Improving Striping Operations through System Optimization

Prepared by
Ronald McGarvey, Ph.D. (principal investigator)
Tim Matisziw, Ph.D. (co-principal investigator)
Charles Nemmers, P.E. (co-principal investigator)
Jim Noble, Ph.D., P.E. (co-principal investigator)
Center for Excellence in Logistics and Distribution, University of Missouri-Columbia

Final Report Prepared for Missouri Department of Transportation
2015 September Project TR201501 Report cmr16-003

TECHNICAL REPORT DOCUMENTATION PAGE

1. Report No. cmr 16-003	**2. Government Accession No.**	**3. Recipient's Catalog No.**
4. Title and Subtitle Improving Striping Operations through System Optimization		**5. Report Date** September 2015 **6. Performing Organization Code**
7. Author(s) Ronald G. McGarvey, Ph.D. http://orcid.org/0000-0002-8825-8553; Tim Matisziw, Ph.D.; Charles Nemmers, P.E.; Jim Noble, Ph.D., P.E.; Gokhan Karakose; Marina Materikina; and Alec Page		**8. Performing Organization Report No.**
9. Performing Organization Name and Address Center for Excellence in Logistics and Distribution (CELDi) University of Missouri E3437 Lafferre Hall Columbia, MO 65211		**10. Work Unit No.** **11. Contract or Grant No.** MoDOT Research Project #TR201501 USDOT/OST-R Grant #DTRT13-G-UTC37
12. Sponsoring Agency Name and Address Midwest Transportation Center 2711 S. Loop Drive, Suite 4700 Ames, IA 50010-8664 Missouri Department of Transportation (SPR) http://dx.doi.org/10.13039/100007251 Construction and Materials Division Research Section P.O. Box 270/1617 Missouri Blvd. Jefferson City, MO 65102 U.S. Department of Transportation http://dx.doi.org/10.13039/100000140 Office of the Assistant Secretary for Research and Technology 1200 New Jersey Avenue, SE Washington, DC 20590		**13. Type of Report and Period Covered** Final Report (June 2014 to July 2015) **14. Sponsoring Agency Code**

15. Supplementary Notes

Visit http://www.intrans.iastate.edu/ for color pdfs of this and other research reports. This report is available at http://www.intrans.iastate.edu/mtc/index.cfm/research/project/project/2139436905. MoDOT research reports are available in the Innovation Library at http://www.modot.org/services/or/byDate.htm. This report is also available at http://library.modot.mo.gov/RDT/reports/TR201501/cmr16-003.pdf.

16. Abstract

Striping operations generate a significant workload for Missouri Department of Transportation (MoDOT) maintenance operations. The requirement for each striping crew to replenish its stock of paint and other consumable items from a bulk storage facility, along with the necessity to make several passes on most of the routes to stripe all the lines on that road, introduce the potential for inefficiencies in the form of "deadhead miles" that striping crew vehicles must travel while not actively applying pavement markings. These inefficiencies generate unnecessary travel, wasted time, and vehicle wear.

The research detailed in this report provides an optimization-based approach to determining a striping schedule that minimizes these deadhead miles. A computer program was developed for scheduling and routing road striping operations. This report contains details on the theoretical foundations of this optimization model, along with a user's guide that details the preparation of input data necessary to utilize this computer program and step-by-step instructions on the use of the model.

17. Key Words Fleet management; Fleet operations; Optimization models. TRT terms: Decision support systems; Highway operations; Pavement maintenance; Road marking materials; Road markings; Striping		**18. Distribution Statement** No restrictions. This document is available through the National Technical Information Service, Springfield, VA 22161.	
19. Security Classif. (of this report) Unclassified.	**20. Security Classif. (of this page)** Unclassified.	**21. No. of Pages** 79	**22. Price**

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

DISCLAIMER STATEMENT

The opinions, findings, and conclusions expressed in this document are those of the investigators. They are not necessarily those of the Missouri Department of Transportation, U.S. Department of Transportation, or Federal Highway Administration. This information does not constitute a standard or specification.

IMPROVING STRIPING OPERATIONS THROUGH SYSTEM OPTIMIZATION

Final Report
September 2015

Principal Investigator
Ronald G. McGarvey, Assistant Professor
University of Missouri

Co-Principal Investigators
Timothy Matisziw, Associate Professor
Charles Nemmers, Program Director, Transportation Infrastructure Center
James Noble, Professor, University of Missouri

Research Assistants
Gokhan Karakose, Marina Materikina, and Alec Page

Sponsored by
the Missouri Department of Transportation,
the Midwest Transportation Center, and
the U.S. Department of Transportation
Office of the Assistant Secretary for Research and Technology

A report from
Institute for Transportation
Iowa State University
2711 South Loop Drive, Suite 4700
Ames, IA 50010-8664
Phone: 515-294-8103 / Fax: 515-294-0467
www.intrans.iastate.edu

TABLE OF CONTENTS

ACKNOWLEDGMENTS ix

1. INTRODUCTION 1

    1.1 MoDOT Operations 1
    1.2 Road Types 3
    1.3 Type of Lines 3
    1.4 Material used for Pavement Marking 5
    1.5 Striping Operations 5
    1.6 Striping Plan 7

2. LITERATURE REVIEW 9

    2.1 Road Striping Operations 9
    2.2 Network Optimization Models 9
    2.3 Genetic Algorithm 11

3. DATA ANALYSIS 14

    3.1 Preparation of ArcGIS data 14
    3.2 Difficult Segments 19
    3.3 Calculation of Number of Passes 23
    3.4 Final Excel Data File: MoDOT Roads 29
    3.5 Overnight Location Distances File 31

4. GENETIC ALGORITHM MODEL 39

    4.1 Brief Description of the GA 39
    4.2 Installation Process of Required Programs and Libraries 41
    4.3 Using the Program Interface 50

5. CONCLUSIONS 60

REFERENCES 61

APPENDIX A: LP OPTIMIZATION MODEL 65

    A.1 Multi-pass RPP Model 65
    A.2 Slow-moving Multi-pass Postman Problem with Overnighting 66

LIST OF FIGURES

Figure 1.1 – Classification of lines based on road type 5
Figure 1.2 – Striping communications system.....6
Figure 1.3 – Striper 7
Figure 1.4 – Old striping plan 8
Figure 1.5 – New striping plan 8
Figure 3.1 – Missouri road network.....14
Figure 3.2 – Central District network with arcs and nodes.....17
Figure 3.3 – Attribute table of D4Roads file 18
Figure 3.4 – Example of "difficult segment" (end and start points are different) 20
Figure 3.5 – Example of "difficult" segment (# of lanes is not the same).....21
Figure 3.6 – Example of "difficult" segment (# of passes and end/start points are different).....22
Figure 3.7 – Example of how road looks in reality with different # of lanes on each side 22
Figure 3.8 – Example of disconnected segment 23
Figure 3.9 – Identification of segments that have centerline only.....26
Figure 3.10 – Part of final Excel data file.....30
Figure 3.11 – Map with overnight locations.....32
Figure 3.12 – Creating the feature class in ArcGIS 33
Figure 3.13 – Loading facilities step.....34
Figure 3.14 – Allocation of facilities and incidents.....35
Figure 3.15 – Solution of closest facility tool.....36
Figure 3.16 – Attribute table of Routes 37
Figure 3.17 – Final Excel file with closest overnight locations to each node 38
Figure 4.1 – Binary crossover.....39
Figure 4.2 – Insertion mutation.....40
Figure 4.3 – Exchange mutation 40
Figure 4.4 – Direction mutation.....40
Figure 4.5 – Installing Python 2.7.2.....41
Figure 4.6 – Installing pypy 42
Figure 4.7 – Preparing pypy on the desktop 42
Figure 4.8 – Adding Python 2.7 to the path variable 43
Figure 4.9 – Adding MoDOT model to the path variable.....44
Figure 4.10 – Installing numpy library 44
Figure 4.11 – Installing steuptools library 45
Figure 4.12 – Downloading PIL library.....45
Figure 4.13 – Adding scripts to the path variable 46
Figure 4.14 – Installing networkx 46
Figure 4.15 – Installing xlrd.....47
Figure 4.16 – Installing xlwt 47
Figure 4.17 – Installing openpyxl 47
Figure 4.18 – Downloading xlrd for pypy 48
Figure 4.19 – Installing xlrd for pypy 48
Figure 4.20 – Finalizing initialization.....49
Figure 4.21 – User interface.....50
Figure 4.22 – Adding new road segments 51

Figure 4.23 – Saving new road segments 51
Figure 4.24 – Confirming network connectivity after adding new road segments.....52
Figure 4.25 – Calculating shortest distances table.....53
Figure 4.26 – Input Data section.....53
Figure 4.27 – Selecting counties for striping scheduling.....54
Figure 4.28 – Selecting initial location of striping crew.....54
Figure 4.29 – Selecting road types for striping scheduling 55
Figure 4.30 – Selecting maximum-allowable computational time 55
Figure 4.31 – Running the model 55
Figure 4.32 – Selecting the road network 56
Figure 4.33 – Selecting distance table for overnighting locations.....56
Figure 4.34 – Selecting the distance table 57
Figure 4.35 – Display when model has finished building running 57
Figure 4.36 – Opening the model output 58
Figure 4.37 – Performing what-if analyses.....59

LIST OF TABLES

Table 3.1 – Calculation of number of passes for divided roads.....24
Table 3.2 – Calculation number of passes for undivided roads.....25
Table 3.3 – Calculation of number of passes for “difficult” segments.....27

ACKNOWLEDGMENTS

The authors would like to thank the Missouri Department of Transportation (MoDOT), the Midwest Transportation Center at Iowa State University, and the U.S. Department of Transportation Office of the Assistant Secretary for Research and Technology for sponsoring this research.

1. INTRODUCTION

1.1 MoDOT Operations

The Missouri Department of Transportation (MoDOT) is responsible for maintaining nearly 34,000 miles of highways and over 10,000 bridges; as a result, Missouri maintains "the nation's seventh largest state highway system" with more miles than the combined systems of Iowa, Nebraska, and Kansas (MoDOT 2015a). One of the annual maintenance activities conducted by MoDOT is road striping, which involves the application of markings (primarily paint) to define lanes and other pieces of traffic related information. According to Montebello and Schroeder (2000), most variations of paint used during striping operations have an estimated life between 9 and 36 months with variation largely dependent on traffic volume. Each year, MoDOT stripe more than 60,000 line-miles of road on a scheduled basis. The majority of the miles of highway maintained by MoDOT are high-volume roadways that necessitate annual striping. Additionally, citizens may place a request for a certain road to receive striping earlier than originally planned; such requests are generally given a high priority, with MoDOT attempting to complete the striping within a few days' time.

Striping operations provide important information while allowing minimal diversion of attention from the roadway. Striping operations include:

- Obliteration of pavement markings (removing existing or temporary pavement marking, which is conflicting or might mislead traffic)
- Application of permanent pavement markings after construction or maintenance of roads
- Removal of permanent marking
- Line-striping for all major and minor roads that require it.
- Symbol Markings, turn markings etc.
- Maintenance of striped lines (keeping track of lines conditions, making a decision, which road should be striped immediately and this year)

For MoDOT, coordinating a plan to accommodate the striping of both major and minor highways on an annual basis represents a significant logistical challenge. In addition, maintenance activities accounted for roughly 21% of expenditures in 2014 (MoDOT 2015a). Therefore, increasing efficiency in striping operations represents a substantial opportunity to decrease annual expenses by MoDOT. Inefficient scheduling can create an excess of "deadhead miles" in which striping crews must travel while not actively striping roads. Minimizing "deadhead miles" is an important aspect of reducing the waste of extraneous travel, time, and vehicle wear.

This report addresses the scheduling of striping operations for a subset of MoDOT roads; namely, those located in the Central District of Missouri. In general, the task of crafting a schedule for striping operations mirrors elements of the Rural Postman Problem, since the objective involves finding the route of minimum cost that traverses a subset of the arcs in a network (Eiselt et. al. 1995a and Monroy-Licht et. al. 2013). For MoDOT, cost is a function of

the distance traveled; therefore, the route of minimum cost is equal to the route of shortest total length, which involves minimizing "deadhead miles."

The task is further complicated due to potential issues such as the restricted interval of time available to complete the striping process for the entire state. For most variations of paint used in striping operations, the temperature of both the air and pavement must be above 50° F (Montebello and Schroeder 2000). In keeping with this environmental constraint, MoDOT generally limits striping operations to the period between March and October of each year. Additional constraints regarding striping operations may present further complications; for example, the varying widths of road segments can require multiple passes (as opposed to the single pass typically assigned in the Rural Postman Problem). As such, a model of striping operations for MoDOT needs to modify the traditional Rural Postman Problem formulation.

In order to build an optimization model representation of this network, certain constraints must be addressed regarding the procedures of striping operations. In general, these constraints include (MoDOT 2015b):

1. MoDOT striping crews can only travel on MoDOT maintained roads. Due to concerns regarding lane width or road quality, MoDOT restricts travel by striping crews to Interstates, Numbered/Lettered Highways, or other roads in which the safe travel of striping vehicles can be ensured. As such, a network formulation should consider only those segments for which general maintenance is the responsibility of MoDOT.
2. Road segments may require a single pass or multiple passes based on the number of lanes, the classification of the road segment, and the presence of certain road features such as a white edge line delineator.
3. Traffic lanes cannot be striped in a direction opposing the normal flow of traffic. In the case of a two-lane road requiring two passes, this restriction indicates that the two passes must occur in opposing directions. As such, each road segment included in the system required a description of both the number and direction of passes in the system.
4. Approximately 10 hours are available each workday for striping crews to actively mark roads.
5. The speed at which striping vehicles travel can vary depending on whether the road needs to be striped and the type of road in which travel is taking place. In general, striping vehicles travel at a speed around 8 miles per hour when striping a divided highway, around 10 miles per hour when striping an undivided highway (the speed difference is primarily due to different physical attributes of the markings applied in each case), and around 35 miles per hour when traveling while deadheading or not actively striping a road.
6. At the end of each workday, striping vehicles must travel to a maintenance building to remain overnight. At the beginning to the next workday, the striping vehicles will return to service, but the striping crews do not necessarily need to begin from the point work ceased during the previous workday. As such, formulation of the model had to account for the travel to and from each maintenance building or overnight location within the 10-hour workday based on a 35 mile per hour deadhead speed limit as an element of the striping schedule.

To accommodate the aforementioned conditions, the following data elements were deemed critical in order to effectively model the preliminary representation of MoDOT striping operations conducted across the Central District of Missouri. The first involved determining the set of roads maintained by MoDOT to both establish the set of potential roads to be striped and establish the network in which striping crews may successfully travel. The second critical data set included the number and direction of passes required for striping operations as determined by the number of lanes and the presence or absence of white edge lines. Finally, the last data set involved finding the location of maintenance buildings/overnight locations positioned throughout the Central District.

In general, most of the information was recorded and available for extraction; however, the information was scattered across various forms. Each form provided portions of the necessary information applicable to the creation of our model, but no individual form provided sufficient information independent of the other available forms. As such, an important element of analysis involved creating a combined data set in which the data relevant to our analysis could be more easily referenced; this work is detailed in Chapter 3 of this report.

1.2 Road Types

Striping operations differ for different road types. The classification of roads that is relevant for pavement marking differences is based on (MoDOT 2015b):

- Road name and designation
- Number of lanes
- Division of traffic directions whether with double yellow lane or “median barrier” (undivided or divided highways accordingly)
- Annual Average Daily Traffic (AADT) factor

Based on AADT factor, number of lanes and designation, roads can be divided into the classes that will be considered in this analysis of the striping system:

- Major roads (interstate roads, most divided highways)
- Regionally Significant roads (most undivided highways, other road types with high traffic)
- Minor roads (roads with relatively low traffic, typically one- or two-lane roads)

1.3 Type of Lines

Lines can be classified based on their pattern, color, width, and functions (MoDOT 2015b):

- A solid line discourages or prohibits crossing and shall be used to delineate two adjacent lanes of traffic; a solid line can also be used to mark an edge line, or gore areas at exit ramps.

- A broken line – normal line segments separated by gaps, should consist of 10 ft. line segments and 30 ft. gaps. It indicates two adjacent lanes of traffic where crossing the line with caution is permitted.
- A dotted line provides guidance or warning of a downstream change in lane function. It's a shorter line segments separated by shorter gaps than used for a broken line.
- A double line – two parallel lines, separated by a discernible space, indicate maximum and special restrictions. It might include two solid lines that separate two opposite lanes and indicate no-passing zone from both sides. A double broken line shall delineate the edge of a lane in which the direction of travel is changed from time to time, such as reversible lanes. A double line is used for undivided roads.

Two colors are used for striping: yellow and white. White markings for longitudinal lines delineate: a separation of traffic flows in the same direction, the right-hand edge of the roadway, channelizing lines, etc. Yellow markings delineate the separation of traffic in opposite directions, the left-hand edge of the roadways of divided highways and one-way streets or ramps, the separation of two-way left turn lanes and reversible lanes from other lanes, etc. (MoDOT 2015b):

- A solid white line separates traffic flow in the same direction and shall be used as the right-hand edge of the roadway (for divided and undivided roads)
- A solid yellow line delineates the left-hand edge of the roadway of a divided highway
- A broken white line shall be used for delineation of the edge of travel path where travel is permitted in the same direction on both sides of the line (mostly for multi-lane roadway of divided and undivided roads)
- A broken yellow line shall be used to delineate the left edge of a travel path where travel on other side of the line is the opposite direction (centerline of a two-way lane, two-way roadway of undivided roads)
- A double yellow line has all the functions of double line that were described above. Double line is always made with yellow color.

Lines can differ based on their width, with MoDOT using 6 in. or 4 in. wide lines, depending upon the road type. Figure 1.1 presents the relationship between road type and line width (note that in this figure, Regionally Significant roads follow the markings specified for minor roads.

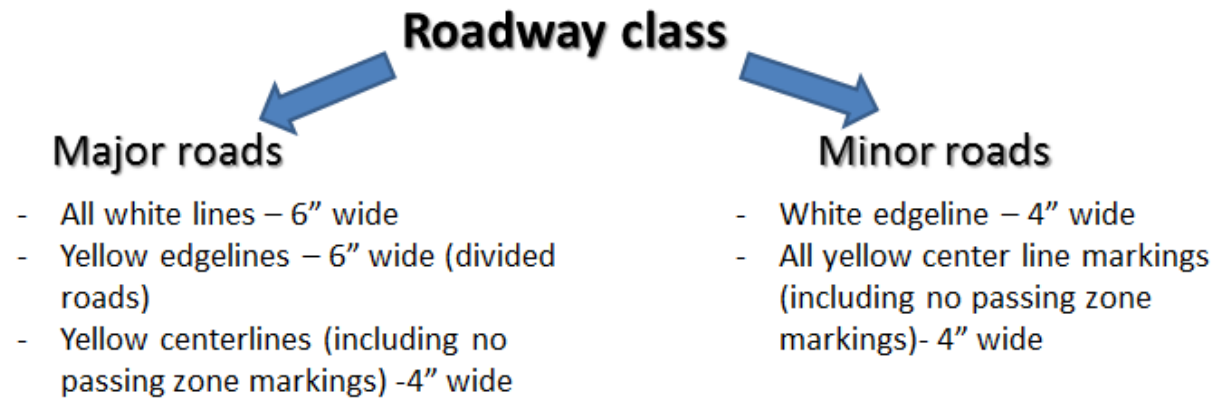


Figure 1.1 – Classification of lines based on road type

1.4 Material used for Pavement Marking

Pavement and curb markings are generally made using paints. Typically, MoDOT uses a waterborne paint for line-striping operations, although a cold weather paint can be used at lower temperatures. The night visibility of pavement markings can be enhanced by embedding the spherical glass or ceramic "beads" in the pavement marking material. There are several types of beads used by MoDOT (MoDOT 2015b):

- Type PM (performance maintenance) beads – intermediate blend of glass beads that is used for all roads except divided highways by MoDOT
- Type P (performance) beads – used by contractors
- Type L beads – "large" glass beads used by MoDOT and contractors, typically for divided highways.

1.5 Striping Operations

MoDOT has 17 stripers that are distributed across its various districts. Generally, each district has two stripers, and each striper has two crews assigned to it. The Central District has two bulk storage facilities for paint; one in Jefferson City and one in Rolla, each with a striper (notionally) assigned to it.

Figure 1.2 presents a representation of the communication system, from the statewide level down to the level of individual striping crews, used to manage striping operations (MoDOT 2015b):

- On a statewide level, regional coordinators and/or district supervisors conduct weekly, statewide conference calls to discuss operations related to striping, plans and progress
- On a regional level, regular phone conferences between the district supervisors in each region are conducted for discussion of weekly progress
- On a district level, supervisors hold weekly meetings with the crews to explain and update current progress as well as the results of the daily quality checks completed during the week.

- At a crew level, constant communication using MoDOT radios between the driver of the striper and the operators regarding the quality of stripe is essential. Communicating via radio with the drivers on the crash truck and nurse truck is also important.

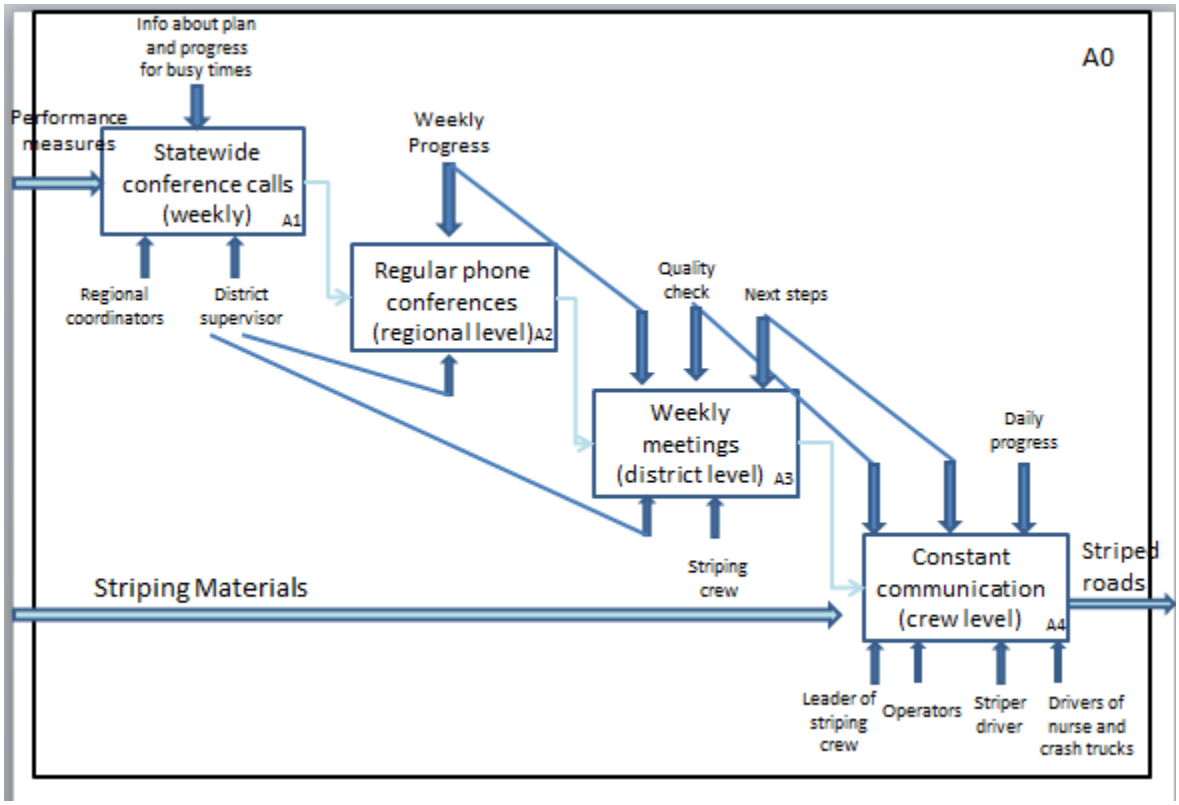


Figure 1.2 – Striping communications system

Crew leaders and operators are expected to perform two quality checks per day and add information to the Missouri Accountability and Performance System (MAPS). The crew leader should record daily progress of lines striped per day and give this information to a district supervisor, who is expected to add this data to a database in order to keep track of the progress and next steps.

The length of a typical scheduled work day is 10 hours; this includes safety meetings, drive time to overnight locations, striping, and reloading time. Weather plays a key role in enabling or limiting striping operations; if a day begins with an expectation of cold weather (less than 35˚F) or rain, striping operations will not typically be performed. This is because the paint and beads may not properly adhere to the road surface if the temperature is low or conditions are too wet.

Striping equipment involved in pavement marking operation includes:

- Striper (see Figure 1.3)
- Middle warning truck
- Rear advanced warning truck

- Shoulder advanced warning truck
- The nurse truck (transports paint from bulk storage facility to location of striping operations)

Figure attribution: http://epg.modot.org/index.php?title=Image:620.2.jpg

Figure 1.3 – Striper

Although striping speeds vary based on factors such as elevation change, for the purpose of this analysis, we will assume that a crew stripes a non-divided highway at a speed of 10 miles per hour, and a crew stripes a divided highway at a speed of 8 miles per hour. The speed when not actively striping (deadheading) is assumed to be 35 miles per hour. When striping, the striper can paint two lines concurrently (as in Figure 1.3, in which the white edge line and double yellow center line are being striped at the same time).

Due to the slow moving nature of striping equipment, striping operations for even a small area (such as a county) take multiple days. As such, it would be inconvenient (and generate excessive deadhead miles) to return the striping vehicles to the assigned bulk storage facility at the end of each day. Instead, a striping crew's vehicles (with the exception of the nurse truck) are parked overnight at the nearest MoDOT maintenance building (there are 25 such facilities in the Central District). The crew members return to the bulk storage facility at the end of the day in either a crew cab pickup or the nurse truck, these vehicles return the crew to their equipment at the start of the following day.

1.6 Striping Plan

Prior to 2015, MoDOT followed an annual striping plan as represented in Figure 1.4

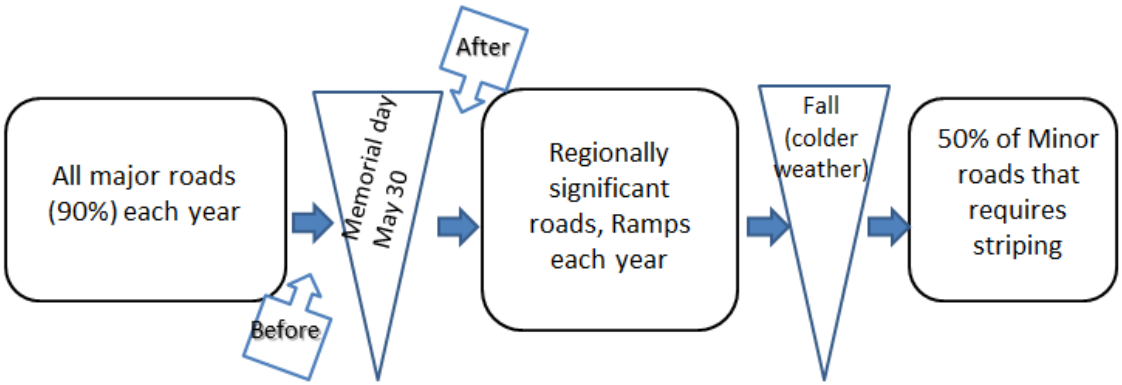


Figure 1.4 – Old striping plan

According to the old plan, striping crews were expected to stripe 90% of the major roads before Memorial Day. After Memorial Day, they striped Regionally Significant roads and ramps. Then, with the remaining striping season, crews striped minor roads with cold weather paint, with a goal of striping 50% of minor roads each year (MoDOT 2015b).

Starting in 2015, striping policy changed. A new type of beads were introduced for divided highways; these beads required a high temperature for drying. Due to the difficulty of switching the type of beads in the striper, divided highways are now striped in one continuous time period. The new striping plan is represented in Figure 1.5.

In the early part of the striping season, when the weather is still cold (before June 1, approximately), crews stripe minor roads. After this date, when the warmer weather is more conducive to striping divided highways with the new bead type (L), all divided highways will be striped. It is expected that striping on all divided highways will be completed by the end of July. From end of July until end of summer (while the weather remains warm), major undivided highways will be striped (with PM beads). As weather get colder, the rest of regional significant and minor roads will be striped until the end of the striping season (typically, mid-November) (MoDOT 2015b).

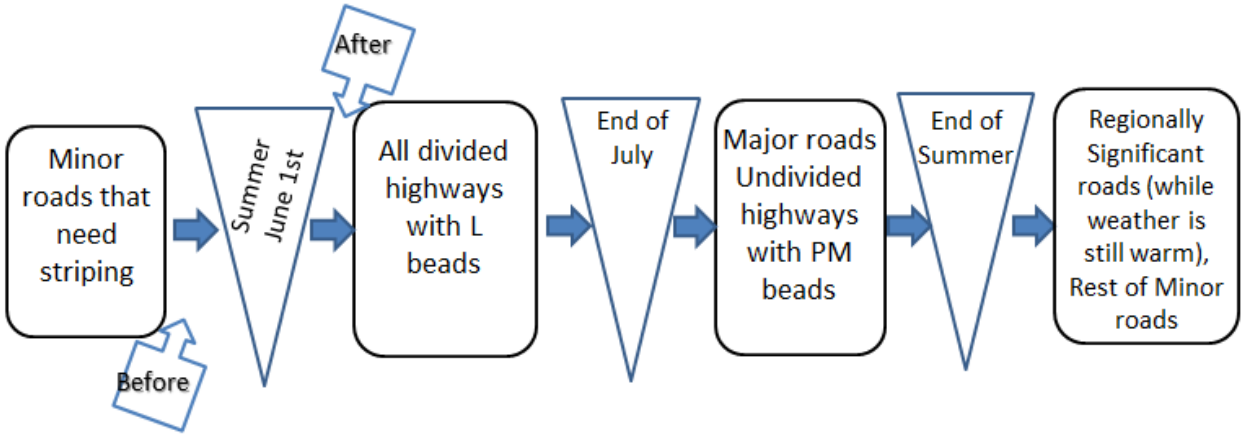


Figure 1.5 – New striping plan

2. LITERATURE REVIEW

2.1 Road Striping Operations

We are aware of no optimization models that have been used for scheduling road striping operations. Most of the research papers examining pavement marking can be divided into the following categories:

- *Safety effectiveness of pavement markings.* In (Smadi et. al.2008), the authors estimated how striping retroreflectivity levels might influence drivers, namely, whether it increases the safety of driving or makes it worse, due to drivers increasing their speeds.
- *Inventory and replacement of striping paint.* In (Kouskoulas 1988), an inventory and replacement cost optimization model is presented for pavement marking system. Another study (Onyango et. al.2014) developed a degradation model of retroreflectivity that was represented as a function of time, environmental condition, and the number of traffic repetition after its application. Based on it, a degradation model was developed.
- *Pavement marking materials.* An evaluation of various pavement marking materials used for longitudinal delineation was presented in (Taek et. al.1999), with the most cost-effective materials chosen based on the level of retroreflectivity and nighttime accidents. The influence of beads on retroreflectivity was examined in (Zhang et. al.2010).

2.2 Network Optimization Models

In the field of operations research, the Traveling Salesman Problem (TSP) is perhaps the best-known network optimization problem. The TSP's objective is to determine the shortest route passing at least once through each of the nodes in a network; this route is further required to be a circuit, that is, the starting and ending node must be identical (Pieterse and Black 2014a, Laporte 2010, and Hoffman et. al.2001). A related problem, known as the Chinese Postman Problem (CPP), involves determining the minimum length route necessary to travel each *arc* of a network at least once (Guan 1962, Pieterse and Black 2014b, and Eiselt et. al.1995b). The Rural Postman Problem (RPP) is a slight variant to the CPP, in which the objective is to identify the minimum-length route that traverses a *subset* of the arcs in a network at least once (Eiselt et. al.1995a and Monroy-Licht et. al.2013). For MoDOT's striping operations, because only a subset of all MoDOT road segments must undergo striping operations in a given year; the RPP provides a useful starting point for modeling.

Striping operations could also be considered a variant of the vehicle routing problem (VRP), first described in the literature in (Dantzig and Ramser 1959). The VRP plays an important role in supply chain management. The VRP's objective is minimization of the total distance traveled by a fleet of vehicles that leave a maintenance building, serve customers in the network, and return to the maintenance building after completion of their routes. More recent studies examining the VRP were concerned with multiple maintenance buildings, with constraint "time windows," capacity constraint, capacity and time and distance constraints, loading/unloading factor for

vehicle, etc. These papers have considered both heuristic (evolutionary algorithm) (Weise et. al.2010) and exact (Cornillier et. al.2012) that could solve different kinds of VRP.

A final similar problem statement was presented in (Orloff 1974) and named the general routing problem (GRP). The GRP determines the minimum cost tour on a network, which starts and ends at the same node, and includes certain prespecified links and nodes in the tour.

2.2.1 Rural Postman Problem

As discussed above, the RPP shares many similarities with the problem of determining a minimum-length striping schedule. Many variations to the classic RPP model have been developed for particular applications.

- *RPP with turn penalties* was introduced in (Benavent and Soler 1999). In this formulation, routes have to be operated on a street network, where some turns are forbidden and other turns are allowed but with some penalties.
- *Windy RPP*, in which the cost of traversing an arc in one direction may be different from the cost of traversing it in the opposite direction, is examined in (Corberán et. al.2005a, 2005b, 2012 and Benavent et al. 2009).
- *RPP with time constraints* (Monroy-Licht et. al.2013 and Letchford and Eglese 1997) considers situations in which the customers (arcs to be traveled) require service to occur within specific time windows.
- *Time-sensitive RPP* (Tan et. al.2012), in which the travel (or service) time of each arc depends on the time interval during which the arc is traversed.
- *Mixed RPP* (Corberán et. al.2005a, 2005b, 2012), in which the network has both arcs that must be traversed and specific nodes that must be visited.

While many applications of the RPP have been discussed in the literature (Eiselt et. al.1995a), no published study has examined the application of the RPP to scheduling striping operations. The RPP framework has been successfully applied to other roadway maintenance activities, such as winter gritting and snow removal operations (Li and Eglese 1996 and Jang et. al.2010). Striping operations and winter maintenance operations share similarities, as both consider road segments that may not require service, both attempt to minimize the total distance traveled by their respective vehicles, and both involve vehicles that begin and end service at a “maintenance building” location. Similarly, multiple maintenance buildings exist within the winter maintenance system, which roughly parallels the multiple maintenance building locations within the striping operation system.

A key difference between these operations is that for winter maintenance, all road segments requiring service are covered within a fairly short window (generally less than one day), so each winter maintenance vehicle typically returns to its starting maintenance building by the end of each day. Because striping operations run continuously over a period of multiple months, at the end of each work day, striping vehicles need to travel to some maintenance building (but not necessarily the location where operations began that day) and remain overnight.

In addition to winter maintenance, other applications of the RPP have explored fields such as delivery vehicle routing (Solomon 1987) and monitoring of roads for black-ice detection (Monroy-Licht et. al.2013). A shared aspect between these various applications of the RPP to vehicle routing problems involves the constraint of deadlines and time windows. In the case of winter maintenance operations, roads differ on classification, which result in variation in deadlines and the time frames available to service various roads (Li and Eglese 1996). For black-ice detection, routing is largely dependent on information received from weather forecasts that determine intervals in which various road segments are available for monitoring (Monroy-Licht et. al.2013). Other instances of vehicle routing such as delivery and school bus routing generally impose time constraints due to customer priority and input regarding acceptable time windows (Solomon 1987).

Another factor differentiating the RPP as applied to striping operations from other applications is the requirement to pass some road segments more than one time. For the cases of garbage collection and mail delivery, as discussed in (Eiselt et. al.1995a), an arc can generally be considered to be "satisfied" with a single pass in either direction, whereas a multi-lane road may require two or more passes in order to completely mark all of the necessary lines. Additionally, the operations required to mark road segments is limited by a constraint requiring operations to move with traffic. For example, for an undivided road segment requiring stripes on the center line and on both edge lines, the first pass could travel from some Node X to Node Y, but, as a result, the second pass would have to travel from Node Y to Node X. One case of an RPP formulation with constraints on the direction of travel over arcs was used with a container storage and retrieval facility in which the directed arcs corresponded to the storage or retrieval of containers across arcs (Vis and Roodbergen 2009).

However, no literature has been found addressing the unique combination of factors present in the system of striping operations such as directed arcs, multiple passes, and particularly slow-moving characteristics. In particular, the slow-moving characteristics that make striping a continuous operation occurring over multiple months requires striping vehicles to overnight at a maintenance building at the end of each workday. This "overnighting" constraint requires a novel modeling approach to striping operations optimization. Accordingly, we will refer to our model as the *slow-moving multi-pass postman problem with overnighting*.

2.3 Genetic Algorithm

While the RPP is easily formulated, its solution is difficult (do Rosário Moreira and Ferreira 2010). Thus, most practical solution techniques make use of heuristic procedures such as Genetic algorithms (GA). As discussed above, our application involves a model that is even more complicated than the classic RPP, thus an approach such as GA is needed to solve our problem..

In (Groves and van Vuuren 2005), somewhat similar to our study, the authors develop a local search algorithm capable of solving TSP and RPP models. In their algorithm, they represent each required arc in the original undirected network as a node of a new network. Namely, every required arc in the original network is represented by two equivalent nodes in the transformed network, one for each direction of the arc. The shortest path algorithm is then used to calculate

minimum distance for a given sequence in this transformed network. The improvement for the order of nodes in the transformed network is achieved by Two-Opt and Three-Opt algorithms. These algorithms enable the authors to reduce computational complexity of calculating the shortest path distance for future generated orders due to nice applicability of the algorithms to the undirected network. Here, the shortest path algorithm applied to the new generated network gives which nodes (again each node represents a directed arc in the original undirected graph) are traversed in the shortest path, and thereby it eliminates the burden of determining direction of each traverse.

Another closely related study to ours that implements GA to the transformed graph is (do Rosário Moreira and Ferreira 2010). Here, the transformation process is achieved with a somewhat similar method to that employed in (Groves and van Vuuren 2005). Each required arc is represented as a node without a direction and the distances between the nodes in the new generated graph are calculated by shortest path algorithm, the new problem then becomes a TSP. Since the graph is undirected, the authors in this study try to sequentially deal with optimizing both sequence and direction of all required arc traverses.

We chose to utilize a GA approach for our solution procedure. GA imitates an evaluation process so as to solve optimization problems, and it was initially proposed in (Holland 1975). This imitation process starts with generating an initial population, which is a set of initial individuals represented by chromosomes. Each generated chromosome has a value attribute known as fitness. Based on fitness score, the best ones are chosen in order to produce better chromosomes for future generations. Here, the selected chromosomes for GA are called parents, and chromosomes produced from parents are known as children. The producing process is done by some operators of GA such as crossover, mutation and immigration. At each subsequent iteration, we implement these GA operators to the newly existing population, and we expect to generate a fitter population. The iteration of GA algorithm is continued in this reproducing process until a stopping criteria (e.g. maximum computational time) is met.

Having good GA operators significantly improves the performance of the algorithm. A well-designed GA should sustain the diversity of a population for the next generation. At the same time, it should also have some local optima search operators to reach improved solution in a reasonable period of time. The exact specifications of a particular problem make the definition of operators a problem-specific issue, thus a single best set of operators cannot be identified in advance for every problem.

Initially, we do not take overnighting considerations into account when developing the GA operators. Instead, overnighting locations are inserted into the striping sequence after the operators have performed on the population. Since these two local optima operators do not significantly change the structure of the sequence, and the overnighting insertion process is done after these operations, they dramatically reduce the computational time as opposed to fully calculated version of fitness of each generated chromosome and gradually assessing the location of overnighting. For the crossover, we use Binary crossover as in (Sadegheih 2006) to sustain the diversity of a population, and thereby expecting to have a good fitness of new generation.

In addition to having good type GA operators and parameters (e.g. crossover type and rate) in the iterations of GA, the performance of GA also depends on the quality of the initial population. Here, the quality strongly relates to not only average fitness of chromosomes but also diversity of the chromosomes in the population (Ahuja et. al.2000). Not having one of these properties leads to a less efficient GA. The underlying reasons of this conjecture are that fitter parents generally produce fitter children and that diversity enables us to avoid getting stuck at a local optima. Hence, determining a strategy for generating initial population and effectively implementing this strategy are crucial. In this regard, a Randomized Greedy Heuristic Method is implemented to generate the initial population for our analysis taking into account special network properties.

A further property that differentiates our problem from previous studies is that the different road types require a different striping strategy, and this causes the use of a different methodology to transform and represent the network (see discussion of "difficult segments" in Chapter 3). As an example, some roads require directed multi-passes (e.g. 2 directed traverses in one direction, 3 in the other), other roads can be traversed in any direction.

To best of our knowledge, no approach has been provided for the implementation of the overnighting concept to RPPs.

3. DATA ANALYSIS

3.1 Preparation of ArcGIS data

ArcGIS files, such as ss_pavement_current, were provided by MoDOT. It includes the whole Missouri road network with MoDOT and non-MoDOT roads (Figure 3.1). The attribute table of this file includes all necessary information about road segments and has records about both directions of road arc (N and S, E and W).

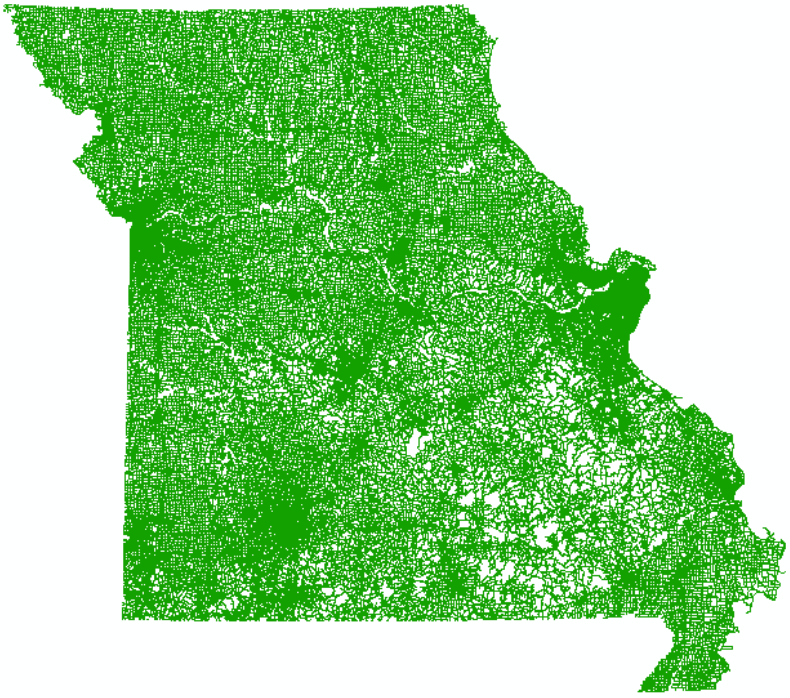


Figure 3.1 – Missouri road network

The first phase of determining a striping schedule for the Central District of Missouri involved consolidating the various forms of data in which aspects of information regarding the road segments was presented. For MoDOT, the data was divided into three different forms across three different program files that included an Access database file, an Excel spreadsheet file, and a geodatabase feature class file. While some road segment details remained present across each of the three files, many differences were found regarding the presentation of the information due to differences in the purpose of each file. For the Access and Excel files, major segments of roads between certain mile-markers were transcribed with formatting changes made to a row designating a specific road segment to reflect changes in the striping progress completed for said segment. This allowed the files to be used to monitor the progress of striping crews during periods of striping operations outside the initial scheduling and planning phase. In the case of the geodatabase feature class file, an ArcGIS program such as ArcMap could be used to visually display the location of road segments across the state of Missouri. Within each file, an extensive set of shared fields describe elements of each road segment including the county in which the road segment is located and the class or type of road such as interstate or lettered/number state

highway. However, the geodatabase feature class file also included additional fields not included in either the Access or Excel file such as the number of lanes and whether a road segment contains divided or undivided lanes. Another difference also includes the means by which road segments are defined. While the former files describe road segments as portions of roads between intersections of MoDOT maintained roads, the geodatabase feature class file describes road segments as portions of roads between intersections of other roads regardless of whether the intersecting road is maintained by MoDOT or not. As a result, a length of road between two intersections of MoDOT maintained roads may be described as a single road segment in the Access and Excel files, but, in the geodatabase feature class file, the road length may be described by multiple road segments due to the various road intersections not present in the former files. The division of road segments in the geodatabase feature class file may vary significantly in size from very small to representative of the entire road segment.

Due to the conditions in which MoDOT striping crews operate, the consolidation of the Access, Excel, and geodatabase feature class files was an important aspect of our preliminary analysis. To consolidate the data files, the first step involved creating a geodatabase feature class from the current MoDOT geodatabase feature class to reduce the contained set of cataloged road segments to only the segments maintained by MoDOT. In general, MoDOT is responsible for most national and state highways across the state of Missouri. Some of the arcs eliminated from the new data set included city and county roads, which are maintained by either local government groups or contracted to outside agencies. In addition, the striping crews examined in our analysis only operated within the Central District, which permitted the further reduction of the data set to arcs contained within the 18 counties comprising the Central District of Missouri.

For this project we consider Central District roads, and only allow for travel to occur on MoDOT roads (with the exception of some nonMoDOT roads that are included to maintain connectivity of the network). We next have to make some modifications to prepare the data for use in our optimization model.

3.1.1 Creating the Junctions

The optimization model requires information about each arc: where it starts and where it ends, and how they are connected to each other. Starting and ending nodes have to be created for each arc. Creating the junctions in ArcGIS is a solution for that. The python code was developed to add junctions in the network. Necessary steps:

1. Creating Feature layers
2. Creating junction nodes – forming the end point for each line (only unique points should be kept)
3. Recording all the end points as JNode ID (making the feature layer for junction nodes)
4. Generating TO and FROM nodes fields in the main file.
5. A starting point of each line is the end point of previous line. An end point of a line is a starting point of next line (connected one)
6. Creating features for START and END nodes
7. Joining FNode (FROM node) to Start node and calculating values

8. Joining TNode (TO node) to End node and calculating values
9. Adding DEGREE field to Junctions layers (calculating the frequency of the node – how many lines has the same start or end point)

After running the code, the files Junctions (nodes) and District4 (segments) are created. The file with all lines includes around 20,000 segments. That's a big size problem and there is a way to decrease the number of segments.

As it was mentioned previously, there are two records about the same segment: in E and W directions, in S or N directions. A divided road network was built based on projection in reality: S and N, E and W segments are a few meters apart from each other, then both records will be kept in the final data file. Concerning undivided roads, only one record is needed for the model (whether N or S, E or W). Keeping only one record will decrease data size, but aggregating of the arcs is needed as well.

3.1.2 Aggregation of the Arcs

Criteria for merging an arc:

- Degree of the node =2
- # of lanes is the same for both directions
- If county name is the same
- If designation and travelway name are the same
- Road class is the same

As it was discussed above, there are three road classes: major, regionally significant, and minor. There is a field in attribute table of the main file MAJOR_MINOR. If it's equal to Major, it means the class of the road is Major. If it's equal to Minor, then the field TW_CNTL_STATE_NAME should be considered. If this field includes CONTINUOUS OPERATION RT, then the segment is Regionally Significant. If the field is blank, then the road class of this segment is Minor.

For aggregation of the arcs another python code was created. Necessary steps:

1. Selecting arcs that intersect with junction of degree 2
2. Selecting fields for merging based on criteria discussed above.
3. Calculating the new length of merged arc, new FTnode information, keeping the minimum and maximum mile markers of segments merged (it will be kept in fields BEG_CONTINUIOS_LOG and END_CONTINUOUS_LOG)
4. Keeping only one record for one direction (whether N or S, E or W)
5. Giving each new created arc (merged) uid (unique ID). When we encounter arcs with degree 2 that weren't merged because they didn't meet one of the criteria above, then they will get the same uid. We will call these kind of segments Difficult segments.

After running the code, D4Roads (all segments) and D4Junctions (nodes) files will be created (ArcGIS shape files). These files will be used as our Data. 6078 segments and 4881 nodes were created. The network with nodes and arcs is represented on Figure 3.2. (Central District MoDOT roads). The attribute table of file D4Roads is represented in Figure 3.3 (only necessary fields for the model are kept).

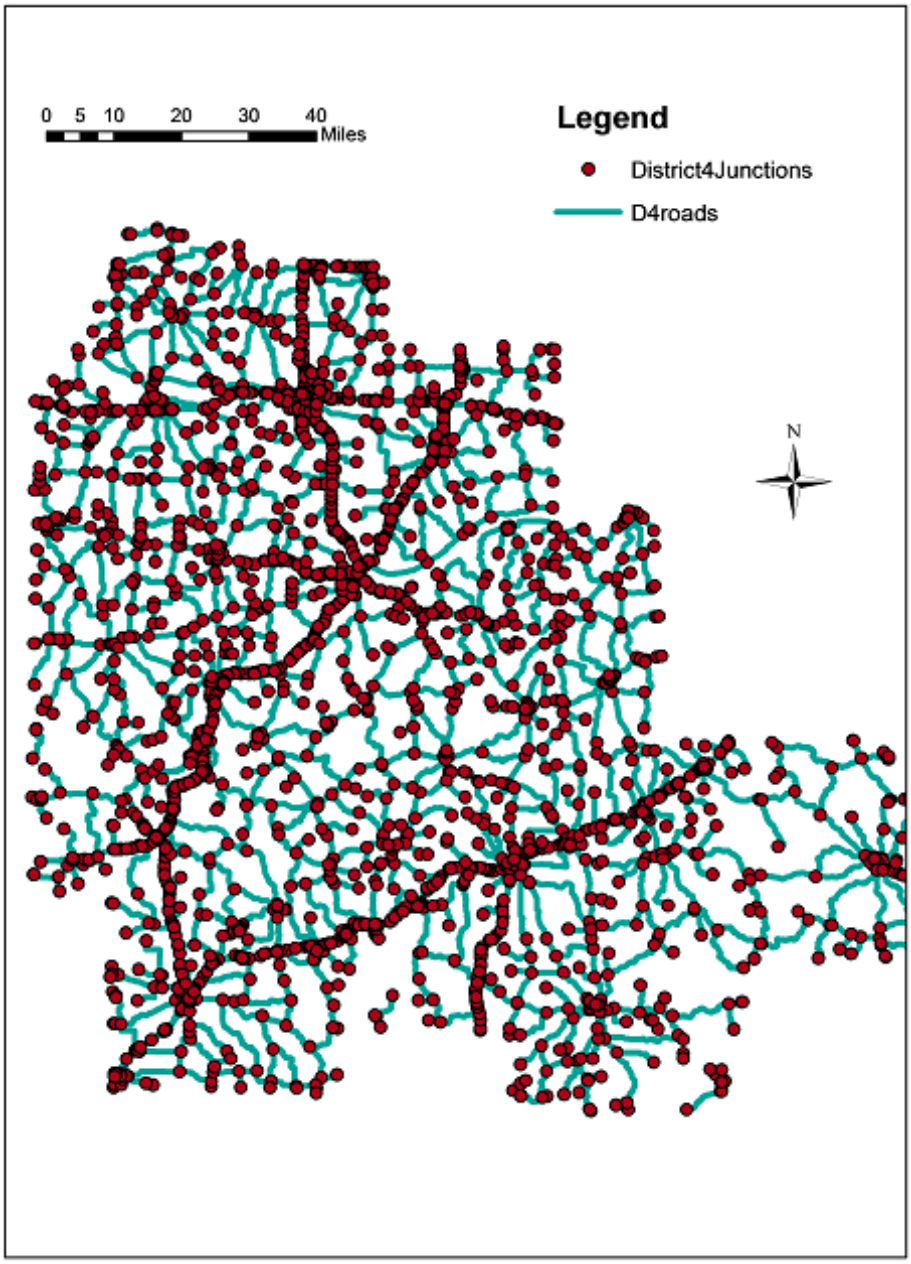


Figure 3.2 – Central District network with arcs and nodes

Table

D4roads

Designation	TRAVELWAY_NAME	TRAVELWAY_DIR	COUNTY_NAME	NUMBER_OF_LANES	DIVIDED_UNDIVIDED	TW_CNTL_STAT_NAME	MAJOR_MINOR	uid	BEG_CONTINUOUS_LOG	END_CONTINUOUS_LOG	FNode	TNode
MO	17	N	COLE	1	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	4452	186.815	186.832	2	1
US	54	W	COLE	2	DIVIDED	CONTINUOUS OPERATION RT	MAJOR	4451	122.767	122.958	1	5231
US	54	W	COLE	2	DIVIDED	CONTINUOUS OPERATION RT	MAJOR	4472	122.564	122.767	5448	1
MO	17	N	COLE	1	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	4457	186.832	186.87101	1	684
MO	28	E	MARIES	1	UNDIVIDED		MINOR	3344	0	0.32501	1003	1004
MO	28	E	MARIES	1	UNDIVIDED		MINOR	3317	30.083	30.428	1003	1010
MO	28	E	MARIES	1	UNDIVIDED		MINOR	3293	28.737	30.083	1604	1003
US	63	N	MARIES	1	UNDIVIDED	CONTINUOUS OPERATION RT	MAJOR	3346	147.598	147.977	1010	1004
US	63	N	MARIES	1	UNDIVIDED	CONTINUOUS OPERATION RT	MAJOR	3552	147.977	149.735	1004	7378
RT	AB	E	CALLAWAY	1	UNDIVIDED		LOWVOL	7921	0	2.19101	10047	10048
RT	D	N	CALLAWAY	1	UNDIVIDED		MINOR	7919	5.801	14.255	11021	10048
RT	B	S	GASCONADE	1	UNDIVIDED		MINOR	3928	3.466	3.474	10055	10056
RT	B	N	GASCONADE	1	UNDIVIDED		MINOR	3850	12.045	18.895	10060	10056
RT	B	N	GASCONADE	1	UNDIVIDED		MINOR	3928	18.895	20.957	10056	10498
RT	C	E	GASCONADE	1	UNDIVIDED		LOWVOL	3848	5.426	7.33701	10821	10056
RT	B	S	PHELPS	1	UNDIVIDED		MINOR	2693	21.445	22.345	10516	10057
RT	B	N	GASCONADE	1	UNDIVIDED		MINOR	3462	11.376	12.045	10059	10060
RT	B	N	PHELPS	1	UNDIVIDED		MINOR	3462	6.071	11.376	10518	10059
RT	U	E	GASCONADE	1	UNDIVIDED		LOWVOL	3494	2.059	5.72701	16169	10060
RT	C	E	CRAWFORD	1	UNDIVIDED		MINOR	3672	2.969	4.053	10067	10068
RT	P	N	CRAWFORD	1	UNDIVIDED		LOWVOL	3670	1.035	9.02101	15290	10068
MO	28	E	MARIES	1	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	3684	44.419	45.543	1007	1008
MO	28	E	MARIES	1	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	3592	39.437	44.419	1607	1007
RT	Z	E	MARIES	1	UNDIVIDED		LOWVOL	3668	0	7.72301	2232	1007
RT	C	E	CRAWFORD	1	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	3378	12.49	12.58001	10071	10072
RT	C	E	CRAWFORD	1	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	3399	12.362	12.49	10817	10071
RT	C	E	PULASKI	1	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	2528	0.836	0.946	10073	10074
RT	AF	N	LACLEDE	1	UNDIVIDED		LOWVOL	15	4.268	4.61301	10076	10075
RT	AD	N	LACLEDE	1	UNDIVIDED		LOWVOL	207	0	8.19901	10075	10390
RT	O	E	LACLEDE	1	UNDIVIDED		LOWVOL	33	10.074	14.136	15045	10075
RT	BB	E	PHELPS	2	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	1997	1.358	1.587	10080	10079
RT	BB	W	PHELPS	1	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	2001	11.567	12.013	10637	10079
RT	BB	E	PHELPS	2	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	2001	1.587	1.82	10079	10639
RT	C	E	MARIES	1	UNDIVIDED		LOWVOL	3686	0	2.244	1008	10832
MO	28	E	MARIES	1	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	3908	45.543	50.389	1008	1610
RT	BB	E	PHELPS	1	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	1983	2.151	2.202	10082	10081
RT	BB	E	PHELPS	1	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	2510	2.202	12.715	10081	10642
RT	BB	E	PHELPS	1	UNDIVIDED	CONTINUOUS OPERATION RT	MINOR	1995	2.033	2.151	10637	10082

1 (0 out of 6078 Selected)

D4roads

Figure 3.3 – Attribute table of D4Roads file

All divided roads are ready to be placed in an Excel file (the final input file for the model). As two records were kept for divided roads, difficult segments were not created. Undivided roads require additional analysis, as discussed below, for "difficult" segments.

3.2 Difficult Segments

As it was discussed above, "difficult" segments can be identified based on uid. The following steps should be performed:

1. Choose one county (e.g., Boone); choose only undivided roads of this county. (Choosing in ArcGIS can be performed with the tool Select by Attribute, and then exporting the data as a new layer).
2. Copy the data from attribute table (i.e., BooneUndivided) to excel
3. Sort the data based on uid from smallest to largest value
4. All uids with same value should be marked with a blue color (As a note, these segments couldn't be merged, and records about both directions were kept for the same segment)
5. In step 4 above, groups based on the same uid were created. For each group find what was the reason of not merging (whether number of lanes for each direction wasn't the same, road class etc.)
6. Retain only one direction for segments (assumption: preferably save N and E if there are no other hard cases)
7. Make sure the connectivity of the network is maintained (when deleting the arc, from and to nodes are deleted as well. Don't delete the nodes that connect "difficult" segments with others).
8. Calculate number of passes that are required for finishing the striping operation.
9. Choose Next County and repeat the algorithm from step 1. Stop when all counties are completed.

Examples of "difficult" segments that might occur:

- *When end (or start) point of a segment in direction W is different from end (or start) point of segment in direction E (for undivided highways).* The same case happens for S and N direction segments as well. The example is represented in Figure 3.4. There are segments from node 1 to node 4 of road RT M that has the same uid. In ArcGIS there is a record about segment 1-3 in W direction, # of lanes =1, segment 3-4 in W direction, # of lanes =1. Also, there is record about segment 1-2 in E direction, # of lanes=1, segment 2-4 in E direction, # of lanes=1. Number of lanes is the same for both direction, but because they act differently in both directions the code couldn't merge these arcs and kept the record about both cases in the attribute table. (end and start points are different).
    - *Solution*: two segments with only one direction should be chosen (E in this case). The records about two other segments (W in this case) will be deleted.
    - *Assumption*: for most cases N and E directions will be chosen. There will be exclusions when the connectivity can't be kept or other harder cases happen (like different number of lanes).

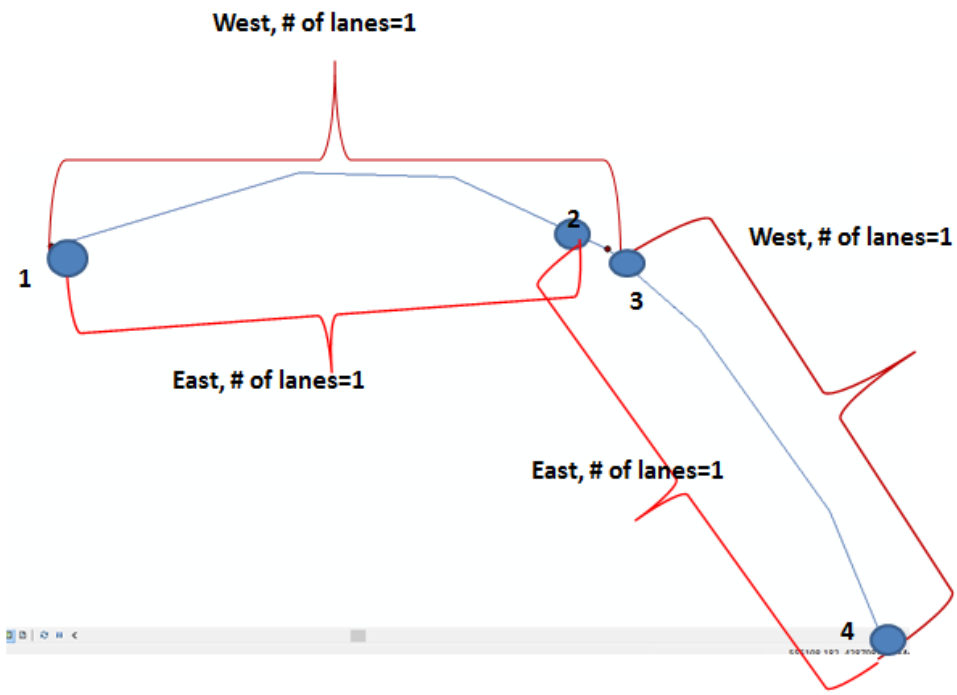


Figure 3.4 – Example of "difficult segment" (end and start points are different)

- *When segments have different number of lanes in both directions*. Example is represented in Figure 3.5. The code couldn't merge the segments from node 1 to node 3 and kept records about both direction (N and S). As we can see segment in N direction has only 1 lane but it's changing in S direction from 2 lanes to 4 lanes. It happens in cases when other roads intersect with this road. Road segment MO 163 from 1-to 3 has a record in N direction and # of lanes =1. Road segment MO 163 from node (1-to 2) = S direction, # of lanes =2. Road segment MO 163 (2-3) = S direction, # of lanes =4.
  - *Solution:* In order to decide how to treat this kind of segments we have to know how many passes should be made for each segment in order to finish the striping operation (calculating number of passes will be discussed in Section 3.3). Segment 1-2 will require 2 passes (if striping starts from one-lane side N (edge line and center line)). Segment 2-3 will require 3 passes (if striping starts from one-lane side N). It's better to keep records about the two segments in S direction because these two segments will be striped differently, and delete the record about the segment in N direction.
  - *Assumption:* Decision where to start striping (which direction) should be made in order to have the minimum number of passes

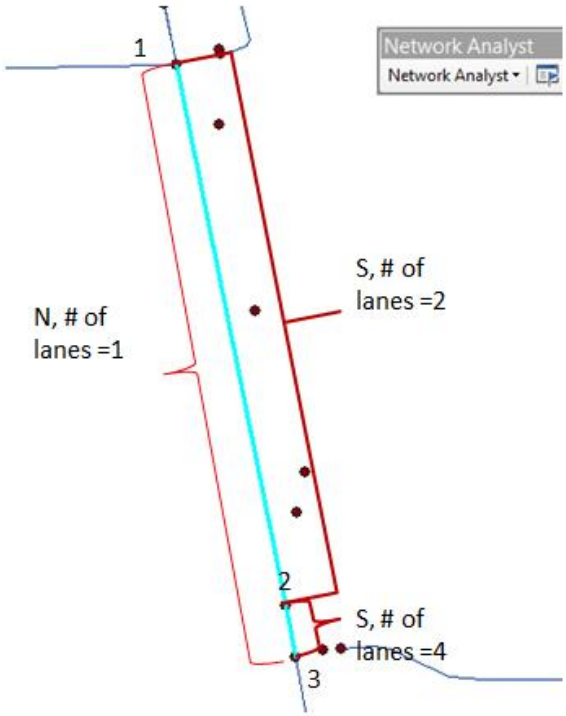


Figure 3.5 – Example of "difficult" segment (# of lanes is not the same)

- *When segments have different number of lanes and passes, and different end or start points in both directions*. The example is represented in Figure 3.6. This is a combination of the first and second type of "difficult" segment when the behavior of segments is totally different in both directions.
  - The segment from node 1 to node 6 couldn't be merged because in N and S they have a different number of passes, and also, start and end points are not the same. Analysis of the number of passes required has to be made for each segment in each direction.
  - Let's choose to save N segments. If striping of double yellow line and white edge line (or white center line) is started from N direction than the number of passes for segment 1-2 will be equal to 3 (S side is one-lane road). With the same, the logic number of passes for segment 2-5 is equal to 2 (for one-lane, two-lane and then one lane segment again). The number of passes for segment 5-6 is equal to 3 (S side is one-lane road)
  - Let's choose to save S segments. The assumption about starting striping from N is kept. It's hard to calculate number of passes for segment 1-3 because for part of segment 1-2 it's required to have 3 passes but for part of segment 2-3 only 2 passes. Number of passes for segment 3-4 = 2. It's hard to calculate number of passes for segment 4-6 as well because in area of segment 4-5 it's required to have 2 passes but for segment 5-6 is 3. It seems it's not preferable to save records for S segments because for some parts of the segments the number of passes won't be the same. It's better to choose the segments in direction where the full segment can be striped with the same number of passes.
  - For this example, it's better to save records about segments in N direction (1-2, 2-5, 5-6) and write down the number of passes that are necessary for finishing the striping.

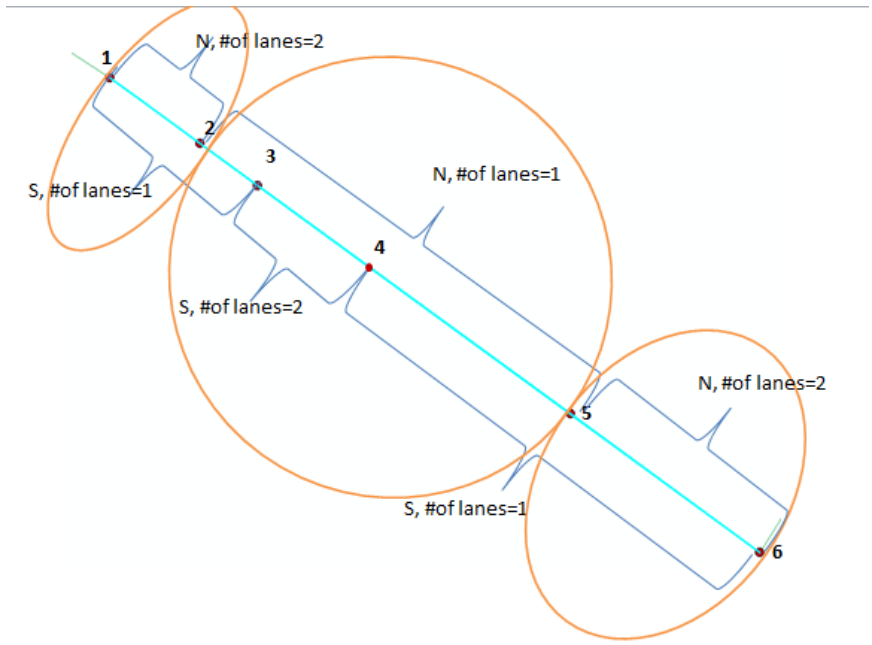


Figure 3.6 – Example of "difficult" segment (# of passes and end/start points are different)

How this assumption (in what direction to start striping the first 2 lanes (usually double yellow line and edge line (white center line))) works in reality is represented in Figure 3.7. We assumed that striping would be started from N, and the record about segments in N direction only will be saved. Now it is easy to calculate the required number of passes.

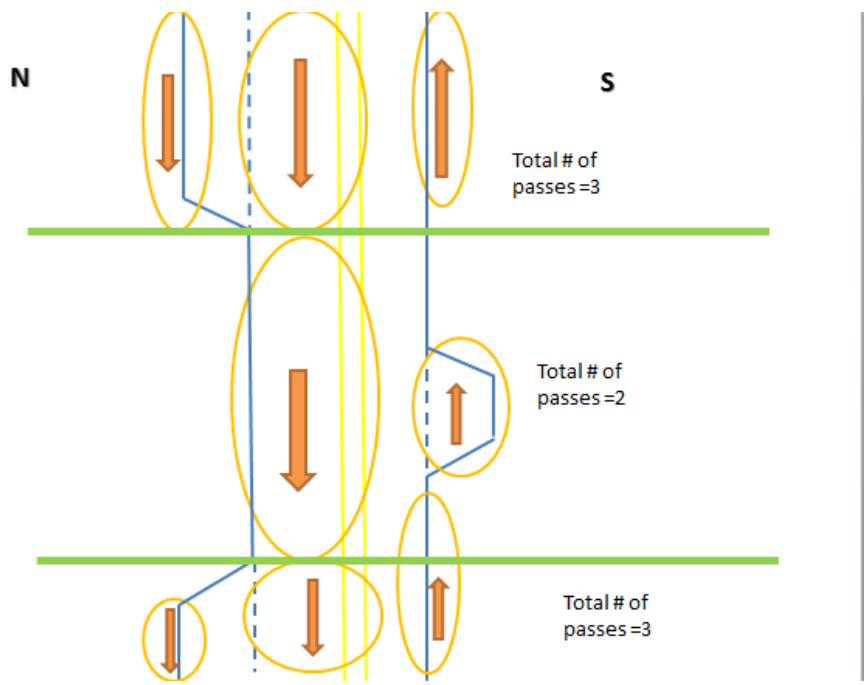


Figure 3.7 – Example of how road looks in reality with different # of lanes on each side

There are longer and harder segments in the network that the code could not merge for similar cases. These were treated the same way as the three previous types. For most of the cases an assumption about the starting direction was made. In the Excel file, the column with information about the opposite direction is added (how many lanes the opposite direction has).

- *When segments are disconnected from the network*. It means there are no MoDOT roads in Central district that can connect them with others. These kind of roads are marked with "pink" color in the Excel file. The example is shown in Figure 3.8.
  - *Solution:* Non-MoDOT roads or roads from other districts can be added to the network. That disconnected segment can be reached by passing these roads. Added roads were marked with a green color, and they don't require striping.

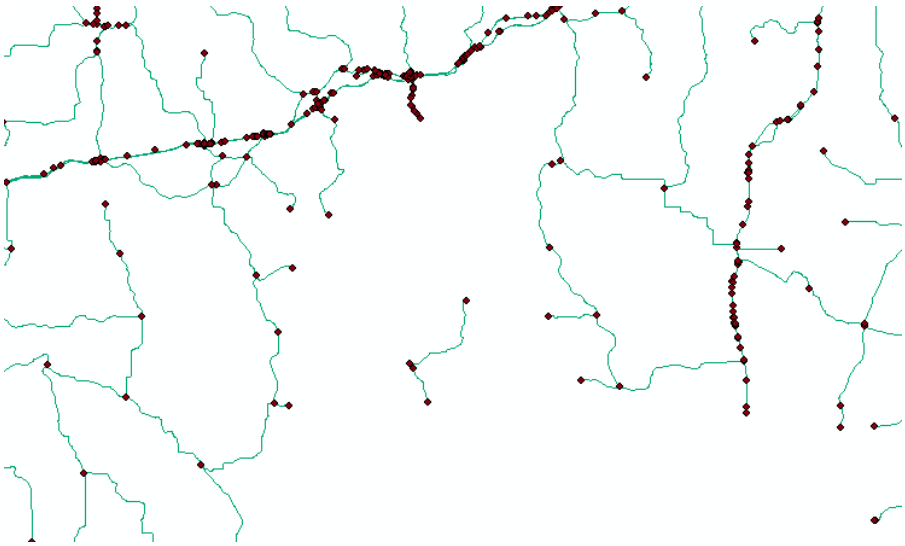


Figure 3.8 – Example of disconnected segment

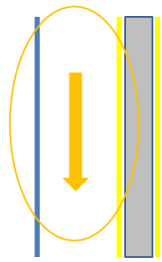
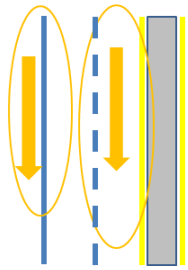
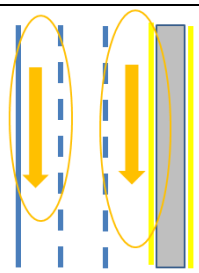
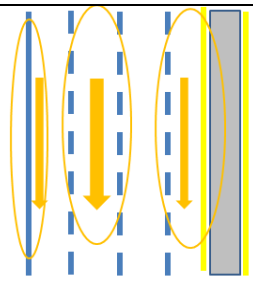
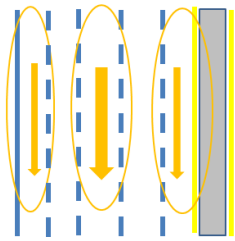
3.3 Calculation of Number of Passes

It's very important to know how many times each segment should be passed for finishing the striping process for optimization model. The strong logic has to be developed.

3.3.1 Calculation Logic for Number of Passes for Divided Roads

We decided to treat each side of divided roads as a separate road, so it's only important how many lanes it includes. It doesn't matter whether it's in N or S, E or W, number of passes will be the same for each side. The logic includes the fact that the machine can stripe two lanes at the same time. Calculation logic for number of passes for divided roads is represented in Table 3.1 where there is a road scheme of striping for each case.

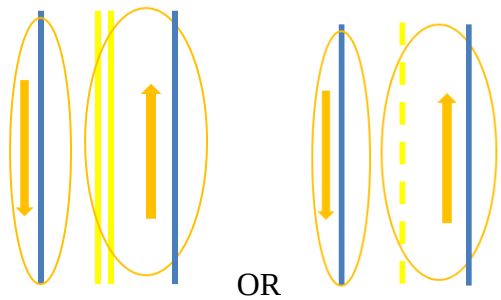
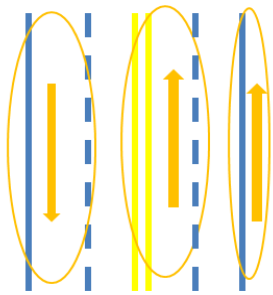
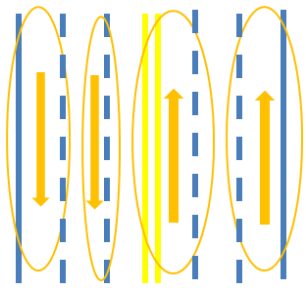
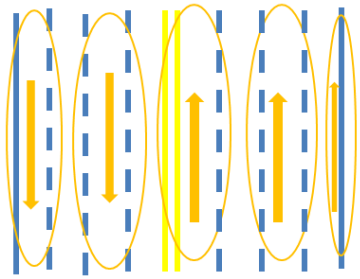
Table 3.1 – Calculation of number of passes for divided roads

# of lanes for one side (N, S, E, W)	# of passes	Scheme for how passes will be made
1 lane	1 pass	
2 lanes	2 passes	
3 lanes	2 passes	
4 lanes	3 passes	
5 lanes	3 passes	

3.3.2 Calculation Logic for Number of Passes for Undivided Roads

We only consider segments for this Section that have the same number of lanes for each side. The calculation logic for number of passes for undivided roads is described in Table 3.2.

Table 3.2 – Calculation number of passes for undivided roads

# of lanes for N or E side	# of lanes for S or W side (opposite direction)	# of passes	Scheme for how passes will be made
1	1	1	If only center line should be striped (no edge lines)
1	1	2	
2	2	3	
3	3	4	
4	4	5	

In this case an assumption about the starting direction (where striping starts) is not required. No matter where the start is, the number of passes will be the same.

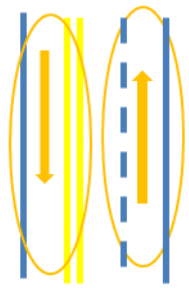
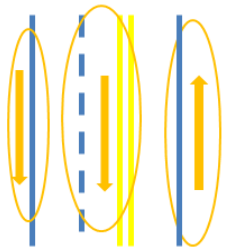
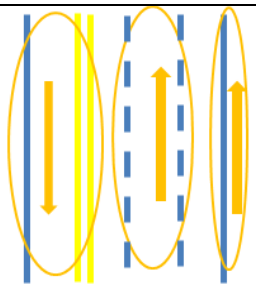
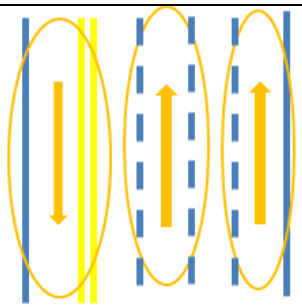
The striper can paint two lines at the same time. Based on discussions with MoDOT, the decision was made that it is more preferable to stripe two lines of different color in terms of usage of paint. If there is a choice it is better to stripe yellow center line and white edge line (or white center line) together.

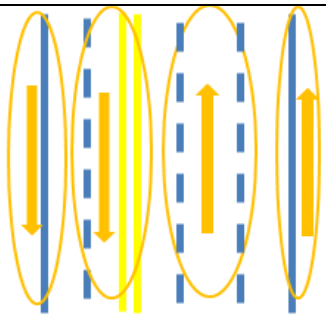
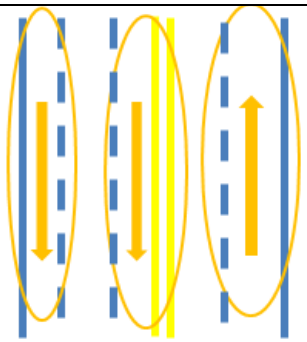
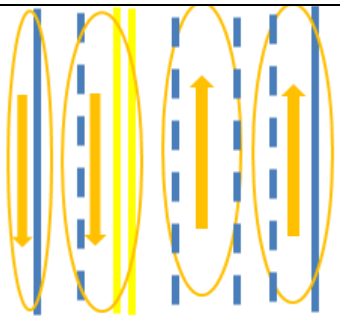
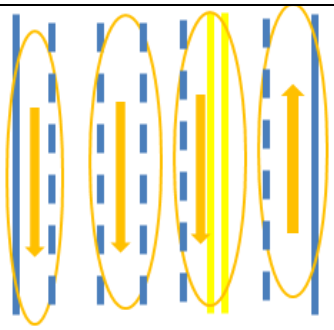
In Table 3.2, the case of two-lane road is represented when number of passes required for striping is equal to one. Usually, this kind of road only has the center line. There are no edges that require striping. In order to identify these segments, the Excel file Striping progress should be checked. If in column White E/L, there are no numbers (length information), it means that no white paint should be used for the edges (Figure 3.9). Only the yellow center line will be painted. Then, only one pass is required to finish this center line. These segments are marked with orange in the Excel file (final data).

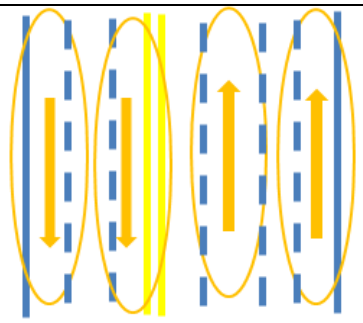
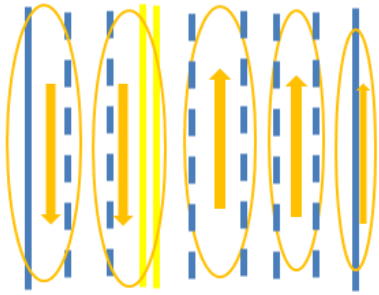
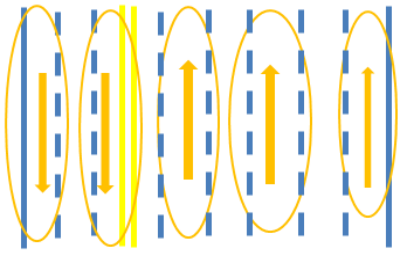
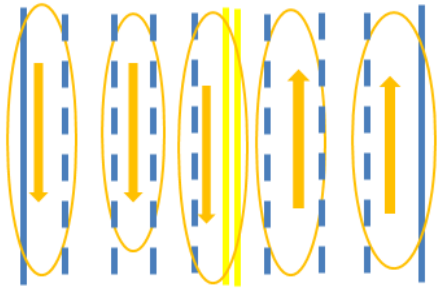
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1		Contract Striping				Striping Program 2014											8%				
2		Maybe Striped by other Di				Not wide enough to sR			outes that may need restriped				Grand Total Completed to Date								
3		Omit Routes				Contract Seal Coat - Modot S 2014							CONTRACT OR MODOT WORK IN 2013								
4		Completed Routes				MoDot ,Seal Coat , Patch - MoDOT Stripe															
5		DON'T TYPE R HERE				CLC / Contract Resurfacing 2014															
6		COUNTY	DESG.	TWAY		Total Length	Yellow E/L	White E/L	C/L	Skips	Yellow Ramps	White Ramps	Lane Lines	One lane Bridge	Gores	Total Line Miles	Mill Thick.	Total Progress	Striping Density	Done ?	Line Miles Completed
28		WOW E/L	RT	CC		10.81		5.2	11.08							16.28		16.28		Done	0
29			RT	D		1.59			2.12							2.12		0		No	0
30		WOW E/L	RT	DD		2.86		5.72	4.39							10.11		10.11		Done	0
31		Significant	RT	E		12.96		27	25.98	0.1						53.08		0		No	0
32			RT	EE		1.98		3.96	3.01							6.97		6.97		Done	0
33		wow/narrow	RT	F		12.91			23.14					0.6		23.74		23.74		Done	0
34			RT	FF		1.58		4	1.98							5.98		0		No	0
35			RT	H		4.09		8.2	5.06							13.26		13.8		Done	0
36			RT	HH		5.75		11.5	9.34							20.84		20.84		Done	0
37			RT	J		12.94		25.8	22.17							47.97		47.97		Done	0
38		Significant	RT	K		7.35		14.4	12.73							27.13		0		No	0
39		Significant	RT	KK		2.52		5.04	4.96							10		0		No	0
40		WOW E/L	RT	M		8.08		16.16	14.52							30.68		30.68		Done	0
41		wow/narrow	RT	MM		4.48			8.53							8.53		8.53		Done	0
42			RT	N		5.79		11.6	11.04							22.64		22.64		Done	0
43			RT	NN		8.44			15.53							15.53		0		No	0
44		wow/narrow	RT	O		6.48			12.37							12.37		12.37		Done	0
45			RT	OO		4.62		8.8	8.36							17.16		0		No	0
46		Significant	RT	PP		6.07		13.6	10.63	0.2						24.43		0		No	0
47			RT	T		1.26			2.52							2.52		0		No	0
48		Significant	RT	TT		1.33		0.78	2.36	0.21			0.45			3.8		0		No	0
49			RT	U		3.44		2.1	5.43							7.53		7.53		Done	0
50		WOW E/L	RT	UU		4.14		5.6	6.97							12.57		12.57		Done	0
51			RT	V		9.4		18.8	14.7							33.5		33.5		Done	0
52			RT	VV		7.18		14.2	12.63							26.83		26.83		Done	0
53		Significant	RT	WW		7.82		17.4	13.08	0.1			0.1			30.68		0		No	0
54		WOW E/L	RT	Y		4.92		9.84	8.99							18.83		18.83		Done	0
55			RT	YY		2.31			4.62							4.62		0		No	0

Figure 3.9 – Identification of segments that have centerline only

Table 3.3 – Calculation of number of passes for "difficult" segments

# of lanes for N or E side*	# of lanes for S or W side (opposite direction)	# of passes	Scheme for how passes will be made	Notes
0	2	2		The same logic as for divided roads because road can be passed only in direction
0	3	2		//-//-//
1	2	2		Assumption: Striping will be started from one-lane side
2	1	3		Assumption: Striping will be started from two-lane side
1	3	3		No matter where striping will be started, number of passes will be the same
1	4	3		Assumption: Striping will be always started from one-lane side in order to have the minimum number of passes

# of lanes for N or E side*	# of lanes for S or W side (opposite direction)	# of passes	Scheme for how passes will be made	Notes
2	3	4		Assumption: Striping will be started from two-lane side
3	2	3		Assumption: Striping will be started from three-lane side
2	4	4		No matter where striping will be started, number of passes will be the same
2	5	4		Assumption: Striping will be always started from two-lane side in order to have the minimum number of passes

# of lanes for N or E side*	# of lanes for S or W side (opposite direction)	# of passes	Scheme for how passes will be made	Notes
3	4	4		Assumption: Striping will be always started from three-lane side in order to have the minimum number of passes
3	5	5		No matter where striping will be started, number of passes will be the same
3	6	5		Assumption: Striping will be always started from three-lane side in order to have the minimum number of passes
5	4	5		Assumption: Striping will be always started from five-lane side in order to have the minimum number of passes

* Assumption: Start striping double yellow line and white edge line (centerline) from this direction

3.4 Final Excel Data File: MoDOT Roads

After all "difficult" segments are identified, only one direction is chosen, extra information deleted, number of passes is calculated, and the connectivity is checked, the final datasheet in Excel is ready for usage in the optimization model. A sample is represented in Figure 3.10.

1	Name	Direction	COUNTY_NAME	NUMBER_OF_LANES	#of lanes in Opos.Dir	uid	BEG_CONTINUOUS_LO(	END_CONTINUOUS_L(	FNode	TNode	Combine	SegmentID	Distance(m
2746	8	E	WASHINGTON	1	1	1668	0	0.13001	8615	8151	8151_8615	E_WASHINGTON_MO_8_0_0.13001	209.2372
2747	U	N	WASHINGTON	1	1	1689	14.763	14.93001	16220	8154	8154_16220	N_WASHINGTON_RT_U_14.763_14.93001	269.076
2748	8	E	WASHINGTON	1	1	1691	55.277	56.518	8190	8151	8151_8190	E_WASHINGTON_MO_8_55.277_56.518	1997.090
2749	8	E	WASHINGTON	2	1	1691	55.274	55.277	8154	8190	8154_8190	E_WASHINGTON_MO_8_55.274_55.277	4.81158
2750	8	E	WASHINGTON	1	1	1715	54.642	55.196	8146	8153	8146_8153	E_WASHINGTON_MO_8_54.642_55.196	892.0844
2751	8	E	WASHINGTON	2	1	1715	55.196	55.274	8153	8154	8153_8154	E_WASHINGTON_MO_8_55.196_55.274	125.6049
2752	21	N	WASHINGTON	1	1	1717	126.069	135.914	842	832	832_842	N_WASHINGTON_MO_21_126.069_135.914	15838.27
2753	8	E	WASHINGTON	1	1	1719	54.608	54.642	832	8146	832_8146	E_WASHINGTON_MO_8_54.608_54.642	54.47714
2754	P	N	WASHINGTON	1	1	1738	13.756	14.03901	15304	8143	8143_15304	N_WASHINGTON_RT_P_13.756_14.03901	456.2925
2755	8	E	WASHINGTON	3	1	1742	54.409	54.503	8143	8144	8143_8144	E_WASHINGTON_MO_8_54.409_54.503	150.9402
2756	8	E	WASHINGTON	2	1	1742	54.503	54.608	8144	832	832_8144	E_WASHINGTON_MO_8_54.503_54.608	169.1440
2757	8	E	WASHINGTON	3	1	1939	54.36	54.409	9248	8143	8143_9248	E_WASHINGTON_MO_8_54.36_54.409	78.20956
2758	8	E	WASHINGTON	1	1	1939	53.608	54.36	8158	9248	8158_9248	E_WASHINGTON_MO_8_53.608_54.36	1207.629
2759	8	E	WASHINGTON	1	1	1943	53.547	53.608	8157	8158	8157_8158	E_WASHINGTON_MO_8_53.547_53.608	97.74926
2760	8	E	WASHINGTON	1	1	2033	53.13	53.547	1197	8157	1197_8157	E_WASHINGTON_MO_8_53.13_53.547	670.4666
2761	O	E	WASHINGTON	1	1	2037	0	2.216	8154	15058	8154_15058	E_WASHINGTON_RT_O_0_2.216	3568.438
2762	8	E	WASHINGTON	1	1	2064	53.007	53.13	9251	1197	1197_9251	E_WASHINGTON_MO_8_53.007_53.13	197.3077
2763	E	N	WASHINGTON	1	1	2109	0	0.784	8158	11976	8158_11976	N_WASHINGTON_RT_E_0_0.784	1261.617
2764	8	E	WASHINGTON	1	1	2117	43.875	53.007	9240	9251	9240_9251	E_WASHINGTON_MO_8_43.875_53.007	14695.77
2765	O	E	WASHINGTON	1	1	2123	2.217	2.58001	15058	15060	15058_15060	E_WASHINGTON_RT_O_2.217_2.58001	583.5407
2766	21	N	WASHINGTON	1	2,1	2181	135.914	137.25	832	845	832_845	N_WASHINGTON_MO_21_135.914_137.25	2148.737
2767	21	N	WASHINGTON	3	1	2181	137.25	137.287	845	846	845_846	N_WASHINGTON_MO_21_137.25_137.287	59.55811
2768	E	N	WASHINGTON	1	1	2183	0.784	1.517	11976	846	846_11976	N_WASHINGTON_RT_E_0.784_1.517	1179.458
2769	F	N	WASHINGTON	1	1	2220	0.728	0.948	12283	12284	12283_12284	N_WASHINGTON_RT_F_0.728_0.948	356.2257
2770	F	N	WASHINGTON	1	1		0.728	0	12283	8157		N_WASHINGTON_RT_F_0.728_0	
2771	185	N	WASHINGTON	1	1	2272	0	1.443	1197	1198	1197_1198	N_WASHINGTON_MO_185_0_1.443	2323.979
2772	AA	E	WASHINGTON	1	1	2465	0	11.48601	9240	1198	1198_9240	E_WASHINGTON_RT_AA_0_11.48601	18483.80
2773	E	N	WASHINGTON	1	1,2	2652	1.517	6.613	846	3479	846_3479	N_WASHINGTON_RT_E_1.517_6.613	8204.234
2774	47	N	WASHINGTON	1	1	2654	7.037	12.297	3485	3479	3479_3485	N_WASHINGTON_MO_47_7.037_12.297	8465.34
2775	21	S	WASHINGTON	1	1	2724	52.267	56.865	1424	1444	1424_1444	S_WASHINGTON_MO_21_52.267_56.865	7400.266
2776	21	S	WASHINGTON	3	1	2724	56.865	57.066	1444	846	846_1444	S_WASHINGTON_MO_21_56.865_57.066	323.225
2777	47	N	WASHINGTON	1	1	2726	12.297	15.55	3479	1424	1424_3479	N_WASHINGTON_MO_47_12.297_15.55	5234.372
2778	F	N	WASHINGTON	1	1	2872	1.23	9.02901	12281	12282	12281_12282	N_WASHINGTON_RT_F_1.23_9.02901	12549.45
2779	F	N	WASHINGTON	1	1	2872	0.949	1.23	12284	12281	12281_12284	N_WASHINGTON_RT_F_0.949_1.23	450.835
2780	185	N	WASHINGTON	1	1	2876	1.443	10.747	1198	1200	1198_1200	N_WASHINGTON_MO_185_1.443_10.747	14972.5
2781	E	N	WASHINGTON	1	1	2926	6.613	12.76001	3479	11301	3479_11301	N_WASHINGTON_RT_E_6.613_12.76001	9889.605

Sheet1 Sheet2 Sheet3

Figure 3.10 – Part of final Excel data file

3.3.3 Calculation Logic for Number of Passes for "Difficult Segments" (Undivided Roads)

All the different cases of "difficult" segments that were met in the network are represented in Table 3.3. For "difficult" segments it's important to know how many lanes the opposite side has. This is going to influence the final number of passes required for striping. A key assumption needs to be made regarding from which side (E or W, N or S) striping operations should be started. In Table 3.3 the starting side (direction) will be in column 1. The second column includes information about opposite side. In most cases, the starting direction will be the one that gives the minimum number of passes. Column Notes details what assumption was used here. Also, there is a scheme how striping passes should be made in reality.

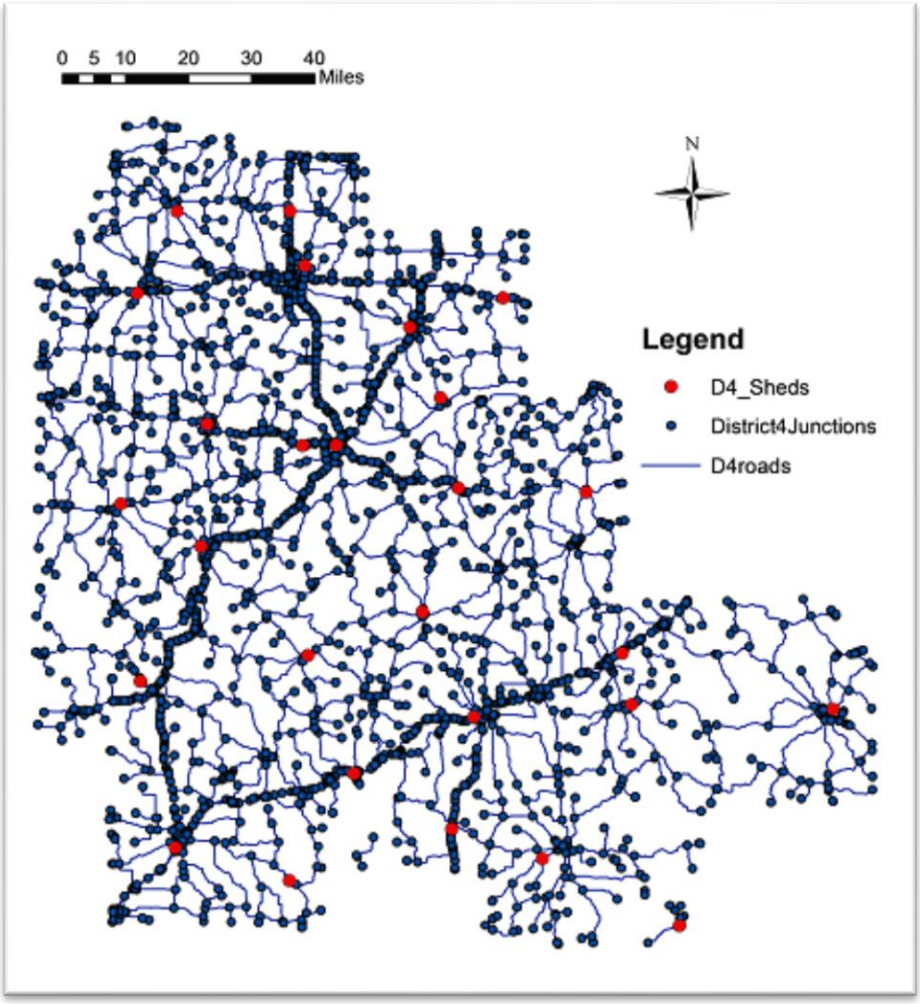
As a summary:

- Blue /dark blue color identifies "difficult" segments
- Orange color identifies two-lane undivided roads with center line only (1 pass)
- Pink color means that segment was disconnected from the road network
- Green color shows added roads (nonMoDOT roads or roads from other districts)

Yellow color just marks all uids. Important fields that will be used in the model: Designation, Name, Direction, County name, Number of lanes, Number of lanes of opposite side, uid, Beg_Continuous_Log, End_Continuous_Log (mile marker), FNode, TNode, Distance in meters and miles, Class of the road, Divided_Undivided class, number of passes, NeedStripe(0,1). NeedStripe(0,1) field defines whether the segment should be striped this year (1) or not (0). This decision has to be made by MoDOT district supervisor. SegmentID field will be the main output from the model: Direction_County_Designation_Name_MileMarker(Beg_End).

3.5 Overnight Location Distances File

As it was discussed above, every day the striping crew has to park their equipment in the closest facility (overnight location). The location of all facilities in central district (27, including one from another district) is represented on the map (Figure 3.11). The map includes D4roads, District4_Junctions (they were created in paragraphs 3.1 and 3.2), and D4_Sheds (facilities) shape files.



Note: subsequent to the creation of this map, we were informed that the maintenance facility at Hallsville was recently closed; this Hallsville facility has been removed from our models.

Figure 3.11 – Map with overnight locations

The striping crew finishes their job at some node (end of a segment) at the end of each day. For this node they should choose the closest overnight location. It's necessary to know the closest facility for each node and distance to it for the model. There is a Network Analysis tool in AcrGIS that can help to calculate all the distances from each node to its closest facility. The following steps shall be made:

1. Make sure that catalog is visible in ArcGIS. There is a folder that includes all our shape files described above (District4Roads.gdb). Find it in catalog-> right click on this folder ->choose New->Feature Dataset. Create a new feature dataset and give it the name (i.e., try).
2. After the first step, the try dataset should appear in catalog in District4Road.gdb folder. Right click on try -> choose import -> Feature Class(single).

3. In processing window choose the input – the network that will be used for finding the best route. If we choose only MoDOT roads distances won't be calculated for all nodes because some segments are disconnected. The network with all MoDOT roads was used (ss_pavement_current). The file all roads will be created (Figure 3.12).

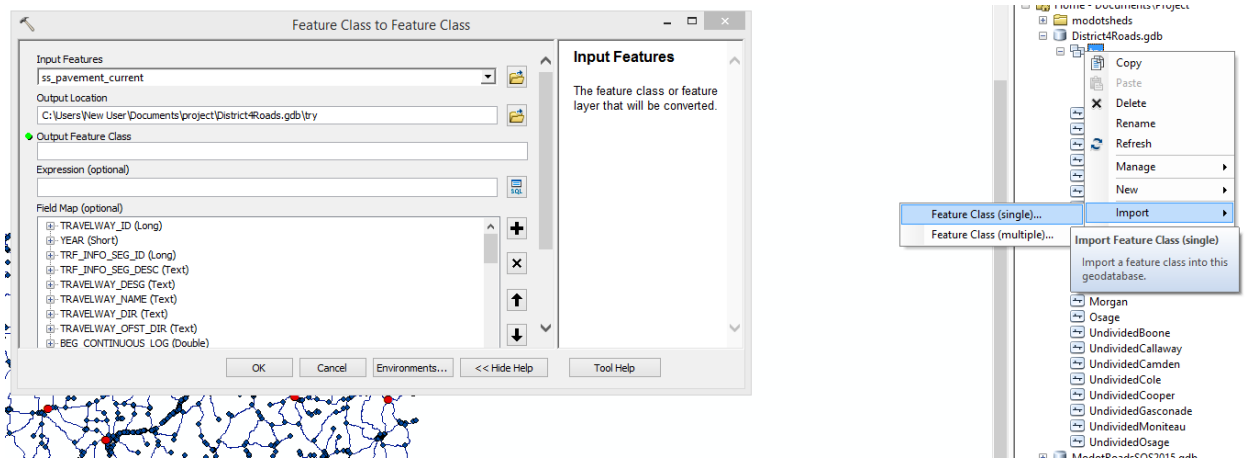


Figure 3.12 – Creating the feature class in ArcGIS

4. Then after necessary data is added, the network dataset should be created. Right click on created folder try -> New -> choose Network Dataset. The files try_ND and try_ND_Junctions will be created. The file with junctions will be the same as District4_Junctions.
5. Make sure that Network Analysis tool is enabled. (Customize->Toolbars). Then go to network analysis panel -> choose the network dataset try_ND. In network analysis catalog choose New Closest facility.
6. The list of different fields will be given. Facilities correspond to the overnight locations. Right click on facilities -> Load Locations -> choose file D4_Sheds. They will be searched within a 5000 meters radius. (Figure 3.13)

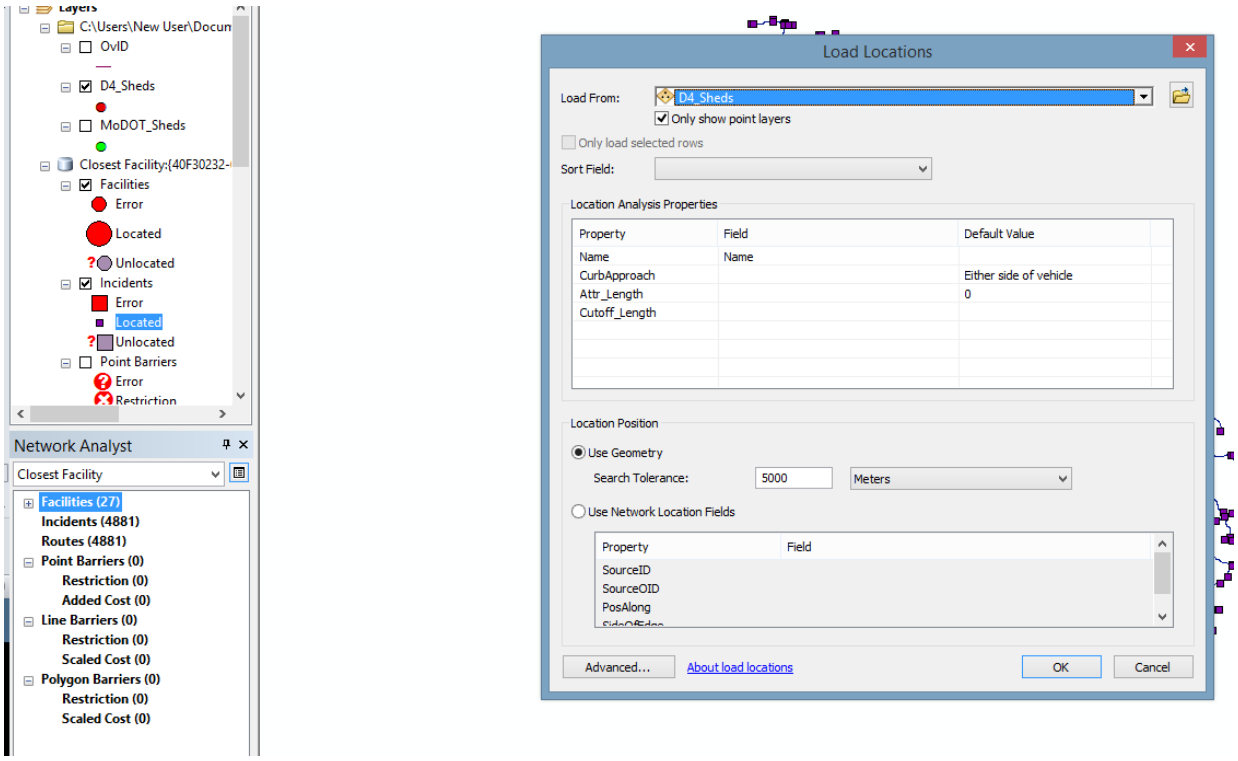


Figure 3.13 – Loading facilities step

7. Incidents correspond to the nodes. Right click -> Load locations -> choose file D4_Junctions.
8. If all facilities could be allocated they would be represented with a big red circle. If all incidents were allocated they would be represented with purple squares. (Figure 3.14)

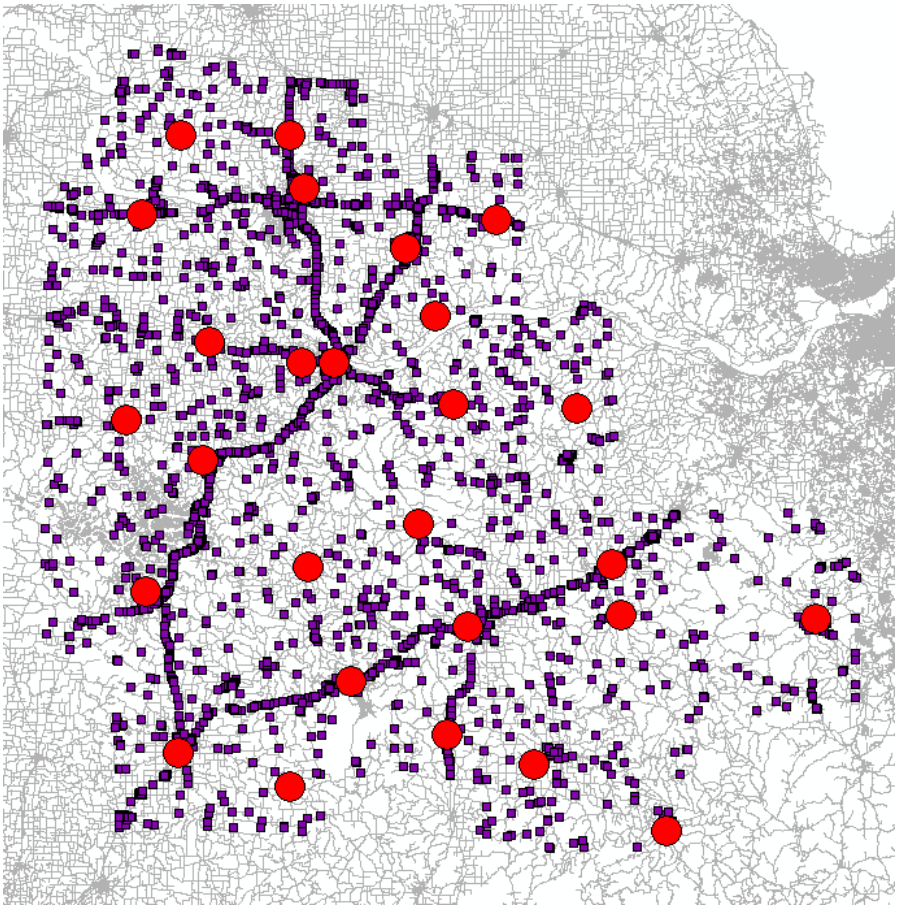


Figure 3.14 – Allocation of facilities and incidents

9. If there are no errors, press the Solve sign in the Network Analysis panel. If the network doesn't have disconnected segments it will calculate the shortest route from each node to the closest overnight location. As there are 4881 node, 4881 routes should be calculated. The solution is represented in Figure 3.15 with all routes with the shortest path.

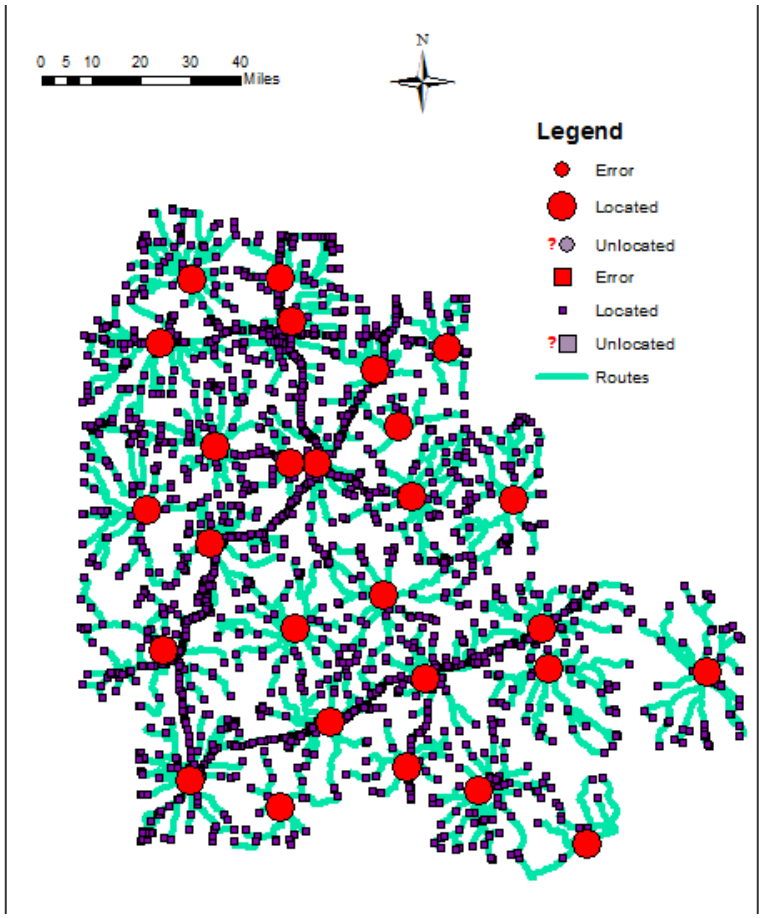


Figure 3.15 – Solution of closest facility tool

10. The next few steps are made to accommodate data modifications that are generated by the Closest Facility tool. These steps are necessary to make the data usable for the model. The created file Routes includes information about the closest overnight location (Name) to each node (ObjectID) and shortest distance to it (Total Length). The attribute table is shown in Figure 3.16. In our data excel file we include information about nodes by using their JNode ID. Routes calculated distances based on the ObjectID. It is also necessary to add data field JNodeID to the Route file, identifying the name of the county corresponding to each facility. The following steps should be performed in ArcGIS:

Routes

ObjectID	Shape	FacilityID	FacilityRank	Name	IncidentCurbApproach	FacilityCurbApproach	IncidentID	Total_Length
1	Polyline M	21	1	Location 1 - ELDON	1	1	1	18523.296696
2	Polyline M	21	1	Location 2 - ELDON	2	1	2	18550.293155
3	Polyline M	23	1	Location 3 - IBERIA	1	1	3	22401.664019
4	Polyline M	23	1	Location 4 - IBERIA	2	1	4	22614.641867
5	Polyline M	21	1	Location 5 - ELDON	1	1	5	22274.216767
6	Polyline M	21	1	Location 6 - ELDON	1	1	6	23614.393672
7	Polyline M	20	1	Location 7 - ST ROBERT	1	2	7	28259.617854
8	Polyline M	20	1	Location 8 - ST ROBERT	2	2	8	28206.018442
9	Polyline M	23	1	Location 9 - IBERIA	1	1	9	10275.278944
10	Polyline M	23	1	Location 10 - IBERIA	2	1	10	8190.884201
11	Polyline M	14	1	Location 11 - NEBO	1	2	11	12415.570936
12	Polyline M	14	1	Location 12 - NEBO	2	2	12	11286.349042
13	Polyline M	20	1	Location 13 - ST ROBERT	1	2	13	27432.458245
14	Polyline M	20	1	Location 14 - ST ROBERT	2	2	14	27402.338212
15	Polyline M	20	1	Location 15 - ST ROBERT	2	2	15	11651.801163
16	Polyline M	20	1	Location 16 - ST ROBERT	1	2	16	11630.674622
17	Polyline M	20	1	Location 17 - ST ROBERT	1	2	17	7329.450814
18	Polyline M	20	1	Location 18 - ST ROBERT	2	2	18	7374.544646
19	Polyline M	6	1	Location 19 - VIENNA	2	2	19	29971.648028
20	Polyline M	6	1	Location 20 - VIENNA	2	2	20	29860.415642
21	Polyline M	20	1	Location 21 - ST ROBERT	2	2	21	12249.157619
22	Polyline M	20	1	Location 22 - ST ROBERT	1	2	22	11733.589079
23	Polyline M	20	1	Location 23 - ST ROBERT	2	2	23	7478.562391
24	Polyline M	20	1	Location 24 - ST ROBERT	1	2	24	12250.768267
25	Polyline M	23	1	Location 25 - IBERIA	2	1	25	15494.934267

Figure 3.16 – Attribute table of Routes

10.1. Right Click on Routes -> Joins and Relates -> Join. In the processing window choose the field in Routes that the Join will be based on – ObjectID. Load table District4_Junctions (there is information in this file about JNodeID and ObjectID). Choose the field in Added Table – ObjectID. All the fields from District4_Junctions will be joined to Routes file. We just need to keep JNodeID.

10.2. In order to make other modifications it's better to create another file identical to routes. Right click on Routes->Export Data-> Give the name OvlD.

10.3. In order to add the field with county names, it's necessary to have a unique name for each facility. In OvlD file add a new field OVLName. Right click on this field-> Field Calculator-> type = (!Name!).replace(Location, " "). Then replace all numbers with this name.

10.4. For removing all extra spaces before Name add new field OName -> Right click on it -> Field calculator -> type =LTrim([OVLName]). This code will remove all spaces and name will be in the required format.

10.5. Right click on file OvlD -> Joins and Relates-> Join. In the processing window choose: OName field, D4_Sheds (file where there is information about counties and names of facilities), and field Name in that file. All attributes from D4_Sheds will be joined based on facility name. We just need to keep county name and can remove other fields.

11. After all these steps with modifications are done, the data is ready to be placed in excel file (Figure 3.17). It includes nodes ID (JnodeID), closest overnight location name, facility county name, FacilityID, Distances in miles and meters, and the node from the network that is closest to each facility.

ObjectID	NodeID	Closest overnight location Name	Overnight location County	FacilityID	Distance to closest facility (m)	Distance to closest facility (mi)	Closest node to facility
1	1	ELDON	MILLER	21	18523.2967	11.50982087	4719
2	2	ELDON	MILLER	21	18550.29316	11.52659566	4719
3	3	IBERIA	MILLER	23	22401.66402	13.91972197	692
4	4	IBERIA	MILLER	23	22614.64187	14.05206002	692
5	5	ELDON	MILLER	21	22274.21677	13.84053007	4719
6	6	ELDON	MILLER	21	23614.39367	14.6732758	4719
7	7	ST ROBERT	PULASKI	20	28259.61785	17.55967875	3045
8	8	ST ROBERT	PULASKI	20	28206.01844	17.52637368	3045
9	11	IBERIA	MILLER	23	10275.27894	6.384750077	692
10	12	IBERIA	MILLER	23	8190.884201	5.089569716	692
11	13	NEBO	LACLEDE	14	12415.57094	7.714663313	2035
12	14	NEBO	LACLEDE	14	11286.34904	7.012998704	2035
13	15	ST ROBERT	PULASKI	20	27432.45825	17.04570658	3045
14	16	ST ROBERT	PULASKI	20	27402.33821	17.02699089	3045
15	17	ST ROBERT	PULASKI	20	11651.80116	7.240079689	3045
16	18	ST ROBERT	PULASKI	20	11630.67462	7.22695229	3045
17	23	ST ROBERT	PULASKI	20	7329.450814	4.554300852	3045
18	24	ST ROBERT	PULASKI	20	7374.544646	4.582320807	3045
19	25	VIENNA	MARIES	6	29971.64803	18.62348294	7382
20	26	VIENNA	MARIES	6	29860.41564	18.55436647	7382
21	27	ST ROBERT	PULASKI	20	12249.15762	7.61125907	3045
22	28	ST ROBERT	PULASKI	20	11733.58908	7.290900246	3045
23	32	ST ROBERT	PULASKI	20	7478.562391	4.646954313	3045
24	34	ST ROBERT	PULASKI	20	12250.76827	7.612259878	3045

Figure 3.17 – Final Excel file with closest overnight locations to each node

4. GENETIC ALGORITHM MODEL

In this chapter, the following are presented: a) brief description of our GA, b) installation process of required programs and libraries for our program, and finally c) usage of our program interface.

4.1 Brief Description of the GA

We do not directly apply a GA to the RPPs. Therefore, as a first step, we converted RPP to equivalent TSP. Here, each required arc to be striped in RPP is represented as a node in TSP. The shortest distances among all nodes in TSP are calculated by Dijkstra's Algorithm. Meanwhile, when we make such conversions, we take the road segment's properties into account such as type of the road (divided/undivided), number of traverses, etc. After completing data preparation, we implemented Randomized Greedy Heuristics to get a better initial solution and thereby to improve the final solution produced by GA. In addition to this, improving GA's performance is strongly related to using appropriate GA operators as mentioned earlier. In this regard, we implement the exchange, insertion and direction mutations and binary crossover operations. Here, we do not take overnighting into account during application of those operators. However, later, overnighting locations are inserted to the striping sequence. By doing so, we take the advantage of similarities between solution sequences and thereby we decrease the computational time. The GA's operators we used are:

Binary Crossover: In binary crossover, we randomly choose 2 individuals (Individual 1 & Individual 2) from the population to produce a child as shown in Figure 4.1.

Positions of Chromosome	1	2	3	4	5	6	7	8	Fit
Individual 1	8	3	6	2	7	1	4	5	72
Randomly-generated Vector	1	1	1	0	1	0	0	1	
Individual 2	8	4	2	3	1	5	7	6	95
Child	8	3	6	4	7	2	1	5	90

Figure 4.1 – Binary crossover

Here, we have 8 required arcs in the original graph and each of them is represented as a node as shown in the table. Later, we randomly generate a binary vector. In this binary vector, whenever we see 1, we copy the corresponding node from individual 1 to the child. We make a list of nodes corresponding to 0 in the binary vector and we check how those nodes are ordered in Individual 2. Finally, we place the remaining nodes corresponding to 0 in the binary vector as seen in the order of Individual 2. The last column shows the fitness values of individuals and child.

Here, the computational expense of this operator is high, but it is important for the algorithm to sustain diversity to not become stuck at a local optima.

Insertion Mutation (IM): We apply IM to the child generated from binary crossover to produce a fitter individual and then we replace the worst individual in the population by this new individual. Here, we examine some possible insertion positions and select the best positions among those considered in terms of fitness value. The computational expense here is low.

Positions of Chromosome	1	2	3	4	5	6	7	8	Fit
Child	8	~~3~~	~~6~~	~~4~~	~~7~~	~~2~~	~~1~~	~~5~~	90
Positions 2,4,6		X		X		X			
Insertion Mutation 1	8	~~4~~	~~7~~	~~2~~	~~3~~	~~6~~	~~1~~	~~5~~	-15
Positions 3,5,6			1		1	1			
Insertion Mutation 2	8	4	3	6	7	2	1	5	-7
Continue Until No Further Improvement			Possible						68

Figure 4.2 – Insertion mutation

Exchange mutation (EM): We again randomly select an individual (I3) from the population to produce a child (C3), replace I3 by C3. We do a somewhat similar approach in Insertion mutation.

Positions of Chromosome	1	2	3	4	5	6	7	8	Fit
Individual 3	~~8~~	~~2~~	1	4	7	~~6~~	~~3~~	~~5~~	90
Positions 3,5			1		1				
Exchange Mutation 1	~~8~~	~~2~~	7	4	1	~~6~~	~~3~~	~~5~~	-6
Positions 2,6		1				1			
Exchange Mutation 2	8	6	7	4	1	2	3	5	-11
Continue Until No Further Improvement			Possible						73

Figure 4.3 – Exchange mutation

Direction mutation (DM): Once the final population is determined, we optimize the direction for each arc requiring an odd number of passes for each individual.

Positions of Chromosome	1	2	3	4	5	6	7	8	Fit
An Actual Pop. Member	(4,5)	(5,4)	(7,2)	(5,8)	(8,5)	(3,8)	(2,7)	(7,2)	90
A Final Population Member	~~8~~	~~2~~	~~3~~	~~6~~	~~7~~	1	~~4~~	~~5~~	68
Direction Mutation Point						1			
A Final Population Member	~~8~~	~~2~~	~~3~~	~~6~~	~~7~~	1	~~4~~	~~5~~	-5
An Actual Pop. Member	(4,5)	(5,4)	(7,2)	(5,8)	(8,5)	(8,3)	(2,7)	(7,2)	-5
Continue Until No Further Improvement			Possible						63

Figure 4.4 – Direction mutation

As a next step, we integrate the overnighting piece of our problem to our GA to calculate the actual fitness of each individual generated by GA's operators. Here, daily working hours, deadheading speed, striping speed are some of the important parameters that affect the end node of each striping day and thereby the overnight location of that day.

With the components of GA we have mentioned, we are able to provide an efficient striping schedule to reduce deadhead miles. It does not require manual intensive work except for entering some input parameters, and it enables the users to do some what-if analysis to examine the impact of resource levels such as daily working hours.

4.2 Installation Process of Required Programs and Libraries

The installation process is shown sequentially. These installation directions assume that the Python program has not already been installed on the computer. First, install **Python 2.7.2** from the website: https://www.python.org/download/releases/2.7.2/ using setup's default setting.

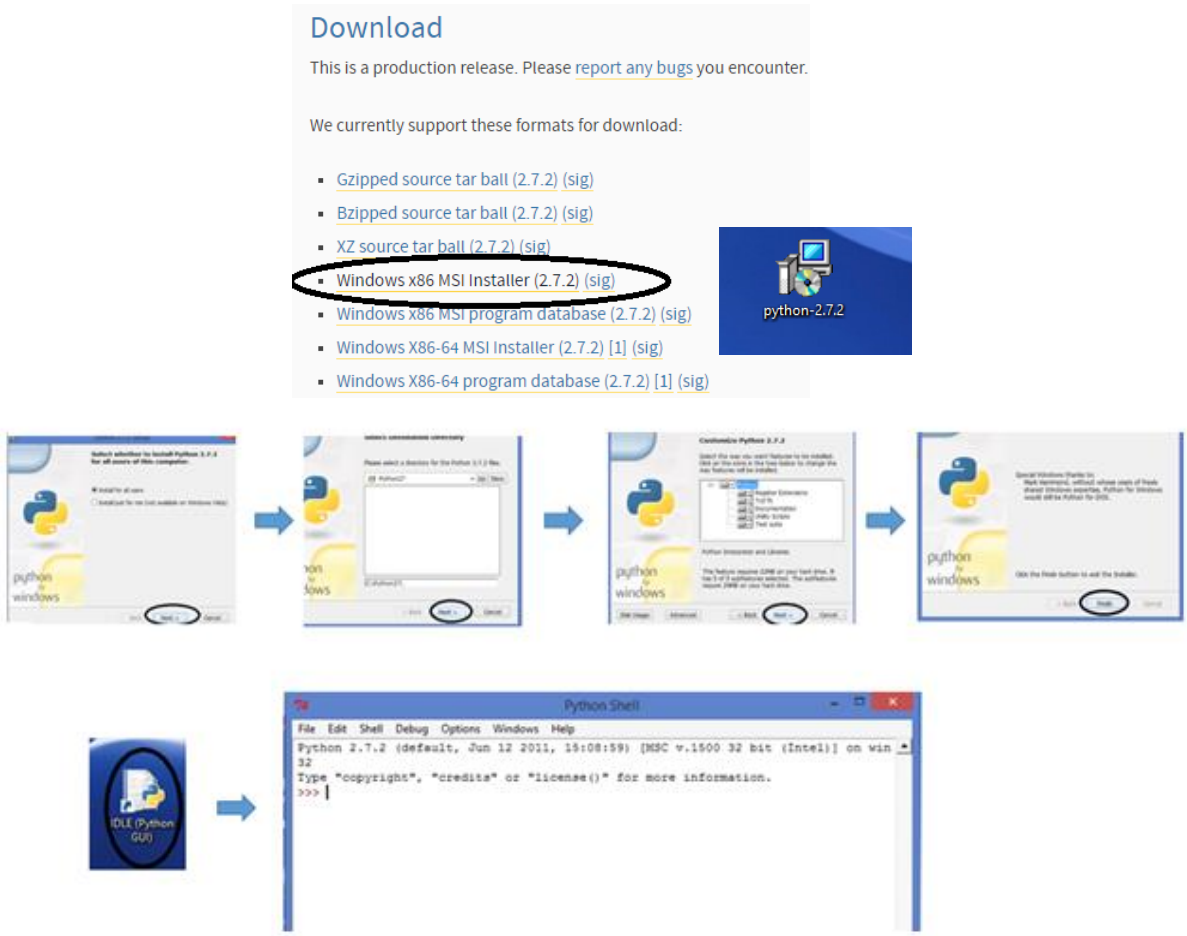


Figure 4.5 – Installing Python 2.7.2

Second, download PyPy to speed up the Python performance from the website: https://bitbucket.org/pypy/pypy/downloads . Here, there is no installation of this program, just download and copy the folder on the desktop. Downloaded version is: pypy-2.5.0-win32.zip.

pypy-2.5.0-linux-armhf-raring.tar.bz2	16.8 MB	mattip	120
pypy-2.5.0-linux-armel.tar.bz2	16.7 MB	mattip	167
pypy-2.5.0-linux.tar.bz2	18.3 MB	mattip	625
pypy-2.5.0-linux-armhf-raspbian.tar.bz2	17.0 MB	mattip	309
pypy-2.5.0-osx64.tar.bz2	17.0 MB	mattip	7972
pypy-2.5.0-win32.zip	19.7 MB	mattip	3548
pypy-2.5.0-src.tar.bz2	14.4 MB	arigo	2597
pypy3-2.4.0-src.zip	18.2 MB	pjenvey	1745
pypy3-2.4.0-src.tar.bz2	14.0 MB	pjenvey	2779
pypy3-2.4.0-win32.zip	17.7 MB	pjenvey	12439
pypy3-2.4.0-osx64.tar.bz2	12.4 MB	pjenvey	6472

Figure 4.6 – Installing pypy

After downloading pypy, the first step is to copy the downloaded pypy zip file to the desktop. Second, unzip the file and delete the zipped one. Third, open the pypy-2.5.0-win32 file and rename the file pypy-2.5.0-win32 in this file as MODOT. Fourth, cut and paste the MODOT file on the desktop. Now, we have the MODOT file on the desktop and delete empty file pypy-2.5.0-win32 on the desktop. The process is shown below in Figure 4.7:

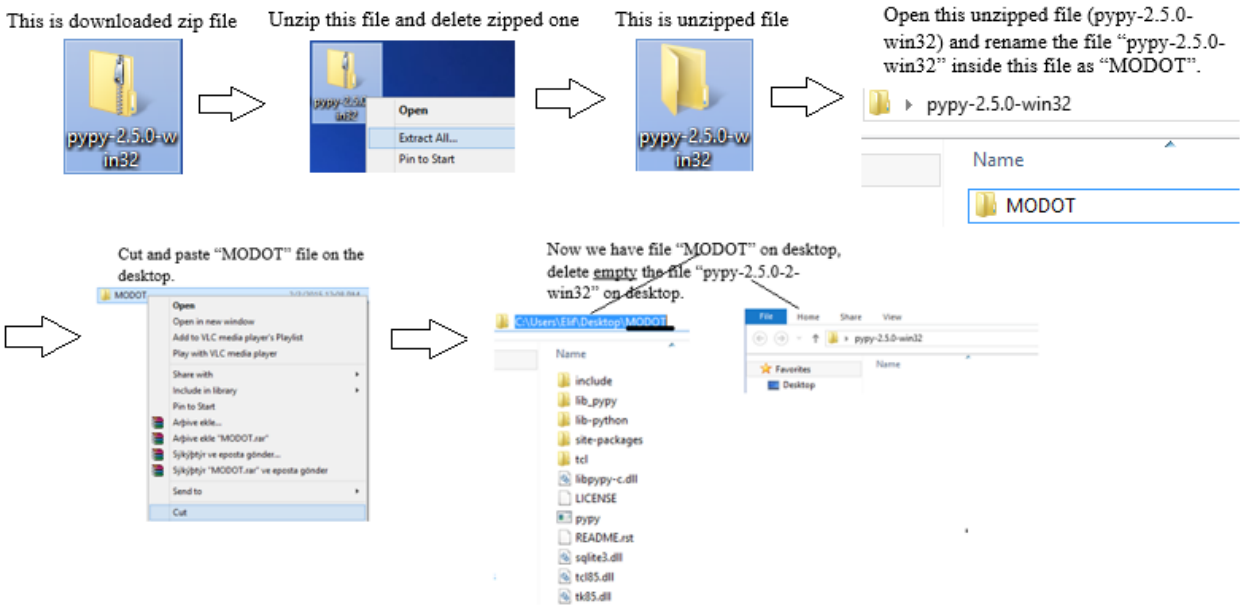


Figure 4.7 – Preparing pypy on the desktop

After downloading Python and pypy, next, add Python 2.7 to the path variable as following: Control Panel - System and Security – System - Advanced system setting - Environment Variables - Path. Screenshots of this process are shown in Figure 4.8.

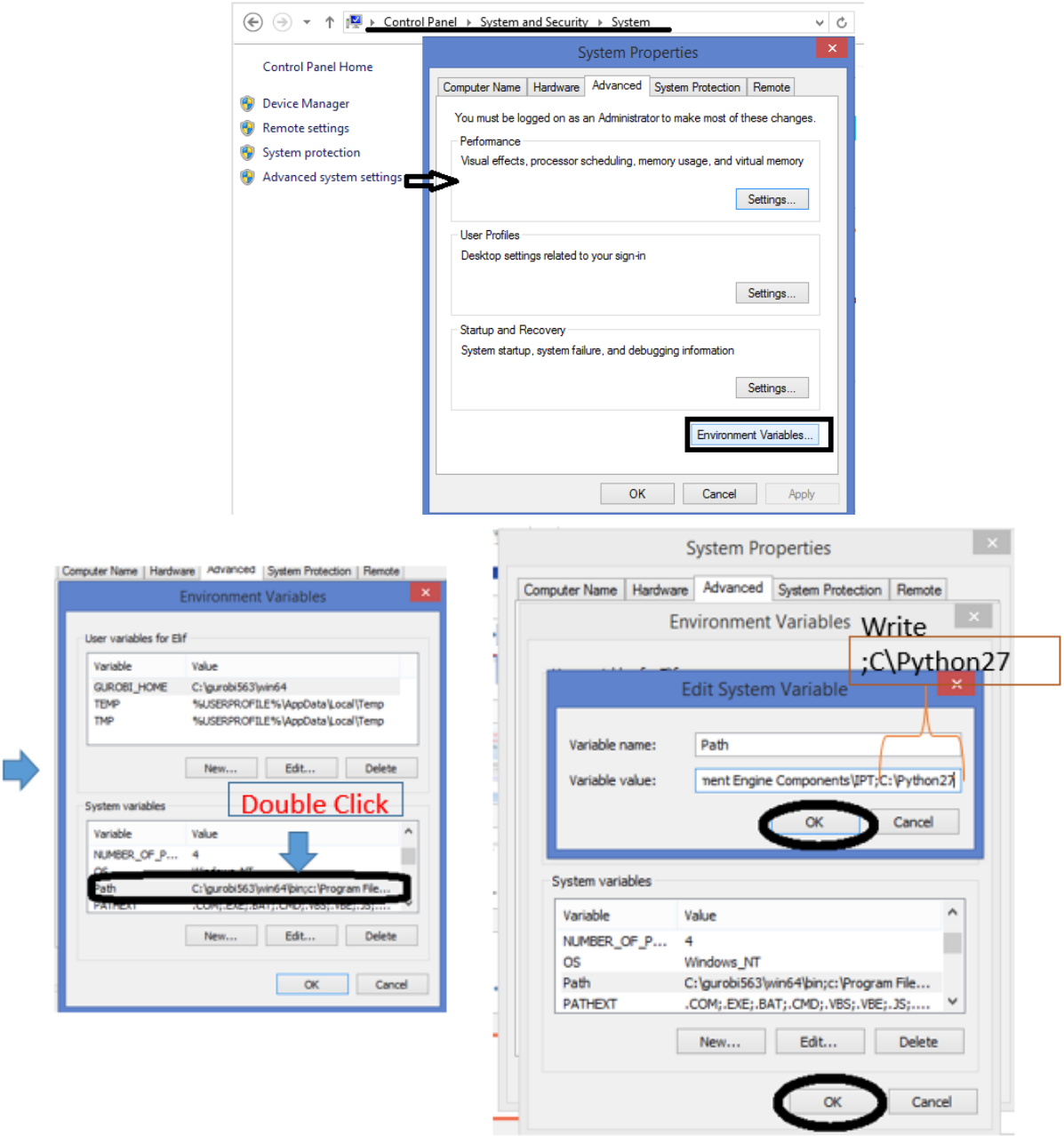


Figure 4.8 – Adding Python 2.7 to the path variable

Now, do the same process for pypy. Namely, add pypy to the path variable as following: Control Panel - System and Security – System - Advanced system setting - Environment Variables – Path (similar to the process shown in Figure 4.8 above). Write where you save the MODOT file as

Variable value, in my case, I saved it to the desktop which is C:\Users\Elif\Desktop\MODOT. Adding it to the path is shown below:

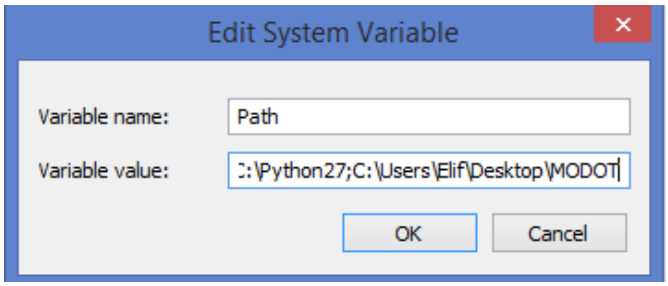


Figure 4.9 – Adding MoDOT model to the path variable

We have installed the required programs. Now, we need to install the Python libraries. First, install the python library numpy from the following website using setup's default settings:

https://pypi.python.org/pypi/numpy/1.6.1

File	Type	Py Version	Uploaded on	Size
numpy-1.6.1-cp27-none-macosx_10_6_intel.macosx_10_9_intel.macosx_10_9_x86_64.whl (md5)	Python Wheel	cp27	2014-07-30	11MB
numpy-1.6.1.tar.gz (md5)	Source		2011-07-24	2MB
numpy-1.6.1.win32-py2.5.exe (md5) Simple installer, no SSE instructions.	MS Windows installer	2.5	2011-07-24	2MB
numpy-1.6.1.win32-py2.6.exe (md5) Simple installer, no SSE instructions.	MS Windows installer	2.6	2011-07-24	2MB
numpy-1.6.1.win32-py2.7.exe (md5) Simple installer, no SSE instructions.	MS Windows installer	2.7	2011-07-24	2MB
numpy-1.6.1.win32-py3.1.exe (md5) Simple installer, no SSE instructions.	MS Windows installer	3.1	2011-07-24	2MB
numpy-1.6.1.win32-py3.2.exe (md5) Simple installer, no SSE instructions.	MS Windows installer	3.2	2011-07-24	2MB
numpy-1.6.1.zip (md5)	Source		2011-07-24	3MB

Figure 4.10 – Installing numpy library

After download, double click the exe file of this library and its installation process is straightforward. Next, we will download setuptools from the following website:

https://pypi.python.org/pypi/setuptools/0.6c11

The installation instruction can also be found from the website, but we will also explain here. Scroll down to the end of the page and download setuptools exe file shown in below.

File	Type	Py Version	Uploaded on	Size
setuptools-0.6c11-1.src.rpm (md5) built for redhat 4.3	RPM	any	2009-10-20	263KB
setuptools-0.6c11-py2.3.egg (md5)	Python Egg	2.3	2009-10-20	1MB
setuptools-0.6c11-py2.4.egg (md5)	Python Egg	2.4	2009-10-20	329KB
setuptools-0.6c11-py2.5.egg (md5)	Python Egg	2.5	2009-10-20	325KB
setuptools-0.6c11-py2.6.egg (md5)	Python Egg	2.6	2009-10-20	325KB
setuptools-0.6c11-py2.7.egg (md5)	Python Egg	2.7	2010-07-08	324KB
setuptools-0.6c11.tar.gz (md5)	Source		2009-10-20	250KB
setuptools-0.6c11.win32-py2.3.exe (md5)	MS Windows installer	2.3	2009-10-20	218KB
setuptools-0.6c11.win32-py2.4.exe (md5)	MS Windows installer	2.4	2009-10-20	222KB
setuptools-0.6c11.win32-py2.5.exe (md5)	MS Windows installer	2.5	2009-10-20	222KB
setuptools-0.6c11.win32-py2.6.exe (md5)	MS Windows installer	2.6	2009-10-20	222KB
setuptools-0.6c11.win32-py2.7.exe (md5)	MS Windows installer	2.7	2010-07-08	222KB

Figure 4.11 – Installing steuptools library

Now, download PIL from the website: http://www.pythonware.com/products/pil/ using setup's default setting.

- Version is: Python Imaging Library 1.1.7 for Python 2.7 (Windows only)

PIL 1.1.7

- **Python Imaging Library 1.1.7 Source Kit** (all platforms) (November 15, 2009)
- **Python Imaging Library 1.1.7 for Python 2.4** (Windows only)
- **Python Imaging Library 1.1.7 for Python 2.5** (Windows only)
- **Python Imaging Library 1.1.7 for Python 2.6** (Windows only)
- **Python Imaging Library 1.1.7 for Python 2.7** (Windows only)

Figure 4.12 – Downloading PIL library

Now, add numpy as we did for Python and pypy path variable: To add numpy, Write ;C\Python27\Script inside the variable value box.

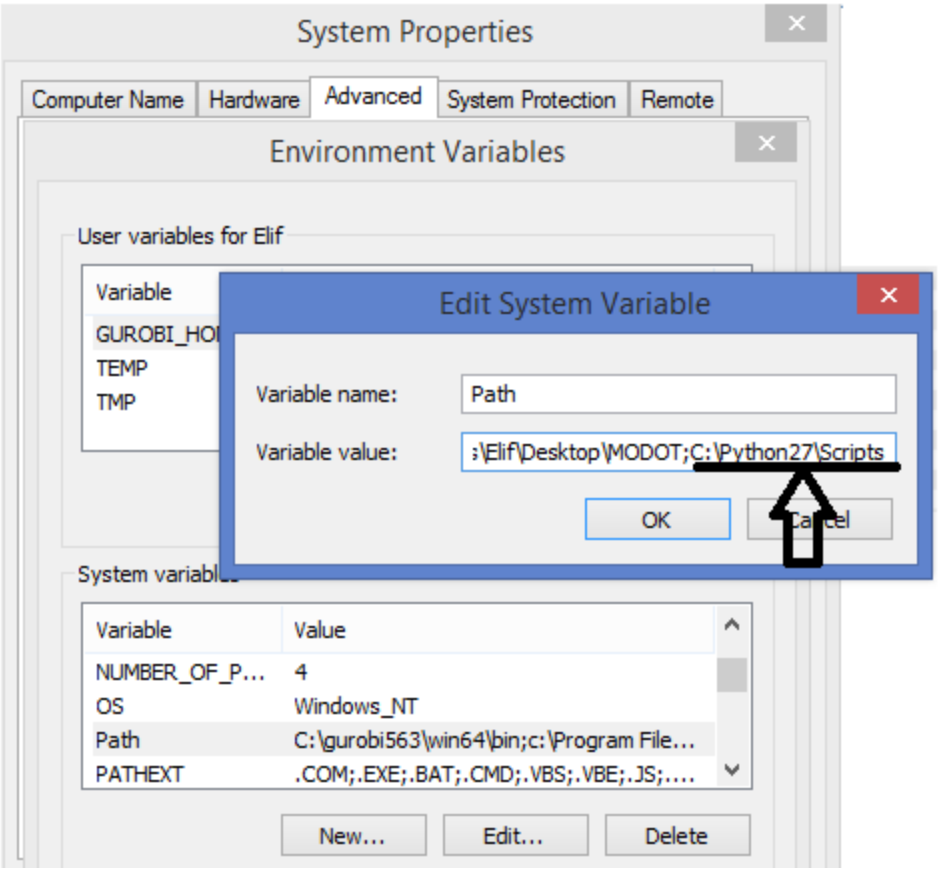


Figure 4.13 – Adding scripts to the path variable

Now, we can easily install other python libraries by using the easy_install tool in the setuptools we installed previously. This tool automatically download appropriate version of selected libraries for python from the internet. Open a command line window and install the following libraries:

Install networkx:

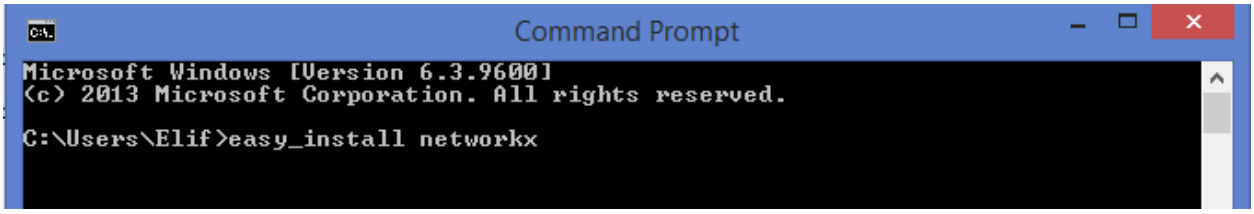


Figure 4.14 – Installing networkx

Install xlrd:

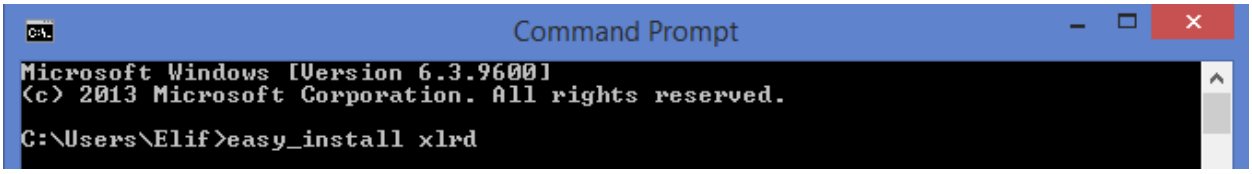


Figure 4.15 – Installing xlrd

Install xlwt:

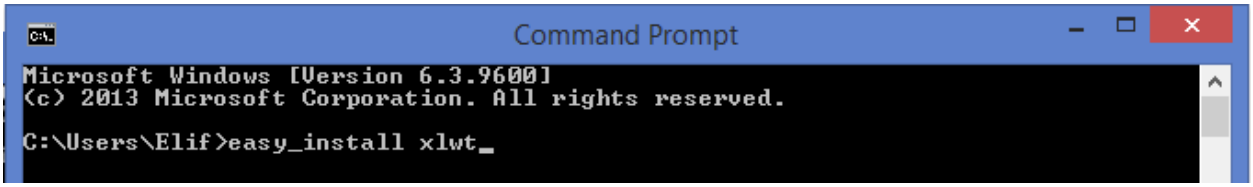


Figure 4.16 – Installing xlwt

Install openpyxl:

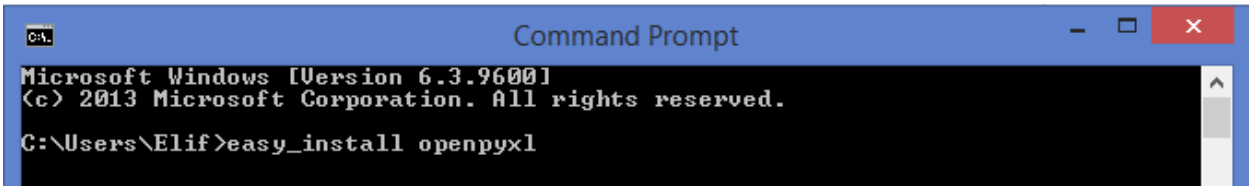


Figure 4.17 – Installing openpyxl

For pypy, we only need to install xlrd, but the installation process of this library to pypy is different from the installation to Python.

First, download xlrd from the website: https://pypi.python.org/pypi/xlrd/0.9.3

If the xlrd file downloads as xlrd-0.9.3.tar.gz, you will also need to install the application WinRAR x86 (32 bit) 5.30 beta 4 to extract the files; this application can be downloaded from the following website: http://www.rarlab.com/download.htm

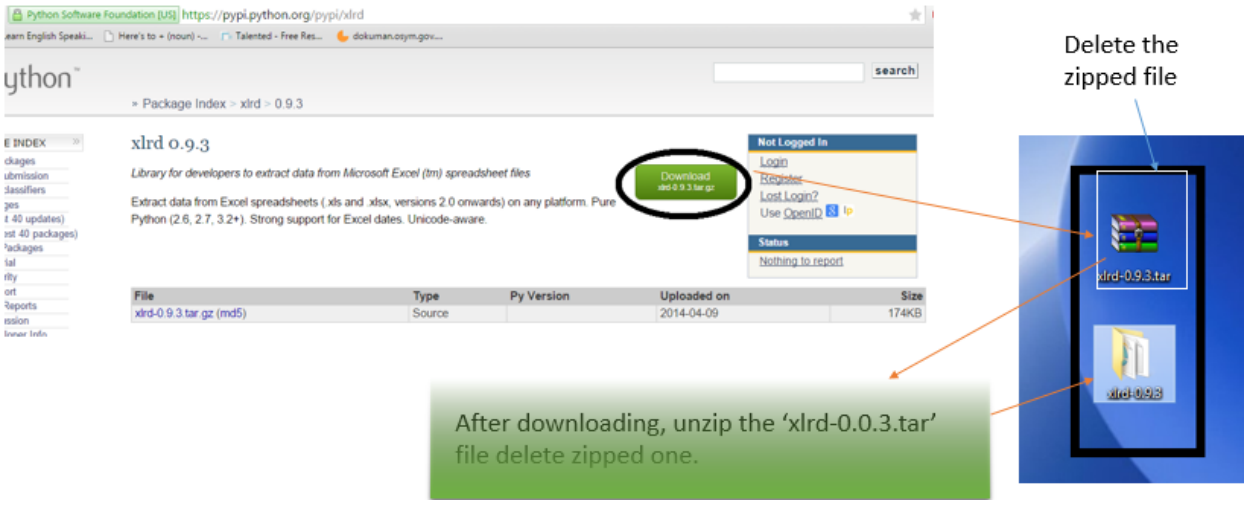


Figure 4.18 – Downloading xlrd for pypy

Second, open a command line window and change the directory to reflect where you unzipped file xlrd. Now, write: *pypy setup.py install* and click enter.

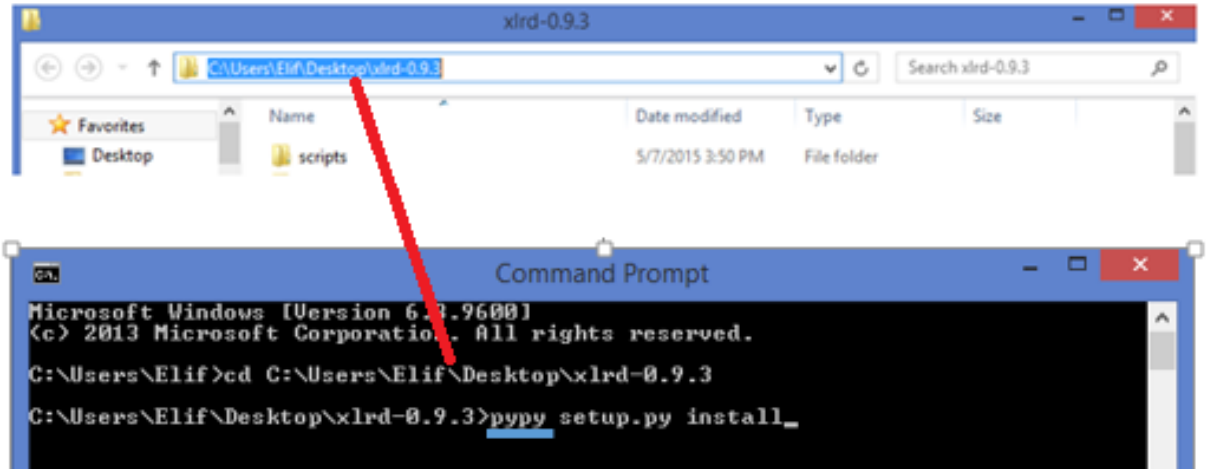


Figure 4.19 – Installing xlrd for pypy

After installation is completed, both the zipped and unzipped xlrd files can be deleted from the desktop.

Finally, we need to install all programs and libraries. The last step is to put all of the files provided by the MU research team (MODOT.py, DISTMATRIX.csv, etc.) inside the MODOT folder we created on the desktop (see Figure 4.7).

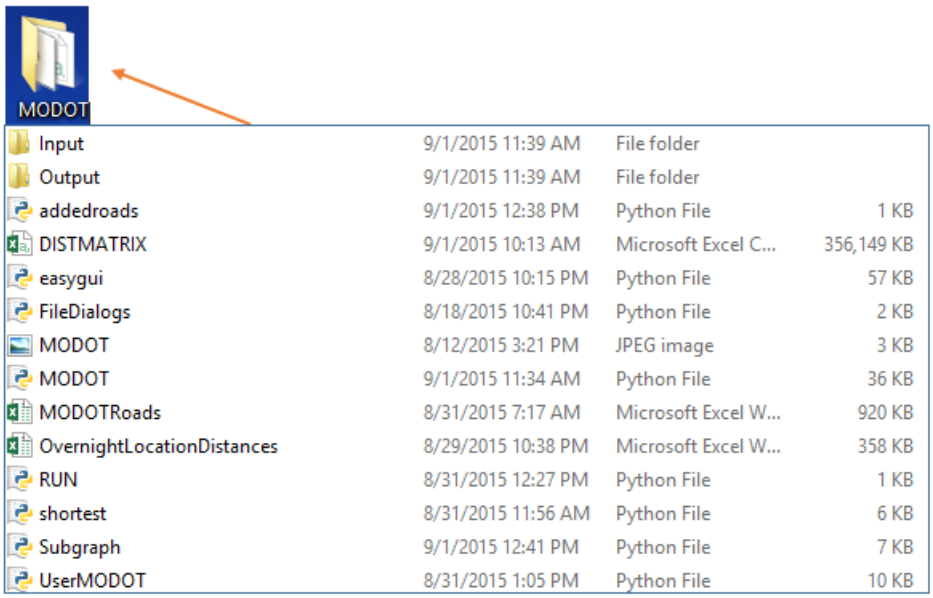


Figure 4.20 – Finalizing initialization

4.3 Using the Program Interface

Double-click UserMODOT.py to open the user interface.

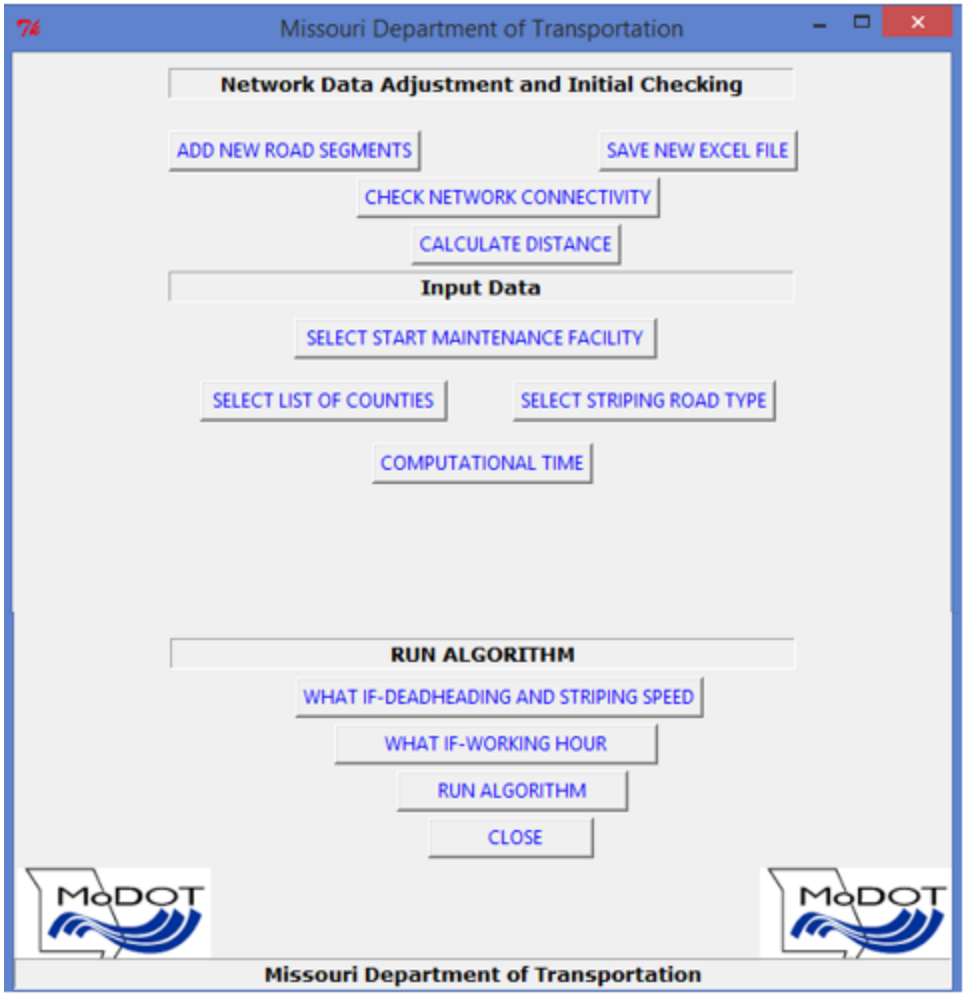


Figure 4.21 – User interface

4.3.1 Adding Arcs to the Network

This Section may be skipped unless the user wishes to modify the underlying network by adding new road segments. By "adding new road segments," we are referring to adding a new arc over which travel may occur, *not* simply changing the set of arcs that require (or don't require) striping.

If the user wishes to add arcs, as shown below, the decision support tool contains the button ADD NEW ROAD SEGMENTS, which will allow the user to add new road segments.

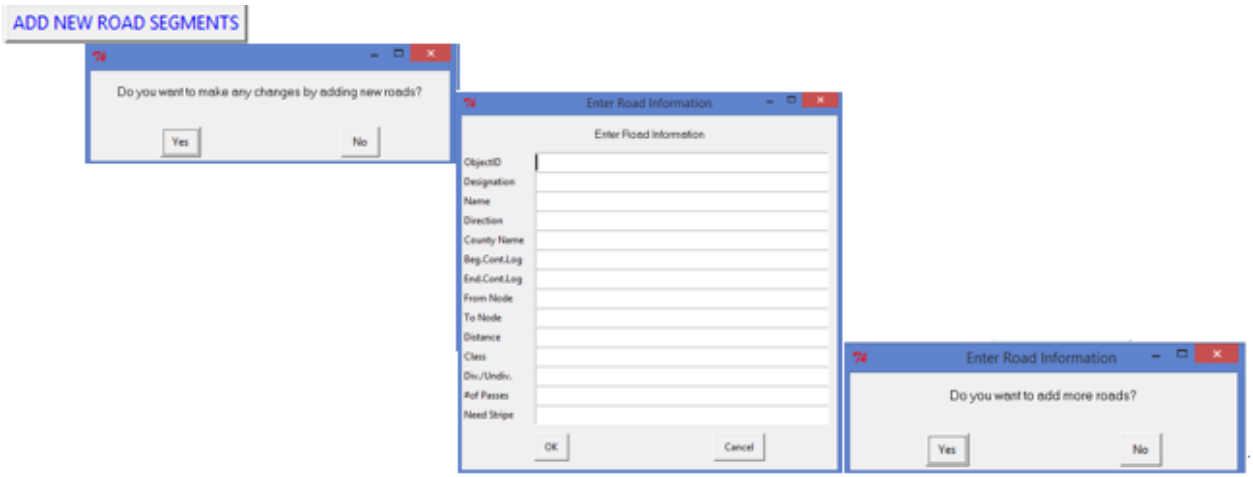


Figure 4.22 – Adding new road segments

Start by adding the necessary input information for the new road segment. After adding new roads, we need to make sure that the file is saved by using the button SAVE NEW EXCEL FILE. It adds the date to the name of the MODOTRoads file. For example, if the initial file name is MODOTRoads.xlsx, if you add new roads to the network the new Excel file will have a timestamp appended to the end of the file name, as shown in Figure 4.23:

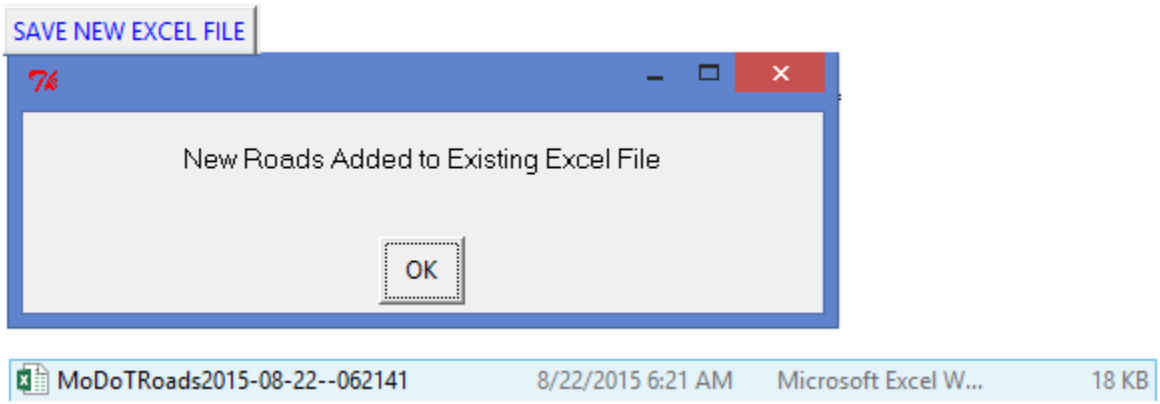


Figure 4.23 – Saving new road segments

After this process, we have to make sure that we have a fully connected network again. In other words, we do not want to have a cut in the network. Therefore, the button CHECK CONNECTIVITY allows us to see if the network is fully connected or not by generating a Connectivity text file, which specifically shows the node numbers for every separated graph if there is a network disconnectivity. Hence, the user is able to identify which segments need to be reconnected.

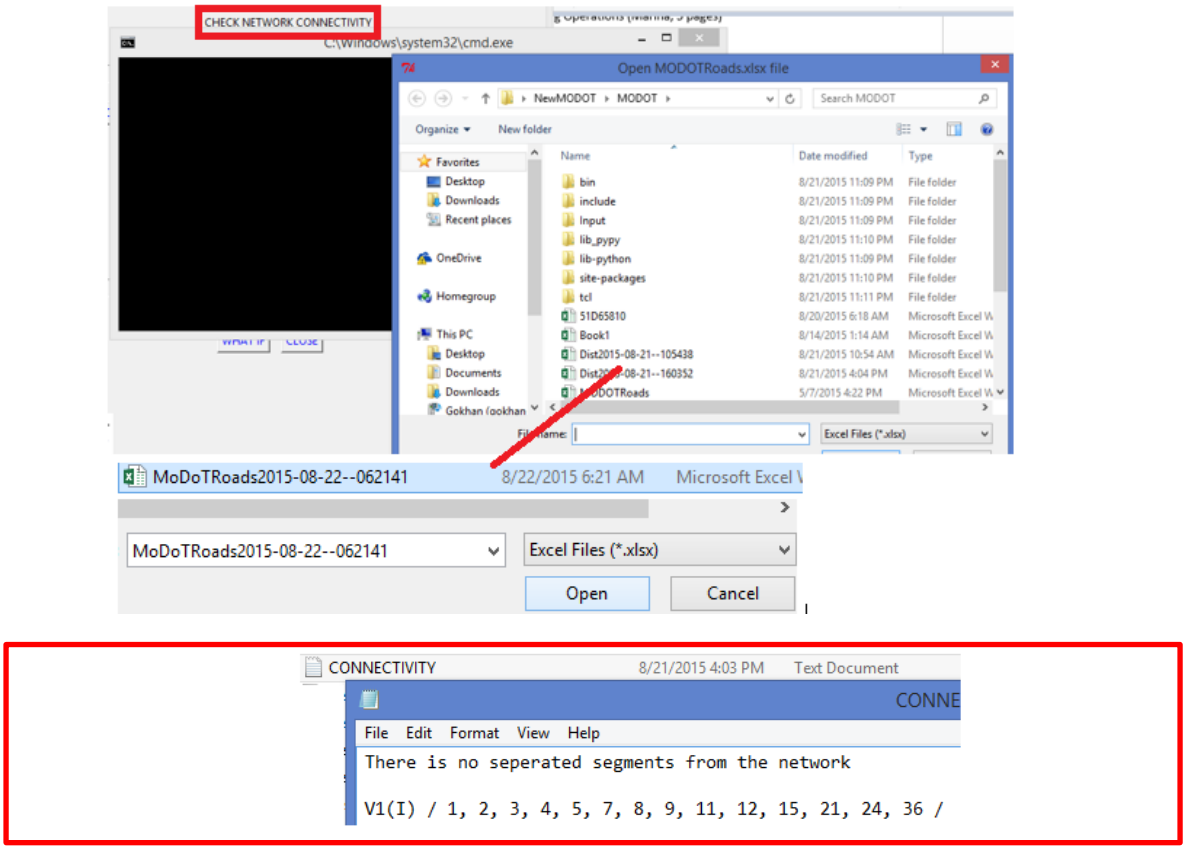


Figure 4.24 – Confirming network connectivity after adding new road segments

Let us explain the above screenshots. Click the button Check Network Connectivity and then open the newly-generated Excel file which has newly added road segments. Finally, it creates a Connectivity text file.

Finally, if MODOT makes any change from the data we already generated, the next step is to calculate new shortest distances. Generating distances takes approximately 2 days using a laptop with Intel 4 CPUs and 4 GB RAM, 64 bit OS. However, this process is done only once. Once this file is obtained, it can be used in all future analysis (until additional arcs are added to the network).

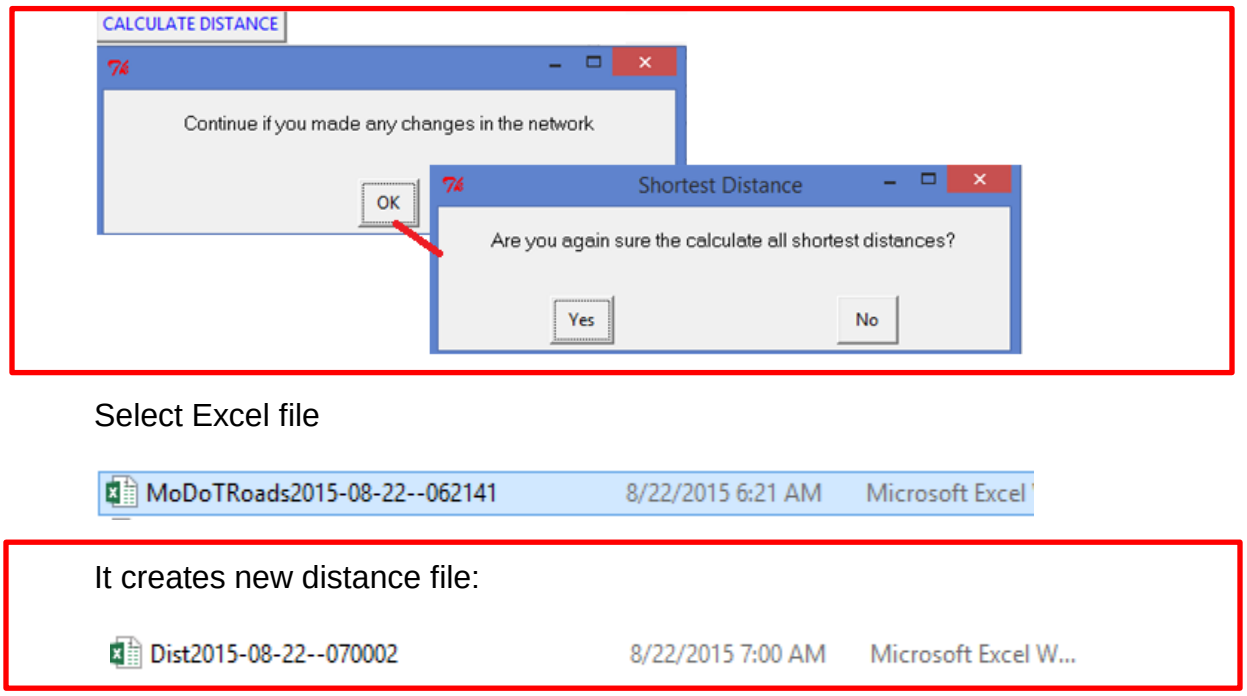


Figure 4.25 – Calculating shortest distances table

4.3.2 Preparing the Model to Run

Next, consider the Input Data section shown as below:

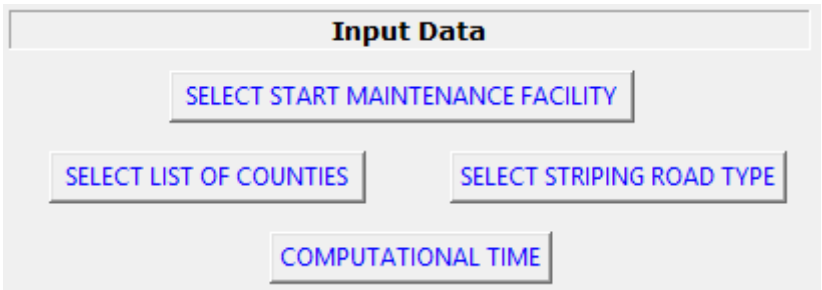


Figure 4.26 – Input Data section

This section is pretty straightforward for the user. First, select the list of counties over which the user wants to determine a striping schedule. In Chapter 3, we explained the data preparation process, how we imported data from ARCGIS and how we created the MODOTRoads Excel file. Note that in this Excel file, the last column showing the striping decision for each segment for the planning period needs to be updated manually.

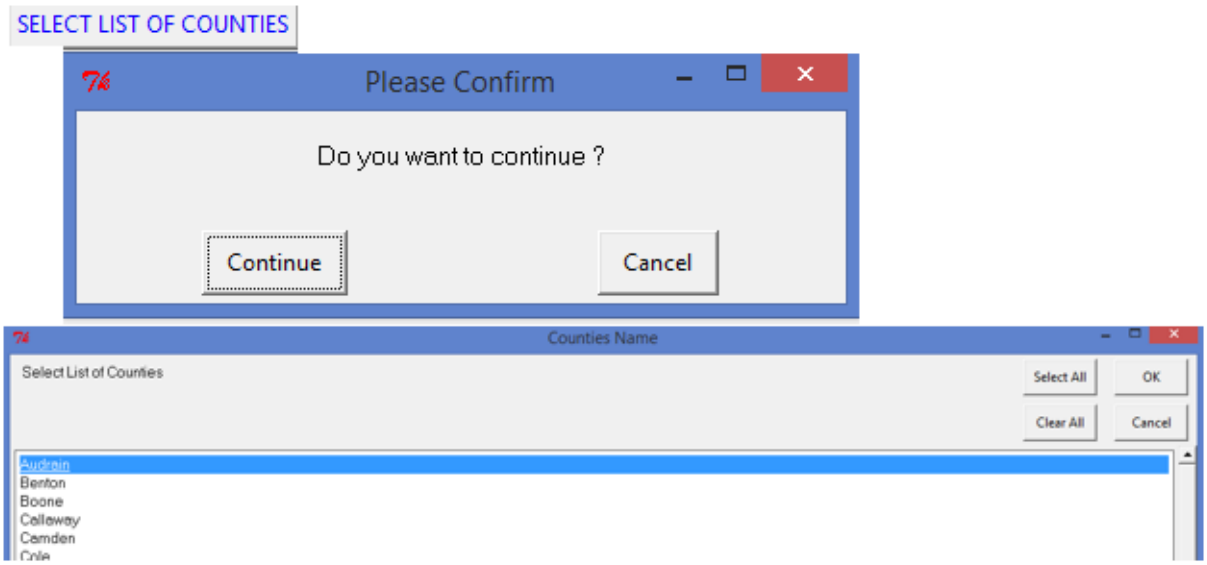


Figure 4.27 – Selecting counties for striping scheduling

Second, select the starting maintenance building where the striping vehicle is parked for the beginning of the striping season.

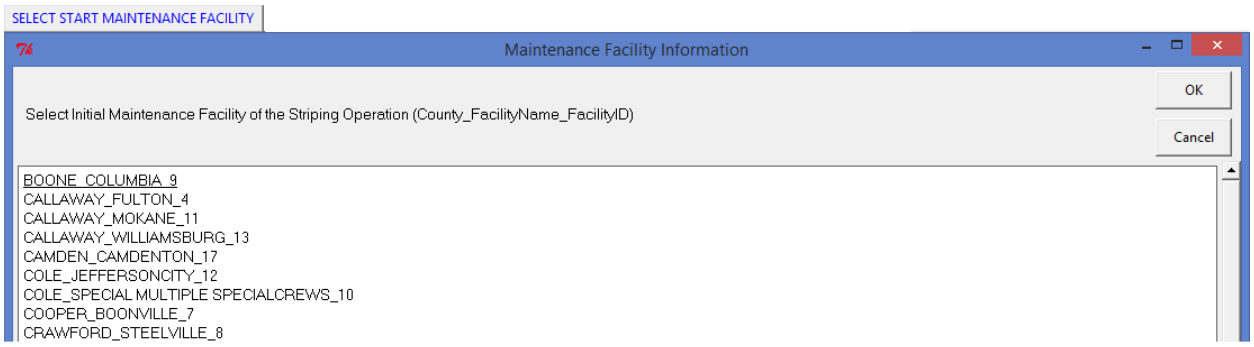


Figure 4.28 – Selecting initial location of striping crew

Third, select road type:

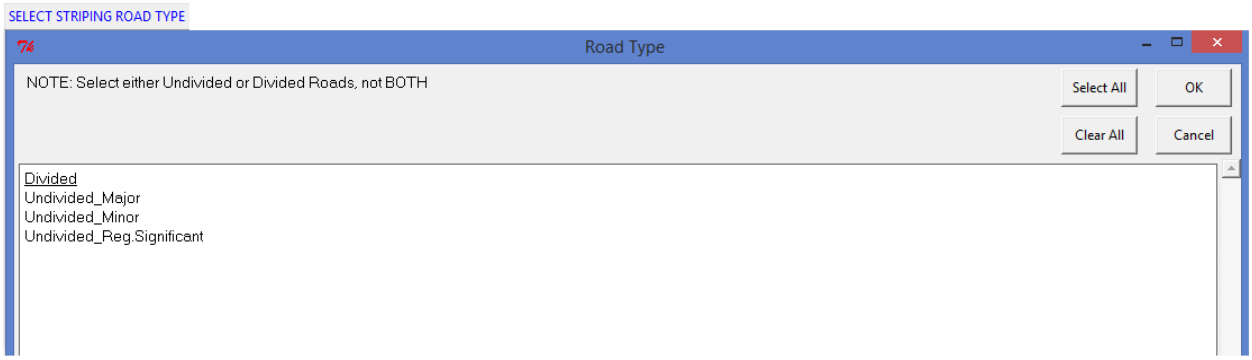


Figure 4.29 – Selecting road types for striping scheduling

Finally, enter maximum-allowable computational time (in minutes):

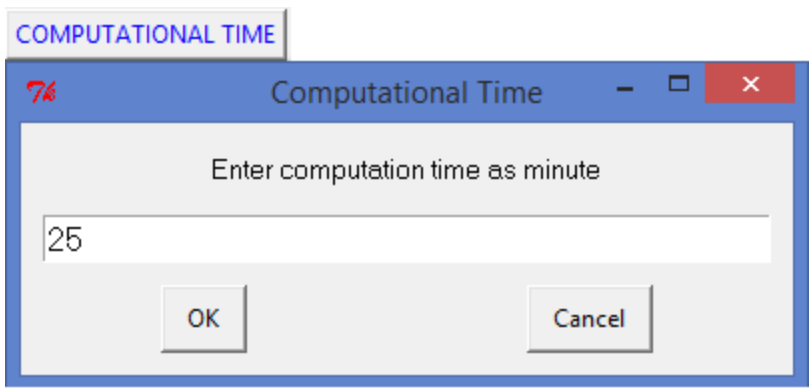


Figure 4.30 – Selecting maximum-allowable computational time

4.3.3 Running the Model

Now, we can run the model by clicking on the RUN ALGORITHM button.

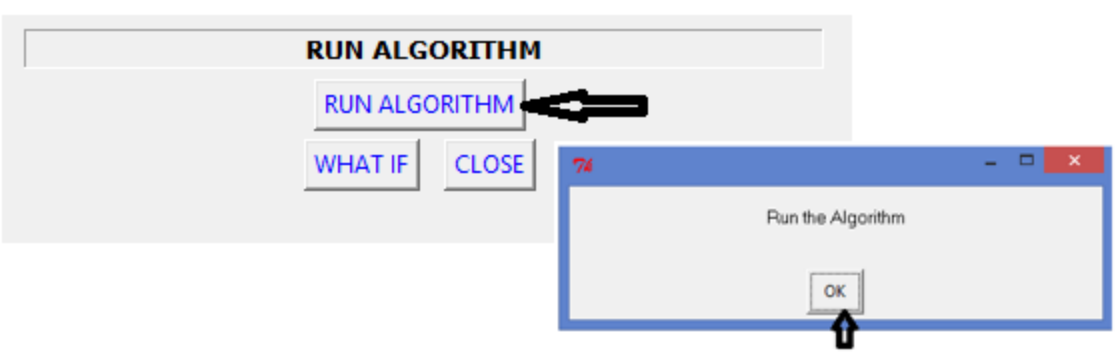


Figure 4.31 – Running the model

We first need to select which road network we want to analyze (typically, MODOTRoads.xlsx).

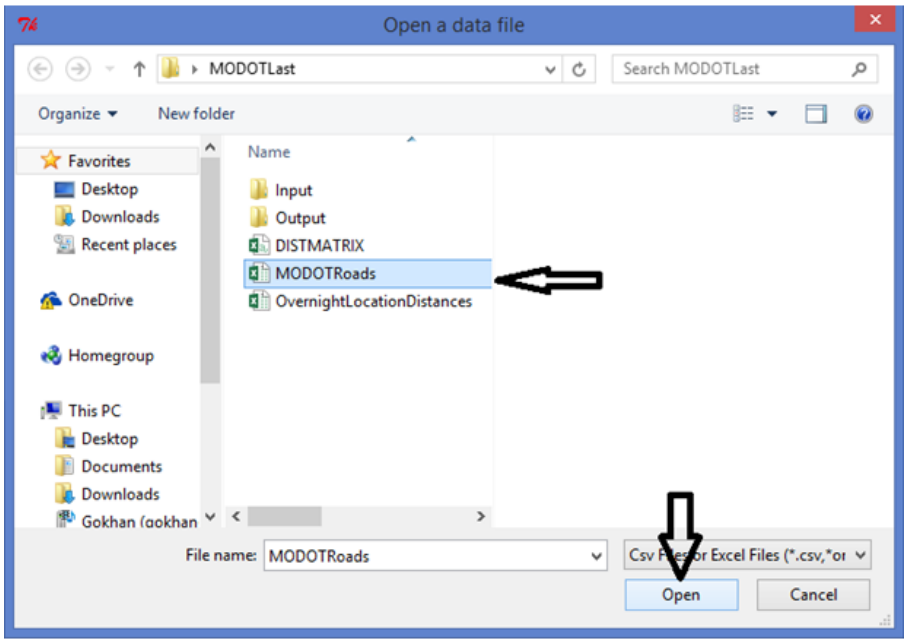


Figure 4.32 – Selecting the road network

We next need to select the distance table between the nodes of our road network and the set of allowable striping vehicle overnighting locations (typically, OvernightLocationDistances.xlsx).

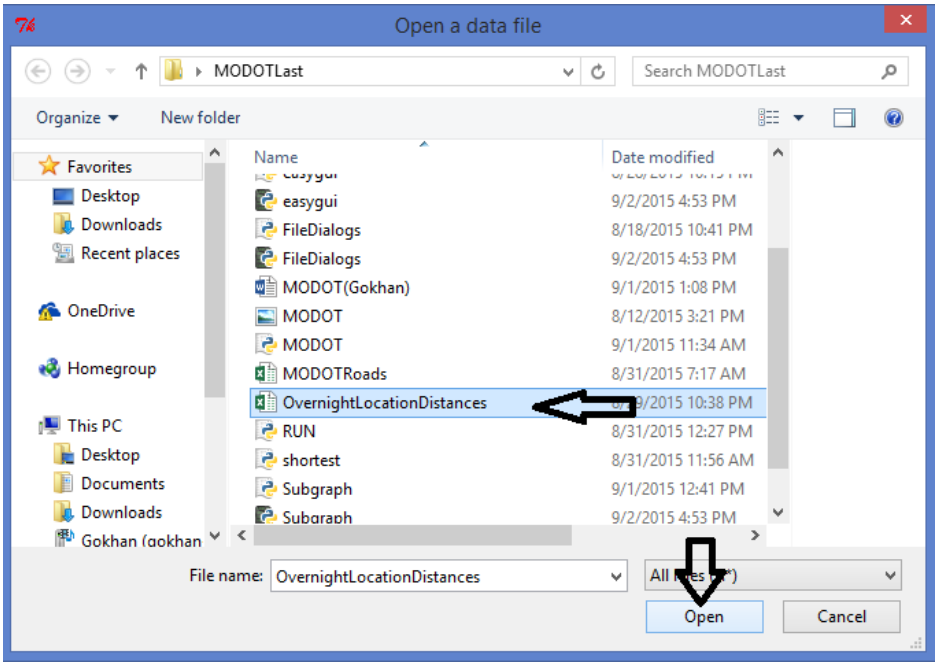


Figure 4.33 – Selecting distance table for overnighting locations

Finally, we need to select the distance table corresponding to our road network (typically, DISTMATRIX.csv).

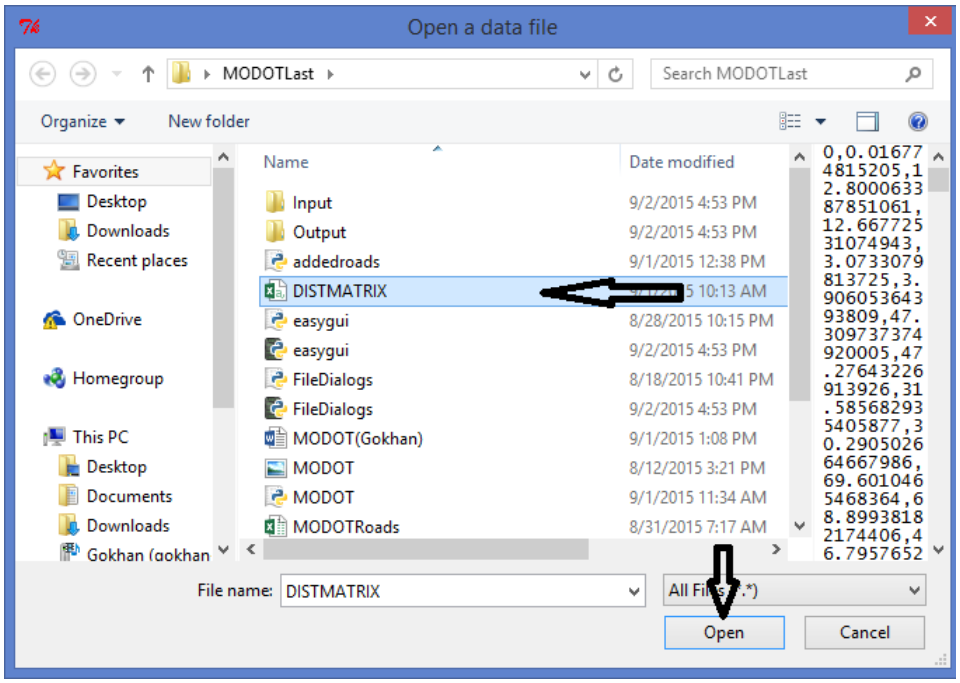


Figure 4.34 – Selecting the distance table

The model now runs, determining an ordered schedule (i.e., a sequence) of road segments to be striped that minimizes the total deadhead miles traveled. When the model has finished running, the PYTHON.EXE window will display: The 'Striping Plan' is in 'Output' folder.

Figure 4.35 – Display when model has finished building running

We can now find the output in the Output folder. The output file first presents overall statistics on the total striping days, total striping time, and the model run's computational time. Then, for each day, the output lists the maintenance building where the day's operations began, and the order in which segments should be striped for that day. Segments are identified as Direction_Highway ID_County Name_starting mile marker_stopping mile marker; for example, S_63C_BOONE_0.468_0.66.

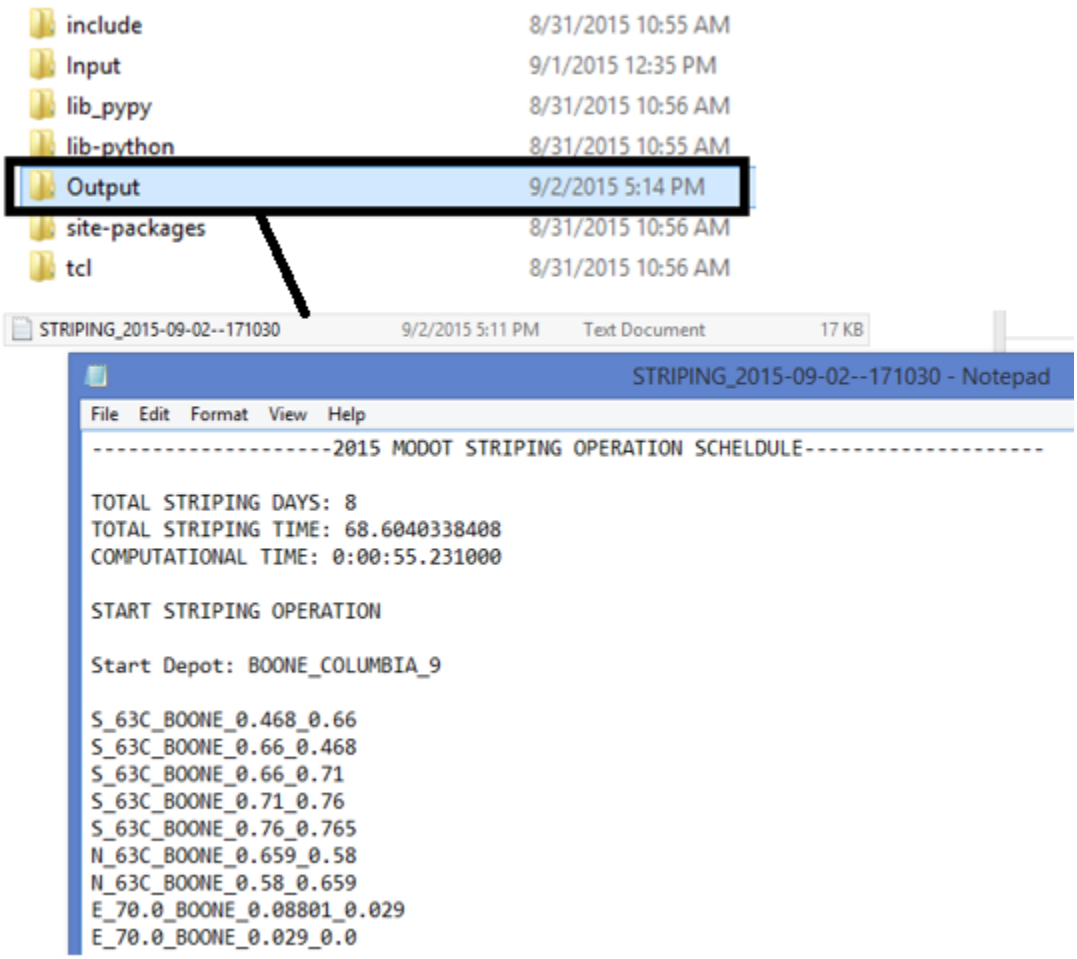


Figure 4.36 – Opening the model output

4.3.4 Performing What-if Analysis

We have designed the tool to allow MoDOT users to perform two types of what-if analyses: (1) the user can change the allowable daily working hours, and (2) the user can change the speed at which striping vehicles travel while striping and deadheading. These what-if analyses can be used to determine the impact of such changes on system outputs such as the number of days required to complete striping operations, the total distance traveled, etc. Other what-if capabilities could be generated in the future, should MoDOT so request.

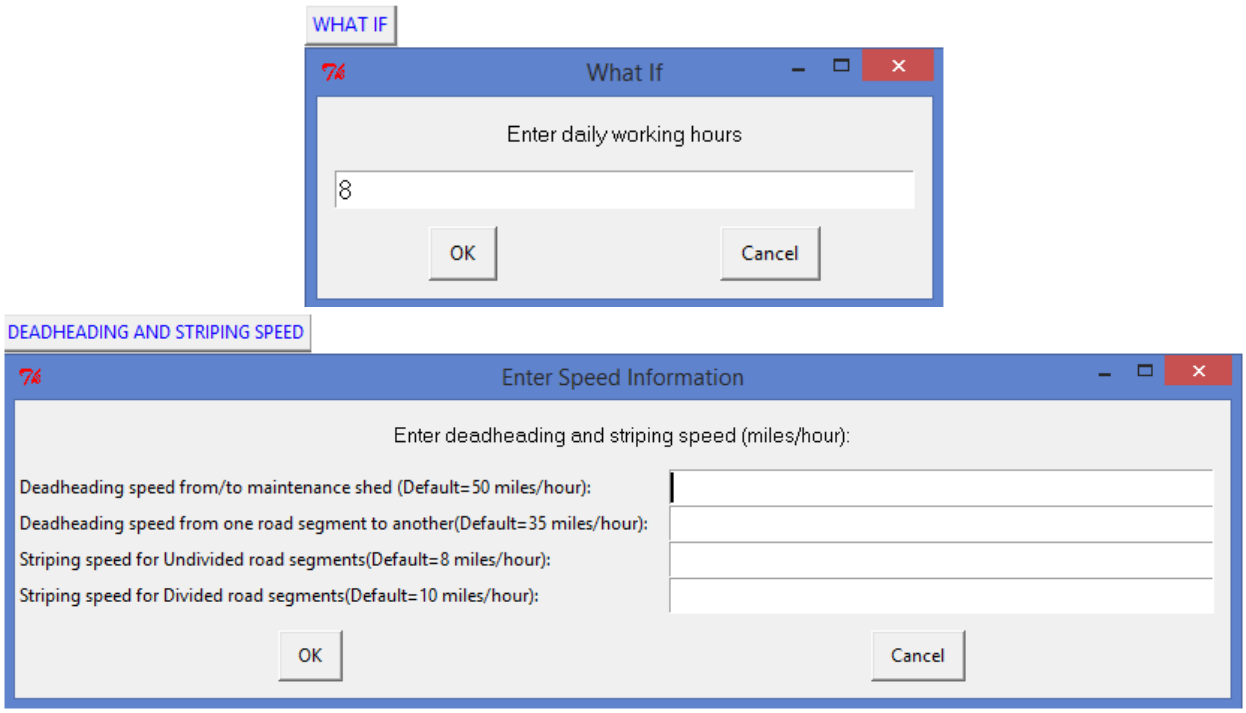


Figure 4.37 – Performing what-if analyses

5. CONCLUSIONS

Road line striping operations generate a significant workload for MoDOT. The requirement for each road striping crew to replenish its stock of paint and other consumable items from a bulk storage facility and the possible requirement of traveling unrequired roads to reach the roads that need striping generate the potential for inefficiencies in the form of "deadhead miles" that road striping crew vehicles must travel while not actively applying pavement markings.

We developed an optimization-based decision support tool, implementing Genetic Algorithm techniques, to identify a minimum-distance striping schedule that satisfies the requirements of MoDOT striping operations. Despite the fact that some factors remain unrepresented in the model (e.g., highway ramps requiring striping), the current results of our model can be used to help MoDOT more quickly calculate a striping schedule and dynamically respond to unexpected conditions, such as schedule disruptions which occur due to, e.g., chip seal operations not completing on the scheduled date. While MoDOT does not have records from which a comparison between the current and proposed system can be evaluated, the advantage of the Genetic Algorithm is apparent in the alleviation in time and effort dedicated to manually developing a striping schedule. Assuming MoDOT is able to manually calculate an optimal striping schedule to minimize the total distance traveled, the task still represents an exceedingly lengthy and laborious task. As such, this model provides an ability to significantly eliminate the effort necessary to produce said striping schedule as well as test what-if scenarios examining the impact of changing resource levels, policies, etc.

The inefficiencies due to deadhead miles are manifested not only in additional, unnecessary miles traveled by road striping crews, but also in the required capacity for road striping crews and equipment. Were a more-efficient utilization of road striping equipment possible, MoDOT could potentially reduce costs by reducing its inventory of road striping assets, without reducing the frequency with which it reapplies pavement markings to Missouri highways. In this regard, the what-if capabilities of our model could be useful beyond solely the creation of striping operations schedules.

From an academic standpoint, the research involved with our modification of the Rural Postman Problem to satisfy the conditions of the *slow-moving multi-pass postman problem with overnighting* constitutes the formulation of a logistics model absent from the literature. As such, this formulation could be useful in the evaluation of other systems.

REFERENCES

Ahuja, R. K., Orlin, J. B., and Tiwari, A. (2000). A greedy genetic algorithm for the quadratic assignment problem. *Computers & Operations Research* 27(10): 917-934.

Benavent, E. and Soler, D. (1999). The Directed Rural Postman Problem with Turn Penalties. *Transportation Science*: 33:4.

Benavent, E., Corberán, A., Plana, I., and Sanchis, J. M. (2009). Min-Max K-vehicles Windy Rural Postman Problem. *Networks*, 54(4): 216-226.

Corberán, A., Oswald, M., Plans, I., Reinelt, G., and Sanchis, J. M. (2012). New results on the Windy Postman Problem. *Mathematical Programming*, 132(1): 309-332.

Corberán, A., Plana, I., and Sanchis, J. (2005a). *A branch and cut algorithm for the windy general routing problem*. Technical report, Department of Statistics and Operational Research, University of Valencia.

Corberán, A., Plana, I., and Sanchis, J. (2005b). *The windy general routing polyhedron: A global view of many known arc-routing polyhedra*. Technical report, Department of Statistics and OR, University of Valencia.

Cornillier, F., Boctor, F., and Renaud, J. (2012). Heuristics for the multi-depot petrol station replenishment problem with time windows. *European Journal of Operational Research* 220: 361–369.

Dantzig, G., and Ramser, J. (1959). The truck dispatching problem. *Management Science* 6: 80–91.

Eiselt, H. A., Gendreau, M., and Laporte, G. (1995a). Arc routing problems, part I: The chinese postman problem. *Operations Research*, 43(2), 231-242. Retrieved from http://www.jstor.org.proxy.mul.missouri.edu/stable/171832

Eiselt, H. A., Gendreau, M., and Laporte, G. (1995b). Arc routing problems, part II: The rural postman problem. *Operations Research*, 43(3), 399-414. Retrieved from http://pubsonline.informs.org/doi/pdf/10.1287/opre.43.3.399

Groves, G. W., and van Vuuren, J. H. (2005). Efficient heuristics for the rural postman problem. *ORiON: The Journal of ORSSA*, 21(1): 33-51.

Guan, M. (1962). Graphic Programming using odd or even points. *Chinese Mathematics*. 1: 273-277.

Hoffman, K. L., Padberg, M., and Rinaldi, G. (2001). Traveling salesman problem. Retrieved from seor.gmu.edu/~khoffman/TSP_Hoffman_Padberg_Rinaldi.pdf

Holland, J. H. (1975). *Adaptations in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor.

Jang, W., Noble, J. S., and Hutsel, T. (2010). An integrated model to solve the winter asset and road maintenance problem. *IIE Transactions*, 42(9), 675-689. Retrieved from http://www.tandfonline.com/doi/abs/10.1080/07408171003705375

Kouskoulas, V. (1988). An optimization model for pavement marking system. Amsterdam: *European Journal of Operational Research* 33:298-303.

Laporte, G. (2010). A concise guide to the traveling salesman problem. *Journal of the Operational Research Society*, 61, 35-40. Retrieved from http://www.jstor.org.proxy.mul.missouri.edu/stable/40540226

Letchford, A. N. and Eglese, R. W. (1997). The rural postman problem with deadline classes. *European Journal of Operational Research* 105: 390-400.

Li. L. Y. O., and Eglese, R. W. (1996). An interactive algorithm for vehicle routeing for winter – gritting. *The Journal of the Operational Research Society*, 47(2), 217-228. Retrieved from http://www.jstor.org.proxy.mul.missouri.edu/stable/2584343

Missouri Department of Transportation. (2015a). Meet MoDOT. Retrieved from http://www.modot.org/newsandinfo/documents/MeetMoDOT.pdf

Missouri Department of Transportation. (2015b). *Engineering Policy Guide, Category 620: Pavement Marking* [Online]. Retrieved from http://epg.modot.org/index.php?title=Category:620_Pavement_Marking

Montebello, D., and Schroeder, J. (2000). Cost of pavement marking materials. *Minnesota Local Road Research Board Synthesis Report*, 2000-11. Retrieved from http://www.lrrb.org/media/reports/200011.pdf

Monroy-Licht, I. M., Amaya, C. A., and Langevin, A. (2013). The rural postman problem with time windows. Centre interuniversitaire de recherche sur les reseaux d'entreprise, la logistique et le transport (CIRRELT), 2013-69. Retrieved from https://www.cirrelt.ca/DocumentsTravail/CIRRELT-2013-69.pdf

Onyango, M. A., Malyuta, D., and Owino, J. (2014). Verification of pavement marking degradation models using eastern tennessee pavement marking retroreflectivity data. *Science, Computing and Telecommunications (PACT): Pan African Conference*, Arusha.

Orloff, C. S. (1974). A Fundamental Problem in Vehicle Routing. *Networks*, 4(1): 35-64.

Pieterse, V., and Black, P. E. (2014a). Chinese postman problem. *Dictionary of Algorithms and Data Structures* [Online]. Retrieved from http://www.nist.gov/dads/HTML/chinesePostman.html

Pieterse, V., and Black, P. E. (2014b). Traveling salesman. *Dictionary of Algorithms and Data Structures* [Online]. Retrieved from http://www.nist.gov/dads/HTML/travelingSalesman.html

do Rosário Moreira, M., and Ferreira, J.S.. (2010). *A Genetic Algorithm for the Undirected Rural Postman Problem*. Technical Report, Instituto de Engenharia de Sistemas e de Computadores do Porto, Portugal.

Sadegheih, A. (2006) Scheduling problem using genetic algorithm, simulated annealing and the effects of parameter values on GA performance. *Applied Mathematical Modelling* 30(2): 147-154.

Smadi, O., Souleyrette, R., Ormand, D., and Hawakins, N. (2008). Pavement Marking Retroreflectivity: Analysis of Safety Effectiveness. *Transportation Research Record: Journal of the Transportation Research Board*. No. 2056. Transportation Research Board of the National Academies, Washington, DC. pp. 17-24. trrjournalonline.trb.org/doi/pdf/10.3141/2056-03

Solomon, M. M. (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, 35(2), 254-265. Retrieved from http://www.jstor.org.proxy.mul.missouri.edu/stable/170697

Taek, J., Maleck, T., and William T. C. (1999). Pavement making material evaluation study in Michigan. *Institute of Transportation Engineers: ITE Journal* 69, 7.

Tan, G., Sun, J., and Hou, G. (2012). The time-dependent rural postman problem: polyhedral results. *Optimization Methods and Software* 28, 4: 855-870.

Vis, I. F. A., and Roodbergen, K. J. (2009). Scheduling of container storage and retrieval. *Operations Research*, 57(2), 456-467. Retrieved from www.jstor.org.proxy.mul.missouri.edu/stable/25614764

Weise, T., Podlich, A., and Gorldt, C. (2010). *Solving real-world vehicle routing problems with evolutionary algorithms*. In R. Chiong, and S. Dhakal (Eds.), Natural intelligence for scheduling, planning and packing problems: 29–53.

Zhang, G., Hummer, J., and Rasdorf, W. (2010). Impact of Bead Density on Paint Pavement Marking Retroreflectivity. *Journal of Transportation Engineering*, 136(8): 773–781.

APPENDIX A: LP OPTIMIZATION MODEL

In order to identify optimal solution for the slow-moving striping process, an LP optimization model was developed. This model minimizes the total traveled distance, including the striping operation, "deadhead" miles and the distance traveled to/from the overnight location. The model is going to stop the work of striping when the shift is almost over and select the closest overnight location. The next day the crew will pick up the striping equipment from the overnight location and model will suggest where to start the work (from which arc). It might be the end point of previous day or some other point that the model decides to be the best starting point in order to have the best route.

A.1 Multi-pass RPP Model

In order to force the LP model to solve faster it is preferable to run the RPP model with multiple passes first. The output (solution) from this model will be used in the final model as *βSolution(j)*.

The standard RPP was modified by adding a new parameter Pij that identifies number of passes required for completion of the striping operation for each arc ij.

Minimize $\sum_{\substack{(v_i,v_j)\in R \\ j>i}} c_{ij}(P_{ij}+x_{ij}) + \sum_{(v_i,v_j)\in A'\setminus R} c_{ij}(x_{ij})$

1) $$\sum_{\substack{(v_i,v_j)\in R \\ j>i}} (P_{ij}+x_{ij}) + \sum_{\substack{(v_i,v_j)\in R \\ j<i}} (P_{ij}+x_{ij}) + \sum_{\substack{(v_i,v_j)\in A'\setminus R \\ j>i}} (x_{ij}) + \sum_{\substack{(v_i,v_j)\in A'\setminus R \\ j<i}} (x_{ij}) = 2 * z_i \ (v_i \in V')$$

$$x_{ij} \geq 0 \ and \ integer \ ((v_i, v_j) \in A')$$

$$z_i \geq 0 \ and \ integer \ ((v_i, v_j) \in A')$$

2) $$\sum_{v_i\in S, v_j\in \bar{S}} x_{ij} \geq 1 \left(S = \bigcup_{k\in P} V_k, \ \bar{S} = \left(\bigcup_{k=1}^{P} V_k \setminus S \right), \ \cdot P \subset \{1, \dots, p\} \right)$$

Parameters:

- vi; vj – vertex (intersection between two roads or nodes on are arcs)
- (vi, vj) – arc between edge vi and vj
- R - number of arcs that has to be passed (painted)
- A - Set of all arcs in the system
- A'\R - number of arcs that don't need to be painted (passed)
- Pij – number of passes required for finishing the striping operation for arc (ij)
- Cij – length of the distance from node i to node j
- Xij – number of times edge (vi,vj)∈A'\R is replicated in the optimal RPP solutions (passed)
- P – number of connected components
- S – set of vertexes that have connection with each other

- OF=minimize the total travel distance (sum of the traveled distance during the striping + sum of traveled distance without painting (deadhead traveling))

Constraint1. Each node has to be passed an even number of times (for RPP we have to start at some point and come back the same point by passing all required edges with the required number of passes)

Constraint2. Ensure that in the optimal cycle all connected components will be linked together.

The GAMS software will be used for writing the code for this model.

The output is Xij parameter. *βSolution(j)*=Xij+Pij = number of passes required for the optimal route with satisfaction of all conditions.

A.2 Slow-moving Multi-pass Postman Problem with Overnighting

Assume we have *kappa* undirected road segments that could potentially be traveled.

Sets

- *J* is the set of all directed road segments
- *I* is a copy of set *J*
- *T* is the set of all schedule positions
- *T'* is a copy of set *T*
- *K is* the set of days required for striping
- *O* is the set of overnight locations
- *NODE* is the set of arcs in a network
- *NODE'* is a copy of set *NODE*

For set *J*, we create two copies of each segment (one for each direction), and we include a dummy segment *maintenance building* that corresponds to the initial starting point at the beginning of the schedule (which is also the final stopping point at the end of the schedule). We order the elements of set J such that:

- Positions *1, …, kappa*: each element corresponds to a different undirected road segment
- Positions *kappa+1* to *2kappa*: each element corresponds to the element *kappa* positions ahead of it in the set, with the direction reversed (that is, the elements in position *1* and *k+1* correspond to the same road segment in different directions, etc.)
- Positions *2kappa+1*: dummy segment *maintenance building*

Data Parameters

- $Head_j$ node at the start of directed arc *j*
- $Tail_j$ node at the end of directed arc *j* (for dummy segment $head_{maintenance\ building}=head_{maintenance\ building}$)

- $\alpha_{ij} = \begin{cases} 1 & \text{if } tail_i = head_j, \\ 0 & \text{otherwise} \end{cases}$
- $\theta_{ij} = \begin{cases} 1, & \text{if for the arc j and arc i (next day)} \\ & \text{the closest overnight location is } O, \\ 0 & otherwise \end{cases}$
- *βSolution(j)* required number of passes for each arc for optimal route (the result from RPP model with multiple of passes)
- γ_j distance across segment *j* (for dummy segment $\gamma_{maintenance\ building}$=0)
- *Speed(j)* speed to pass or stripe arc j
- λ_j time to pass segment *j* (stripe or just travel) = $\gamma_j/speed(j)$
- d_{Oj} distance from overnight location *O* to the head of arc *j*
- e_{jO} distance from tail of arc *j* to overnight location *O*
- Speedovl speed to travel from/to overnight location
- f_{Oj} time to travel from overnight location *O* to head of arc *j*=$d_{Oj}/Speedovl$
- h_{jO} time to travel from tail of arc *j* to overnight location *O*=$e_{Oj}/Speedovl$
- Φ_k the length of striping shift in day k (hours)

Decision variables

- Binary variable for free

$$p_{tjk} = \begin{cases} 1 & \text{if at time } t \text{ on day } k \text{ we start the day from overnight location } O \\ & \text{and drive till } head \text{ of segment } j \text{ to start striping,} \\ 0 & \text{otherwise} \end{cases}$$

- Binary variable

$$y_{tk} = \begin{cases} 1 & \text{if time } t \text{ occurs on day } k, \\ 0 & \text{otherwise} \end{cases}$$

- Binary variable for free

$$z_{tkj} = \begin{cases} 1 & \text{if at time } t \text{ on day } k \text{ we stripe (pass)} \\ & \text{the segment } j \text{ during the shift,} \\ 0 & \text{otherwise} \end{cases}$$

- Binary variable

$$s_{tj} = \begin{cases} 1 & \text{if we stripe (pass)segment } j \text{ at time } t \text{ during the shift,} \\ 0 & \text{otherwise} \end{cases}$$

- Binary variable for free

$$g_{tkj} = \begin{cases} 1 & \text{if at time } t \text{ on day } k \text{ we travel from tail of arc } j \\ & \text{to the closest overnight location } O, \\ 0 & \text{otherwise} \end{cases}$$

- Binary variable for free

$$q_t = \begin{cases} 1 & \text{if at time } t \text{ we travel from tail of arc } j \\ & \text{to the closest overnight location } O, \\ 0 & \text{otherwise} \end{cases}$$

Objective function

Min $\sum_t \sum_k \sum_j d_{Oj} p_{tkj} + \sum_t \sum_k \sum_j \gamma_j z_{tkj} + \sum_t \sum_k \sum_j e_{jO} g_{tkj}$ (1)

Constraints

$\sum_t (s_{tj} + s_{t,j+kappa}) \geq \beta Solution(j)$ (j≤kappa) (2)

$s_{tj} \leq \sum_i \alpha_{ij} s_{t-1,i} + \sum_{ki} \theta_{ij} g_{t-1,ki}$ (t>1) (3)

$\sum_t s_{tj} - \sum_t s_{t,j+kappa} \leq 1$ (j≤kappa) (4)

$\sum_t s_{t,j+kappa} - \sum_t s_{tj} \leq 1$ (j≤kappa) (5)

$\sum_t \sum_j f_{Oj}\, p_{(t)kj} + \sum_t \sum_j \lambda_j\, z_{(t)kj} + \sum_t \sum_j h_{jO}\, g_{(t)kj} \leq \phi_k$ (6)

$\sum_k y_{tk} = 1$ (7)

$y_{tk} + q_t - 1 \leq y_{t+1,k+1}$ (8)

$y_{tk} - q_t \leq y_{t+1,k}$ (9)

$y_{tk} + q_{t-1} + s_{tj} - 2 \leq p_{tkj}$ (t>1, k>1) (10)

$p_{tkj} \geq 0$ (11)

$\sum_{tj} p_{tkj} = 1$ (k>1) (12)

$\sum_j (s_{tj} + q_t) = 1$ (13)

$q_t - s_{t-1,j} \leq 0$ (t>1) (14)

$q_t \geq 0$ (15)

$\sum_t q_t = k$ (16)

$y_{tk} + s_{tj} - 1 \leq z_{tkj}$ (17)

$z_{tkj} \geq 0$ (18)

$y_{tk} + q_t + s_{t-1,j} - 2 \leq g_{tkj}$ (t>1) (19)

$g_{tkj} \geq 0$ (20)

$\sum_{tj} g_{tkj} = 1$ (21)

Objective function (1) minimizes the total traveled distance including driving to/from the overnight location at the end/start of each day, actual striping operation and "deadhead" miles traveled during the shift. Constraints can be divided into a few groups:

- Linking the day and time (7,8,9): Equation (7) makes sure that period of time t can happen only once in day k. Equation (8) shows if at time t on day k the job was stopped, then the next time period (t+1) will happen in day (k+1). If the day's work wasn't finished on time t, then next period (t+1) will be still happening on day k (9).

- Linking the striping process and traveling to the overnight location (at the end of day) (3,13,14,15,16): Constraint (3) helps to keep the connectivity of the network when sequence is chosen. The arc j that will be striped in period t has to be connected with arc i that was striped at period (t-1). Also, the arc j that will be striped at the beginning of the day (from overnight location) has to be close to the overnight location (arcs I and J have to be in one set of OVL). Constraints (13) and (14) prove that at time period t only one action should happen: whether striping or traveling to the overnight location at the end of the day. The model will

stop the work at time period t only if previously at time (t-1) the crew was working. It helps to prevent the model from stopping the work a few times per day. Constraint (16) makes sure that the work can be stopped only k number of times (k-how many days of work we have overall), it means only once in a day. According to equation (15) q(t) is a free positive variable. With help of constraints (13,14,16) it becomes "binary variable for free." It becomes 1 when the job has to be stopped and equipment should be parked in the closest overnight location, value of 0 when the crew is either striping or traveling. It helps to decrease size of the problem and computational time.

- Satisfaction of striping requirements (2,4,5): According to the constraint (2) number of total passes should be equal at least *βSolution(j)* times. *βSolution(j)* was calculated in the RPP model with multiple passes – number of passes required for the optimal sequence (including number of times required for striping p(j)). Constraints (4,5) help to deal with the direction for each arc in order to prevent striping against traffic (if the total number of passes is even, only 2 passes in each direction have to be made).

- Starting the day from overnight location (10,11,12): If the striping job was finished at time (t-1) and equipment was parked in the overnight location, at time period t on next day k, the work will be started from the facility, the crew will drive to segment j where striping will begin at period t on day k (10). Constraint (12) makes sure that the day k at time period t will be started from the overnight location only once. According to the equation (11) P(t,k,j) is a free positive variable. With constraints (10,12) it becomes a "binary variable for free." It's equal to 1 when the day k should be started from the overnight location, and when previously at (t-1) striping equipment was parked in that facility. It's equal to 0 when other actions happen at the same time period t (just striping).

- Striping plan for a day (17,18): Constraint (17) proves that striping of arc j will happen at time t on day k. According to the equation (18) z(t,k,j) is a free positive variable with adding constraint (17) it becomes a "binary variable for free" (0 or 1).

- When to end the working day (19,20,21): If the day should end at time period t, striping of arc j was happening on time (t-1) and time period happens on day k then the day k will be over at time period t at the end point of arc j (19). Constraint (21) proves that the day k will be ended by the model only once at time period t at the end node of arc j. According to constraint (20) g(t,k,j) is a free positive variable. By adding (19) and (21) equations it becomes a "binary variable for free." It equals to 1 only when the striping crew drives to the facility at the end of day at period t, period happens on day k, striping of arc j ended at time (t-1).

- Control of the length of the shift (6): This constraint makes sure that the time for striping and time for traveling from/to overnight location doesn't exceed the length of the shift on day k.