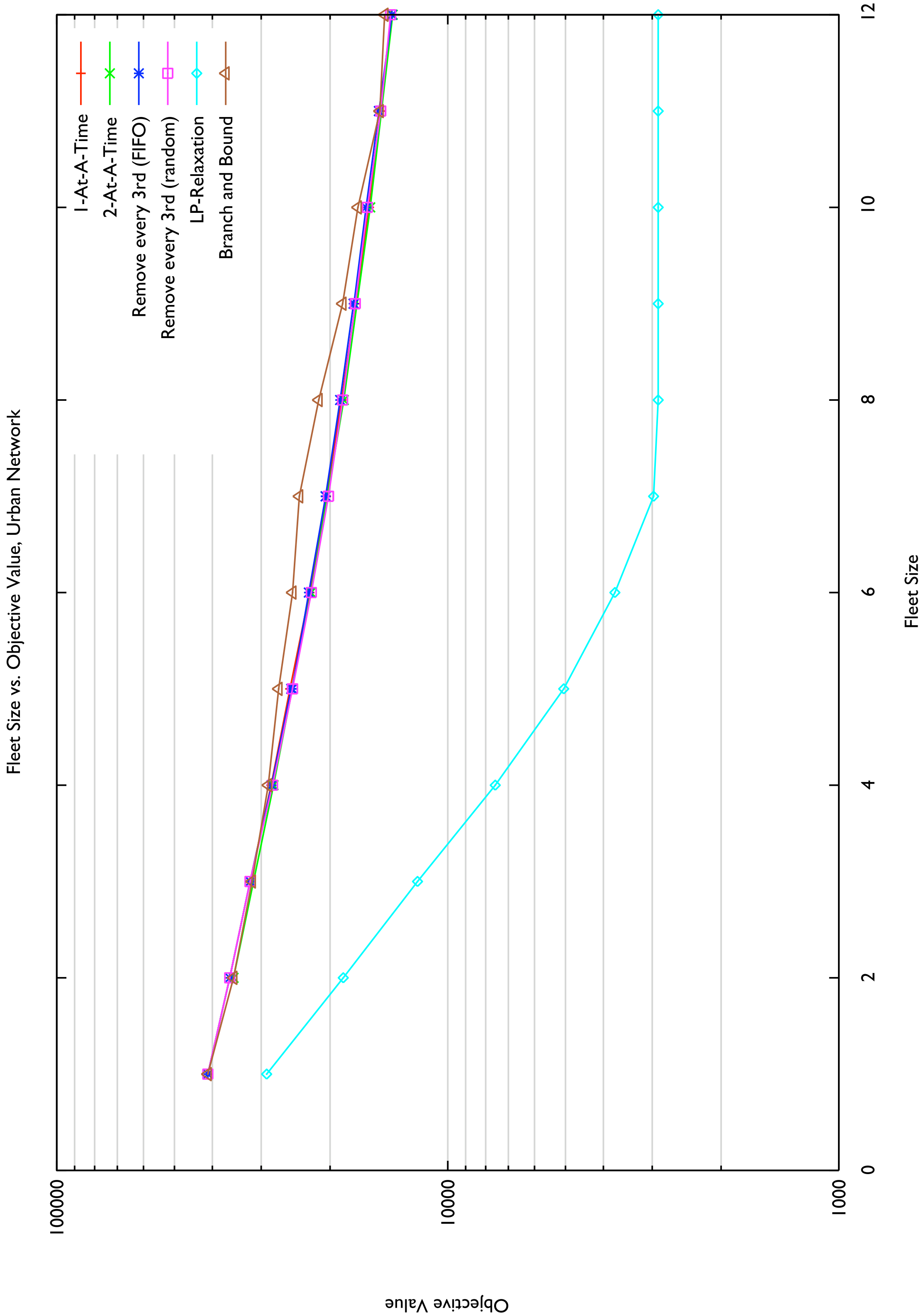

Figure 4.8: Objective value vs. fleet size for various heuristics in an urban network

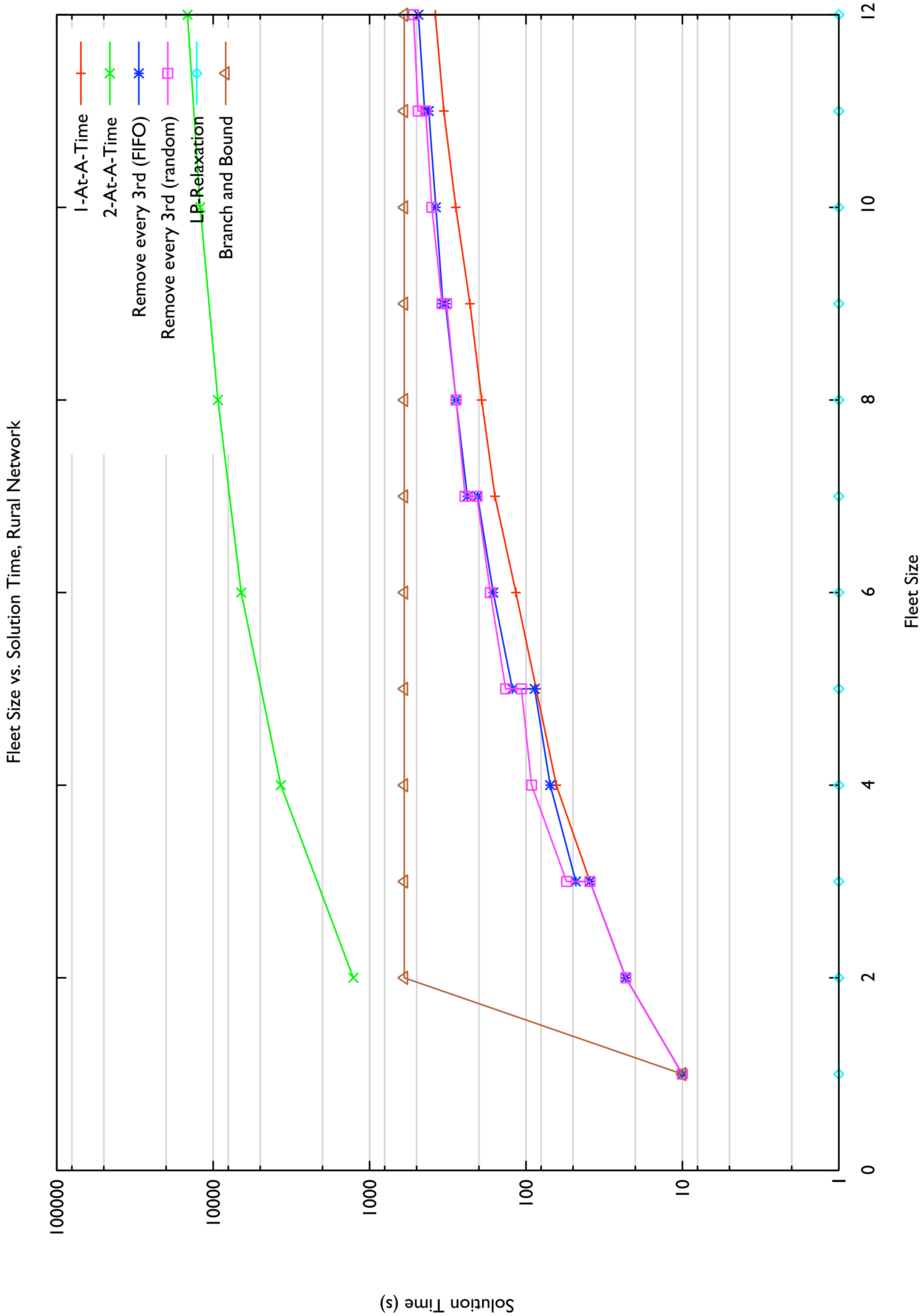

Figure 4.9: Runtime vs. fleet size for various heuristics in a rural network

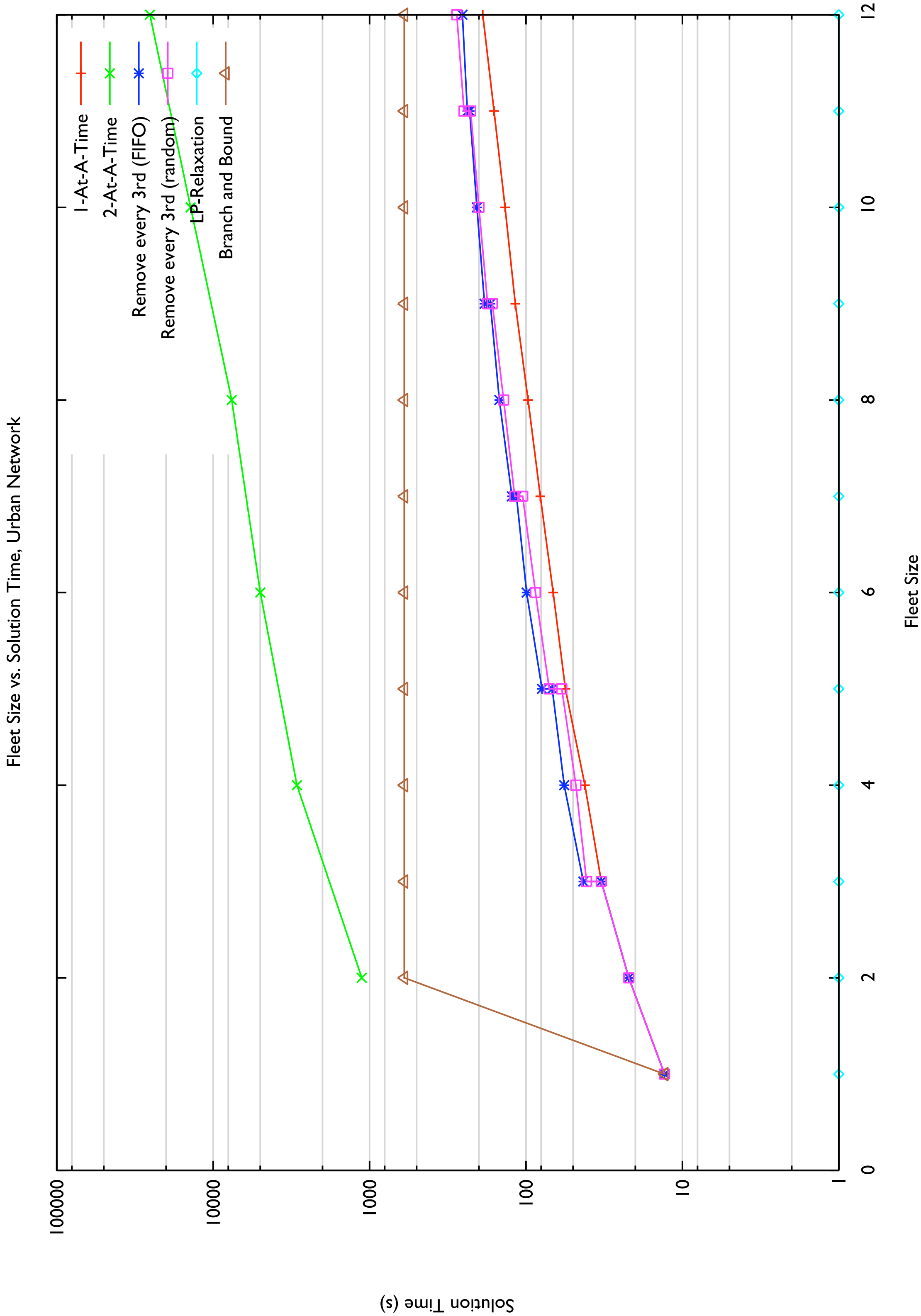

Figure 4.10: Runtime vs. fleet size for various heuristics in an urban network

Chapter 5

Conclusion

5.1 Summary of Work

A comprehensive summary of existing research, both theoretical and practical, was undertaken in Chapter 2. This summary explored existing research in three major areas: weather forecasting, treatment effectiveness modelling, and existing route design strategies. As these three areas form the foundation on which our model is built, a concrete understanding of them is critical. In addition, the introductory chapter discussed the present state of maintenance practise, and outlined the problems the WRMSP is designed to solve.

The WRMSP was described in detail as an Integer Linear Program (ILP). The model accepts inputs describing current and predicted weather conditions, road topology information, road priorities and service level contracts, fleet size restrictions, and several other secondary inputs. The output of the model consists of two parts. The first of these is a dispatch matrix, which describes when and on which routes service vehicles should depart to realise an optimal treatment schedule. The second component of the WRMSP output is in the form of a condition matrix, which describes the anticipated road condition values over time for each section of road, given that the treatment schedule is followed. This information is useful as a visualisation tool for maintenance supervisors and emergency responders, and also as an informational tool for travellers.

Being an ILP, the WRMSP is intrinsically hard to solve. Therefore, heuristics are used to lower the time required to find a viable solution to the problem. The heuristics considered in Chapter

4 all exploit the fact that the WRMSP becomes tractable when only scheduling for one or two vehicles. Several heuristics based on this notion were proposed, which were termed Add-One-At-A-Time, Add-Two-At-A-Time, and two Remove and Reinsert heuristics. The Add-n-At-A-Time heuristics worked by adding n vehicles into the solution at every iteration, stopping when the number of vehicles added is equal to the fleet size bound established for the problem as a whole. The Remove and Reinsert heuristic builds on this idea, stopping every fixed number of iterations to remove an already scheduled vehicle from the solution, and reinserting it in an attempt to counteract any globally bad scheduling decisions which may have been made previously.

An analysis of the solution quality and runtimes obtained from these four algorithms was undertaken, with the results suggesting that the Remove and Reinsert heuristics, in particular the non-stochastic FIFO queue based heuristic, are generally the best choice, as they provide extremely high quality results, with a runtime that is only a small fixed constant worse than the fastest heuristic studied. In a practical sense, the runtimes of all of the heuristics except for the Add-Two-At-A-Time heuristic should be fast enough for use in most operational settings.

5.2 Major Contributions

We have developed a model that represents winter maintenance operations in a formal mathematical context. This model is flexible in terms of the operations it models, and also the metrics that it uses to establish optimal treatment plans. Such a model fills a niche that was otherwise vacant in the existing research, and as such we believe it to be a useful contribution to the field.

The WRMSP model allows for the efficient modelling and quantification of the overall system-wide effectiveness of a given treatment plan for an arbitrary weather event. This ability proves useful both for developing a plan of action in real-time during a storm event, and for estimating the effects of altering various system parameters such as fleet size or treatment policies. In addition to providing recommended schedules for vehicle dispatch, road condition estimates are also provided, which can prove useful for emergency response and public information purposes.

From an application standpoint, the WRMSP has the potential to function as an effective component of a decision support system, primarily in interpreting existing weather forecast information in a network aware manner. This research has provided the first steps towards this eventual goal, however, a great deal of work remains to be done before such deployments could become a reality. Accordingly, the Future Work section below is concerned primarily with extensions of the model in the direction of real-world applicability.

From a research standpoint, perhaps the principal contributions of this model to current research is the mathematical system that maps vehicle departures along a route to visitation times on individual road segments (discussed in Section 3.4). This particular system is a new work, and fulfils a function that was previously absent in the literature. This system has applications in many route optimisation fields, as it allows for the selection of a route from a fixed palette of choices rather than any arbitrary path through the network.

This work is the first in the literature that explicitly models winter maintenance effectiveness as a direct consequence of scheduling decisions, and as such serves as a framework for future work in this area. In particular, the network based structure of winter maintenance operations is formally represented in this model, an aspect that is lacking in most existing research in the field to date.

5.3 Future Work

5.3.1 Model Validation

In this research, we have attempted to produce a model that is both extensible and whose accuracy is quantifiable. Such quantification is a necessary step on the way to making this model deployable in a production setting. Although a formal validation of the model's fidelity to real-world operations has not been done, the foundations for such have been laid out in this research. At a minimum, the weather and cost impacts estimated by the model for a wide range of treatment plans should be compared against those experienced in a real-world execution of the same treatment plans. In this way, the model can be brought into closer alignment with the reality it seeks to represent.

5.3.2 Alternate Treatment Operations

In the initial stages of this research, the model was concerned exclusively with the operation of plowing, as it affords both a clear and easy to understand performance metric (snow depth) and an easy to model effect (plowing removes all snow present on the roadway at the time of plowing). In reality, however, many other treatment operations such as salting and anti-icing are used extensively in most jurisdictions, and often play a more important role in the overall road maintenance picture than does plowing. As such, it is critical that the WRMSP be extended to model alternative treatment types. This is a large undertaking, but should be able to be accomplished within the framework provided by the WRMSP.

Road salting, as discussed in Chapter 2, is a complex treatment to model. The effects of treatment are not immediately evident, and may persist for several hours as the chemical effects of the salt run their course. While models exist to quantify this effectiveness over time, they do not immediately meet all the criteria needed to be integrated into the WRMSP. Nonetheless, it is anticipated that they can be integrated into the WRMSP as part of future research.

5.3.3 Alternative Performance Metrics

Modelling treatment types other than plowing also introduces the problem of establishing an effective metric for measuring their performance. For example, reducing snow depth on a roadway is not the primary goal of salting operations, nor is it an effective direct measure of a salting operations' performance. Thus, alternative performance metrics such as percentage of bare pavement or the average coefficient of friction along a road segment should be considered. Indeed, such metrics may well represent a closer alignment of road condition estimation with driver perception of such condition (in other words, such measurements may more closely represent what drivers consider to be 'good' or 'bad' road conditions than the existing WRMSP metric of snow depth).

It should also be mentioned that the choice of metric used to determine performance need not be a physical quantity. For example, a numerical notion of the safety of a section of road could be used as a performance metric, provided that there was both an effective way to model the road condition using this metric, and that a direct and quantifiable change in this metric could be derived from a treatment event. Work is being done[12] that seeks to relate winter maintenance operations to accident rates, and to find a statistically valid relation between the two. Such a metric could conceivably be used in future work to construct maintenance schedules which directly seek to minimise predicted accident rates in the road network without the intermediary step of predicting snow depth or road coverage percentage. This would be an interesting development, and would likely make this work more attractive to prospective maintenance planners and policy makers.

5.4 Final Remarks

In developing the WRMSP, our goal was to lay a solid foundation for future work in both practical and research areas of winter maintenance. The results of this research are largely preliminary, and as such, the ultimate extent of the contribution of this work to the field remains to be seen. Although the WRMSP is still in its infancy, it is hoped that it will mature into a useful tool for both researchers and practitioners.

Appendix A

Mathematical Summary of the WRMSP

The following is a brief summary of the mathematical formulation of the WRMSP. For a detailed description of the formulation, please consult Chapter 3. Specifically, the formulation of the Q and S matrices is not defined here, as the description of their construction is beyond the scope of this summary. For summary purposes, it suffices to note that Q and S are used to map the route departure times indicated in X to road arrival and location times indicated in Y and Y', respectively.

minimise

$$\sum_{i=0}^{n-1}\sum_{t=0}^{T-1}\boldsymbol{\lambda}_i C_{i,t} \quad + \quad \beta\sum_{t=0}^{T-1}\sum_{k=0}^{m-1}\boldsymbol{d}_k X_{k,t} \tag{3.1}$$

subject to

$$C_{i,0} = \boldsymbol{w}_i + W_{i,0} - \delta Y'_{i,0} \qquad \forall i \in \{0,\dots,n-1\} \tag{3.9}$$

$$C_{i,t} = C_{i,t-1} + W_{i,t} - \delta Y'_{i,t} \qquad \forall i,t \in \{\{0,\dots,n-1\}\times\{0,\dots,T-1\}\} \tag{3.10}$$

$$C_{i,t} \le \boldsymbol{\phi}_i \qquad \forall i,t \in \{\{0,\dots,n-1\}\times\{0,\dots,T-1\}\} \tag{3.12}$$

$$\sum_{i=0}^{n-1} Y_{i,t} \le b \qquad \forall t \in \{0,\dots,T-1\} \tag{3.13}$$

$$Z_{k,i} = Q_{k,i}(X_k^T + P_k^T) \qquad \forall i,k \in \{\{0,\dots,n-1\}\times\{0,\dots,m-1\}\} \tag{3.3}$$

$$Y_{i,t} = \sum_{k=0}^{m-1} Z_{k,i,t} \qquad \forall i,t \in \{\{0,\dots,n-1\}\times\{0,\dots,T-1\}\} \tag{3.4}$$

$$Z'_{k,i} = S_{k,i}(X_k^T + P_k^T) \qquad \forall i,k \in \{\{0,\dots,n-1\}\times\{0,\dots,m-1\}\} \tag{3.6}$$

$$Y'_{i,t} = \sum_{k=0}^{m-1} Z'_{k,i,t} \qquad \forall i,t \in \{\{0,\dots,n-1\}\times\{0,\dots,T-1\}\} \tag{3.7}$$

$$C_{i,t} \in \mathbb{R}^+ \qquad \forall i,t \in \{\{0,\dots,n-1\}\times\{0,\dots,T-1\}\} \tag{3.11}$$

$$X_{k,t} \in \{0,1\} \qquad \forall t,k \in \{\{0,\dots,T-1\}\times\{0,\dots,m-1\}\}$$

$$Y_{i,t} \in \mathbb{Z}^+ \qquad \forall i,t \in \{\{0,\dots,n-1\}\times\{0,\dots,T-1\}\}$$

$$Y'_{i,t} \in \mathbb{Z}^+ \qquad \forall i,t \in \{\{0,\dots,n-1\}\times\{0,\dots,T-1\}\}$$

$$Z_{k,i,t} \in \mathbb{Z}^+ \qquad \forall i,t,k \in \{\{0,\dots,n-1\} \times \{0,\dots,T-1\}\times\{0,\dots,m-1\}\}$$

$$Z'_{k,i,t} \in \mathbb{Z}^+ \qquad \forall i,t,k \in \{\{0,\dots,n-1\} \times \{0,\dots,T-1\}\times\{0,\dots,m-1\}\}$$

Table A.1: Summary of variables in WRMSP

Variable	Domain	Description
$\boldsymbol{\lambda}_i$	$\mathbb{R}^+$	Importance of road segment i
$C_{i,t}$	$\mathbb{R}^+$	Condition of road segment i during time t
β	$\mathbb{R}^+$	Weighs cost of providing treatment against resultant road condition in solution
$\boldsymbol{d}_k$	$\mathbb{R}^+$	Cost of deploying a service vehicle along route k
$X_{k,t}$	$\{0,1\}$	Whether to dispatch a vehicle along route k at time t
$\boldsymbol{w}_i$	$\mathbb{R}$	Weather conditions at the beginning of time period 0 on road segment i
$W_{i,t}$	$\mathbb{R}$	Change in road conditions due to weather on road segment i during time t
δ	$\mathbb{R}$	Maximum change in road conditions due to a treatment event
$Y'_{i,t}$	$\mathbb{Z}^+$	Number of service vehicles beginning service of road segment i during time t
$\boldsymbol{\phi}_i$	$\mathbb{R}^+$	Maximum permissible road condition on road segment i
$Y_{i,t}$	$\mathbb{Z}^+$	Number of service vehicles on road segment i during time t
b	$\mathbb{Z}^+$	Number of service vehicles available
$Z_{k,i,t}$	$\mathbb{Z}^+$	Number of service vehicles on road segment i during time t due to route k
$Q_{k,i,t,t'}$	$\{0,1\}$	A table relating route departure times with vehicle locations
$P_{k,t}$	$\mathbb{Z}^+$	Number of vehicles pre-dispatched along route k at time t
$Z'_{k,i,t}$	$\mathbb{Z}^+$	Number of service vehicles beginning service on road segment i during time t due to route k
$S_{k,i,t,t'}$	$\{0,1\}$	A table relating route departure times with road segment arrival times

Appendix B

Implementation of the WRMSP

The WRMSP is implemented as an ILP specified in the AMPL programming language[13]. The use of AMPL allows for the use of standard ILP solvers while maintaining a clear and easy to read file format. The WRMSP program is solved by the Open Source GNU Linear Programming Kit solver[14], although any solver package which supports the AMPL language could be used in its place. The use of AMPL as a modelling language affords several practical benefits, most notably the formal separation of model and data information. This allows for the model to be expressed symbolically, and data for a particular instance of the problem to be expressed separately. AMPL facilitates this through the use of model files and data files. The model file for the WRMSP is included as Appendix C, and describes the WRMSP in terms of the input variables expressed in Chapter 3. Data files for the WRMSP problems described in Section 4.5 are included as Appendix D and Appendix E.

Appendix C

WRMSP Model File

```
#
# WRMSP.mod
#
# This AMPL / GNU Mathprog model file implements the WRMSP algorithm as an ILP.
#
# Data files to accompany this model file are REQUIRED to provide data for the
# following input parameters beta, delta, b, lambda, phi, l, and R (as well as
# the contents of the Segments, Routes, and Periods sets)
#
# Data files MAY provide the following input parameters
# w, P, W

#############################################
# Sets -- These sets describe the indices used to define arrays of input
# parameters
#############################################

# Segments is the set of road segments to model. Indexes based on Segments are
# denoted by i
set Segments;

# Routes is set of routes available to the scheduler. Indexes based on Routes
# are denoted by j
set Routes;

# Periods is the set of time periods to model. Indexes based on Periods are
# denoted by t
set Periods;

#############################################
# Parameters -- These values are required to be provided by an input data file,
# unless they have defaults associated with them in which case inputs may
# override the default values
#############################################
```

```
# The amount of weight assigned to the cost component of the objective function
param beta;

# The maximum change in road condition due to a single service event
param delta;

# The number of vehicles available to schedule
param b > 0 integer;

# The relative importance of each road segment, used to ascribe classes to roads
param lambda{i in Segments};

# Phi gives the upper bound on the condition of each road segment, used to
# ascribe service level constraints to each road section
param phi{i in Segments};

# The absolute value of initial road conditions for each road segment at time 0
param w{i in Segments} default 0;

# Previously deployed vehicles which we cannot plan for or control, do not
# contribute to fleet size restrictions, but do contribute to condition levels
# and cost estimates. Used by the incremental solver heuristics. Expressed as
# the time that the vehicles left on a particular route
param P{j in Routes, t in Periods} integer default 0;

# The changes in road condition during each period (expressed as a relative
# change for each time period) due to all non-controllable factors (weather,
# outside treatment, etc.)
param W{i in Segments, t in Periods} default 0;

# The length in time periods that a segment takes to traverse
param l{i in Segments};

# A table specifying the order in which segments appear in a given route, if at
# all. For a given pair of route and segment, the entry is set to the ordinal
# value of that segments occurrence order in the route. If a route does not
# contain a given segment, the entry is set to 0. See accompanying example data
# files for further clarification
param R{j in Routes, i in Segments};

##############################################
# Internally computed parameters -- These values are computed by the AMPL parser
# based on the data in the parameters above
##############################################

# Q is a 4D table of departure times. Entries in Q are set to 1 iff a service
# vehicle which begins route j at time tprime is on segment i at time t, and 0
# otherwise. This table, while crucial to the proper functionality of the rest
# of the algorithm, is built entirely from the entries in the routes and length
# input tables. Thus, the ILP itself sees the resultant 4D table of binary
# coefficients, while the user of this model sees only the input route and
# segment length tables. For details of the structure and function of this
# table, consult the master thesis document
param Q{j in Routes, i in Segments, t in Periods, tprime in Periods}
```

```
    binary :=
    if (
        R[j,i] > 0
    &&
        t >= tprime + sum{iprime in Segments} (
            if (R[j,iprime] > 0 && R[j,iprime] < R[j,i]) then
                l[iprime]
            else
                0
        )
    &&
        t < tprime + l[i] + sum{iprime in Segments} (
            if (R[j,iprime] > 0 && R[j,iprime] < R[j,i]) then
                l[iprime]
            else
                0
        )
    ) then
        1
    else
        0;

# S is based directly on Q, and consists of entries which are 1 iff the
# corresponding entry in Q is set to 1, and is the element of lowest index t
# that is so. We use S to define the instantaneous treatment time for segments.
# In the slice formed by fixing j and i, the row index k of the entry in the
# column set to 1 is the time at which segment i gets treated by a vehicle
# leaving on route j at time tprime. For details of the structure and function
# of this table, consult the master thesis document
param S{j in Routes, i in Segments, t in Periods, tprime in Periods}
    binary :=
    if (Q[j,i,t,tprime] = 1 && sum{u in 0..t} Q[j,i,u,tprime] = 1) then
        1
    else
        0;

# The cost in real terms to treat route j. Currently computed as the length of
# the route in time periods
param d{j in Routes} :=
    sum{i in Segments} if (R[j,i] > 0) then l[i] else 0;


##############################################
# Variables -- These are the variables which are manipulated by the AMPL solver
# to optimise the ILP
##############################################

# X is the departure table. Entries are set to 1 iff a vehicle should depart
# along the corresponding route at the corresponding time
var X{j in Routes, t in Periods} binary;

# C is the recursive condition table. Entries indicate the estimated road
# condition at each time interval
var C{i in Segments, t in Periods} >= 0;
```

```
#############################################
# ILP Definition -- This defines the actual structure of the WRMSP in terms of
# the above inputs
#############################################

# This objective function seeks to minimise a weighted combination of road
# conditions, road importance, and the cost of providing treatment
minimize impact:
    (sum{i in Segments, t in Periods}
        (lambda[i] * C[i,t]))
    + beta * (sum{j in Routes, t in Periods}
        (d[j] * (P[j,t] + X[j,t])));

# The following two constraints model the road condition values as a recursion.
# The first defines the base condition for the recursion, while the second
# defines the general recursion. It should be noted that the >= inequality will
# be forced to tightness by the fact that the objective function seeks to
# minimise C[] values wherever possible
subject to ConditionBase{i in Segments}:
    C[i,0] >= w[i] + W[i,0]
    - delta * (sum{j in Routes} (sum{u in Periods}
        (S[j,i,0,u] * (P[j,u] + X[j,u]))));

subject to ConditionRecursive{i in Segments, t in Periods : t <> 0}:
    C[i,t] >= C[i,t-1] + W[i,t]
    - delta * (sum{j in Routes} (sum{u in Periods}
        (S[j,i,t,u] * (P[j,u] + X[j,u]))));

# This constraint ensures that road segments never deteriorate beyond a given
# condition threshold. Cases where service levels cannot be met will therefore
# result in no feasible solution being found. This may not be the desired
# behaviour of the WRMSP for a particular task, and so it is common practice for
# this constraint to be removed
subject to ServiceConstraint{i in Segments, t in Periods}:
    C[i,t] <= phi[i];

# This constraint ensures that fleet size restrictions will be honoured for
# every time period, and that no more than b vehicles will ever be on the road
# at a time
subject to FleetSize{t in Periods}:
    sum{i in Segments} (sum{j in Routes} (sum{u in Periods}
        (Q[j,i,t,u] * X[j,u]))) <= b;

end;
```

Appendix D

WRMSP Data File for the City of Waterloo

```
#
# City of Waterloo case study data file
#
# This data file contains a street network and sample weather event to model
# a winter storm in a sub-section of the City of Waterloo. Intended to be used
# alongside a WRMSP AMPL model file. See master thesis document for details

# The set of road segments which comprise the road network
set Segments :=
67C-67C 67C-67B 67B-67C 67B-67B 67B-67A 67A-67B 67A-67A 67B-88C
88C-67B 88C-88C 88C-67A 67A-88C 67A-88B 88B-67A 88B-88B 88B-88C
88C-88B PY-88B 88B-PY 88B-88A 88A-88B 88A-88A 88B-98C 98C-88B
98C-98C 98C-98A 98A-98C 98A-98A 98A-98B 98B-98A 98B-98B 98B-88A
88A-98B 98B-98C 98C-98B;

# The set of time periods to model
set Periods :=
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17;

# The set of routes through the road network. Layout of the routes is described
# by the R parameter below. This set simply lists them by name
set Routes :=
R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11;

# Assign a 0.5 weight factor to the cost associated with providing treatment
param beta := 0.5;

# Assume the plow can remove an essentially arbitrary amount of snow
param delta := 2000;

# We have a single plow to schedule (in this case, we're being used as a
# sub-step of the add-1-at-a-time heuristic)
param b := 1;

# Lengths of the various road segments
```

```
param l :=
67C-67C 6
67C-67B 1
67B-67C 1
67B-67B 5
67B-67A 1
67A-67B 1
67A-67A 4
67B-88C 1
88C-67B 1
88C-88C 6
88C-67A 1
67A-88C 1
67A-88B 1
88B-67A 1
88B-88B 3
88B-88C 1
88C-88B 1
PY-88B  1
88B-PY  1
88B-88A 1
88A-88B 1
88A-88A 7
88B-98C 1
98C-88B 1
98C-98C 5
98C-98A 1
98A-98C 1
98A-98A 5
98A-98B 1
98B-98A 1
98B-98B 5
98B-88A 1
88A-98B 1
98B-98C 1
98C-98B 1;

# Importance values for each road segment -- in this example, they all have the
# same importance ascribed to them
param lambda :=
67C-67C 1
67C-67B 1
67B-67C 1
67B-67B 1
67B-67A 1
67A-67B 1
67A-67A 1
67B-88C 1
88C-67B 1
88C-88C 1
88C-67A 1
67A-88C 1
67A-88B 1
88B-67A 1
```

```
88B-88B 1
88B-88C 1
88C-88B 1
PY-88B  1
88B-PY  1
88B-88A 1
88A-88B 1
88A-88A 1
88B-98C 1
98C-88B 1
98C-98C 1
98C-98A 1
98A-98C 1
98A-98A 1
98A-98B 1
98B-98A 1
98B-98B 1
98B-88A 1
88A-98B 1
98B-98C 1
98C-98B 1;

# Service levels for each road segment, expressed as road condition thresholds
param phi :=
67C-67C 20
67C-67B 20
67B-67C 20
67B-67B 20
67B-67A 20
67A-67B 20
67A-67A 20
67B-88C 20
88C-67B 20
88C-88C 20
88C-67A 20
67A-88C 20
67A-88B 20
88B-67A 20
88B-88B 20
88B-88C 20
88C-88B 20
PY-88B  20
88B-PY  20
88B-88A 20
88A-88B 20
88A-88A 20
88B-98C 20
98C-88B 20
98C-98C 20
98C-98A 20
98A-98C 20
98A-98A 20
98A-98B 20
98B-98A 20
```

```
98B-98B 20
98B-88A 20
88A-98B 20
98B-98C 20
98C-98B 20;

# Weather condition deltas for each road segment
param W
:       0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 :=
67C-67C 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
67C-67B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
67B-67C 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
67B-67B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
67B-67A 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
67A-67B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
67A-67A 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
67B-88C 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88C-67B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88C-88C 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88C-67A 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
67A-88C 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
67A-88B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88B-67A 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88B-88B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88B-88C 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88C-88B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
PY-88B  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88B-PY  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88B-88A 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88A-88B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88A-88A 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88B-98C 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
98C-88B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
98C-98C 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
98C-98A 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
98A-98C 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
98A-98A 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
98A-98B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
98B-98A 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
98B-98B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
98B-88A 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
88A-98B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
98B-98C 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
98C-98B 5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0;

# Route table which expresses the various route structures in terms of the order
# in which they are visited within each route
param R
: 67C-67C 67C-67B 67B-67C 67B-67B 67B-67A 67A-67B 67A-67A 67B-88C
88C-67B 88C-88C 88C-67A 67A-88C 67A-88B 88B-67A 88B-88B 88B-88C
88C-88B PY-88B 88B-PY 88B-88A 88A-88B 88A-88A 88B-98C 98C-88B
98C-98C 98C-98A 98A-98C 98A-98A 98A-98B 98B-98A 98B-98B 98B-88A
88A-98B 98B-98C 98C-98B :=
R1  0 0 0 4 0 0 0 5 3 0 0 0 0 0 0 2 6 1 7 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
R2   0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 1 3 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R3   0 0 0 0 0 0 3 0 0 0 0 0 4 2 0 0 0 1 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R4   0 0 0 0 0 0 0 0 0 3 0 0 0 0 0 2 4 1 5 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R5   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 7 0 0 0 2 6 0 3 5 4 0 0 0 0 0 0 0
R6   0 0 0 0 5 4 0 0 0 0 3 6 0 0 0 2 7 1 8 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R7   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 5 0 0 0 2 4 3 0 0 0 0 0 0 0 0 0 0
R8   0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 7 2 6 0 0 0 0 0 0 0 0 0 4 5 3 0 0
R9   5 6 4 0 0 0 0 7 3 0 0 0 0 0 0 2 8 1 9 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R10  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 5 2 4 3 0 0 0 0 0 0 0 0 0 0 0 0 0
R11  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 8 0 0 0 2 7 0 0 0 0 5 4 0 0 0 6 3;

# List of pre-established route departures to assume as being present. Used to
# enable incremental heuristics, but initially empty
param P
:   0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17:=
R1  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
R2  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
R3  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
R4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
R5  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
R6  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
R7  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
R8  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
R9  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
R10 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
R11 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0;

end;
```

Appendix E

WRMSP Data File for a Rural Network

```
#
# Rural road network case study data file
#
# This data file contains a street network and sample weather event to model
# a winter storm in a semi-fictitious rural road network. Intended to be used
# alongside a WRMSP AMPL model file. See master thesis document for details

# The set of road segments which comprise the road network
set Segments :=
QA AQ AB BA BF FB AD DA BE EB DE ED EF FE DG GD FK KF GG GH HG HJ JH JK KJ
GL LG HM MH JN NJ KP PK LM ML MN NM NP PN RR LS SL PT TP QR RQ RS SR ST TS;

# The set of time periods to model
set Periods :=
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17;

# The set of routes through the road network. Layout of the routes is described
# by the R parameter below. This set simply lists them by name
set Routes :=
R1 R2 R3 R4 R5 R6 R7 R8 R9 R10 R11;

# Assign a 0.5 weight factor to the cost associated with providing treatment
param beta := 0.5;

# Assume the plow can remove an essentially arbitrary amount of snow
param delta := 2000;

# We have a single plow to schedule (in this case, we're being used as a
# sub-step of the add-1-at-a-time heuristic)
param b := 1;

# Lengths of the various road segments
param l :=
QA 3
AQ 3
AB 1
BA 1
```

```
BF 2
FB 2
AD 1
DA 1
BE 1
EB 1
DE 1
ED 1
EF 1
FE 1
DG 1
GD 1
FK 1
KF 1
GG 1
GH 1
HG 1
HJ 1
JH 1
JK 1
KJ 1
GL 1
LG 1
HM 1
MH 1
JN 1
NJ 1
KP 1
PK 1
LM 1
ML 1
MN 1
NM 1
NP 1
PN 1
RR 1
LS 1
SL 1
PT 1
TP 1
QR 2
RQ 2
RS 1
SR 1
ST 1
TS 1;

# Importance values for each road segment -- in this example, they all have the
# same importance ascribed to them
param lambda :=
QA 1
AQ 1
AB 1
BA 1
```

```
BF 1
FB 1
AD 1
DA 1
BE 1
EB 1
DE 1
ED 1
EF 1
FE 1
DG 1
GD 1
FK 1
KF 1
GG 1
GH 1
HG 1
HJ 1
JH 1
JK 1
KJ 1
GL 1
LG 1
HM 1
MH 1
JN 1
NJ 1
KP 1
PK 1
LM 1
ML 1
MN 1
NM 1
NP 1
PN 1
RR 1
LS 1
SL 1
PT 1
TP 1
QR 1
RQ 1
RS 1
SR 1
ST 1
TS 1;
```

 # Service levels for each road segment, expressed as road condition thresholds
param phi :=

```
QA 20
AQ 20
AB 20
BA 20
BF 20
```

```
FB 20
AD 20
DA 20
BE 20
EB 20
DE 20
ED 20
EF 20
FE 20
DG 20
GD 20
FK 20
KF 20
GG 20
GH 20
HG 20
HJ 20
JH 20
JK 20
KJ 20
GL 20
LG 20
HM 20
MH 20
JN 20
NJ 20
KP 20
PK 20
LM 20
ML 20
MN 20
NM 20
NP 20
PN 20
RR 20
LS 20
SL 20
PT 20
TP 20
QR 20
RQ 20
RS 20
SR 20
ST 20
TS 20;

# Weather condition deltas for each road segment
param W
:   0  1  2  3  4  5  6  7  8  9  10 11 12 13 14 15 16 17 :=
QA  0  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0
AQ  0  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0
AB  0  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0
BA  0  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0
BF  0  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0
```

```
FB  0  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0
AD  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
DA  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
BE  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
EB  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
DE  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
ED  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
EF  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
FE  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
DG  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
GD  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
FK  0  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0
KF  0  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0
GG  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
GH  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
HG  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
HJ  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
JH  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
JK  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
KJ  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
GL  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
LG  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
HM  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
MH  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
JN  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
NJ  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
KP  0  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0
PK  0  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0
LM  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
ML  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
MN  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
NM  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
NP  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
PN  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
RR  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
LS  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
SL  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
PT  0  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0
TP  0  0  0  0  0  0  0  0  0  0  0   0  0  0  0  0  0  0
QR  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
RQ  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
RS  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
SR  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
ST  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0
TS  5  6  7  8  9  11 13 13 12 11 10  8  6  4  4  3  2  0;

# Route table which expresses the various route structures in terms of the order
# in which they are visited within each route
param R
:  QA AQ AB BA BF FB AD DA BE EB DE ED EF FE DG GD FK KF GG GH HG HJ
JH JK KJ GL LG HM MH JN NJ KP PK LM ML MN NM NP PN RR LS SL PT TP QR
RQ RS SR ST TS :=
R1  0 0 0 4 0 0 5 0 0 3 6 0 7 2 0 0 8 1 0 0 0 0 0 0 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
```

```
R2  0 0 0 0 0 0 7 6 4 3 0 5 0 2 8 0 0 1 9 10 0 11 0 12 0 0
    0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
R3  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 0 2 0 1 4
    0 7 6 10 9 0 12 5 0 8 0 11 0 0 0 0 0 0 0 0 0 0 0 0
R4  0 0 8 0 0 0 0 7 9 0 0 0 10 0 0 6 11 0 0 0 0 0 0 0 0 0
    5 0 0 0 0 1 0 0 4 0 3 0 2 0 0 0 0 0 0 0 0 0 0 0
R5  0 0 0 0 0 0 0 0 0 0 7 0 8 0 0 6 9 0 0 0 0 0 0 0 0 0
    5 0 0 0 0 1 0 0 0 0 0 0 0 0 0 4 2 0 0 0 0 0 0 3
R6  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0
    0 0 0 2 0 0 13 0 4 0 3 0 0 7 5 0 0 12 9 8 10 6 11 0
R7  0 0 0 0 0 0 0 0 0 0 0 3 0 2 4 0 0 1 5 6 0 0 0 14 0 0
    0 7 0 0 13 0 0 0 8 0 0 0 12 0 9 0 0 11 0 0 0 0 10 0
R8  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
    0 0 0 0 0 1 8 5 0 6 0 7 0 0 0 4 2 0 0 0 0 0 0 3
R9  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 3 14 2 15 1 4
    0 0 13 0 0 0 0 12 0 0 0 0 0 9 5 11 0 0 8 7 10 6 0 0
R10 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 1 10
    9 3 0 0 0 0 14 11 4 12 0 13 0 0 5 8 0 0 0 0 0 0 6 7
R11 0 0 0 0 0 0 11 10 14 13 12 3 15 2 4 9 16 1 0 0 0 0 0 0 0 5
    8 0 0 0 0 0 0 0 0 0 0 0 0 0 6 7 0 0 0 0 0 0 0 0
```

 # List of pre-established route departures to assume as being present. Used to
 # enable incremental heuristics, but initially empty

param P

```
:   0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 :=
R1  0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0
R2  0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0
R3  0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0
R4  0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0
R5  0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0
R6  0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0
R7  0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0
R8  0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0
R9  0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0
R10 0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0
R11 0 0 0 0 0 0 0 0 0 0 0  0  0  0  0  0  0  0;
```

end;

Appendix F

Details of Heuristic Analysis

Tests to compare various approximation heuristics were run using the WRMSP model file included in Appendix C, using the various heuristics described in Section 4.5, and input data based on the that presented in Appendices D and E. Tables F.1 and F.2 detail the results of those runs, which are also summarised in Figures 4.8 and 4.7.

These tests were run on an AMD Athlon 1.25GHz computer with 256MB of memory, using the GLPK[14] libraries running on Mandrake Linux 10.1[15].

Table F.1: Urban case study numerical results

Heuristic	Fleet Size	Time To Solve (s)	Objective Value
Add-One-At-A-Time	1	13	4.10725e04
	2	22	3.6105e04
	3	33	3.2036e04
	4	42	2.8287e04
	5	56	2.53235e04
	6	67	2.25545e04
	7	81	2.0457e04
	8	97	1.8683e04
	9	117	1.72175e04
	10	136	1.58895e04
	11	160	1.4818e04
	12	189	1.39485e04

Table F.1: Urban case study numerical results (continued)

Heuristic	Fleet Size	Time To Solve (s)	Objective Value
Add Two-At-A-Time	2	1122	3.5306e04
	4	2914	2.78985e04
	6	4996	2.23835e04
	8	7610	1.84485e04
	10	13929	1.5789e04
	12	25376	1.38355e04
Remove FIFO	1	13	4.10725e04
	2	22	3.6105e04
	3	43	3.2008e04
	4	57	2.82275e04
	5	79	2.4985e04
	6	99	2.26605e04
	7	123	2.0529e04
	8	148	1.88465e04
	9	184	1.7377e04
	10	206	1.6174e04
	11	237	1.49935e04
	12	255	1.3891e04

Table F.1: Urban case study numerical results (continued)

Heuristic	Fleet Size	Time To Solve (s)	Objective Value
Remove Random	1	13	4.10725e04
	2	22	3.6105e04
	3	41	3.20235e04
	4	48	2.80635e04
	5	71	2.49575e04
	6	87	2.2333e04
	7	118	2.01925e04
	8	139	1.85715e04
	9	176	1.72435e04
	10	200	1.60435e04
	11	249	1.48475e04
	12	277	1.3998e04
Abbreviated Branch and Bound	1	13	4.10725e04
	2	600	3.5306e04
	3	600	3.1672e04
	4	600	2.87745e04
	5	600	2.70745e04
	6	600	2.4939e04
	7	600	2.3969e04
	8	600	2.1377e04
	9	600	1.85745e04
	10	600	1.6972e04
	11	600	1.49155e04
	12	600	1.45045e04

Table F.1: Urban case study numerical results (continued)

Heuristic	Fleet Size	Time To Solve (s)	Objective Value
Linear Relaxation	1	1	2.904617e04
	2	1	1.850113e04
	3	1	1.194955e04
	4	1	7.55978e03
	5	1	5.0512225e03
	6	1	3.73789e03
	7	1	2.9757775e03
	8	1	2.8964975e03
	9	1	2.8964975e03
	10	1	2.8964975e03
	11	1	2.8964975e03
	12	1	2.8964975e03

Table F.2: Rural case study numerical results

Heuristic	Fleet Size	Time To Solve (s)	Objective Value
Add-One-At-A-Time	1	10	4.5026e04
	2	23	3.72005e04
	3	39	3.14685e04
	4	64	2.68335e04
	5	86	2.25135e04
	6	116	1.94705e04
	7	158	1.73225e04
	8	192	1.54325e04
	9	228	1.355e04
	10	281	1.25475e04
	11	335	1.16375e04
	12	380	1.08515e04
Add-Two-At-A-Time	2	1270	3.68025e04
	4	3693	2.59765e04
	6	6611	1.9507e04
	8	9330	1.5752e04
	10	12094	1.31985e04
	12	14597	1.11545e04

Table F.2: Rural case study numerical results (continued)

Heuristic	Fleet Size	Time To Solve (s)	Objective Value
Remove FIFO	1	10	4.5026e04
	2	23	3.72005e04
	3	48	3.08705e04
	4	70	2.63155e04
	5	122	1.9184e04
	6	162	1.7054e04
	7	237	1.51975e04
	8	280	1.3798e04
	9	340	1.2544e04
	10	377	1.1609e04
	11	445	1.0704e04
	12	487	9.999e03
Remove Random	1	10	4.5026e04
	2	23	3.72005e04
	3	39	3.14685e04
	4	92	2.63155e04
	5	107	2.21545e04
	6	169	1.91115e04
	7	206	1.70255e04
	8	280	1.46935e04
	9	322	1.3426e04
	10	400	1.23625e04
	11	438	1.15155e04
	12	524	1.07485e04

Table F.2: Rural case study numerical results (continued)

Heuristic	Fleet Size	Time To Solve (s)	Objective Value
Abbreviated Branch and Bound	1	10	4.5026e04
	2	600	3.69735e04
	3	600	3.13495e04
	4	600	2.56205e04
	5	600	2.42105e04
	6	600	2.094905e04
	7	600	1.73725e04
	8	600	1.62615e04
	9	600	1.51715e04
	10	600	1.4114e04
	11	600	
	12	600	
Linear Relaxation	1	1	2.9486665e04
	2	1	1.37132335e04
	3	1	6.702220833e03
	4	1	4.855105e03
	5	1	4.1597125e03
	6	1	4.055295e03
	7	1	4.055295e03
	8	1	4.055295e03
	9	1	4.055295e03
	10	1	4.055295e03
	11	1	4.055295e03
	12	1	4.055295e03

Bibliography

[1] Russell Alger, Edward Adams, and Phillip Beckwith. Anti-icing study: Controlled chemical treatments. Technical Report SHRP-H-683, Michigan Technological University, 1994.

[2] J. Andrey, B. Mills, B. Jones, R. Hass, and W Hamlin. Adaptation to climate change in the canadian transportation sector. Technical report, Natural Resources Canada, Adaptation Liaison Office, 1999.

[3] Lennart Axelson and Ronald Wahlgreen. Expert system based road salting - a preliminary study and a prototype. Technical report, Swedish National Road Administration, 1991.

[4] MDSS National Laboratory Consortium. Maintenance decision support system (http://www.rap.ucar.edu/projects/rdwx_mdss/).

[5] MDSS National Laboratory Consortium. The maintenance decision support system (MDSS) project: MDSS prototype release-3 technical description. Technical report, MDSS National Laboratory Consortium, 2004.

[6] Stantec Consulting. *Winter Maintenance Policy and Procedures, Municipalities of the Geographic Region of Waterloo.*

[7] University Corporation for Atmospheric Research. The unidata local data manager suite.

[8] Frederick Hillier and Gerald Lieberman. *Introduction to Operations Research.* Holden-Day Inc., 3rd edition, 1980.

[9] R. Jordan. A one-dimensional temperature model for a snowcover: Technical documentation for sntherm.89. Technical Report CRREL Special Report 91-16, U.S. Army Cold Regions Research and Engineering Laboratory, Hanover, N.H., 1991.

[10] S. Ketcham, D. Minsk, R. Blackburn, and E. Fleege. Manual of practice for an effective anti-icing program. Technical report, United States Federal Highway Administration, 1995.

[11] J Hode Keyser. *Handbook of Snow*, chapter Chemicals and Abrasives for Snow and Ice Control, pages pp. 580–612. Pergamon Press, 1981.

[12] Fu L. Quantifying effects of winter storms on safety and mobility. Technical report, Department of Civil Engineering, University of Waterloo, 2004.

[13] Bell Labs. The AMPL programming language (http://www.ampl.com).

[14] Andrew Makhorin. The GNU Linear Programming Kit (http://www.gnu.org/software/glpk/glpk.html).

[15] Mandrakesoft. Mandrake linux (http://www.mandrakelinux.com).

[16] Transportation Asssociation of Canada. *Salt SMART Training Guuide*. Ottawa, Ontario, 2004.

[17] Ontario Ministry of Transportation. Maintenance technology project 2003-2004 overview.

[18] Ontario Ministry of Transportation. Maintenance technology project 2003-2004 handout. Technical report, Ontario Ministry of Transportation, 2004.

[19] Kynric Pell. An improved displacement snowplow. Technical report, Department of Mechanical Engineering, University of Wyoming, 1994.

[20] Max Perchanok, George Comfort, and Roland Wahlgren. Decision support system for winter maintenance: Fesibility demonstration. Technical report, Aurora Consortium, 2000.

[21] Max Perchanok, Curt Pape, Lee Smithson, Dan Eriksson, Tim Bullock, and Alfred Uzokwe. Decision support system for winter maintenance: The DART database. Technical report, AURORA Consortium, 2002.

[22] M.S. Perchanok. Design and maintenance procedures to minimise impacts from drifting snow on highways - provisional guidelines. Technical report, Ontario Ministry of Transportation, 1998.

[23] M.S. Perchanok, D.G. Manning, and J.J. Armstrong. Highway de-icers: Standards, practice, and research in the province of ontario. Technical report, Ontario Ministry of Transportation, 1991.

[24] Aurora Program. Aurora program website (http://www.aurora-program.org/).

[25] SHRP Anti-Icing / RWIS Lead States Team. 1999-2000 technology transfer and usage survey results. Technical report, Strategic Highway Research Program, 2000.

[26] Mathieu Trudel. Minutes of meeting with phil hewittson, city of waterloo maintenance manager, May 2004.

[27] Mathieu Trudel. A real-time scheduling model for winter road maintenance operations. In *Proceedings of the 84th Annual Meeting of the Transportation Research Board*. Transportation Research Board, 2004.