SENSOR LOCATION FOR NETWORK FLOW AND ORIGIN-DESTINATION ESTIMATION WITH MULTIPLE VEHICLE CLASSES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Qing Zhao

August 2015

SENSOR LOCATION FOR NETWORK FLOW AND ORIGIN-DESTINATION ESTIMATION WITH MULTIPLE VEHICLE CLASSES

Qing Zhao, Ph. D.

Cornell University 2015

The need for multi-class origin-destination (O-D) estimation and link volume estimation requires multi-class observations from sensors. This dissertation has established a new sensor location model that includes: 1) multiple vehicle classes; 2) a variety of data types from different types of sensors; and 3) a focus on both link-based and O-D based flow estimation. The model seeks a solution that maximizes the overall information content from sensors, subject to a budget constraint. An efficient two-phase metaheuristic algorithm is developed to solve the problem.

The model is based on a set of linear equations that connect O-D flows, link flows and sensor observations. Concepts from Kalman filtering are used to define the information content from a set of sensors as the trace of the posterior covariance matrix of flow estimates, and to create a linear update mechanism for the precision matrix as new sensors are added or deleted from the solution set. Sensor location decisions are nonlinearly related to information content because the precision matrix must be inverted to construct the covariance matrix which is the basis for measuring information. The resulting model is a nonlinear knapsack problem.

The two-phase search algorithm proposed addresses this nonlinear, nonseparable integer sensor location problem. A greedy phase generates an initial solution, feeding into a Tabu Search phase which swaps sensors along the budget

constraint. The neighbor generation in Tabu search is a combination of a fixed swap-out strategy with a guided random swap-in strategy.

Extensive computational experiments have been performed on a standard test network. These tests verify the effectiveness of the problem formulation and solution algorithm. A case study on Rockland County, NY demonstrates that the sensor location method developed in this dissertation can successfully allocate sensors in realistic networks, and thus has significant practical value.

BIOGRAPHICAL SKETCH

Qing Zhao was born in 1989 in Xingjiang, China, and moved with the whole family to Shanghai when she was 3, where she finished her primary and secondary education and later earned undergraduate degree in Transportation Engineering at Tongji University. With a great love for her major, she continued her education at Cornell University as a MS-Ph.D student and earned her Master of Science degree in 2013 and her Doctor of Philosophy degree in 2015 under the supervision of Professor Mark A. Turnquist.

致我的父亲: 赵兴斌

母亲: 田静

For my father, Xingbin Zhao

and my mother, Jing Tian

ACKNOWLEDGMENTS

I wish to express here my indebtedness and gratitude to those who have helped me to bring this dissertation to fruition.

I am enormously grateful to my committee chair Dr. Mark Turnquist, who continuously provides intellectual and methodological guidance, as well as moral support even when his own health condition is concerning. His far-reaching vision and deep insights helped me to form the idea from the very beginning and steer my project aright. His incredible patience and rich knowledge helped me to solve the problems and make progress along the way. His high standard and detailed revision helped to bring the dissertation to an end. His mentorship is paramount in providing a well-rounded experience consistent my long-term career goals. I am extremely grateful for being the very fortunate recipient of much time and attention. Dr. Turnquist has been a paragon to me. His perseverance and dedication for perfection encouraged me all through my 5 years at Cornell and will stay with me as I leave.

I wish to offer my heartfelt thanks to my other committee members, Dr. Oliver Gao and Dr. Huseyin Topaloglu. The academic guidance and suggestions to the dissertation from them are invaluable to this process.

I also wish to express my sincere thanks to Dr. Peter I. Frazier and Dr Charles Van Loan in particular for giving me suggestions and inspirations on algorithms. The discussions were very insightful and enjoyable.

I also wish to express my special thanks to Mrs. Turnquist. She has been a great help for supporting Professor Turnquist both in work and in life. Without her

support, this dissertation cannot come to fruition. Her optimism and love make all this possible.

To my relatives and friends, thanks for the abundance of love and always being a great source of inspiration and support.

Finally and most importantly, I would like to thank my mom for her support, encouragement, and unwavering love. I would also like to thank my dad for having faith in me and allowing me to grow. It was under their watchful eye that I gained so much drive and an ability to tackle challenges head on.

My feelings, looking back, is of immense gratitude to all these people.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

AVI	Automatic Vehicle Identification
agg	aggregated
cls.	classified
DOT	Department of Transportation
GPS	Global Positioning System
GSP	Garden State Parkway
hr.	hour
NYS	New York State
O-D	Origin-Destination
PIP	Palisades Interstate Parkway
RFID	Radio-frequency identification
SF	Sioux Falls
SNL	Stochastic Network Loading
SUE	Stochastic User Equilibrium
TAZ	Traffic Analysis Zone
TS	Tabu Search
UMZ	Unit Marginal Z
veh.	vehicle
VMT	Vehicle Miles Traveled
vol.	volume

CHAPTER 1

INTRODUCTION

Effective traffic management in networks depends on the ability to sense and interpret volumes and flow patterns of vehicles using the network. Existing practice is based primarily on use of loop counters embedded in the roadway as a means of obtaining counts and average speed data for a small fraction of network links. The data from these relatively crude sensors has at least five major limitations: 1) the type of data available is very limited; 2) there is very little ability to distinguish vehicles in different size classes; 3) the sensors tend to produce the least reliable information under the most critical high-volume conditions; 4) there are too few sensors and their deployment is often based on need for very local data rather than being designed to provide useful network-wide data; and 5) there is limited ability to synthesize the data from a collection of sensors into a coherent picture of what is happening on the network as the basis for effective traffic management.

The expectations for effectiveness of traffic management are rising for several reasons. First, traffic demand is growing faster than available physical capacity, creating increasing levels of congestion. Second, there is increasing concern with truck movements in urban networks because trucks impose different levels of pavement damage than automobiles, they have different emission characteristics, different accident patterns, and may be subject to different flow controls. Third, there is increasing interest in pricing policies for use of the network, with prices that may vary by location, time-of-day and vehicle class. And fourth, we may be on the verge of a

fundamental change in how the roadway network is funded, changing from a fuel-tax based system to a system based on vehicle-miles-traveled (VMT).

To meet these rising expectations, traffic management must be based on more and better information about network flow patterns. Modern traffic sensing technology offers increasing ability to classify vehicles as they are counted, as well as to create data that are more informative than simple link counts, including output from video detectors, GPS-based vehicle location systems, automatic vehicle identification (AVI) systems, etc. As sensing technology advances, applications in traffic management are growing and many important questions about sensor deployment and data use are becoming more critical: What type of sensors should be used? How many of them are needed and where should they be installed in order to get required information economically? How can traffic system managers synthesize the data from different types of sensors, located in various parts of the network, to create network-level information?

The network information of interest is partly link-based, partly path-based, and partly origin-destination (O-D) based. Good estimates of link volumes (including links that may not be observed directly) is important for evaluating speeds and travel times, total emissions, VMT, etc. Estimating O-D flows provides a more complete picture of demand on the network and the basis for evaluating possible responses to traffic management strategies. Path-based information provides a connection between link-based data and O-D based demand, and is also important for traffic management assessment.

In a recent review of sensor location issues, Gentili and Mirchandani (2012)

identify two general classes of objectives for sensor location models: flow observability and flow estimation. Flow observability focuses on a specific set of flows in the network (which may include link flows, O-D flows, path flows, etc.) and seeks the minimum set of sensors and locations required to solve uniquely for the desired flows from the sensor observations. Flow estimation, on the other hand, focuses on situations where sensor observations cannot uniquely determine flows, but seeks sensor locations that can best improve the quality of flow estimates available from the sensor observations. The work in this thesis is in the latter category and focuses on locating sensors to improve estimates of both link flows and O-D flows in a network.

The purpose of this dissertation is to accomplish five important goals:

1) Consider data collected on multiple vehicle classes;
2) Consider a variety of data types (from different types of sensors) that include not just link-based data, but also turning movements and other partial path data;
3) Consider an objective that may include both link-based and O-D based flow estimation;
4) Formulate the optimization problem for choosing sensor types and locations to maximize the overall information content of the resulting data, subject to budget constraints on available sensor deployment; and
5) Develop an efficient solution algorithm for the problem so that it becomes a practical tool for traffic management and planning.

To accomplish these goals, the research builds on several previous efforts in

the rapidly growing literature on network sensor location. Particular attention is paid to constructing a model of the information content from different types of sensors and reflecting the interdependence of information from multiple sensor locations. We also pay special attention to the desire to sense multiple classes of vehicles and differentiate automobile flows from truck flows, for example.

The model formulation that includes multiple vehicle classes and integrates data from a wide variety of sensor types is one major contribution of this research. Another important element is the design of a solution algorithm for the problem. This solution algorithm:

1) Addresses the nonlinear, nonseparable discrete character of the objective function in the optimization using a meta-heuristic search method;
2) Is capable of starting from an arbitrary point, representing existence of some prior set of sensors already in place, and find the optimal (or near-optimal) way to augment them within an available budget; and
3) Is efficient enough to scale in a reasonable way to networks of realistic size.

Chapter 2 provides background and a literature review on sensors and sensor location modeling. Chapter 3 describes the problem formulation in detail and Chapter 4 discusses the literature on algorithm design and developed a solution algorithm for the problem. Chapter 5 describes a battery of computational experiments on a test network and Chapter 6 illustrates application of the model in a realistic network (Rockland County, New York). Chapter 7 presents conclusions and outlines directions for further research.

CHAPTER 2

BACKGROUND AND LITERATURE REVIEW

In most urban areas, the primary source of traffic flow data is single-loop counters buried in the pavement on selected links or near major intersections. The data from these counters are generally total vehicles and average speed over relatively short (e.g., 10-15 minute) periods and the intended use of these data is often quite local (traffic signal setting, etc.). Single-loop counters do not typically provide any vehicle classification information, but in some places dual-loop installations have been used. These are capable of limited vehicle classification counts based on vehicle length, but are highly unreliable in congested stop-and-go conditions.

The available technology for traffic flow sensing has improved quite significantly in recent years, along with rising expectations for improved network-level traffic management. The combination of more capable sensors and higher expectations for using the data has motivated increasing interest in the problem of sensor location. How many of what types of sensors should be deployed, and in what locations across the network, to provide the real-time information necessary for more effective traffic management and control? Various aspects of this general question have been addressed by researchers since the late 1990's.

Yang and Zhou (1998) and Yim and Lam (1998) focused on identifying link count locations for estimating O-D matrices. Bianco, *et al*. (2001) also focused on O-D estimation using link counters, but added assumed turning probabilities at intersections to formulate a problem of identifying a subset of links on which counters

should be placed.

Gentili and Mirchandani (2005) addressed the sensor location problem as a set-covering problem using active sensors. The intended use of the model is for monitoring or managing particular classes of identified traffic streams and only one special case is shown to be polynomially solvable. Sherali, *et al*. (2006) formulated a quadratic 0-1 optimization problem for locating Automatic Vehicle Identification (AVI) tag readers by maximizing the benefit that would accrue from measuring travel times. Eisenman, *et al*. (2006) used a simulation-based real-time network traffic estimation and prediction system based on dynamic traffic assignment (DTA) methodology to analyze different levels of detection and different sensor locations in a portion of the Chesapeake Highway Advisories Routing Traffic (CHART) network.

Zhou and List (2010) extended the work of Eisenman, *et al*. (2006) and proposed an information-theoretic model to maximize the expected information gain from a set of link and point-to-point sensors, subject to budget constraints. A heuristic beam search algorithm was used to solve the problem. Fei and Mahmassani (2011) proposed a Kalman-filtering-based bi-objective model, which considers link information gains and O-D demand coverage in the context of dynamic traffic assignment, to locate a minimal number of passive point sensors given an available budget.

The work by Wang, *et al*. (2012) focused on locating link count sensors to estimate route flows. They assumed that a (relatively small) set of possible routes is enumerated for each O-D pair (defined as the route choice set), and that prior estimates of flows on all these routes are available, along with confidence limits on

those estimates. They then proposed a two-part model, one of which constructs route flow estimates from observed link counts, and the other which selects link sensor locations to maximize reduction in uncertainty of the resulting route flows. Their focus on flows for an enumerated route choice set represents a level of analysis that is different from either Zhou and List (2010) or Fei and Mahmassani (2011), but the core ideas that sensors produce observations that can be used to update prior estimates, and that the sensor location problem is one of maximizing some information measure from the sensors, are common among all three efforts.

These three previous efforts provide important building blocks for the extended analysis in the current research. In the research proposed here, the information-theoretic model designed by Zhou and List (2010) is extended to include multiple vehicle classes and a broader range of potential sensors and data types. The extension uses some of the ideas of a route choice set propsed by Wang, *et al*. (2012). Elements of the formulation by Fei and Mahmassani (2011) that include a focus on both O-D volumes and link volumes are also included. Fei and Mahmassani (2011) developed an approximation for the information gain with respect to O-D pair *w* from placing a sensor on link *a*. By treating this value as a constant for each *w*-*a* combination (independent of what other sensors are chosen), they were able to express the sensor location model as a linear knapsack problem, which greatly simplifies the optimization. Wang, *et al*. (2012) also constructed a linear measure of information gain from individual sensors, creating an even simpler form of linear knapsack problem for locating sensors. However, as shown by Zhou and List (2010), the actual information content of a collection of sensors is not additive. The basic approaches

taken by Fei and Mahmassani (2011) and Wang, *et al*. (2012) are useful, but the actual objective (maximizing information) is nonlinear. Thus, the resulting overall optimization has the form of a nonlinear, nonseparable discrete knapsack problem. Incorporating this nonlinearity is an important part of the work described in this thesis.

Collection and processing of video data presents particularly interesting opportunities and challenges. Surveillance cameras have versatile uses (e.g. crime detection and prevention, license plate tracking, etc). With the development of image processing technology, cameras can also distinguish vehicles in different classes effectively and provide data on turning movements at intersections. In addition, multiple cameras at different locations may be able to provide path information for selected individual vehicles. These data can be used to improve understanding and management of network traffic flows. However, installing and maintaining video data collection systems is more expensive than deploying traditional loop counters, and processing the video data to extract useful traffic information is also expensive (although becoming less so). The work in this thesis places significant emphasis on integrating observations from several different types of sensors (intersection-based as well as link-based) and on allocating limited resources most effectively to locate sensors that can yield information of varying levels of accuracy on multiple vehicle classes at varying costs. This represents a significant generalization of previous work.

Gentili and Mirchandani (2012) provide an extensive review of work on sensor location models and a classification structure for different types of modeling efforts. It is useful to place the work presented here within that structure. They distinguish between what they term the “Sensor Location Flow-Observability Problem” and the

"Sensor Location Flow-Estimation Problem". The former deals with the problem of determining how to locate sufficient sensors to guarantee that a specified set of network flows can be determined uniquely. The latter addresses the problem of identifying some set of sensor locations on the network to optimize a specified evaluation criterion, subject to a budget constraint. The work presented here is of the second type.

Within the general category of sensor location flow-estimation models, Gentili and Mirchandani define ten different types of optimization models for sensor location, based on choosing different evaluation criteria, or "rules" (in their terminology). The closest model type to the work presented here is defined as Model M10: Locate a given number *k* of sensors to minimize the estimation error in the O-D flow estimates. The model developed in this research does not assume a fixed number of sensors to be located, although it is assumed that there is a budget constraint and that sensors of different types and locations have varying (known) costs, so that the selection of types and locations is constrained. Also, the model formulated here does not concentrate exclusively on O-D flow estimates, but also includes link flow estimates.

Recent work on models in the M10 category includes Zhou and List (2010), Fei and Mahmassani (2011) and Wang, *et al*. (2012) mentioned above, and also papers by Li and Ouyang (2011), Fei, *et al*. (2013), Lu, *et al*. (2013), Sayyady, *et al*. (2013) and Hu and Liou (2014).

Li and Ouyang (2011) developed a compact linear integer programming model to determine optimal locations for vehicle ID sensors for travel time estimation along known paths, with the assumption that sensors may fail, causing loss of coverage for

some path flows. Although the setting and objectives were somewhat different from the work presented here, their work is interesting because of the recognition of sensor errors and potential failure.

Fei, *et al*. (2013) extended the model of Fei and Mahmassani (2011) to include the effects of random traffic incidents that may cause flow changes in a dynamic network. They formulated a two-stage stochastic optimization model to locate a fixed number of link sensors under uncertain future flow conditions. All the core elements of the model developed by Fei and Mahmassani (2011) are retained, but the set of coefficients used to evaluate the marginal information benefit of a sensor placed on link *a* is expanded to allow computation of an expected value across multiple traffic flow condition scenarios. This is a useful and ambitious piece of work, but moves in a different direction from what is undertaken in this thesis. The focus here is on better representation of the information content of a collection of various types of sensors for estimating flows of multiple vehicle classes, rather than representing the effects of flow disruptions resulting from traffic incidents.

Lu, *et al*. (2013) focused on camera location for traffic surveillance and enforcement functions (speeding, red light infractions, etc.). They included multiple user classes, although their definition of classes was not by vehicle type, but classes of drivers distinguished by their degree of concern to avoid intersections with cameras. The objective of the model formulated was to maximize the observed flow on monitored links, given that drivers may change routes to avoid monitoring. The underlying premise of their model was different from the flow estimation problem of interest here, but the work does represent one of the few instances of multiple user

classes in the sensor location literature.

Sayyady, *et al*. (2013) were concerned with locating weigh-in-motion sensors to estimate axle loads on highway links by choosing a subset of p links that are most representative of a larger set. They used a p-median formulation to select optimal locations. This work is of interest because it focused on truck flows in multiple classes, but the character of the flow estimation problem is considerably different from what is of interest here.

Hu and Liou (2014) considered a problem of deploying what they term passive sensors (link flow counters) and active sensors (license plate readers) to capture a set of link volumes (from the passive sensors) and path or route volumes (from the active sensors) that would best support estimating an O-D matrix for a network. Their work is interesting both because it considered integration of multiple sensor types and observations and because the method takes advantage of explicitly constructed variables to determine link or path flows that can be inferred from flow conservation if other flows are observed directly. This approach reduces reliance on link/path/O-D incidence matrices for connecting observations from the sensors to the unknown flows to be estimated and represents a novel way of trying to represent the information content of a collection of sensors. However, the method does not incorporate errors in the sensor observations and how such errors may degrade the quality of information being obtained.

Against the backdrop of the previous work described above, the following chapter creates a new formulation of the sensor location flow-estimation problem that:

1) Considers data collected on multiple vehicle classes;

2) Considers a variety of data types (from different types of sensors) that include not just link-based data, but also turning movements and other path-based data;
3) Includes estimation of both link flows and O-D flows; and
4) Chooses sensor types and locations to maximize the overall information content of the resulting data, subject to budget constraints on available sensor deployment.

CHAPTER 3

A SENSOR LOCATION MODEL

3.1 Introduction

The purpose of the model developed in this chapter is to choose sensor types and locations to maximize the overall information content of the resulting data, subject to budget constraints on available sensor deployment. At the center of the model is a set of linear equations that relate O-D flows, link flows and sensor observations. The sensor information content is defined using concepts from Kalman filtering, resulting in a set of nonlinear equations that relate choices of sensor types and locations to total information gain. The final optimization is then a nonlinear knapsack model in which the sensors are chosen subject to a budget constraint. The following sections describe the elements of this model and how these elements are integrated to create the overall problem formulation.

3.2 Relating Flows and Observations

If zones are defined as the origin and destination points in a network, the O-D flows are often summarized as a square matrix (i.e., from-to), but it is also possible to express the relevant entries (ignoring the intrazonal trips on the diagonal, as well as any other combinations that are known or assumed to be zero) as a column vector, Q, and we will adopt that convention. The length of this vector will be denoted N, and individual entries will be indexed by n. If the network contains L directed links, the vector of link volumes, V, can be expressed as:

$$V = PQ \qquad \textbf{Equation 3- 1}$$

where P is an L x N link utilization matrix whose entries give the proportion of trips for O-D pair n that use link l, l = 1,…, L.

For the model discussed here, the values of P are assumed known, creating a linear relationship between Q and V. Q is treated as unknown and is to be estimated by a constructed vector, Q. This estimate is assumed to have a covariance matrix S_Q. The link volume estimate: $V = PQ$, has a covariance matrix, S_V, that can be related to S_Q through Equation 3-2 (Fei and Mahmassani, 2011):

$$S_V = PS_QP^T \qquad \textbf{Equation 3- 2}$$

The matrix *P* can be determined through some mechanism of identifying paths (either implicitly or explicitly) between each origin and destination in *Q*, and assigning relative likelihoods for traffic using each of those paths. This can be done in several possible ways, including both deterministic and stochastic traffic assignment models, reflecting various sets of assumptions about how traffic flows are distributed on the network. The sensor location model can work with *P* matrices developed in many different ways, but the end results (sensor types and locations) will be somewhat sensitive to the assumptions underlying whatever *P* matrix is used.

In the computational experiments described later in this thesis, *P* has been determined using a process that uses link lengths and free-flow travel times to construct *efficient* links (i.e., links that move further from the origin and closer to the destination, both in terms of distance and time) for each O-D pair. Based on the set of efficient links for each O-D pair, paths can be identified explicitly and path choice

probabilities are constructed using a probit model. The link utilization probabilities in *P* can then be constructed by aggregating the path choice probabilities for all paths that include a given link. Because path impedance (as a function of distance and travel time) for automobiles and trucks is different, the values in *P* for the same link and O-D pair, but different vehicle classes, will differ, but these values are in different rows of *P*, corresponding to link flow for the separate vehicle classes.

This process for constructing the matrix *P* is consistent with an assumption that the process of locating sensors in the network should not depend on detailed observations of network flows. That is, the sensors must be located *before* the observations are available. If the actual flows in the network are assumed to represent either deterministic or stochastic user equilibrium, then in general, the elements of *P* are functions of *Q*. In this case, Equation 3-1 is a set of nonlinear equations and the optimal set of sensor locations *based on prior knowledge of the flows* might not be the same as the set constructed without that knowledge. However, the model developed here proceeds under the assumption that the *P* matrix must be computed before detailed flows are available.

A collection of sensors in the network produces a vector of observations, *Y*. These observations can be related to link volumes and O-D flows by Equation 3-3.

$$Y = A_V V + A_Q Q + \varepsilon_Y \qquad \textbf{Equation 3- 3}$$

where A_Q and A_V are coefficient matrices and ε_Y is a column vector of observation errors.

Using Equation 3-1, we can reduce Equation 3-3 to

$$Y = HQ + \varepsilon_Y \qquad \textbf{Equation 3- 4}$$

where H is a coefficient matrix that contains characteristics of P, A_Q, and A_V. Each column of H corresponds to an O-D pair. Rows or collections of rows in H are associated with the observations from a particular sensor, and we will denote the total number of rows in H as G. The problem of choosing sensor types and locations can be defined as choosing a particular set of rows to make up the matrix H, subject to a budget constraint.

3.3 A Measure of Information from the Sensors

Different choices of rows to construct H can be evaluated using a measure of information contained in the sensor observations. If a classic Kalman filter is used to integrate new sensor information into an estimate of Q, the objective of minimizing mean-square error is equivalent to minimizing the trace of the posterior covariance matrix (Zhou and List, 2010). Each vector, Q, can be decomposed into its mean, $\overline{Q}$, and deviation component, d_Q. $d_Q = Q - \overline{Q} \bullet 1$, where 1 is a vector of 1. The length of deviation vector, L_{d_Q} can be calculated by $L_{d_Q}^2 = d_Q^T d_Q = \sum_{j=1}^{n} (Q_{ij} - \overline{Q}_i)^2$, where index i represents the element of vector, j represents the repetition. The information comprising S_Q is obtained from the deviation vector d_Q and trace of variance covariance matrix equals to the sum of squared deviations(Johnson and Wichern, 2007).

If there is some prior estimate of the O-D matrix, $\bar{Q}$, with covariance matrix

S_Q^-, and a set of sensors produce observations *Y* as given by equation 3-4, with covariance matrix *R*, the Kalman filter is a linear updating mechanism to create a posterior estimate of *Q*, denoted Q^+. The update is of the form:

$$Q^+ = Q^- + \Phi\left(Y - HQ^-\right) \qquad \textbf{Equation 3- 5}$$

The term $\left(Y - HQ^-\right)$ is the error between what is observed and what was expected, based on the prior estimate, Q^-. The matrix Φ is a gain matrix, and if the entries in that matrix are selected to minimize mean square errors in the posterior estimate, Q^+, the result is:

$$\Phi = S_Q^- H^T (HS_Q^- H^T + R)^{-1} \qquad \textbf{Equation 3- 6}$$

The corresponding update to the covariance matrix S_Q is

$$S_Q^+ = S_Q^- - \Phi H S_Q^- = \left(I - \Phi H\right) S_Q^- = \left[\left(S_Q^-\right)^{-1} + H^T R^{-1} H\right]^{-1} \qquad \textbf{Equation 3- 7}$$

As a special case, if no prior covariance matrix on *Q* is available, the prior precision matrix, $\left(S_Q^-\right)^{-1}$ is zero, and Equation 3-7 simplifies to

$$S_Q^+ = (H^T R^{-1} H)^{-1} \qquad \textbf{Equation 3- 8}$$

The information content associated with a set of sensor choices (types and locations) is the reduction in $tr\left(S_Q^+\right)$ achieved by deploying those sensors. Because the covariance matrix update does not require the observations *Y* themselves, but only specification of *H* and *R*, the problem of choosing sensor locations to maximize information content can be addressed separately from the actual update of the

estimated O-D matrix.

It is also useful to compare the idea of information content from a set of sensors (and maximization of that information as the objective of a sensor location model) with several previous efforts to create objectives for sensor location. The survey paper by Gentili and Mirchandani (2012) summarized seven location evaluation criteria ("rules") proposed in previous papers on sensor location flow estimation. It should be noted that each of these rules was developed for a situation where only link counters are considered and there is a single vehicle class, so none of them focus on separating vehicles by class nor on sensors located at intersections. However, considering them as possible criteria (as contrasted with maximizing information content from the sensors) provides additional insight into the model proposed here.

1) **Rule 1:** O-D Covering Rule (Yang *et al*., 1991). Maximize the total number of O-D pairs *covered* by the solution set. An O-D pair is covered if some route connecting the O-D pair includes a sensor, so that some fraction of trips for that O-D pair is observed. This is a simple criterion and measurement of attainment is easy. It also avoids need for any prior estimates of O-D volumes. However, as a result no distinction is made between important large-volume O-D pairs and less important small-volume pairs. By focusing on reducing variance, the information criterion places emphasis on O-D pairs for which the estimated volumes are most uncertain, and these are generally high-volume pairs.

2) **Rule 2:** Maximal Flow Fraction Rule (Yang and Zhou, 1998). Counting sensors should be located on links so that, for each covered O-D pair, the flow fraction for this O-D pair on measured links is as large as possible. The idea behind this rule is to choose links for counting where the link flow is dominated by one (or a few) O-D pairs, so that the counted values can be used to estimate O-D volumes with greater confidence. However, to use this rule in practice, it is necessary to have prior information on link flows, so in general it is not a very useful criterion.

3) **Rule 3:** Route Covering Rule or O-D Separation Rule (Yang *et al*., 2006; Ma *et al*., 2006). Counting sensors should be located on links so that the total number of routes covered by the location set or the total number of O-Ds separated by the location set is maximized. This is an amplification of Rule 1, since the objective is to not only observe *some* route connecting each O-D pair, but to observe *all* routes. An O-D pair is *separated* by the sensor location set if the sensors cover all routes connecting that O-D pair. Implementation of this rule requires explicit enumeration of all routes connecting each O-D pair, so it is difficult to use in a large network. This rule is also somewhat at odds with information maximization, because locating a sensor to count traffic on a low volume link that is part of a route connecting a minor O-D pair (in order to separate that O-D pair) may yield much less information than a sensor on a higher volume link used by several other O-D pairs.

4) **Rule 4:** Maximal Flow Intercepting Rule (Yang and Zhou, 1998). Counting sensors should be located on links so that the total observed flow is as large as possible. This rule focuses on link flows rather than O-D volumes. If some prior information on likely link flows is available, the sensor location solution is a simple "greedy" process of ranking the links by volume and selecting the largest until the available budget is exhausted. Although easy to implement computationally, this rule requires prior information on the link flows that the sensors are designed to collect, which is problematic, and it does not recognize that many high volume links are likely to be in sequence along a corridor and selecting all of them for sensing counts the same traffic multiple times. This may not yield much new information. The maximization of information from the collection of sensors will generally produce a much more effective solution.

5) **Rule 5:** Maximal O-D Demand Fraction Rule (Cipriani *et al*., 2006). Counting sensors should be located on links that maximize the sum of intercepted O-D demand fractions. This rule is a variation of Rule 2, but focuses on the total fraction of demand measured for each O-D pair covered. The idea behind this rule is that the estimated link utilization coefficients in the matrix P are likely to contain errors, and it may be difficult to eventually estimate O-D volumes from having observed only a small fraction of the volume (a small coefficient in P). Thus, this rule focuses on choosing links with large p_{an} values (i.e., the

fraction of trips for O-D pair *n* that use link *a*). However, focusing only on the p_{an} values without considering the volume on O-D pair *n* can lead this rule to locate a sensor to capture nearly all flow for a minor O-D pair (because the p_{an} value is near 1), but yield much less total information than a sensor on a link used by several high volume O-D pairs (but with smaller p_{an} values).

6) **Rule 6:** Maximal Net Route Flow Captured Rule (Yang and Zhou, 1998). Counting sensors should be located on links so that, for a given number of counting sensors, the largest net O-D route flows are measured. The idea of *net* O-D route flows is to eliminate multiple counting of the same route flows on different links. This is a useful conceptual generalization of Rule 4, and shifts the focus from link flows to route flows, but it is difficult to implement in large networks because it requires route enumeration in order to determine overlaps and compute net flow observed. The concept of information maximization also accomplishes the goal of discounting multiple observations of the same flows, but does so without requiring route enumeration.

7) **Rule 7:** Maximal Net O-D Flow Captured Rule (Yim and Lam, 1998). Counting sensors should be located on links so that, for a given number of counting sensors, the largest number of net O-D trips is measured. This is a variation of Rule 6, but focuses directly on O-D volumes rather than route volumes. In concept, this makes the rule easier to implement than Rule 6, but

in practice the computation of net flows observed is still difficult without route enumeration.

Using a direct measure of information from the set of sensor locations offers an opportunity for an objective that is more comprehensive than these previous criteria, and thus sets the stage for a formal definition of the problem of sensor location as viewed in this thesis.

3.4 **Problem Formulation**

Our overall objective includes the estimation of link volumes across the entire network, as well as the O-D matrix, so we are also concerned with $tr\left(S_V^+\right)$, where Equation 3-2 is used to construct S_V^+ from S_Q^+. The objective is then to choose a collection of rows of *H* to solve:

$$MinZ = \lambda tr(S_V^+) + (1-\lambda) tr(S_Q^+) \qquad \textbf{Equation 3- 9}$$

$$= \lambda tr(PS_Q^+P^T) + (1-\lambda) tr(S_Q^+)$$

where λ is a weighting parameter ($0 \le \lambda \le 1$), so that the sensors can be selected considering the accuracy of both link volumes and O-D volumes.

When multiple vehicle classes, *m*, are included, the O-D vector *Q* contains blocks of elements for each vehicle class and a row of *H* (corresponding to an observation from a sensor) may contain non-zero elements pertaining to a single vehicle class or to multiple classes, depending on the nature of the sensor.

Consider K different types (technologies) for sensors, indexed $k = 1, \ldots, K$. For sensor type k, assume there are D_k different potential locations in the network that are feasible. The total number of possible type-location possibilities is then $D = \sum_{k=1}^{K} D_k$. We will index these possible choices by $d = 1, \ldots, D$. If the observations from different sensors are independent, R is a block-diagonal matrix. A sensor that produces multiple measurements simultaneously may have correlated errors among those measurements, but those will be independent of blocks corresponding to any other sensor. We can define R_d as the block within R that corresponds to sensor d. The update to the precision matrix can then be written as:

$$\left(S_Q^+\right)^{-1} = \left(S_Q^-\right)^{-1} + \sum_{d=1}^{D} \left(h_d\right)^T R_d^{-1} h_d \qquad \textbf{Equation 3- 10}$$

where h_d is the collection of rows in H that pertains to measurements at sensor d. That is, the effects of individual sensors on the posterior precision matrix can be decomposed into separate matrices $\left(h_d\right)^T R_d^{-1} h_d$ and the net effect of a collection of sensors can be written as the sum of the individual matrices.

This does not mean that the effect of individual sensors on the trace of the covariance matrix is separable. The precision matrix must be inverted before the trace computation, and this complicates the evaluation of the objective function for the sensor location problem (Equation 3-9). However, the separability in the computation of the precision matrix is still useful.

For each d, there is a matrix Δ_d that expresses its effect on the posterior

precision matrix $\left(S_Q^+\right)^{-1}$. We can also create a decision vector *X* with integer elements x_d, that indicate the number of sensor possibility *d* has been implemented. Then the posterior precision matrix can be represented as:

$$\left(S_Q^+\right)^{-1} = \left(S_Q^-\right)^{-1} + \sum_{d=1}^{D} x_d \Delta_d \qquad \textbf{Equation 3- 11}$$

Combining Equation 3-10 and 3-11, we see that *Z* is a nonlinear function of *X*, and the sensor location problem can be expressed as:

$$\textbf{Min} \quad Z(X) \qquad \textbf{Equation 3- 12}$$

$$\textbf{s.t.} \quad \sum_{d=1}^{D} c_d x_d \le B \qquad \textbf{Equation 3- 13}$$

$$x_d \in \{0,1\} \qquad \textbf{Equation 3- 14}$$

where c_d is a cost associated with deploying sensor option *d*, and *B* is an overall budget limit.

This is a nonlinear knapsack problem. The variables are discrete and the objective function is nonseparable, which makes it a difficult problem to solve. Further discussion of potential approaches to solution can be facilitated by an example, so we'll consider a small network described in the following section.

3.5 An Example Network

Figure 3- 1 shows a small network with nine nodes and 12 two-way links. The values alongside the links are lengths, in miles. For analysis purposes, each of the two-way links is replaced by a pair of directional links, so the network has 24 directed links. We will assume that there are four O-D pairs for each vehicle class in this

network: 1-6, 1-9, 4-3 and 4-9. Three vehicle classes are assumed to be of interest – automobiles and two truck classes. The first truck class is two-axle, six-tire medium trucks. The second class includes all heavier trucks. Light trucks are included with the automobiles.

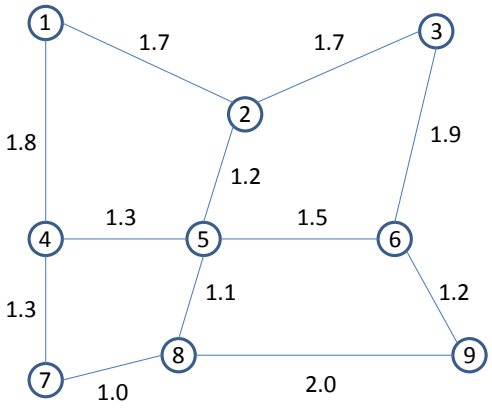


Figure 3- 1: Example network.

Four types of sensors will be considered for this example:

1) point sensor (i.e., link volume counter) with vehicle classification capability;
2) point sensor with only aggregate volume capability;
3) intersection surveillance camera providing through movements and turning movements, but without vehicle classification information; and
4) Automatic Vehicle Identification (AVI) sensors on a pair of links, providing counts of vehicles (by vehicle class) that appear in both places within a specified time interval.

These sensors provide data of differing types and with different levels of detail. They

are also assumed to have different costs. Table 3- 1 provides details on the possible locations, nature of observations available, and costs for each potential sensor. The problem is to select sensor types and locations in the network so that the variance of the resulting link volume estimates and O-D estimates are minimized while keeping the total expense within the available budget. For the example, it is assumed that the total available sensor budget is 8 units, implying that between two and four of the possible sensors can be chosen.

Table 3- 1: Potential locations, observations and costs for sensors

Index, d	Type	Location	Cost	Observations
1	Point sensor classified	Link 4-5	3	Link count on 4-5 for three vehicle classes
2	Point sensor aggregate	Link 4-7	1	Aggregated link count on 4-7
3	Point sensor aggregate	Link 7-8	1	Aggregated link count on 7-8
4	Point sensor aggregate	Link 5-6	1	Aggregated link count on 5-6
5	Point sensor classified	Link 1-2	3	Link count on 1-2 for three vehicle classes
6	Surveillance camera	Node 5 Facing west	5	Aggregated traffic counts on path 4-5-2, 4-5-6, and 4-5-8
7	AVI	Link 1-2 & link 6-9	5	Traffic counts that show up both on link 1-2 and link 6-9 for three vehicle classes

Choosing to implement a particular sensor results in one or more observations, and each of these observations are connected to the unknown O-D volumes (by vehicle class) by the sub-matrices h_d defined in section 3.4. These coefficients are constructed from the multiclass P matrix. Table 3- 2 summarizes the rows of H for the sensors considered in this example, and Table 3- 3 shows the error variances

associated with the various potential observations. For purposes of this example, it will be assumed that all observations (even multiple observations from a single sensor) are independent of one another.

Table 3- 2: Potential rows of H matrix resulting from sensor choices

Sensor		Obs.	OD pair and vehicle class											
Index	Observation	Label	1-6(1)	1-9(1)	4-3(1)	4-9(1)	1-6(2)	1-9(2)	4-3(2)	4-9(2)	1-6(3)	1-9(3)	4-3(3)	4-9(3)
1	1	4-5(1)	0.227	0.198	0.737	0.429	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	2	4-5(2)	0.000	0.000	0.000	0.000	0.207	0.173	0.757	0.371	0.000	0.000	0.000	0.000
	3	4-5(3)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.212	0.191	0.699	0.414
2	4	4-7	0.000	0.292	0.000	0.571	0.000	0.329	0.000	0.629	0.000	0.300	0.000	0.586
3	5	7-8	0.000	0.292	0.000	0.571	0.000	0.329	0.000	0.629	0.000	0.300	0.000	0.586
4	6	5-6	0.594	0.274	0.178	0.253	0.581	0.258	0.143	0.226	0.545	0.249	0.182	0.236
5	7	1-2(1)	0.773	0.509	0.263	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	8	1-2(2)	0.000	0.000	0.000	0.000	0.793	0.499	0.243	0.000	0.000	0.000	0.000	0.000
	9	1-2(3)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.788	0.509	0.301	0.000
6	10	4-5-2	0.000	0.000	0.559	0.000	0.000	0.000	0.614	0.000	0.000	0.000	0.517	0.000
	11	4-5-6	0.227	0.105	0.178	0.253	0.207	0.092	0.143	0.226	0.212	0.097	0.182	0.236
	12	4-5-8	0.000	0.094	0.000	0.176	0.000	0.081	0.000	0.146	0.000	0.094	0.000	0.178
7	13	1-2-6-9(1)	0.000	0.357	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000
	14	1-2-6-9(2)	0.000	0.000	0.000	0.000	0.000	0.352	0.000	0.000	0.000	0.000	0.000	0.000
	15	1-2-6-9(3)	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.360	0.000	0.000

Table 3- 3: Error variances for possible sensor choices

Sensor		Obs.	Obs. Error
Index	Observation	Label	Variance
1	1	4-5(1)	16.014
	2	4-5(2)	0.69
	3	4-5(3)	1.268
2	4	4-7	2.815
3	5	7-8	2.815
4	6	5-6	3.891
5	7	1-2(1)	14.95
	8	1-2(2)	0.993
	9	1-2(3)	1.026
6	10	4-5-2	3.317
	11	4-5-6	5.965
	12	4-5-8	2.22
7	13	1-2-6-9(1)	2.499
	14	1-2-6-9(2)	0.352
	15	1-2-6-9(3)	0.144

The information contained in Table 3- 2 and Table 3- 3 can be used to specify the Δ_d matrices, and Eqs. 3-15 and 3-16 illustrate the computations for sensor option 1 (classified vehicle counts on link 4-5) and sensor option 6 (aggregated turning movements at node 5). The collection of these update matrices for each potential

sensor can be pre-computed and stored for later use in the optimization algorithm.

$$\Delta_1 = \begin{bmatrix} 0.227 & 0 & 0 \\ 0.198 & 0 & 0 \\ 0.737 & 0 & 0 \\ 0.429 & 0 & 0 \\ 0 & 0.207 & 0 \\ 0 & 0.173 & 0 \\ 0 & 0.757 & 0 \\ 0 & 0.371 & 0 \\ 0 & 0 & 0.212 \\ 0 & 0 & 0.191 \\ 0 & 0 & 0.699 \\ 0 & 0 & 0.414 \end{bmatrix} \begin{bmatrix} 16.014 & 0 & 0 \\ 0 & 0.69 & 0 \\ 0 & 0 & 1.268 \end{bmatrix}^{-1} \begin{bmatrix} 0.227 & 0.198 & 0.737 & 0.429 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.207 & 0.173 & 0.757 & 0.371 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0.212 & 0.191 & 0.699 & 0.414 \end{bmatrix} =$$

$$\begin{bmatrix} 0.0032 & 0.0028 & 0.0104 & 0.0061 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0028 & 0.0025 & 0.0091 & 0.0053 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0104 & 0.0091 & 0.0339 & 0.0198 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0061 & 0.0053 & 0.0198 & 0.0115 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0618 & 0.0517 & 0.2266 & 0.1111 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0517 & 0.0432 & 0.1893 & 0.0928 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.2266 & 0.1893 & 0.8306 & 0.4073 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.1111 & 0.0928 & 0.4073 & 0.1997 & 0.0000 & 0.0000 & 0.0000 & 0.0000 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0353 & 0.0319 & 0.1167 & 0.0692 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0319 & 0.0288 & 0.1053 & 0.0624 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.1167 & 0.1053 & 0.3852 & 0.2284 \\ 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0000 & 0.0692 & 0.0624 & 0.2284 & 0.1355 \end{bmatrix}$$

Equation 3- 15

$$\Delta_6 = \begin{bmatrix} 0 & 0.227 & 0 \\ 0 & 0.105 & 0.094 \\ 0.559 & 0.178 & 0 \\ 0 & 0.253 & 0.176 \\ 0 & 0.207 & 0 \\ 0 & 0.092 & 0.081 \\ 0.614 & 0.143 & 0 \\ 0 & 0.226 & 0.146 \\ 0 & 0.212 & 0 \\ 0 & 0.097 & 0.094 \\ 0.517 & 0.182 & 0 \\ 0 & 0.236 & 0.178 \end{bmatrix} \begin{bmatrix} 3.317 & 0 & 0 \\ 0 & 5.965 & 0 \\ 0 & 0 & 2.2 \end{bmatrix}^{-1} \begin{bmatrix} 0 & 0 & 0.559 & 0 & 0 & 0 & 0.614 & 0 & 0 & 0 & 0.517 & 0 \\ 0.227 & 0.105 & 0.178 & 0.253 & 0.207 & 0.092 & 0.143 & 0.226 & 0.212 & 0.097 & 0.182 & 0.236 \\ 0 & 0.094 & 0 & 0.176 & 0 & 0.081 & 0 & 0.146 & 0 & 0.094 & 0 & 0.178 \end{bmatrix} =$$

0.0086	0.0040	0.0068	0.0096	0.0078	0.0035	0.0054	0.0086	0.0080	0.0037	0.0069	0.0090
0.0040	0.0058	0.0031	0.0119	0.0036	0.0050	0.0025	0.0101	0.0037	0.0057	0.0032	0.0117
0.0068	0.0031	0.0995	0.0075	0.0062	0.0027	0.1078	0.0067	0.0063	0.0029	0.0926	0.0071
0.0096	0.0119	0.0075	0.0247	0.0088	0.0103	0.0060	0.0211	0.0090	0.0116	0.0077	0.0242
0.0078	0.0036	0.0062	0.0088	0.0072	0.0032	0.0049	0.0078	0.0073	0.0033	0.0063	0.0082
0.0035	0.0050	0.0027	0.0103	0.0032	0.0044	0.0022	0.0088	0.0032	0.0049	0.0028	0.0101
0.0054	0.0025	0.1078	0.0060	0.0049	0.0022	0.1172	0.0054	0.0051	0.0023	0.1001	0.0057
0.0086	0.0101	0.0067	0.0211	0.0078	0.0088	0.0054	0.0181	0.0080	0.0098	0.0069	0.0206
0.0080	0.0037	0.0063	0.0090	0.0073	0.0032	0.0051	0.0080	0.0075	0.0034	0.0065	0.0084
0.0037	0.0057	0.0029	0.0116	0.0033	0.0049	0.0023	0.0098	0.0034	0.0056	0.0029	0.0114
0.0069	0.0032	0.0926	0.0077	0.0063	0.0028	0.1001	0.0069	0.0065	0.0029	0.0861	0.0072
0.0090	0.0117	0.0071	0.0242	0.0082	0.0101	0.0057	0.0206	0.0084	0.0114	0.0072	0.0237

Equation 3- 16

In Equation 3-15, where the sensor provides class-specific observations, the precision update matrix is block-diagonal, with separate updates for each vehicle class. However, when the observations are aggregated across vehicle classes, as in sensor 6, the precision update matrix (Equation 3-16) does not have that block structure.

Precision updates will always be non-negative (i.e., a sensor cannot provide "negative information"), so adding another sensor will never reduce precision in the estimate of either link volumes or O-D volumes. Translated to the covariance matrices, this means that an additional sensor will never increase a variance, or the trace of the covariance matrices. An additional sensor is always "worth something," even if the incremental value is small. In this example, the available budget is a multiple of the smallest sensor cost, so we can be sure that no budget will go unused in an optimal solution. This allows us to focus only on possible solutions that use the entire budget.

For example, if sensors 5 and 6 (a classified point sensor on link 1-2 and a surveillance camera at node 5) are selected, the sensor budget is exhausted. If it is further assumed that the prior precision matrix contains essentially no information (i.e., a small value of 0.00001 for each diagonal entry), then the precision matrix resulting from the selection of these two sensors can be computed as follows:

$$\left(S_Q^+\right)^{-1} = \left(S_Q^-\right)^{-1} + \Delta_5 + \Delta_6 =$$

0.0486	0.0303	0.0204	0.0096	0.0078	0.0035	0.0054	0.0086	0.0080	0.0037	0.0069	0.0090
0.0303	0.0232	0.0121	0.0119	0.0036	0.0050	0.0025	0.0101	0.0037	0.0057	0.0032	0.0117
0.0204	0.0121	0.1041	0.0075	0.0062	0.0027	0.1078	0.0067	0.0063	0.0029	0.0926	0.0071
0.0096	0.0119	0.0075	0.0247	0.0088	0.0103	0.0060	0.0211	0.0090	0.0116	0.0077	0.0242
0.0078	0.0036	0.0062	0.0088	0.6415	0.4018	0.1991	0.0078	0.0073	0.0033	0.0063	0.0082
0.0035	0.0050	0.0027	0.0103	0.4018	0.2549	0.1242	0.0088	0.0032	0.0049	0.0028	0.0101
0.0054	0.0025	0.1078	0.0060	0.1991	0.1242	0.1767	0.0054	0.0051	0.0023	0.1001	0.0057
0.0086	0.0101	0.0067	0.0211	0.0078	0.0088	0.0054	0.0181	0.0080	0.0098	0.0069	0.0206
0.0080	0.0037	0.0063	0.0090	0.0073	0.0032	0.0051	0.0080	0.6130	0.3942	0.2377	0.0084
0.0037	0.0057	0.0029	0.0116	0.0033	0.0049	0.0023	0.0098	0.3942	0.2578	0.1522	0.0114
0.0069	0.0032	0.0926	0.0077	0.0063	0.0028	0.1001	0.0069	0.2377	0.1522	0.1745	0.0072
0.0090	0.0117	0.0071	0.0242	0.0082	0.0101	0.0057	0.0206	0.0084	0.0114	0.0072	0.0237

Equation 3- 17

The precision matrix can then be inverted to produce the covariance matrix, as follows:

23685	-27640	-16096	118	-11273	12411	11345	-2024	-11128	14920	3919	2535
-27640	45933	-7617	324	13957	-22148	-122	3657	13708	-26198	8382	-3533
-16096	-7617	62112	-993	6114	6413	-33136	-1150	6168	6882	-27774	-627
118	324	-993	62724	-574	-173	2228	-32663	562	61	-1576	-35510
-11273	13957	6114	-574	22011	-30580	-9122	-2514	-10656	14008	4227	1743
12411	-22148	6413	-173	-30580	54642	-12278	2701	11438	-22252	7656	-3412
11345	-122	-33136	2228	-9122	-12278	55003	2666	11328	-77	-29525	1311
-2024	3657	-1150	-32663	-2514	2701	2666	70929	-1487	3442	-1926	-30605
-11128	13708	6168	562	-10656	11438	11328	-1487	25467	-27541	-20130	2789
14920	-26198	6882	61	14008	-22252	-77	3442	-27541	47020	-7347	-3800
3919	8382	-27774	-1576	4227	7656	-29525	-1926	-20130	-7347	65110	-881
2535	-3533	-627	-35510	1743	-3412	1311	-30605	2789	-3800	-881	65590

The trace of this matrix is 600,226.

Using the link utilization coefficients, P, and Equation 3-2, we can compute the covariance matrix of the link volumes and its trace. Since this matrix is 24 x 24, it is not shown completely here, but the trace value is 10,202.

Then, assuming $\lambda = 0.5$, the value of the objective function for this selection of sensors is Z = 305,215.

If different sensors were selected, different subsets of the pre-computed Δ_d

matrices would be used to update the O-D flow precision matrix. These would lead to different resulting covariance matrices, different trace values, and a different objective function value. The objective function value clearly also depends on the assumed values for the λ weights, reflecting varying importance placed on O-D vs. link flow uncertainty.

In this small example, there are only nine combinations of sensor choices that fully utilize the available budget (but do not exceed it), so the selection of optimal sensor types and locations can be done by enumeration. Table 3- 4 summarizes the trace computations for O-D and link flows for each of the combinations.

Table 3- 4: Trace calculations for selections of sensors that use the full budget in the test example

Case	Sensors selected	Cost	$tr(S_V^+)$	$tr(S_Q^+)$
1	2-3-4-6	8	12,184	701,748
2	1-2-4-5	8	2,657	400,177
3	1-3-4-5	8	2,657	400,177
4	1-2-3-5	8	62,845	500,061
5	5-6	8	10,202	600,226
6	2-3-4-7	8	390,569	700,031
7	1-6	8	263,404	700,101
8	1-7	8	753,083	600,048
9	5-7	8	846,603	600,058

In this case, there are two solutions that are clearly best – using sensors 1-2-4-5 or 1-3-4-5. Sensors 2 and 3 provide identical information (see Table 3- 4), so they can be used interchangeably. The combination of a set of classified link counts on links 1-2 and 4-5, plus aggregate link counts on links 5-6 and either 4-7 or 7-8, provide the most information within the available budget.

However, looking beyond the dominant solutions in this example, it is clear from Table 3- 4 that some sensor choices may be better for reducing variance in O-D estimates, while others may do better at reducing link volume variance (compare cases 4 and 5 in Table 3- 4, for example). Thus, as situations become more complicated, using the weighted combination of the two traces can provide solutions that may be quite different from focusing on either one or the other exclusively.

This small example illustrates the general process of selecting sensors, computing the resulting covariance matrices and then the traces and the objective function value; and how the solution process needs to search for a budget-feasible solution that minimizes that objective. It also emphasizes that the computation of the objective function for a given solution is not trivial, so constructing an effective search process for a solution is important.

3.6 **Summary**

The sensor location model is based on a set of linear equations that relate O-D flows, link flows and sensor observations. The sensor information content is then defined using concepts from Kalman filtering, resulting in a measure that focuses on the trace of the covariance matrix for the O-D flows and the trace of the covariance matrix for the link flows. The Kalman filter also provides a linear updating scheme for the precision matrices associated with O-D flows, in which the effect of each potential sensor choice can be represented as a matrix Δ_d. The Δ_d matrix for each sensor can be pre-computed before the optimization begins. Because the information content of the sensors is measured on the covariance matrices, the precision matrix for the O-D

flows must be inverted in order to compute the information gain for any potential set of sensor choices, implying a nonlinear relationship between sensor choices and the objective of maximizing information gain. However, the updating process for the precision matrices provides a structure around which the optimization for sensor location can be formulated in an effective way.

The optimization problem is a nonlinear knapsack problem with a nonseparable objective function and integer decision variables. Problems of this class are known to be difficult to solve exactly, so primary attention has been devoted to developing an effective heuristic solution method. This is the focus of the following chapter.

CHAPTER 4

A SOLUTION ALGORITHM FOR THE SENSOR LOCATION MODEL

4.1 Introduction

The sensor location model developed in Chapter 3 is a nonlinear nonseparable knapsack problem. Although the knapsack structure of the problem is potentially helpful in constructing an efficient solution method, the complexity of the objective function presents difficulties. In this chapter, previous research on solution of various types of knapsack problems is reviewed, and a new two-phase algorithm is proposed for the sensor location problem. The first phase uses a greedy approach to construct a relatively good starting solution for the second phase, which uses Tabu Search to find a near-optimal solution.

The literature on solution of nonlinear knapsack problems includes several general approaches, but most focus on situations where the objective function is separable (e.g., Horst and Tuy, 1990; Hochbaum, 1995; Li, *et al*. 2009) and the most effective solution methods also assume that the variables are continuous rather than discrete (e.g., Bretthauer and Shetty, 1995, 2002; Kodialam and Luss, 1998). Bretthauer and Shetty (2001) provided a survey of algorithms and applications. Powell, *et al*. (2005) discuss approximate dynamic programming as an approach for resource allocation problems with high dimensionality, but the required approximation of the objective function using linear or piecewise linear structure seems problematic for the situation of interest here. Jahangiri, *et al*. (2006) developed a procedure for improving the computational efficiency of dynamic programming for non-convex

separable integer problems, but this also appears to have limited applicability to the current problem.

There is some literature addressing nonseparable quadratic problems. Gallo, *et al*. (1980) and Caprara, *et al*. (1999) developed exact methods for quadratic problems. Hochbaum (2007) proposed a scheme to separate integer quadratic problems using a transformation matrix, which can only be used for relatively small problems. Romeijn (2007) and Sharkey and Romeijn (2011) focused on nonseparable continuous problems, based on the analysis of a family of linear programs which are closely related to the problems.

The literature on quadratic knapsack problems has been of particular interest because quadratic approximations to the objective function are often used in nonlinear optimization. An approach based on the procedure described by Caprara, *et al*. (1999) has been explored, but does not seem to be effective for the problem faced here. The primary challenge is that combinations of more than two sensors have important joint benefits, but these combinations are not evaluated correctly using the quadratic approximation.

The exact methods in the current literature are unable to address the nonlinear, nonseparable problem posed for sensor location in realistic size road networks, so it is necessary to seek a heuristic method that will produce good (if not necessarily optimal) solutions in a reasonable amount of computation time. This chapter proposes a two-phase meta-heuristic method based on Tabu Search, which performs well on the sensor location problem. Section 4.2 presents the algorithm, starting with general ideas and working up through the details of the implementation.

4.2 The Two-Phase Solution Method

The general idea of the algorithm is based on moving from one solution to another along the budget constraint by exploring potential swaps of one or more sensors currently in the solution for an equal-cost subset not currently part of the solution. To evaluate such a swap, the current precision matrix, $\left(S_Q^+\right)^{-1}$, is revised by subtracting the Δ_d matrices for the sensors being swapped out and then adding those for the sensors being added. The resulting tentative precision matrix is inverted, the link covariance matrix is computed and the traces of the two covariance matrices are recorded for computation of the objective function, Z. Based on this value, the new solution is either kept or discarded and the process continues.

The algorithm proposed here is consisted of two phases. The first phase generates an initial solution using a greedy procedure, while the second phase uses a Tabu Search method to perform the swaps. These phases are discussed separately in the following subsections.

4.2.1. Greedy Phase

In the first phase, the unit marginal change that each potential new sensor *d* can bring to the objective value, UMZ_d, is calculated (see Equation 4-1) and ranked.

$$UMZ_d = \frac{Z_d - Z_0}{c_d} \qquad \textbf{Equation 4-1}$$

where Z_0 is the objective value of the current solution set (which may include pre-existing sensors). Z_d is the new objective value with an additional sensor d implemented and c_d is the cost of sensor d. Because implementing an additional

sensor can never increase variances in S_Q^+ and S_V^+, Z_d is always smaller than or equal to Z_0. Thus, UMZ_d can never be positive. The most negative values of UMZ_d are associated with the sensors that contain the most information relative to their cost, given the current solution set. The general idea of using the ratio of change in the objective function over cost to rank items is common in many knapsack algorithms, and selecting items in order of their ranking is a "greedy" procedure that often produces relatively good solutions.

For the current problem, there is an additional element of creating a good starting solution for the Tabu Search procedure – having a mix of different sensor types (and costs) in the solution. When multiple sensor types (at varying costs) are available for potential deployment, the swapping process works more efficiently if the starting solution contains a mixture of sensor types because there is a richer set of initial swapping opportunities.

In order to generate a diverse initial solution, we subdivide the available budget among all available sensor types and choose sensors in a greedy way within each type. The budget allowance for each sensor type *k*, B_k, is calculated as shown in Equations 4-2 and 4-3.

$$B_k = B \frac{C_k}{\sum_{k=1}^{K} C_k} \qquad \textbf{Equation 4-2}$$

$$C_k = \sum_{d \in k} c_d \qquad \textbf{Equation 4-3}$$

where *B* is the total budget allowance and c_d is the cost of sensor d. Thus C_k is

the total cost of implementing all candidate sensors of type k. Then for each sensor type k, sensors are added to the solution set in order of their UMZ_d from the smallest (most negative) to the largest until B_k is exhausted.

When multiple sensor types are available, it is also likely that the set of potential choices includes different types at a single location (link or intersection). Since the values of UMZ_d are not updated during the greedy selection process as new sensors are added to the solution set, some important locations may be selected multiple times with different sensor types and this degrades the quality of the overall solution. In order to prevent duplication of sensors at a single location in the greedy phase, a location prohibition is introduced. Locations already covered by previous sensor choices cannot be selected again. This location prohibition creates a bias where sensor types processed first are likely to be selected for critical locations, and the following sensor types have fewer location choices. To avoid overly restricting the initial solution, the process is repeated for all sensor type permutations so that each type can have the opportunity to be selected at important locations. We then compute the objective function value for all the solutions of different permutation. The best solution is selected as the initial solution for the Tabu Search phase.

The location constraint is only used in the greedy phase. In the Tabu Search phase, UMZ values are updated at each iteration. Thus no constraint regarding the number of sensors to be implemented at a certain location is imposed. The duplicated information will be automatically reflected in the revised UMZ.

This greedy procedure creates an effective starting solution for the swapping

process in the second phase of the algorithm, and does so with relatively modest computation. Testing the Tabu Search with a randomly generated initial solution has demonstrated that the greedy procedure improves both the quality of the final solution and the speed with which it is reached.

4.2.2. Tabu Search

The second phase is a Tabu Search algorithm. Local searches check the immediate neighbors of a current trial solution in the hope of finding an improved solution. At each step, worsening moves can be accepted if no improving move is available, which can help the algorithm to escape a suboptimal region or plateau. In addition, prohibitions represented by a Tabu list, are introduced to discourage the search from coming back to previously-visited solutions. The general procedure of a Tabu Search algorithm is shown in Figure 4- 1.

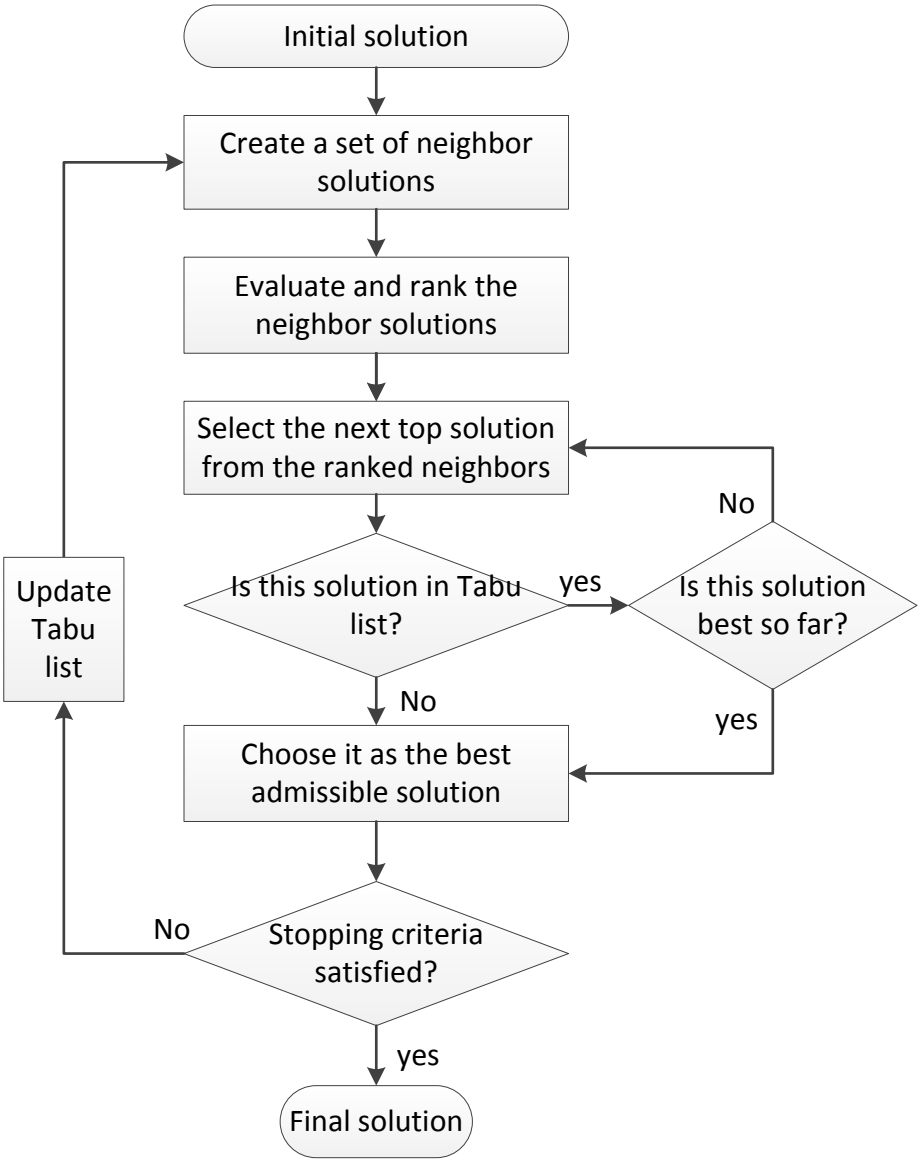


Figure 4- 1: Flow chart of Tabu search.

The two major building blocks of a Tabu Search algorithm are creation of a set of neighboring solutions and the definition of the Tabu list. In the sensor location application, neighbors are defined as the result of a simple swap.

To determine a sensor to be swapped out, all sensors that are currently in the solution set are examined and their UMZ_d value is calculated, with the

understanding that the change in Z is from deleting the sensor rather than adding it. These UMZ_d values are positive, rather than negative. The sensors to be swapped out follow the ranked list of UMZ, starting from the sensors contributing the least information (smallest positive UMZ_d value). N_T is the tenure length of the Tabu list, and sensors on that list are not eligible to be swapped out.

Normally there are far more sensors not selected than selected, so it is computationally expensive to calculate UMZ values for all sensors currently not in the solution set in order to decide which ones to swap in. Therefore, we only evaluate a subset of N_c candidate sensors at each iteration. Those N_c sensors are selected randomly from all candidates at the beginning of each iteration so that each time, a different subset is considered.

Within the set of N_c candidate sensors, the probability that a sensor d is selected for swapping in is based on the *UMZ* values within the candidate subset, S_c. The probability, p_d, is computed using Equation 4-4.

$$p_d = \frac{UMZ_d}{\sum_{i \in S_c} UMZ_i} \qquad \textbf{Equation 4-4}$$

The UMZ_d is defined as adding an extra sensor d, regardless of the number of sensor d currently implemented. Therefore the *UMZ* values of sensors are all negative, and the most negative values are associated with sensors contributing the most information. Thus, Equation 4-4 provides a higher probability of selection for sensors that are likely to contribute more information.

The randomness in selecting the candidate sensor set and selecting actual sensors to be swapped in prevents the algorithm from being trapped in local optima. The direction that *UMZ* provides helps the algorithm to converge faster than if neighbors were selected completely at random.

At the beginning of a neighbor generation, the sensors implemented in the current solution set are evaluated. The sensor with the smallest positive *UMZ* will definitely be swapped out. Then a sensor in the candidate set is selected to be swapped in following the distribution calculated from Equation 4-4. If the sensor selected to enter the solution is less expensive than the sensor being swapped out, additional selections are made from the candidate list until the available budget is consumed. Alternatively, if the sensor being swapped out is less expensive than the one selected to enter the solution, additional sensors are picked to leave from the ranked *UMZ* list until the budget is again balanced. This process generates one neighboring solution to the current solution. At each iteration, a fixed number of neighbors are evaluated and the best available neighbor is considered for adoption as the current incumbent solution.

As additional protection against having the algorithm become "stuck" in a local optimum, the algorithm includes multi-starts. The Tabu search performs multiple times and the best solution among different trials is output as the final solution.

The overall process clearly has several control parameters:

v: the number of neighbors at each iteration in Tabu Search

N_T: the tenure length of the Tabu list

N_c: the total number of candidates considered for entry at each

iteration

N_{tri} : the number of trials performed.

Selection of these parameters is likely to influence the effectiveness of the search process, and experiments to evaluate parameter settings are described in Chapter 5. However, at this stage it is important to state the algorithm carefully and completely.

4.2.3. Algorithm Statement

The description of parameters

D is total number of potential sensor type and location combinations

x_0 is a $D \times 1$ column vector denote pre-existing sensors with integer entries indicating the number of sensor d implemented

S_Q^{m-} is the prior variance-covariance matrix of O-D volume for vehicle class *m*

λ^m is the weights on precision of link volume for vehicle class *m*, $\lambda^m \in [0,1]$

P^m is the link utilization coefficients matrix for vehicle class *m*.

c_d is the cost of sensor d

p_d is the probability of sensor d to be swapped in

Δ_d is the precision update of sensor d

UMZ_d is the unit marginal change in the objective value that sensor d can bring

B is the total budget

B_k is the budget allowance of sensor type k

v is the number of neighbors at each iteration in Tabu Search

T is Tabu list

N_T is the tenure length of the Tabu list

N_c is the total number of candidates considered for each iteration

N_{tri} is the number of trials performed

totFE is the total number of function evaluations allowed per trial

S_{out} is the set of sensors to be swapped out

S_{in} is the set of sensors to be swapped in

S_c is the set of candidate sensors to be considered at current iteration

x_{cur} is the current solution set

Z_{cur} is the objective value for x_{cur}

x_{best} is the best solution so far

AL is the aspiration level, the best objective value so far

totSp is the total spending of x_{cur}

Two-phase algorithm for a sensor allocation problem

Input: $x_0, S_Q^{m-}, c_d, B, \lambda^m, P^m, \Delta_d, v, N_T, N_c, totFE, N_{tri}$

Output: x_{best} and AL

Procedure:

Phase 1: Greedy Search

Step 1: Find an initial solution, x_1, using Greedy procedure.

Step 1.1: Compute Z_0 from x_0 and UMZ_d for $d = 1,...,D$ using Equation 4-1.

Step 1.2: Rank all D sensors according to their UMZ_d s from smallest (negative, with biggest reduction on Z) to biggest and categorize them according to their sensor type.

Step 1.3: Compute B_k for each sensor type k using Equation 4-2, 4-3.

Step 1.4: For each sensor type permutation $k_1,...,k_K$, starting from k_1, select the sensors of type k_1 according to the ranked UMZ from top to bottom until reach the budget allowance B_{k1}. Repeat the process for k_2 to k_K with location constraint.

Step 1.5: Compute objective function values of the solutions for each permutation. Keep the solution with the lowest objective function value and denote it as x_1.

Phase 2: Tabu Search

Step 2: Initialize T, set $AL = +\infty$, $x_{cur} = x_1$

Step 3: Generate v neighbors from x_{cur}

Step 3.1: Randomly select N_c candidate sensors from all D candidate sensors and put them into S_c. Compute p_d using Equation 4-4 for sensors in S_c.

Step 3.2: For each sensor implemented in x_{cur} (does not include pre-installed sensors), compute UMZ. Sort sensors by UMZ from the biggest to the smallest.

Step 3.3: Pick the first sensor in the sorted UMZ and add it to S_{out}. Randomly select one sensor from S_c following p_d and add it to S_{in}

Step 3.4: Compute the total spending, $totSp$,

1) If $totSp > B$, take out one more sensor on the top of UMZ list and go back to the beginning of step 3.4

2) If $totSp < B$, and $B - totSp \geq \min_{\forall d}(C_d)$, randomly pick a sensor with cost smaller than $B - totSp$ following p_d and add it to S_{in}. Go back to the beginning of step 3.4.

3) If $totSp \leq B$ and $B - totSp \leq \min_{\forall d}(C_d)$, proceed to step 3.5.

Step 3.5: Record the neighbor and corresponding S_{out}, S_{in}. Reset S_{out}, S_{in} to $\varnothing$. Continue to generate next neighbor by going back to step 3.3 until all v neighbors are generated.

Step 4: Evaluate the objective function value for all v neighbors and rank neighbors according to their objective value from smallest to the biggest. Pick the top most solution in v neighbors.

Step 5: If the S_{out} for this neighbor is not in the Tabu list T, set this neighbor as x_{cur}, add S_{in} to T and keep newest N_T entries. If $Z_{cur} < AL$, set $AL = Z_{cur}$, $x_{best} = x_{cur}$

Step 6: If the S_{out} for this neighbor is in the Tabu list T, but $Z_{cur} < AL$, set this neighbor as x_{cur}, add S_{in} to T and keep newest N_T entries, set $AL = Z_{cur}$, $x_{best} = x_{cur}$

Step 7: If the S_{out} for this neighbor is in the Tabu list T, and $Z_{cur} \geq AL$, move to the next neighbor on the ranked list and go back to step 5

Step 8: If all neighbors violate T and none of them can override, go back to step 3 with step 3.3 modified as "Pick the first sensor not in the Tabu list in the sorted UMZ and add it to S_{out}. Randomly select one sensor from S_c and add it to S_{in}." If all currently implemented sensors are part of T, thus S_{out} definitely violate T. Keep the neighbor with the smallest objective function value.

Step 9: If the current number of function evaluation is smaller than $totFE$, increment the iteration number and go back to step 3

Step 10: Repeat step 2 to step 9 for N_{tri} times. Select the best solution as x_{best} and output its corresponding AL.

4.3 **Summary**

The search algorithm proposed in this chapter is designed to solve the nonlinear nonseparable knapsack problem for sensor location formulated in Chapter 3. The algorithm consists of two parts: a greedy phase to generate an initial solution and a Tabu Search phase to swap sensors along the budget constraint. The neighbor generation in Tabu search is a combination of a fixed swap-out strategy with a guided random swap-in strategy.

Because we cannot guarantee that the objective function Z is convex or that there is particular structure involved in that function, there is no good way to generate a performance guarantee on the heuristics. That is we cannot guarantee that we will

find a solution within certain percent of optimal. The evaluation of the heuristics must be strictly experimental. The purpose of the experiments in the following two chapters is to test its ability to get reasonable solutions in different situations. It is to be expected that the structure of the network on which the sensor location has been done might have effects on the nature of the Z function. The mechanism that's present in this problem for representing that structure is through the coefficient matrices P and H. Part of the testing is to look at the way in which those coefficient matrices are constructed and how that affects the solution.

Both the sensor location model and the search algorithm for its solution contain parameters. The parameters of the formulation affect the character of the solutions for sensor choice and location. The parameters of the search algorithm affect how efficiently those solutions can be found. The next chapter uses an example network as a test bed for experiments to evaluate the effects of these parameters.

CHAPTER 5

TESTING AND EVALUATION OF THE MODEL

To test the concepts and solution algorithm, a series of experiments has been developed. These experiments are designed to evaluate parameter settings for the solution method and to provide insights into sensor location solutions under varying budgets and with varying importance placed on information related to O-D volumes and link flows.

5.1 Test Network

All the tests are performed on the "Sioux Falls" (SF) network, shown in Figure 5- 1. It originated from a representation of part of the street network in Sioux Falls, South Dakota. This network, first constructed and used by LeBlanc, et al. (1975), has since become a "standard" test network for many types of transportation network algorithms. We have borrowed the basic structure of the network from the original version used by LeBlanc, et al. (1975), but we have created O-D tables and link characteristics that enhance the network's usefulness as a test bed for the sensor location algorithm.

Three vehicle classes (see Table 5- 1) are used, representing automobiles, medium trucks and heavy trucks. The O-D tables are based on 7 origin and destination zones (node 1, 6, 7, 10, 13, 15, and 20) and exhibit different flow patterns between automobiles and trucks (see Table 5- 2, Table 5- 3, and Table 5- 4). Node 10 is assumed to be a zone in a downtown area, where many trips (both automobile and

truck) begin and end. Node 13 is a loading area at the periphery of the city, and is more important for truck trips than for automobile trips.

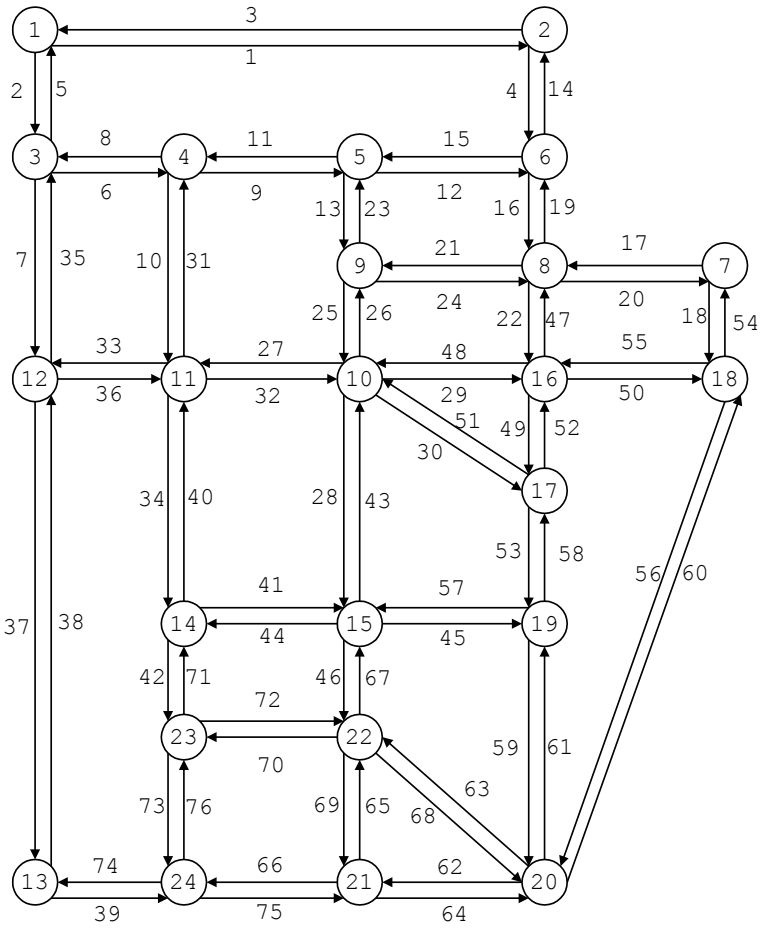


Figure 5- 1: Sioux Falls network

Table 5- 1: Vehicle class information in Sioux Falls network

Class number	Type	Cost of time Coefficient	Cost of distance Coefficient	Vehicle equivalents
1	Automobile	0.2	0.25	1
2	Medium truck	0.33	1	2
3	Heavy truck	0.5	1.5	3

Table 5- 2: Seven-zone O-D table for vehicle class 1 (veh/hr)

O\D	1	6	7	10	13	15	20
1	0	491	440	1070	532	839	359
6	491	0	306	269	224	266	264
7	442	308	0	1348	90	723	90
10	1064	267	1343	0	532	533	794
13	526	221	90	537	0	708	91
15	846	267	714	534	719	0	1069
20	355	267	88	807	89	1071	0

Table 5- 3: Seven-zone O-D table for vehicle class2 (veh/hr)

O\D	1	6	7	10	13	15	20
1	0	14	37	121	69	24	19
6	25	0	19	37	16	7	10
7	30	19	0	50	37	7	18
10	116	45	62	0	116	38	49
13	68	23	30	113	0	27	25
15	19	7	14	33	25	0	15
20	19	3	18	33	28	3	0

Table 5- 4: Seven-zone O-D table for vehicle class 3 (veh/hr)

O\D	1	6	7	10	13	15	20
1	0	42	53	121	41	48	39
6	38	0	8	19	13	6	8
7	68	20	0	37	18	10	18
10	115	34	40	0	26	28	23
13	33	15	8	36	0	18	14
15	27	7	17	32	8	0	14
20	32	13	21	23	5	8	0

For each O-D pair, free flow conditions on the links are used to compute path choice probabilities for each vehicle class using a probit model of path choice. These

calculations create the values for the P matrix in Equation 3-2, and hence the h_d matrices for each of the sensor types and locations considered.

5.2 Sensor Types and Candidate Sensors

As sensing technology advances, more and more types of data are available to be used for travel demand and traffic volume estimation. In terms of vehicle classification ability, sensors can be divided into classification sensors, dual sensors and aggregated sensors. The classification sensors can differentiate all vehicle classes assumed. The dual sensors can only provide observations of automobiles and non-automobiles. The aggregated sensors only give the total counts regardless of vehicle class observed. Sensors can also be divided into several categories according to their implementation locations and observations they provide.

Point sensors associated with links

These sensors are normally implemented along the road side (e.g. microwave and laser radar sensors), underneath the pavement (e.g. inductive loop detectors, magnetic sensors), or mounted overhead above the road surface (e.g. infrared sensors). They can provide traffic counts on this link either classified or aggregated.

Point sensors associated with intersections

These sensors are normally implemented at intersections so that they can provide turning movements at the intersection. The most widely used type is a surveillance camera. Together with video image processors, cameras can provide the number of vehicles turning left, going straight, or turning right from each leg of the

intersection. Those counts can either be aggregated or classified. Normally at least two cameras are needed for each intersection to ensure the quality of the image for processing. Other sensors of this type can be a set of Bluetooth detectors at an intersection.

Partial path sensors

These sensors provide partial path information by identifying specific vehicles in order to trace their trajectories. For example, GPS-based systems obtain data from online GPS users, RFID sensors only detect vehicles with an RFID tag, and cell-phone data follows users with a cell phone from a specific carrier. All of these data are subject to the problems of market penetration and sample representation when used for travel demand and traffic volume estimation purpose. We didn't assume any sensor from this category in our tests, but this type of data can definitely fit into our model frame and algorithm if desired.

For the purpose of testing, we have assumed 5 types of sensors (see Table 5- 5). Sensor costs and error characteristics are estimated based on a report by Middleton, *et al*. (2009), as well as papers by Rabiu (2013), Zhang, et al. (2007), and Taghvaeeyan, *et al*.(2014). For the current tests, these characteristics are assumed to be the same for all locations in the network. Sensor type 1 and 2 in Table 5- 5 can be assumed as any types of point sensors associated with link. Sensor types 3 to 5 are assumed to be different types of surveillance cameras together with image processing software, capable of separating varying numbers of vehicle classes. To get a good quality image for processing, two cameras are assumed to be installed at each intersection.

Table 5- 5: Sensor types assumed in tests

Index	Location type	# class	Counting error	Overcount ratio	Classification error	Cost ($/Lane or $/Node)
1	Link	1	2%	50%	0%	1800
2	Link	5	2%	50%	5%	4550
3	Intersection	1	2%	50%	0%	11800
4	Intersection	2	2%	50%	9%	14160
5	Intersection	5	2%	50%	13%	16992

The third column of the table is the number of vehicle classes these sensors can differentiate. As we can see, sensor types 2 and 5 can differentiate up to 5 vehicle classes. Since we only assumed three vehicle classes for our tests, the actual observations for these types of sensors only differentiate 3 vehicle classes. The R matrix associated with each sensor can be generated using the method described in the Appendix.

5.3 Parameter Tuning

The first set of experiments is intended for tuning the parameters of the Tabu Search algorithm. These parameters are: the tenure length of the Tabu list (N_T), the number of neighbors explored at each iteration (ν), the size of candidate sensor pool to be swapped in when generating neighbors (N_c), the maximum number of objective function evaluations for each trial (*totFE*), and the number of trials performed for the multi-start purpose (N_{tri}). The parameters and candidate values are listed in Table 5-6.

Table 5- 6: Parameters and candidate values

Parameter	Range
Tenure length of the Tabu list	$\{N_T \in \mathbb{Z} \mid 1 \le N_T \le 10\}$
Number of neighbors explored at each iteration	$\{\nu \in \mathbb{Z} \mid 1 \le \nu \le 20\}$
Size of candidate sensor pool	$N_c \in \{50, 70, 90, 110, 130, 150, 170, 190, 210\}$
Number of trials	$N_T \in \{1, 2, 3, 4, 5\}$
Maximum number of function evaluations	$totFE \in \{50000, 25000, 16667, 12500, 10000\}$

These five parameters can be divided into two groups. The first three are more concerned with the performance of the algorithm, while the latter two are more about how to allocate the limited computational effort. Since they have different purposes and tuning them together with all possible combinations can be computationally very expensive, we tuned these two groups separately.

For the parameter tuning experiments, the problem-specific constants – the weight on link volume estimation in the objective function (λ) and total budget to spend (B) – are set to intermediate values for the test network. We set λ = 0.5, meaning that we focus equally on link volume estimation and travel demand estimation. After some trial runs, a budget of \$125,000 seems to be in the middle of a budget range-of-interest. Therefore, the budget is set to be \$125,000 during the whole parameter tuning process. More tests will be performed on these two parameters in the next two sections to see how solutions change with different objective emphasis and budget availability.

After some trial runs, the maximum number of function evaluations of 25000 and the number of trials of 2 seems performs well. Therefore, we set those two

parameters as above when tuning the tenure length, number of neighbors and size of candidate set. We explore all the combinations of the possible values mentioned in Table 5- 6 for the first three parameters. Altogether there are 1800 possibilities. The average posterior objective function value across all trials is used to evaluate different cases in order to pick a set of parameters that generate a stable result.

The top 10 combinations are shown in Table 5- 7. Over all 1800 cases, there seems to be a pattern that the algorithm performs better when the number of neighbors, ν, is bigger than 15, the tenure length, N_T, is less than or equal to 3 and the size of candidate pool, N_c, is around 70. However, among the top performed cases, the average Z^+ seems to be in a similar range. This indicates that some randomness still plays a role. We set the number of neighbors to be 19, tenure length to be 2 and size of candidate pool to be 70 in all following parameter tuning and testing since this case came to the top. However, some other combinations could also produce very acceptable results.

Table 5- 7: Top 10 combinations of ν, N_T, and N_c

ν	N_T	N_c	$\overline{Z^+}$
19	2	70	34383
11	1	70	34584
16	2	90	34881
20	2	70	34921
20	2	150	34964
15	2	90	35132
20	3	70	35214
17	3	70	35249
13	3	110	35283
20	3	90	35321

We tuned the parameters concerning computational effort with the first group of parameters fixed to their best value. The parameter candidates are designed in a way that the overall computational effort (i.e. the total number of function evaluations per trial times the number of trials) is around 50,000 for all cases tested here. We first tested $totFE$ from 1000 to 10,000, resulting to a N_{tri} ranging from 50 to 5. The result is shown in Table 5- 8.

Table 5- 8: Sorted result of parameter tuning for computational effort 1

$totFE$	N_{tri}	Total computational effort	$\overline{Z^+}$
10000	5	50000	23686
8000	6	48000	25122
6000	8	48000	25372
7000	7	49000	25466
9000	6	54000	25487
4000	13	52000	28091
5000	10	50000	28670
3000	17	51000	29588
2000	25	50000	29954
1000	50	50000	47198

It seems that algorithm performs better when more computational effort is spent on each trial, rather than using it to increase the number of trials. Therefore, we expanded the range of $totFE$ tested. The result is shown in Table 5- 9. It seems that this expanded test further confirms our conclusion. Therefore we pick the $totFE$ as 25000 and N_{tri} as 2 for all later tests.

Table 5- 9: Sorted result of parameter tuning for computational effort 2

$totFE$	N_{tri}	Total computational effort	$\overline{Z^+}$
25000	2	50000	22544
50000	1	50000	22791

12500	4	50000	24052
10000	5	50000	24359
16667	3	50001	25307

The parameters used for subsequent tests are summarized in Table 5- 10.

Table 5- 10: Parameter configuration

Parameter	Value
ν	19
N_T	2
N_c	70
totFE	25000
N_{tri}	2

5.4 Changes of Budget

We want to see how the solution changes when the budget availability changes. Therefore, we performed a series of tests from a budget of $50,000 to $250,000 at $25,000 increments. A budget of $50,000 means that around a quarter of the links in the network can be covered if the budget is spent solely on aggregated sensors. An increment of $25,000 can add around 10 more aggregated link count sensors or 4 classified link count sensors or 2 aggregated cameras or 1 dual or classified camera. Figure 5- 2 summarizes the result for the objective function value, *Z*. Increasing the budget clearly reduces Z^+, but at a much slower rate as the budget exceeds $100,000, meaning that the marginal effect an extra dollar brings after $100,000 is relatively small. The primary range of interest might be between $75,000 and $150,000.

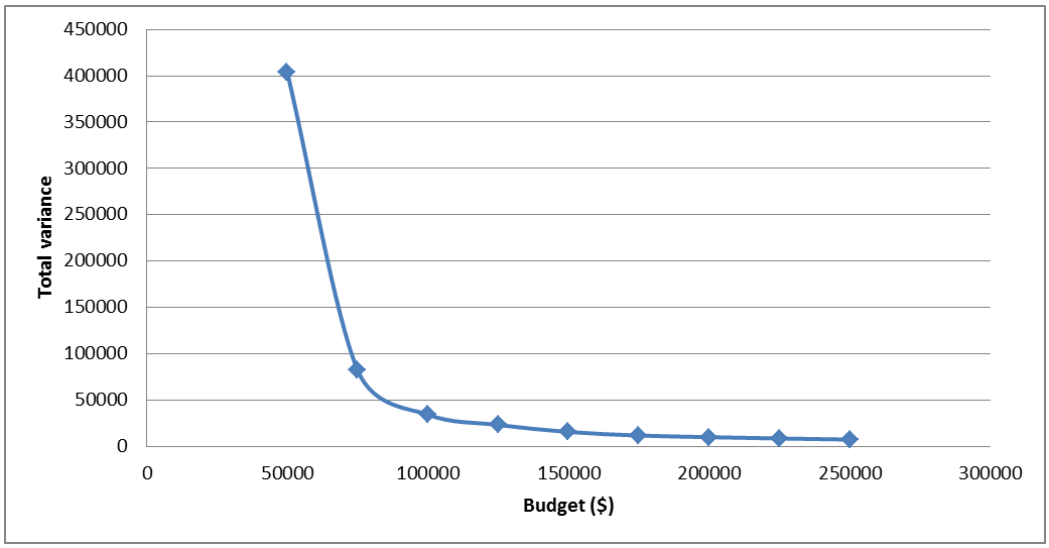


Figure 5- 2: Z^+ under different budget level

Table 5- 11 shows the breakdown of Z^+ into its components, $tr\left(S_Q^+\right)$ and $tr\left(S_V^+\right)$, as the budget changes. Because these two components are equally weighted in the objective, it is not surprising that their values are somewhat similar in the optimal solutions at each budget level, but as the budget increases, there is somewhat greater emphasis on reducing $tr\left(S_V^+\right)$ and the ratio shown in the last column of the table increases slightly.

Table 5- 11: Objective function value and breakdown under different budgets

B	Z^+	$tr\left(S_Q^+\right)$	$tr\left(S_V^+\right)$	$tr\left(S_Q^+\right)/tr\left(S_V^+\right)$
50000	404265	410860	397670	1.03
75000	82583	72727	92439	0.79
100000	34579	38587	30570	1.26
125000	23375	25656	21093	1.22
150000	15803	18098	13507	1.34
175000	11937	14185	9689	1.46
200000	10178	11957	8399	1.42

225000	8768	10523	7013	1.50
250000	7578	9625	5531	1.74

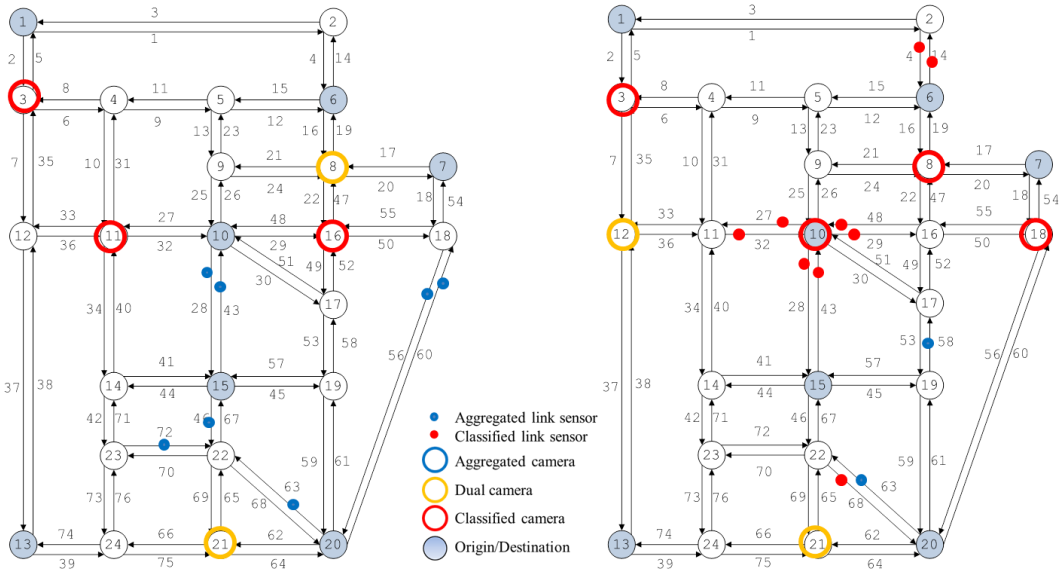
Table 5- 12 shows the number of sensors of each type chosen in the best solution for each budget level. The first column is the budget level, with columns 2 to 6 corresponding to the number of sensors chosen for aggregated link count sensors, classified link count sensors, aggregated cameras, dual cameras and classified cameras respectively. The solutions include many more link count sensors than cameras. This reflects the fact that link sensors are much less expensive and it is necessary to deploy sensors across several parts of the network to successfully reduce Z^+. As the budget increases, classified link counters replace aggregate counters and more classified intersection sensors are deployed. Cameras capable of producing only aggregate counts of turning movements at intersections are hardly used at all. The results summarized in Table 5- 12 contain at least three important conclusions. First, use of sensors capable of providing vehicle classification is highly desirable when the available budget allows such choices. Second, an optimal solution is likely to contain a mix of sensor types, deployed to complement one another. Third, if intersection surveillance is used as part of the sensor solution, it is important to use that to obtain vehicle classification information in turning counts, not just aggregate counts.

Table 5- 12: Number of sensors chosen for each sensor type

B	# agg link	# cls link	# agg cam	# dual cam	# cls cam
50000	11	0	0	2	0
75000	6	0	1	1	2
100000	7	0	0	2	3

125000	1	5	0	3	3
150000	2	9	0	2	4
175000	4	13	0	1	5
200000	1	16	0	1	6
225000	1	19	0	3	5
250000	0	27	0	1	5

Based on the data in Figure 5- 2 and Table 5- 12, two solutions have been selected for further analysis and comparison – those for budgets of $100,000 and $150,000. The solution at $100,000 contains a mix of aggregate link counters and intersection surveillance for vehicle classification information. The solution at $150,000 shows substitution of classified link counters for many of the aggregate counters and achieves a total Z^+ value that is approximately one-half of the value at a budget of $100,000. Figure 5- 3 shows the sensor deployment under both solutions.



B=100,000 *B*=150,000

Figure 5- 3: Sensor deployment with budgets of $100,000 and $150,000

Three of the five nodes at which intersection cameras are installed in the solution for B = $100,000 are retained in the solution for B = $150,000 (nodes 3, 8 and 21), although the installation at node 8 is upgraded from dual class to full classification. Much of the additional budget is used in the vicinity of node 10 (the major origin and destination for all traffic classes). Because there are many trips that do not go "through" this intersection, but terminate there, the combination of link counters and intersection cameras for through movements allows much greater reduction of uncertainty regarding the various O-D volumes for all vehicle classes.

Table 5- 13 provides more detail regarding this point, showing the initial variances and final variances for all O-D pairs with origin or destination at node 10. The first column of the table indicates the O-D pair and vehicle class, and the second column shows the initial variance for each O-D volume. The very large variances are all for the automobile vehicle class because the anticipated volumes are much higher for auto trips than for the two truck classes. The third column in the table shows the final variances in the run with budget of $100,000. All the auto O-D variances have been reduced quite dramatically. The truck O-D volume variances have also been reduced, although for some O-D pairs (e.g., 15-10 and 20-10) the final values are only slightly smaller than the initial values. For these O-D pairs, the sensors deployed provide little information.

Table 5- 13: Prior and posterior O-D volume variance related to node 10

O-D	$tr\left(S_Q^-\right)$	$tr\left(S_Q^+\right)_{100,000}$	$tr\left(S_Q^+\right)_{150,000}$
10-1(1)	377370	706	292
10-6(1)	23763	328	291

10-7(1)	601220	443	340
10-13(1)	94341	223	65
10-15(1)	94696	1481	394
10-20(1)	210150	1786	918
1-10(1)	381630	722	298
6-10(1)	24120	328	274
7-10(1)	605700	424	268
13-10(1)	96123	221	64
15-10(1)	95052	1434	282
20-10(1)	217080	1533	586
10-1(2)	4485	433	227
10-6(2)	675	189	222
10-7(2)	1281	196	123
10-13(2)	4485	357	217
10-15(2)	481	479	154
10-20(2)	800	631	264
1-10(2)	4880	451	250
6-10(2)	456	160	194
7-10(2)	833	176	106
13-10(2)	4256	364	233
15-10(2)	363	362	142
20-10(2)	363	327	247
10-1(3)	4408	113	87
10-6(3)	385	72	43
10-7(3)	533	31	22
10-13(3)	225	72	59
10-15(3)	261	261	52
10-20(3)	176	153	99
1-10(3)	4880	121	91
6-10(3)	120	49	33
7-10(3)	456	31	21
13-10(3)	432	79	66
15-10(3)	341	340	61
20-10(3)	176	155	126

The last column in Table 5- 13 shows the final variances for the run with a budget of $150,000. There is general improvement for all vehicle classes, relative to the results at a $100,000 budget. This is a direct result of the additional classified sensors placed near node 10 in the solution for the $150,000 budget.

If we look at the best solutions for both trials in all 9 budget cases (18 total runs), we note that more than half of the sensor candidates (123 out of 224, or 55%) are never chosen in any trial. The top ten sensor choices (based on the number of times they appear in the 18 solutions) are summarized in Table 5- 14. All of these are classified sensors, and seven of the ten are in the vicinity of node 10. This further emphasizes the importance of sensing flows in the center of this network.

Table 5- 14: Top 10 sensors frequently chosen among all trials for all budget cases

Sensor	Location	Sensor type	Times chosen
90	6-2	2	12
119	15-10	2	12
210	10	5	12
104	10-15	2	11
208	8	5	11
80	2-6	2	10
105	10-16	2	10
124	16-10	2	10
103	10-11	2	9
108	11-10	2	9

5.5 Changes of Relative Weight on O-D Volumes and Link Flows

In this set of tests, we examined the effects that the changes in λ, the relative weight of O-D variances and link flow variances in the objective function. As λ increases, the weight on O-D volume variance decreases and more emphasis is put on link volumes. Eleven values of λ, from 0 to 1 at 0.1 increments, were tested using a budget of $100,000. The result is organized in Table 5- 15. The overall value of the objective Z^+ is relatively insensitive to λ. As λ increases, the measure of O-D volume variance ($tr\left(S_Q^+\right)$) increases slightly and the measure of link flow variance ($tr\left(S_V^+\right)$) decreases slightly, but it is only when $\lambda = 1$ (the objective function value depends entirely on link volume variance) that there is a marked change, with $tr\left(S_Q^+\right)$ increasing substantially as it no longer factors into the objective.

Table 5- 15: Changes on objective value for different λ

λ	Z^+	$tr\left(S_Q^+\right)$	$tr\left(S_V^+\right)$	$tr\left(S_Q^+\right)/tr\left(S_V^+\right)$
0	31950	31950	36414	0.88
0.1	36543	36231	39346	0.92
0.2	36604	36167	38350	0.94
0.3	35237	35352	34967	1.01
0.4	35270	36584	33298	1.10
0.5	34579	38587	32755	1.06
0.6	33559	34766	32755	1.06
0.7	31420	34006	30311	1.12
0.8	35289	38895	34388	1.13
0.9	31058	37490	30343	1.24
1	29271	53493	29271	1.83

Additional insight can be gained by examining the sensor deployment solutions for three values of λ (0, 0.5 and 1). The results are shown in Figure 5- 4, Figure 5- 5, and Figure 5- 6 respectively. When $\lambda = 0$, the solution makes considerable use of aggregate link sensors. In the intermediate solution ($\lambda = 0.5$), there is a shift away from these aggregate link sensors and an additional intersection has been instrumented. As the weight on link flow variance continues to increase to $\lambda = 1$, the use of aggregate link counters further diminishes. These results indicate that the more interested we are in the variance of link flow volumes, the more budget should be expended on intersection surveillance with classified counts and the less useful are the aggregate link counters. This result is somewhat surprising, because we might expect that if the primary interest is in link volumes, it should be desirable to deploy counters on more links, even if the counts are aggregate. However, when there are multiple vehicle classes of interest, it is vital to obtain vehicle classification information and the intersection surveillance is particularly useful for that purpose.

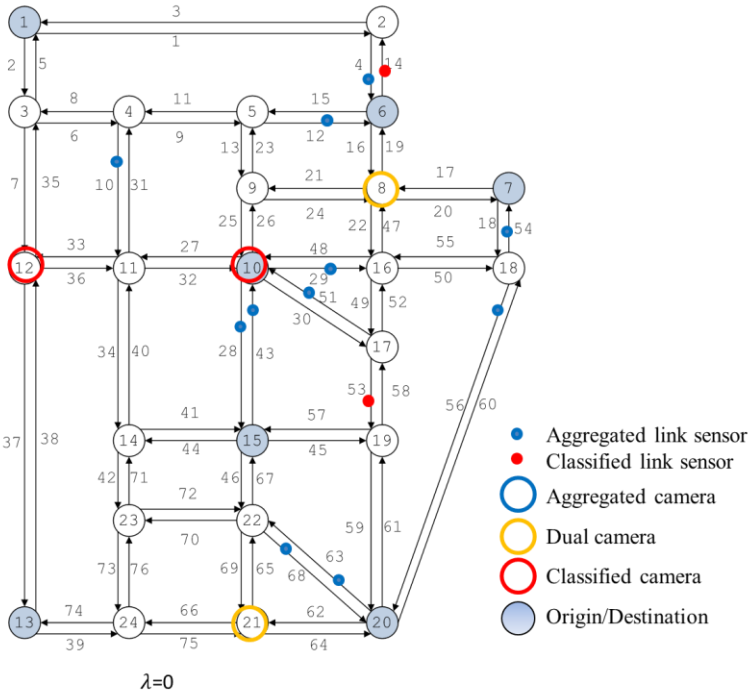


Figure 5- 4: Sensor allocation for $\lambda = 0$

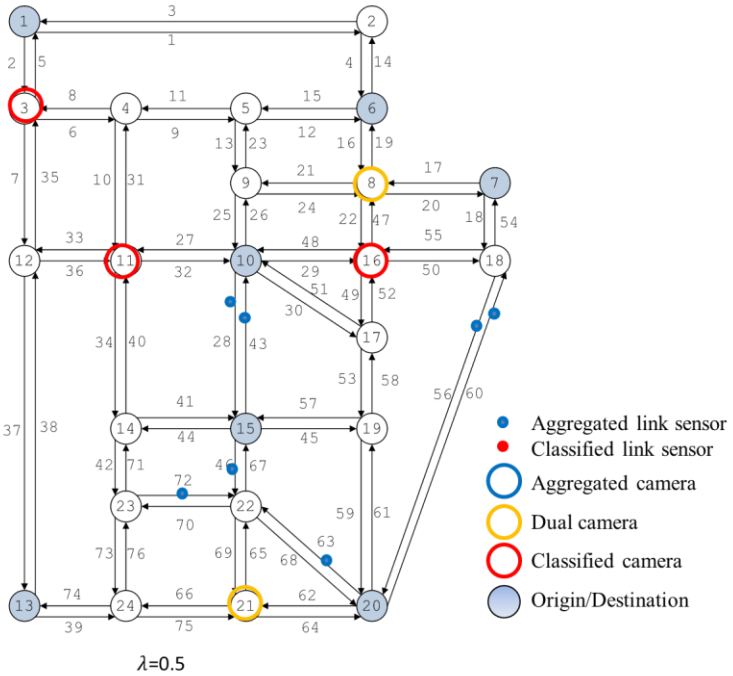


Figure 5- 5: Sensor allocation for $\lambda = 0.5$

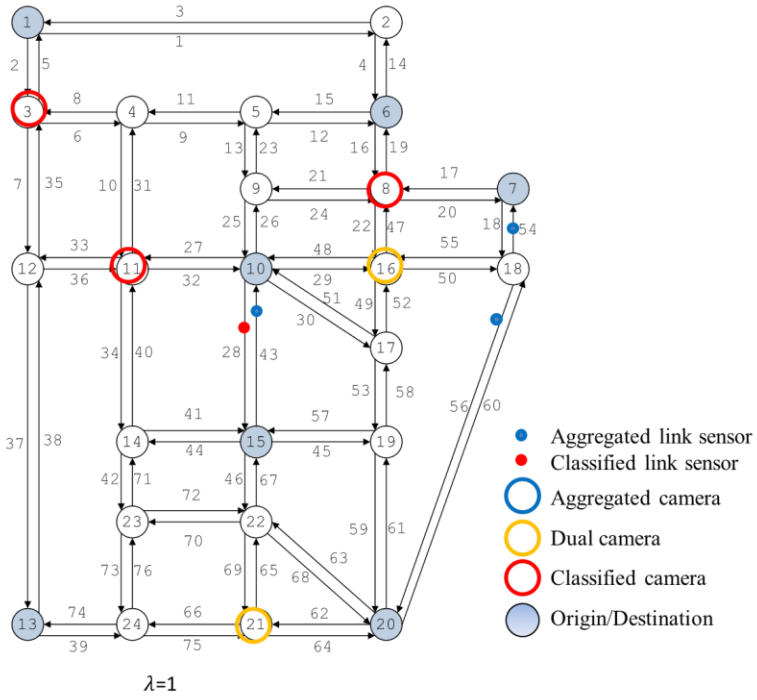


Figure 5- 6: Sensor allocation for $\lambda = 1$

5.6 Sensitivity Analysis for the P Matrix

In all experiments to this point, we have been using the link utilization coefficients derived from an empty network, which means the link costs are not updated by the actual traffic volume. We assumed this because it is much easier information to get than the link utilization of an equilibrium situation. We only need to know the network characteristics, e.g. link length, capacity, topology, to get the link utilization from the empty network. However, in order to get the utilization coefficients from equilibrium, we need to know flow information, which is hard to get before sensor implementation. However, it is useful to evaluate how sensitive the sensor location decisions are to the assumed link utilization coefficients. Therefore, we perform a set of tests comparing use of Stochastic Network Loading (SNL)

coefficients in the matrix P with use of coefficients from a Stochastic User Equilibrium (SUE) flow pattern. This is possible in this test situation because an actual set of O-D tables is available and can be assigned to the network to obtain the SUE solution.

A budget of $100,000 and a weight of 0.5 are used for the tests. Table 5- 16 and Figure 5- 7 show the comparison. The objective function value of SUE is generated from the SUE solution evaluated under the SNL utilization coefficients for calculation of $tr\left(S_V^+\right)$ in order to make two solutions comparable. The overall objective value of the two solutions is very similar, indicating that the information content of the two solutions is essentially the same.

Table 5- 16: Prior and posterior for SNL and SUE

Case	Z^+	Z^-	Z^+ / Z^-
SNL	34579	8090050	0.0043
SUE	34455	8090050	0.0043

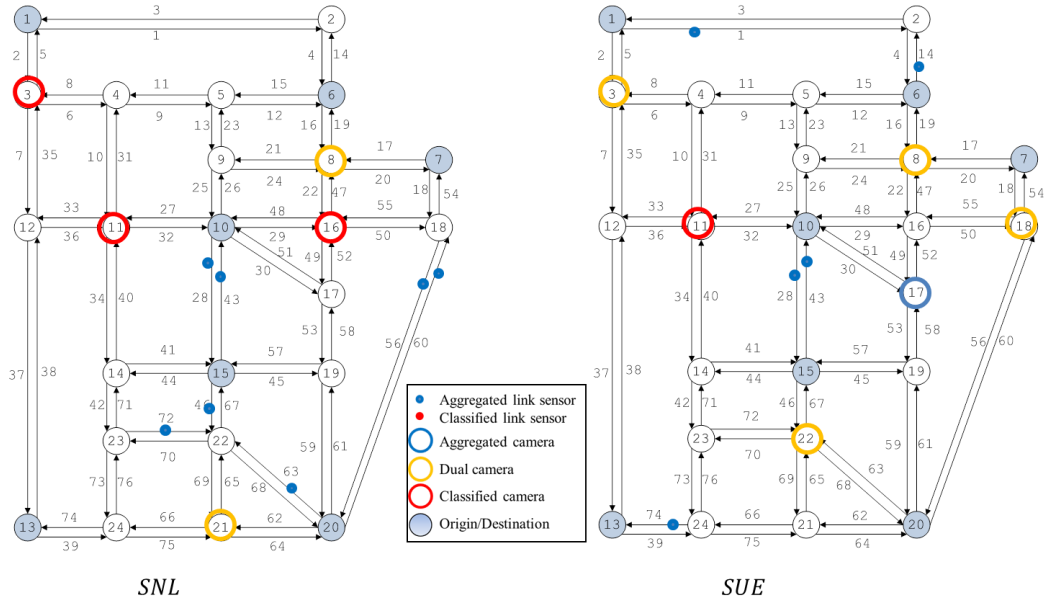


Figure 5- 7: Sensor allocation for SNL and SUE

As we can see from Figure 5- 7, there are clearly some differences in the sensor selection and location, particularly for the aggregated link counters. The intersections chosen for camera implementation are either the same or adjacent with the chosen sensor type a little bit different. We conclude from this test that the sensor locations chosen, especially for link counters, are likely to be somewhat sensitive the use of unloaded network link utilization coefficients, but that the information content of the solutions is very insensitive. That is, there are likely to be alternative sensor deployment solutions that yield approximately the same information. Using values in the *P* matrix that do not reflect actual current flows may result in choosing a different sensor location solution, but may not be harmful in reducing the actual information obtained.

5.7 Summary

This chapter analyzes the characteristics of the sensor allocation solutions on the Sioux Falls network. Tests on different budget level, B, and relative weight, λ, have been presented. Sensitivity analysis on utilization coefficients in the P matrix is also performed.

As more budget is available, the classified link count sensors and classified cameras are more frequently chosen to address the variance of multiple vehicle classes. Meanwhile, inexpensive aggregated sensors are less likely to be chosen. There is a clear preference for more detailed information on the various vehicle classes, and this allows the overall variance measure to be improved. In the Sioux Falls network, there is a concentration of activity (trip ends by all vehicle classes) around one major node, and the solutions tend to focus sensors in that area, regardless of budget. However, as the available budget increases, the types of sensors deployed around that node change and allow much better resolution of the various vehicle class O-D volumes. The solutions across different levels of available budget also show the importance of an integrated solution using multiple sensor types.

In terms of λ, different weight put emphasis differently on $tr\left(S_V^+\right)$ and $tr\left(S_Q^+\right)$ resulting in different spatial arrangement of the solution. As more emphasis is put on link volume variances in a multi-class context, the number of aggregated link count sensors selected is smaller, and more emphasis is placed on obtaining vehicle classification information.

The sensitivity analysis at the last part of this chapter show us that the utilization coefficients generated using empty network is a good substitute for actual utilization coefficients. The information content of the resulting solutions is quite similar.

In the next chapter, a case study on a real network is presented to shown the scalability of the model and solution method.

CHAPTER 6

A CASE STUDY IN ROCKLAND COUNTY, NEW YORK

Rockland County is on the west side of the Hudson River, just north of New York City, located as shown in Figure 6- 1. The county is roughly triangular in shape, covering 199 square miles with a total population of just over 310,000. Figure 6- 2 shows additional detail about the county, including some of the main highways. The county contains a portion of the New York State Thruway (I-87) leading to the Tappan Zee Bridge and is roughly bisected by the Palisades Interstate Parkway (PIP), running north-south. It also contains the connections between I-87 and two other major facilities going south into New Jersey – the Garden State Parkway (GSP) and I-287. The PIP and the GSP are closed to commercial traffic. However, in addition to I-87 and I-287, portions of several state highways in the county are heavily used truck routes.

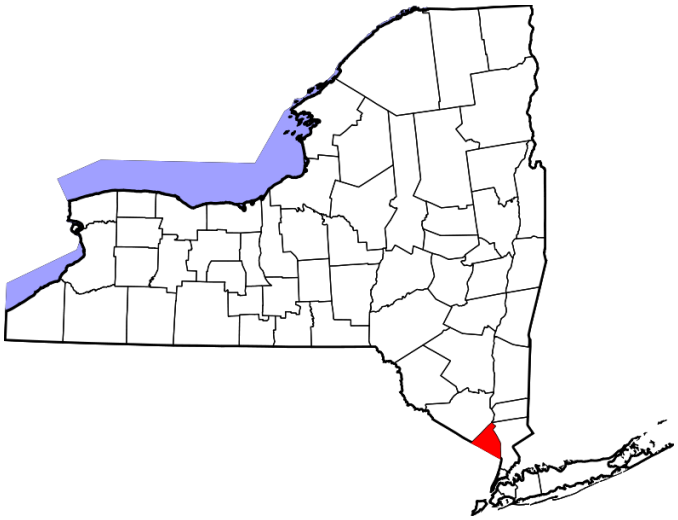


Figure 6- 1: Location of Rockland County, New York.

The principal sources of primary data for this test case are O-D and link utilization estimates generated by Zhao (2013). Those estimates were calculated from the average daily vehicle counts collected by the New York State Department of Transportation, and estimates of truck trip generation based on land use. The experiments with this test network are designed to answer two basic questions:

1) Does the methodology developed in this research scale reasonably to a much larger network than used in Chapter 5, with a much more complex structure and variety of facilities?
2) How might a traffic management or transportation planning agency in an area like Rockland County (a suburban county near a major metropolitan area) use the sensor location model most effectively?

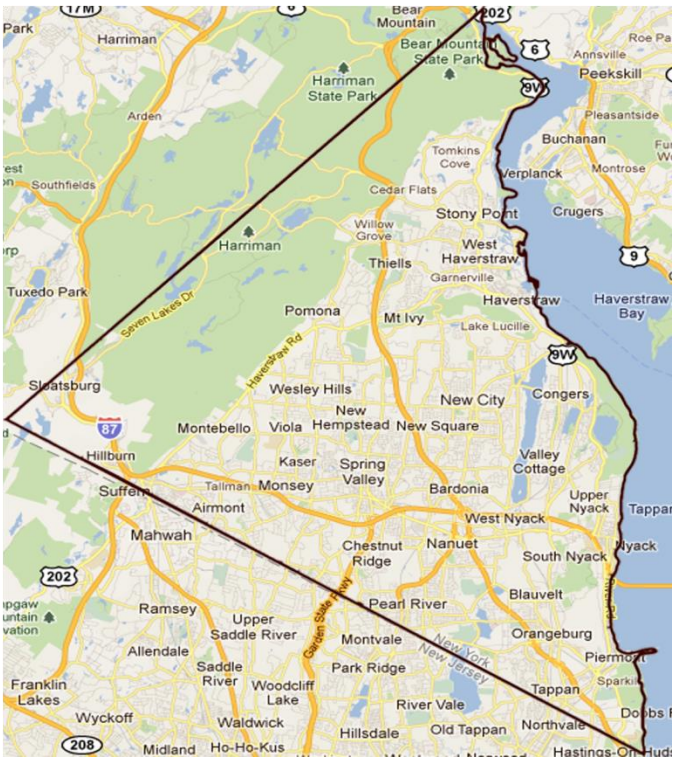


Figure 6- 2: Rockland County, New York.

6.1 Network Construction

The network representation used contains 25 traffic analysis zones (TAZs), where trips originate and terminate. 14 of these zones are internal and the remaining 11 represent the entry/exit points to the county from adjoining areas, including the Tappan Zee Bridge. There are thus 600 O-D pairs for each vehicle class, although some of these (origins and destinations for trucks at the external zones representing the parkways) are required to have zero trips. Table 6- 1 summarizes the location of the TAZs.

Table 6- 1: Traffic Analysis Zones

TAZ code	Description
301	External zone on the north of the Rockland county, linked by highway 9W
302	External zone on the northwest of the Rockland county, linked by Palisades Interstate Parkway
303	Local zone representing Stony Point
304	Local zone representing Haverstraw town
305	Local zone representing Haverstraw village
306	Local zone representing New Hempstead
307	Local zone representing New Square
308	Local zone representing Clarkstown
309	Local zone representing Spring valley
310	Local zone representing Nyack
311	External zone on the east of Rockland County, linked by Tappan Zee Bridge
312	External zone on the south of Rockland County, linked by I- 287
313	Local zone representing Suffern
314	External zone on the south of Rockland county, representing NJ local
315	Local zone representing Airmont

316	Local zone representing Chestnut Ridge
317	External zone on the south of Rockland County, linked by Garden State parkways
318	External zone on the south of Rockland county, representing NJ local
319	Local zone representing Orangetown
320	External zone on the south of Rockland county, representing NJ local
321	External zone on the south of Rockland County, linked by NJ Palisades Interstate Parkway
322	Local zone representing Sloatsburg
323	External zone on the west of Rockland county, representing Orange county local
324	External zone on the west of Rockland County, linked by I-87
325	Local zone representing West Haverstraw village

Two vehicle classes are assumed, automobiles and trucks. Theoretically there are 1200 O-D pairs for two vehicle classes. However, prior work on estimating O-D tables for this network (Zhao, 2013) produced estimates in which many O-D pairs have zero volume and many others have very small volumes. For the experiments conducted here, the set of O-D pairs of interest has been limited to 288 O-D pairs that are likely to have volumes of at least 10 veh/h for automobiles or 5 veh/h for trucks. The resulting problem is still of substantial size, but a large number of potential O-D pairs that contribute little to the understanding of traffic flows in the network have been eliminated. This type of reduction to focus on pairs of significant interest in determining sensor locations is likely to be an important part of most realistic implementations of the ideas developed in this thesis.

The network includes 277 nodes and 771 links (including centroid connectors to the TAZs). The network is designed to represent the designated state and county

highways, with other links added as necessary to make connections. The overall network is shown in Figure 6- 3.

Each link has attributes of length, number of lanes, free-flow speed and overall capacity. The capacity values are based on the functional classes of the links, using the class definitions in Table 6- 2 (from NYSDOT). Centroid connectors for the internal TAZs are assumed to be 0.5 mile in length and have a fixed speed of 25 mph.

Table 6- 2: Road classes included in the Rockland model.

Functional Classification Code	Description
11	Urban Interstate
12	Urban other Freeway and Expressway
14	Urban Principal Arterial
16	Urban Minor Arterial
17	Urban Major Collector
20	Ramp (speed limit below 45 mph)
21	Ramp (speed limit greater or equal to 45 mph)
999	Centroid Connector

A total of 304 links and 26 intersections are identified as candidate locations for potential sensor deployment (see Figure 6- 4). Selected links include state highways (i.e., 9W, 17, 45, 59, 202, 303, 304, 306, and 340), important county routes (i.e., Routes 20, 74, 80, 108, and 210), links that connect to external zones (representing entrances to and exits from the county) and other principal arterials. On each candidate link, either an aggregate link counter or a dual-loop counter (that can count cars and trucks separately) can be implemented. There is also no constraint regarding the number of sensors can be implemented on a single link. Because link counts contain

errors and separate counters are assumed to produce independent observations, duplicating sensors on a single link has the effect of reducing the error variance for flows. With a limited budget for sensor deployment, it is usually preferable to deploy sensors on different links, but if the greatest information gain can be obtained by duplicating one or more sensors, the optimization allows that.

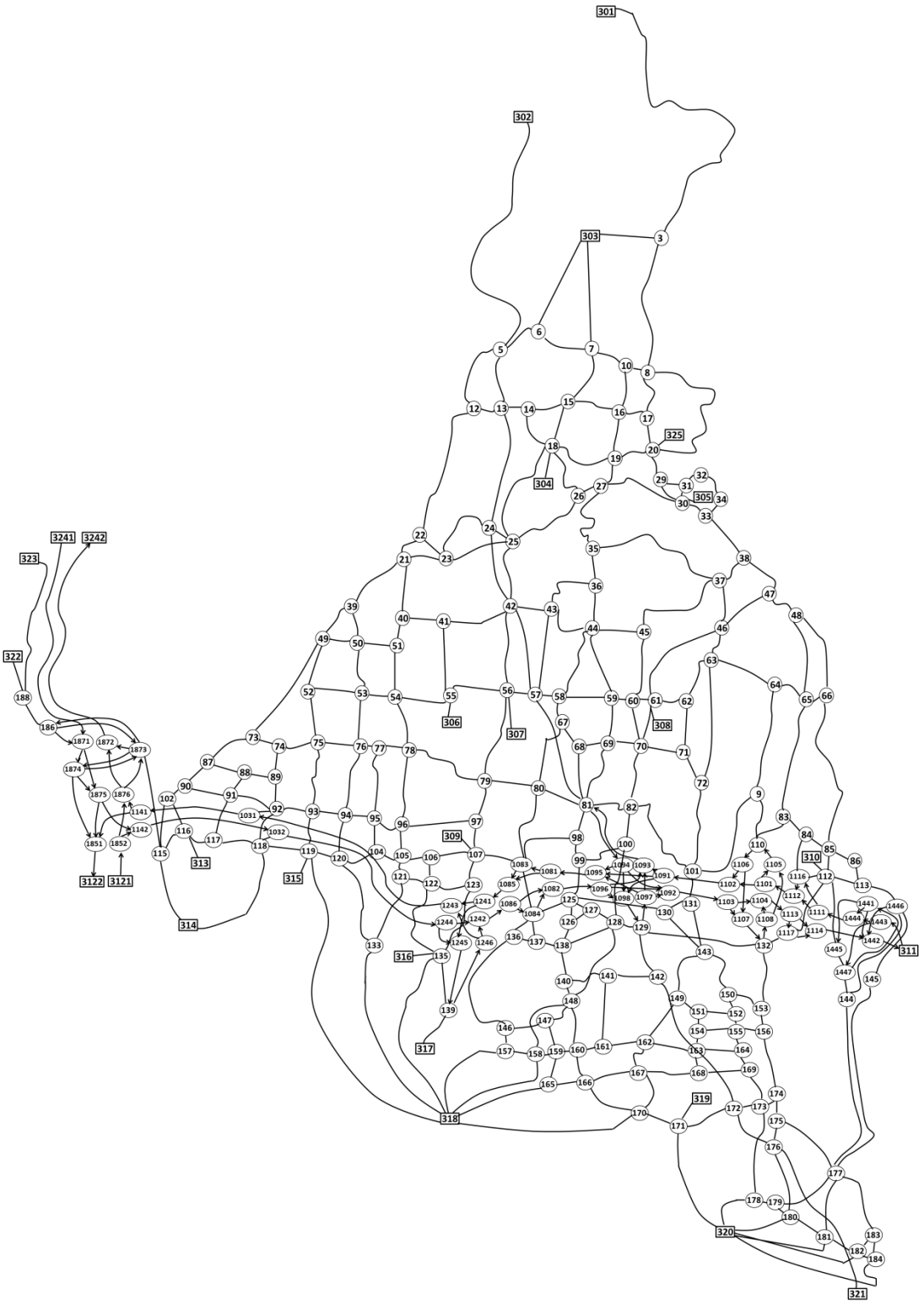


Figure 6- 3: Rockland County network.

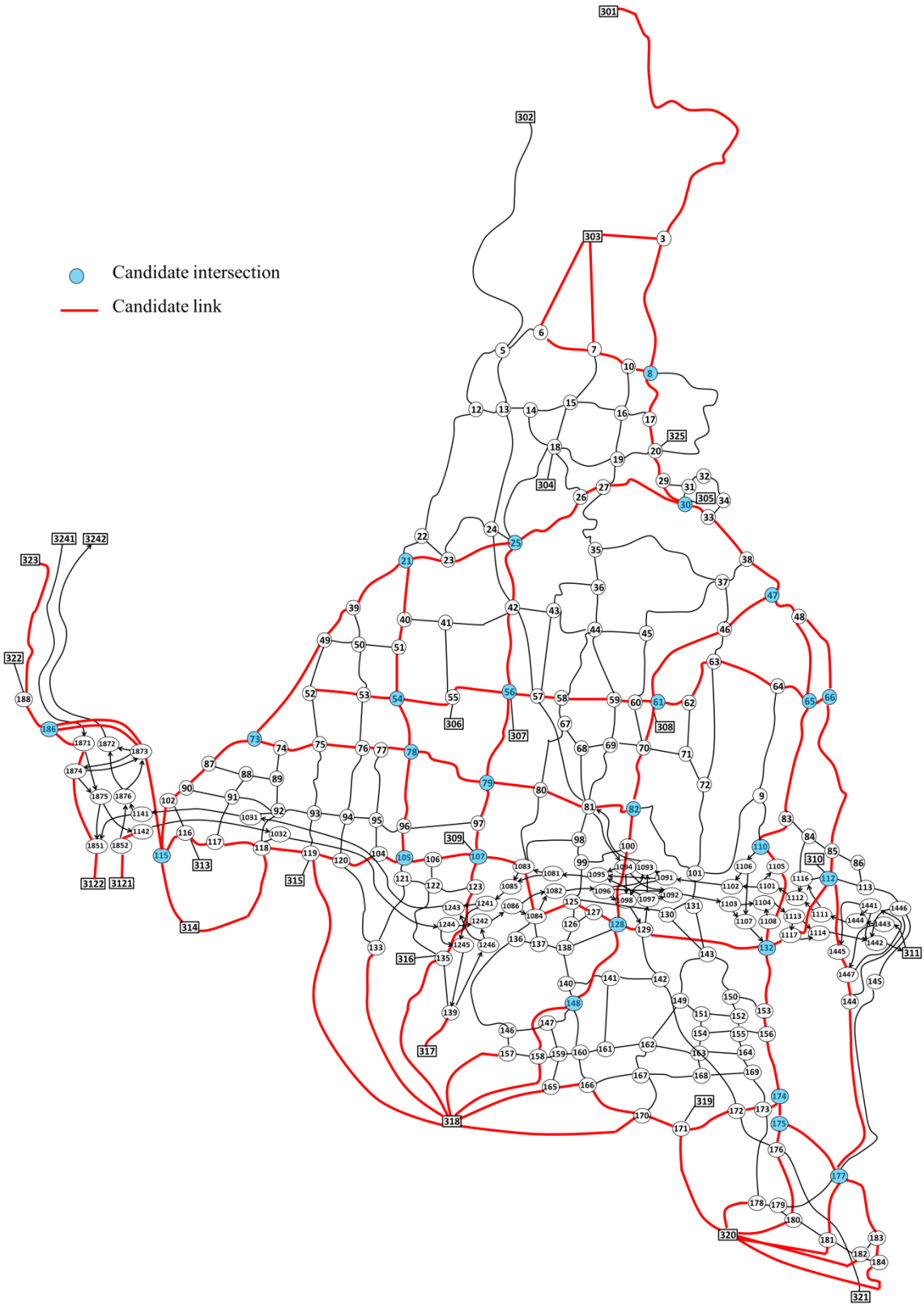


Figure 6- 4: Candidate links and intersections

Links on the New York State Thruway and the PIP are not considered for sensor placement because in this case study the perspective taken is that of a county-level agency and Rockland County does not have jurisdiction over either the Thruway or the PIP. However, the prior conditions for the sensor placement optimization assume that accurate data is already being collected on the Thruway by the agencies that operate the facility.

The candidate intersections for camera placement are primarily the intersections of the state and county highways mentioned above. These are likely to be among the most important intersections in the network.

6.2 Prior Covariance Matrices

The sensor location model operates by updating prior covariance matrices for O-D flows and link volumes into posterior matrices that have minimum trace values (sums of variances). To implement this process, it is necessary to specify what the prior matrices (either covariance or precision) are. It is possible to use a zero precision matrix as a representation of no prior information, but in practice that is somewhat unrealistic. The fact that the network exists and is in operation implies some level of prior information, even if the network is not well instrumented. In this case study, some prior information has already been asserted as the basis for reducing the number of O-D pairs of interest.

For this analysis, prior estimates of O-D volume variances for the 288 O-D pairs of interest have been constructed by assuming that the values estimated by Zhao (2013) represent an average and the actual (unknown) values are uniformly distributed

between 0 and twice the estimate. This results in a prior variance estimate for each O-D pair that is one-third the estimated volume squared. These prior variance estimates distinguish between O-D volumes that are likely to be relatively large (and have large prior variances) and those that are likely to be relatively small (and have small prior variances). The placement of sensors in the network is then more likely to focus on the higher-volume O-D pairs in an effort to reduce the trace of the posterior covariance matrix. All off-diagonal elements (covariances) in the prior covariance matrix are assumed to be zero. The prior covariance matrix for the link flows is computed from the covariance matrix of O-D volumes using Equation 3-2.

In addition to the assumed prior information on O-D volume and link flow variances, the effects of the assumed pre-existing sensors on the Thruway (I-87) are incorporated into the initial covariance matrices S_Q^- and S_V^-. The Thruway is assumed to have vehicle counts and classification information along each link, with small error rates of 0.5% for vehicle counts and 0.5% for vehicle classification, with classification errors assumed to be equally distributed in each direction.

6.3 Possible Sensors and Characteristics

Three types of available sensors are assumed: aggregated link count sensors, classified link count sensors and classified cameras. These sensors have characteristics shown in Table 6- 3. For each sensor type, the simulation method described in Appendix has been used to construct an *R* matrix for the sensor, based on the counting error rate, classification error rate, and overcount ratio shown in Table 6- 3.

Table 6- 3: Sensor types assumed in tests

Index	Location type	# class	Counting error	Overcount ratio	Classification error	Cost ($/Lane or $/Node)
1	Link	1	2%	50%	0%	1800
2	Link	5	2%	50%	5%	4550
3	Intersection	2	2%	50%	9%	14160

Entries for the rows of *H* corresponding to the link count sensors are developed from the link utilization probabilities in the matrix *P*, constructed using free-flow impedances on the network links for each vehicle class. The *H* matrices for cameras need partial path (turning movement) probabilities at the relevant intersections. These turning movements probabilities are derived using a simple Markovian approximation based on link probabilities. Figure 6- 5 provides an example.

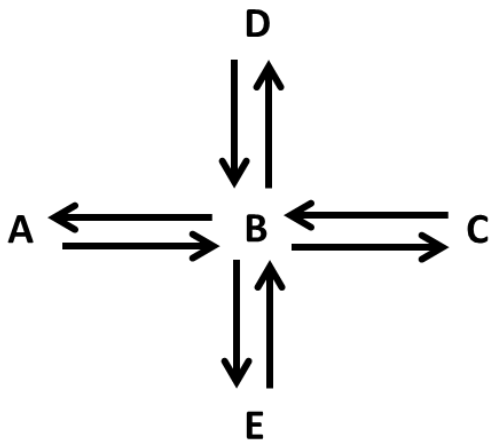


Figure 6- 5: Example intersection

The probability of turning movement A → B →D, P_{ABD} for flow on a specific O-D pair is computed as:

$$P_{ABD} = P_{AB} \frac{P_{BD}}{P_{BD} + P_{BC} + P_{BE}} \qquad \textbf{Equation. 6-1}$$

where P_{AB} is the link utilization coefficient (link probability) for link *AB* for that O-D pair, and the other terms are defined similarly. Other turning movement probabilities (e.g. P_{ABE}, P_{CBD}) can be calculated in the same way. These values provide entries in the rows of *H* corresponding to the camera sensors at intersections.

6.4 Experiments and Results

Two sets of computational experiments are done for the Rockland County network. The first is sensor allocation using only the two types of link count sensors under varying levels of available budget. This set of experiments accomplishes three objectives: 1) it demonstrates the efficacy of the sensor location method on a realistic network; 2) it establishes a baseline result constructed with only the most basic sensor types available; 3) it demonstrates how the solution changes with a varying budget. The second set of experiments adds cameras as a potential sensor type, and provides an ability to evaluate the role of video sensing of vehicle classification within an overall sensing plan.

In each experiment, we allocate 10000 objective function evaluations per trial and 15 trials for each budget case. We also use a tenure length of 3, and 10 neighbors for the Tabu search. Lambda (the relative weight of O-D variances and link flow variances) is set as 0.5, putting equal weight on both.

6.4.1 Link Count Sensors and Budget Variations

In the first experiment, at each candidate link, one or more aggregated or classified link count sensors can be implemented. The available sensor budget is varied from \$100,000 to \$450,000 at \$50,000 increments. Figure 6- 6 summarizes the result for the objective function value, *Z*, as well as its two components, $tr\left(S_Q^+\right)$ and $tr\left(S_V^+\right)$.

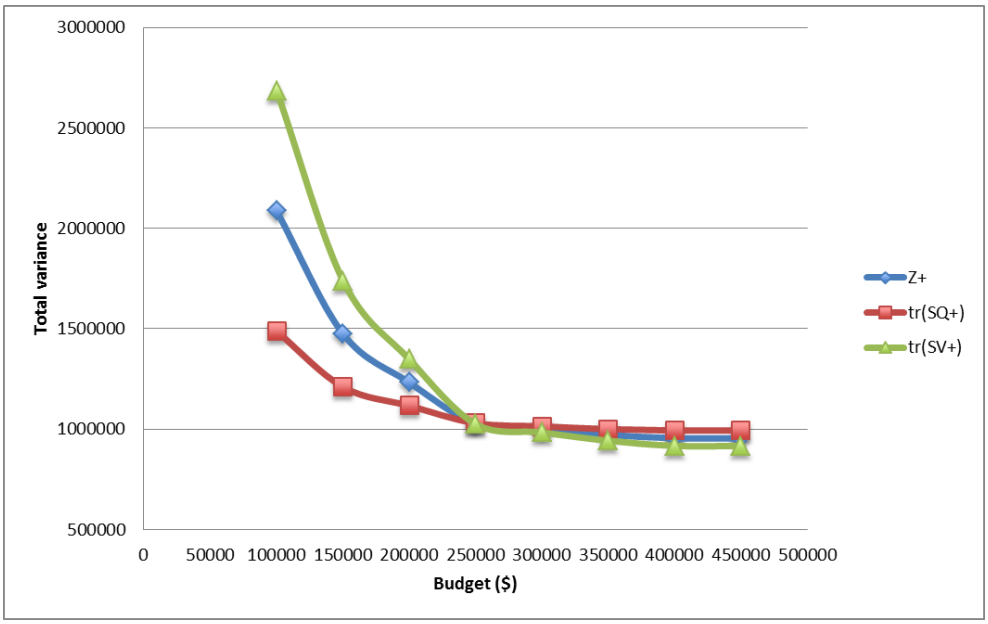


Figure 6- 6: Objective value over different budgets

For budgets between \$100,000 and \$250,000, each budget increment produces substantial improvement in the variances of both O-D volumes and link flows. However, at budgets beyond \$250,000, the further improvements are quite modest. Thus, for this network the budget of \$250,000 appears to be a very important point.

Some further interpretation of the Z value is also useful. At a budget of \$250,000, both $tr\left(S_Q^+\right)$ and $tr\left(S_V^+\right)$ have values of approximately 1,000,000. Since there are 288 O-D pairs included in the trace computation for S_Q^+, the average variance for an O-D pair volume is about 3500. This implies the average standard deviation on an estimated O-D volume is approximately 59 vehicles/hour. For the link flows, there are 771 links in the network, with two vehicle classes, so there are 1542 volumes included in the trace computation. Thus, the average variance is about 650, corresponding to an average standard deviation on a link flow estimate of approximately 25 vehicles/hour. Of course, these average standard deviations do not apply to every O-D pair and link, but the magnitudes give some general meaning to the value of Z in the optimization.

The solution *covers* 81% of O-D pairs (234 out of 288). That is, some fraction of the volume for 234 O-D pairs is observed at one or more of the sensors. Among the 54 O-D pairs that are not covered, 17 cannot be covered by any sensor in the current candidate set. The other 37 uncovered O-D pairs have small O-D volume estimates, and hence small variances. The O-D pairs that cannot be covered by any candidate locations are related to I-87, I-287, the PIP and the Garden State Parkway. If some ramp links for these limited-access facilities were included in the candidate location set, coverage of those 17 O-D pairs could be achieved, but for the current experiments the ramps are not considered accessible locations.

The results summarized in Table 6- 4 and Table 6- 5 provide additional insight into the effects of sensors on variances for specific O-D pairs and links. Table 6- 4

shows results for the 20 O-D pairs that have largest prior estimated volumes. The column labeled "O-D" indicates the origin and destination nodes, and the "1" indicates that the flows are for vehicle class 1 (automobiles). All of these high-volume O-D pairs are external-external flows; i.e., both the origin and destination are outside Rockland County, but the trips move across the County. The "prior_initial" column indicates the variance on the estimated O-D flow based on the prior estimate, and the column "prior_preinstall" shows the variance after the effect of the assumed pre-existing information on I-87. The values in this column are entries in S_Q^-. The column labeled "diff_preInstall" contains the reduction in O-D variance between the prior values and the "preinstall" values. Eleven of the top 20 O-D pairs are affected by the data from the Thruway. The last eight columns show the reduction of O-D variance (relative to S_Q^-) from the sensor installations at varying budget levels (from \$100,000 to \$450,000). The darker the green color, the more reduction is achieved. The sensor placement decisions are quite effective at reducing the variance in estimated volumes for these large O-D flows, and most of the effect for these large O-D pairs is obtained even at relatively small budgets. With a limited budget, the most effective way to reduce $tr\left(S_Q^+\right)$ is to locate sensors that reduce the largest variances first.

Table 6- 4: Variance reduction on top volume O-D pairs

						(SQ+) - (SQ-)							
Index	OD	Volume	prior_initial	prior_preinstall	diff_preInstall	100000	150000	200000	250000	300000	350000	400000	450000
177	3241-3122(1)	3193	3398600	1110	-3397490	-1	-1	-1	-1	-1	-1	-1	-1
89	3121-3242(1)	1499	749250	67	-749183	0	0	-1	-1	-2	-4	-3	-3
74	3111-3182(1)	1055	371020	113320	-257700	-111477	-111584	-112458	-112543	-112990	-113069	-113058	-113091
57	3091-3102(1)	990	327020	37442	-289578	-20563	-20615	-20565	-20854	-20896	-20903	-20906	-20911
36	3061-3182(1)	986	324060	324060	0	-323314	-323817	-323838	-323910	-323811	-323975	-323974	-323974
54	3081-3112(1)	959	306490	40674	-265816	-36323	-36040	-38354	-40112	-40177	-40136	-40363	-40388
11	3031-3052(1)	931	288740	288740	0	-287887	-287878	-286900	-288023	-287934	-288119	-288129	-288130
69	3101-3202(1)	861	247330	116	-247214	0	0	0	0	0	0	-1	0
76	3111-3212(1)	848	239820	27	-239793	0	0	0	0	0	0	-2	0
112	3161-3062(1)	834	231810	231810	0	-230049	-231663	-231679	-231669	-231685	-231752	-231761	-231757
160	3211-3202(1)	771	198210	198210	0	-190340	-192664	-197883	-197982	-197998	-197884	-197880	-197880
166	3221-3232(1)	764	194460	194460	0	-159383	-159383	-159633	-159383	-159383	-159383	-159383	-159383
4	3021-3092(1)	763	194070	194070	0	-180036	-182705	-188035	-189329	-190091	-190692	-190874	-190635
171	3231-3122(1)	753	189160	70812	-118348	-43546	-43557	-43559	-44122	-43565	-44127	-44129	-44128
91	3131-3072(1)	690	158580	158580	0	-155791	-158417	-157184	-158442	-158430	-158452	-158468	-158462
156	3211-3112(1)	671	149900	17	-149883	0	0	0	0	0	0	0	0
45	3071-3132(1)	657	143840	143840	0	-143113	-142499	-143212	-143547	-143271	-143698	-143730	-143722
77	3111-3242(1)	655	143130	51411	-91719	-23244	-23346	-23483	-23678	-23721	-23753	-23745	-23763
67	3101-3112(1)	618	127200	46720	-80480	-38636	-38631	-38636	-39622	-39648	-39696	-39725	-39731
26	3051-3082(1)	615	126250	126250	0	-124987	-126070	-125721	-125999	-125995	-126008	-126007	-126017

Table 6- 5 presents similar information, but for link flows. The links shown are the 20 highest volume flows in the network. All are vehicle class 1 (automobiles), as indicated in the second column of the table. Most of these high-volume links are on the Thruway and the assumed sensor data from the Thruway reduces most of the variances to small values before the location of additional sensors. The sensor location model successfully addresses the remaining links, and again most of the effect on these high-volume links is achieved even at small budgets. As with the elements of $tr\left(S_Q^+\right)$, the most effective way to reduce $tr(S_V^+)$ is to focus on the largest values first.

Table 6- 5: Variance reduction on top volume links

index	ODIB	Loc	volume	prior_ini	prior	diff preInstall	diff 100000	150000	200000	250000	300000	350000	400000	450000
693	1242-1086(1)	1242-1086	5791	622570	17	-622553	0	0	0	0	0	0	0	0
703	1442-3112(1)	1442-3112	5559	1029400	30	-1029370	0	0	0	0	0	0	0	0
651	1092-1103(1)	1092-1103	5372	833570	13	-833557	0	0	0	0	0	0	0	0
667	1103-1104(1)	1103-1104	5372	833570	13	-833557	0	0	0	0	0	0	0	0
669	1104-1113(1)	1104-1113	5372	833570	13	-833557	0	0	0	0	0	0	0	0
634	1082-1096(1)	1082-1096	5253	587540	18	-587522	0	0	0	0	0	0	0	0
681	1114-1442(1)	1114-1442	5080	891730	19	-891711	0	0	0	0	0	0	0	0
770	3241-1871(1)	3241-1871	4841	3537500	61	-3537439	0	0	0	0	0	0	0	0
631	1032-1244(1)	1032-1244	4819	496070	22	-496048	0	0	0	0	0	0	0	0
716	1851-3122(1)	1851-3122	4711	3706400	44	-3706356	0	0	0	0	0	0	0	0
657	1096-1092(1)	1096-1092	4382	526760	40531	-486229	-36179	-35886	-38241	-39960	-40021	-39985	-40210	-40237
690	1142-1032(1)	1142-1032	4371	452580	86	-452494	0	0	0	-1	-2	-2	-1	-1
646	1086-1082(1)	1086-1082	4345	463410	16	-463394	0	0	0	0	0	0	0	0
746	3121-1852(1)	3121-1852	4338	1069300	195	-1069105	0	0	-1	-2	-4	-6	-4	-4
721	1872-3242(1)	1872-3242	4326	1141500	41	-1141459	0	0	0	0	0	0	0	0
377	112-3102(1)	112-3102	4205	749150	119840	-629310	-108188	-111522	-110806	-112466	-112562	-112586	-112645	-112677
720	1871-1875(1)	1871-1875	4205	1677100	24	-1677076	0	0	0	0	0	0	0	0
645	1085-1241(1)	1085-1241	4151	545260	19	-545241	0	0	0	0	-1	-1	-1	-1
730	1875-1851(1)	1875-1851	3989	2685900	30	-2685870	0	0	0	0	0	0	0	0
745	3111-1443(1)	3111-1443	3847	941040	27	-941013	0	0	0	0	0	0	-1	0
679	1113-1114(1)	1113-1114	3800	678460	16	-678444	0	0	0	0	0	0	0	0
695	1244-1242(1)	1244-1242	3797	402390	19	-402371	0	0	0	0	0	0	0	0
663	1098-129(1)	1098-129	3773	394260	203810	-190450	-162405	-162985	-171454	-175024	-175568	-176081	-176463	-176516

As further exploration of the effect of additional budget, an experiment was performed with a very large budget of $25,000,000. This is 100 times larger than the critical budget of $250,000. The solution in this case includes 1963 aggregated link count sensors and 3776 classified link count sensors, far more than the number of candidate locations in the network (304). On average, more than 18 sensors are implemented at each location. The very large budget is expended by duplicating sensors. As discussed above, this results in a small decrease in variances for estimated O-D volumes and link flows, but the overall effectiveness is low. With this very large budget, the resulting Z^+ is 899,510, only 5.7% lower than the value achieved at a budget of $250,000. This is mainly because a lot of the links and O-D pairs related to PIP have relatively large variances, contributing to a large Z^+. However, those links are not eligible for sensor deployment.

The remaining relatively large variances are primarily associated with the internal centroid connector links in the network. These are not considered candidate links for sensors because they are "virtual" links rather than single physical facilities, so the sensor location model cannot operate to reduce the variances on these links. Two useful practical conclusions can be drawn from this experiment. First, a mechanism of sensing total trip ends in a zone (analogous to sensing volume on a centroid connector) would be useful in reducing $tr\left(S_Q^+\right)$. This could stimulate useful thought about incorporating a different type of sensor information. Second, it probably is useful to exclude centroid connectors when computing $tr(S_V^+)$. They are not "real"

links, but are introduced into the network for modeling purposes, and $tr(S_V^+)$ should probably be computed only using actual physical links.

For further analysis, the focus is on the results achieved at budgets of $250,000 and below. Figure 6- 7 shows how many sensors are implemented of each type under different budgets. As the budget increases, more sensors of each type are implemented. However, the number of classified sensors implemented increases faster than that of the aggregated sensors. It is probably because automobiles far outnumber trucks, contributing to the majority of the uncertainty in both link volumes and O-D volumes. The aggregated link count can be an acceptable rough estimate for automobiles when budget is scarce. Under a small budget like $100,000, much of the budget is spent on relatively inexpensive aggregated link count sensors to cover as many O-D pairs and links as possible. As the budget increases, more money can be spent on classified sensors to address the uncertainty in truck volumes, especially on route 80 and 59 where truck volumes are high.

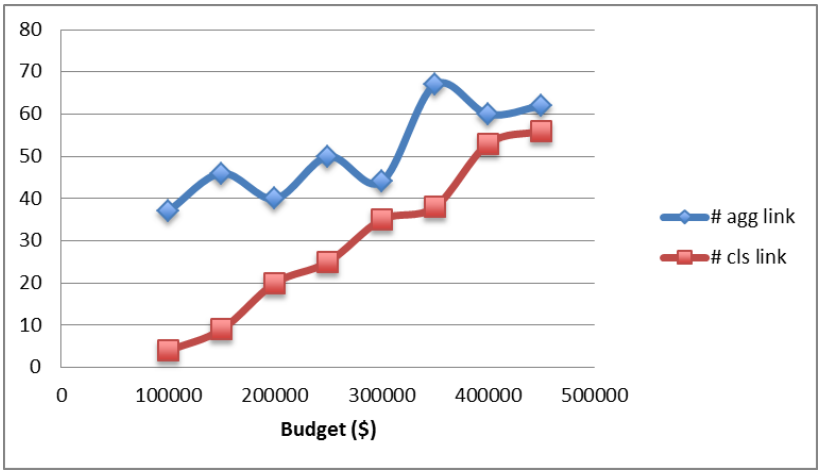


Figure 6- 7: Number of sensors implemented under different budgets

We are interested in where the classified sensors are implemented. Therefore, we pick the top 10 classified link count sensors that are most frequently chosen among all eight cases of different budget (see Table 6- 6). As we can see, those sensors are implemented mostly on links with high truck volumes and low implementation cost (i.e. links with only one lane).

Table 6- 6: Top 10 frequently chosen classified sensors

Index	Type	Location	Cost	Auto volume	Truck volume	Total volume	Frequency
330	2	3201-171	4550	466	220	686	7
435	2	59-60	4550	897	265	1162	6
329	2	171-3202	4550	483	223	706	6
369	2	61-70	9100	1903	336	2239	5
422	2	61-60	4550	656	323	979	5
482	2	33-30	4550	1034	239	1273	5
331	2	3181-170	4550	281	214	495	5
519	2	55-56	4550	1336	207	1542	5
334	2	135-3182	4550	419	182	601	5
421	2	60-61	4550	942	324	1267	4

Figure 6- 8 shows the selected locations and sensor types for the budget level of $250,000. Altogether, 50 aggregated link count sensors and 25 classified link count sensors are implemented. Primary areas for sensor deployment are:

- the area around zones 305 and 325 in the northeast part of the network (the villages of Haverstraw and West Haverstraw)
- US highway 202 (links connecting nodes 25, 26, 27 and 30)
- State highway 45 (links connecting nodes 25, 42, 56, 79, 97, 107 and 123)

- State highway 59 (links connecting nodes 115-120)
- County route 74 (links connecting nodes 74-82)
- County route 80 (links connecting nodes 55-61) and
- connecting links to the external zones 314, 317, 318 and 320 at the southern edge of the network (the border with New Jersey).

We also compared the solution for a budget of $100,000 to the solution for the budget of $250,000 to see how the sensor allocation changes as the budget increases. The solution for the budget of $100,000 obviously deploys fewer sensors (and especially fewer classified sensors), but in general focuses on similar areas of the network, as shown in Figure 6- 9. Only 23 sensors are exactly the same in the two solutions, but in the larger budget solution classified sensors are often substituted for aggregate sensors in similar locations. As the budget increases, sensor density along the main corridors in the network also increases.

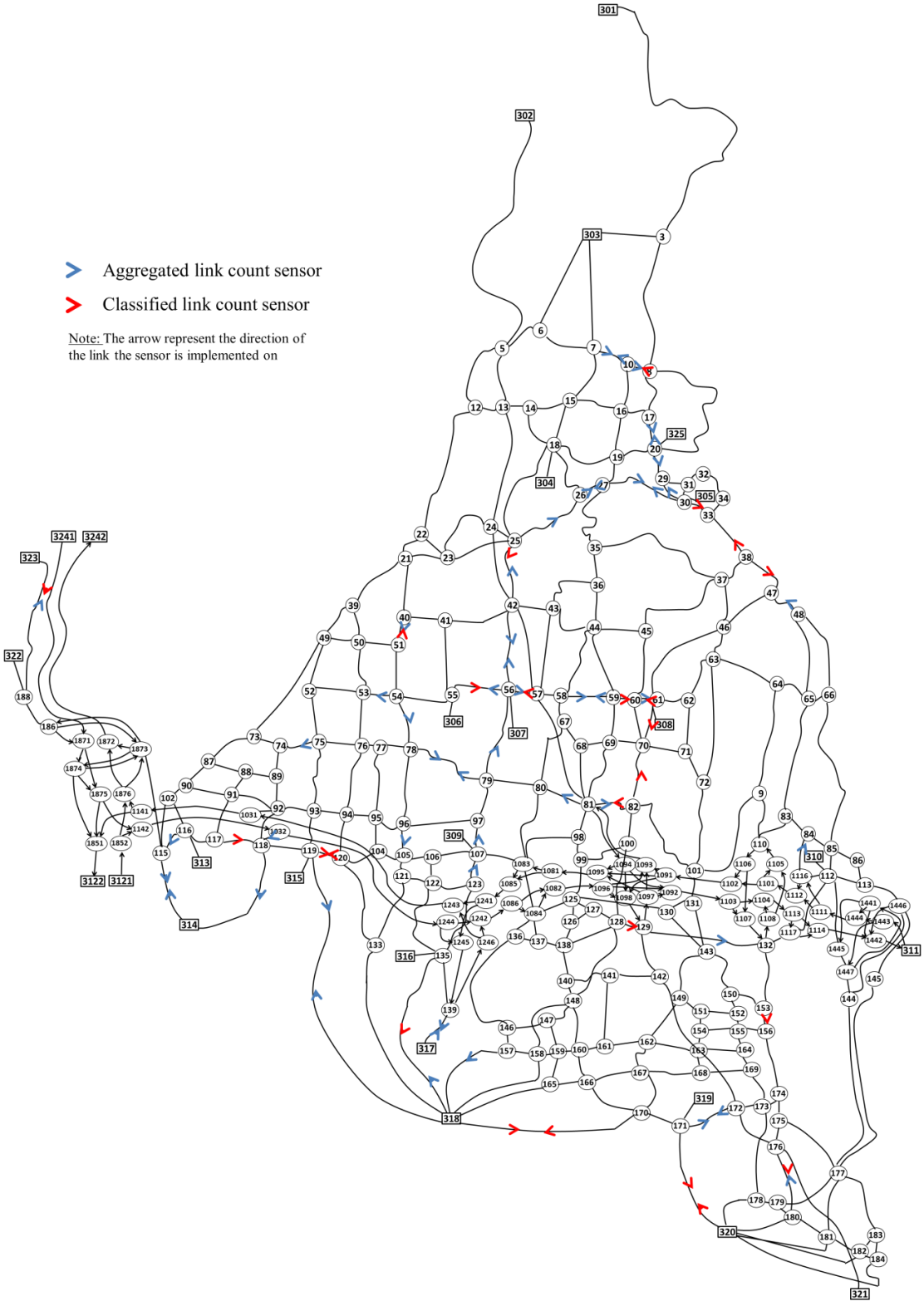


Figure 6- 8: Sensor allocation for budget of $250,000

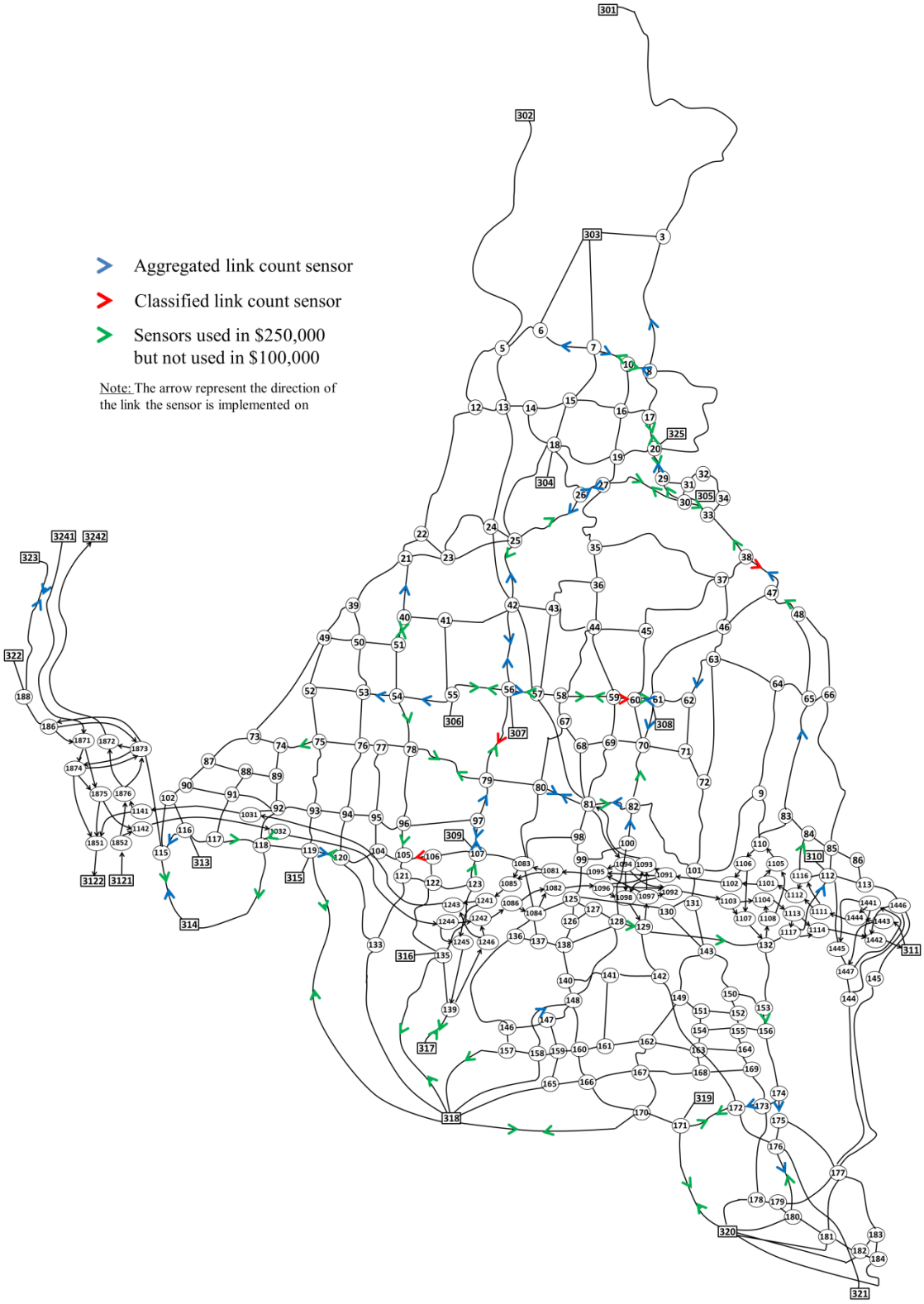


Figure 6- 9: Sensor allocation for budget of $100,000

6.4.2 Including Video Sensors at Intersections

We now consider surveillance cameras at intersections as a potential sensor type. Together with image processing software, they can provide classified turning movements at the candidate intersections shown in Figure 6- 4. This data may be quite valuable, but the cameras are considerably more expensive than link counters. In this experiment, it is assumed surveillance at an intersection results in observability of all movements at the intersection (through movements and turns from each approach), and that the video processing software associated with the sensor can distinguish vehicle classes for all movements.

With the cameras added to the possible sensor set and the budget set at $250,000, the result of the optimization is shown in Figure 6- 10. Altogether, 22 aggregated link count sensors, 16 classified link count sensors, and 8 cameras have been chosen. Comparing Figure 6- 8 and Figure 6- 10, we see that cameras are used in places where originally many link count sensors are allocated. They are used to substitute for those link count sensors in key areas of the network. After cameras are added, the final Z^+ has been reduced by approximately 10%, with $tr(S_Q^+)$ reduced by 7% and $tr(S_V^+)$ reduced by 13%. Thus, availability of the video sensors leads to a superior overall solution and the sensor location model developed here incorporates them into the process of deploying sensors across the network in a seamless fashion.

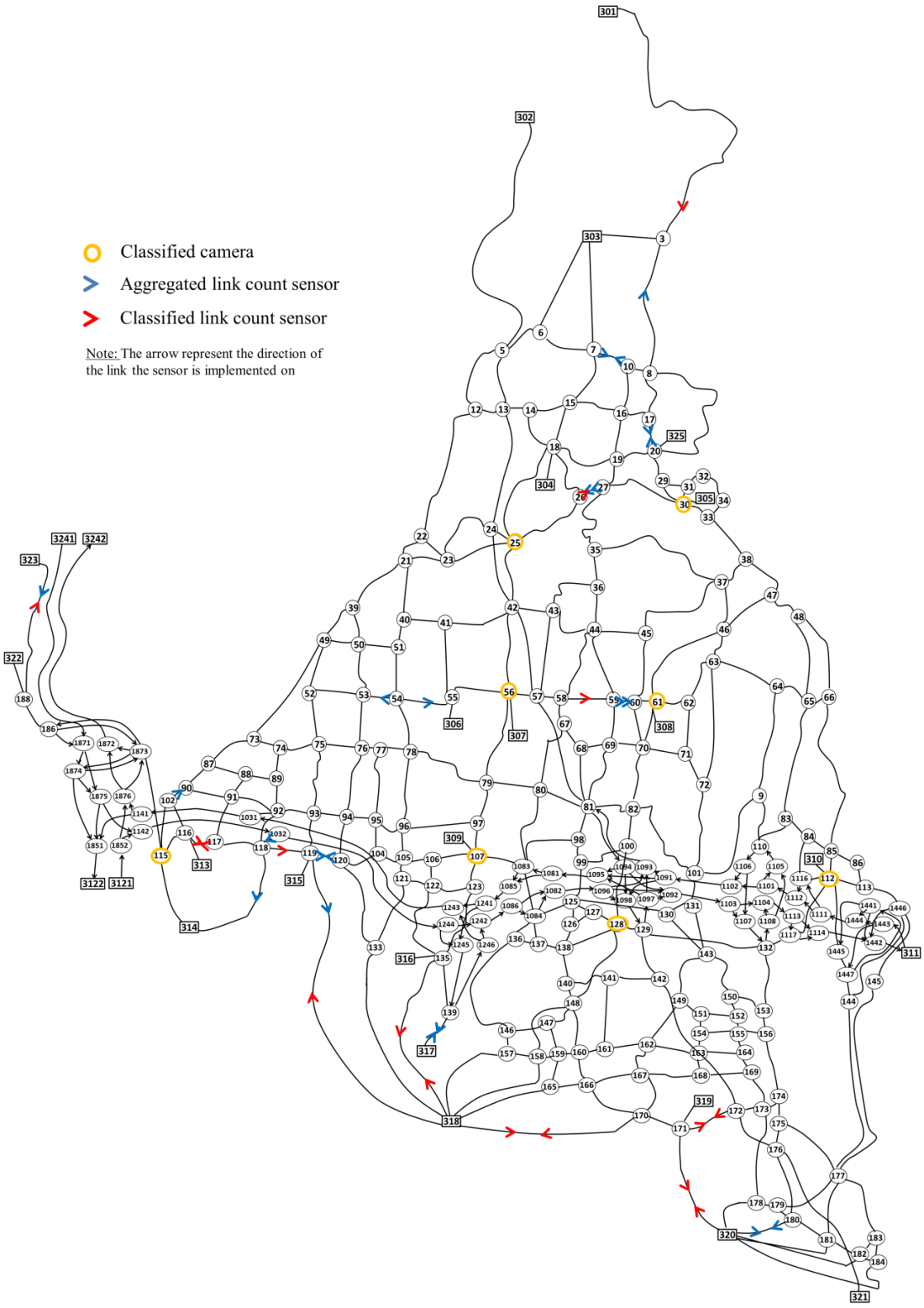


Figure 6- 10: Sensor allocation for budget of $250,000 with cameras added

6.5 Summary

The Rockland County test case demonstrates that the sensor location method developed in this research can successfully allocate sensors in realistic networks, and thus has significant practical value. It also demonstrates that the addition of classified video sensors at intersections can increase the total information content of a solution without requiring extra money. The classified turning movements provided by cameras have considerable value and should be considered more in practice.

CHAPTER 7

CONCLUSIONS

Effective traffic management in networks depends on the ability to sense and interpret volumes and flow patterns of vehicles using the network. Traffic management needs to become more effective to deal with increasing levels of congestion, emission control, special concern with truck movements, increasing interest in pricing policies for use of the network, and the potential for a fundamental change in how the roadway network is funded, changing from a fuel-tax based system to a system based on vehicle-miles-traveled (VMT).

Modern traffic sensing technology offers increasing ability to classify vehicles as they are counted, as well as to create data that are more informative than simple link counts, including output from video detectors, GPS-based vehicle location systems, automatic vehicle identification (AVI) systems, etc. Good estimates of link volumes (including links that may not be observed directly) are important for evaluating speeds and travel times, total emissions, VMT, etc. Estimating O-D flows (by vehicle class) provides a more complete picture of demand on the network and the basis for evaluating possible responses to traffic management strategies. Path-based information provides a connection between link-based data and O-D based demand, and is also important for traffic management assessment.

This dissertation establishes a new sensor location model that focuses on multiple vehicle classes and an objective that includes both link-based and O-D based flow estimation. It includes capability to locate a variety of sensor types in an

integrated way to maximize the information content of the entire set of sensors. The model formulation has the form of a nonlinear knapsack problem. An effective solution method is developed using a two-phase process. In the first phase, a greedy algorithm creates a good starting point for the second phase. The second phase is a Tabu Search algorithm that seeks improvement in the solution by swapping sensors while maintaining the overall budget constraint on the solution.

Extensive computational experiments have been performed on a small test network with 24 nodes and 76 links. These tests verify the effectiveness of the problem formulation and solution algorithm. They also indicate some important aspects of sensor deployment. When the sensor budget is quite small, the need for spatial coverage in the network is dominant and the sensors chosen are mostly the least expensive ones – simple aggregate link flow counters. As the available budget increases, classified link count sensors and classified cameras are more frequently chosen to address the need to construct flow estimates separated by vehicle class. Concentration of sensor allocation also happens around nodes with concentrated activity (e.g. trip ends or link flows by all vehicle classes) at any budget level. However, as the available budget increases, the types of sensors deployed around that node change and allow much better resolution of the various vehicle class O-D volumes.

The solutions across different levels of available budget also show the importance of an integrated solution using multiple sensor types. As the weight in the objective function changes and more emphasis is put on link volume variances in a multi-class context, the number of aggregated link count sensors selected is smaller,

and more emphasis is placed on obtaining vehicle classification information. The sensitivity analysis on utilization coefficients shows that the coefficients generated using free-flow speeds and travel times in the network are a good substitute for actual utilization coefficients. The information content of the resulting solutions is quite similar.

A case study on Rockland County, NY demonstrates that the sensor location method developed in this dissertation can successfully allocate sensors in realistic networks, and thus has significant practical value. It also demonstrates that the addition of classified video sensors at intersections can increase the total information content of a solution without requiring extra money. The classified turning movements provided by cameras have considerable value and should be considered more in practice.

The research in this dissertation contributes an important capability for locating sensors in a multiclass environment. This is likely to be of particular interest in urban areas where there is a desire to better understand differences between auto movements and truck movements so that more effective policies on emissions, pavement maintenance and energy consumption can be designed. In the experiments conducted here, equal weight has been placed on all vehicle classes. This is consistent with a measure of overall information content from the set of sensors, but in some applications there may be special interest in locating sensors to understand truck movements. This can be accommodated within the framework established here, by introducing another set of vehicle class weights in the evaluation of the trace values from the O-D volume and link flow covariance matrices. Further exploration of this

possibility, and the impacts on sensor type and location choices, is certainly worthwhile.

There might also be useful effort in simulating the ability of different sensor deployment strategies to support OD and link flow estimation. By simulating the observations, estimating the OD volumes and estimating the link flows in a test bed where the underlying answers are known, we are able to test whether there is a significant correlation between the objective function and actual ability to estimate those matrices and flows. There are also additional sensor types and opportunities that can be incorporated into the model formulation, but which have not been included in the computational experiments conducted here. Observations from AVI-equipped vehicles in the traffic stream or GPS-related data for specific vehicles are two types of data that have not been included in the experiments done here.

A useful extension of the core model in future work is to consider route flows more directly in the overall objective. Route flows represent an intermediate construct between the link flows and O-D flows considered here. Building variances on route flow estimates directly into the model is likely to be useful for incorporating path-oriented data (like AVI and GPS data) into the model framework.

There is also opportunity for use of cell-phone based data in understanding network flows. Such data are more aggregated geographically than the types of observations considered in this dissertation, but they can fit into the general information-theoretic framework used here. This presents another opportunity for future research and development efforts.

APPENDIX

CONSTRUCTION OF SENSOR ERROR COVARIANCE MATRICES

As discussed in chapter 3, each sensor is characterized by a delta precision matrix, Δ_d, that reflects its effect on the precision matrix for the O-D volumes in the network. The matrix Δ_d is calculated using Equation A-1.

$$\Delta_d = \left(h_d\right)^T R_d^{-1} h_d \qquad \textbf{Equation A-1}$$

The h_d matrix is constructed from the link utilization coefficients, as described in the example in Chapter 3. The R_d matrix is the covariance matrix for the errors in the observations from sensor d. The purpose of this appendix is to describe a process for constructing R_d.

For this purpose, the 9-node network used as an example in Chapter 3 is used again. Figure A- 1 illustrates the network. Four O-D pairs are assumed to be of interest in this network: 1-6, 1-9, 4-3 and 4-9. Three vehicle classes are assumed – automobiles and two truck classes. The first truck class is two-axle, six-tire medium trucks. The second class includes all heavier trucks. Light trucks are included with the automobiles.

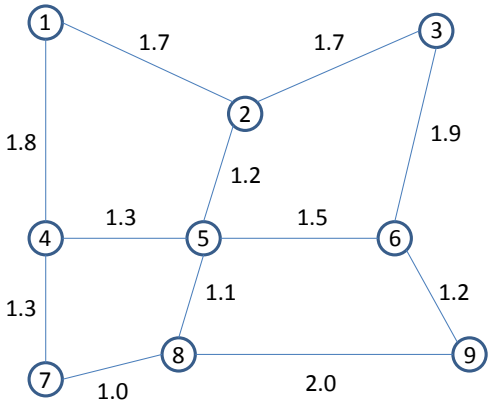


Figure A- 1: Example network.

We consider four types of potential sensors: aggregated link count sensors, classified link count sensors, aggregated intersection surveillance and classified intersection surveillance. When an intersection is instrumented, we assume that all the movements (through and turns) from all approaches at that node are recorded. This may involve use of multiple cameras. The basic information on the accuracy of observations from the sensors is shown in Table A- 1.

Table A- 1: Sensor types used for illustration

Index	Location type	# class	Counting error	Overcount ratio	Classification error
1	Link	1	2%	50%	0%
2	Link	3	2%	50%	5%
3	Intersection	1	2%	50%	0%
4	Intersection	3	2%	50%	9%

The third column of the table is the number of vehicle classes these sensors can differentiate. The counting errors are the percentage of time when an error record occurs. We assumed two types of counting error, overcounting and undercounting. In the case of overcounting, a vehicle is recorded when there is actually no vehicle present. This can generally happen during heavy traffic, where a lot of stop-and-go

happens, and a vehicle can be double counted. In the case of undercounting, a count is missed when a vehicle passes a sensor. This can happen with small vehicles, vehicles changing lanes, etc. Since different sensors have different error patterns, we do not make specific assumptions here regarding the vehicle classes associated with overcounting and undercounting. But those assumptions can be easily added to our framework if needed. The overcounting ratio is the percentage of counting error that is an overcount. The rest will be undercount. The classification error is the percentage of time when an observed vehicle is misclassified into a wrong category.

The R_d matrix (covariance matrix for errors in the observations) for different sensors can be derived through simulation. Each record generated is an observation (or lack of observation in the case of undercounting) of a vehicle. We simulate an hour's worth of traffic to calculate the R_d matrix. Since we don't know the actual traffic volume on each link before the sensor installation, we used the link capacity as an upper bound estimate.

Consider a classified link count sensor as an example. Each simulated entry contains three flags (whether it is a counting error, whether it is an overcount or undercount, whether it is a classification error) and two fields (true vehicle class and observed vehicle class). To generate an entry, we first generate two random variables to decide whether this record is a counting error or not, and if it is a counting error, whether it is an overcount or undercount. An overcount entry is automatically flagged as a classification error and the true vehicle class is marked as 0. The observed class is generated randomly from the distribution of vehicle class market share. An undercount entry is assumed not to be a classification error since the count does not

actually show up. The observed vehicle class is marked as 0. The true class is randomly generated using vehicle class market share just as for non-counting-error entries.

The vehicle class market share is the percentage of time that a vehicle observed belongs to a certain vehicle class. It is estimated from total travel demand and is assumed to be the same across all links. These data can also be easily derived from some manual traffic counts sampled across the network, if needed.

For records that are not counting errors, we then use a third random variable to decide whether an entry is a classification error or not. If it is not a classification error, the observed class of this entry is assumed to be the same as the true vehicle class. Otherwise we make the assumption that the observed vehicle class is one vehicle class away from the true vehicle class. That is, the (erroneous) observed class can only be 2 if the true vehicle class is either 1 or 3. If the true vehicle class is 2, we assume a 50% chance it will be misclassified as vehicle class 1 and 50% chance to be misclassified as vehicle class 3. All these assumptions can be modified in a real situation depending on the specific sensors under consideration.

We summarize the result in a confusion matrix (see Table A- 2) and compare the total number of vehicles of a certain vehicle class in the observation and true situation (see Table A- 3). The error column in Table A- 3 gives one sample of observation errors. We repeat the process multiple times and then compute the sample covariance matrix from all the samples as an estimate of the R_d matrix.

Table A- 2: Example confusion matrix of one trial of classified link counter simulation

True\Observed	Class 0	Class 1	Class 2	Class 3	Sum

Class 0	0	9	0	1	10
Class 1	6	988	65	0	1059
Class 2	1	3	58	2	64
Class 3	0	0	2	65	67
Sum	7	1000	125	68	

Table A- 3: Example simulation result summary of one trial

Vehicle class	True count	Observed count	Error
1	1059	1000	-59
2	64	125	61
3	67	68	1

The sampling distribution of a sample covariance matrix is called the Wishart distribution. It is defined as the sum of independent products of multivariate normal random vectors. Johnson and Wichern (2007) have shown that in the case of large sample, the law of large numbers applies and it has a distribution that is nearly normal. In this case, samples are quite easy to generate, so it is possible to use large sample sizes without incurring large computation times.

Table A- 4 shows an example of observation errors for a sample of 1,000,000 trials. An example R_d matrix of observation errors is shown in Table A- 5. Note that for a sensor that records vehicle class (and makes classification errors), the covariances are negative. One of the important parts of this method for estimating the R_d matrix is that it accounts for the fact that errors from classified counts are correlated, and that information is then included in the computation of the Δ_d matrix for the sensor.

Table A- 4: Example observation errors for 1,000,000 trials

Trials\Error	Class 1	Class 2	Class 3
1	-59	61	1
2	-41	44	2
3	-44	55	0
4	-57	53	-2
5	-59	58	-1
…	…	…	…
999999	-53	48	-1
1000000	-42	48	0

Table A- 5: Example R_d matrix

R_d	Class 1	Class 2	Class 3
Class 1	72.74	-51.32	-0.10
Class 2	-51.32	57.57	-5.12
Class 3	-0.10	-5.12	6.78

A simplified version of the same process is used to construct a variance estimate for an aggregated link count sensor. Because an aggregated link counter produces only a single observed value, R_d is a scalar, rather than a matrix and there are no covariance terms. The simulation treats an aggregated link counter as a simplified version of classified link count sensor where the total number of vehicle classes is 1 and the classification error rate is 0.

The R_d matrix of an intersection surveillance sensor can be performed in a similar way. Consider a classified sensor at node 5 in the network as an example. There are 15 observations associated with this sensor (5 relevant movements for each of the three vehicle classes). The resulting R_d matrix is a 15 by 15 matrix (See Table A- 6). We assume that the errors made by the sensor are counting errors and vehicle classification errors, but not mis-recording turning movements. That is, if a vehicle is

recorded correctly, its actual movement (through, left turn, right turn) is also recorded properly. As a result of this assumption, the R_d matrix is block-diagonal. The submatrix concerning different observations of one turning movement can be derived in a similar way as a classified link counter. The capacity of a path used in the calculation is the smallest capacity of included links. Since all links in the 9-node network have the same capacity, the submatrices are all the same.

Table A- 6: Example R_d matrix of a classified camera at node 5

	2-5-6(1)	2-5-6(2)	2-5-6(3)	2-5-8(1)	2-5-8(2)	2-5-8(3)	4-5-2(1)	4-5-2(2)	4-5-2(3)	4-5-6(1)	4-5-6(2)	4-5-6(3)	4-5-8(1)	4-5-8(2)	4-5-8(3)
2-5-6(1)	72.74	-51.32	-0.1	0	0	0	0	0	0	0	0	0	0	0	0
2-5-6(2)	-51.32	57.57	-5.12	0	0	0	0	0	0	0	0	0	0	0	0
2-5-6(3)	-0.1	-5.12	6.78	0	0	0	0	0	0	0	0	0	0	0	0
2-5-8(1)	0	0	0	72.74	-51.32	-0.1	0	0	0	0	0	0	0	0	0
2-5-8(2)	0	0	0	-51.32	57.57	-5.12	0	0	0	0	0	0	0	0	0
2-5-8(3)	0	0	0	-0.1	-5.12	6.78	0	0	0	0	0	0	0	0	0
4-5-2(1)	0	0	0	0	0	0	72.74	-51.32	-0.1	0	0	0	0	0	0
4-5-2(2)	0	0	0	0	0	0	-51.32	57.57	-5.12	0	0	0	0	0	0
4-5-2(3)	0	0	0	0	0	0	-0.1	-5.12	6.78	0	0	0	0	0	0
4-5-6(1)	0	0	0	0	0	0	0	0	0	72.74	-51.32	-0.1	0	0	0
4-5-6(2)	0	0	0	0	0	0	0	0	0	-51.32	57.57	-5.12	0	0	0
4-5-6(3)	0	0	0	0	0	0	0	0	0	-0.1	-5.12	6.78	0	0	0
4-5-8(1)	0	0	0	0	0	0	0	0	0	0	0	0	72.74	-51.32	-0.1
4-5-8(2)	0	0	0	0	0	0	0	0	0	0	0	0	-51.32	57.57	-5.12
4-5-8(3)	0	0	0	0	0	0	0	0	0	0	0	0	-0.1	-5.12	6.78

REFERENCE

Bianco, L., Confessore, G., Reverberi, P., 2001. A network based model for traffic sensor location with implications on O/D matrix estimates. Transportation Science 35 (1), 50–60.

Bretthauer, K.M., Shetty, B., Syam, S., 1995. A branch and bound algorithm for integer quadratic knapsack problems. ORSA Journal on Computing 7, 109–116.

Bretthauer, K.M., Shetty, B., 2001. A pegging algorithm for the nonlinear resource allocation problem. Computers and Operations Research 29 (5) 505–527.

Bretthauer, K.M., Shetty, B., 2002 The nonlinear knapsack problem – algorithms and applications. European Journal of Operational Research 138 459–472.

Caprara, A., Pisinger, D., Toth, P., 1999. Exact solution of the quadratic knapsack problem. INFORMS Journal on Computing 11, 125–137.

Cipriani, E., Fusco, G., Gori, S., Petrelli, M., 2006. Heuristic methods for the optimal location of road traffic monitoring stations. In: IEEE Intelligent Transportation Systems Conference, pp. 1072–1077.

Eisenman, S. M., Fei, X., Zhou, X., Mahmassani, H. S., 2006. Number and location of sensors for real-time network traffic estimation and prediction: sensitivity analysis. Transportation Research Record, 1964, 253–259.

Fei,X., Mahmassani, H. S., 2011. Structural analysis of near-optimal sensor locations for a stochastic large-scale network. Transportation Research, Part C, 19, 440-453.

Fei,X., Mahmassani, H. S., Murray-Tuite, P., 2013 Vehicular network sensor placement optimization under uncertainty. Transportation Research, Part C, 29, 14–31.

Gallo, G., Hammer, P.L., Simeone, B., 1980. Quadratic knapsack problems. Mathematical Programming 12, 132–149.

Gentili, M., Mirchandani, P.B., 2005. Locating active sensors on traffic network. Annals of Operations Research 136 (1), 229–257.

Gentili, M., Mirchandani, P.B, 2012. Locating sensors on traffic networks: Models, challenges and research opportunities. Transportation Research, Part C, 24, 227-255.

Hochbaum, D.S., 1995. A nonlinear knapsack problem. Operations Research Letters 17, 103–110.

Hochbaum, D.S., 2007. Complexity and algorithms for nonlinear optimization problems. Annals of Operations Research 153: 257–296

Horst, R., Tuy, H., 1990. Global Optimization: Deterministic Approaches. Springer, Berlin.

Hu, S., Liou, H., 2014. A generalized sensor location model for the estimation of network origin–destination matrices. Transportation Research, Part C, 40, 93–110

Jahangiri, E., Ghassemi-Tari, F., 2006. A dynamic programming approach for solving nonlinear knapsack problems. Journal of Industrial Engineering International 2(1), 31 – 37

Johnson, R. A., Wichern, D. W.2007. Applied multivariate statistical analysis, 6th ed. Prentice Hall, Upper Saddle River, N.J. 114-119, 173-176

Klein, L. A., Mills, M. K., Gibson, D. R. P., 2006. Traffic Detector Handbook: Third Edition (No. FHWA-HRT-06-108). Retrieved from: http://www.fhwa.dot.gov/publications/research/operations/its/06108/index.cfm

Kodialam, M.S., Luss, H., 1998. Algorithms for separable nonlinear resource allocation problems. Operations Research 46, 272–284.

LeBlanc, L.J., Morlok, E.K., Pierskalla, W., 1975. An efficient approach to solving the road network equilibrium traffic assignment problem, *Transportation Research*, 9:5, 309-318.

Li, D., Sun, X.L., Wang, J., McKinnon, K.I.M., 2009. Convergent Lagrangian and domain cut method for nonlinear knapsack problems. Computational Optimization and Applications 42: 67–104

Li, X., Ouyang, Y., 2011. Reliable sensor deployment for network traffic surveillance. Transportation Research, Part B, 45, 218–231

Lu, X., Huang, H., Long, J., 2013. Camera location optimisation for traffic surveillance in urban road networks with multiple user classes. International journal of systems science 44 (12), 2211-2222.

Ma, Guang-Ying, Li, Ping, Yao, Yun long, 2006. Research on location of traffic counting points for estimating origin-destination matrix. In: 6th International Conference on ITS Telecommunications, pp. 1216–1219.

Middleton, D., Charara, H., Longmire, R., 2009. Alternative vehicle detection technologies for traffic signal systems: technical report (No. FHWA/TX-09/0-5845-1). Retrieved from: http://tti.tamu.edu/documents/0-5845-1.pdf

New York State Department of Transportation (NYSDOT), 2015. Functional Classification. Retrieved July 9th, 2015, from https://www.dot.ny.gov/divisions/engineering/technical-services/highway-data-services/functional-class-maps

Powell, W.B., George, A., Bouzaiene-Ayari, B., Simao, H.P., 2005. Approximate dynamic programming for high dimensional resource allocation problems. Proceedings of the IJCNN, Montreal.

Rabiu, H., 2013. Vehicle detection and classification for cluttered urban intersection. International Journal of Computer Science, Engineering and Applications 3(1), 37–47.

Romeijn, H.E., Geunes, J., Taaffe, K., 2007. On a nonseparable convexmax imization problem with continuous knapsack constraints. Operations Research Letters 35, 172 – 180

Sayyady, F., Fathi, Y., List, G.F., Stone, J.R., 2013. Locating traffic sensors on a highway network models and algorithms. Transportation Research Record 2339, 30–38.

Sharkey, T.C., Romeijn, H. E., Geunes, J., 2011. A class of nonlinear nonseparable continuous knapsack and multiple-choice knapsack problems. Mathematical Programming A 126:69–96

Sherali, H., Desai, J., Rakha, H., El-Shawarby, I., 2006. A discrete optimization approach for locating automatic vehicle identification readers for the provision of roadway travel times. Transportation Research Part B 40 (10), 857–871.

Taghvaeeyan, S., Rajamani, R., 2014. Portable roadside sensors for vehicle counting, classification, and speed measurement. IEEE Transactions on Intelligent Transportation Systems 15(1), 73–83.

Wang, N., Gentili, M., Mirchandani, P., 2012. Model to locate sensors for estimation of static origin–destination volumes given prior flow information. Transportation Research Record 2283, 67–73.

Yang, H., Iida, Y., Sasaki, T., 1991. An analysis of the reliability of an origin-destination trip matrix estimated from traffic counts. Transportation Research Part B 25, 351–363.

Yang, H., Zhou, J., 1998. Optimal traffic counting locations for origin–destination matrix estimation. Transportation Research Part B 32, 109–126.

Yang, H., Yang, C., Gan, L., 2006. Models and algorithms for the screen line-based traffic counting location problem. Computers and Operations Research 33, 836–858.

Yim, P., Lam, W., 1998. Evaluation of count location selection methods for estimation of O–D matrices. Journal of Transportation Engineering 124(4), 376–383.

Zhang, G., Avery, R. P., Wang, Y., 2007. Video-based vehicle detection and classification system for real-time traffic data collection using uncalibrated video cameras. Transportation Research Record 1993, 138–147.

Zhao, Q., 2013. Multiclass origin-destination estimation using multiple data types (Master's thesis). Retrieved from https://newcatalog.library.cornell.edu/catalog/8267567

Zhou, X., List, G.F., 2010. An information-theoretic sensor location model for traffic origin-destination demand estimation applications, Transportation Science, 44:2, 254-273.