


**University of Alberta**

**GIS Based Facility Location Planning with Different Types of Consumers**

by

**Weiping Zeng** 

A thesis submitted to the Faculty of Graduate Studies and Research  
in partial fulfillment of the requirements for the degree of

**Doctor of Philosophy**

Department of Earth & Atmospheric Sciences

Edmonton, Alberta, Canada

**Fall 2007**



Library and  
Archives Canada

Bibliothèque et  
Archives Canada

Published Heritage  
Branch

Direction du  
Patrimoine de l'édition

395 Wellington Street  
Ottawa ON K1A 0N4  
Canada

395, rue Wellington  
Ottawa ON K1A 0N4  
Canada

*Your file* *Votre référence*  
*ISBN: 978-0-494-33097-5*  
*Our file* *Notre référence*  
*ISBN: 978-0-494-33097-5*

#### NOTICE:

The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

#### AVIS:

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protègent cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

---

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

  
**Canada**

## **Abstract**

This dissertation integrates geographic information systems (GIS), optimization modeling, aggregation, and heuristic methodologies to study facility location planning on a network with different types of consumers.

Traditional flow-interception location models (FILM) locate facilities to intercept as much traffic as possible, without considering where. Chapter Two develops a model that accounts for consumer desires to receive services at or near specific locations along their trips.

Location researchers tend to introduce changes in objective functions or assumptions by developing new models, hampering the development of standardized software that would encourage widespread use of flow interception models. Chapter Three formulates a generalized model encapsulating all known and many proposed FILM problems through simple data, or occasionally, constraint changes.

Traditional location theory views consumers as travelling from fixed points; their convenience is measured by distance from these points to the nearest facility. FILM theory views consumers as flows traveling on predetermined paths; their convenience is measured by distance from these paths to a facility. Chapter Four accommodates a more realistic view of consumers, that they choose a facility based on its greater convenience to either their home or their travel path, substantially improving the location modeling outcome. Location researchers have traditionally developed different models for different consumer types. Chapter Four further develops a strategy for unifying consumer types and location models. A generalized

model encompasses at least sixty existing models, including the  $p$ -median model, the maximal covering location model, and FILM.

Flow databases are often too large for location models to handle. Chapter Five integrates GIS, optimization, and heuristics to develop a system of efficiently aggregating flow data for location models. We apply this system to the classic FILM model using 2001 Edmonton afternoon peak traffic data and find it to be effective and free of aggregation error.

## **Acknowledgments**

My first acknowledgement goes to my supervisor Dr. M. John Hodgson for his patience, guidance and encouragement during the preparation and writing of this dissertation. I am very grateful to Dr. Ignacio Castillo, from the School of Business and Economics at Wilfrid Laurier University, for providing co-supervisor service throughout this study. I greatly appreciate Dr. Charles Robin Lindsey, my supervisory committee member, from the Department of Economics at University of Alberta, for his careful reading and valuable insights on this dissertation. I acknowledge Dr. Armann Ingolfsson, my candidate and oral examiner, from the School of Business at University of Alberta for his valuable insights on this dissertation. I would also like to express my thanks to Dr. Michael Kuby, the external examiner, from the School of Geographical Sciences at Arizona State University for his helpful suggestions and comments on this dissertation. Thanks to the City of Edmonton Transportation and Streets Department, particularly Dr. Alan Brownlee and Ms. Lorraine Doblanko for providing Edmonton afternoon peak traffic data. I also acknowledge that this study was supported in part by the National Science and Engineering Research Council of Canada. Special appreciation must go to my parents, two brothers, and two sisters for their endless support and encouragement. Finally, I thank Dr. Martin Sharp, Dr. Philippe Erdmer, Dr. Edgar Jackson, Dr. Di Zhou, Ms. Carol Cooper, Ms. Kimberly Arndt, Ms. Marsha Boyd, and all the people who have helped and supported me over the length of this study.

## Table of Contents

<b>Chapter 1: Introduction</b>	1-6
References	4
<b>Chapter 2: The Pickup Problem: Consumers' Locational Preferences in Flow Interception</b>	7-29
2.1. Introduction	8
2.2. Background	9
2.2.1 The Flow-Interception Location Model	9
2.2.2 The Pickup Problem	10
2.2.3 The Benefit Function	11
2.3. Model Formulation	11
2.4. Numerical Experimentation	13
2.4.1. Morning Peak Traffic Flows for Edmonton, 1989	13
2.4.2. Afternoon Peak Traffic Flows for Edmonton, 2001	16
2.5. Conclusions	19
References	20
<b>Chapter 3: A Generalized Model for Locating Facilities on a Network with Flow-Based Demand</b>	30-53
3.1. Introduction	31
3.2. The Flow-Interception Location Model	32
3.3. The Generalized Flow-Interception Location-Allocation Model	33
3.4. Special Cases of GFIM	34
3.4.1. The Basic Flow-Interception Case	34
3.4.2. Protection Cases	36
3.4.3. Generalized Preference Cases	37
3.4.4. Generalized Deviation Cases	37
3.4.4.1 Deviation Case 1	37
3.4.4.2 Deviation Case 2	38
3.4.4.3 Deviation Case 3	38
3.4.5. Generalized Capacity Cases	39
3.4.6. Combinations of Cases	39
3.4.7. Multi-Counting Cases	40
3.4.8. Group-Counting Cases	41
3.4.9. Other Potential Cases	42
3.5. Computational Experimentation Comparison	42
3.6. Conclusion	46
References	47
<b>Chapter 4: A New Type of Consumer and an Efficient Strategy for Unifying Network Location Models</b>	54-78
4.1. Introduction	55

4.2. A Generalized and Efficient Strategy for Unifying Consumer Types	56
4.3. A Generalized Location-Allocation Model	58
4.4. The Efficiency of GES in Unifying Current and Future Location Models	59
4.4.1. GES Enables GLAM to Solve Many Maximization Problems	60
4.4.1.1. Maximum Potential Service User Problems	60
4.4.1.2. Maximum Actual Service User Problems	63
4.4.1.3. Capacitated Maximization Problems	64
4.4.2. GES Enables GLAM to Solve Many Minimization Problems	65
4.4.3. Application of GES to Other Location Models	66
4.5. The Importance of Type C Consumers in Location Modeling	67
4.6. Conclusions	71
References	72

**Chapter 5: An Integrated GIS, Optimization, and Heuristic System for Flow-Based Demand Aggregation** 79-104

5.1. Introduction	80
5.2. Background	81
5.2.1. Flow-Interception Location Problems	81
5.2.2. Aggregation Errors	82
5.2.3. GIS, Optimization and Heuristics	83
5.3. A Case Study	84
5.3.1. Flow Paths	86
5.3.2. Potential Facility Sites	88
5.4. An Integrated Multitechnologies System for Flow demand Aggregation	89
5.5. Conclusion	90
References	91

**Chapter 6: Conclusions and Future Research** 105-107

References	107
------------	-----

**Appendices: Codes** 108-133

Appendix 1: Implementing FILM in AMPL CPLEX	108
Appendix 2: Implementing GFIM in AMPL CPLEX	109
Appendix 3: Calculate Shortest Paths in CPLEX	110
Appendix 4: ShortestPath.cpp	112
Appendix 5: CalculateInterceptedFlows.cpp	117
Appendix 6: UnionLink.cpp	119
Appendix 7: ChangeSolutionID.cpp	121
Appendix 8: CalculateSolutionFlows.cpp	123
Appendix 9: 1-opt interchange heuristic	125
Appendix 10: AggregatePaths.cpp	131

## List of Tables

Table 2-1: Flow and benefit obtained in each scenario ( $p = 3, \alpha = 0.002$ )	15
Table 2-2: Benefit % ( $p = 1 \dots 10, \alpha = 0.002$ )	15
Table 2-3: 27 important nodes	17
Table 2-4: Facility location movement, benefit, and flow as $\alpha$ increases ( $p = 5$ )	19
Table 3-1: OD flow paths	34
Table 3-2: Input matrix of G and output results for each case of GFIM	35
Table 3-3: Computational comparison (afternoon peak traffic, Edmonton)	45
Table 3-4: Computational comparison (morning peak traffic, Edmonton)	46
Table 4-1: Consumers, paths and distances	58
Table 4-2: Q and G in section 4.4.1	62
Table 4-3: Optimal solutions in section 4.4.1.1	62
Table 4-4: Q and G in section 4.4.1.2	64
Table 4-5: Optimal solutions in section 4.4.1.2	64
Table 4-6: Q and G in section 4.4.2	66
Table 4-7: Table 4-7: Optimal solutions in section 4.4.2	66
Table 4-8: Q in section 4.5	68
Table 4-9: Optimal solutions in section 4.5	69
Table 4-10: Consumers covered by each solution	69
Table 4-11: Superiority of SC	70
Table 4-12: The Solution Robustness	70
Table 5-1: Optimal Solutions of the Original FILM Problem	85
Table 5-2: Five Transportation Networks	86
Table 5-3: Computation time and errors ( $p = 1 \dots 20$ )	87



## List of Figures

Figure 2-1: The four pickup scenarios	22
Figure 2-2: Sample two-origin, one-destination trees	22
Figure 2-3: Optimal locations ( $p = 3$ , constant = 0.002)	23
Figure 2-4: Zoomed in optimal locations ( $p = 3$ , constant = 0.002)	24
Figure 2-5: "Coffee" optimal locations with increasing constant ( $p = 3$ )	25
Figure 2-6: Cost-effectiveness of the optimal solutions ( $1 \leq p \leq 148$ )	26
Figure 2-7(colour): Network flow structure (2001)	27
Figure 2-8(colour): Optimal locations ( $p = 5$ , constant = 0.0005)	28
Figure 2-9: Superiority of PUP	29
Figure 3-1: A test 7-node network	49
Figure 3-2: The GFIM model (GFIM.mod)	49
Figure 3-3: Data for the GFIM model (GFIM.dat)	49
Figure 3-4: 2001 afternoon peak traffic network (Edmonton, Alberta, Canada)	50
Figure 3-5: 1989 morning peak traffic network (Edmonton, Alberta, Canada)	51
Figure 3-6: CPU minutes of FILM and GFIM (Edmonton afternoon peak traffic)	52
Figure 3-7: CPU minutes of FILM and GFIM (Edmonton morning peak traffic)	53
Figure 4-1: A test 7-node network	75
Figure 4-2: GLAM.mod	75
Figure 4-3: GLAM.txt (example in section 4.1.1)	76
Figure 4-4: GLAM.run	76
Figure 4-5: Consumer patterns in scenarios {A} and {B}	77
Figure 4-6: Optimal solution at each scenario ( $p = 4$ )	78
Figure 5-1: CPU Minutes with CPLEX	93
Figure 5-2: Aggregation Errors in aggregated networks	94
Figure 5-3(colour): Removed zones at each network	95
Figure 5-4: Network flow structure (395 zones)	96
Figure 5-5: Removed network flow structure (network 1-->2)	97
Figure 5-6: Link flow structure (network 1)	98
Figure 5-7: Locations movement and network flow structure (network 2)	99
Figure 5-8: Removed network flow structure (Network 2-->3)	100
Figure 5-9: Network flow structure (network 3)	101
Figure 5-10: High flow network nodes	102
Figure 5-11: 270 high flow nodes	103
Figure 5-12: An integrated system for flow demand aggregation	104

## Chapter 1

### Introduction

Almost every enterprise in the private and public sectors faces the problem of strategically locating facilities to provide services to consumers on a transportation network. Industrial firms must identify locations for plants and warehouses that minimize total fixed costs and transportation costs. Retail outlets must locate stores that have geographical advantages. Government agencies must build public service facilities such as schools, hospitals, public libraries, post offices, bus stops, fire stations, vehicle inspection stations, and landfills in locations convenient to users. Location theory provides decision makers with strategic, analytical, and quantitative tools for seeking locations where fixed and operating costs can be kept low and accessibility to markets can be kept high. Due to the complexity, strategic importance, and widespread application of these tools, this field has attracted many researchers from the disciplines of operations research/management science, geography, economics, marketing, urban and regional planning, transportation planning, and engineering, among others.

Location theory has been of interest for a long time. The tradition of determining optimal location patterns for specific facilities goes back as far as Fermat (1601–1665), who put the problem on a mathematical basis, and continued through Weber (1909) who located industrial firms to minimize transportation costs. Other seminal location publications include: Von Thünen (1826), Hotelling (1929), Christaller (1933), Weiszfeld (1937), Cooper (1963), Hakimi (1964, 1965), Balinski (1965), ReVelle and Swain (1970), Toregas *et al.* (1971), and Church and ReVelle (1974). It is generally acknowledged that the roots of agricultural location theory can be traced back to 1826 when Von Thünen published his classic work *The Isolated State* (O’Kelly and Bryan 1996). His research uncovered laws that govern the interaction of agricultural prices, land uses and distance, as farmers seek to maximize profit. Hotelling (1929), a renowned economist, considered the problem of locating competing facilities through the simple example of two ice-cream vendors along a beach strip. Christaller (1933), a Germany geographer, introduced central place theory to explain how location influences the evolution of systems of cities and towns. Weiszfeld (1937) provided the simplest and most common used technique (called “Weiszfeld procedure”) to solve the Weber problem. Cooper (1963) introduced a simple location-allocation problem, which optimally locates service facilities and allocates demand to them. Hakimi (1964), an operations researcher, introduced and brought invaluable insight to the  $p$ -median problem on a network – minimizing the total (therefore average) distance that is traveled by those who utilize the facilities. Hakimi (1964, 1965) addressed the  $p$ -center problem which minimizes the maximal distance that is traveled by those who utilize the facilities. Balinski (1965) introduced the plant location problem which seeks the location of an unknown number of facilities so that the sum of manufacturing costs and delivery costs is a minimum. ReVelle and Swain (1970) first formulated the  $p$ -median problem as an integer linear

program. Toregas *et al.* (1971) introduced the set covering location problem which seeks the minimum number of facilities in a manner that all demand points are covered by at least one facility within a distance standard. Church and ReVelle (1974) introduced the maximal covering location problem which aims to locate  $p$  facilities in a manner that maximizes the number of covered consumers.

A network location problem may be characterized as the problem of identifying the placement of  $p$  facilities on a network to serve a spatially distributed set of demands in a manner that optimizes a designated objective function. Since the early 1960s, hundreds of network location models have been proposed in thousands of academic publications due to the ubiquity of locational decision-making, as well as the general availability of high-speed computers and efficient optimization methods. Brandeau and Chiu (1989) identified over 50 problem types in location theory appearing in over 40 different scholarly journals. Hale (2006) listed over 3400 location theory references. Several representative books edited by Ghosh and Rushton (1987), Mirchandani and Francis (1990), Dicken and Lloyd (1990), Daskin (1995), Drezner (1995) and Drezner and Hamacher (2002), provide a rich collection of papers on location theory. Scholarly journals that historically have played an important role in the area of facility location include *Annals of Operations Research*, *Computers and Operations Research*, *Environment and Planning*, *European Journal of Operational Research*, *Geographical Analysis*, *IIE Transactions*, *INFOR*, *Journal of Retailing*, *Journal of Retailing and Consumer Services*, *Journal of the Operational Research Society*, *Location Science*, *Management Science*, *Naval Research Logistics*, *Operations Research*, *Papers of the Regional Science Association*, *Socio-economic Planning Sciences*, *The Professional Geographer*, *The Annals of Regional Science*, *Transportation Research*, and *Transportation Science*.

Traditional location theory views consumers as travelling from static and fixed points (e.g., homes); their convenience is measured by distance from these points to the nearest facility (point-based demand). Since the early 1990s, there has been considerable research interest, represented by about 40 academic publications, in flow-interception location theory (e.g., Hodgson 1990; Berman, Larson, and Fouska 1992; Berman, Bertsimas, and Larson 1995), which views consumers as flows travelling on predetermined origin-destination (OD) paths (e.g., daily commute between home and workplace); their convenience is measured by distance from these paths to a facility (flow-based demand). Flow-interception theory has been applied to the strategic location of automatic teller machines and convenience stores (Berman, Hodgson, and Krass 1995; Hodgson, Rosing, and Storrier 1996; Wang, Batta, and Rump 2002; Turner 2006), advertising billboards (Averbakh and Berman 1996; Hodgson and Berman 1997), vehicle inspection stations (Hodgson, Rosing, and Zhang 1996; Gendreau, Laporte, and Parent 2000; Miller and Shaw 2001), park-and-ride facilities (Horner and Grove 2006), gasoline stations and refuelling facilities (Kuby and Lim 2005, 2007; Kuby 2006; Upchurch, Kuby, and Lim 2006), and cellular base stations (Erdemir *et al.* 2006). In addition, Berman, Bertsimas, and Larson (1995) developed several models to address generalizations of flow-interception problems where flows are allowed to deviate from predetermined origin-destination paths. The

reader is referred to Berman, Hodgson, and Krass (1995) and Hodgson (1998) for more detailed reviews of these models.

Facility location planning is a key decision in the long-term efficiency of operations. This dissertation integrates geographic information systems (GIS) such as ArcGIS, optimization modeling techniques such as AMPL/CPLEX, aggregation, and heuristic methodologies to study facility location planning on a network with different types of consumers. A brief description of each chapter is as follows.

Chapter 1: This chapter introduces the background of location theory and the objective of each chapter.

Chapter 2: The standard FILM problems implicitly assume that there is no indication of where in the journey the flows are intercepted, nor is there any impetus to prefer one location over another. However, for most real-world facilities (e.g., convenience stores, fast food outlets, gasoline and refueling stations), this assumption is tenuous because consumers often desire to obtain a product or service at or near a specific location along their trip, frequently at their trip origin or destination. I note that the classic flow-interception location model (FILM) (Hodgson 1990; Berman, Larson, and Fouska 1992) seeks to optimally locate service facilities, but does not explicitly consider the allocation of flow-based demand to facilities. That is to say that FILM does not consider *where* a consumer is served. Thus, in location theory, FILM is considered to be a *location* model rather than a *location-allocation* model. The implication of this observation is that FILM cannot directly take into account consumers' locational preferences. The objective of chapter 2 is to formulate a flow-interception location-*allocation* model for considering consumers' locational preferences. In other word, it focuses on shaping our understanding of geographical advantages and consumers' behaviours.

Chapter 3: Location researchers tend to introduce changes in objective function and/or assumptions by developing new models. Over 30 different flow-interception models spanning about 40 academic publications have been proposed during the past 17 years. This has created numerous disparate models, each viewed as requiring its own solution method, challenging the development of standardized software that would encourage widespread use of location models in real-world, strategic, decision-making processes. I note that the structure of most flow-interception models is similar. The objective of chapter 3 is to formulate a *generalized* flow-interception location-allocation model for solving different kinds of flow-interception problems using a single framework. I expect that this single, generalized framework will also ease the burden on flow-interception decision makers.

Chapter 4: Traditional network location theory assumes that consumers patronize facilities near demand points (Type A consumer). Flow-interception location theory assumes that consumers patronize facilities near predetermined paths (Type B consumer). In the real world, however, consumers often choose a facility based on its greater convenience to either their homes or their travel path. Chapter 4 calls these consumers Type C consumers. Most people in the real world are Type C consumers – they are not as selective of the actual location as Type A and Type B consumers. The literature has neglected Type C consumers. The major objective of chapter 4 is to

study the importance of Type C consumers in location analysis. Location researchers have traditionally proposed models for different types of consumers in isolation. This research progression has split flow-interception theory and traditional network location theory. Another major objective of this chapter is to develop a generalized and efficient strategy for unifying consumer types and associated models. In other words, chapter 4 focuses on satisfying consumers' diverse desires and needs while simultaneously easing the burden on location decision makers.

Chapter 5: Large-scale location problems often cannot be solved optimally. The volume of flow-based demand data grows very quickly as the number of origins and destinations increases, and even with the most efficient and specialized heuristics, good solutions to large flow-interception problems will be beyond the capability of the personal computer. The objective of chapter 5 is to solve large real-world flow-interception problems by aggregating flow-based demands using GIS. In other words, it aims to overcome the difficulties of applying flow-interception problems to real-world situations and reducing the burden on flow-interception decision makers.

Chapter 6: The most important contributions of each chapter and future directions are reviewed.

The body of this dissertation (chapters 2 through 5) is written with the intent to enable readers to read selected chapters without having to read the entire dissertation. A concise and factual summary (maximum length 300 words) at the beginning of each chapter is also able to stand alone. The chapters are presented in order of increasing complexity. Readers without a background in location theory are encouraged to go through them in numerical order.

## References

- Averbakh, I., O. Berman. 1996. Locating flow-capturing units on a network with multi-counting and diminishing returns to scale. *European Journal of Operational Research* **91** 495-506.
- Balinski, M.L. 1965. Integer programming: methods, uses, computation. *Management Science* **12** 253-313.
- Berman, O. 1997. Deterministic flow-demand location problems. *Journal of Operational Research Society* **48** 75-81.
- Berman, O., D. Bertsimas, R. C. Larson. 1995. Locating discretionary service facilities, II: maximizing market size, minimizing inconvenience. *Operations Research* **43** 623-632.
- Berman, O., M. J. Hodgson, D. Krass. 1995. Flow-interception problems. In *Facility Location: A Survey of Applications and Methods*, edited by Z. Drezner. Springer-Verlag, New York, 389-426.
- Berman, O., R. C. Larson, N. Fouska. 1992. Optimal location of discretionary service facilities. *Transportation Science* **26** 201-211.
- Brandeau, M.L., S.S. Chiu. 1989. An overview of representative problems in location research. *Management Science* **35** 645-674.
- Christaller, W. 1933. *Central place in southern Germany*, Translated in 1966, Prentice-Hall, NJ.

- Church, R., C. ReVelle. 1974. The maximal covering location problem. *Papers of the Regional Science Association* **32** 101-1018.
- Cooper, L. 1963. Location-allocation problems. *Operations Research* **11** 331-343.
- Daskin, M.S. 1995. *Network and discrete location*. John Wiley & Sons. Inc. 1-498.
- Dicken, P., P. E. Lloyd (1990). *Location in space the theoretical perspectives in Economic Geography*. Harper& Row, New York, 1-431.
- Drezner, Z. 1995. *Facility location: a survey of applications and methods*. Springer-Verlag, New York, 1-571.
- Drezner, Z., H. W. Hamacher. 2002. *Facility location: applications and theory*. Springer-Verlag, New York, 1-457.
- Erdemir, E. T., R. Batta, S. Spielman, P.A. Rogerson, A. Blatt, M. Flanigan. 2006. Location coverage models with demand originating from nodes and paths: application to cellular network design. Manuscript for *European Journal of Operational Research*.
- Gendreau, M., G. Laporte, I. Parent. 2000. Heuristics for the location of inspection stations on a network. *Naval Research Logistics* **47** 287-303.
- Ghosh, A., G. Rushton. 1987. *Spatial analysis and location allocation models*. NY, Van Nostrand Reinhold.
- Hakimi, S.L. 1964. Optimum locations of switching centres and the absolute centres and medians of a graph. *Operations Research* **12** 450-459.
- Hakimi, S.L. 1965. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. *Operations Research* **13** 462-475.
- Hale, T. S. 2006. Trevor Hale's location science references.  
<http://gator.dt.uh.edu/~halet/>
- Hillsman, E.L. 1984. The  $p$ -median structure as a unified linear model for location-allocation analysis. *Environment and Planning A* **16** 305-318.
- Hodgson, M. J. 1990. A flow-capturing location-allocation model. *Geographical Analysis* **22** 270-279.
- Hodgson, M. J. 1998. Developments in flow-based location-allocation models. In *Economic Advances in Spatial Modelling and Methodology: Essays in Honour of Jean Paelinck*, edited by D.A. Griffith, C.G. Amrhein, J-M Huriot, Kluwer Academic Publishers, 119-132.
- Hodgson, M. J., K. E. Rosing, A. L. G. Storrier. 1996. Applying the flow-capturing location-allocation model to an authentic network: Edmonton, Canada. *European Journal of Operational Research* **90** 427-443.
- Hodgson, M. J., K. E. Rosing, J. Zhang. 1996. Locating vehicle inspection stations to protect a transportation network. *Geographical Analysis* **28** 299-314.
- Hodgson, M. J., O. Berman. 1997. A billboard location model. *Geographical & Environmental Modeling* **1** 25-45.
- Horner, M. W., S. Groves. 2007. Network flows-based strategies for identifying rail park-and-ride facility locations. *Socio-Economic Planning Sciences* **41** 255-268.
- Hotelling, H. 1929. Stability in Competition. *The Economic Journal* **39** 41-57.
- Kuby, M. 2006. Prospects for geographical research on alternative-fuel vehicles.

- Journal of Transport Geography* **14**, 234-236.
- Kuby, M., S. Lim. 2005. The flow-refuelling location problem for alternative-fuel vehicles. *Socio-Economic Planning Sciences* **39** 125-145.
- Kuby, M., S. Lim. 2007. Location of alternative-fuel stations using the flow-refueling location model and dispersion of candidate sites on arcs. *Networks and Spatial Economics* **7** 129-152.
- Miller, H. J., S. Shaw. 2001. *Geographic Information Systems for Transportation*. New York, NY: Oxford University Press.
- Mirchandani, P.B., R. L. Francis. 1990. *Discrete location theory*. John Wiley & Sons. Inc. 1-549.
- O'Kelly, M., D. Bryan. 1996. Agricultural location theory: von Thünen's contribution to economic geography. *Progress in Human Geography* **20**, 457-475.
- ReVelle, C.S., R. Swain. 1970. Central facilities location. *Geographical Analysis* **2** 30-42.
- Schilling, D. A., V. Jayaraman, R. Barkhi. 1993. A review of covering problems in facility location. *Location Science* **1** 25-55.
- Toregas, C., R. Swain, C. ReVelle, and L. Bergman. 1971. The location of emergency service facilities. *Operations Research* **19** 1363-1373.
- Turner, D. 2006. Implementing the flow-interception location model with geographic information systems. Master Thesis, University of Texas at Dallas, USA.
- Upchurch, C., M., Kuby, S. Lim. 2006. A model for location of capacitated alternative-fuel stations. Forthcoming at *Geographical Analysis*.
- Wang, Q., R. Batta, C. M. Rump. 2002. Algorithms for a facility location problem with stochastic customer demand and immobile servers. *Annals of Operations Research* **111** 17-34.
- Weber, A. 1909. *Über den Standort der Industrien*, Tübingen, (English translation by Friedrich, C. J. (1929). *Theory of the location of industries*, University of Chicago Press.
- Weiszfeld, E. 1937. Sur le point pour lequel la somme des distances de  $n$  points donnés est minimum. *Tohoku Mathematical Journal* **43** 355-386.

## Chapter 2 \*

### The Pickup Problem: Consumers' Locational Preferences in Flow Interception

**Summary:** In this chapter, I address what I call the *pickup problem* wherein patrons briefly interrupt a predetermined journey to obtain a simple good or service, such as fast food or a video, and then resume their journey. This is a problem of the class known as flow-interception location problems. Traditional flow-interception models are used to select service locations such that the flows that are intercepted are maximized. I note that in these traditional models only flow quantities are considered and where in the journey the pickup is made is not considered. However, in the real world, consumers often wish to obtain a product or service at or near a specific location along their trip. In this chapter, I propose a pickup model (PUP) that considers consumers' locational preferences, providing a much broader, more realistic approach than FILM (a special case of PUP) to problems in the private and public sectors. By considering which patrons are served where, PUP transforms the flow-interception location model to a flow-interception location-allocation model, providing a fruitful garden for further research. I demonstrate and apply the PUP model to morning and afternoon peak traffic flows in Edmonton, Alberta, Canada. I integrate geographic information systems (GIS) and optimization engines to investigate the PUP model in real-world transportation systems. My numerical experimentation demonstrates that the optimal locations identified by traditional models arise solely from network flow structure, whereas the optimal locations identified by PUP result from the tradeoffs between the network flow structure and the importance of proximity to preferred locations. I discover that solutions of PUP are superior to those of traditional FILM if consumers have locational preferences. The up-to-date, real world transportation networks provide a realistic test-bed for this and other models of the flow-interception type.

\* A version of this chapter has been **accepted** for publication.

**Zeng, Weiping, M. John Hodgson, Ignacio Castillo.** 2007. The pickup problem: consumers' locational preferences in flow interception. Accepted on February 10, 2007 for publication in the special issue of *Geographical Analysis*, edited by John R. Current, Richard L. Church, and Mark S. Daskin (in memory of Charles S. ReVelle, a leader in location science)



## 2.1. Introduction

One of the most important ways an industrial firm, retail outlet, or government agency can enhance its chances of success is to identify a good location. One approach is to use location-allocation models that optimally locate service facilities and allocate demand to them according to a specific objective. Traditional location models such as the  $p$ -median model (ReVelle and Swain 1970) and the maximal covering location model (Church and ReVelle 1974) deal with demands expressed at fixed locations in the network (point-based demand). Demands for many services are, however, expressed by flows in a network. Since the early 90's, there has been considerable research interest, represented by over 30 published academic articles, in the flow-interception location model (FILM), in which demand is represented as flows traveling on origin-destination (OD) paths of a network (flow-based demand). The applications of flow-interception theory have covered the strategic location of automatic teller machines and convenience stores, advertising billboards, vehicle inspection stations, park-and-ride facilities, gasoline stations and refuelling facilities, and cellular base stations.

Standard flow-interception models involve the placement of  $p$  service facilities aimed at maximizing the gross amount of intercepted flow: flows are intercepted or not, there is no indication of where in the journey they are intercepted. Nor is there any impetus to prefer any location over another. In the real world, however, consumers often wish to obtain a service at or close to a specific location along their OD path. In this chapter, I address the optimal location of facilities at which products are picked up, or services received, along a pre-determined trip, such as the daily commute between home and workplace. Unlike in the traditional approach, however, patrons often express locational and proximity preferences (referred to hereafter simply as locational) for their visits. Fundamental to the modelling of locational decisions is some measure of proximity. I envision four types of consumer locational preference scenarios representing a wide spectrum of consumer choice.

- (i) The "Video" scenario: patrons have no locational preferences – they simply wish to pick up a video on their journey home to or from work.
- (ii) The "Coffee" scenario: patrons wish to pick up their cup of coffee as early in their trip as possible so that they may enjoy it while driving from home to work.
- (iii) The "Pizza" scenario: patrons want to pick up their pizza as late in their trip as possible so that it will be as warm as possible when they get home.
- (iv) The "Hamburger" scenario: patrons want to pick up their hamburger as close to where they will be at lunchtime during their journey.

I use the term "Where" scenario to generalize these scenarios in which the consumers have preferences as to where in the journey the pickup is made, e.g., scenario (ii), (iii) and (iv).

In this chapter, I introduce a common principle that embraces any "Where" scenario: the benefit arising from locational preferences as to where the potential service is received. Based on this principle, I proposed a *benefit-maximizing pickup location-allocation model*. This model, like FILM, is context free; it is potentially

applicable to any flow-based facility location system where consumers have locational preferences. I believe the pickup concept expressed through the four scenarios to be an excellent way of introducing this principle and model to the literature, thus I refer to my model simply as PUP.

Commercial optimization engines (e.g., CPLEX 9.1) are reliable and easy-to-use engines for solving linear and integer optimization models such as the PUP model. A geographic information system (GIS) is a system for management, analysis, and display of geographic knowledge, which is represented using a series of information sets such as maps and globes, geographic data sets, processing and work flow models, data models, and metadata (ESRI 2007). GIS engines (e.g., ArcGIS 9.1) provide powerful tools for users to visualize and examine spatial relationships among entities, and to represent data in a way that may reveal patterns and relationships that are hard to detect using nonvisual approaches. In this chapter, I integrate ArcGIS 9.1 and CPLEX 9.1 to examine the PUP model with two traffic networks for Edmonton, Alberta, Canada. A smaller network is trimmed to a suitable size for demonstrating the workings of PUP; the other network tests the model's capabilities in a middle-size, realistic network setting.

## 2.2. Background

### 2.2.1 The Flow-Interception Location Model

Hodgson (1990) and Berman, Larson, and Fouska (1992) independently developed the flow-interception location model aimed at demand represented as flow traveling on various paths of the network. FILM is designed to locate  $p$  facilities so as to maximize the number of consumers who encounter at least one facility along their trips. The problem is formulated as (Hodgson 1990):

$$\text{Maximize: } Z^F = \sum_{q \in Q} f_q X_q \quad (1)$$

s.t.

$$X_q \leq \sum_{j \in q} Y_j, \forall q \in Q \quad (2)$$

$$\sum_{j \in J} Y_j = p \quad (3)$$

$$X_q \in \{0, 1\}, \forall q \in Q \quad (4)$$

$$Y_j \in \{0, 1\}, \forall j \in J \quad (5)$$

In this formulation, the input data is:

$Q$  = the set of nonzero flow paths indexed by  $q$

$J$  = the set of potential facility sites indexed by  $j$

$j \in q$  = the set of potential facility sites along path  $q$

$f_q$  = the flow volume on the path  $q$

$p$  = the number of facilities to be located

and the objective function and decision variables are:

$Z^F$  = the objective function, total flows intercepted at least once

$$X_q = \begin{cases} 1 & \text{if the flow on path } q \text{ is intercepted by a facility along the path } q \\ 0 & \text{otherwise} \end{cases}$$

$$Y_j = \begin{cases} 1 & \text{if there is a facility located at potential facility site } j \\ 0 & \text{otherwise} \end{cases}$$

The objective function (1) is aimed at intercepting as much flow as possible, subject to the constraints that flow on path  $q$  cannot be intercepted unless there is at least one facility along path  $q$  (2), and that exactly  $p$  facilities are located (3). Constraints (4) and (5) are the standard integrality conditions.

The FILM model is mathematically equivalent to the classic maximal covering location model (Church and ReVelle 1974). FILM is a fruitful garden for research. About 40 academic articles based on FILM have been published during the last 17 years. Yet, FILM itself is under-defined as a location-allocation model, which should consider where facilities should be *located* and which demands should be *allocated* to which facilities. FILM seeks to optimally locate service facilities, but does not explicitly consider allocation of demand to the open facilities. For this reason, I call Hodgson's model a flow-interception *location* model, although it is often incorrectly called a *location-allocation* model in the literature (Hodgson *et al.* 1990, 1993, 1996, 1998, ReVelle and Eiselt 2005).

### 2.2.2 The Pickup Problem

In this section, I introduce the *pickup problem*. Within simple network situations, I begin by considering the four pickup scenarios which illustrate how I use a benefit function to represent the locational preferences of patrons. In the "Video" scenario, patrons have no locational preferences – they do not care where in the journey the pickup is made. The benefit of intercepting a unit of flow is constant at each node along the trip (Figure 2-1-A). I can directly apply the traditional FILM to this situation. In any "Where" scenario, the benefit of intercepting a unit of flow is different at each potential facility site on a path: the traditional FILM is thus not appropriate. I introduce a subscript  $j$  to indicate location at potential facility site  $j$  in path  $q$ , and a term  $b_{qj}$  to indicate the benefit of intercepting one unit of flow on path  $q$  at potential facility site  $j$ . I introduce a new decision  $X_{qj}$  variable:  $X_{qj} = 1$  if the flow on path  $q$  is intercepted by a facility at site  $j$  and 0 otherwise. The objective function then becomes:

$$\text{Maximize: } Z^F = \sum_{q \in Q} \sum_{j \in q} f_q b_{qj} X_{qj} \quad (6)$$

The goal becomes to maximize the total benefit of intercepting flows in a network considering where in the journey they are intercepted.

I use simple contrived examples to illustrate the relationship between benefits to preferred locations. In any scenario, benefit is greatest (a value of 1.0) at the most preferred location and decreases by 0.2 (or any other predefined factor) with each increment of distance from it. For instance, the benefit of "Coffee" pickup decreases from origin to destination (Figure 2-1-B,  $b_{q1} = 1.0$ ,  $b_{q2} = 0.8$ , etc.); the benefit of "Pizza" pickup increases from origin to destination (Figure 2-1-C,  $b_{q1} = 0.2$ ,  $b_{q2} = 0.4$ , etc.). If patrons wish to pick up a hamburger at a particular time, the node nearest to

where they will be at that time (I assume node 4) is the preferred location and the benefit of “Hamburger” pickup decreases with distance away from that node (Figure 2-1-D).

Simple two-origin, one-destination trees (Figure 2-2) better demonstrate the flow interception characteristics of this problem. Suppose there is one unit of flow on each OD path. At any node, the benefits of intercepting both flows may be aggregated. In the “Video” scenario (Figure 2-2-A), nodes 1 and 3, and nodes 2 and 4 intercept only flows from one origin, while nodes 5, 6 and 7, intercept both flows: benefits are aggregated to  $1.0 + 1.0 = 2.0$ , any of these three nodes (nodes 5, 6 and 7) is optimal. In the “Coffee” scenario (Figure 2-2-B), benefits decrease with distance away from the origin. The optimal location for the system is at node 5, which provides a benefit of 0.6 from each path; neither is served optimally but the total benefit for the system is maximized. In the “Pizza” scenario (Figure 2-2-C), the destination gets 1.0 unit from each OD path and is optimal for the system. In the “Hamburger” scenario, benefits are calculated as illustrated in Figure 2-2-D. I consider much more complex networks in the numerical experimentation section.

### 2.2.3 The Benefit Function

The value that a customer obtains from a service is, in many cases, a function of how close the service facility is to the preferred location. The benefit of locating a service facility for pickup arises from the various consumer locational preferences. As I have defined it,  $b_{qj}$  is the benefit of intercepting one unit of flow along path  $q$  at node  $j$ . From origin to destination,  $b_{qj}$  is constant for “Video;” decreasing for “Coffee;” increasing for “Pizza;” and variable for “Hamburger.” The common thread is that in each scenario  $b_{qj}$  decreases with distance away from the preferred node(s) on path  $q$ . I define  $d_{qj}$  as the distance from node  $j$  to the preferred location for patrons of trip  $q$ ; thus  $b_{qj}$  is, in general, a decreasing function of  $d_{qj}$ .

In real life, the actual benefit function will depend on the specifics of the particular study area, product, consumer, and so on. An appropriate function might be obtained from a variety of sources such as marketing and expert surveys. Regardless of how complex the benefit function might be, however,  $b_{qj}$  is expressed in the objective function of my model exogenously as a parameter: the structure of the model is not changed. Here, I define  $b_{qj} = e^{-\alpha d_{qj}}$ ,  $\alpha > 0$ . The exponential decay is appropriate for loss of heat from coffee (or pizza) although utility may not decline at the same rate. That is, I follow the suggestion proposed by Berman (1995) in the context of a model entertaining deviation from predetermined flow paths. I note that  $\alpha$ , the scaling constant of the benefit function, is used to measure the importance of proximity to the preferred location for consumers. I emphasize that when  $\alpha = 0$ , the value of  $b_{qj}$  is 1.0 despite the value of  $d_{qj}$ , in which case (e.g., the “Video” scenario) the objective of my model reduces to that of FILM.

### 2.3. Model Formulation

The traditional FILM model is aimed at maximizing the number of consumers who

encounter at least one facility on their trips. The traditional  $X_q$  variables in FILM identify whether there is at least one facility along path  $q$ . My PUP model is aimed at maximizing the benefits arising from the location where consumers encounter facilities, determined by their locational preferences. The new  $X_{qj}$  variables in PUP identify at which facility flow along path  $q$  is intercepted:  $X_{qj} = 1$  if the flow on path  $q$  is intercepted by a facility at node  $j$  and 0 otherwise. The mathematical formulation of PUP is:

$$\text{Maximize: } Z^p = \sum_{q \in Q} \sum_{j \in q} f_q b_{qj} X_{qj} \quad (7)$$

s.t.

$$\sum_{j \in q} X_{qj} \leq 1, \forall q \in Q \quad (8)$$

$$X_{qj} \leq Y_j, \forall q \in Q, j \in q \quad (9)$$

$$\sum_{j \in J} Y_j = p \quad (10)$$

$$X_{qj} \in \{0, 1\}, \forall q \in Q, j \in q \quad (11)$$

$$Y_j \in \{0, 1\}, \forall j \in J \quad (12)$$

The objective function (7) aims at maximizing the benefit that arises from the different consumer locational preferences. Constraint (8) ensures that flow on path  $q$  is intercepted by at most one facility located in the path. Constraint (9) ensures that flow on path  $q$  may be intercepted at a node only if there is a facility located there. Constraint (10) ensures that exactly  $p$  facilities are located. Constraints (11) and (12) are the standard integrality conditions. Like the traditional FILM model, the PUP model is not particularly “integer friendly.” My real-world examples below show that relaxing variables  $X_{qj}$  of PUP ( $0 \leq X_{qj} \leq 1$ ) obtains, in general, integer values and better solution speed due to a reduction in the number of branching iterations needed.

Before getting to the numerical experimentation, I discuss the node-optimality of PUP. For the “Video” scenario, the objective function aims at intercepting as much flow as possible. This scenario has node-optimality – location at a node is always at least as good as location on a link because either endpoint of a link can intercept all its flow. For the “Where” scenarios, the objective function aims at intercepting as much flow at or close to preferred locations as possible. When all preferred locations are at network nodes, PUP has node-optimality – at least one endpoint of a link is better than location on the link because at least one endpoint of a link intercepts all its flow and is closer to the preferred location for its flow. When some preferred locations are on links, PUP does not have node-optimality – a preferred location on a link may intercept the same flows as either endpoint of a link, and, obviously, be closer to the preferred location than are the endpoints. Here, I can simply add each preferred location on links as a network node to guarantee at least one network node that is better than location on a link. In a word, the “Video” scenario always has node-optimality, while the other three “Where” scenarios have node-optimality only if all preferred locations are at original or added network nodes. I note that with large real-world data, the “Hamburger” scenario may have more preferred locations on links

than there are original network nodes because the number of OD flow pairs may be much larger than the number of network nodes. For advanced ways of adding network intersection points in other related models, readers are referred to Kuby, Lim and Upchurch (2005).

#### **2.4. Numerical Experimentation**

My study uses traffic data for Edmonton, Alberta, Canada, a prairie city of about 700,000 population in an area of around 700 km<sup>2</sup> (Census of Edmonton, 2005). My data were provided by the City of Edmonton Transportation and Streets Department (TSD), who aggregated the data into traffic zones represented as centroids. Zone centroids are connected to the transportation network by one or more feeder links and interzonal traffic. Travel times and distances are estimated for each traffic zone's centroid. Flows are vehicle flows for all pairs of traffic zones in the full Edmonton area. TSD states that their data have been produced according to industry standards and that their forecasting model is highly recognized throughout North America. My CPLEX implementation simply reads flow paths as data and maximizes the benefit of intercepting flows along the paths however they are defined. Sophisticated methods of traffic assignment (of OD flows to paths) exist, but in this study I use the most commonly used methodology in the transportation literature and simply assign all OD flows to the network over the least-time paths using a network simplex model in AMPL CPLEX 9.1.

First, I use a very simple subset of the Edmonton network's morning peak traffic for 1989; here my goal is to demonstrate how the model works by relating the solutions to simple flow patterns. Second, I use the full afternoon peak traffic data sets for 2001; here my goal is to produce a middle-size, realistic flow system and to explore how the PUP model performs in this system. I solve all examples optimally using AMPL CPLEX 9.1, and spatially analyze and map the results with ArcGIS 9.1. I note that in my numerical experimentation, I use driving time (seconds) as my "distance" measure.

##### **2.4.1. Morning Peak Traffic Flows for Edmonton, 1989**

I first demonstrate the workings of PUP with the 1989 morning peak network, comprising 703 nodes and 2198 links and described by Hodgson, Rosing and Storrier (1996). The entire database is too detailed to map clearly enough for a visual demonstration, so I highly simplified the network by designating only the top 10 destination traffic zones, quite central to the city, as destinations. As origins, I use all other traffic zones with nonzero flow to these 10 destinations, 148 in all. This subset of the network comprises 1458 OD pairs, representing 20% of all total traffic. Each of the 10 destination nodes arbitrarily serves only as a destination and each of the 148 origin nodes arbitrarily serves only as an origin, allowing origin and destination nodes to be clearly mapped. In this simplified structure, the reader can relate to the structure of benefits arising from tradeoffs between flow bundling at well-placed intersections and proximity to preferred locations.

The input data for PUP are the 1458 OD flow pairs, the 703 potential facility

sites, the set of nodes along each OD path  $q$ , and  $b_{qj}$ . Recall that  $b_{qj} = e^{-\alpha d_{qj}}$  is the benefit of intercepting one unit of flow along path  $q$  at node  $j$ . It is important to grasp the characteristics of the function and the nature of  $d_{qj}$ , which activates the solution properties under each scenario. Each node along each path is a satisfactory location for “Video” consumers, I ignore the value of  $\alpha$  and set  $b_{qj} = 1.0$ . For each node, I calculate the driving time from the origin for the “Coffee” scenario and the driving time from the destination for the “Pizza” scenario. I assume that the preferred location under the “Hamburger” scenario is exactly at the center of each path, from which I calculate driving time.

Considering the tradeoffs between flow structure and the importance of proximity to preferred locations, I would expect to observe the following spatial distribution patterns: “Coffee” facilities will be oriented to origins; “Hamburger” facilities to the centers of trips; “Pizza” facilities to destinations; and “Video” facilities will be located at nodes that intercept maximum flow.

As an example of how the model performs, I consider the optimal locations identified by each scenario with  $p = 3$  and  $\alpha = 0.002$ . Circles with areas proportional to the total out-flows at origins illustrate the underlying flow structure of this problem (Figure 2-3). The three “Coffee” facilities are located somewhat centrally to the origins in the north, west, and south of the city. The three “Hamburger” facilities are located toward the centers of trips, on the major network routes between origins and destinations. None of these optimal facilities occurs at a trip origin or destination but at intermediate nodes, a characteristic of traditional FILM solutions. This analysis is conducted in an artificial situation with only a few destinations which are clustered centrally: the three “Pizza” facilities are located at the top three destinations. The three “Video” facilities are also concentrated near the destinations: two are located at the top two destinations, and the other is located at a node between the top 3 and 4 destinations. The small difference between the “Video” and “Pizza” solutions helps us to grasp the different effects of flow interception and benefit maximization.

I consider the flow patterns in the area encompassing the top four destinations (Figure 2-4). Link widths are proportional to the flow on them and circle areas are proportional to the flows passing through nodes. Because the destination nodes serve only as destinations, the links connected to them can be considered to be one-way. The objective of the “Video” scenario is the same as in the traditional FILM objective, to maximize flow interception by avoiding flow cannibalization, wasteful redundant flow-interception. Node 11 is a better location for a “Video” facility than is destination 3 because with it the model intercepts more flows ( $12902 - 12657 = 245$ ) than with destination 3 (Table 2-1). Most of the flows intercepted by nodes 11 and destination 3 are not intercepted by destinations 1 or 2: intercepting flow at destinations 1 and 2 greatly improves the objective. The objective of the “Pizza” scenario is to reap benefits by intercepting as many consumers at or close to destinations as possible. Destination 3, where all “Pizza” consumers are served at the preferred location, is a better location for a “Pizza” facility than is node 11 because with it, the model achieves more benefits ( $12657 - 11593 = 1064$ ) than with node 11

(Table 2-1). The importance of proximity to trip destinations pulls the “Pizza” facility from node 11 to destination 3. Table 2-1 illustrates the obvious result that each optimal scenario provides the greatest benefits for its particular consumers, and the inappropriateness of the other “Where” solutions to particular scenarios.

**Table 2-1: Flow and benefit obtained in each scenario ( $p = 3$ ,  $\alpha = 0.002$ )**

	Video	Pizza	Coffee	Hamburger
Flow Interception	<b>12902</b>	12657	9571	6535
Pizza Benefit	11593	<b>12657</b>	3901	690
Hamburger Benefit	3617	3030	<b>7264</b>	3064
Coffee Benefit	1124	929	2226	<b>2923</b>

**Table 2-2: Benefit % ( $p = 1 \dots 10$ ,  $\alpha = 0.002$ )**

$p$	Video	Pizza	Coffee	Hamburger
1	15.8	15.2	3.6	8.1
2	30.3	29.7	6.7	15.9
3	41.8	41.3	9.5	23.5
4	52.7	52.6	11.8	29.8
5	63.6	63.6	14.0	35.4
6	73.0	73.0	16.1	38.7
7	81.3	81.3	18.1	42.0
8	88.0	88.0	20.0	45.0
9	94.1	94.1	22.0	47.8
10	100.0	100.0	23.9	50.5

To further demonstrate the tradeoffs between network flow structure and the importance of proximity to preferred locations ( $\alpha$ ), I explore how solutions are affected by increasing the value of  $\alpha$  from 0.0 to a large factor. As  $\alpha$  increases, the optimal locations identified by each “Where” scenario move from the traditional FILM locations toward the preferred locations; when  $\alpha$  is large enough, the optimal locations are at the nodes most exemplifying the ideal preferences. I demonstrate this by observing the migration of three “Coffee” facilities from a flow orientation to a source orientation as the value of  $\alpha$  increases (Figure 2-5). At  $\alpha = 0.0$  the three “Coffee” facilities are located as in the traditional FILM; however, at higher values of  $\alpha$ , flow volumes at intersections and distance to the preferred location trade off. At  $\alpha = 0.025$ , locations are very close to the origins; at  $\alpha = 0.035$ , they are at the major origins. These patterns are observed because the network flow structure is fixed but the power of the importance of proximity to preferred location increases with  $\alpha$ .

I now consider the relationship between benefits and the number of facilities under each scenario. Cost-effectiveness diagrams plot the benefit achieved for each



value of  $p = 1 \dots 148$  under each scenario (Figure 2-6). The “Video” and “Pizza” scenarios produce similar relationships regardless of the value of  $\alpha$  because of the strong concentration of flow near the destinations. Up to  $p = 4$ , the “Video” scenario provides a little more benefit than the “Pizza” scenario (Table 2-2) because of the difficulty of finding locations that combine flow interception and proximity to preferred location with small  $p$ . At  $p \geq 5$ , they have the same results; at  $p \geq 10$ , all possible benefit is obtained. The curves for the “Coffee” and “Hamburger” scenarios demonstrate the diminishing returns expected of FILM-type problems, and are smooth with no sharp breakpoints to suggest ideal tradeoffs between facility numbers and performance. The “Coffee” curve rises smoothly to become asymptotic to full benefit. The “Hamburger” curves do not approach full benefit, because the preferred locations are usually not nodal – facilities cannot be located exactly at the path centers.

#### 2.4.2. Afternoon Peak Traffic Flows for Edmonton, 2001

TSD provided vehicle flows for a traffic network of 395 traffic zones, 2211 nodes, 6211 links, and  $(395 \times 395 - 395) = 155,630$  OD flow pair for the afternoon peak period in 2001. The full OD flows require too much CPU memory for CPLEX to solve the FILM or PUP model. The number of OD flow pairs grows very quickly with the number of traffic zones, and even with the most efficient and specialized heuristics, good solutions to large flow-interception problems will be beyond the capability of a personal computer. I am currently working on the use of heuristics, standard optimization engines and geographical information techniques together for solving large flow-interception problems. Here my goal is to produce a mid-size, realistic flow system and to explore how the PUP model performs in this system. Thus, I used ArcGIS 9.1 and C++ to reduce the network size – I discarded OD flow pairs with less than 1 unit of flow, reducing the number of zones to 290 and the number of OD flow pairs to 16,488 representing 75% of the total traffic flow. I pared the network down to 1746 nodes and 4606 links by removing all nodes and links which do not fall on these least-time paths (Colour Figure 2-7). These afternoon flows are dominated by movement from the central to the peripheral areas, but each of the 290 zone centroids serves both as an origin and a destination, producing what I view as a realistic test-bed for PUP.

My major interest in this section is to see whether the types of solution characteristics observed in the early example are borne out in this large realistic database. I optimally solve all scenarios with  $p = 1 \dots 14$ . I solve for the “Pizza” and “Coffee” Scenarios with  $\alpha$  increasing from zero to a value at which all facilities are located at the top  $p$  preferred nodes. For the “Hamburger” scenario, I increase  $\alpha$  until the facilities no longer shift – they are as close to, but not at, the preferred centers of major flow paths. (These centers are not usually at nodes, and to add nodes would greatly complicate the network, so the “Hamburger” scenario has no node-optimality to compare with “origin” or “destination” for the other two “Where” scenarios.)

My findings are exemplified by the results for  $p = 5$ , for which I present maps and tabulate particular solution characteristics. To relate to solution characteristics under each scenario, it is important that the readers grasp the important elements of

the network flow structure, especially the locations of major origins and destinations. I have therefore identified what I consider to be the most important 27 nodes in the network (Table 2-3).

**Table 2-3: 27 important nodes**

Node	F	O	D	TNF (%)	Node	F	O	D	TNF (%)
1	1	24	23	6.73	15	15	--	--	3.59
2	2	--	--	5.27	16	16	--	--	3.50
3	3	--	--	4.90	17	17	--	--	3.49
4	4	--	--	4.43	18	18	--	--	3.45
5	5	--	--	4.27	19	19	--	--	3.43
6	6	--	--	4.23	20	20	--	--	3.35
7	7	--	--	3.93	21	51	2	181	2.72
8	8	--	--	3.78	22	27	4	4	3.03
9	9	--	--	3.75	23	36	5	6	2.84
10	10	1	16	3.74	24	25	6	17	3.14
11	11	--	--	3.69	25	39	10	2	2.83
12	12	--	--	3.67	26	74	11	5	2.54
13	13	3	1	3.65	27	58	15	3	2.66
14	14	--	--	3.60					

*F*—Rank of flow at node; *O*—Rank of out-flow from Origin;  
*D*—Rank of in-flow to destination; *TNF*—the percent of total network flow

I indicate the rank of node in respect to flows through it, the total numbers of trip origins out of it, or the total number of trip destinations into it. Node 1, for instance, is also known as F1 (highest flow, 6.73% of total flow), O24 (24<sup>th</sup> highest origin, 0.94% of total outflow), and D23 (23<sup>rd</sup> highest destination, 0.73% of total inflow). I map these 27 important nodes which include the top 20 flows nodes, the top 6 origins, and the top 6 destinations (Colour Figure 2-7). A major cluster comprises 19 of the 27 nodes, concentrated near the city center (inset, Colour Figure 2-7) representing the downtown, university and government center areas. A second cluster is near node 13 which represents West Edmonton Mall (until recently, the world's largest shopping centre). The third cluster is along two major arteries in the south (nodes 22, 15, and 23). Because the afternoon peak is the rush-hour journey from work, trip destinations are more dispersed than trip origins. I may roughly imagine that most flows come from the three clustered areas to disperse throughout the entire map, yet recognize that many major destinations are near or at the three cluster areas.

I compare the spatial distribution patterns under each scenario for the solution with  $p = 5$  and  $\alpha = 0.0005$  (Colour Figure 2-8). "Video" facilities are located at nodes with high flow (F1, F5, F12, F13, and F14), but not simply at the five nodes with the highest flow: clearly the optimal solution to PUP avoids flow cannibalization. Traffic

tends to be bundled into a few major arteries and nodes. The “Video” facilities are dispersed to intercept flows throughout the network: F1 intercepts flow to the North, F14 to the West, F5 to the South, F12 to the Southeast; and F13 to and from the West Edmonton Mall. In contrast, “Coffee” facilities are located at origins with very high out-flow (O1, O2, O3, O5, and O6) and “Pizza” facilities are located at destinations with very high in-flow (D1, D2, D3, D4, and D23). Flow interception maintains its traditional role however. A “Coffee” facility chooses O6, which is central to many major origins, but intercepting more flow than O4. A “Pizza” facility chooses D23 rather than D5 because D23 (also F1) can intercept much more flow than D5 (also F74).

The fact that all the optimal “Coffee” and “Pizza” facilities are located at major preferred nodes indicates that the importance of proximity to preferred locations exerts more influence on location than does flow structure for high values of  $\alpha$ . Because most preferred locations for the “Hamburger” scenario have no node-optimality, I cannot make the same sort of assessment, but I observe in the map that “Hamburger” facilities are located toward the center of trips, on major arteries between origins and destinations.

To further demonstrate the tradeoffs between network flow structure and the importance of proximity to preferred locations ( $\alpha$ ), I explored how “Coffee” and “Pizza” locations move by augmenting  $\alpha$  from 0.0 by increments of 0.00005 and noting solution changes. At  $\alpha = 0.0$ , the “Pizza” and “Coffee” solutions are, of course, identical to the “Video,” or FILM solution. As the value of  $\alpha$  increasingly jumps from one step to another step, one “Pizza” facility always moves to a node with less flow. For instance, node F14 jumps to node F15 when the value of  $\alpha$  jumps from step 1 to step 2 (Table 2-4). At steps 5 through 8, one “Pizza” facility always moves to a more preferred node (a node with higher in-flow). For instance, Node D23 jumps to node D5 when the value of  $\alpha$  jumps from step 7 to step 8. Similar movement patterns of “Coffee” facilities are observed (Table 2-4). I thus observe that the optimal solutions of PUP conform to my intuition about the tradeoffs between intercepting flows in general and doing so in preferable locations. This is the first time that such intuition about spatial distribution patterns has been experimentally verified.

Finally, I investigate the importance of the PUP model, by determining how much better it serves the locational preferences of consumers than FILM. I do this with a simple index:

$$\frac{Z_{\text{PUP Solution}}^p - Z_{\text{FILM Solution}}^p}{Z_{\text{FILM Solution}}^p} \times 100\% \quad (13)$$

The numerator indicates how much more benefit PUP solution provides to consumers than FILM solution does. The denominator indicates the benefit obtained by FILM solutions. I term the scale index *Superiority of PUP*. The index evaluates the importance of the PUP model in terms of how much more preference-based benefit the model provides than would the simple FILM. For  $p = 5$ , and varying values of  $\alpha$  (Figure 2-9), I observe that the superiority of PUP solutions for “Coffee” and “Pizza” rises quickly to over 100 per cent for values of  $\alpha$  approaching those where the most highly preferred locations host facilities.

**Table 2-4:** Facility location movement, benefit, and flow as  $\alpha$  increases ( $p = 5$ )

Scenario	Step	Facility Location Movement with an Increasing $\alpha$					
		$\alpha$	Facility Location Movement				
Video	0	$-\infty \leq \alpha \leq +\infty$	F1	F13	F5	F12	F14
Pizza	1	$0.00000 \leq \alpha \leq 0.00002$	D23/F1	D1/F13	F5	F12	<u>F14</u> ↓
	2	$0.00003 \leq \alpha \leq 0.00004$	D23/F1	D1/F13	F5	<u>F12</u> ↓	F15
	3	$0.000045 \leq \alpha \leq 0.000055$	D23/F1	D1/F13	<u>F5</u> ↓	F14	F15
	4	$0.00006 \leq \alpha \leq 0.0001$	D23/F1	D1/F13	<u>F6</u> ↓	F14	F15
	5	$0.00015 \leq \alpha \leq 0.0002$	D23/F1	D1/F13	D3/F58	<u>F14</u> ↓	F15
	6	$0.0003 \leq \alpha \leq 0.0004$	D23/F1	D1/F13	D3/F58	D2/F39	<u>F15</u> ↓
	7	$0.0005 \leq \alpha \leq 0.0008$	<u>D23/F1</u> ↓	D1/F13	D3/F58	D2/F39	D4/F27
	8	$\alpha \geq 0.0009$	D5/F74	D1/F13	D3/F58	D2/F39	D4/F27
Coffee	1	$0.00000 \leq \alpha \leq 0.00002$	O24/F1	O3/F13	F5	F12	<u>F14</u> ↓
	2	$0.00003 \leq \alpha \leq 0.00005$	O24/F1	O3/F13	<u>F5</u> ↓	F12	F15
	3	$0.00006 \leq \alpha \leq 0.0001$	O24/F1	O3/F13	O1/F10	F12	<u>F15</u> ↓
	4	$0.00015 \leq \alpha \leq 0.0002$	O24/F1	O3/F13	O1/F10	<u>F12</u> ↓	O5/F36
	5	$0.0003 \leq \alpha \leq 0.0004$	<u>O24/F1</u> ↓	O3/F13	O1/F10	<u>O4/F27</u> ↓	O5/F36
	6	$0.0005 \leq \alpha \leq 0.0012$	O6/F25	O3/F13	O1/F10	O2/F51	<u>O5/F36</u> ↓
	7	$0.0013 \leq \alpha \leq 0.0014$	<u>O6/F25</u> ↓	O3/F13	O1/F10	O2/F51	O4/F27
	8	$\alpha \geq 0.0015$	O5/F36	O3/F13	O1/F10	O2/F51	O4/F27

*F*—Rank of Flow at node; *O*—Rank of Out-flow from Origins;  
*D*—Rank of In-flow to Destinations

### 2.5. Conclusions

A basic underlying assumption of conventional flow-interception location models is that consumers do not care where service facilities are located along their trip. In the real world, however, consumers often wish to obtain a product or service at or near a

specific location along their trip, often at their trip origins or destinations. The particular preferences depend on the nature of the products or services demanded. In this chapter, I propose the PUP model, which transforms the traditional flow-interception location model to a flow-interception location-allocation model. The PUP considers various locational preferences, providing a much broader, realistic approach to enterprise in private and public sectors. I note that the traditional flow-interception models are special cases of my proposed model. Moreover, the PUP introduces a common principle that considers a wide range of preferential scenarios with a model driven by an exogenously calculated proximity preference function.

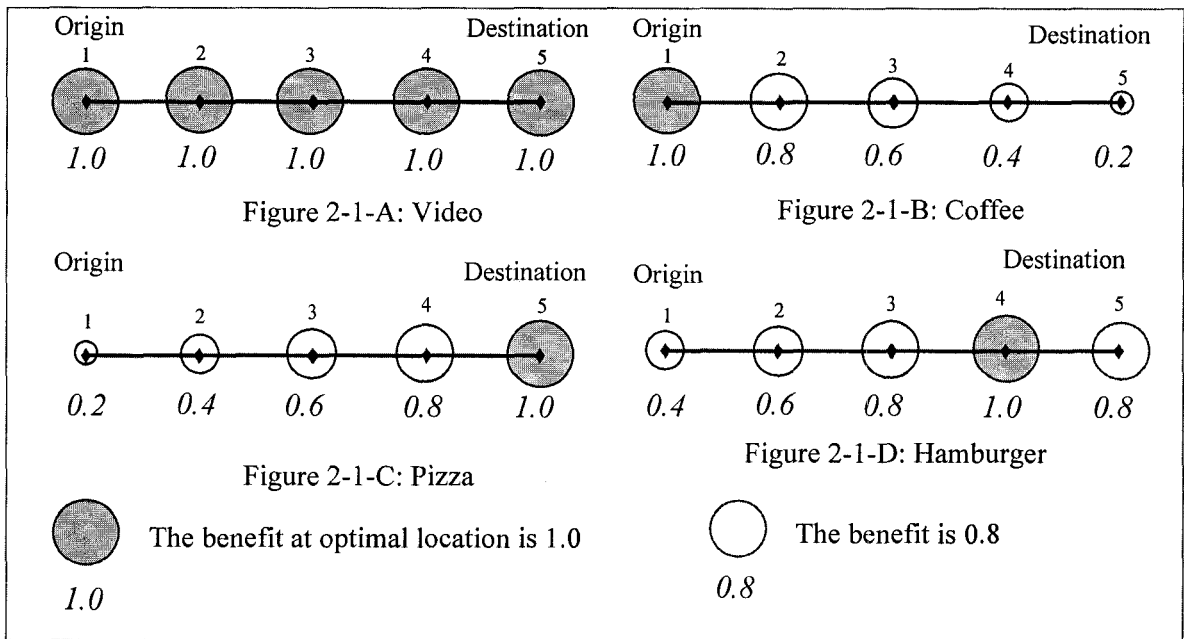
I applied PUP to morning (1989) and afternoon (2001) peak traffic flows in Edmonton, Alberta, Canada. My observations of the spatial distribution and cost-effectiveness characteristics of four scenarios, indicating varying consumer preference structures, demonstrate that the optimal locations identified by the “Video” scenario arise solely from network flow structure (as in the traditional models), whereas optimal locations identified by “Where” scenarios result from a trade-off between the network flow structure and the importance of proximity to preferred locations. I note that this is the first time that my intuition about spatial distribution patterns has been experimentally verified. In short, my proposed model enhances spatial decisions by incorporating consumer locational and proximity preferences, providing a fruitful garden for further research. I note, in closing, that ignoring consumer preferences can greatly impair the benefits of flow interception modelling.

## References

- Berman, O., D. Bertsimas, R. C. Larson. 1995. Locating discretionary service facilities, II: maximizing market size, minimizing inconvenience. *Operations Research* **43** 623-32.
- Berman, O., R. C. Larson, N. Fouska. 1992. Optimal location of discretionary service facilities. *Transportation Science* **26** 201-11.
- Census of Edmonton. 2005. Oliver is home to more seniors and gold bar residents are most loyal, says the census. B3, *Edmonton Journal* October 22.
- Church, R. L., C. ReVelle. 1974. The maximal covering location problem. *Papers of the Regional Science Association* **32** 101-18.
- ESRR. 2007. GIS: Getting Started. [http://www.esri.com/getting\\_started/index.html](http://www.esri.com/getting_started/index.html)
- Hodgson, M. J. 1990. A flow-capturing location-allocation model. *Geographical Analysis* **22** 270-79.
- Hodgson, M. J. 1998. Developments in flow-based location-allocation models. In *Economic Advances in Spatial Modelling and Methodology: Essays in honour of Jean Paelinck*, edited by D.A. Griffith, C.G. Amrhein, J-M Huriot, Kluwer Academic Publishers, 119-132.
- Hodgson, M. J., K. E. Rosing, A. L. G. Storrier. 1996. Applying the flow-capturing location-allocation model to an authentic network: Edmonton, Canada. *European Journal of Operational Research* **90** 427-43.
- Kuby, M., S. Lim, and C. Upchurch. (2005). Dispersion of nodes added to a network. *Geographical Analysis* **37** 383-409.

- ReVelle C.S., H. A. Eiselt. 2005. Location analysis: a synthesis and survey. *European Journal of Operational Research* **165** 1-19.
- ReVelle, C.S., R. Swain. 1970. Central facilities location. *Geographical Analysis* **2** 30-42.
- Transportation and Streets Departments. 2005. 2001 Afternoon peak traffic data, City of Edmonton, Edmonton, Alberta, Canada.

**Figure 2-1: The four pickup scenarios**



**Figure 2-2: Sample two-origin, one-destination trees**

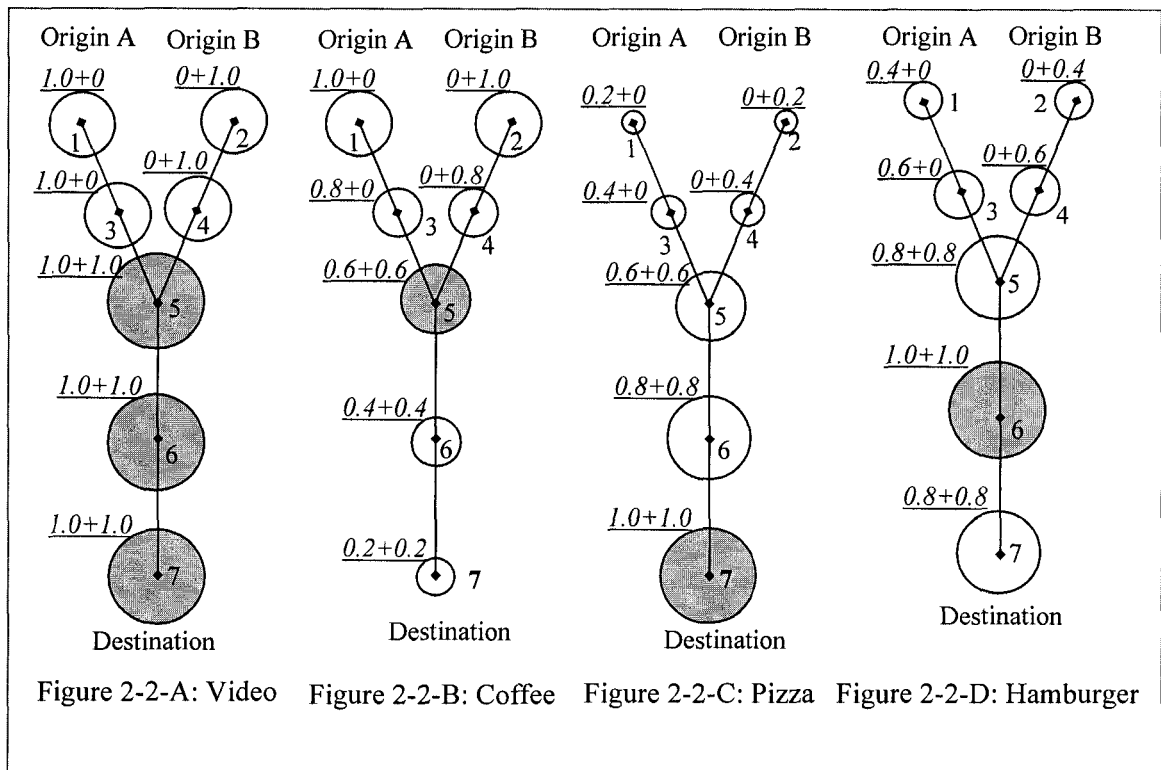


Figure 2-3: Optimal locations ( $p = 3$ , constant = 0.002)

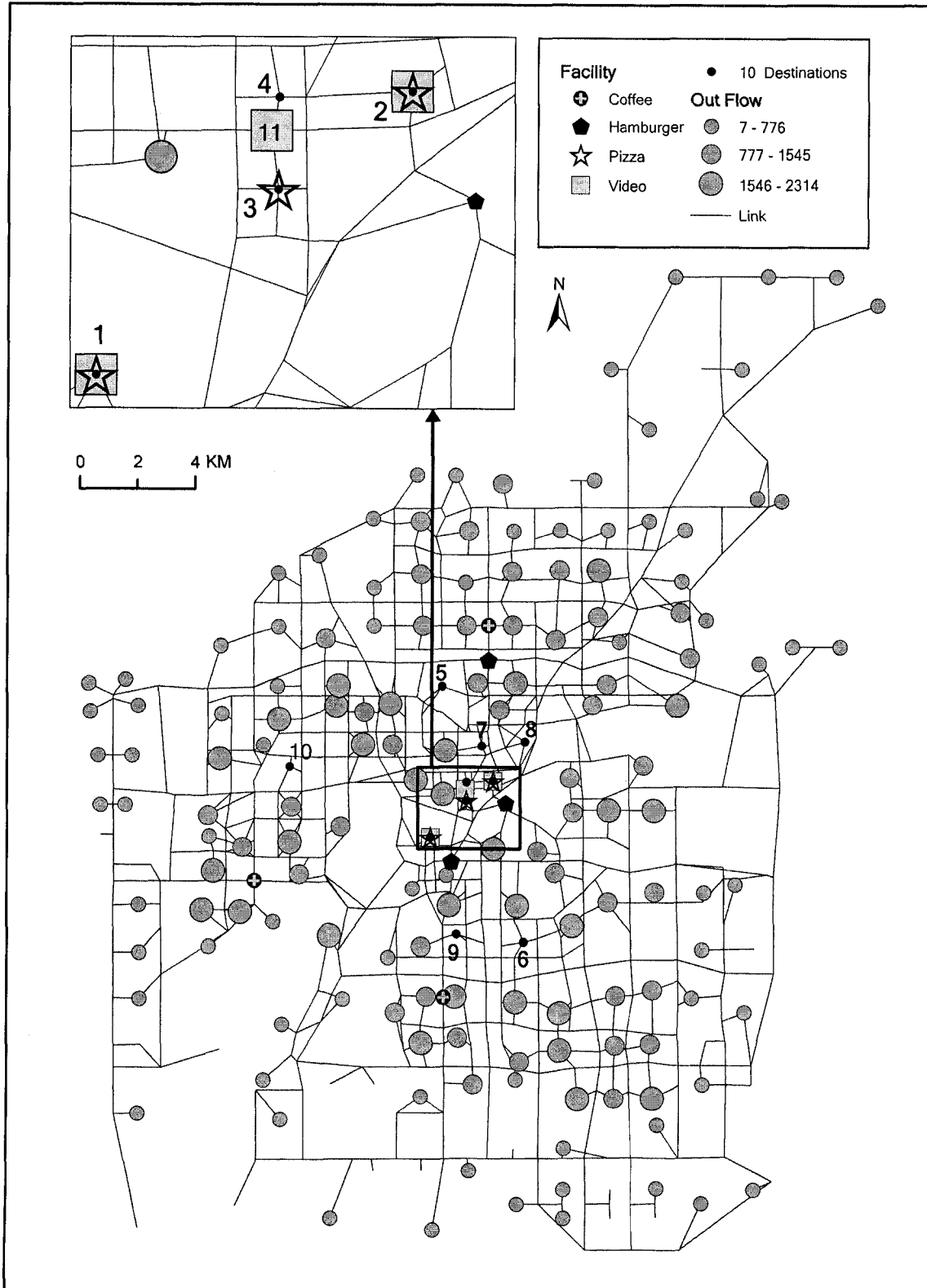




Figure 2-4: Zoomed in optimal locations ( $p = 3$ , constant = 0.002)

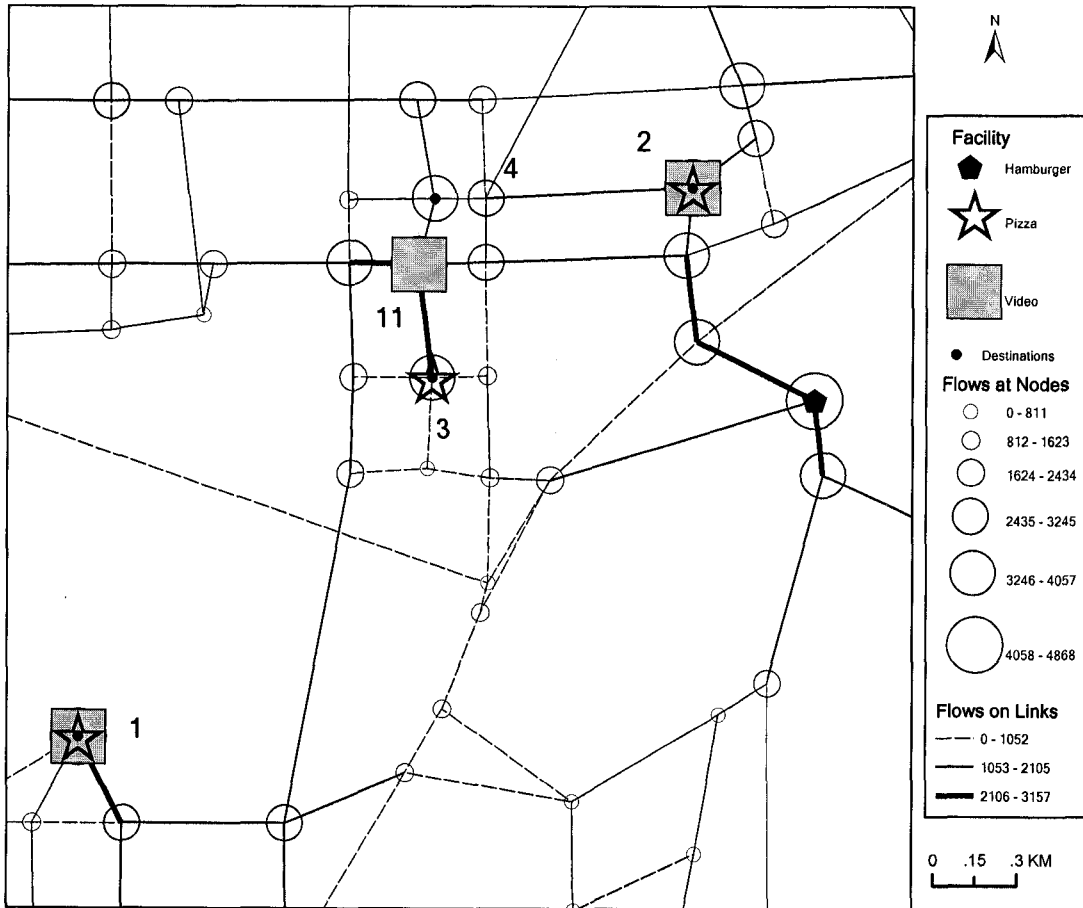


Figure 2-5. "Coffee" optimal locations with increasing constant ( $p = 3$ )

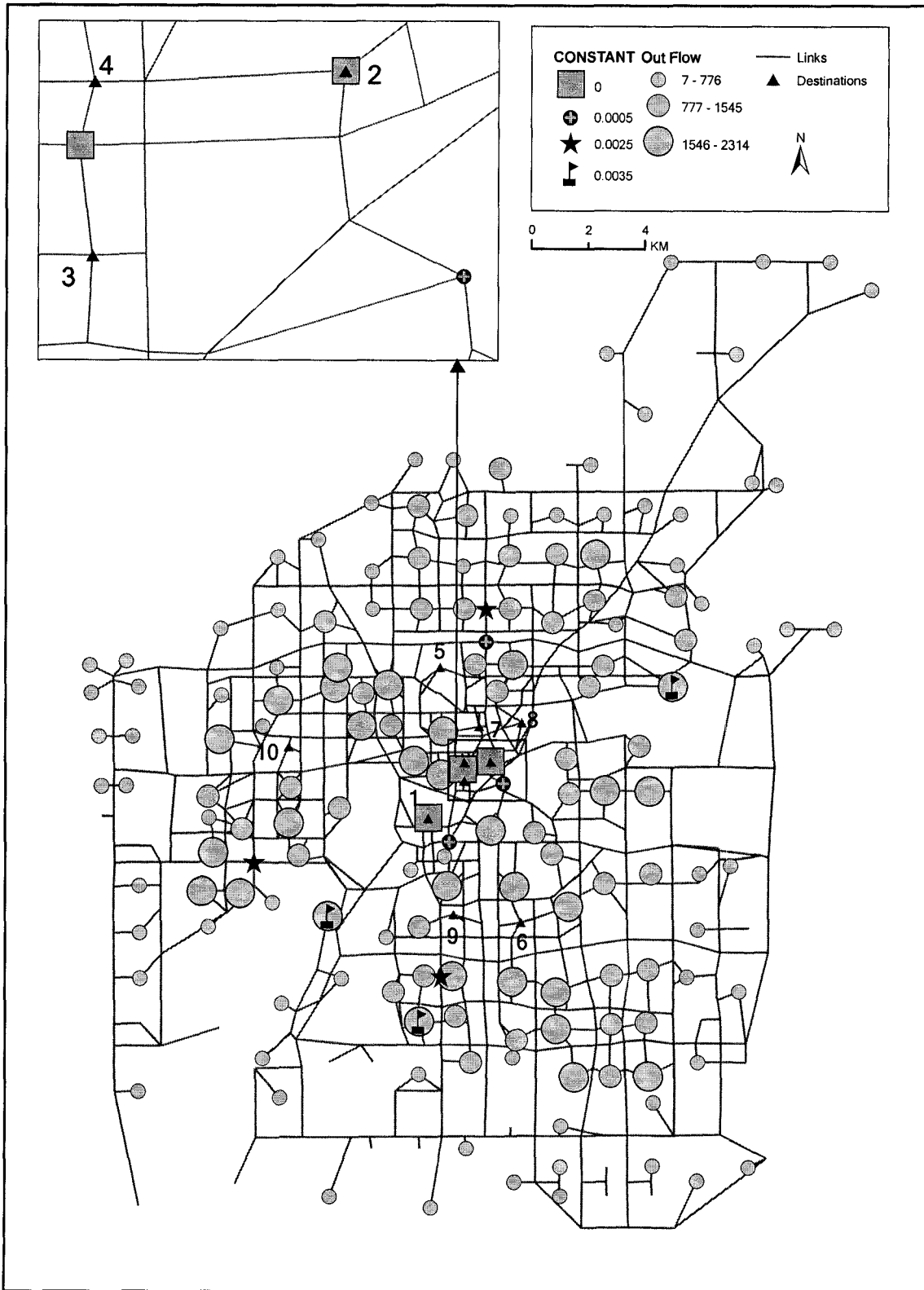


Figure 2-6: Cost-effectiveness of the optimal solutions ( $1 \leq p \leq 148$ )

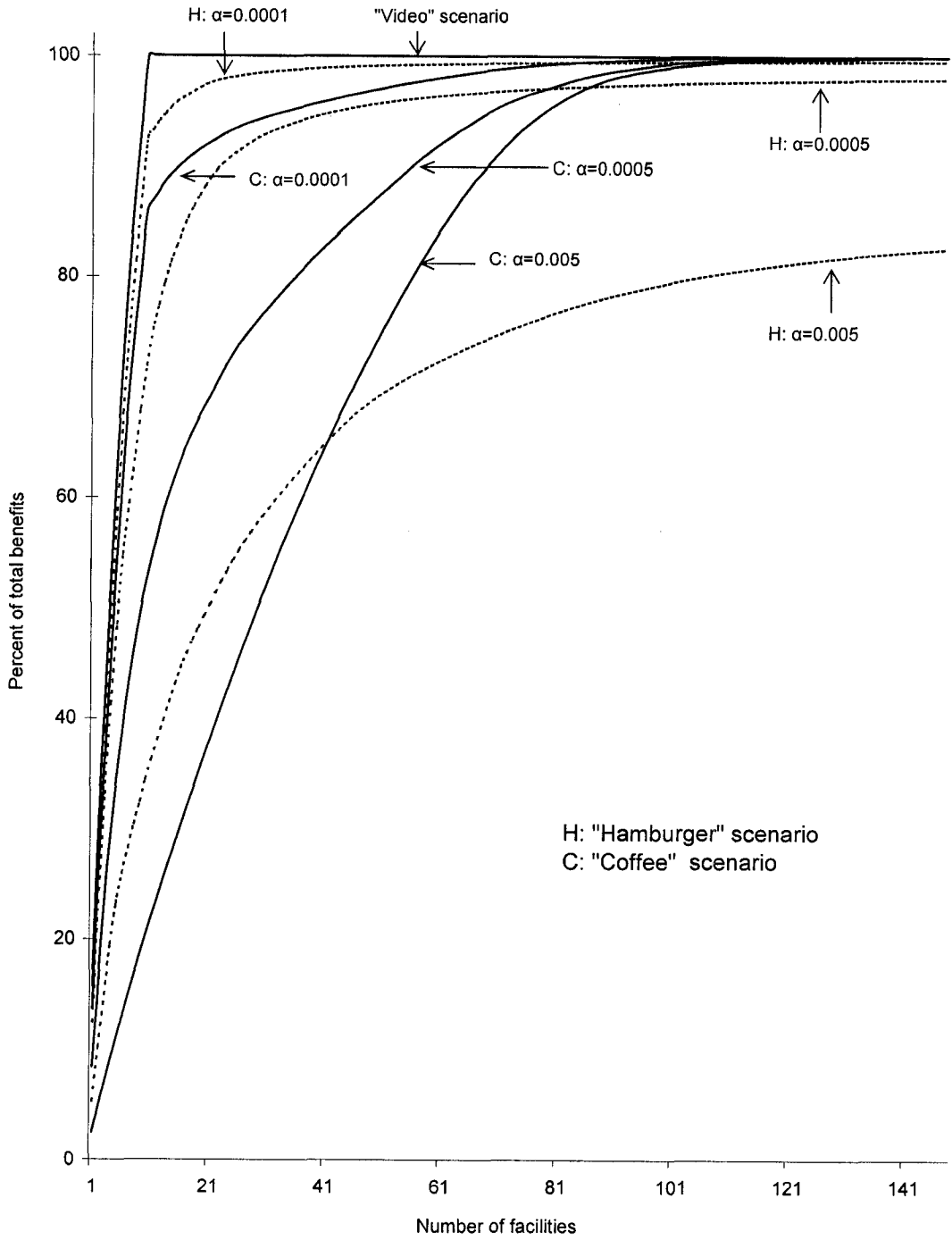


Figure 2-7 (colour): Network flow structure (2001)

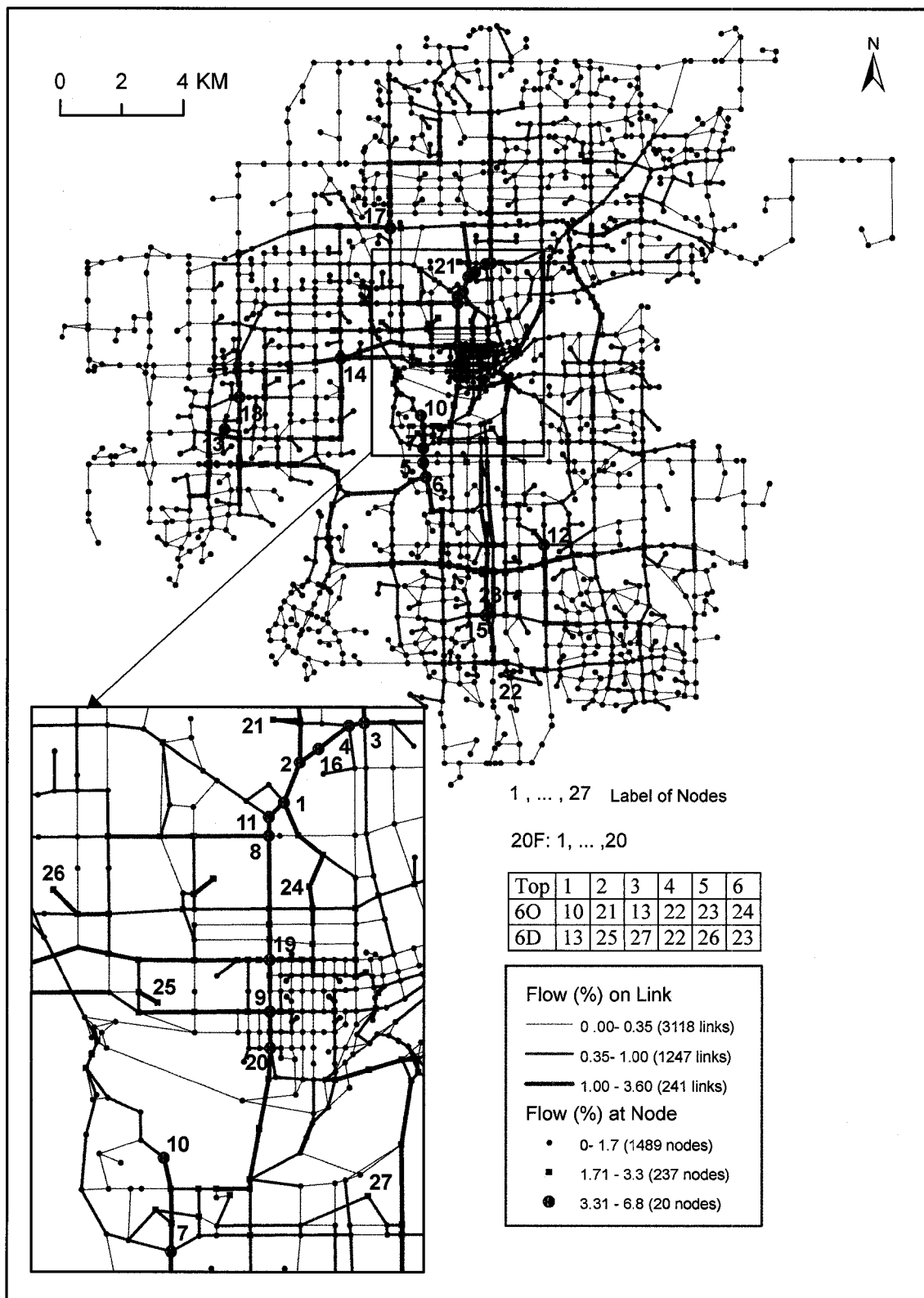
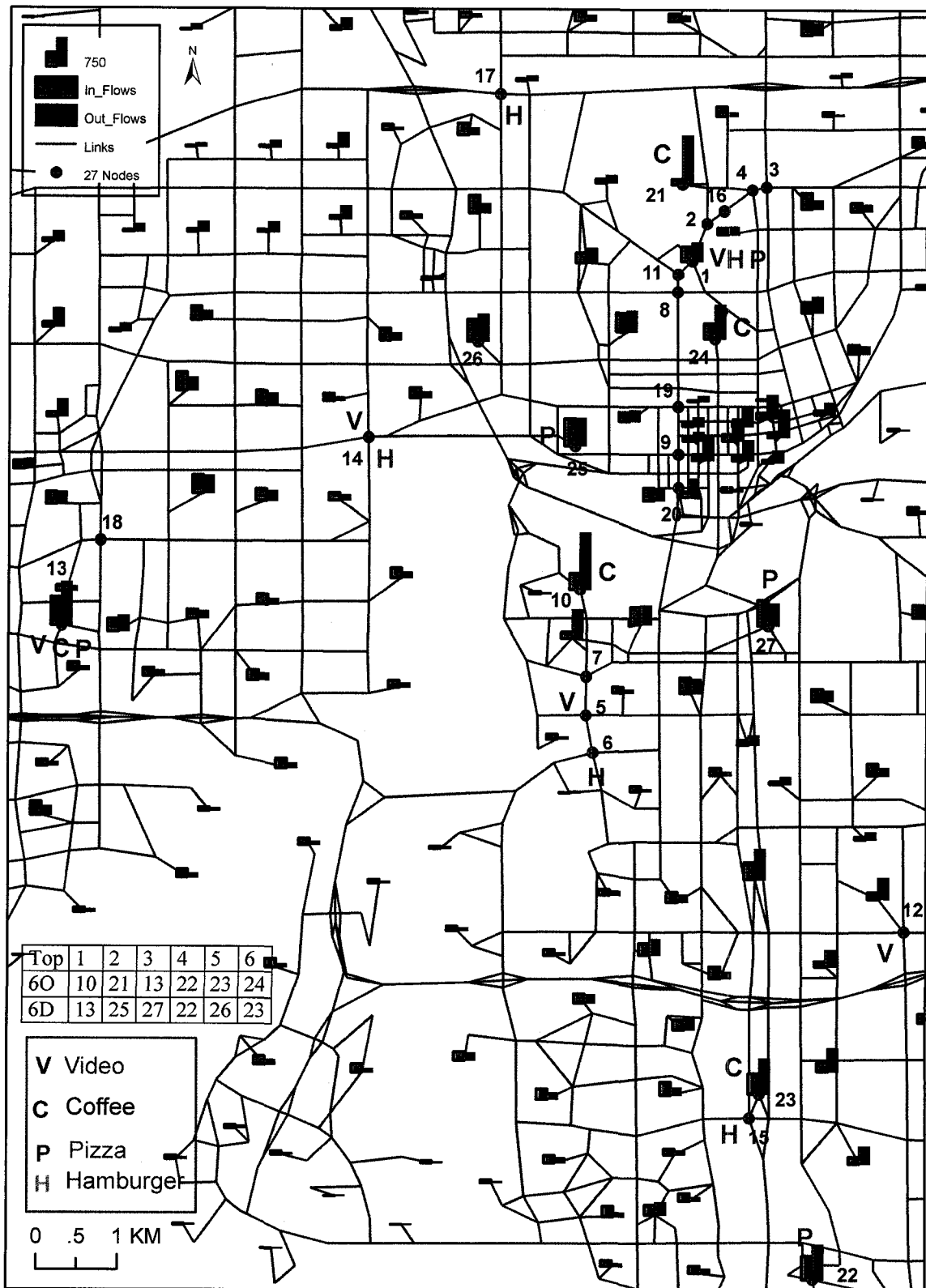
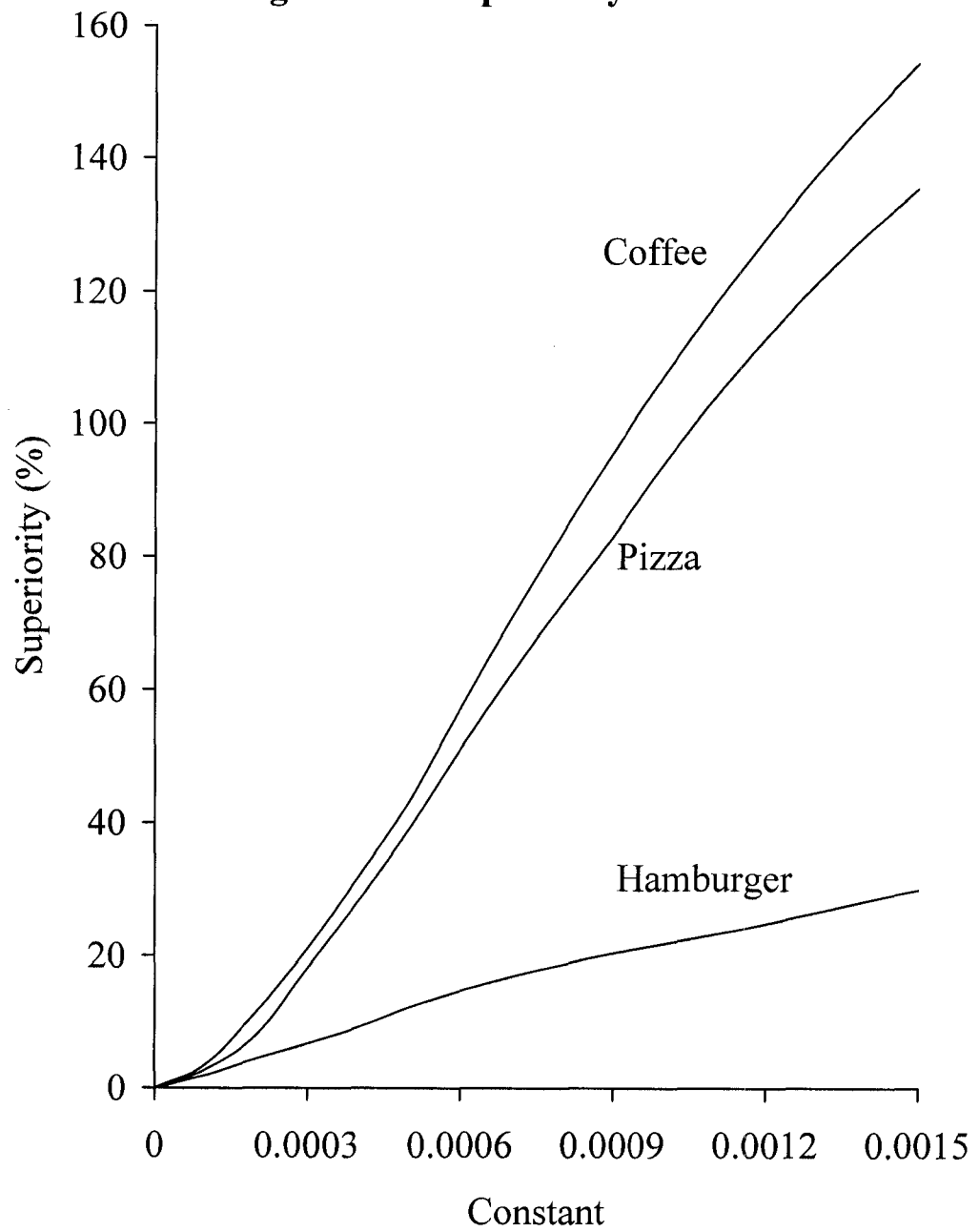


Figure 2-8 (colour): Optimal locations ( $p = 5$ , constant = 0.0005)



**Figure 2-9: Superiority of PUP**

### Chapter 3 \*

#### A Generalized Model for Locating Facilities on a Network with Flow-Based Demand

**Summary:** Flow-interception location problems can be characterized as identifying good or optimal facility locations on a network with flow-based demand. Since the early 1990s, about 30 different flow-interception location models have appeared in about 40 academic publications. Location researchers tend to introduce changes in objective functions and/or assumptions by developing new models. This has led to many disparate models, each requiring a somewhat different solution method, challenging the development of standardized software that would encourage widespread use in real-world, strategic decision making processes. In this chapter, I formulate a generalized flow-interception location-allocation model (GFIM) which, with few exceptions, requires only simple modifications to its input data to effectively solve all current deterministic flow-interception problems. Additional flow-interception problems can be solved by simple model manipulation or addition of constraints. Moreover, several critical considerations in flow-interception models – such as deviation from predetermined journeys, locational and proximity preferences, and capacity issues – can be handled within the single framework. Two real-world examples reported in the literature (morning and afternoon peak traffic for the city of Edmonton in Canada) show that CPLEX optimally solves GFIM much more efficiently than it does the classic flow-interception location model.

**\* A version of this chapter has been submitted for publication.**

**Zeng, Weiping,** Ignacio Castillo, M. John Hodgson. 2007. A generalized model for locating facilities on a network with flow-based demand. Submitted to *Transportation Science* on May 4, 2007, submission No:TS-2007-0105, under review.

### 3.1. Introduction

Facility location is a central problem in real-world, strategic decision making processes. In traditional location theory, demand for service is assumed to occur at fixed locations on a traffic network. This process is generally termed point-based demand. The main purpose of travel for point-based demand is to obtain or provide service. That is, consumers residing at nodes on the network travel to facility locations to obtain service (e.g., going to school or workplace), or, alternatively, service providers located at network nodes travel to consumer locations to provide service as requested (e.g., ambulance, police, and repair services).

This chapter focuses on demand for service that is expressed by flows traveling on origin-destination paths of a traffic network, flow-based demand. In contrast to traditional point-based demand, the main purpose of travel for flow-based demand is not necessarily to obtain service, but if there is a facility on the predetermined journey, consumers may choose to obtain service. Fast food outlets, automatic teller machines, and gasoline stations are examples of services that experience flow-based demand. Flow-interception location problems can be characterized as identifying good or optimal facility locations on a network with flow-based demand. The most basic location model that incorporates flow-based demand is the flow-interception location model (FILM) developed by Hodgson (1990) and Berman, Larson, and Fouska (1992).

Over the past 17 years, in order to formally characterize a wide spectrum of consumer desires and needs, about 30 flow-interception location models have appeared in about 40 academic publications. Location researchers tend to introduce changes in objective functions or assumptions by developing new models. This has created numerous disparate models, each viewed as requiring its own solution method, challenging the development of standardized software that would encourage widespread use of location models in real-world, strategic, decision-making processes.

This chapter introduces a generalized flow-interception location-allocation model (GFIM), into which most current deterministic flow-interception models can be transformed. Several critical considerations in flow-interception models – such as deviation from a predetermined journey, locational and proximity preferences, and capacity issues – are handled within GFIM's single modeling framework. Interestingly, real-world examples, using 1989 morning (Hodgson, Rosing, and Storrier 1996) and 2001 afternoon (Zeng, Hodgson, and Castillo 2007) peak traffic data for the city of Edmonton, Canada, show that a standard optimization engine such as ILOG-CPLEX optimally solves GFIM much more efficiently than the classic flow-interception location model in at least two instances of large real-world data. GFIM provides a new way of looking at location problems relative to flow-based demand and a new way of identifying similarities and differences among flow-interception location problems. GFIM also provides a fruitful garden for future research; thus, making a substantial contribution to the flow-interception problem literature.

The remainder of article is organized as follows. Section 3.2 considers FILM, the basic flow-interception location model. Section 3.3 introduces GFIM, the proposed generalized model. Section 3.4 demonstrates a variety of current and future models to



be special cases of GFIM. Section 3.5 examines the performance of GFIM using 1989 morning and 2001 afternoon peak traffic data for the city of Edmonton, Canada. The final section offers our conclusion.

### 3.2. The Flow-Interception Location Model

Hodgson (1990) and Berman, Larson, and Fouska (1992) developed the basic classic flow-interception location model. This basic model is aimed at maximizing the number of consumers who encounter at least one facility along their predetermined journeys. FILM is formulated as:

$$\text{Maximize: } Z^F = \sum_{q \in Q} f_q X_q \quad (1)$$

s.t.

$$X_q \leq \sum_{j \in q} Y_j, \forall q \in Q \quad (2)$$

$$\sum_{j \in J} Y_j = p \quad (3)$$

$$X_q \in \{0, 1\}, \forall q \in Q \quad (4)$$

$$Y_j \in \{0, 1\}, \forall j \in J \quad (5)$$

In this formulation, the parameters are:

$Q$  = the set of nonzero flow paths indexed by  $q$ ;

$J$  = the set of potential facility sites indexed by  $j$ ;

$j \in q$  = the set of all nodes on path  $q$ ;

$f_q$  = the flow volume along path  $q$ ;

$p$  = the number of facilities to be located.

The objective function and the decision variables are denoted by:

$Z^F$  = the objective function, total flows intercepted at least once;

$$X_q = \begin{cases} 1 & \text{if } f_q \text{ is intercepted by a facility along path } q \\ 0 & \text{otherwise} \end{cases};$$

$$Y_j = \begin{cases} 1 & \text{if there is a facility located at node } j \\ 0 & \text{otherwise} \end{cases}.$$

Note that it is possible to relax the binary requirement on the  $X_q$  variable as  $0 \leq X_q \leq 1$  because the 100% of total flows along path  $q$  could be intercepted by a set of facilities (rather than one facility) on the path  $q$ . The objective function (1) is aimed at intercepting as much flow as possible, subject to the constraints that flow along path  $q$  cannot be intercepted unless there is at least one facility on path  $q$ , (2); and that exactly  $p$  facilities be located, (3). Constraints (4) and (5) are the standard integrality conditions.

I note that FILM seeks to optimally locate service facilities, but does not explicitly consider the allocation of flow-based demand to the open facilities. That is to say that FILM does not consider *where* a flow-based demand is served. Thus, in location theory, FILM is considered a *location* model rather than a *location-allocation* model. The implication of this observation is that FILM cannot directly take into

account many critical considerations in location analysis (for example, deviation from a predetermined journey, locational and proximity preferences, and capacity issues). Furthermore, FILM is aimed at maximizing the number of consumers who encounter at least one facility on their trips. In real-world settings, there are often more complex objectives such as maximizing the amount of network protection against risks (hazardous cargos and drunk drivers for example), maximizing benefits arising from different consumer locational or proximity preferences, and maximizing benefits of multiple exposures to billboards.

### 3.3. The Generalized Flow-Interception Location-Allocation Model

The generalized flow-interception location-allocation model is formulated as:

$$\text{Maximize: } Z^G = \sum_{q \in Q} \sum_{j \in N_q} G_{qj} X_{qj} \quad (6)$$

s. t.

$$\sum_{j \in N_q} X_{qj} \leq 1, \forall q \in Q \quad (7)$$

$$X_{qj} \leq Y_j, \forall q \in Q, j \in N_q \quad (8A)$$

$$\sum_{j \in J} Y_j = p \quad (9)$$

$$X_{qj} \in \{0, 1\}, \forall q \in Q, j \in q \quad (10)$$

$$Y_j \in \{0, 1\}, \forall j \in J \quad (11)$$

Parameters  $Q$ ,  $J$ ,  $p$  and decision variable  $Y_j$  in GFIM are the same as in FILM. In comparison with FILM, GFIM uses two new parameters and one new decision variable as follows.

$G_{qj}$  = the contribution to the objective function when flow along path  $q$  is intercepted by a facility at node  $j$ , ( $G$  denotes the matrix  $G_{qj}$ );

$N_q$  = the set of nodes capable of intercepting the flow along path  $q$  (e.g., the set of all nodes on path  $q$ );

$X_{qj}$  = the proportion of the total objective function contribution where flow along path  $q$  is intercepted by the facility at node  $j$  ( $0 \leq X_{qj} \leq 1$ ).

The objective function (6) is aimed at intercepting as much total objective contribution as possible. Constraint (7) ensures that at most, 100% of total flows along path  $q$  can be intercepted by the set of facilities in  $N_q$ . Constraint (8A) ensures that flow along  $q$  can only be intercepted by the facility located at node  $j$ . Variants of this constraint, denoted with the letters “B” and “C,” will be introduced below. Constraint (9) ensures that exactly  $p$  facilities are located. Constraints (10) and (11) are the standard integrality conditions.

The three new generalized terms are the major innovations of this formulation. The value  $f_q$  in FILM represents only the number of consumers along path  $q$ . The value of  $G_{qj}$  in GFIM represents the available objective function contribution on path  $q$  at specific node  $j$ . Thus,  $G_{qj}$  can represent the number of consumers in the FILM case, the total benefit of intercepting these consumers, or even persons at risk along a path where flows are hazardous cargos or drunk drivers. The decision variable  $X_q$  in FILM represents only whether all the consumers along path  $q$  are intercepted. The

new decision variable  $X_{qj}$  in GFIM identifies the proportion of the contribution along path  $q$  obtained by the facility at node  $j$ . FILM considers only the set of nodes on a path ( $j \in q$ ), while GFIM considers the set of nodes ( $N_q$ ) capable of intercepting flow along path  $q$ . This is important in situations where consumers are allowed to deviate from predetermined OD flow paths. In short, the three new generalized terms address an individual consumer's consideration of any specific facility in the network, providing the great potential of GFIM to effectively consider all kinds of consumers' desires and needs in real-world situations.

### 3.4. Special Cases of GFIM

Most flow-interception location models are structurally identical to GFIM: the key difference hinges on the interpretation of  $G$ . In this section I show that a variety of flow-interception location models are indeed special cases of GFIM. My findings are demonstrated with a 7-node test example, for which I present a map (Figure 3-1), flow paths (Table 3-1), input data  $G$  and output results (Table 3-2). In the implementation, I coded GFIM in the AMPL language (Fourer, Gay, and Kernighan 2002) using the well-known ILOG-CPLEX optimizer (Figure 3-2). Note that  $N_q$  is the set of nodes capable of intercepting the flow along path  $q$ . The set of  $N_q$  is  $\{j \mid j \in q\}$  for FILM and may be  $\{j \mid j \in J\}$  for other problems. Figure 3-2 obtains  $N_q$  directly from input data  $G_{qj}$  because the set of nodes capable of intercepting the flow along path  $q$  can always be simply represented as all  $G_{qj} > 0$ . In the large real-world examples (see section 5 below), I coded GFIM in a more memory efficient way. However, the model file (Figure 3-2) and the data file (Figure 3-3) make it easy for readers to grasp the nature of  $G$ , which activates GFIM. The results in Table 3-2 are all solved by modifying input data (Figure 3-3). Readers are encouraged to compare the results (the objective function value, the least number of facilities required to capture all objective value in the network, and other solution results) for these special cases.

**Table 3-1: OD flow paths**

Path $q$	Flow	Num	Path by Nodes	Path by Links
1	2	4	1 3 5 7	1 2 3
2	1	3	2 3 6	4 5
3	1	3	4 5 6	6 7
4	2	2	4 7	8

*Num: The total number of nodes on each path*

#### 3.4.1. The Basic Flow-Interception Case

The original basic flow-interception location problem can, of course, be solved in GFIM by calculating the value of  $G_{qj}$  as:

$$G_{qj} = \begin{cases} f_q, \forall j \in q \\ 0, \forall j \notin q \end{cases}$$

The value of  $G_{qj}$  indicates that each facility on path  $q$  provides the same objective contribution ( $f_q$ ) for consumers along path  $q$ ; only a facility on path  $q$  can intercept the

flow along path  $q$ . In my example, nodes 1, 3, 5, and 7 are on path 1 so the facility at each node can intercept 2 units of flow from path 1; nodes 2, 4, and 6 are not on path 1 so they cannot intercept flow along path 1 (Figure 3-1, Table 3-2). The optimal location for a single facility ( $p = 1$ ) is at node 7 which intercepts the two largest flows. The optimal locations for two facilities ( $p = 2$ ) are at nodes 6 and 7 which intercept all flows in the network.

**Table 3-2: Input matrix of  $G$  and output results for each case of GFIM**

Cases of GFIM	Input Matrix of $G$ at each node								Output Results		
	path	1	2	3	4	5	6	7	$p$	$Z$	Solution
Basic Flow-Interception	1	2	.	2	.	2	.	2	1	4	{7}
	2	.	1	1	.	.	1	.	2	6	{6, 7}
	3	.	.	.	1	1	1	.			
	4	.	.	.	2	.	.	2			
Protection	1	12	.	8	.	2	.	0	1	12	{1}
	2	.	3	2	.	.	0	.	2	19	{1, 4}
	3	.	.	.	3	2	0	.	3	22	{1, 2, 4}
	4	.	.	.	4	.	.	0			
Preference	1	0.10	.	0.28	.	1.22	.	2.00	1	4	{7}
	2	.	0.22	0.37	.	.	1.00	.	2	6	{6, 7}
	3	.	.	.	0.22	0.37	1.00	.			
	4	.	.	.	0.67	.	.	2.00			
Deviation Case 1	1	2	2	2	2	2	2	2	1	6	{5}
	2	.	1	1	.	1	1	.			
	3	.	.	1	1	1	1	1			
	4	.	.	.	2	2	.	2			
Deviation Case 2	1	2.00	0.74	2.00	0.74	2.00	1.22	2.00	1	5.22	{5}
	2	0.14	1.00	1.00	0.08	0.22	1.00	0.08	2	6.00	{6, 7}
	3	0.03	0.08	0.22	1.00	1.00	1.00	0.37			
	4	0.02	0.04	0.10	2.00	2.00	0.28	2.00			
Deviation Case 3	1	0	4	0	4	0	1	0	1	3	{5}
	2	4	0	0	5	3	0	5	2	0	{3, 4}
	3	7	5	3	0	0	0	2			
	4	20	16	12	0	0	8	0			
Deviation Case 2 (Preference)	1	0.10	0.02	0.27	0.10	1.21	0.16	2.00	1	4.03	{7}
	2	0.00	0.22	0.37	0.00	0.02	1.00	0.00	2	6.00	{6, 7}
	3	0.00	0.00	0.02	0.22	0.37	1.00	0.03			
	4	0.00	0.00	0.00	0.74	1.21	0.00	2.00			
Deviation Case 2 (Capacity $\leq 2.6$ )	1	2.00	0.74	2.00	0.74	2.00	1.22	2.00	1	2.5	{6}
	2	0.14	1.00	1.00	0.08	0.22	1.00	0.08	2	4.59	{5, 7}
	3	0.03	0.08	0.22	1.00	1.00	1.00	0.37	3	6.00	{5, 6, 7}
	4	0.02	0.04	0.10	2.00	2.00	0.28	2.00			

Entries with "." represent not available or zero;  $p$  is the number of facilities;  $Z$  is the objective value

### 3.4.2. Protection Cases

Vehicle inspection models are designed to protect networks against risks (hazardous cargos and drunk drivers for example). These types of risks are clearly flow-based and FILM could be used to optimally intercept offenders. FILM enacts a *punitive* approach aimed at catching as many violators as possible, regardless of where they are in their journeys. In contrast, vehicle inspection programs use a *preventive* approach aimed at inspecting and removing violators from networks as early in their trips as possible, thus reducing risk of network hazards as quickly as possible. Mirchandani, Rebello, and Agnetis (1995) suggested the principle of preventive inspection in some heuristic experiments but did not formalize it in a model. Hodgson, Rosing, and Zhang (1996) formulated the protection model for locating inspection stations. Gendreau, Laporte, and Parent (2000) developed several solution methods for the protection model. Horner and Groves (2007) applied the protection model to identify the optimal location of park-and-ride facilities, which is aimed at effectively reducing roadway congestion by intercepting the maximum amount of vehicle flow as early as possible in its OD journey.

The protection model can be transformed into GFIM by making  $G_{qj}$  the protection available to path  $q$  at node  $j$ . It is calculated as the product of the flow along path  $q$  and the number of persons at risk along path  $q$  between node  $j$  and the destination. On any link there is a risk density: the number of persons per unit distance who are exposed to the hazard on the link. Risk density is a function of the number of persons occupying (traveling, living, working, shopping, and visiting on or within a critical distance of) the link. For assessment of risk density, readers are referred to Zhang, Hodgson, and Erkut (2000). Recall that assessed risk density enters  $G_{qj}$  exogenously as a parameter. In this protection case,

$$G_{qj} = \begin{cases} f_q \times R(d_{jq_s}), \forall j \in q \\ 0, \forall j \notin q \end{cases}$$

where

$d_{jq_s}$  is the distance from node  $j$  to the destination of path  $q$ ;

$R(d_{jq_s})$  is the risk density along path  $q$  between node  $j$  and the destination.

Hodgson, Rosing, and Zhang (1996) assume a risk density of 1.0 throughout the network: the number of persons at risk is proportional to link length. In this case,  $R(d_{jq_s}) = d_{jq_s}, \forall j \in q$ . For my example, node 1 provides  $2 \times (1+2+3) = 12$  units of protection from flow 1 and node 7 does not provide any protection because it is obviously too late for node 7 to capture flows 1 and 4 (Figure 3-1, Table 3-2). Node 2 provides 3 units of protection from flow 2 but it does not provide any protection from flow 1 because node 2 is not on path 1. The optimal location for a single facility is at node 1 which protects the longest and largest volume of flow along path 1. The optimal locations for two facilities are at nodes 1 and 4 which provide 19 units of protection. It takes at least three facilities to protect the whole network.

### 3.4.3. Generalized Preference Cases

FILM implicitly assumes that there is no indication of where in the journey the flows are intercepted, nor is there any preference for one location over another. This is not an unrealistic assumption for some facilities such as advertising billboards. However, for most facilities (e.g., gasoline and refuelling stations, automatic teller machines, convenience stores, and fast food outlets), this assumption is tenuous because consumers often desire to obtain a product or service at or near a specific location on their trip. Zeng, Hodgson, and Castillo (2007) outline several “pickup” problems that involve various consumers locational and proximity preferences.

Consumer locational and proximity preferences are common issues for all flow-interception location problems. GFIM considers these issues by treating the value of  $G_{qj}$  as the benefit available to path  $q$  at node  $j$ . It is calculated as the product of the flow along path  $q$  and the benefit of intercepting one unit of flow on path  $q$  at node  $j$ . Regardless of how complex the benefit function might be, it is expressed as  $G_{qj}$  exogenously, as a parameter. In the Zeng, Hodgson, and Castillo (2007) model:

$$G_{qj} = \begin{cases} f_q e^{-\alpha d_{jq_p}}, \forall j \in q \\ 0, \forall j \notin q \end{cases}$$

where  $\alpha$  is a scaling constant reflecting the importance of distance from a preferred location and  $d_{jq_p}$  is the distance from node  $j$  to the preferred location for patrons of path  $q$ .

In my example, when the preferred location is at the destination and the value of  $\alpha$  is 0.5, node 1 provides  $2e^{-0.5 \times 6} = 0.10$  units of objective contribution from flow 1; node 7 provides  $2e^{-0.5 \times 0} = 2.00$  units (Figure 3-1, Table 3-2). The optimal location for a single facility is at node 7, the destination for the two largest flow volumes. It takes at least two facilities to obtain all the benefit available in the network.

### 3.4.4. Generalized Deviation Cases

FILM assumes that consumers will make no deviations, no matter how small, from the predetermined trip to visit a service facility. In reality, consumers may deviate from their predetermined trip to obtain services if there are no facilities on their paths. Berman, Bertsimas, and Larson (1995) discussed three generalizations of the problem where deviations from predetermined trips are considered.

#### 3.4.4. 1 Deviation Case 1

In the first generalization, flow along any path  $q$  is regarded as being intercepted only if there is a facility within a maximum distance  $\Delta$  of  $q$  (measured from the closest node on  $q$  to the facility). The second assumes that as the deviation distance increases, less and less flow will be intercepted. The third generalization assumes that all potential consumers will deviate if necessary to the closest facility regardless of the actual deviation distance. Deviation distance is defined as the extra distance incurred when customers deviate from their predetermined paths. The term  $d_{jq_d}$  denotes the

deviation distance between node  $j$  and path  $q$ . Consumers wishing to visit a facility are assumed to first take the shortest path to the facility, and then from the facility to take the shortest path to the destination. The sum of these two shortest distances minus the shortest origin-destination distance is the deviation distance. For instance, in my example, the deviation distance between node 6 and path 1 is  $(2 + 2) + (2 + 1) - (2 + 3 + 1) = 1$  (Figure 3-1, Table 3-1).

Berman, Bertsimas, and Larson (1995) developed three separate models to deal with the three generalizations. GFIM, however, directly deals with them via the modification of  $G_{qj}$  data, here defined as the expected number of customers along path  $q$  who become actual users of the facility at node  $j$ . The three generalizations simply represent three different ways of calculating the value of  $G_{qj}$  based on the deviation distance. For FILM, protection, and preference problems without deviation,  $N_q$  is the set of nodes on path  $q$ . For the first generalization,  $N_q$  is the set of nodes on path  $q$  and the set of nodes within distance  $\Delta$  of path  $q$ . The value of  $G_{qj}$  is calculated as:

$$G_{qj} = \begin{cases} f_q, \forall j \in \{j \mid d_{jq_d} \leq \Delta\} \\ 0, \forall j \in \{j \mid d_{jq_d} > \Delta\} \end{cases}$$

Recall that  $d_{jq_d}$  denotes the deviation distance between node  $j$  and path  $q$ . In my example, with  $\Delta = 3$ , consumers along path 1 travel less than 3 units of deviation distance to any node in the network; thus, any node can intercept 2 units of flow along path 1. On the other hand, the deviation distance between node 1 and path 2 is 4 units; thus, node 1 intercepts no flows from path 2 (Figure 3-1, Table 3-2).

#### 3.4.4.2 Deviation Case 2

For the second generalization,  $G_{qj}$  is the fraction of consumers along path  $q$  who would deviate to use the closest facility at node  $j$ . In this case,  $N_q$  is the set of potential facility sites in the network ( $N_q = J$ ). Berman, Bertsimas, and Larson (1995) suggest a popular negative exponential function of deviation distance to calculate the probability that a random path- $q$  consumer will deviate to use that facility, thus,  $G_{qj} = f_q e^{-\alpha d_{jq_d}}$ ,  $\forall j \in J$ . In my example (with  $\alpha = 0.5$ ), nodes 1, 3, 5, or 7 intercept  $2e^{-0.5 \times 0} = 2$  units of flow along path 1; node 2 intercepts  $2e^{-0.5 \times 2} = 0.74$  units of flow from path 1 (Figure 3-1, Table 3-2).

#### 3.4.4.3 Deviation Case 3

The third generalization is to minimize the total deviation distance traveled. In this case,  $N_q$  is the set of potential facility sites in the network ( $N_q = J$ ). To address the equivalence of GFIM to this case, I redefine GFIM as a minimization problem, select “=” in constraint (5), and define  $G_{qj} = f_q \times d_{jq_d}$ ,  $\forall j \in J$ . In my example, node 2 provides  $2 \times 2 = 4$  units of weight deviation from consumers along path 1 (Figure 3-1, Table 3-2).

The optimal location of a single facility for the first generalization is at node 5 which intercepts all four paths. The optimal location for the second generalization is at

node 5 which intercepts the full flow along paths 1, 3, and 4, and 0.22 units of flow along path 2. For the third, the optimal location is at node 5 where only consumers along path 2 make 3 units of weight deviation and the other three trips do not need to make deviations. The first generalization needs at least one facility to intercept all flows in the network, while the other two generalizations need at least 2 facilities to capture all flows in the network.

### 3.4.5. Generalized Capacity Cases

A critical characteristic of facilities is their capacity. But, all deterministic flow-interception location models formulated to date implicitly assume that the facilities being sited have infinite capacity. In many applications, however, this is unrealistic. As mentioned before, as a simple location model, FILM cannot consider how much flow is intercepted by each facility. Constraint (8A) of GFIM can be reformulated more generally to consider capacity issues as follows.

$$\sum_{q \in Q} G_{qj} X_{qj} \leq C_j Y_j, j \in J \quad (8B)$$

Where  $C_j$  is the capacity of the facility at node  $j$ . Constraint (8A) is a special case of constraint (8B) with  $C_j = \infty$ . No new models are required; constraint (8B) of GFIM can capture the capacity issues in any problem by inputting facility capacities as data if needed.

The 7-node example under deviation case 2 illustrates the characteristics of the capacity issue. Suppose that the capacity of each open facility is no more than 2.6 units of flow. For expository purposes, I consider all decision variables to be binary in this simple test problem. The optimal location for a single facility is at node 6 which intercepts  $1.22 + 1.00 + 0.28 = 2.5$  units of flow (Table 3-2). The optimal locations for two facilities are at nodes 5 and 7 which intercept 4.59 units of total flow: node 5 intercepts 2.00 units of flow along path 1 and 0.22 units of flow along path 2; node 7 intercepts 0.37 units of flow from path 2 and 2.00 units of flow along path 4. It takes at least 3 facilities to intercept all flows in the network. In contrast, in the uncapacitated problem under deviation case 2, the optimal location for a single facility is at node 5 which intercepts  $2.00 + 0.22 + 1.00 + 2.00 = 5.22$  units of flow; the optimal locations for two facilities are at nodes 6 and 7 which intercept all flow in the network.

### 3.4.6. Combinations of Cases

I have shown that GFIM effectively solves five different types of flow-interception problem. A real-world problem may simultaneously involve several considerations such as preference, deviation, and capacity cases. The number of combinations of five

elements is  $\sum_{e=1}^5 {}_5C_e = 31$ . A special case of GFIM may contain several subcases (e.g.,

three deviation cases) and there are many other potential cases of GFIM discussed below. Therefore, the number of combinations of cases could be very large.

Fortunately, by modifying the input data rather than by creating numerous new mathematical models, GFIM can effectively solve any combination of these cases.



A simple example is a flow-interception problem considering both the preference and deviation situations: consumers wish to patronize a facility at or near a specific location on their trip; if there is not a facility at their desired location they may consider a facility on their path or a facility at an acceptable deviation distance from their predetermined trip. For a facility at node  $j$ , consumers along path  $q$  consider the preferred distance ( $d_{jq_p}$ ) between node  $j$  and the preferred location, and the deviation distance ( $d_{jq_d}$ ) between node  $j$  and the predetermined path. The value of  $G$  is calculated based on the two types of distances. Following the suggestion of Berman, Bertsimas, and Larson (1995) and Zeng, Hodgson, and Castillo (2007), I set  $G_{qj} = f_q e^{-\alpha(d_{jq_p} + \beta d_{jq_d})}$ , where  $\alpha$  and  $\beta$  are scaling constants,  $\beta$  the trade-off between the relative importance of the two types of distance. The value of  $\beta$  may be greater than 1 because the deviation distance cost may include extra fees such as gasoline purchase.

Suppose the preferred location is at the destination, I consider the generalized deviation case 2 (preference) with  $\alpha = 0.5$  and  $\beta = 2.0$ . Using my example, node 1 provides  $2e^{-0.5 \times (6+0)} = 0.10$  units of objective contribution from path 1 and  $e^{-0.5 \times (4+2 \times 7)} = 0.00$  units of objective contribution from path 3. Node 2 provides  $2e^{-0.5 \times (5+2 \times 2)} = 0.02$  units of objective contribution to path 1. The optimal location for a single facility ( $p = 1$ ) is at node 7 which provides 4.00 units of objective contribution on paths 1 and 4, 0.03 units of objective contribution on path 3, and 0.00 units of objective contribution on path 2.

### 3.4.7. Multi-Counting Cases

FILM avoids flow cannibalization; that is, wasteful redundant flow-interception. In some situations, however, redundant flow intercepting may be advantageous. For example, multiple exposures to the same advertising billboard can increase its impact. In this situation, the benefit of exposure to the message depends not only on the existence of a facility located on the path but also on the number of facilities located on the path. Averbakh and Berman (1996) developed a nonlinear integer model to locate advertising billboards on the links of a network with multi-counting and diminishing returns to scale. Hodgson and Berman (1997) formulated a linear integer model to locate advertising billboards aimed at maximizing the benefit of exposure to the message. I have modified their formulation to make the relationship to GFIM more apparent.

$$\text{Maximize: } Z^G = \sum_{q \in Q} \sum_{j \in M} f_q b_j X_{qj} \quad (12)$$

s.t.

$$\sum_{j \in M} X_{qj} \leq 1, \forall q \in Q \quad (13)$$

$$jX_{qj} \leq \sum_{k \in q} Y_k, \forall q \in Q, j \in M \quad (8C)$$

$$\sum_{k \in K} Y_k = p \quad (14)$$

$$X_{qj} \in \{0, 1\}, \forall q \in Q, j \in q \quad (15)$$

$$Y_j \in \{0, 1\}, \forall j \in J \quad (16)$$

In comparison with GFIM, the new terms are:

$M = \{1, \dots, m\}$ , where  $m$  is the maximum number of effective exposures;  $m+1$  will not increase total effectiveness

$b_j$  = the benefit of the  $j$ th exposure on a path

$$X_{qj} = \begin{cases} 1 & \text{if } f_q \text{ is exposed exactly } j \text{ times} \\ 0 & \text{otherwise} \end{cases}$$

$K$  = the set of all links that are potential billboard locations indexed by  $k$

$$Y_k = \begin{cases} 1 & \text{if there is a facility located at link } k \\ 0 & \text{otherwise} \end{cases}$$

The objective function (12) maximizes the benefit of exposure to advertising billboards. Constraint (13) ensures that at most one value  $X_{qj}$  is chosen. Constraint (8C) ensures the correct value of  $b_j$  is chosen. Constraint (14) ensures that the desired number of billboards is located. We note that this formulation is structurally identical to GFIM with input data  $G_{qj} = f_q \times b_j$ ,  $N_q = M$ ,  $J = K$ , and with constraint (8C) replacing (8A). Constraints (15) and (16) are the standard integrality conditions.

In my 7-node example, with  $b_1 = 1$ ,  $b_2 = 1.6$ , and  $b_3 = 2$  the optimal location for a single facility is on link 1 which intercepts only the flow along path 1; the optimal locations for three facilities are on links 1, 2, and 8, which provide 3.2 units of benefit from path 1 and 2.0 units of benefit from path 2.

### 3.4.8. Group-Counting Cases

In some situations, a group of facilities must be open to serve a consumer. For instance, the limited range of vehicles implies that certain paths may only be refuelled by a group of adequately spaced facilities. Kuby and Lim (2005, 2007) developed a flow refuelling location model (FRLM) to locate  $p$  refuelling stations on a network aimed at maximizing the number of vehicles that can be successfully refuelled. FRLM departs from FILM by specifying eligible groups of facilities, able to cover or refuel a path, rather than a single facility. GFIM can solve group-counting flow-interception problems by adding constraints (17) and (18) to the GFIM model as follows.

$$\sum_{j \in q} X_{qj} \leq \sum_{h \in q} V_h, \forall q \in Q \quad (17)$$

$$V_h \leq Y_j, \forall h \in H, j \in h \quad (18)$$

Where,

$H$  = set of all potential facility groups indexed by  $h$

$$V_h = \begin{cases} 1 & \text{if all facilities in group } h \text{ are open} \\ 0 & \text{otherwise} \end{cases}$$

Constraint (17) ensures that at least one eligible group must be open for path  $q$

to be refuelled. Constraint (18) holds  $V_h$  to zero unless all the facilities in group  $h$  are open. I note that the original FRLM is a location, not a location-allocation model; it cannot consider at which facilities the flow is intercepted. Therefore, FRLM cannot further consider capacity, locational and proximity preference or deviation: critical issues in location analysis. GFIM, can, however account for these critical issues.

In my 7-node example, following the suggestions of Kuby and Lim (2005), I set the vehicle range as 3 units and the remaining fuel range at the origin as 2 units. The algorithm for determining the groups of nodes that can refuel a path is found in Kuby and Lim (2005). The groups required for round-trip paths 1 through 4 are  $\{(3, 5)\}$ ,  $\{(2, 6), (3, 6)\}$ ,  $\{(4, 6), (5, 6)\}$ , and  $\{(7)\}$ , respectively. The total flow refuelled for one facility is 2 units of flow: only path 4 is refuelled at node 7. The optimal locations for three facilities are at nodes 3, 5, and 6 where only path 4 cannot be refuelled. The optimal locations for four facilities are at nodes 3, 5, 6, and 7 where all 4 paths are refuelled.

### 3.4.9. Other Potential Cases

Recall that the value of  $G_{qj}$  represents an individual consumer's consideration of a specific facility; it enables GFIM to consider consumers' many specific desires and needs, including locational and proximity preferences, deviation from predetermined journeys, and capacity issues. In my example of the generalized deviation case 2, suppose that an individual path 1 consumer wishes to drop by his/her grandparents' house at node 6 and that this will add 0.5 units of objective contribution to his/her trip.

This individual consumer's specific desire can be satisfied by setting  $G_{16} = 2 \times e^{-0.5 \times 1} + 0.5 = 1.72$ . By simply setting new values of  $G_{qj}$  for individual consumers' specific desires, GFIM might be applied to other potentially useful location problems such as competitive flow-interception problems. Berman and Krass (1998, 2002), and Wu and Lin (2003) studied competitive, probabilistic flow-interception problems, but there is no competitive, deterministic flow-interception study reported in the literature.

### 3.5. Computational Experimentation and Comparison

GFIM is a *location-allocation* model designed to solve many different kinds of flow-interception problems, so we might expect it to be much more complex and computationally demanding than a simple *location* model. In this section, I compare and analyze the solution times of FILM and special cases (represented by the protection case) of GFIM with real-world examples. There are two real-world examples reported in the flow-interception literature. A recent one is the 2001 afternoon peak traffic network for the city of Edmonton in Canada, comprising 16,488 flow paths, 290 zones, 1,746 nodes and 4,606 links described by Zeng, Hodgson, Castillo (2007). An earlier one is the 1989 morning peak traffic network for the city of Edmonton in Canada, comprising 23,958 flow paths, 703 nodes, and 2,198 links (Hodgson, Rosing, and Storrier 1996; Hodgson and Berman 1997; Hodgson, Rosing, and Storrier 1997). The original data for these two networks were provided by the City of Edmonton Transportation and Streets Department, who state that their data have been produced according to industry standards and that their forecasting model is

highly recognized throughout North America. Flows in these two data sets are vehicle flows among traffic zones in the full Edmonton area. Figure 3-4 illustrates the percent of flows along 4,606 links and the percent of total flows through 290 zones in the 2001 afternoon peak traffic network. Figure 3-5 illustrates the 2198 links and the percent of total flows through 177 zones in the 1989 morning peak traffic network.

Both FILM and GFIM were coded in the AMPL language and solved with version 9.1.0 of the ILOG-CPLEX optimizer on a 2.80 GHz Intel Pentium processor with 1024 MB of RAM in Microsoft Windows 2000. I specified that FILM and GFIM be solved using the default primal algorithm and the dual simplex algorithm, respectively; our experience verified that these choices provided the most efficient solutions. Tables 3-3 and 3-4 provides the CPU minutes, branch-and-bound nodes, mixed-integer programming (MIP) simplex iterations of FILM and GFIM at  $Y_j =$  binary (FILM and GFIM),  $0 \leq X_q \leq 1$  (FILM), and  $0 \leq X_{qj} \leq 1$  (GFIM) for these two examples. Table 3-3, 3-4, Figures 3-6, and Figure 3-7 reveal drastic contrasts between the solution characteristics of the two models. At small size problem ( $p < 13$  for afternoon data,  $p < 11$  for morning data), CPLEX optimally solve FILM more efficiently than the protection case of GFIM. However, at large size problem ( $p > 14$  for afternoon data,  $p > 10$  for morning data), CPLEX optimally solve the protection case of GFIM more efficiently than FILM. In both examples, FILM's computation times rise sharply as  $p$  increases, GFIM's rise much more slowly, approximately linearly (Figures 3-6 and 3-7). In the 2001 afternoon traffic network, CPLEX optimally solves the protection case of GFIM more efficiently than FILM for  $p > 14$ , with a much lower increase in the numbers of branch-and-bound nodes and MIP simplex iterations. In the 1989 morning traffic network, CPLEX optimally solves the protection case of GFIM much more efficiently than FILM for  $p > 10$ , with a much lower increase in the numbers of branch-and-bound nodes and MIP simplex iterations. This runs counter to our intuition that as a generalized, location-allocation model, GFIM would be less efficient than FILM.

The reasons why FILM's computation times, branch-and-bound nodes and MIP simplex iterations rise sharply as  $p$  increases and why GFIM's rise much more slowly, approximately linearly are very interesting and complex. I will work on a formal mathematical proof in the future. Here, I provide some simple explanations of the computational behaviour of FILM and GFIM.

The formulations of FILM and GFIM can be modified with the following new terms to make their relationships to the classic formulation of the  $p$ -median model (ReVelle and Swain 1970) more apparent:  $Q = I$ ;  $f_q = w_i$ ;  $N_q = N_i$ ;  $G_{qj} = w_i \times d_{ij}$ ;  $X_q = X_i$ ;  $X_{qj} = X_{ij}$ ;  $Y_j = X_{jj}$ ; and  $j \in q \rightarrow j \in N_i$ . The constraints (2) and (8A) are, respectively, transformed to constraints (2D) and (8D) as follows.

$$X_i - \sum_{j \in N_i} X_{jj} \leq 0, \forall i \in I \quad (2D)$$

$$X_{ij} - X_{jj} \leq 0, \forall i \in I, j \in N_i, i \neq j \quad (8D)$$

Constraint (8D) at  $N_i = J$  is exactly the Balinski constraint described by Morris (1978), whereas (2D) is structurally very similar to the Efromson Ray type constraint.

Several researchers (e.g., Divas and Ray 1969; Williams 1974; Morris 1978; Rosing, ReVelle, and Rosing-Vogelaar 1979; ReVelle 1993; Church 2003) determined that the Balinski type constraint is “integer-friendly,” meaning that the relaxed formulation produces a high proportion of integer solution variables. There are many fewer Efromson Ray constraints in FILM, but this formulation is not “integer-friendly,” and requires time-consuming branch and bound iterations.

Williams (1974) concluded, “Obviously the second formulation (a Balinski type constraint) is far superior to the first (an Efromson Ray type constraint) because of the unimodularity property.” He explained, “the second formulation (a Balinski type constraint) has the property that each restraint now contains exactly one coefficient equal to +1 and exactly one coefficient equal to -1. The dual problem will therefore have exactly one +1 and exactly one -1 in each column. [...] The matrix of both this problem and its dual are therefore unimodular.” ReVelle (1993) indicated that integer or mixed-integer programs which have unimodular constraint matrices are “integer-friendly.” Based on practical experience, Rosing, ReVelle, and Rosing-Vogelaar (1979) and Church (2003) concluded that not all constraints need be of the Balinski type to make a model “integer-friendly.” Our research is the first to experimentally verify these “integer-friendly” properties in flow-interception location models (Tables 3-3 and 3-4; Figures 3-6 and 3-7).

Now suppose we consider GFIM to be a re-formulation of FILM. To facilitate comparison between the two formulations, we will only use the  $X_{qj}$  variables. That way, both formulations use the same decision variables. The two formulations are:

$$\begin{array}{ll} \text{GFIM} & \text{FILM} \\ \text{Maximize:} & \sum_{q \in Q} f_q \sum_{j \in N_q} X_{qj} \end{array} \quad \text{Maximize:} \quad \sum_{q \in Q} f_q \sum_{j \in N_q} X_{qj} \quad (19)$$

$$\begin{array}{ll} \text{s.t.} & \text{s.t.} \\ \sum_{j \in N_q} X_{qj} \leq 1, \quad \forall q \in Q & \sum_{j \in N_q} X_{qj} \leq 1, \quad \forall q \in Q \end{array} \quad (20)$$

$$\begin{array}{ll} X_{qj} \leq Y_j, \quad \forall q \in Q, j \in N_q & \sum_{j \in N_q} X_{qj} \leq \sum_{j \in N_q} Y_j, \quad \forall q \in Q \end{array} \quad (21)$$

$$\sum_{j \in J} Y_j = p \quad \sum_{j \in J} Y_j = p \quad (22)$$

$$0 \leq X_{qj} \leq 1, \quad \forall q \in Q, j \in N_q \quad 0 \leq X_{qj} \leq 1, \quad \forall q \in Q, j \in N_q \quad (23)$$

$$Y_j \in \{0,1\}, \quad j \in J \quad Y_j \in \{0,1\}, \quad j \in J \quad (24)$$

From a side-by-side comparison, we see that FILM is obtained from GFIM by summing the second set of constraints over  $j$ , seen in constraint (21). Considering the linear programming relaxations of both formulations, we see that any solution that is feasible to GFIM will also be feasible to FILM, but the reverse is not true. Therefore, the feasible set for the linear programming relaxation of GFIM is contained in the feasible set for the relaxation of FILM. In other words, the GFIM formulation is tighter than the FILM formulation. A simple example illustrates why the GFIM constraints are tighter than the FILM constraints. Consider the constraints  $0 \leq X \leq 1$ ,

and  $0 \leq Y \leq 1$  which specify the unit square. Moving from GFIM to FILM is like replacing these constraints with  $X \geq 0, Y \geq 0, X + Y \leq 2$ , which specifies a triangle that contains the unit square and has twice the area of the unit square.

**Table 3-3: Computational comparison** (afternoon peak traffic, Edmonton)

$p$	The classic FILM model			Protection case, GFIM		
	Minute	Branch	MIP iterations	Minute	Branch	MIP iterations
1	0.1	0	17185	2.8	0	323080
2	0.3	0	22913	5.6	0	331198
3	0.2	0	18928	6.9	0	336505
4	0.4	0	24582	6.7	0	328942
5	0.3	0	24687	7.5	0	326573
6	0.4	0	25261	10.9	0	336977
7	0.4	0	25583	9.7	0	332557
8	0.5	0	26027	11.4	0	336150
9	0.6	10	26589	14.2	2	339775
10	0.9	111	32082	16.0	0	350686
11	1.0	64	33439	18.6	0	356895
12	4.0	1028	94947	18.3	0	342589
13	16.8	3073	356459	22.1	0	346294
14	3.6	420	79278	27.0	0	350418
15	69.9	15435	1451224	27.4	0	346564
16	64.2	5657	1285721	33.2	2	346953
17	77.8	7896	1615058	38.3	2	355691
18	69.7	8237	1372720	36.9	0	344834
19	105.3	12240	2073856	42.6	0	351875
20	364.1	28172	7333811	56.7	4	362394
21	1008.4	192555	18647468	64.5	4	368355
22				60.2	4	362409
23				64.7	2	362197
24				79.2	4	372695
25				92.1	8	386159
26				96.9	4	377892
27				104.7	2	386704
28				143.2	24	422796
29				169.4	20	446673
30				194.5	18	463703

*Minute: CPU Minutes; Branch: Branch-and-bound nodes;  
MIP: MIP simplex iterations*

**Table 3-4: Computational comparison** (morning peak traffic, Edmonton)

$p$	The classic FILM model			Protection case, GFIM		
	Minute	Branch	MIP iterations	Minute	Branch	MIP iterations
1	0.3	0	25728	4.9	0	445691
2	0.5	0	27328	9.8	0	437754
3	0.2	0	22412	8.8	0	413171
4	0.3	0	22918	11.4	0	409773
5	0.3	0	22297	9.8	0	387256
6	0.4	0	22827	14.0	0	391095
7	0.5	0	25023	15.4	0	380291
8	1.6	48	40740	20.2	0	395562
9	2.7	54	53139	18.6	0	381923
10	3.0	209	58863	24.8	0	391419
11	216.7	13253	3426214	39.3	2	425254
12	1043.5	123291	15331821	46.8	2	425289
13				47.0	2	425765
14				53.4	2	433251
15				48.1	0	414934
16				56.1	0	416713
17				46.3	0	406230
18				61.6	2	425265
19				78.1	2	442567
20				93.6	10	441506
21				106.7	14	461006
22				120.8	13	466902
23				108.3	4	448141
24				116.0	6	452277
25				125.2	4	451776
26				122.3	11	447947
27				133.7	6	446673
28				154.4	14	454800
29				195.0	14	468113
30				194.8	13	471270

*Minute: CPU Minutes; Branch: Branch-and-bound nodes;  
MIP: MIP simplex iterations*

### 3.6. Conclusion

In this chapter I introduce GFIM, a generalized and efficient model for locating facilities on a network with flow-based demand. First, GFIM is a generalized model for effectively solving most current and future flow-interception location problems. Most current flow-interception location problems can be solved by simple parameter manipulations in GFIM's input. Additional flow-interception problems can be solved by manipulating or adding constraints to GFIM.

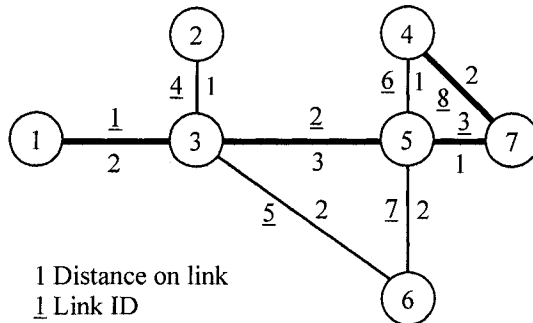
Furthermore, several critical considerations in flow-interception models – such as deviation from predetermined journeys, locational and proximity preferences, and capacity issues – can be handled within the single framework. Secondly, GFIM is an effective and efficient model for locating facilities on a network with flow-based demand. In comparison with FILM, which is a *location* model without having “integer-friendly” properties, GFIM is a *location-allocation* model having “integer-friendly” properties. These “integer-friendly” properties enable a standard optimization engine such as ILOG-CPLEX to optimally solve GFIM more efficiently than the classic FILM. The “location-allocation” property, together with the “integer-friendly” properties, enables GFIM to provide a much broader, realistic approach to problems in the private and public sectors than do other current flow-interception models. In short, in comparison with current flow-interception location models reported in the literature, the use of GFIM significantly reduces the solution burden on decision makers, without degrading solution quality. GFIM clearly provides a standardized benchmark for current and future models in the academic literature.

## References

- Averbakh, I., O. Berman. 1996. Locating flow-capturing units on a network with multi-counting and diminishing returns to scale. *European Journal of Operational Research* **91** 495-506.
- Berman, O., D. Bertsimas, R. C. Larson. 1995. Locating discretionary service facilities, II: maximizing market size, minimizing inconvenience. *Operations Research* **43** 623-632.
- Berman, O., D. Krass. 1998. Flow intercepting spatial interaction model: a new approach to optimal location of competitive facilities. *Location Science* **6** 41-65.
- Berman, O., D. Krass. 2002. Locating multiple competitive facilities: spatial interaction model with variable expenditures. *Annals of Operations Research* **111** 197-225.
- Berman, O., R. C. Larson, N. Fouska. 1992. Optimal location of discretionary service facilities. *Transportation Science* **26**, 201-211.
- Church, R. L. 2003. COBRA: a new formulation of the classic  $p$ -median location problem. *Annals of Operations Research* **122** 103-120.
- Davis, P., T. Ray. 1969. A branch-bound algorithm for the capacitated facilities location problem. *Naval Research Logistics Quarterly* **16** 331-344.
- Fourer, D., D. M., Gay, B.W., Kernighan. 2002 *AMPL: A Modeling Language for Mathematical Programming*. Second Edition. Thomson.
- Gendreau, M., G. Laporte, I. Parent. 2000. Heuristics for the location of inspection stations on a network. *Naval Research Logistics* **47** 287-303.
- Hodgson, M. J. 1990. A flow-capturing location-allocation model. *Geographical Analysis* **22** 270-279.
- Hodgson, M. J., K. E. Rosing, J. Zhang. 1996. Locating vehicle inspection stations to protect a transportation network. *Geographical Analysis* **28** 299-314.
- Hodgson, M. J., O. Berman. 1997. A billboard location model. *Geographical & Environmental Modeling* **1** 25-45.



- Horner, M. W., S. Groves. 2007. Network flows-based strategies for identifying rail park-and-ride facility locations. *Socio-Economic Planning Sciences* **41** 255-268.
- Kuby, M., S. Lim. 2005. The flow-refuelling location problem for alternative-fuel vehicles. *Socio-Economic Planning Sciences* **39** 125-145.
- Kuby, M., S. Lim. 2007. Location of alternative-fuel stations using the flow-refueling location model and dispersion of candidate sites on arcs. *Networks and Spatial Economics* **7** 129-152.
- Mirchandani, P.B., R. Rebello, A. Agnetis. 1995. The inspection station location problem in hazardous material transportation: some heuristics and bounds. *INFOR* **33** 100-113.
- Morris, J.G., 1978. On the extent to which certain fixed-charged depot location problems can be solved by LP. *Journal of the Operational Research Society* **29** 71-76.
- ReVelle, C. 1993. Facility siting and integer-friendly programming. *European Journal of Operational Research* **65** 147-158.
- ReVelle, C.S., R. Swain. 1970. Central facilities location. *Geographical Analysis* **2** 30-42.
- Rosing, K.E., C.S. ReVelle, H. Rosing-Vogelaar. (1979). The  $p$ -median and its linear programming relaxation: an approach to large problems. *Journal of the Operational Research Society* **30** 815-823.
- Williams, H. P., 1974. Experiments in the formulation of integer programming problems. *Mathematical Programming Study* **2** 180-197.
- Wu, Tai-His, Jen-Nan Lin. 2003. Solving the competitive discretionary service facility location problem. *European Journal of Operational Research* **144** 366-378.
- Zeng, W., M. J. Hodgson, I. Castillo. 2007. The pickup problem: consumers' locational preferences in flow interception. Accepted on February 10, 2007 for publication in the special issue of *Geographical Analysis*, edited by John R. Current, Richard L. Church, and Mark S. Daskin (in memory of Charles S. ReVelle, a leader in location science)
- Zhang, J., M. J. Hodgson, E. Erkut. 2000. Using GIS to assess the risks of hazardous materials transport in networks. *European Journal of Operational Research* **121** 316-329.

**Figure 3-1: A test 7-node network****Figure 3-2: The GFIM model (GFIM.mod)**

```

-----
param Q >=0;
param J >=0;
param p >=0;
param C >=0;
param G {q in 1..Q,j in 1..J};
var Y {1..J} binary;
var X {q in 1..Q,j in 1..J: G[q,j]>0} binary;
maximize Z:
    sum {q in 1..Q} sum {j in 1..J: G[q,j]>0} G[q,j]*X[q,j];
subject to constraint_7 {q in 1..Q} :
    sum {j in 1..J: G[q,j]>0} X[q,j] <=1.0;
subject to constraint_8B {j in 1..J}:
    sum {q in 1..Q: G[q,j]>0} G[q,j]*X[q,j] <=C*Y[j];
subject to constraint_9:
    sum{j in 1..J} Y[j] <= p;
-----

```

**Figure 3-3: Data for the GFIM model (GFIM.dat)**

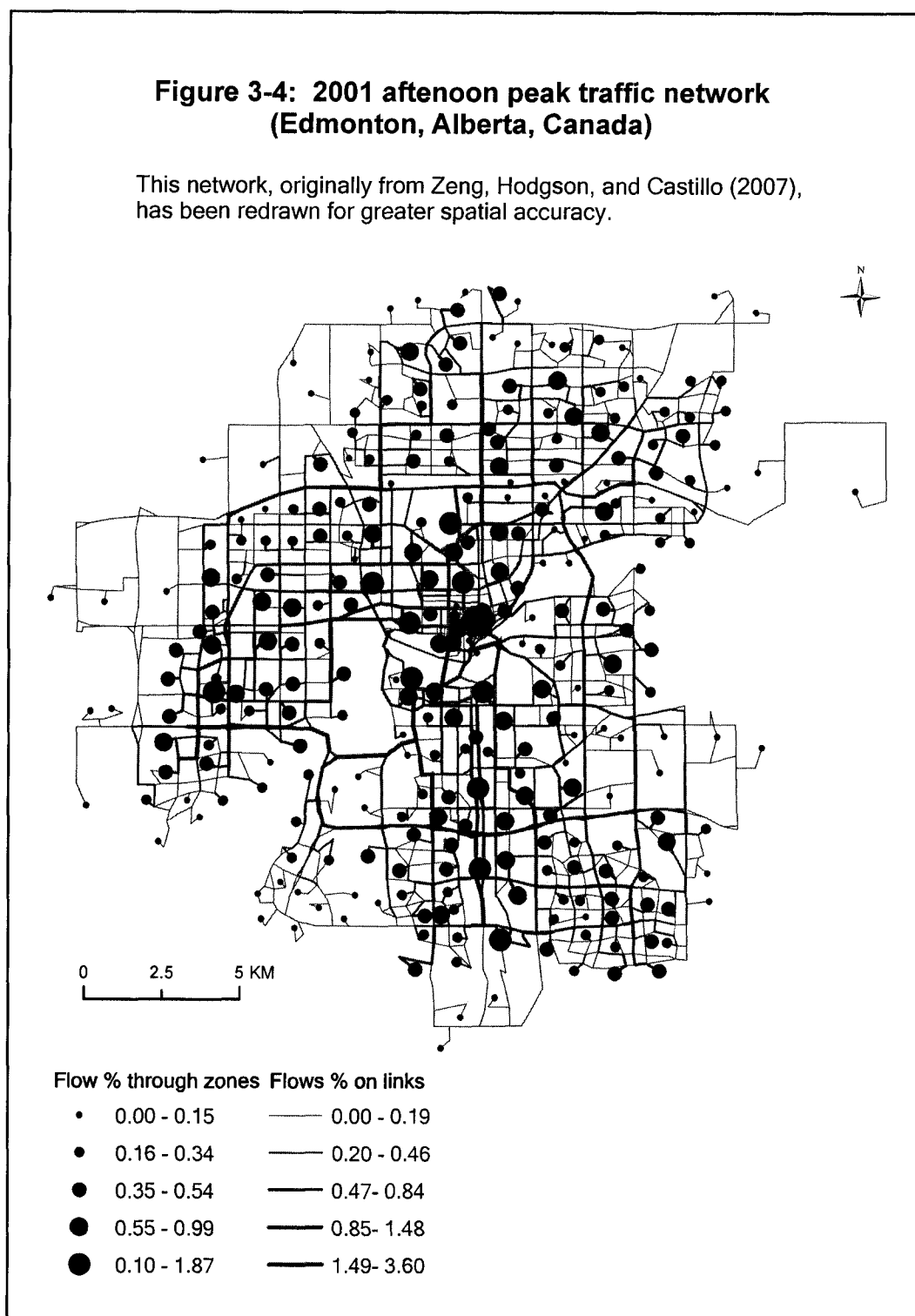
```

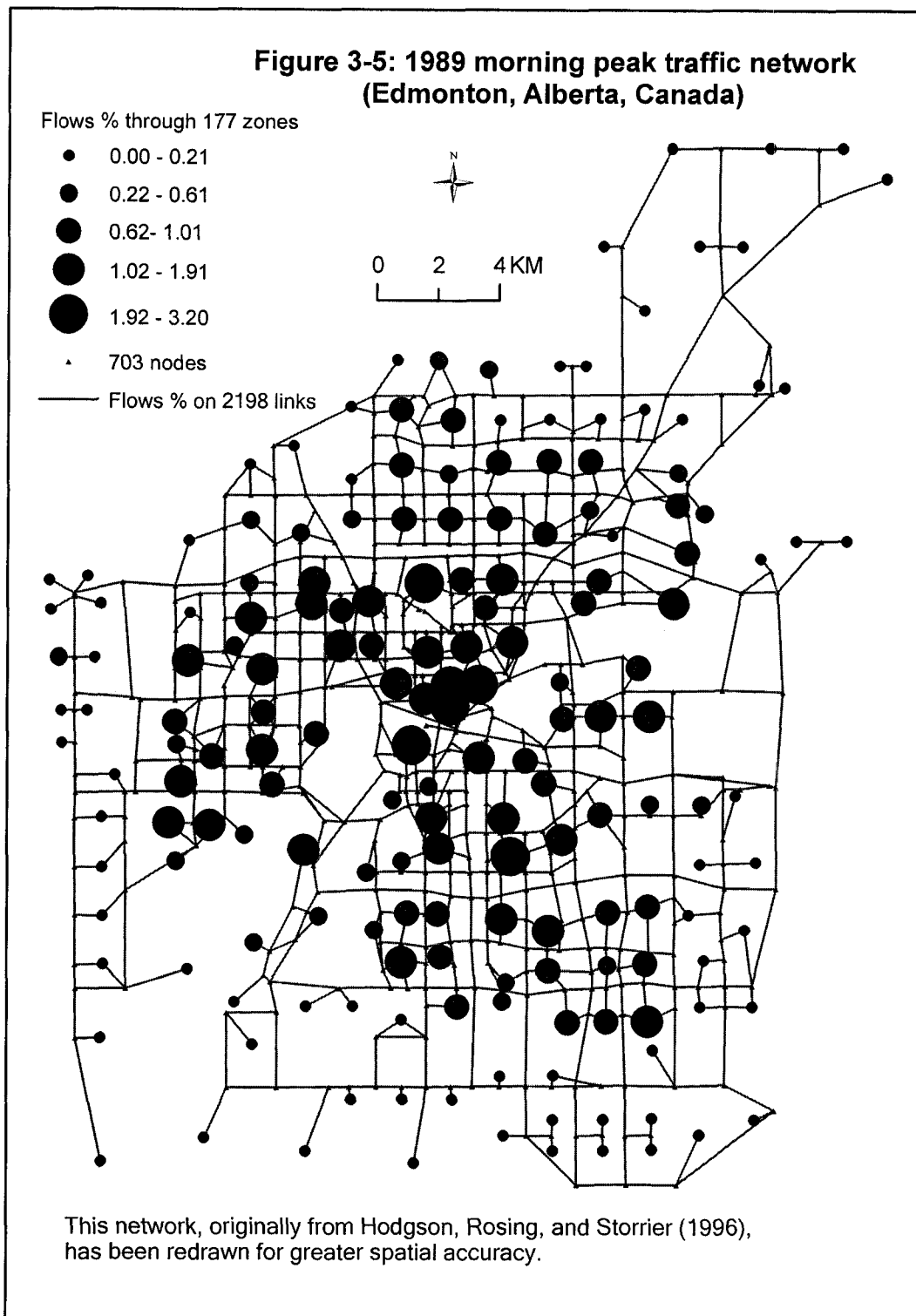
-----
param Q = 4;
param J = 7;
param C = 2.6;
param p := 2;
param G default 0: # matrix of G
    1    2    3    4    5    6    7:=
1  2.00  0.74  2.00  0.74  2.00  1.22  2.00
2  0.14  1.00  1.00  0.08  0.22  1.00  0.08
3  0.03  0.08  0.22  1.00  1.00  1.00  0.37
4  0.02  0.04  0.10  2.00  2.00  0.28  2.00;
-----

```

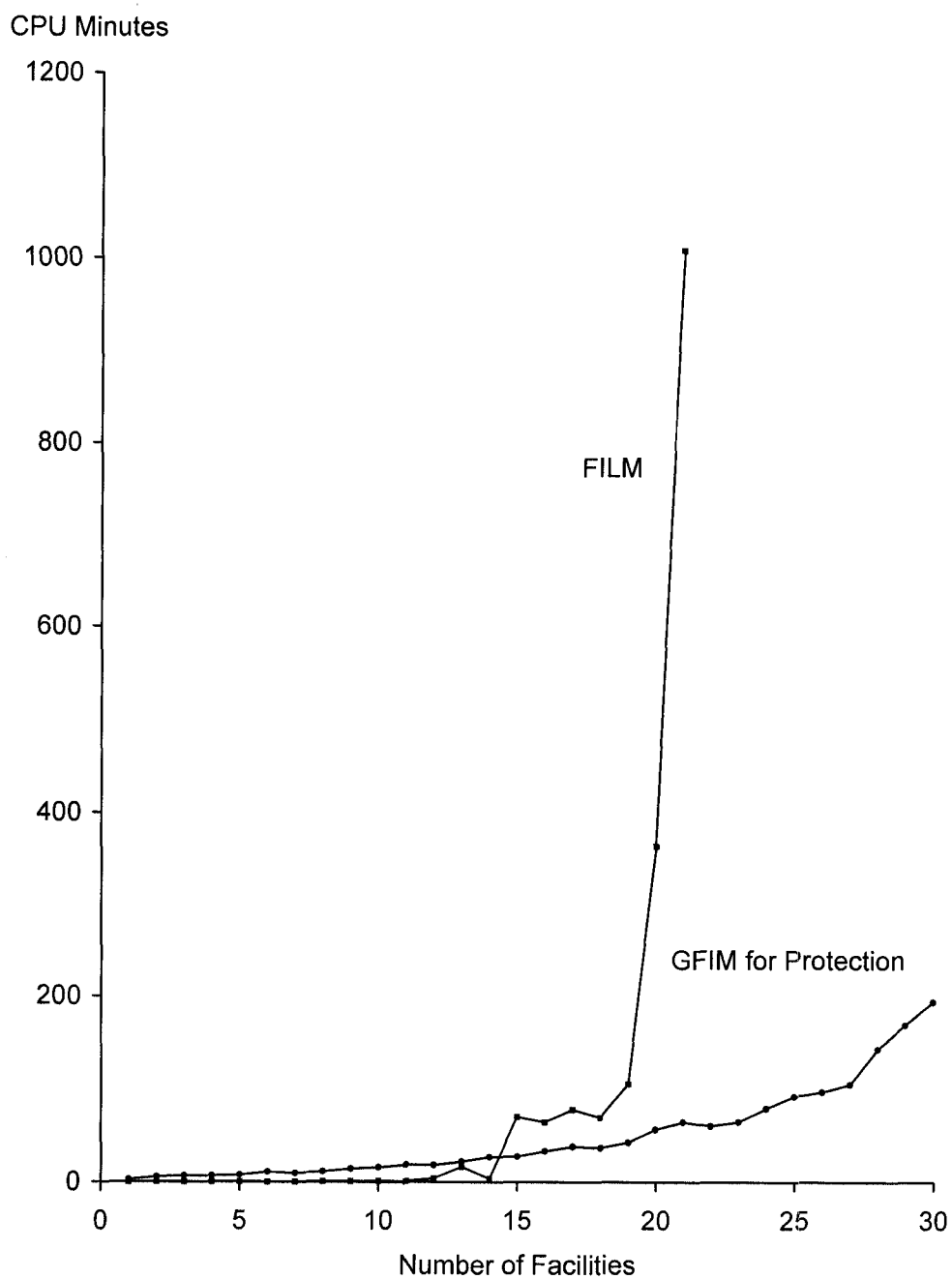
**Figure 3-4: 2001 afternoon peak traffic network  
(Edmonton, Alberta, Canada)**

This network, originally from Zeng, Hodgson, and Castillo (2007), has been redrawn for greater spatial accuracy.

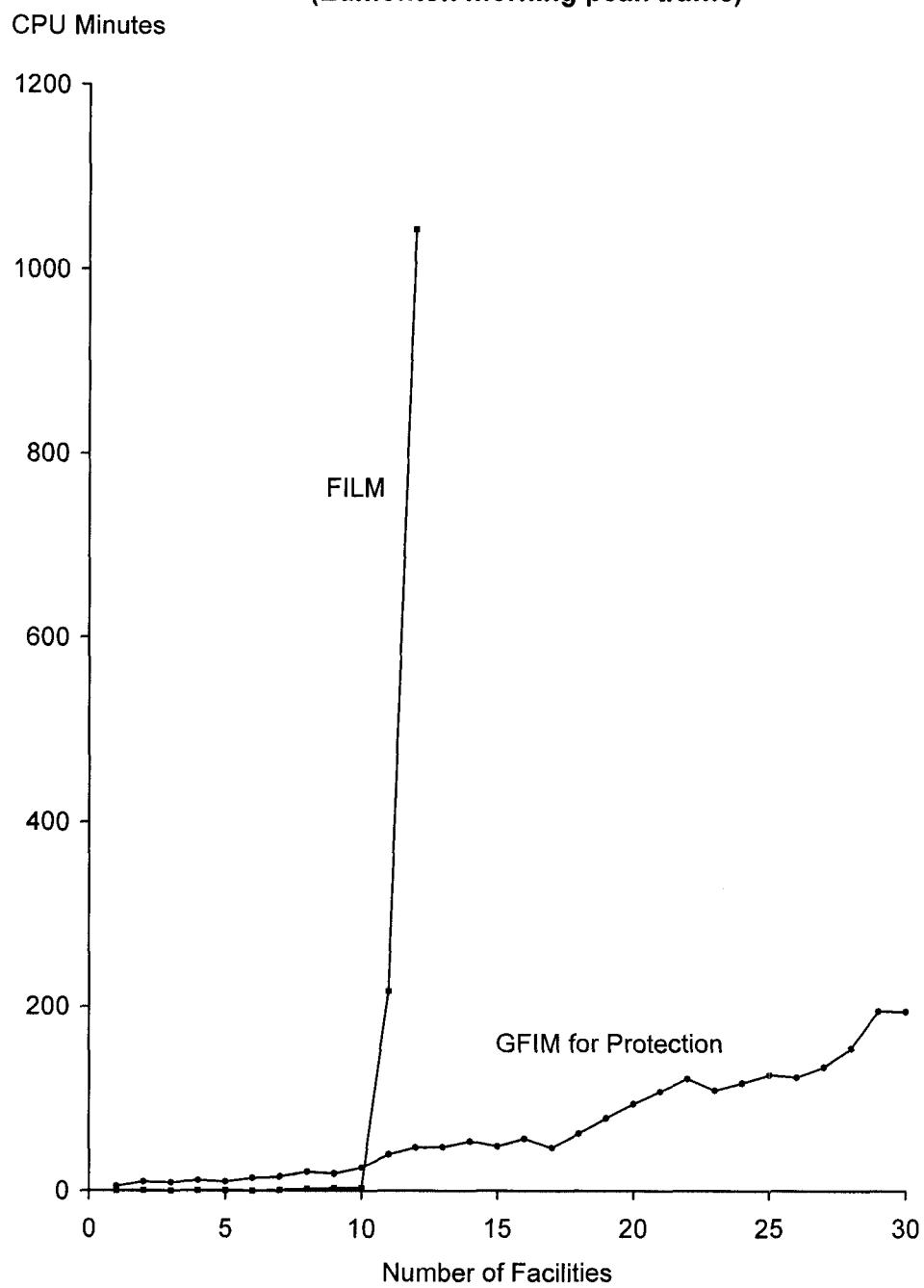




**Figure 3-6: CPU minutes of FILM and GFIM  
(Edmonton afternoon peak traffic)**



**Figure 3-7: CPU minutes of FILM and GFIM  
(Edmonton morning peak traffic)**



## Chapter 4 \*

### A New Type of Consumer and an Efficient Strategy for Unifying Network Location Models

**Summary:** Traditional network location theory assumes that consumers patronize facilities as close as possible to demand points (Type A consumer). Flow-interception location theory assumes that consumers patronize facilities near or on these predetermined paths (Type B consumer). In the real world, however, if a facility is close to their homes, consumers may patronize it; if a facility is close to their predetermined trips consumers may also patronize it. I call these consumers Type C consumers. Most people in the real-world are Type C consumers – they choose a facility based on its greater convenience to either their home or their travel path. The literature has neglected Type C consumers. My examples, using afternoon peak traffic data for the city of Edmonton in Canada, show that the solutions identified by Type C consumers are more sensible than solutions identified by Type A and Type B consumers.

Location researchers have traditionally proposed models for different types of consumers in isolation and tend to introduce changes in objective functions and/or assumptions by developing new models. This article introduces a generalized and efficient strategy for unifying consumer types and location models (GSUM). Using GSUM principle, numerous location models can be unified. For instance, a generalized location-allocation model is formulated to effectively and efficiently encompass at least 60 existing models, including the  $p$ -median, maximal covering location model, flow-interception location model, and numerous variants of these models.

**\* A version of this chapter has been submitted for publication.**

**Zeng, Weiping.** 2007. A new type of consumer and an efficient strategy for unifying network location models. Submitted to the *European Journal of Operational Research* on April 28, 2007, submission No:EJOR-D-07-00630, under review.

#### 4.1. Introduction

Almost every enterprise in the private and public sectors faces the problem of strategically locating facilities to provide services to consumers on a transportation network. Traditional network location theory (e.g., the  $p$ -median model formulated by ReVelle and Swain in 1970) assumes that demand for service occurs at fixed points (e.g., home) and that consumers patronize facilities only near demand points (Type A consumer). Flow-interception location theory assumes that demand for services is expressed by flows travelling on predetermined origin-destination (OD) paths (e.g., daily commute between home and workplace) and that consumers patronize facilities near or on predetermined OD paths (Type B consumer). In the real-world, however, if a facility is close to their home consumers may patronize it; if a facility is close to their predetermined trips consumers may also patronize it. I call these consumers Type C consumers. Most people in the real-world are Type C consumers – they choose a facility based on its greater convenience to either their home or their travel path. Except for a brief mention by Berman (1997), the literature has neglected Type C consumers. My examples, using afternoon peak traffic data for the city of Edmonton in Canada, show that solutions identified by Type C consumers are more expedient than solutions identified by Type A and Type B consumers.

The number of combinations of three types of consumers is seven consumer scenarios:  $\{A\}$ ,  $\{B\}$ ,  $\{A, B\}$ ,  $\{C\}$ ,  $\{A, C\}$ ,  $\{B, C\}$  and  $\{A, B, C\}$ . Conventional network location models have been proposed exclusively for solving problems in scenario  $\{A\}$  where all demands for service are from Type A consumers. Flow-Interception Location Models (FILM) are used exclusively for solving problems in scenario  $\{B\}$  where all demand for service are from Type B consumers. The Type B consumer was first identified by Hodgson (1990) and Berman, Larson, and Fouska (1992) and has received considerable research interest, represented by about 30 location models spanning about 40 academic publications. Their applications have covered the strategic location of automatic teller machines and convenience stores (Berman, Hodgson, and Krass 1995; Hodgson, Rosing, and Storrier 1996; Wang, Batta, and Rump 2002; Turner 2006), advertising billboards (Averbakh and Berman 1996; Hodgson and Berman 1997), vehicle inspection stations (Hodgson, Rosing, and Zhang 1996; Gendreau, Laporte, and Parent 2000; Miller and Shaw 2001), park-and-ride facilities (Horner and Grove 2007), gasoline stations and refuelling facilities (Kuby and Lim 2005, 2007; Kuby 2006; Zeng, Castillo, and Hodgson 2007; Upchurch, Kuby, and Lim 2006), pickup and fast food outlets (Zeng, Hodgson, and Castillo 2007), and cellular base stations (Erdemir et al. 2006). In addition, Berman, Bertsimas, and Larson (1995) developed several models to address generalizations of FILM where flows are allowed to deviate from predetermined origin-destination paths. The reader is referred to Berman, Hodgson, and Krass (1995) for more detailed reviews of these models. Several researchers (Hodgson and Rosing 1992; Hodgson, Rosing, and Storrier 1997; Berman 1997; Erdemir et al. 2006) have developed several disparate models exclusively for solving problems in scenario  $\{A, B\}$ . Known location models consider scenario  $\{A\}$ ,  $\{B\}$ , or  $\{A, B\}$ .

Location researchers have traditionally proposed models for the three scenarios



in isolation and tend to introduce changes in objective functions and/or assumptions by developing new models. This has created numerous disparate models, each viewed as requiring a somewhat different solution method, thus impeding the development of standardized software that would encourage widespread use of location models in real-world, strategic decision-making processes.

Another fundamental contribution of this chapter is the introduction of GSUM, a generalized and efficient strategy for unifying consumer types and network location models. First, GSUM transforms Type A and Type C consumers into Type B consumers. Thus, many location models and theories based on a particular consumer type can be unified, for example, the two divergent flow-interception and conventional network location theories can be unified. Second, GSUM enables current and future location models to handle consumer types and many objective functions through exogenous parameters. Using GSUM principle, I formulate a Generalized Location-Allocation Model (GLAM) that allows us to effectively encompass at least 60 existing models, including the maximal covering location model (MCLM) (Church and ReVelle 1974),  $p$ -median, FILM, and numerous variants of these models. Third, GSUM reduces the location problem size in terms of the number of variables and constraints needed through the use of a sorted strings data structure (e.g., Densham and Rushton 1992; Sorensen and Church 1996) that offers great benefits in memory savings and solution times. Actually, GSUM becomes more effective as the number of located facilities to be located increases.

In short, this chapter aims to show that the Type C consumer and GSUM have the potential to substantially improve the merits and applicability of location modeling, while simultaneously significantly reducing the solution burden on decision makers. GLAM is an excellent example to showcase the effectiveness of GSUM and the Type C consumer on many location problems. The GLAM model itself is an interesting, but not necessarily the primary, contribution of this chapter.

The remainder of the chapter is organized as follows. Section 2 introduces GSUM. Section 3 describes the formulation of GLAM. Section 4 shows that GSUM enables GLAM and other location models to unify a wide variety of current and future location models. Section 5 shows the importance of Type C consumers in location modeling with real-world examples. The final section offers major conclusions.

#### **4.2. A Generalized and Efficient Strategy for Unifying Consumer Types**

A network location problem may be characterized as identifying the placement of  $p$  facilities on a network to serve a spatially distributed set of demand nodes in a manner that optimizes a designated objective function. In general, the objective function consists of terms involving distances or transportation costs (referred to hereafter simply as distances) between consumers and facilities. Type A consumers consider distance from demand points to facility points. Type B consumers consider “distance” of facilities to the predetermined origin-destination paths. The different ways in which consumers understand distance is the major reason that location researchers have proposed isolated models for different types of consumers in isolation.

Essentially, a facility is to serve a spatially distributed set of demand nodes. In

other words, each potential facility site (candidate) provides a potential objective function value (POV) (e.g., demand-weighted distance) to each demand node. A demand node may represent all Type A consumers residing in a node or all Type B consumers along an OD path. Although objective functions have a variety of real-world interpretations with respect to consumers' types and needs, a POV often enters deterministic location models exogenously as a parameter. Accordingly, deterministic models themselves need not distinguish among consumer types. The Generalized and Efficient Strategy (GSUM) for unifying different consumer types and many objective functions is as follows. A dummy path is built for each demand node. In each dummy path, all candidates on the network that can serve a given demand node are listed in descending order of POV for a maximization problem or in ascending order of POV for a minimization problem. All candidates that cannot serve a given demand node are removed from that dummy path. The total number of candidates in each dummy path is specified. A POV string is also compiled to record each candidate's POV along the dummy path. Recall that a dummy path is not actually a path in physical space but rather a list of candidates in decreasing order of preference. A "list" perhaps would be a more descriptive term. However, I believe that the dummy path concept is an excellent way of understanding how Type A and Type C consumers are transformed into Type B consumers. GSUM is applied to many examples in section 4 below.

The most obvious advantage of GSUM is that it makes use of dummy paths and POV strings (rather than models themselves) to deal with various consumer types and objective functions. Another advantage of GSUM pertains to the use of a sorted strings data structure to save memory and decrease solution time. Many solution techniques (e.g., linear/integer programming, interchange heuristics, the algorithms in the ARC/INFO GIS system) use a sorted distance strings data structure to replace a standard distance matrix in order to speed processing time. Here I describe three simple approaches for reducing the volume of dummy paths and POV strings without affecting the properties of the optimal solutions.

First, in multiple-facility location problems, candidates near the end of a dummy path often do not affect the properties of the optimal solution. These candidates can thus be removed. In a maximization problem, the removed candidates have zero or very low POVs. In a minimization problem, the removed candidates have high POVs. An example is given by the classic  $p$ -median problem (ReVelle and Swain 1970), which locates  $p$  facilities in a manner that minimizes the total distance which is traveled by those who utilize the facilities. In this problem, the last  $p - 1$  candidates along each dummy path can always be removed because the farthest  $p - 1$  candidates from demand node  $i$  will never serve demand node  $i$ . In large problem instances, I can apply advanced strategies for cutting distance strings (Densham and Rushton 1992; Sorensen and Church 1996) to further remove candidates.

Second, some demand nodes (even those residing in different network nodes or along different OD paths) may have the same dummy path and POV string. These demand nodes can be combined and given a greater demand weight. This approach can reduce the total number of dummy paths. The following is an example. The MCLM problem aims to maximize the total number of consumers that are covered.

Here I consider the simultaneous existence of Type A and Type B consumers: Type A consumers can be covered by a facility within 4 units of distance; Type B consumers can be covered by the facilities along OD paths (See Figure 4-1 and Table 4-1). In Figure 4-1, Type A consumers at nodes 5 and 7 both have four visited candidates (4, 5, 6, and 7) and each candidate provides 1.0 POV. Although the two consumers reside in different nodes, they can be combined. If a Type B consumer has an OD path (e.g., 4, 5, 6, and 7), the three consumers can be combined. By removing candidates and combining paths, the problem size (the volume of data, decision variables, and constraints) is reduced. The two above approaches are analogous to assigning consumers who have the same travel plan into a large bus, and removing all unnecessary bus stops. Note that the associated POV strings are also reduced with dummy paths.

Third, POV strings can further drastically reduce data requirements in specific problems. For instance, in the classic MCLM and FILM models, candidates along a dummy path always provide the same value of POV (e.g., demand weight). In these situations, a POV string can be reduced to a single value.

**Table 4-1: Consumers, paths and distances**

Node	{A}	{B}	{C}	#	Flow Paths	$d_{qj}$	1	2	3	4	5	6	7	$V_{qj}$	1	2	3	4	5	6	7
1	1	1	1	4	1 3 5 7	1	0	4	3	8	8	9	10	1	0	1	0	1	0	3	0
2	1	1	1	3	2 3 6	2	4	0	2	4	6	8	7	2	5	0	0	1	1	0	3
3	1	1	1	3	3 5 7	3	3	2	0	6	5	6	7	3	6	2	0	2	0	3	0
4	1	1	1	3	4 5 6	4	8	4	6	0	2	5	3	4	12	7	7	0	0	0	2
5	1	1	1	2	5 6	5	8	6	5	2	0	3	2	5	14	11	8	4	0	0	3
6	1	1	1	2	6 7	6	9	8	6	5	3	0	4	6	15	11	9	5	1	0	0
7	1	1	1	3	7 4 2	7	10	7	7	3	2	4	0	7	7	0	2	0	2	5	0

*Each node has a different type of consumer; #: The total number of candidates along each real flow path;  $d_{qj}$ : Distance matrix;  $V_{qj}$ : Deviation distance matrix.*

### 4.3. A Generalized Location-Allocation Model

Using the GSUM principle, I formulate a generalized location-allocation model (GLAM) that locates  $p$  facilities in a manner that optimizes the total objective function values that arise from intercepting consumers traveling on flow paths. The formulation of GLAM is:

$$\text{Maximize (or minimize): } Z = \sum_{q \in Q} \sum_{j \in N_q} G_{qj} X_{qj} \quad (1)$$

s.t.

$$\sum_{j \in N_q} X_{qj} \leq 1, \forall q \in Q \quad (2)$$

$$X_{qj} \leq Y_j, \forall q \in Q, j \in N_q \quad (3)$$

$$\sum_{j \in J} Y_j = p \quad (4)$$

$$Y_j \in \{0, 1\}, \forall j \in J \quad (5)$$

$$0 \leq X_{qj} \leq 1, \forall q \in Q, j \in N_q \quad (6)$$

In this formulation, the parameters are:

$Q$  = the set of real and dummy paths indexed by  $q$

$J$  = the set of candidates indexed by  $j$

$p$  = the number of facilities to be located

$G_{qj}$  = the POV (e.g., demand weight distance) where flow along dummy path  $q$  is assigned to a facility at node  $j$

$G$  = denotes the matrix of data  $G_{qj}$

$N_q$  = the set of all nodes along dummy path  $q$

and the decision variables are:

$$Y_j = \begin{cases} 1 & \text{if there is a facility located at node } j \\ 0 & \text{otherwise} \end{cases}$$

$X_{qj}$  = the proportion of the POV along dummy path  $q$  which is assigned to the facility at node  $j$  ( $0 \leq X_{qj} \leq 1$ )

The objective function (1) aims to maximize or minimize total objective values. In a maximization problem, constraint (2) ensures that all POV along path  $q$  is intercepted at most once by the set of nodes in  $N_q$ . There is only a very small distinction between the maximization version of GLAM and its minimization version. In a minimization problem, constraint (2) must select “=” rather than “≤”, which ensures that all POV along path  $q$  must be intercepted by the set of nodes in  $N_q$ . Constraint (3) ensures that flow along the dummy path  $q$  can only be assigned to the node that is located at a facility. Constraint (4) ensures that exactly  $p$  facilities are located. Constraint (5) represents the standard integrality conditions.

Two previous generalized models and their variants are special cases of GLAM. Hillsman (1984) developed the unified linear model (ULM) which can encompass about 20 models, spanning 28 academic articles. ULM is structurally identical to the minimization version of GLAM. The primary drawback to ULM is that its solution times are often considerably greater than the times of more specific models because of the distance and objective function coefficient matrices. The size of each matrix is the number of demand nodes multiplied by the number of candidates. A secondary drawback is that the efficiency of ULM is drastically degraded when it has to give an arbitrarily large number to the distance matrix to prevent the assignment of a demand node to an unreasonable candidate (e.g., a candidate beyond a maximum distance from a demand point in the MCLM problem). However, the two drawbacks are eliminated by GLAM because it uses a structure of dummy paths and POV strings (see Section 4.1 Scenario {A} below). Zeng, Castillo and Hodgson (2007) proposed the generalized flow-interception location-allocation model (GFIM) which is exclusively for solving problems in scenario {B}. With few exceptions, GFIM effectively and efficiently solves all deterministic flow-interception location problems reported in the literature – about 20 different models spanning about 40 academic publications. GFIM is structurally identical to the maximization version of GLAM.

#### 4.4. The Efficiency of GSUM in Unifying Current and Future Location Models

This section aims to demonstrate that GSUM is able to significantly reduce the

solution burden on decision makers by unifying numerous current and future location models. The first part shows that GSUM enable GLAM to effectively and efficiently encompass many current and future models, including the maximal covering location model, flow-interception location model,  $p$ -median, and numerous variants of these models. The second part shows that GSUM can be directly applied to many classic known location models.

#### 4.4.1. GSUM Enables GLAM to Solve Many Maximization Problems

With GSUM unifying consumer types, GLAM can solve many maximization problems that involve the placement of  $p$  facilities on a network. The MCLM and FILM problem are two classic maximization problems reported in the literature. Schilling, Jayaraman, and Barkhi (1993) identified 65 articles on MCLM appearing in 25 different journals, and many more contributions have appeared since then. Zeng, Hodgson, and Castillo (2007) identified about 40 articles on FILM problems. Many MCLM and FILM problems reported in the literature can be classified as four types.

- (i) Maximum potential service user problems (MaxP1): consumers are either fully covered (by a facility within a specified distance from demand node or a facility along predetermined OD paths), or not covered at all.
- (ii) Maximum actual service user problems (MaxP2): the number of actual service users travelling to a facility is typically viewed as a decreasing step function of the distance to the facility.
- (iii) Capacity MaxP1: incorporate facility capacity issues into the MaxP1.
- (iv) Capacity MaxP2: incorporate facility capacity issues into the MaxP2.

MCLM and FILM studies have grown apart and researchers tend to propose isolated models for each different consumer scenario. Note that each type of problem has seven consumer scenarios based on considerations of three types of consumers. Considering the four types of problems, there are at least  $7 \times 4 = 28$  different problems. GSUM enables GLAM to effectively solve all of these problems. In selecting existing problems, I have not attempted to describe all possible problems and their variants that can be solved by GLAM; rather, I have attempted to cite a few early articles addressing problems in scenarios  $\{A\}$ ,  $\{B\}$  and  $\{A, B\}$  and to propose similar problems in scenarios  $\{C\}$ ,  $\{A, C\}$ ,  $\{B, C\}$ , and  $\{A, B, C\}$ .

To illustrate the effectiveness GLAM on different kinds of problems, I present a simple 7-node example (Figure 4-1 and Table 4-1) for each scenario. Large real-world problems are presented in section 5. All models in this chapter are coded in the AMPL language (Fourer, Gay, and Kernighan 2002) and solved with ILOG-CPLEX optimizer version 9.1.0 on a 2.80 GHz Pentium processor with 1024 MB of RAM in Microsoft Windows 2000. Figures 4-2, 4-3, and 4-4 are the code, data, and run scripts, respectively. As mentioned above, GLAM is able to solve the problems by simply modifying the data file (Figure 4-3).

##### 4.4.1.1. Maximum Potential Service User Problems

In scenario  $\{A\}$ , only Type A consumers exist, thus MaxP1 reduces to the classic

MCLM problem (Church and ReVelle 1974) which is to locate  $p$  facilities in a manner that maximizes the number of Type A consumers that are covered by at least one facility within a specified critical distance ( $\Delta$ ). GLAM solves this kind of problem when  $G$  is characterized as:

$$G_{qj} = \begin{cases} w_q, & \text{if } d_{qj} \leq \Delta \\ 0, & \text{otherwise} \end{cases}$$

where  $d_{qj}$  = the distance between consumer  $q$  and node  $j$ , and  $w_q$  = demand weight. That is, every candidate within a maximum distance from a consumer provides the POV for that consumer by  $w_q$ . In my example  $\Delta = 4$ , only nodes 1, 3, and 2 are within 4 units from node 1, thus  $G_{11} = G_{13} = G_{12} = w_1 = 1$  (Figure 4-1 and Table 4-2). The dummy path for the consumer residing in node 1 is  $1 \rightarrow 3 \rightarrow 2$ . (When candidates provide the same value, it does not matter which candidate is visited first.) Table 4-2 and Figure 4-3 provide the dummy paths and POV strings for all consumers. The optimal solutions for two facilities provided by GLAM are at nodes 2 and 6 which can cover all nodes in the network (Table 4-3).

As discussed earlier, when candidates along a dummy path provide the same value of POV, each POV string can be reduced to a single value (=  $w_q$  in this example). Pentium processor floating points require 4 bytes and integers require 2 bytes. POV and demand weight require floating points and the other data require integers. The data size of GLAM is  $3 \times 2 + (\sum_{i \in n} N_i + 1) \times 2 + n \times 4$ , where  $n$ ,  $m$ , and  $N_i$  are the total

number of demand nodes, candidates, and candidates within  $\Delta$  distance from node  $i$ , respectively. This 7-node example requires  $3 \times 2 + 32 \times 2 + 7 \times 4 = 98$  bytes (see the first 8 lines in Figure 4-3). The ULM model (Hillsman 1984) can also solve this problem, but has to use two  $n \times m$  matrixes to store distances and objective function coefficients. In this example, the data size for ULM is  $3 \times 2 + n \times m \times 2 + n \times m \times 4 = 3 \times 2 + 7 \times 7 \times 2 + 7 \times 7 \times 4 = 300$  bytes. Recall that the data size of ULM grows very quickly with the number of demand nodes and candidates.

In scenario {B}, only Type B consumers exist, thus MaxP1 reduces to the classic FILM model (Hodgson 1990; Berman, Larson, and Fouska 1992) which is to locate  $p$  facilities in a manner that maximizes the number of Type B consumers who encounter at least one facility along their OD paths. GLAM solves this problem when  $G$  is characterized as:

$$G_{qj} = \begin{cases} w_q, & \text{if } j \in N_q \\ 0, & \text{otherwise} \end{cases}$$

That is, every candidate along the predetermined path  $q$  provides POVs for that consumer by  $w_q$ . Accordingly, the dummy paths for Type B consumers are the same as their actual OD paths (Tables 4-1 and 4-2). Essentially, the only distinction between scenarios {A} and {B} is the concept of satisfactory distance: scenario {A} is based on whether a facility is within a critical distance from a consumer's fixed location, and scenario {B} is based on whether a facility is along the consumer's OD path. This distinction is reflected in dummy paths and POV strings, rather than in the GLAM

model. In my example, optimal solutions provided by GLAM for two facilities are at nodes 6 and 7, which can intercept all Type B consumers in the network (Table 4-3).

**Table 4-2:  $Q$  and  $G$  in section 4.4.1**

Scenario	Dummy Path $Q$					POV Strings						
	$q$	#	1	2	3	4	5	1	2	3	4	5
{A}	1	3	1	3	2			1	1	1		
	2	4	2	3	1	4		1	1	1	1	
	3	3	3	2	1			1	1	1		
	4	4	4	5	7	2		1	1	1	1	
	5	4	5	4	7	6		1	1	1	1	
	6	3	6	5	7			1	1	1		
	7	4	7	5	4	6		1	1	1	1	
{B}	1	4	1	3	5	7		1	1	1	1	
	2	3	2	3	6			1	1	1		
	3	3	3	5	7			1	1	1		
	4	3	4	5	6			1	1	1		
	5	2	5	6				1	1			
	6	2	6	7				1	1			
	7	3	7	4	2			1	1	1		
{C}	1	5	1	3	5	7	2	1	1	1	1	1
	2	5	2	3	6	1	4	1	1	1	1	1
	3	5	3	5	7	2	1	1	1	1	1	1
	4	5	4	5	6	7	2	1	1	1	1	1
	5	4	5	6	4	7		1	1	1	1	
	6	3	6	7	5			1	1	1		
	7	5	7	4	2	5	6	1	1	1	1	1

**Table 4-3: Optimal solutions in section 4.4.1.1**

Scenario	{A}	{B}	{C}	{A, B}	{A, C}	{B, C}	{A, B, C}
$p$	1 2	1 2	1 2	1 2 3	1 2	1 2	1 2 3
$Z$	4 7	4 7	6 7	8 13 14	10 14	10 14	14 20 21
Location	2 2,6	6 6,7	5 1,6	7 2,5 2,4,6	7 2,6	7 6,7	7 2,5 3,4,6

Perhaps Berman (1997) was the first researcher to notice the existence of Type C consumers. In scenario {C}, only Type C consumers exist. GLAM solves the problem when  $G$  is characterized as:

$$G_{qj} = \begin{cases} w_q, & \text{if } d_{qj} \leq \Delta \text{ or } j \in N_q \\ 0, & \text{otherwise} \end{cases}$$

That is, every candidate within a maximum “distance” of the demand point or of the OD path provides the POV for that consumer by  $w_q$ . In this example, for the consumer residing in node 1, nodes 3, 5, and 7 are along the OD path, and nodes 1, 2, and 3 are near node 1 ( $\Delta \leq 4$ ). Therefore, the nodes along the dummy path are the union of the two set of nodes (1, 3, 5, 7, and 2; Table 4-2). The optimal location for a single facility is at node 5 which covers 6 demand nodes (Table 4-3). In contrast, the optimal locations for a single facility in scenario {A} covers only 4 demand nodes.

Berman (1997) tried to consider the MaxP1 problem in scenario {A, B, C}. However, he artificially split Type C consumers into Type A and Type B consumers by adding some assumptions. Indeed, he was concerned with the problem in scenario {A, B}. His article formulated a new model that considers MaxP1 in scenario {A, B}. In comparison with MCLM and FILM, his model is more complex and computationally demanding. Rather than developing new models, GLAM can effectively solve MaxP1 for any scenario by simple pre-processing the input data: compile the dummy paths for various consumers and then enter all dummy paths into GLAM. Table 4-3 summarizes the results of the seven scenarios in my example.

#### 4.4.1.2. Maximum Actual Service User Problems

In scenario {A}, only Type A consumers exist, MaxP2 reduces to generalized maximal covering problems (e.g., Church and Roberts 1983; Berman and Krass 2002; Berman, Krass, and Drezner 2003; Karasakal and Karasakal 2004). In scenario {B}, only Type B consumers exist, MaxP2 reduces to generalized flow-interception problems (e.g., Berman, Bertsimas, and Larson 1995; Zeng, Castillo, and Hodgson 2007). Several researchers (Hodgson and Rosing 1992; Hodgson, Rosing, and Storrer 1997; Berman 1997; Erdemir *et al.* 2006) formulated several models for MaxP2 in scenario {A, B}. Their models are more complex and computationally demanding than GLAM. A wide variety of coverage functions in the real-world, together with seven consumer scenarios, have provided a rich collection of models but conceptually similar mathematical representations of the MaxP2 problems. GLAM effectively solves those problems simply by characterizing  $G$  in different ways. Note that  $G$  can be any function of the distance to facilities. A typical coverage function described by Karasakal and Karasakal 2004 is: consumers are fully covered within a minimum critical distance, decreasingly covered (partially covered) with distance beyond the minimum critical distance until the maximum critical distance, and not covered beyond this range. In this case,  $G$  is characterized as follows:

$$G_{qj} = \begin{cases} w_q, & \text{if } d_{qj} \leq R \\ w_q e^{-\alpha d_{qj}}, & \text{if } R < d_{qj} \leq T \\ 0, & \text{otherwise} \end{cases}$$

where:  $R$  = the maximum full coverage distance;

$T$  = the maximum partial coverage distance;

$\alpha$  = a scaling constant; and

$d_{qj}$  = the distance between node  $q$  and  $j$ , or the deviation distance between path



$q$  and node  $j$ .

Note that when  $R = T$  the problem reduces to the classic MCLM or FILM.

**Table 4-4:  $Q$  and  $G$  in section 4.4.1.2**

Scenario	Dummy Path $Q$							POV Strings								
	$q$	#	1	2	3	4	5	6	7	1	2	3	4	5	6	7
{A}	1	3	1	3	2					1	0.22	0.14				
	2	4	2	3	1	4				1	1	0.14	0.14			
	3	4	3	2	1	5				1	1	0.22	0.08			
	4	5	4	5	7	2	6			1	1	0.22	0.14	0.08		
	5	5	5	4	7	6	3			1	1	1	0.22	0.08		
	6	4	6	5	7	4				1	0.22	0.14	0.08			
	7	4	7	5	4	6				1	1	0.22	0.14			
{B}	1	7	1	3	5	7	2	4	6	1	1	1	1	1	1	0.22
	2	7	2	3	6	4	5	7	1	1	1	1	1	0.22	0.08	
	3	6	3	5	7	2	4	6		1	1	1	1	0.22		
	4	4	4	5	6	7				1	1	1				
	5	4	5	6	7	4				1	1	0.22	0.14			
	6	4	6	7	5	4				1	1	1	0.14			
	7	6	7	4	2	3	5	6		1	1	1	1	1	0.08	

**Table 4-5: Optimal solutions in section 4.4.1.2**

Scenario	{A}				{B}	{A, B}				
	$p$	1	2	3	4	1	1	2	3	4
$Z$		3.30	5.44	6.22	7.00	7.00	10.30	12.44	13.22	14.00
Location		5	3,5	1,2,5	1,2,5,6	5	5	3,5	1,2,5	1,2,5,6

In my example, Table 4-4 summarizes  $Q$  and  $G$  with  $\alpha = 0.5$ ,  $R = 2$ , and  $T = 5$  in scenarios {A} and {B}. In scenario {A}, consumers measure distance from a fixed location: for example, the distance between node 2 and the consumer residing in node 1 is 4 units. Thus,  $G_{12} = e^{-0.5 \times 4} = 0.14$ . In scenario {B}, the consumers wishing to visit a facility are assumed to first take the shortest path to the facility, and then from the facility to take the shortest path to the destination. The sum of these two shortest distances minus the shortest origin-destination distance is the deviation distance. For instance, the deviation distance between node 2 and the consumer coming from node 1 is  $4 + (4 + 3) - (3 + 5 + 2) = 1$  (Figure 4-1 and Table 4-1). Thus,  $G_{12} = 1$ . GLAM can effectively solve any scenario. Table 4-5 summarizes the solutions provided by GLAM in scenarios {A}, {B}, and {A, B}.

#### 4.4.1.3. Capacitated Maximization Problems

The capacitated maximization problem is a very important class of problem arising in

many contexts. Service facilities often have workloads which push them to the limit of their ability to provide effective service. The maximum capacity problems incorporate facility capacity issues into MaxP1 and MaxP2. In scenario {A}, only Type A consumers exist, this is the capacitated maximal covering problem (e.g., Chung 1986; Current and Storbeck 1988; Pirkul and Schilling 1991). In scenario {B}, only Type B consumers exist, this is the capacitated flow-interception problem (e.g., Zeng 2004; Zeng, Castillo and Hodgson 2007). GLAM effectively solves these problems in any scenario by replacing constraint (3) with the following constraint:

$$\sum_{q \in Q} G_q X_{qj} \leq C_j Y_j, j \in J \quad (7)$$

where  $C_j$  is the capacity of the facility at node  $j$ . Constraint (7) ensures that the total objective values assigned to a facility do not exceed the capacity of that facility.

#### 4.4.2. GSUM Enables GLAM to Solve Many Minimization Problems

GSUM also enables GLAM to solve many minimization problems. Perhaps the most commonly studied and used minimization problem is the  $p$ -median problem. Hundreds of articles on the  $p$ -median problem have been published since its debut in the 1960s. The  $p$ -median problem arises naturally in locating plants, warehouses, postal offices, schools, hospitals, and public buildings.

The classic  $p$ -median model (ReVelle and Swain 1970) is designed for solving problems in scenario {A}. GLAM effectively solves the  $p$ -median problem by defining the value of  $G_{qj} = w_q \times d_{qj}$ . Tables 4-6 and 4-7 provide dummy paths, POV strings, and results for the 7-node example.

Hodgson (1981) and Berman, Bertsimas, and Larson (1995) addressed the  $p$ -median problem in scenario {B}. They assumed that all consumers will travel from their predetermined trips to a service facility which is the closest in terms of the deviation distance. The goal of their models is to minimize the total deviation distance that is traveled by those who utilize the facilities. GLAM effectively solves these problems by defining the value of  $G_{qj} = w_q \times V_{qj}$ , where  $V_{qj}$  = the deviation distance between path  $q$  and node  $j$  (Table 4-6 and 4-7). Note that when the location of two facilities is involved,  $Z = 12$  in scenario {A}, but  $Z = 0$  in scenario {B}.

Berman (1997) addressed  $p$ -median problems in scenario {A, B}. The goal of his model is to minimize the total distances and deviation distances that are traveled by those who utilize the facilities. GLAM solves this problem by simply characterizing consumers using  $Q$  and  $G$ .

The use of  $Q$  and  $G$  to characterize the types of consumer rather than the model, simplifies the solutions to these problems. Table 4-7 summarizes the optimal solutions provided by GLAM.

A number of articles (e.g., Lorena and Senne 2003, 2004; Diaz and Fernandez 2006) have addressed the capacitated  $p$ -median problem where facilities have finite capacity. Zeng (2004) and Zeng, Castillo, and Hodgson (2007) considered capacitated flow-interception location-allocation problems. GLAM can effectively solve the two types of capacitated problems by replacing constraint (3) with the above constraint (7).

**Table 4-6:  $Q$  and  $G$  in section 4.4.2**

Scenario	$q$	Dummy Path $Q$							POV Strings							
		#	1	2	3	4	5	6	7	1	2	3	4	5	6	7
{A}	1	7	1	3	2	4	5	6	7	0	3	4	8	8	9	10
	2	7	2	3	1	4	5	7	6	0	2	4	4	6	7	8
	3	7	3	2	1	5	4	6	7	0	2	3	5	6	6	7
	4	7	4	5	7	2	6	3	1	0	2	3	4	5	6	8
	5	7	5	4	7	6	3	2	1	0	2	2	3	5	6	8
	6	7	6	5	7	4	3	2	1	0	3	4	5	6	8	9
	7	7	7	5	4	6	2	3	1	0	2	3	4	7	7	10
{B}	1	7	1	3	5	7	2	4	6	0	0	0	0	1	1	3
	2	7	2	3	6	4	5	7	1	0	0	0	1	1	3	5
	3	7	3	5	7	2	4	6	1	0	0	0	2	2	3	6
	4	7	4	5	6	7	2	3	1	0	0	0	2	7	7	12
	5	7	5	6	7	4	3	2	1	0	0	3	4	8	11	14
	6	7	6	7	5	4	3	2	1	0	0	1	5	9	11	15
	7	7	7	4	2	3	5	6	1	0	0	0	2	2	5	7

**Table 4-7: Optimal solutions in section 4.4.2**

Scenario	{A}							{B}	{A, B}							
	$p$	1	2	3	4	5	6	7	1	2	1	2	3	4	5	6
$Z$	26	12	9	6	4	2	0	4	0	30	14	10	6	4	2	0
Location	5	3,	3,5,	1,3,	1,2,5,	1,3,4,	1...	5	6,	5	2,	3,6,	1,2,	1,2,5,	1,3,4,	1...
		5	6	5,6	6,7	5,6,7	7	7	7		5	7	5,6	6,7	5,6,7	7

*\*All optimal solutions are identified by Figure 4-2 replacing "maximize" with "minimize", "<=" with "=" in constraint\_2*

#### 4.4.3. Application of GSUM to Other Location Models

We have seen that GSUM unifies consumer types and many objective functions by creating dummy paths and POV strings. GSUM enables GLAM to effectively solve a broad variety of location problems. GSUM can be applied to many current location models. An example is the set covering location model (SCLM) (Toregas *et al.* 1971), which seeks the minimum number of facilities so that all demands are covered by at least one facility within a specified distance. SCLM has received considerable attention in the location literature. Schilling, Jayaraman, and Barkhi (1993) identified 40 articles on SCLM appearing in 25 different journals, and many more contributions have appeared since then. Holmes, Williams, and Brown (1972) and Hodgson and Doyle (1978) extended SCLM to locate day-care centers. Although the latter article indicated that a high proportion of automobile users and public transit users rely on day-care centers along the daily commute between home and workplace, the literature

on location of day-care centers assumes that all service users are Type A consumers. With GSUM, this assumption can be easily relaxed.

Here I consider the 7-node example, where Type A consumers can be covered by facilities within 4 units of distance; Type B consumers can be covered by facilities within 4 units of deviation distance; and Type C consumers can be covered by facilities within 4 units of distance or deviation distance. (Note that the distance threshold need not be the same for the different consumers.) Table 4-8 summarizes the dummy paths for the three types of consumer. For instance, in scenario {A}, for the consumer at node 1, nodes {1, 3, 2} are within 4 units of distance, thus the dummy path is {1, 3, 2}. In scenario {B}, for the consumer coming from node 1, all 7 nodes are within 4 units of deviation distance, thus the dummy path is 1, 3, 5, 7, 2, 4, 6. For the consumer at node 4 in scenario {C}, nodes {4, 5, 7, 2} are within 4 units of distance, and nodes {4, 5, 6, 7} are within 4 units of deviation distance, thus the dummy path is {2, 4, 5, 6, 7}. In this simple problem, all POV strings can be removed because they are all equal to one. Table 4-9 summarizes the results identified by SCLM in the seven scenarios.

#### **4.5. The Importance of Type C Consumers in Location Modeling**

It is assumed to locate service facilities such as public libraries, postal offices, day-care centers, video rental stores, or automatic teller machines in the city of Edmonton, Alberta, Canada. The goal is to maximize the total number of potential service users. Some people (e.g., senior citizens or children) use facilities only from home, Type A consumers; some people patronize facilities only from their daily commute, Type B consumers; and some people patronize facilities either from home or along their predetermined trips, Type C consumers. In this section, I compare the importance of the three types of consumer in location modeling.

My real-world example is the 2001 afternoon peak traffic network for the city of Edmonton in Canada, comprising 16,488 OD flow paths, 290 zones, 1,746 nodes and 4,606 links (Figure 5) described by Zeng, Hodgson, Castillo (2007). The original data were provided by the City of Edmonton Transportation and Streets Department, who state that their data have been produced according to industry standards and that their forecasting model is highly recognized throughout North America. Flows in these two data sets are vehicle flows among traffic zones in the full Edmonton area. These afternoon flows are dominated by movement from the central to the peripheral areas, but each of the 290 zone centroids serves both as an origin and a destination, producing what I view as a realistic test-bed for the flow-interception location problem. Because the afternoon peak is the rush-hour journey from work, each trip destination can be viewed as a home.

The consumers in each scenario are described as follows. In scenario {A}, the 51375 inflows to 290 destinations are viewed as Type A consumers, who are either fully covered by the facility within one kilometre Euclidean distance from their homes or not covered at all. In scenario {B}, the 51375 flows along 16488 OD paths are viewed as Type B consumers, who are either fully intercepted by the facility along their OD paths or not intercepted at all. Figure 4-5 illustrates the Type A and Type B

consumer patterns. Circles with areas are proportional to the percent of Type A consumers. Link widths are proportional to the percent of Type B consumers. In scenario {C}, I add an extension path to each of the 16,488 OD paths. For each OD path, the extension path is to visit the nodes which are within one kilometre of the destination excluding the nodes along that OD path. Therefore the 51,375 flows along the 16,488 OD and extension paths can be viewed as Type C consumers, who are either fully covered by the facility along their OD and extension paths, or not covered at all. Intuitively, the high flow nodes near large circles are highly preferred locations for Type C consumers (Figure 4-5). I assume that each type of consumer in scenarios {A, B}, {A, C}, and {B, C} are half of that in each scenario {A}, {B}, or {C}. For instance, the number of Type A consumers at each destination in scenario {A, B} is half of that in scenario {A}, and the number of Type B consumers along each OD path in scenario {A, B} is half of that in scenario {B}. Each type of consumer in scenario {A, B, C} is one third of that in each scenario {A}, {B}, or {C}. Note that each scenario has the same total number of consumers. My findings are exemplified by the results identified by GLAM at  $p = 4$ . I select a small number of facilities because it is easier for readers to visualize several solution patterns in a figure.

**Table 4-8:  $Q$  in section 4.4.3**

Scenario	q	#	1	2	3	4	5	6	7
{A}	1	3	1	3	2				
	2	4	2	3	1	4			
	3	3	3	2	1				
	4	4	4	5	7	2			
	5	4	5	4	7	6			
	6	3	6	5	7				
	7	4	7	5	4	6			
{B}	1	7	1	3	5	7	2	4	6
	2	6	2	3	6	4	5	7	
	3	6	3	5	7	2	4	6	
	4	4	4	5	6	7			
	5	4	5	6	4	7			
	6	3	6	7	5				
	7	5	7	4	2	3	5		
{C}	1	7	1	2	3	4	5	6	7
	2	7	1	2	3	4	5	6	7
	3	7	1	2	3	4	5	6	7
	4	5	2	4	5	6	7		
	5	4	4	5	6	7			
	6	3	5	6	7				
	7	6	2	3	4	5	6	7	

**Table 4-9: Optimal solutions in section 4.4.3**

Scenario	{A}	{B}	{C}	{A,B}	{A,C}	{B,C}	{A,B,C}
Z	2	1	1	2	2	1	2
location	2,6	5	5	2,6	2,6	5	2,6

First, I analyze solution patterns in scenarios {A}, {B}, and {C}. Figure 4-6 illustrates the optimal locations in the seven scenarios. The small, black points illustrate the covered destinations in scenario {A}. In scenario {A}, there is no facility at a high flow node but each facility is central to several major destinations, a characteristic of the classic MCLM solution. In scenario {B}, four facilities are located in the north, west, south, and southeast of the city; clearly optimal FILM solutions avoid flow cannibalization. In scenario {C}, all facilities are located at high flow nodes near several major destinations, clearly incorporating benefits of the two solutions in scenarios {A} and {B}. Table 4-10 illustrates the obvious result that each optimal solution covers the largest number of consumers in that particular scenario. SA, SB, SC, SAB, SAC, SBC, and SABC are the seven optimal solutions for scenarios {A}, {B}, {C}, {A, B}, {A, C}, {B, C} and {A, B, C}, respectively.

**Table 4-10: Consumers covered by each solution**

Scenario	{A}	{B}	{C}	{A,B}	{A,C}	{B,C}	{A,B,C}
SA	<b>7017</b>	3026	8404	5022	7710	5715	7550
SB	3203	<b>9307</b>	10772	6255	6988	10040	9556
SC	5189	8226	<b>11766</b>	6708	8478	9996	10354
SAB	5216	8704	11456	<b>6960</b>	8336	10080	10368
SAC	5951	7630	11354	6790	<b>8652</b>	9492	10204
SBC or SABC	4986	8668	11652	6867	8319	<b>10160</b>	<b>10377</b>

\* SBC and SABC are the same by coincidence.

Second, I compare the influence of the three types of consumer on real-world location modeling, by considering two simple indices:

$$I_{CA} = \frac{Z_{SC} - Z_{SA}}{Z_{SA}} \times 100\%, \text{ and } I_{CB} = \frac{Z_{SC} - Z_{SB}}{Z_{SB}} \times 100\% \quad (8)$$

$I_{CA}$  equals the proportion of additional consumers covered by SC compared to SA;  $I_{CB}$  is the proportion of additional consumers covered by SC compared to SB. I term the index Superiority of SC. These indices (Table 4-11) indicate the influence of solutions {A}, {B} and {C} in each of the seven scenarios. SC always covers more consumers than SA except for scenario {A}. In scenarios {B}, SC covers 171.8% more consumers than SA. Considering the seven scenarios, SC covers an average of 48.8% more consumers than SA. SC always covers more consumers than SB except for scenarios {B} and {B, C}, where SC covers 11.6% and 0.4% fewer consumers than SB. Considering the seven scenarios, SC covers on average 13.7% more consumers than SB. When there is more than one type of consumers, SC nearly always covers

more consumers than SA or SB: SB covers a few more consumers than SC in scenario {B, C}. Note that SC covers more consumers than SA or SB even in scenario {A, B}. Figure 4-6 also illustrates that SC outperforms SA or SB: for instance, SA chooses node 4 which is not a good location for Type B and Type C consumers; SB chooses node 5 which is not a good location for Type A and Type C consumers; SC chooses node 11 which is a good location for Type A, Type B, and Type C consumers. Moreover, the fact that three of four facilities in each scenario {A, C}, {B, C} or {A, B, C} are the same as those in scenario {C} (Figure 4-6), also indicates that Type C consumers exert the most influence on location in this particular example. In short, these findings show that Type C consumers have much greater impact than Type A and Type B consumers and that ignoring Type C consumers greatly impairs the benefits of location modeling.

**Table 4-11: Superiority of SC**

Scenario	{A}	{B}	{C}	{A,B}	{A,C}	{B,C}	{A,B,C}	Average
$I_{CA}$	-26.1%	<b>171.8%</b>	<b>40.0%</b>	<b>33.6%</b>	10.0%	<b>74.9%</b>	<b>37.1%</b>	<b>48.8%</b>
$I_{CB}$	<b>62.0%</b>	-11.6%	9.2%	7.2%	<b>21.3%</b>	-0.4%	8.4%	<b>13.7%</b>

**Table 4-12: The Solution Robustness**

$I_{optimal}$	{A}	{B}	{C}	{A,B}	{A,C}	{B,C}	{A,B,C}	Mean	SD
SA	0.0	<b>-67.5</b>	<b>-28.6</b>	<b>-27.8</b>	-10.9	<b>-43.8</b>	<b>-27.2</b>	<b>-29.4</b>	<b>21.9</b>
SB	<b>-54.4</b>	0.0	-8.4	-10.1	<b>-19.2</b>	-1.2	-7.9	-14.5	18.7
SC	-26.1	-11.6	0.0	-3.6	-2.0	-1.6	-0.2	-6.4	9.5
SAB	-25.7	-6.5	-2.6	0.0	-3.7	-0.8	-0.1	-5.6	9.1
SAC	-15.2	-18.0	-3.5	-2.4	0.0	-6.6	-1.7	-6.8	7.1
SBC or SABC	-28.9	-6.9	-1.0	-1.3	-3.8	0.0	0.0	-6.0	10.4

*SD: Standard Deviation*

Finally, once a facility is built, it is expensive to relocate it in the future. However, many inputs (and consequently the outputs as well) to real-world location problems depend on time and are likely to be uncertain. In other words, while most facilities are static, inputs (especially demands) to real-world location problems are dynamic and probabilistic. A robust solution should remain a good solution when the scenario has changed. Therefore, it is highly important to assess the robustness of each solution against the optimal solution in each scenario. I do this evaluation by using a simple index:

$$I_{optimal} = \frac{Z - Z_{optimal}}{Z_{optimal}} \times 100\% \quad (9)$$

The numerator indicates the consumers not covered by a solution compared to the optimal solution in that scenario. The denominator indicates the number of consumers covered by the optimal solution in that scenario. I term the index Solution Robustness. Table 4-12 indicates the robustness of the seven solutions in each scenario. SC (an

average of  $-6.4\%$   $I_{optimal}$ ) is less negative than SA (an average of  $-29.4\%$   $I_{optimal}$ ) or SB (an average of  $-14.5\%$   $I_{optimal}$ ). The index standard deviation of SA (21.9) or SB (18.7) is much larger than that of SC (9.5). The solutions identified in the traditional scenarios {A} and {B} are not robust, but the solutions in other five scenarios are robust. Therefore, it is important to consider the simultaneous existence of different types of consumers, particularly Type C consumers, in location modeling.

#### 4.6. Conclusions

In network location theory, it is traditionally assumed that consumers patronize facilities as close as possible to demand points (Type A consumer). In flow-interception location theory, it is traditionally assumed that consumers patronize facilities near or on predetermined trips (Type B consumer). In the real world, however, if a facility is close to demand points (e.g., home), consumers may patronize it; if a facility is close to predetermined trips (e.g., daily commute between home and workplace) consumers may also patronize it. The novelty of this article is to consider this new type of consumers, termed Type C consumers. Most people in the real-world are Type C consumers – they are not as selective of location as Type A and Type B consumers. The literature has either neglected Type C consumers or artificially split them into Type A and Type B consumers by adding some assumptions. This article considers the influence of incorporating the Type C consumers in location models with 2001 afternoon peak traffic data for the city of Edmonton in Canada. My findings show that Type C consumers are critical in facility location problems.

Location researchers have traditionally proposed models for different types of consumers in isolation and tend to introduce changes in objective functions and/or assumptions by developing new models. This article introduces a generalized and efficient strategy for unifying consumer types and location models, called GSUM. GSUM can transform Type A and Type C consumers into Type B consumers. Thus, many location models and theories based on a particular consumer type can be unified. An example is that the two divergent flow-interception and conventional network location theories can be unified. Moreover, using GSUM principle, numerous location models can be unified. For instance, a generalized location-allocation model is formulated to effectively and efficiently encompass at least 60 existing models, including the  $p$ -median, maximal covering location model, flow-interception location model, and numerous variants of these models.

#### References

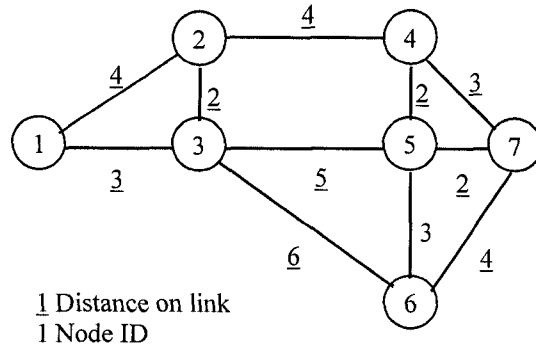
- Averbakh, I., O. Berman. 1996. Locating flow-capturing units on a network with multi-counting and diminishing returns to scale. *European Journal of Operational Research* **91** 495-506.
- Berman, O. 1997. Deterministic flow-demand location problems. *Journal of Operational Research Society* **48** 75-81.
- Berman, O., D. Bertsimas, R. C. Larson. 1995. Locating discretionary service facilities, II: maximizing market size, minimizing inconvenience. *Operations Research* **43** 623-632.



- Berman, O., D. Krass, Z. Drezner. 2003. The gradual covering decay location problem on a network. *European Journal of Operation Research* **151** 474-480.
- Berman, O., D. Krass. 2002. The generalized maximal covering location problem. *Computer & Operational Research* **29** 563-581.
- Berman, O., M. J. Hodgson, D. Krass. 1995. Flow-interception problems. In *Facility Location: A Survey of Applications and Methods*, edited by Z. Drezner. Springer-Verlag, New York, 389-426.
- Berman, O., R. C. Larson, N. Fouska. 1992. Optimal location of discretionary service facilities. *Transportation Science* **26** 201-211.
- Chung, C-H, D. 1986. Recent applications of the maximal covering location planning (M.C.L.P.) model. *Journal of the Operational Research Society* **37** 735-746.
- Church, R., C. ReVelle. 1974. The maximal covering location problem. *Papers of the Regional Science Association* **32** 101-118.
- Church, R.L., K. L. Roberts. 1983. Generalized coverage models and public facility location. *Papers of the regional science association* **53** 117-135.
- Current, J. R., J. E. Storbeck. 1988. Capacitated covering models. *Environment and Planning B: Planning and Design* **15** 153-163.
- Densham, P. J., G. Rushton. 1992. Strategies for solving large location-allocation problems by heuristic method. *Environment and Planning A* **24** 289-304.
- Diaz, J.A., E. Fernandez. 2006. Hybrid scatter search and path relinking for the capacitated  $p$ -median problem. *European Journal of Operational Research* **169** 570-585.
- Erdemir, E. T., R. Batta, S. Spielman, P.A. Rogerson, A. Blatt, M. Flanigan. 2006. Location coverage models with demand originating from nodes and paths: application to cellular network design. Manuscript for *European Journal of Operational Research*.
- Fourer, D., D. M., Gay, B.W., Kernighan. 2002 *AMPL: A Modeling Language for Mathematical Programming*. Second Edition. Thomson.
- Gendreau, M., G. Laporte, I. Parent. 2000. Heuristics for the location of inspection stations on a network. *Naval Research Logistics* **47** 287-303.
- Hillsman, E. 1980. Heuristic solutions to location-allocation problems: A user's Guide to Alloc IV, V, and VI, *Monograph No.7*, Department of Geography, The University of Iowa, Iowa City.
- Hillsman, E.L. 1984. The  $p$ -median structure as a unified linear model for location-allocation analysis. *Environment and Planning A* **16** 305-318.
- Hodgson, M. J. 1981. The location of public facilities intermediate to the journey to work. *European Journal of Operational Research* **6** 199-204.
- Hodgson, M. J. 1990. A flow-capturing location-allocation model. *Geographical Analysis* **22** 270-279.
- Hodgson, M. J., K. E. Rosing, A. L. G. Storrier. 1996. Applying the flow-capturing location-allocation model to an authentic network: Edmonton, Canada. *European Journal of Operational Research* **90** 427-443.
- Hodgson, M. J., K. E. Rosing, A. L. G. Storrier. 1997. Testing a bicriterion location-allocation model with real-world network traffic: the case of Edmonton, Canada. In:

- Multicriteria Analysis*, edited by J. Climaco. Springer, 484-495.
- Hodgson, M. J., K. E. Rosling, J. Zhang. 1996. Locating vehicle inspection stations to protect a transportation network. *Geographical Analysis* **28** 299-314.
- Hodgson, M. J., K. E. Rosling. 1992. A network location-allocation model trading off flow capturing and  $p$ -median objectives. *Annals of Operations Research* **40** 247-260.
- Hodgson, M. J., O. Berman. 1997. A billboard location model. *Geographical & Environmental Modeling* **1** 25-45.
- Hodgson, M. J., P. Doyle. 1978. The location of public services considering the model of travel. *Socio-Economic Planning Sciences* **12** 49-54.
- Holmes, J. Williams F. B., Brown L. A. 1972. Facility location under a maximum travel restriction: an example using day care facilities. *Geographical Analysis* **4** 258-266.
- Horner, M. W., S. Groves. 2007. Network flows-based strategies for identifying rail park-and-ride facility locations. *Socio-Economic Planning Sciences* **41** 255-268.
- Karasakal, O., E.K. Karasakal. 2004. A maximal covering location model in the presence of partial coverage. *Computers & Operational Research* **31** 1515-1526.
- Kuby, M. 2006. Prospects for geographical research on alternative-fuel vehicles. *Journal of Transport Geography* **14**, 234-236.
- Kuby, M., S. Lim. 2005. The flow-refuelling location problem for alternative-fuel vehicles. *Socio-Economic Planning Sciences* **39** 125-145.
- Kuby, M., S. Lim. 2007. Location of alternative-fuel stations using the flow-refueling location model and dispersion of candidate sites on arcs. *Networks and Spatial Economics* **7** 129-152.
- Lorena, L. A.N., E. L.F. Senne. 2004. A column generation approach to capacitated  $p$ -median problems. *Computers & Operations Research* **31** 863-876.
- Lorena, L.A.N., E.L.F. Senne. 2003. Local search heuristics for capacitated  $p$ -median problems, *Networks and Spatial Economics* **3** 409-419.
- Miller, H. J., S. Shaw. 2001. *Geographic Information Systems for Transportation*. New York, NY: Oxford University Press.
- Pirkul, H., D. A. Schilling. 1991. The maximal covering location problem with capacities on total workload. *Management Science* **37** 233-248.
- ReVelle, C.S., R. Swain. 1970. Central facilities location. *Geographical Analysis* **2** 30-42.
- Schilling, D. A., V. Jayaraman, R. Barkhi. 1993. A review of covering problems in facility location. *Location Science* **1** 25-55.
- Sorenson, P. A., R. L. Church. 1996. A comparison of strategies for data storage reduction in location-allocation problems. *Geographical Systems* **3** 221-242.
- Toregas, C., R. Swain, C. ReVelle, L. Bergman. 1971. The location of emergency service facilities. *Operations Research* **19** 1363-1373.
- Turner, D. 2006. Implementing the flow-interception location model with geographic information systems. Master Thesis, University of Texas at Dallas, USA.
- Upchurch, C., M., Kuby, S. Lim. 2006. A model for location of capacitated alternative-fuel stations. Forthcoming in *Geographical Analysis*.

- Wang, Q., R. Batta, C. M. Rump. 2002. Algorithms for a facility location problem with stochastic customer demand and immobile servers. *Annals of Operations Research* **111** 17-34.
- Zeng, W. 2004. Flow-interception problems addressing where flows are intercepted, the Institute for Operations Research and the Management Sciences / Canadian Operational Research Society (INFORMS/CORS) Joint Meeting at Banff, May 19 2004.
- Zeng, W., I. Castillo, M. J. Hodgson. 2007. A generalized model for locating facilities on a network with flow-based demand. Manuscript for *Transportation Science*.
- Zeng, W., M. J. Hodgson, I. Castillo. 2007. The pickup problem: consumers' locational preferences in flow interception. Forthcoming in *Geographical Analysis*.

**Figure 4-1: A test 7-node network****Figure 4-2: GLAM.mod**


---

```

param Q>=0;
param J>=0;
param p>=0;
param N{1..Q};
param path{q in 1..Q,j in 1..N[q]};
param G{q in 1..Q,j in 1..N[q]};
var Y{1..J} binary;
var X{q in 1..Q,j in 1..N[q]}>=0,<=1;
maximize Z: sum {q in 1..Q} sum {j in 1..N[q]} G[q,j] *X[q,j];
subject to constraint_2 {q in 1..Q}: sum {j in 1..N[q]} X[q,j] <= 1;
subject to constraint_3 {q in 1..Q,j in 1..N[q]}: Y[path[q,j]] >= X[q,j];
subject to constraint_4: sum{j in 1..J} Y[j] = p;
  
```

---

**Figure 4-3: GLAM.txt (example in section 4.1.1)**

```

-----
7 7 2
3 1 3 2
4 2 3 1 4
3 3 2 1
4 4 5 7 2
4 5 4 7 6
3 6 5 7
4 7 5 4 6
1 1 1
1 1 1 1
1 1 1
1 1 1 1
1 1 1 1
1 1 1
1 1 1 1
-----

```

- \* The first line is the total number of dummy paths, nodes and facilities, respectively.
- \* The last 7 lines are POV strings. The others are dummy paths.

**Figure 4-4: GLAM.run**

```

-----
# include GLAM.run;
option solver cplex;
model GLAM.mod;
read Q, J, p < GLAM.txt; # read Q, J, p
read {q in 1..Q} (N[q], {t in 1..N[q]} path[q,t]) < GLAM.txt; # read N, path
read {q in 1..Q} ({j in 1..N[q]} G[q,j]) < GLAM.txt; # read G[q,j]
option cplex_options 'dual';
option print_separator ","; # separator with "," in writing file
printf "p,?Optimal,Second,Z,Solution\n">GLAMresults.txt;
      # write title line
set SetP := 1 ..4; # set the range of p facilities
for {c in SetP} {
  let p := c;
  solve;
  # The follow commands are to print results in two files
  printf
"%u,%u,%0.1f,%0.2f",p,solve_result_num,_solve_user_time,Z>GLAMresults.txt;
  print {j in 1..J:Y[j]>=1} j>GLAMresults.txt; # location
  printf "p= %u solve_result_num= %u \n",p,solve_result_num>GLAMbranchMIP.txt;
  display solve_message>GLAMbranchMIP.txt; }
-----

```

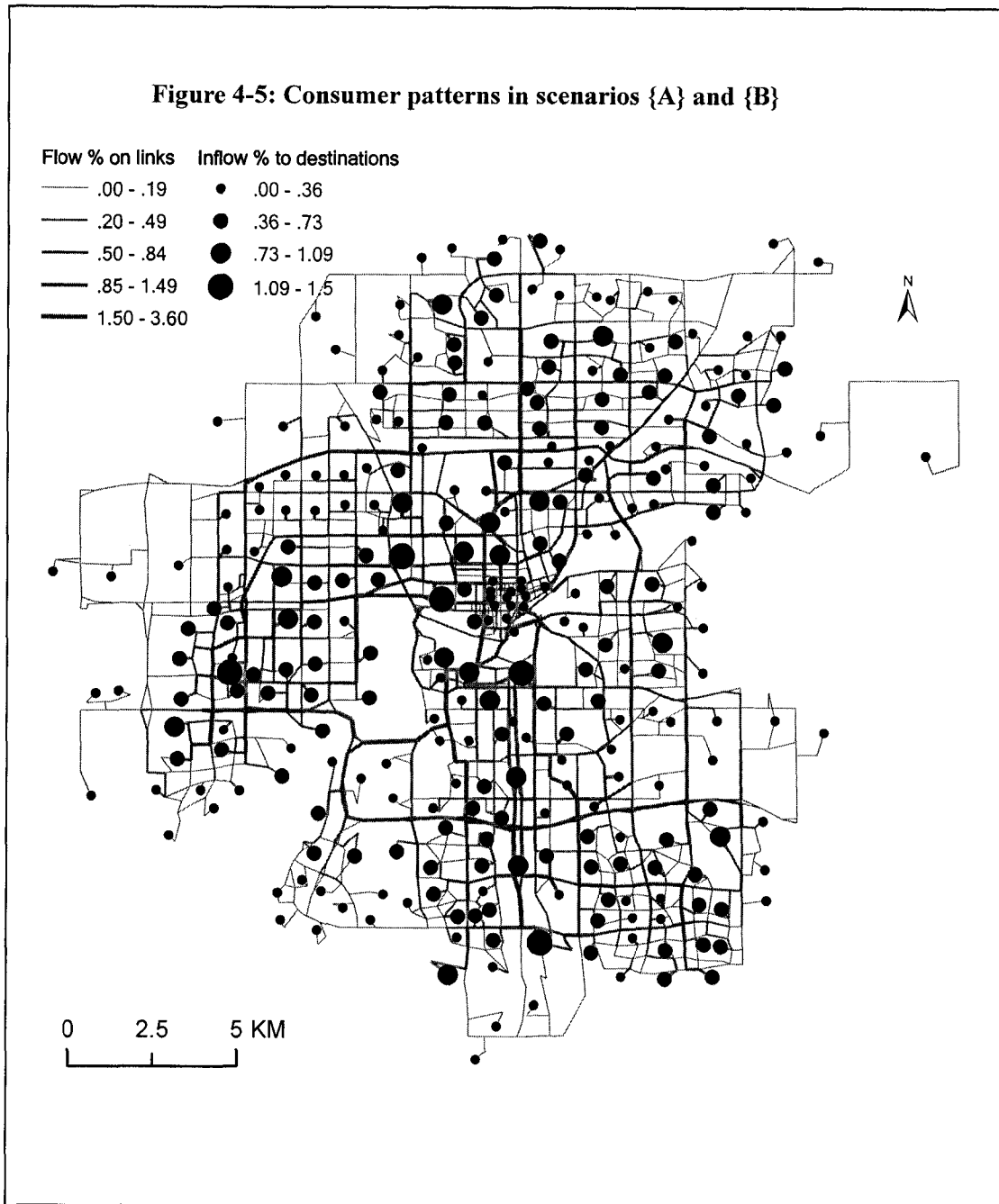
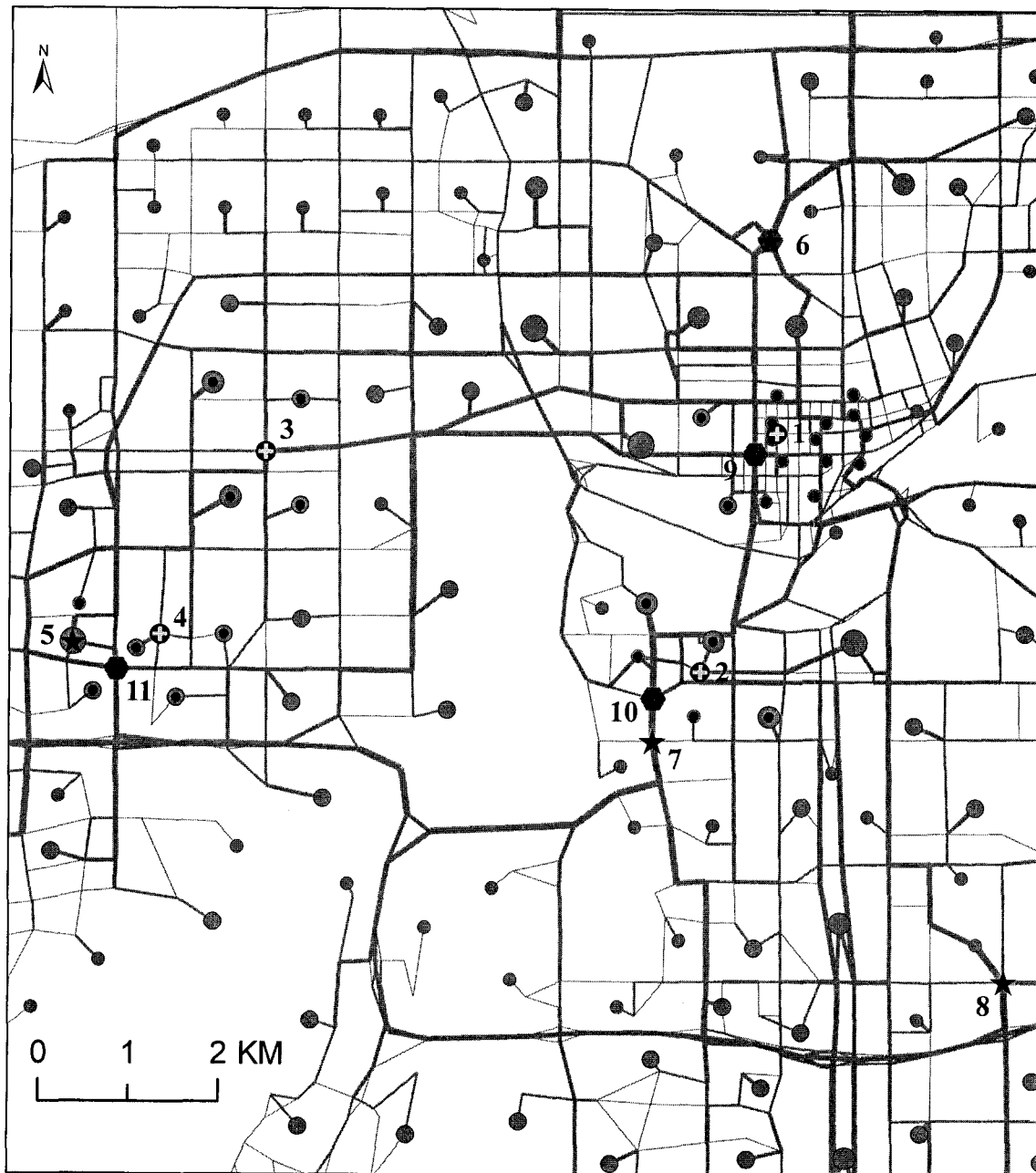
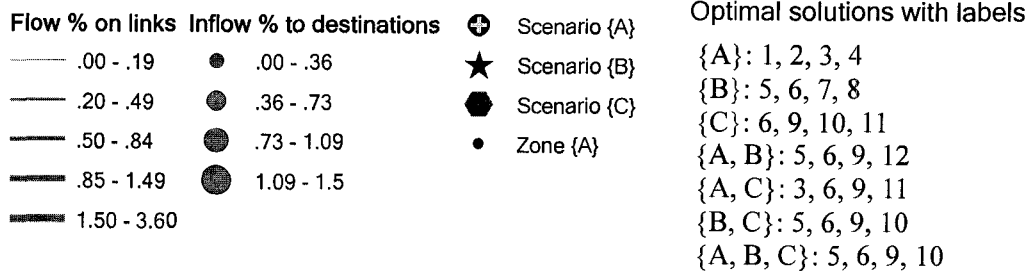


Figure 4-6: Optimal solution at each scenario (p = 4)



## Chapter 5 \*

### **An Integrated GIS, Optimization, and Heuristic System for Flow-Based Demand Aggregation**

**Summary:** Traditional network location problems seek optimal or good facility locations on a network with point-based demand. Flow-interception location problems seek optimal or good facility locations on a network with flow-based demand. Flow-interception problems have been of considerable recent interest, represented by about 40 academic publications over the past 17 years. In most real-world location studies, spatially aggregated data is used due to its original dimensionality. Point-based demand aggregation has received considerable research interest in both industry and academia. Systematic studies of flow-based demand aggregation have not, however, been reported to date. In this chapter, I integrate geographic information systems (GIS), optimization, and heuristic methodologies to examine the special network flow structure of real-world transportation systems and to develop an integrated system for aggregating flow-based demand data. I apply this integrated system to the standard flow-interception model with 2001 afternoon peak traffic data for Edmonton, Alberta (the sixth largest Canadian city). I find this application to be extremely efficient and, most notably, totally free of aggregation error.

**\* A version of this chapter has been submitted for publication.**

**Zeng, Weiping,** Ignacio Castillo. 2007. An integrated GIS, optimization, and heuristic system for flow-based demand aggregation. Submitted to *Computers & Operations Research* on May 19, 2007, submission No:COR-D-07-00267, under review.



### 5.1. Introduction

A business profits from a good location: whether it is a small coffee shop with a local clientele or a multinational network of widget factories with distribution centers and a worldwide chain of retail outlets. Location theory provides decision makers with quantitative tools for seeking locations where fixed and operating costs can be kept low and accessibility to markets high. Traditional network location theory (e.g., the  $p$ -median model formulated by ReVelle and Swain in 1970) assumes that demand for service occurs at fixed points and that consumers patronize facilities as close as possible to them (this type of demand is also referred to as point-based demand). Flow-interception location theory assumes that demand for service is expressed by flows travelling on predetermined origin-destination (OD) paths and that consumers patronize facilities on or near these predetermined OD paths (this type of demand is also referred to as flow-based demand). Since flow-interception theory was identified by Hodgson (1990) and Berman, Larson, and Fouska (1992) in the early 1990s, flow-based demand has received considerable research interest, represented by about 40 academic publications.

Most real-world location studies use spatially aggregated data. Data is aggregated to reduce computational effort, for ease of collection and analysis, to facilitate understanding, or for confidentiality. In many cases, the only data available is already aggregated. Point-based demand aggregation has received a considerable interest in both industry and academia. Systematic studies of flow-based demand aggregation have not, however, been reported in the literature.

Traffic flow data is essential for planning and management of transportation systems. Traffic flow data provided by government agencies is usually already highly aggregated, presented for entities such as traffic zones represented as centroids. Flow-based demand is typically presented for origin-destination (OD) flow paths. The simplest and most widely used approximation for estimating flows is to assign all OD flow to the shortest path for each OD pair (Doblas and Benitez 2005). If the number of traffic zones is  $n$ , the total number of OD flow paths is  $n \times (n-1)$ , thus increasing rapidly with the number of zones. For example, the 395 traffic zones in the city of Edmonton, Alberta (the sixth largest Canadian city) produce 155,630 OD pairs; the 1790 traffic zones in the Chicago region produces 3,202,310 OD pairs. The average number of nodes on each OD path also grows with the number of network nodes. Even with the most efficient and specialized heuristics, good solutions to large real-world flow-interception location problems are beyond the capability of the personal computer. Therefore, large real-world flow-interception modeling and analysis must rely on aggregated data.

In traditional facility location, aggregation is performed by identifying sets of demand points that are close to one another spatially, and representing these sets by single points. Aggregation reduces instances to computationally tractable size, but also introduces errors into the value of the objective function and into the selection of optimal locations. In this chapter, I show that, in contrast to traditional facility location, the special network flow structure of real-world transportation systems allows flow-based demand data to be aggregated with few or no errors. This chapter reports on an

effort towards integrating geographic information systems (GIS), optimization modeling, and heuristics to examine the special network flow structure of real-world transportation systems and to develop efficient methods of aggregating flow-based demand data.

The remainder of this chapter is organized as follows. Section 5.2 introduces some background of the standard flow-interception location model, aggregation errors, GIS, optimization, and heuristics. Section 5.3 integrates GIS, optimization, and heuristics to examine the special network flow structure of 2001 afternoon peak traffic data for Edmonton, Alberta, and to aggregate data for the standard flow-interception location model. Section 5.4 proposes an integrated system for flow-based demand aggregation. The final section offers major conclusions.

## 5.2. Background

### 5.2.1. The Standard Flow-Interception Location Model

This chapter aims to show the importance of flow-based demand aggregation and to develop a framework for aggregating such a demand: representing the first systematic study on flow-based demand aggregation.

The standard flow-interception location model (FILM) (Hodgson 1990; Berman, Larson, and Fouska 1992) is the perfect model for my goals – its aggregation errors are easy to understand, and its outputs are easy to measure and compare. This classic model is aimed at maximizing the number of consumers who encounter at least one facility along their predetermined journeys. The mathematical formulation of FILM (Hodgson 1990; Berman, Larson, and Fouska 1992) is:

$$\text{Maximize: } Z = \sum_{q \in Q} f_q X_q \quad (1)$$

s.t.

$$X_q \leq \sum_{j \in q} Y_j, \forall q \in Q \quad (2)$$

$$\sum_{j \in J} Y_j = p \quad (3)$$

$$X_q \in \{0, 1\}, \forall q \in Q \quad (4)$$

$$Y_j \in \{0, 1\}, \forall j \in J \quad (5)$$

In this formulation, the input data is:

$Q$  = the set of nonzero flow paths indexed  $q$

$J$  = the set of potential facility sites indexed  $j$

$j \in q$  = the set of potential facility sites along path  $q$

$f_q$  = the flow volume on the path  $q$

$p$  = the number of facilities to be located

and the objective function and decision variables are:

$Z$  = the objective function, total flows intercepted at least once

$$X_q = \begin{cases} 1 & \text{if the flow on path } q \text{ is intercepted by a facility along the path } q \\ 0 & \text{otherwise} \end{cases}$$

$$Y_j = \begin{cases} 1 & \text{if there is a facility located at potential facility site } j \\ 0 & \text{otherwise} \end{cases}$$

The objective function (1) is aimed at intercepting as much flow as possible, subject to the constraints that flow on path  $q$  cannot be intercepted unless there is at least one facility along path  $q$  (2), and that exactly  $p$  facilities are located (3). Constraints (4) and (5) are standard integrality conditions.

### 5.2.2. Aggregation Error

Since the early 1970s, two major issues in aggregation analysis in location modeling have received much attention in the literature: the identification and investigation of errors introduced by a given aggregation procedure; and the development of approaches for doing aggregation well. This literature is the subject of an excellent recent review by Francis et al. (2007). To my knowledge, no article has addressed the problem of aggregation in flow intercepting models. In fact, to my knowledge, only one large, real-world dataset has been applied to flow intercepting models, which described in Hodgson Rosing and Storrier (1996) and applied in several other articles by Hodgson.

Unlike point-based demand data, which is readily available at various levels of aggregation from census and other sources, authentic network flow data is difficult to obtain. Such data is garnered by traffic engineering teams; their collection is far beyond the capabilities of individuals or small groups of researchers. Fortunately, such data is important to transportation planning agencies and is collected in some form or another for most large North American cities. This data is sometimes made available to researchers; often the data is regarded by municipalities as “proprietary.”

The complexities of urban street networks results in monumentally large numbers of OD pairs within a city of even modest size; data is highly aggregated in the transportation planning process. Standard practice involves the creation of systems of traffic zones, aggregations of many origins and destinations into highly generalized networks of real traffic arteries and artificial “feeder” arcs. Flow data is usually highly aggregated in time as well as in space, and is presented, for instance, for an entire day, or the morning or evening peak period. Flow data is complex – it requires the knowledge of the assignment of OD flows to the individual links in the networks. Traffic engineers estimate the number of trips originating in, or destined to, each traffic zone (trip generation), the number of trips between each origin and each destination (trip distribution), and the exact paths that they take between origin and destination (trip assignment). The procedures to develop flow databases have been developed over the past half century, their inaccuracies are an accepted concomitant of their great utility in planning procedures.

As do municipal planners, location analysts must accept municipal traffic data as given. I term the data received from the municipal data base “unaggregated” – I consider the data to represent “true” flows. Even in its generalized form, municipal

flow data may be too large for tractable FILM problems. I address this problem in this chapter, as in the point-based aggregation literature: how may I reduce problem size, and what are the error consequences of doing so?

I take a straightforward approach to measuring the error arising from flow-based demand aggregation. For a smaller problem, based on smaller values of  $p$ , I assess how well solutions based on aggregated data compare with solutions based on unaggregated data. Defining:

$Z_{at}$  = True objective function values intercepted by model using aggregated data

$Z_{tt}$  = True objective function values intercepted by model using true data

I use a simple error measure:

$$E = \left[ \frac{Z_{tt} - Z_{at}}{Z_{tt}} \right] * 100$$

that represents the degree to which the model based on aggregated demand data fails to intercept true objective function values. The error term measures the effect of locational changes induced by using aggregated data.

### 5.2.3. GIS, Optimization and Heuristics for Location Analysis

A location problem may be characterized as identifying the placement of  $p$  facilities to serve a spatially distributed demand in a manner that optimizes a designated objective function. Spatial relations (e.g., topological and distance relations such as “neighbourhood,” “near,” and “far”) between demand points and facilities play a central role in location theory. A GIS is a system for management, analysis, and display of geographic knowledge, which is represented using a series of information sets such as maps and globes, geographic data sets, processing and work flow models, data models, and metadata (ESRI 2007). Commercial GIS engines (e.g., ArcGIS and MapInfo) provide a powerful tool for users to visualize and examine spatial relationships among entities, and to represent data in a way that may reveal patterns and relationships that are hard to detect using nonvisual approaches. However, location models with thousands of decision variables and constraints can be an information overload for commercial GIS engines. Commercial optimization engines (e.g., CPLEX) are reliable and easy-to-use engines for solving location models. However, optimization engines are not visual tools and are unable to solve large size location problems. In many cases, the use of optimization engines is not practical because it requires extensive CPU time and memory. Heuristics are approximate techniques to arrive at good solutions of large complex problems. But heuristics (e.g., 1-opt interchange) cannot guarantee optimal solutions; furthermore, algorithms for heuristics are problem-related, a universal heuristic algorithm is not available.

Location analysis, including aggregation analysis, can benefit tremendously from the integration of GIS, optimization, and heuristic technologies. A number of researchers (e.g., Hodgson and Neuman 1993; Jong and Eck 1996; Yeh and Chow 1996; Benoit and Clarke 1997; Vlachopoulou, Silleos and Manthou 2001; Church 2002; Yao and Thill 2006; Cheng, Li and Yu 2007; Murray et al 2007; Hernandez

2007) have studied the integration of GIS and optimization technologies to support facility location planning. A number of researchers (e.g., Densham and Rushton 1991; Li and Yeh 2007) have studied the integration of GIS and heuristic technologies to support facility location planning. This chapter reports on an effort toward integrating GIS, optimization, and heuristic technologies to study the special network flow structure of real-world transportation systems and to develop efficient methods for aggregating flow-based demand data. In an integrated system, each technology contributes to the system with distinctive features. In my case, ArcGIS 9.1 is the central system which is used to collect, organize, and visualize spatial data, to export data for models in CPLEX, to aggregate flow paths and networks, to visualize solutions, and to analyze how errors are introduced into the aggregated procedure. CPLEX 9.1 is used to optimally solve aggregated location problems and unaggregated problems with small  $p$  (the number of facilities). An efficient and robust 1-opt interchange heuristic is used to find approximate answers to difficult problems that cannot be given exact solutions in a reasonable amount of time.

### 5.3. A Case Study

My study uses 2001 afternoon peak traffic data for Edmonton, Alberta (the sixth largest city in Canada). The original data was provided by the City of Edmonton Transportation and Streets Department (TSD). TSD data has been gathered according to industry standards and the TSD forecasting model is respected throughout North America. TSD provided vehicle flows for a traffic network of 395 traffic zones, 2211 network nodes, 6211 links, and 149644 nonzero OD flow pairs for the afternoon peak in 2001. Flows are estimated vehicle flows for all pairs of traffic zones in the Edmonton area for 2001. Sophisticated methods of traffic assignment (of OD flow pairs to paths) exist, but in this study I use the most simple and commonly used methodology in the transportation literature by simply assigning all OD flows to the network over the least-time paths. I do this using CPLEX 9.1 and ArcGIS 9.1.

In my experience, the unaggregated network for FILM problems for any  $p < 21$  could be solved with CPLEX within 1000 minutes and aggregation errors should be zero or few. Optimal FILM solutions for the original problem for  $p = 1 \dots 20$  are documented in Table 5-1. The solutions for  $p = 1 \dots 11$  are optimally solved by ILOG-CPLEX 9.1 optimizer on a 2.8 GHz Intel Pentium processor with 1024 MB of RAM in Microsoft Windows 2000. The solutions for  $p = 12 \dots 20$  are the best solutions from 100 random-start runs of a 1-opt interchange heuristic written in FORTRAN. This heuristic is known to perform well on FILM problems (Hodgson, Rosing, and Storrier 1996). For known optimal solutions in my original and aggregated problems, the 1-opt interchange heuristic always finds an optimal solution within 10 runs. Heuristic concentration cannot improve these solutions: I use CPLEX to solve the original FILM problem where potential facility sites are limited to a concentration set created from the union of facility locations for  $p = 1 \dots 20$  over 100 times. My aggregation methods described below cannot improve these solutions either. Thus, I believe these solutions for  $p = 12 \dots 20$  are optimal solutions. Recall that even with the most efficient and specialized heuristics, good solutions to large flow-interception location

**Table 5-1: Optimal Solutions of the Original FILM Problem**

$p$	Flow %	Locations																			
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	6.7	115																			
2	12.0	115	1142																		
3	17.0	115	1140	2113																	
4	21.9	115	1142	2059	2113																
5	25.6	115	542	1142	2059	2113															
6	28.8	115	542	683	1142	2059	2113														
7	31.9	115	542	683	1140	2059	2113	2168													
8	34.7	115	542	683	1140	1236	2059	2113	2168												
9	37.2	115	542	683	482	1140	1236	2059	2113	2168											
10	39.6	115	542	683	482	695	1140	1236	2059	2113	2168										
11	41.8	115	542	683	482	695	999	1140	1236	2059	2113	2168									
12	43.7	115	542	683	482	695	999	1140	1236	1306	2059	2113	2168								
13	45.5	115	542	683	482	695	999	536	1140	1236	1306	2059	2113	2168							
14	47.3	115	542	683	482	695	999	536	1140	1236	1306	1608	2059	2113	2168						
15	49.0	115	542	683	482	695	999	536	1140	1236	1306	1608	1669	2059	2113	2168					
16	50.8	115	542	683	511	695	999	536	1008	1142	1211	1306	1608	2059	2113	2147	2168				
17	52.3	115	542	683	482	695	999	536	1007	1140	1259	1306	1608	1669	2059	2113	2147	2168			
18	53.9	115	542	683	482	695	999	536	1007	20	1140	1259	1306	1608	1669	2059	2113	2147	2168		
19	55.4	115	542	683	482	695	999	536	1007	20	754	1140	1259	1306	1608	1669	2059	2113	2147	2168	
20	56.7	115	542	683	511	695	999	536	1007	754	1142	1202	1211	1259	1306	1608	2059	2078	2113	2147	2168

problems rapidly exceed the capability of the personal computer. I believe that the 1-opt interchange heuristic cannot solve very large real-world FILM problems such as the traffic network of the U.S. Chicago region (Boyce and Bar-Gera 2003) which has 1790 traffic zones, 13,000 network nodes, 3,202,310 flow pairs, and a volume of flow data 100 times larger than the one in this chapter.

### 5.3.1. Flow Paths

Section 5.2.1 indicates that the number of paths is the most important factor in CPU times for FILM: the number of decision variables is the number of paths plus the number of network nodes (potential facility sites); the number of constraints is the number of paths plus one. Therefore, the first strategy for reducing CPU times for FILM is to reduce the total number of flow paths.

In afternoon peak traffic data for Edmonton, a large number of flow paths have very small flows and major flows are concentrated into limited number of paths. If 0.1%, 0.5%, 1%, 2%, 5%, and 10% total flows are removed by taking out smallest flow paths from the original network, 32%, 45%, 52%, 60%, 68%, and 77% of OD paths are removed, respectively. The implication of this observation is that removing a large number of small flow paths could only introduce few aggregation errors. If  $k\%$  total flows are removed, at most,  $k\%$  aggregation errors will be introduced to the original problem. I produce four aggregated networks by removing paths with less than 0.1000, 1.0000, 2.0000, and 5.0000 units of flow. I use ArcGIS 9.1 and C++ to reduce the associated OD paths and networks – removing all nodes and links which do not fall on the least-time paths. Table 5-2 describes flows, OD paths, and the structure of these networks. Network 1 is the original network. Network 2 removes 58% of paths by discarding only 2% of total flows. Network 3 removes 89% of paths by discarding only 26% of total flows. Network 4 removes 95% of paths by discarding 43% of total flows. Network 5 removes 98% of paths by discarding 69% of total flows. In these aggregated networks, a large number of paths are removed, while not many zones, nodes and links are removed (Table 5-2).

**Table 5-2: Five Transportation Networks**

Network		Total flow		OD path			Network Structure					
Flow	G%	Flow	F%	#	RP%	Avg	Max	Zones	RZ	Nodes	Links	
1	0.0000	0	69886	100	149644	0	35	79	395	0	2211	6211
2	0.1000	2	68261	98	62659	58	27	69	335	60	1922	5089
3	1.0000	26	51375	74	16488	89	19	59	290	45	1746	4606
4	2.0000	43	39688	57	8165	95	16	57	275	15	1692	4338
5	5.0000	69	21903	31	2423	98	11	44	242	33	1489	3403

*G%: percent of maximum errors; F%: percent of total flows; #: number of OD paths; RP%: percent of removed OD paths; Max: number of nodes on the longest OD path; Avg: average number of nodes on a path; RZ: number of removed zones.*

Table 5-3, Figure 5-1, and Figure 5-2 reveal the CPU time and errors for these networks. The CPU times for networks 1 to 3 rise sharply as  $p$  increases. Network 1

(the original network) is free of errors, but it cannot be solved within 1000 minutes for  $p > 11$ . The aggregated network 2 reduces the problem size sharply and has very few errors (an average of 0.2% errors, see Table 5-3). However, it still cannot be solved within 1000 minutes for  $p > 13$ . The aggregated network 3 can be solved within 365 minutes for  $p < 21$ , but it has an average of 10% errors. The aggregated network 4 can be solved within 15 minutes for  $p < 21$ , but it has very large errors (an average of 23.5% errors). The aggregated network 5 can be solved within 0.1 minutes for  $p < 21$ , but it has very large errors (an average of 34.5% errors). Although the aggregation error generally decreases with an increase in the number of facilities for  $p > 4$ , the relationship is not monotonic (Figure 5-2). I discover that if k% total flows are removed, the introduced aggregation error is much less than k%. Networks 2, 3, 4, and 5 induce an average of 0.2 %, 10.2%, 23.5%, and 34.5% errors after 2%, 26%, 43%, and 69% flows removals, respectively. Network 2 induces zero or few errors although it removes 58% paths, 2% flows, and 60 traffic zones.

**Table 5-3: Computation time and errors ( $p = 1...20$ )**

$p$	CPU minutes in network						errors % in network				
	1	2	3	4	5	#6	2	3	4	5 #6	
1	5.9	1.5	0.1	0.0	0.0	0.1	0.0	0.0	0.0	57.1	0
2	4.9	2.6	0.3	0.1	0.0	0.2	1.2	7.0	20.2	51.8	0
3	6.3	2.7	0.2	0.1	0.0	0.2	1.0	12.4	27.2	27.2	0
4	6.7	2.4	0.4	0.1	0.0	0.2	0.0	19.4	32.8	33.4	0
5	10.6	4.1	0.3	0.1	0.0	0.3	0.0	17.0	34.0	34.0	0
6	40.5	81.9	0.4	0.1	0.0	0.4	0.2	15.0	29.2	34.8	0
7	21.8	29.3	0.4	0.1	0.0	0.6	0.0	8.3	29.0	35.2	0
8	38.7	14.5	0.5	0.1	0.0	0.3	0.0	9.9	29.4	35.2	0
9	109.4	7.9	0.6	0.1	0.0	1.0	0.3	11.2	28.6	34.2	0
10	115.0	12.9	0.9	0.1	0.0	0.3	0.1	11.3	27.8	35.2	0
11	191.3	35.3	1.0	0.6	0.0	0.7	0.0	10.8	24.0	34.6	0
12	*1770	35.4	4.0	0.2	0.0	3.9	0.1	10.2	23.8	34.4	0
13		788.7	16.8	0.7	0.0	8.8	0.2	9.7	24.1	33.4	0
14		*1920	3.6	0.5	0.0	4.6		8.9	20.9	32.4	0
15			69.9	0.4	0.0	18.2		10.6	21.3	30.6	0
16			64.2	1.3	0.0	19.5		10.6	21.7	29.8	0
17			77.8	0.4	0.0	54.4		9.1	21.1	30.0	0
18			69.7	0.3	0.1	79.7		8.3	19.9	29.2	0
19			105.3	11.1	0.0	75.3		8.9	17.2	29.2	0
20			364.1	5.4	0.0	239.3		5.1	17.1	28.1	0
<b>Average</b>	$\infty$	$\infty$	<b>39.0</b>	<b>1.1</b>	<b>0.0</b>	<b>25.4</b>	<b>0.2</b>	<b>10.2</b>	<b>23.5</b>	<b>34.5</b>	<b>0</b>

\*: cannot be solved within these times; #6: an aggregated network in Section 5.3.2

I further investigate the removed network flow structure and how these removed flows introduce errors to the FILM problem. First, the average number of nodes on a path decreases as removed paths increase (Table 5-2), thus most removed



flow paths are very long paths. This pattern conforms to gravity spatial interaction models (e.g., Fotheringham and O’Kelly 1989) – OD flows are reversed proportional to distances between origins and destinations. Second, these removed zones orient toward the central city as the number of removed OD flow paths increases (Colour Figure 5-3). Most removed traffic zones are at the periphery of the city and have small flows (Figures 5-3 and 5-4). Finally, these 2% total flows from network 1 to network 2 disperse on  $(5089/6211)*100\% = 82\%$  links of the city (Figure 5-5). These 2% flows do not obviously change the original network flow structure (Figures 5-6 and 5-7). Thus the 2% flows have negligible effects on facility locations – only 7 facilities moved total 10 times for any  $p = 1 \dots 13$  (Figure 5-7). However, considering network 3, these 24%, removing flows (Figure 5-8) from network 2 to network 3 introduces many more errors because these removed flows greatly change the original network flow structure (Figure 5-7 and Figure 5-9).

In short, a large number of small and long flow paths have negligible effects on facility locations and associated objective function values. However, many errors will be induced if flow removals obviously change the original network flow structure. In this section, I reduce the number of paths by simply removing small flow paths. A future work is to reduce the number of paths as many as possible, changing network flow structures as less as possible.

### 5.3.2. Potential Facility Sites

Section 5.2.1 indicates that network nodes (potential facility sites) are second important factor in FILM computation times. The number of decision variables is the number of nodes plus the number of paths. The number of summation operations for path  $q$  in constraint (2) is equal to the number of nodes on path  $q$ . The number of summation operations in constraint (3) is equal to the total number of nodes in the network. Therefore, the second strategy for reducing FILM computation times is to reduce the number of network nodes. At first glance this strategy seems not as efficient as the first strategy: the number of constraints is equal to the number of paths plus one; the number of decision variables is the number of paths plus the number of nodes but the number of nodes is generally many fewer than the number of paths. However, many paths may have the same nodes after many nodes are removed and the total number of paths may be sharply reduced by aggregating paths. Therefore, this strategy may reduce the computation time sharply.

I indicate the rank of node with respect to the percent of total flow through it. The highest flow node intercepts 6.7% of total flows of the original network. If a node intercepting at least 2.0% of total flows is defined as a high flow node, there are 270 high flow nodes in network 1 (Figure 5-10). The number of high flow nodes decreases sharply as the requirement for high flow nodes increases (Figure 5-10). As an example, I select the top 270 high flow nodes as potential facility sites and remove all other low flow nodes. I further remove all low flow nodes on each path and aggregate paths which have the same nodes. The original 149,644 OD paths are aggregated into 15,936 OD paths. I termed this aggregated network as network 6. The 270 high flow nodes intercept 95% of paths and 87% of flows. Recall that the optimal solution at  $p =$

20 intercepts 56.7% of total flow (Table 5-1). Network 6 with 15,936 paths, 270 nodes and 87% of flows is much smaller than network 3 with 16,488 paths, 1746 nodes, and 74% of flows. This is to say that I can greatly aggregate the original network without missing much of total flow by using this method. CPLEX solves network 6 within 240 minutes for  $p = 1 \dots 20$  (Table 5-3) and finds the same solutions as Table 5-1. Therefore, the original network is aggregated without error at  $p = 1 \dots 20$  by this method.

I examine why this method is so efficient in my study area. FILM is aimed at maximizing the number of consumers who encounter at least one facility along their predetermined journeys. Thus, FILM solutions avoid flow cannibalization. The 270 high flow nodes are concentrated into a few large arterial roads, not avoiding flow cannibalization (Figure 5-11). However, the 270 high flow nodes include the union (26 nodes) of optimal locations of the original FILM problem at  $p = 1 \dots 20$  (Figure 5-11). This is the reason why network 6 does not induce aggregation errors for the FILM problem at  $p = 1 \dots 20$ .

I further examine why all FILM locations are at a limited number of high flow nodes and whether other real-world transportation networks have these characteristics. FILM has nodal manifestation—location at a node is always at least as good as location on a link because either endpoint of a link can intercept all its flow. In real-world transportation systems, travel patterns tend to focus on central business districts and movements tend to converge from local streets onto larger arterial roads and expressway systems. There is no point in locating FILM facilities on local streets since almost all flows intercepted by a node on a local street are intercepted by at least one intersection with a larger arterial road. A real-world transportation network has many more local streets than arterial roads, thus a large number of low flow nodes can be removed. Therefore, I suspect that in most real-world transportation systems FILM may have this high flow nodal manifestation – a FILM location is always at a network node which is good at intercepting many total flows.

#### **5.4. An Integrated System for Flow-Based Demand Aggregation**

Here, I propose an integrated GIS, optimization, and heuristic system for flow-based demand aggregation (see Figure 5-12). Most of these steps are coded in Microsoft Visual C++ 6.0 (see Appendices).

- Step 1: Map and examine the original network flow structure. Remove  $k\%$  total flows by taking out smallest flow paths from the original network. Note that if  $k\%$  total flows are removed, at most,  $k\%$  aggregation errors will be introduced to the original problem. The value of  $k\%$  could be selected according to acceptable aggregation errors and the size of network.
- Step 2: Map and compare the aggregated network flow structure with the original network flow structure to roughly guess aggregation errors. The less the original network flow structure is changed by flow removals; the less aggregation errors could be introduced. Note that if  $k\%$  removed flows do not obviously change the original network flow structure, the introduced aggregation error could be much less than  $k\%$ ; if  $k\%$  removed flows

obviously change the original network flow structure,  $k\%$  aggregation errors may be introduced.

- Step 3: Rank every potential facility site with respect to the total flow through it. Map the number of high flow nodes vs. the least % flow intercepted by a high flow node such as Figure 5-5. Analyze the map and designate the top  $m$  nodes as high flow nodes and others as low flow nodes.
- Step 4: Designate the  $m$  high flow nodes as potential facility sites. Aggregate paths by removing low flow nodes on each path and combining paths having the same high flow nodes.
- Step 5: Output aggregated paths for heuristics.
- Step 6: Output high flow nodes for heuristics.
- Step 7: Solve the flow-interception model with a heuristic
- Step 8: If the problem is too large for heuristics to solve, go back to step 4 and reduce  $m$ . Repeat the process until the heuristic could solve the model.
- Step 9: If the problem can be solved, run the heuristic a number of times and create a concentration set from the union of all locations of these solutions.
- Step 10: Designate the concentration set as part of the potential facility sites.
- Step 11: Designate the  $m$  high flow nodes as part of the potential facility sites.
- Step 12: Map and analyze the concentrated set and high flow nodes. Create a full set of potential facility sites and further aggregate flow paths.
- Step 13: Output aggregated paths for CPLEX.
- Step 14: Output the full set of potential facility sites for CPLEX.
- Step 15: Solve the flow-interception model with CPLEX.
- Step 16: If the problem is too large for CPLEX to solve, go back to step 8 and reduce  $m$ . Repeat the process until CPLEX could solve the model.
- Step 17: If CPLEX can solve the problem, map and report results in ArcGIS.

As shown in Figure 5-12, GIS is central to this integrated multi-technologies system for flow-based demand aggregation – most analyses and operations are completed in the GIS environment. Heuristics are used to find a solution set for part of the potential facility sites. Optimization engines are used to solve small size flow-interception problems. Note that this aggregation system can be used to aggregate other flow-interception location problems where flows are objective function values and network flow structures are network objective function value structures.

### 5.5. Conclusion

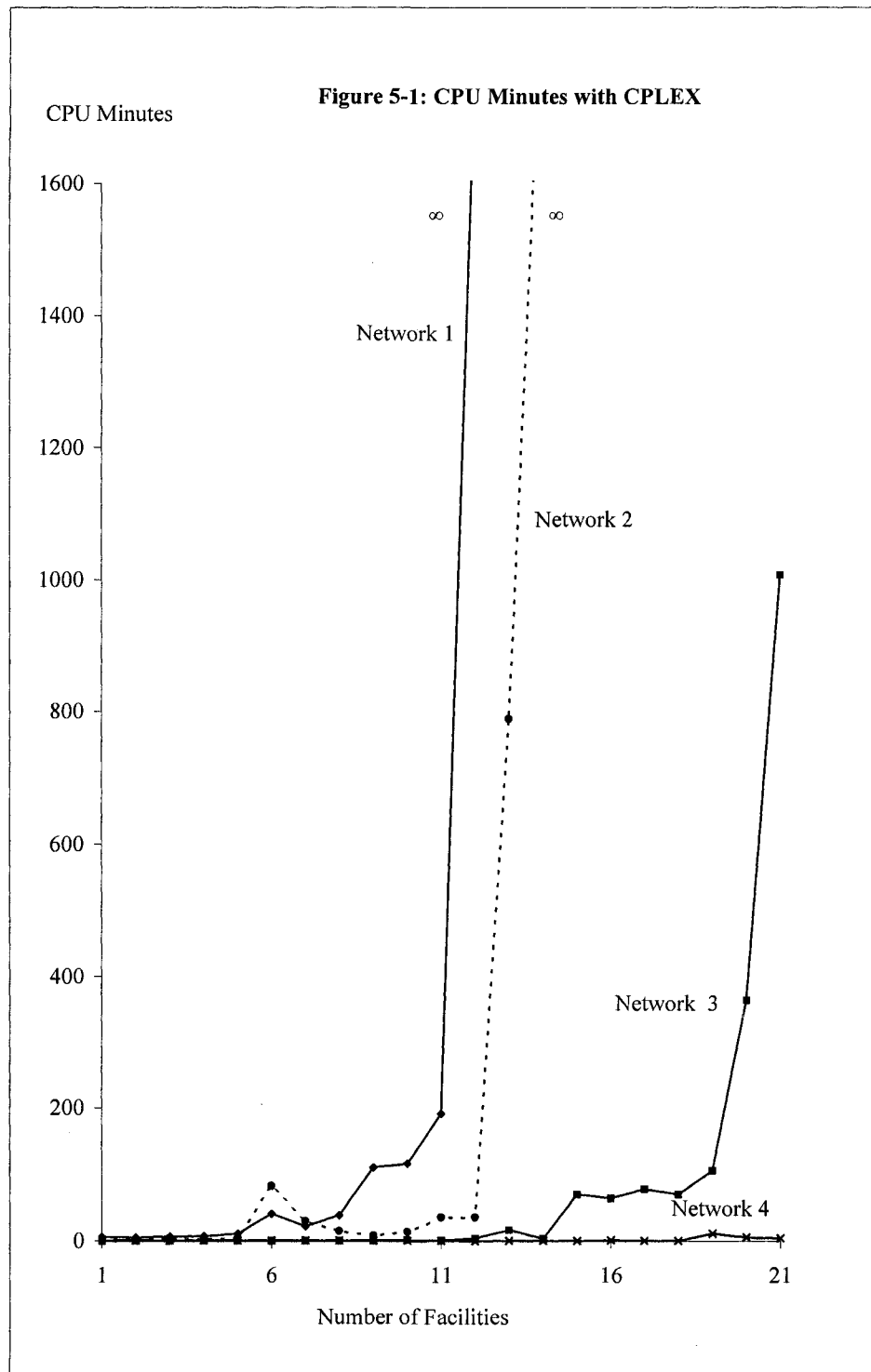
Location analysis as a field can benefit tremendously from the integration of GIS, optimization, and heuristic technologies. Flow-based demand aggregation is extremely important because large real-world flow-interception modeling and analysis rely mostly on aggregated data. In this chapter, I integrate these technologies to examine the real-world network flow structure of afternoon peak traffic data for Edmonton, Alberta in the year 2001. I discovered that real-world transportation systems may have very special network flow structures: urban residential, workplace, and shopping distributions may process considerable order, and the corresponding transportation networks and travel patterns result from complex processes that react to that order.

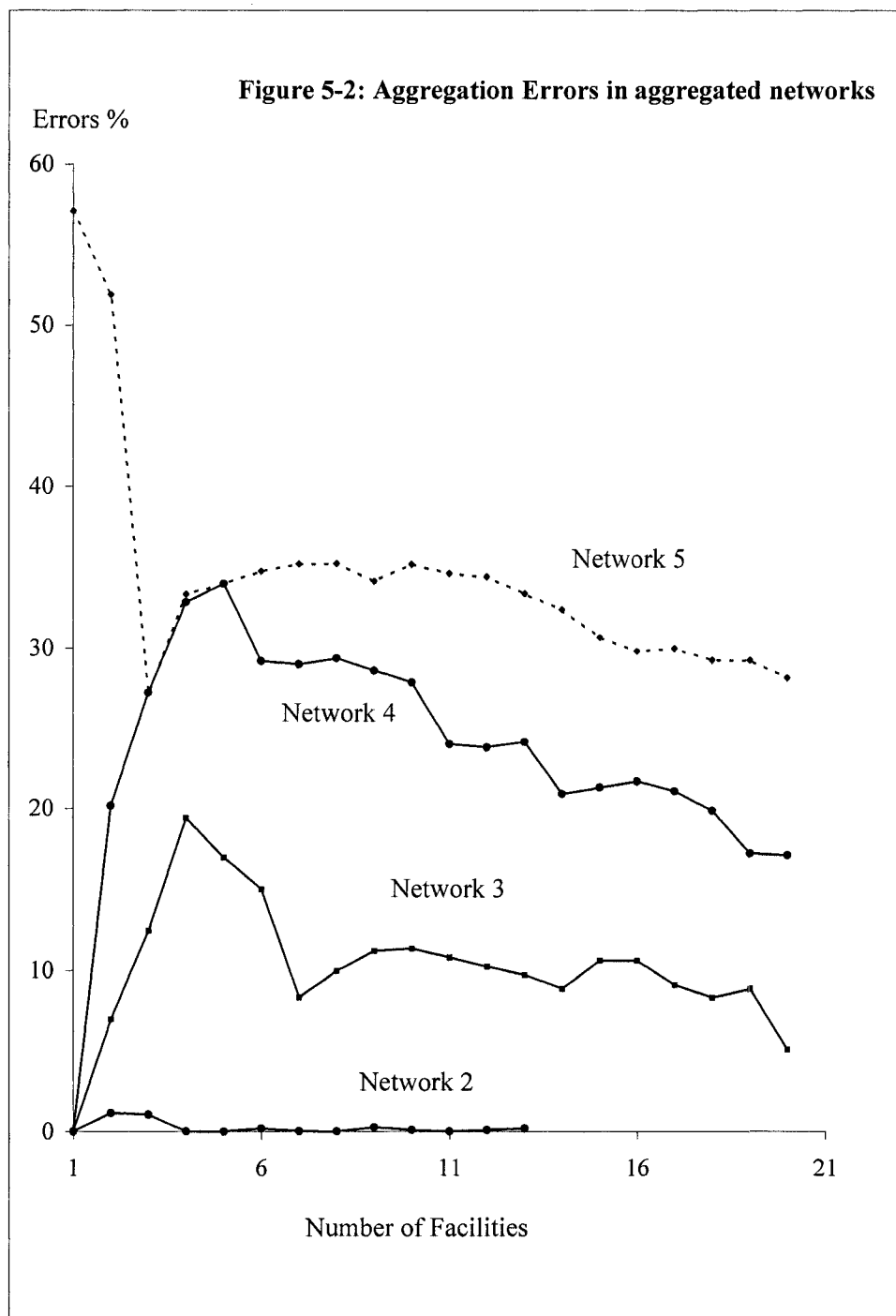
Travel patterns tend to focus on central business districts, and movement tends to converge from local streets onto larger arterial roads and expressway systems. In these urban systems, a large number of paths have very small flows and major flows are concentrated into a limited number of paths; network flows are highly concentrated into several larger arterial roads. Because of these special network flow structures, a small number of nodes can intercept most flows in a transportation network. FILM has high flow nodal manifestation (every FILM location is a high flow network node). This chapter further integrates GIS engines, optimization engines, and heuristics to develop a system of efficiently aggregating data for large real-world flow-interception location problems. Application of the integrated system to the standard flow-interception model with Edmonton, Alberta afternoon peak traffic data is very efficient, inducing zero aggregation errors.

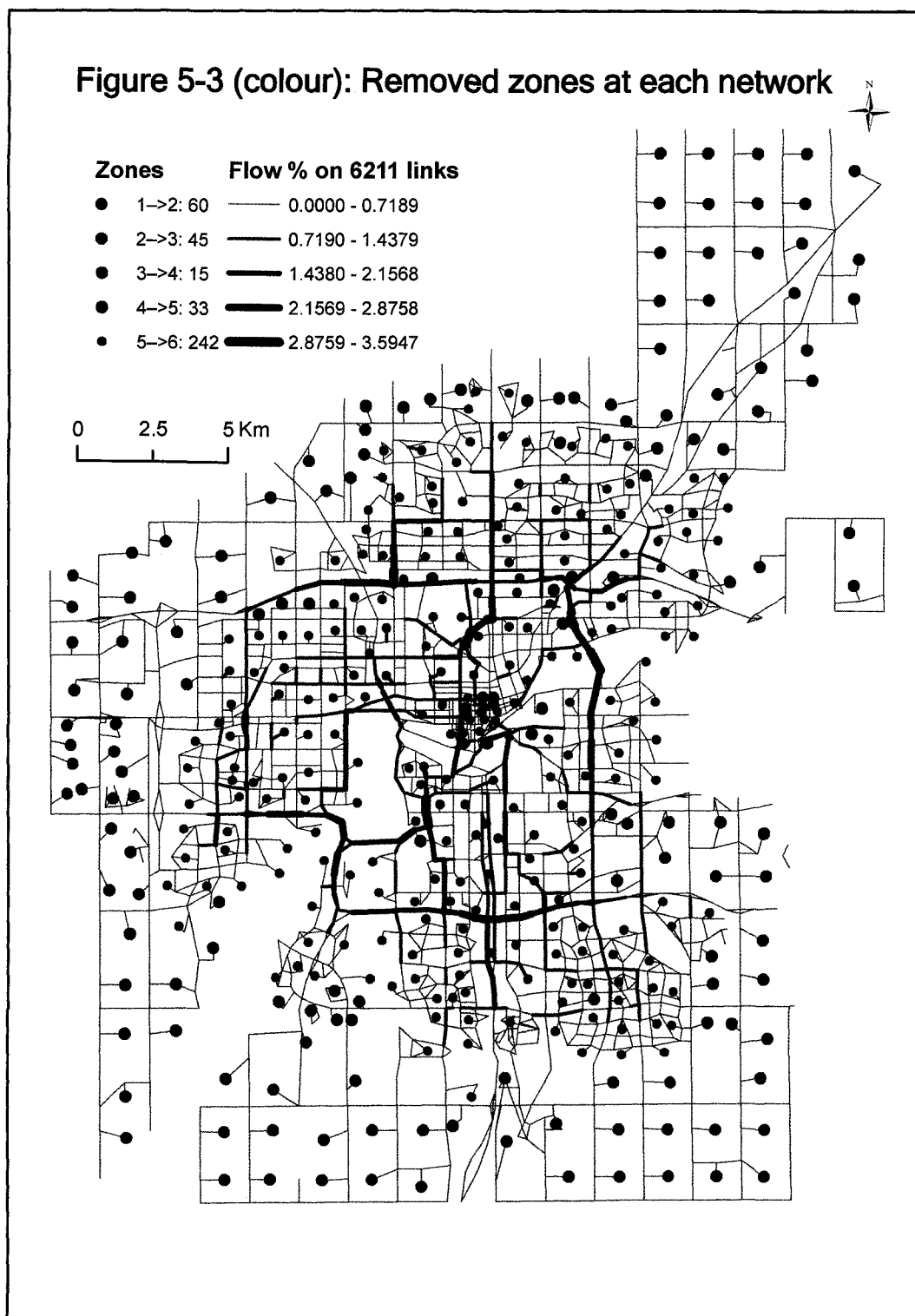
### References

- Benoit, D., G P Clarke. 1997. Assessing GIS for retail location planning. *Journal of Retailing and Consumer Services* 4 239-258.
- Berman, O., R. C. Larson, N. Fouska. 1992. Optimal location of discretionary service facilities. *Transportation Science* 26 201-211.
- Boyce, D., H. Bar-Gera. 2003. Validation of multiclass urban travel forecasting models and combining origin-destination, modal, and route choice. *Journal of Regional Science* 43, 517-540.
- Cheng, W.L., H. Li, L. Yu. 2007. A GIS approach to shopping mall location selection. *Building and Environment* 42, 884-892.
- Church, R. L. 2002. Geographical information systems and location science. *Computers & Operations Research* 29 541-562.
- Densham, P.J., G., Rushton. 1991. Designing and implementing strategies for solving large location-allocation problems with heuristic methods. Technical Report 91-100, National Center for Geographical Information and Analysis. University of California, Santa Barbara, CA.
- Doblas, J., and F. G. Benitez. 2005. An Approach to estimating and updating origin-destination matrices based upon traffic counts preserving the prior structure of a survey matrix. *Transportation Research B* 39 565-591.
- ESRI. 2007. GIS: Getting Started. [http://www.esri.com/getting\\_started/index.html](http://www.esri.com/getting_started/index.html)
- Fotheringham, A.S., M.E. O'Kelly. 1988. *Spatial Interaction Model: Formulations, and Applications*. Kluwer.
- Francis, R.L., T.J. Lowe, M.B. Rayco, A. Tamir. 2007. Aggregation error for location models: survey and analysis. *Annals of Operations Research*, in press.
- Hernandez, T. 2007. Enhancing retail location decision support: The development and application of geovisualization. *Journal of Retailing and Consumer Services*, article in press.
- Hodgson, M. J. 1990. A flow-capturing location-allocation model. *Geographical Analysis* 22 270-279.
- Hodgson, M. J., K. E. Rosing, A. L. G. Storrier. 1996. Applying the flow-capturing location-allocation model to an authentic network: Edmonton, Canada. *European*

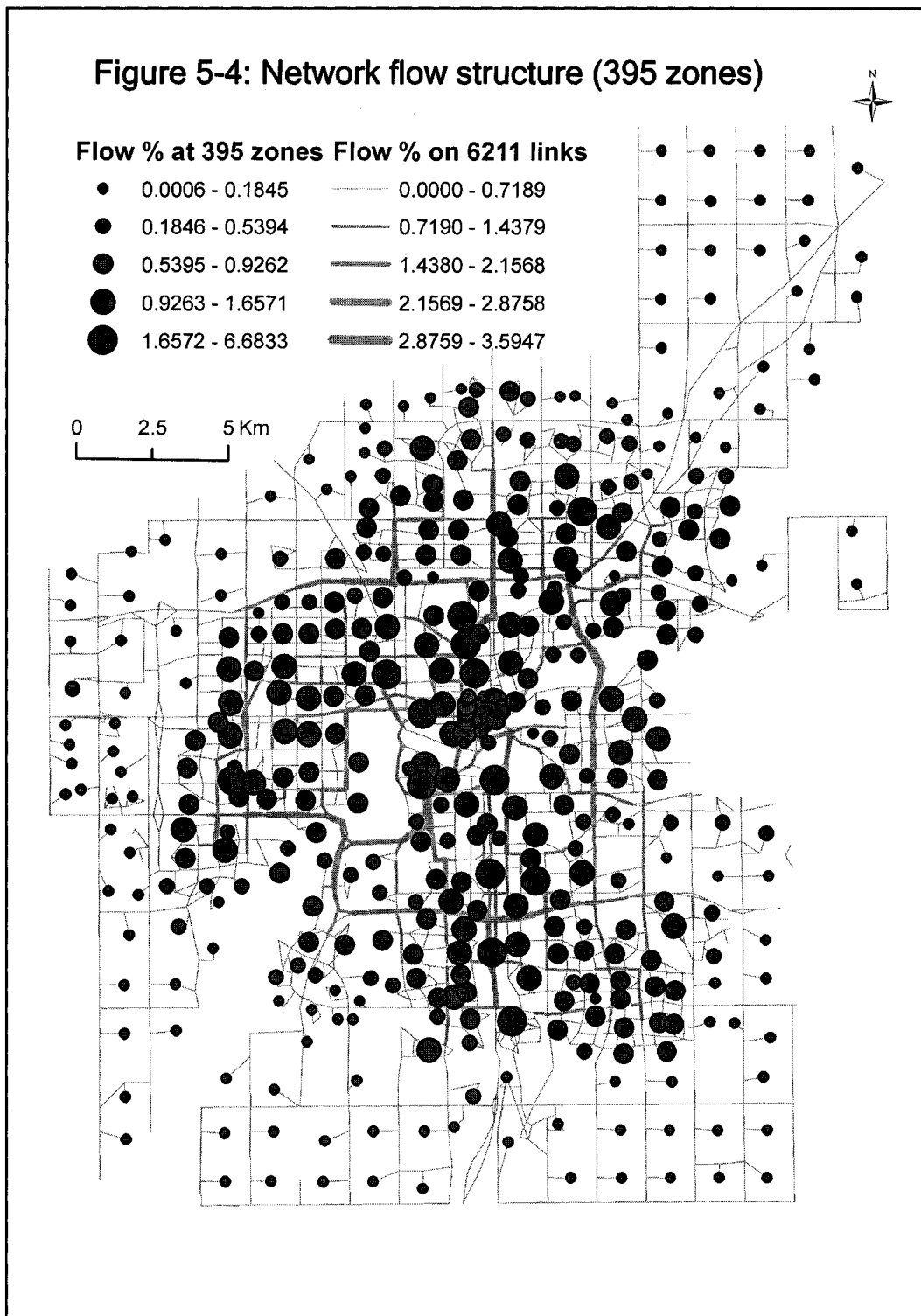
- Journal of Operational Research* **90** 427-443.
- Hodgson, M. J., S. Neuman. 1993. A GIS approach to elimination Source C aggregation error in  $p$ -median models. *Location Science* **1**, 155-170.
- Jong, T. D., J. R.V. Eck. 1996. Location profile-based measures as in improvement ton accessibility modelling in GIS. *Computers, Environment and Urban Systems* **20** 181-190.
- Li, X., A. G. Yeh. 2007. Integration of genetic algorithms and GIS for optimal location search. *International Journal of Geographical Information Science* **19** 581-601.
- Murray, A. T., K. Kim, J. W. Davis, R. Machiraju, R. Parent. 2007. Coverage optimization to support security monitoring. *Computers, Environment and Urban Systems* **31** 133-147.
- ReVelle, C.S., R. Swain. 1970. Central facilities location. *Geographical Analysis* **2** 30-42.
- Vlachopoulou, M. G. Silleos, V. Manthou. 2001. Geographic information systems in warehouse site selection decisions. *International Journals of Production Economics* **71** 205-212.
- Yao, X., J. Thill. 2006. Spatial queries with qualitative locations in spatial information systems. *Computers, Environment and Urban Systems* **30** 485-502.
- Yeh, A. G., M. H. Chow. 1996. An integrated GIS and location-allocation approach to public facilities planning – an example of open space planning. *Computers, Environment and Urban Systems* **20** 339-350.

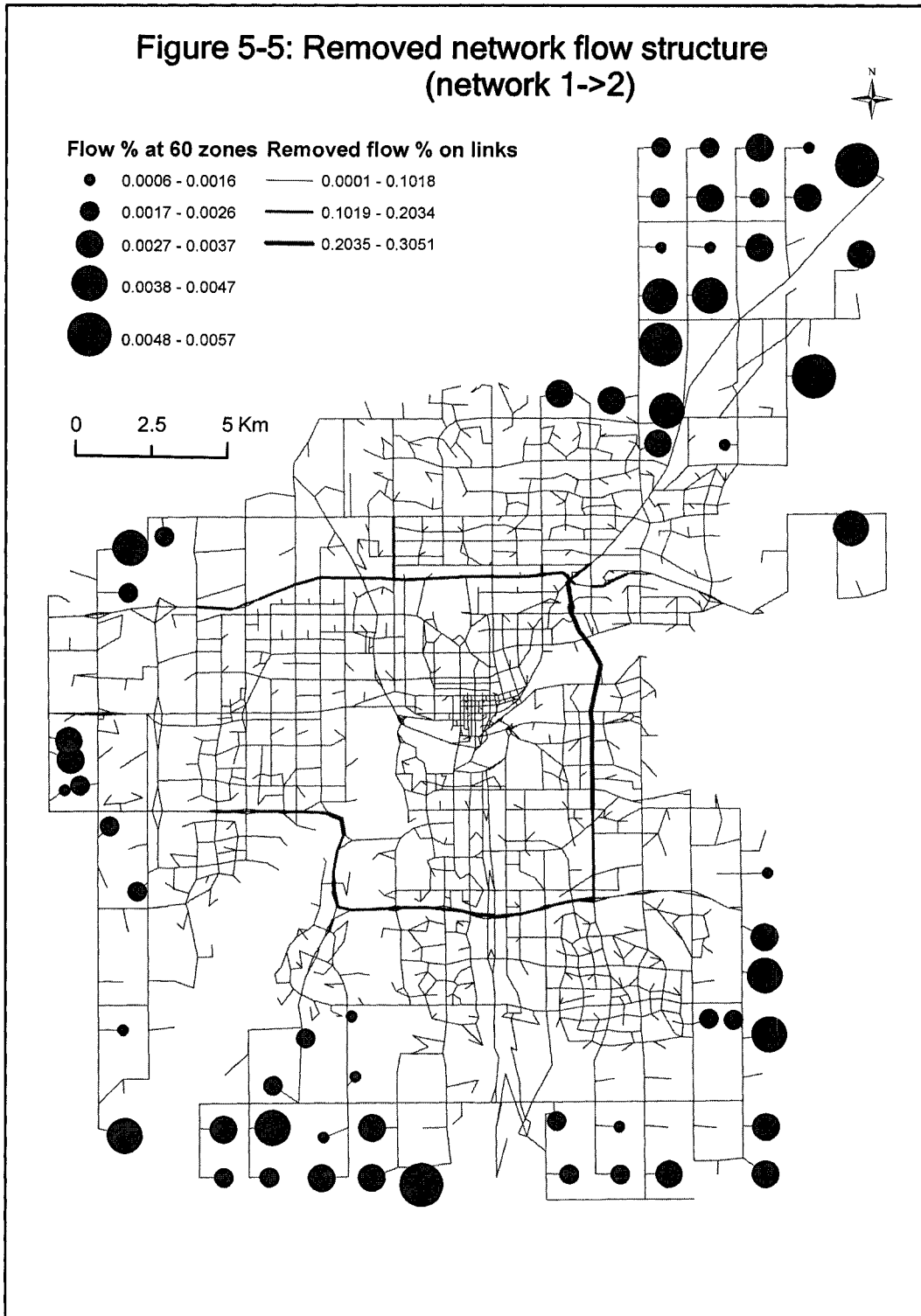


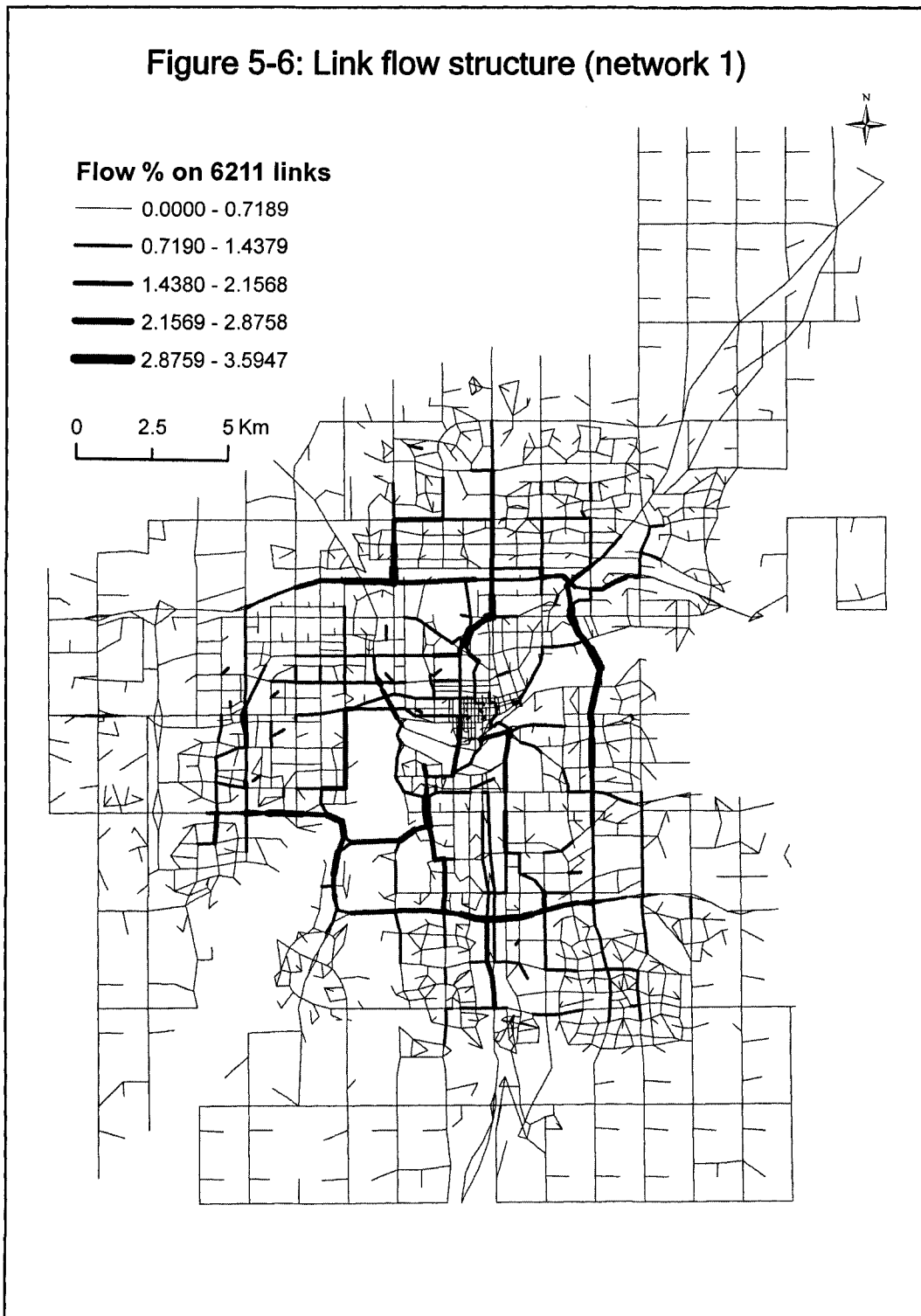




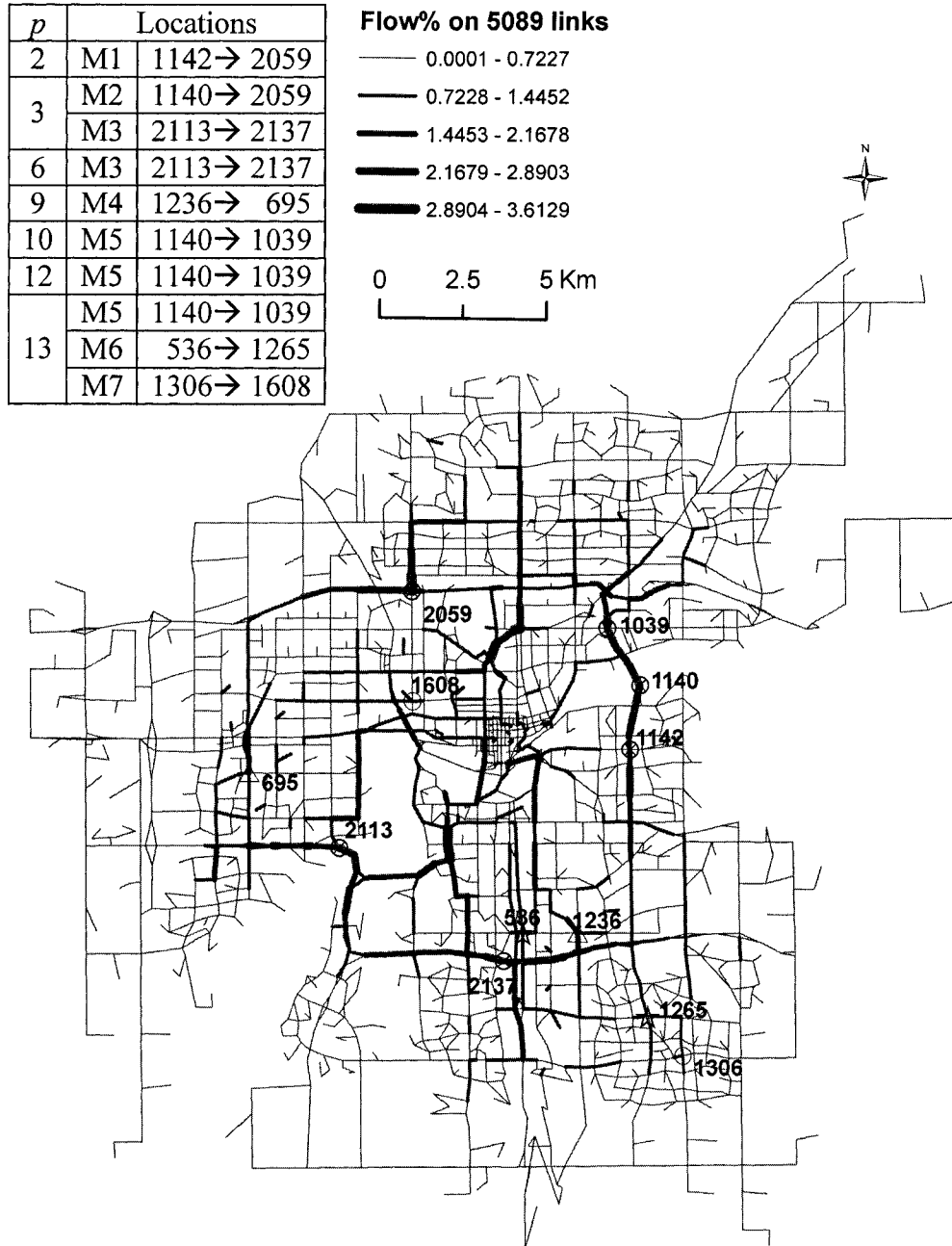


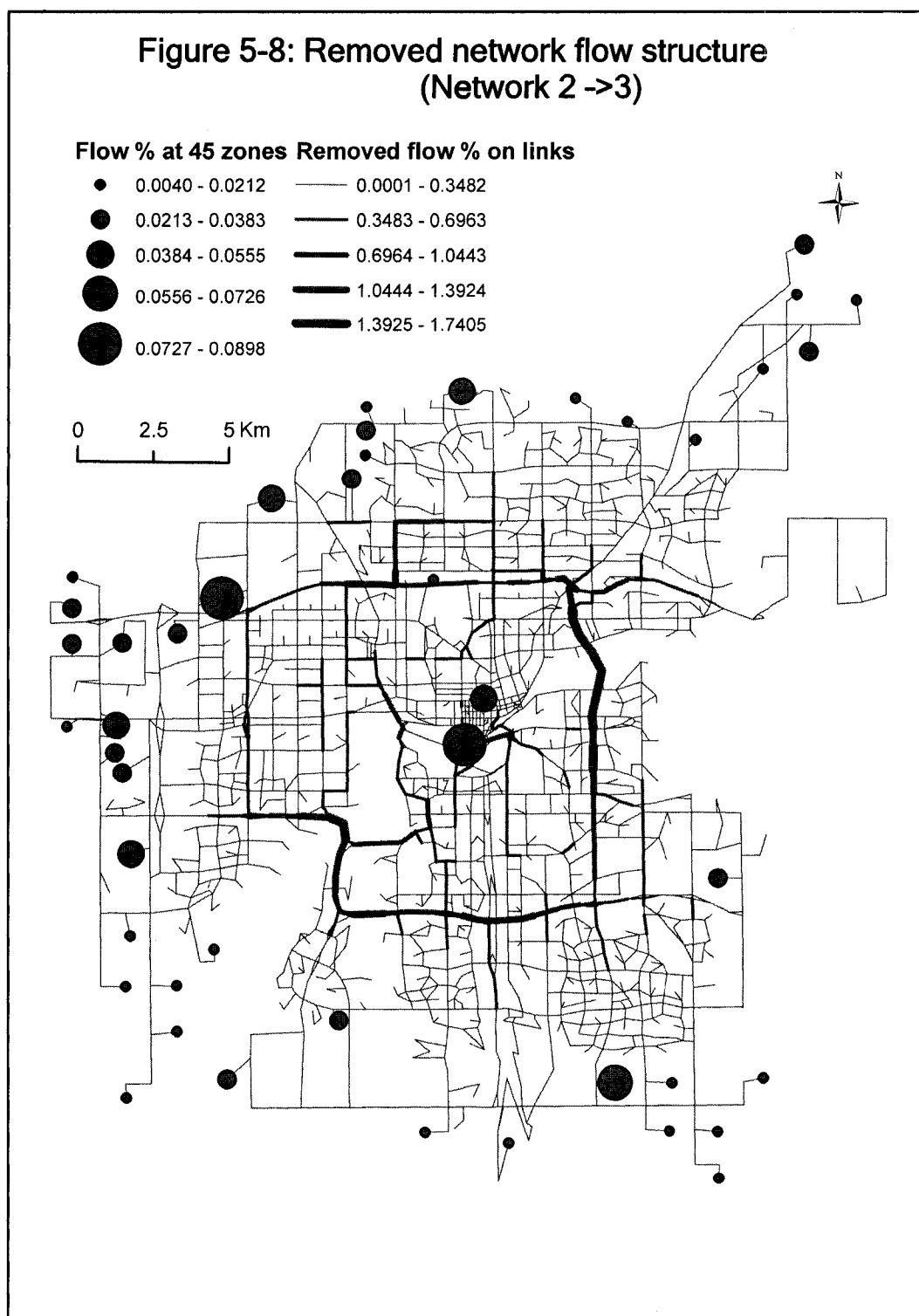




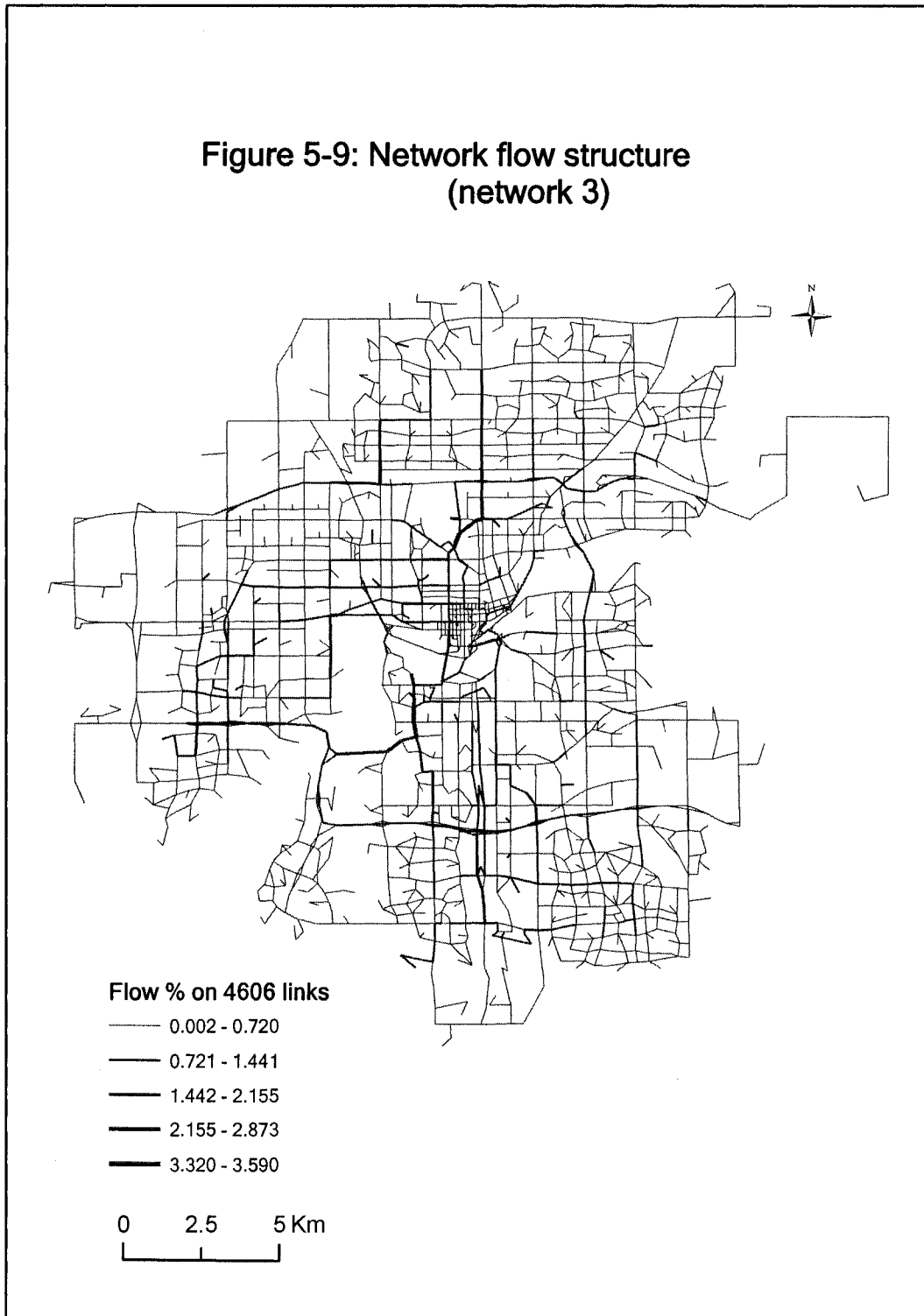


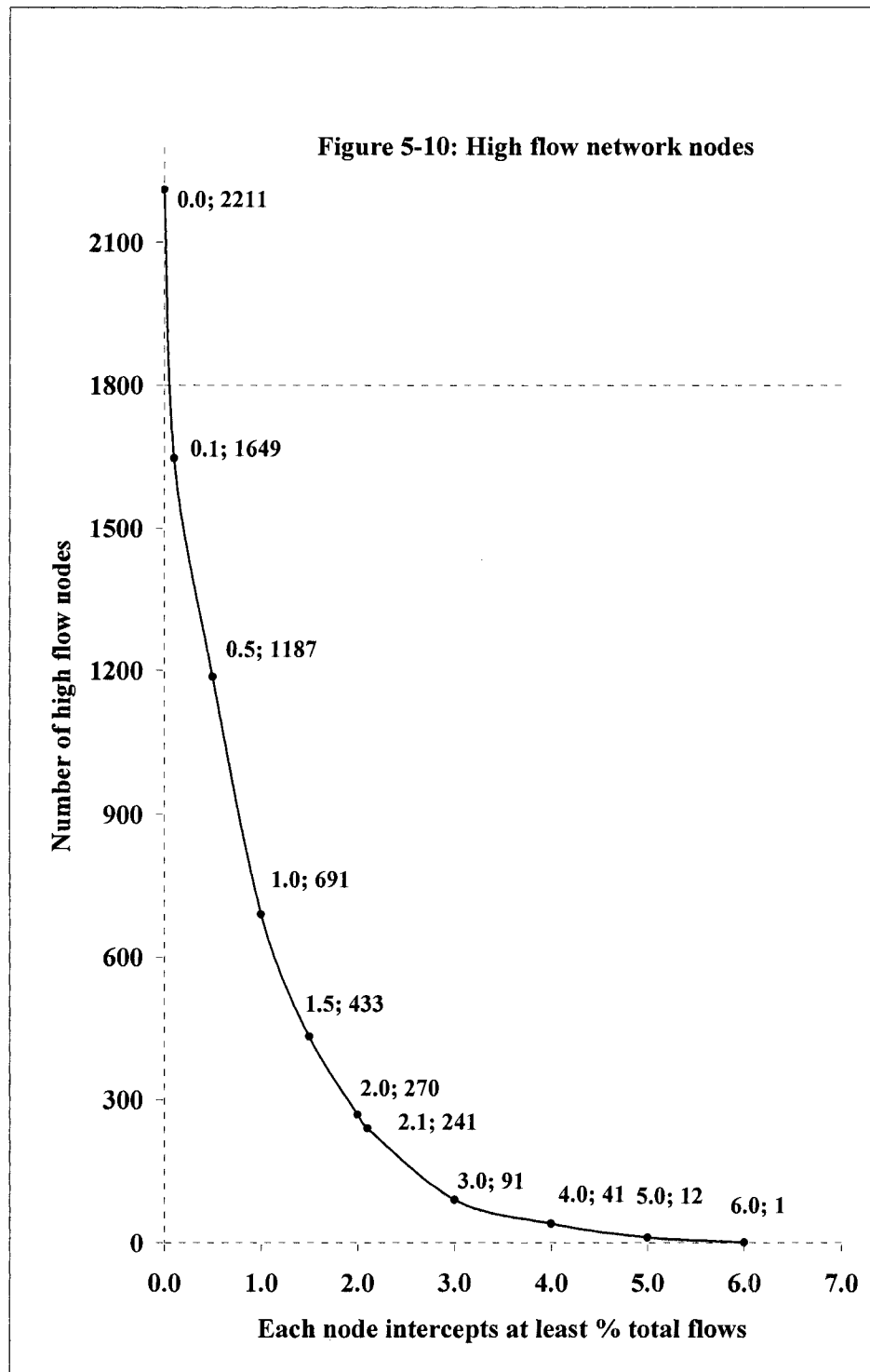
**Figure 5-7: Locations movement and network flow structure (network 2)**





**Figure 5-9: Network flow structure  
(network 3)**





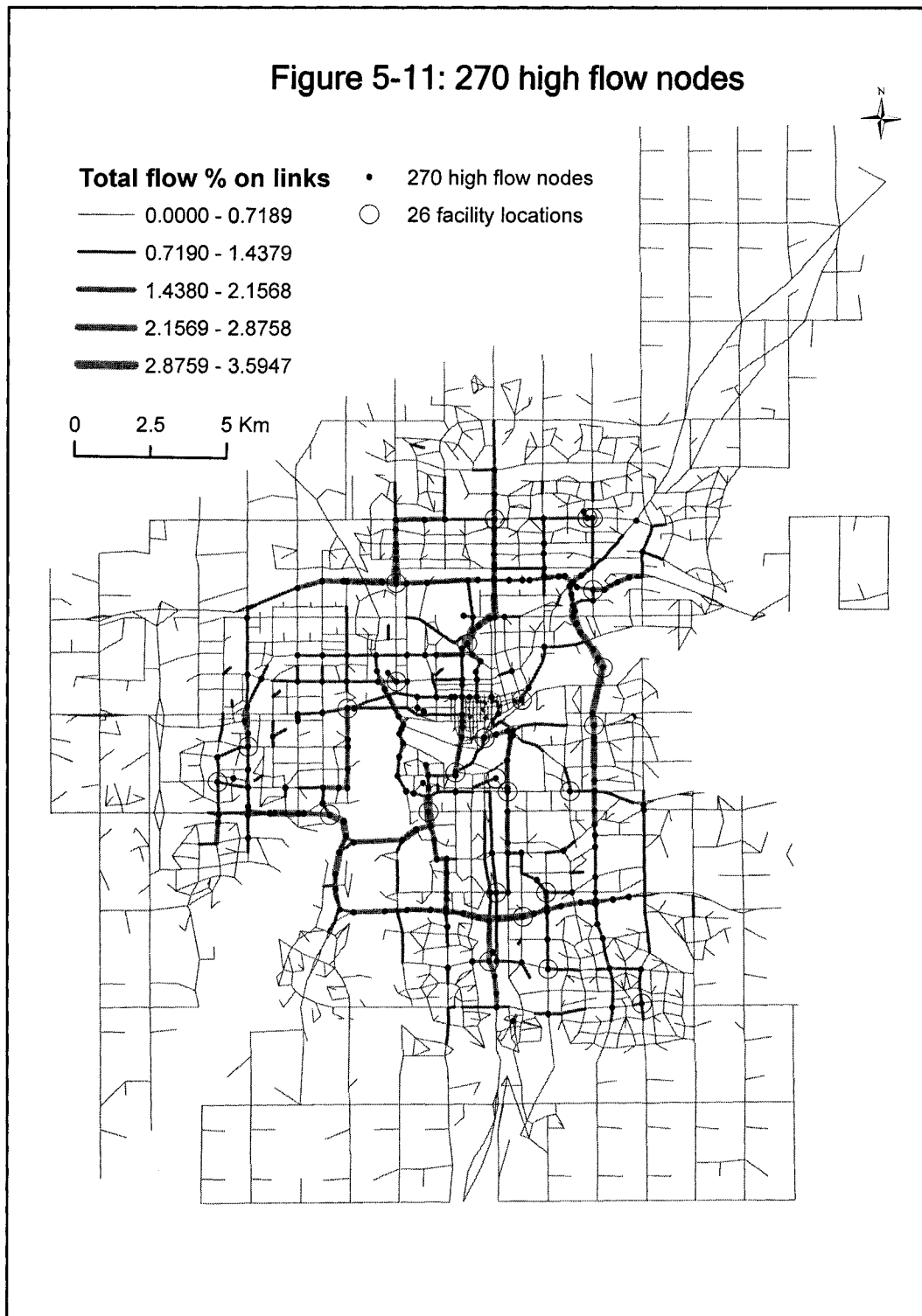
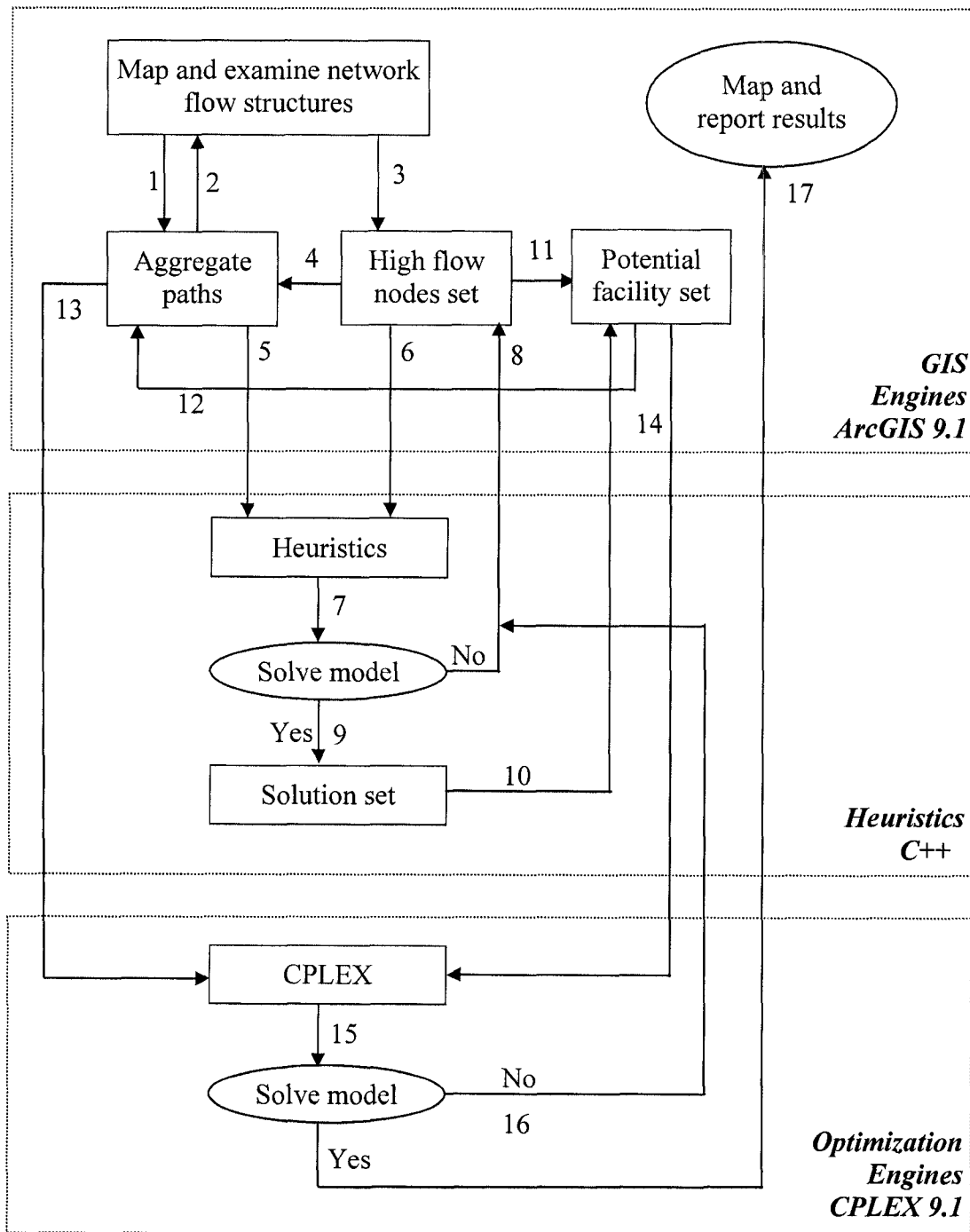




Figure 5-12: An integrated system for flow demand aggregation



## Chapter 6

### Conclusions and Future Research

Facility location planning is a key decision in the long-term efficiency of operations. Location models provide decision makers with strategic and quantitative support in seeking locations where fixed and operating costs can be kept low, while accessibility to markets can be kept high. Geographic information systems (GIS) provide powerful tools for users to visualize, examine, and analyze spatial location patterns and relationships. Heuristics and data aggregations are approximate techniques used to arrive at good solutions of large complex problems. This dissertation integrates GIS, optimization modeling, aggregation, and heuristic methodologies to study facility location planning on a network with different types of consumers. The major contributions of this dissertation are as follows.

**Chapter 2:** Traditional flow-interception location models (FILM) locate facilities such that the gross amount of intercepted flows is maximized; flows are intercepted or not, there is no indication of where in the trip flows are intercepted. Nor is there any impetus to prefer any location over another. In the real world, however, consumers often desire to receive services at or near a specific location along their trips, frequently at trip origin or destination. This chapter extends the traditional notion of flow interception to propose a pickup model (PUP) for considering consumers' locational and proximity preferences, accommodating our understanding of geographical advantages and consumer behaviour. PUP transforms the standard flow-interception *location* model (Hodgson 1990; Berman, Larson, and Fouska 1992) to a flow-interception *location-allocation* model, providing a fruitful garden for further research. This chapter integrates GIS and optimization engines to investigate PUP in real-world transportation systems. My examples, using morning and afternoon peak traffic flows in Edmonton (the sixth largest Canadian city), demonstrate that solutions of PUP are superior to solutions of traditional flow-interception models if consumers have locational preferences.

**Chapter 3:** Location researchers tend to introduce changes in objective functions and/or assumptions by developing new models. About 30 FILM models have been proposed in about 40 academic publications during the past 17 years. This has led to many disparate models, each requiring a somewhat different solution method, hampering the development of standardized software that would encourage widespread use in real-world, strategic decision making processes. This chapter formulates a generalized flow-interception location-allocation model (GFIM) to effectively solve current and future flow-interception location problems. Most current flow-interception location problems can be solved by simple parameter manipulations in GFIM's input. Additional flow-interception problems can be solved by manipulating or adding simple constraints to GFIM. Several critical considerations in flow-interception models – such as deviation from predetermined journeys, locational and proximity preferences, and capacity issues – can be handled within the single framework. Two real-world examples, morning and afternoon peak traffic for the city

of Edmonton in Canada, show that CPLEX optimally solves GFIM much more efficiently than it does the classic flow-interception location model. GFIM clearly provides a standardized benchmark for current and future flow-interception models in the academic literature.

**Chapter 4:** Traditional location theory views consumers as travelling from static and fixed points (e.g., homes); their convenience is measured by distance from these points to the nearest facility (Type A consumers). FILM theory views consumers as flows traveling on predetermined paths (e.g., daily commute between home and workplace); their convenience is measured by distance from these paths to a facility (Type B consumers). In the real world, the decision to use a facility based on its proximity to home or to a work trip is probably not predetermined. More likely, consumers choose a facility based on its greater convenience to either their home or their travel path. This dissertation identifies this type of consumer for the first time in the literature, calling them Type C consumers. Most people in the real world are Type C consumers; they are not as selective of location as point-based consumers (Type A) and flow-based consumers (Type B). The literature has neglected Type C consumers. My examples, using afternoon peak traffic data for the city of Edmonton in Canada, show that solutions identified by Type C consumers are more robust than solutions identified by Type A and B consumers. Therefore, considering Type C consumers substantially improves the location modeling outcome. Location researchers have traditionally developed different models for different consumer types. A generalized and efficient strategy is developed to unify numerous network location models by unifying various consumer types and objectives. A generalized location-allocation model (GLAM) is formulated to effectively and efficiently encompass at least 60 existing models, including the  $p$ -median (ReVelle and Swain 1970), maximal covering location model (Church and ReVelle 1974), FILM, and numerous variants of these models.

**Chapter 5:** Most location studies use spatially aggregated data, and demand point aggregation has received considerable interest in both industry and academia. Systematic studies of flow demand aggregation have not, however, been reported to date. The huge volume of flow demand data is the bottleneck of applying flow-interception problems to real-world situations. This chapter integrates GIS, optimization, and heuristics to examine the special network flow structure of real-world transportation systems and to develop a system of efficiently aggregating flow-based demand data for location models. This chapter applies this system to the classic FILM model using 2001 Edmonton afternoon peak traffic data and finds it to be effective and free of aggregation error.

Aside from the development of these theories and models, my development of up-to-date, real-world transportation networks provides a realistic test-bed for flow-interception location models.

This dissertation provides a rich background for future research as follows.

- i. Chapter 4 accommodates a more realistic view of consumers, Type C consumers. The literature has neglected Type C consumers. It will be valuable to further investigate, by surveys and other means, Type C consumers in real-

- world spatial analysis problems.
- ii. GIS supports a wide range of spatial queries that can be used to support location research. Church (2002) concluded that GIS will play a significant role in future location model development and applications. There are six location models available in ARC/INFO. Since at least 60 existing location models, including most location models in ARC/INFO, can be transformed into GLAM, it will be valuable to implement GLAM in ARC/INFO or ArcGIS. Since the generalized and efficient strategy in chapter 4 provides a theory for unifying location models and reduces location problem size, it is applicable to facility location and planning in geographic information systems.
  - iii. I note that Kuby's flow refuelling location model (FRLM) (Kuby and Lim 2005, 2007) is a location, not a location-allocation model; it cannot consider at which facilities flows are intercepted. Therefore, FRLM cannot directly further consider capacity, locational and proximity preferences or deviation from predetermined journeys: all critical issues in location analysis. Chapter four has demonstrated that GFIM effectively solves FRLM problems. Since GFIM is a location-allocation model, it can expand the study of refueling to consider these critical issues.
  - iv. Since large real-world flow-interception problems may be aggregated with small or no aggregation errors, it is valuable to further study the special network flow structure of real-world transportation systems and to develop clever integrated systems of efficiently aggregating flow demand data.
  - v. There is no competitive, deterministic flow-interception study reported in the literature. Since GLAM is able to consider an individual consumer's consideration of a specific facility, it could be applied to competitive flow-interception problems.

### References

- Berman, O., R. C. Larson, N. Fouska. 1992. Optimal location of discretionary service facilities. *Transportation Science* **26** 201-211.
- Church, R. L. 2002. Geographical information systems and location science. *Computers & Operations Research* **29** 541-562.
- Church, R., C. ReVelle. 1974. The maximal covering location problem. *Papers of the Regional Science Association* **32** 101-1018.
- Hodgson, M. J. 1990. A flow-capturing location-allocation model. *Geographical Analysis* **22** 270-279.
- Kuby, M., S. Lim. 2005. The flow-refuelling location problem for alternative-fuel vehicles. *Socio-Economic Planning Sciences* **39** 125-145.
- Kuby, M., S. Lim. 2007. Location of alternative-fuel stations using the flow-refueling location model and dispersion of candidate sites on arcs. *Networks and Spatial Economics* **7** 129-152
- ReVelle, C.S., R. Swain. 1970. Central facilities location. *Geographical Analysis* **2** 30-42.

## Appendices: Codes

### Appendix 1: Implementing FILM in AMPL CPLEX

#### Running Script File: FILMfast.run

```

-----
# include FILMfast.run;
option solver cplex;
model FILMfast.mod;
read Q,J, p < FILMfast.txt; # read Q, J, P
read {q in 1..Q} # read flow, N, path from data file FILMfast.txt
    (flow[q],N[q],{t in 1..N[q]}path[q,t]) < FILMfast.txt;
option print_separator ","; # separator with "," in writing file
printf "p,?Optimal,Second,Z,Solution\n">FILMresults.txt; # write title line
set SetP := 1 ..4; # set the range of p facilities
for {c in SetP} {
    let p := c;
    solve; # The follow commands are to print results in two files
    printf "%u,%u,% .1f,% .2f," p,solve_result_num,_solve_user_time,Z>FILMresults.txt;
    print {j in 1..J:Y[j]>=1} j>FILMresults.txt; # location
    printf "p= %u solve_result_num= %u \n",p,solve_result_num>FILMbranchMIP.txt;
    display solve_message>FILMbranchMIP.txt;
}
-----

```

#### Model File: FILMfast.mod

```

-----
param Q >=0;
param J >=0;
param p >=0;
param flow{1..Q};
param N{1..Q};
param path{q in 1..Q,j in 1..N[q]};
var Y{1..J} binary;# location decision variable
var X {1..Q} >=0,<=1;# flow interception decision variable
maximize Z: sum {q in 1..Q} flow[q] * X[q];
subject to Location {q in 1..Q} : sum{j in 1..N[q]} Y[path[q,j]]>=X[q];
subject to total_number: sum{j in 1..J} Y[j] = p;
-----

```

#### Data File: FILMfast.txt

```

-----
4 7 2
2 4 1 3 5 7
1 3 2 3 6
1 3 4 5 6
2 2 4 7
-----

```

\* First line:  $Q, J, p$ ; Line 4-5: flow,  $N$ , path; Data: the 7-node example.

**Appendix 2: Implementing GFIM in AMPL CPLEX****Running Script File: GFIMfast.run**

```

-----
# include GFIMfast.run;
option solver cplex;
model GFIMfast.mod;
read Q, J, p < GFIMfast.txt;# read Q, J, p
read {q in 1..Q} (N[q],{t in 1..N[q]}path[q,t]) < GFIMfast.txt; # read N, path
read {q in 1..Q}({j in 1..N[q]}G[q,j]) < GFIMfast.txt; # read G[q,j]
option cplex_options 'dual!';
option print_separator ","; # separator with "," in writing file
printf "p,?Optimal,Second,Z,Solution\n">GFIMresults.txt; # write title line
set SetP := 1 ..4; # set the range of p facilities
for {c in SetP} {
  let p := c;
  solve;
  # The follow commands are to print results in two files
  printf "%u,%u,% .1f,% .2f," p,solve_result_num,_solve_user_time,Z>GFIMresults.txt;
  print {j in 1..J:Y[j]>=1} j>GFIMresults.txt; # location
  printf "p= %u solve_result_num= %u \n",p,solve_result_num>GFIMbranchMIP.txt;
  display solve_message>GFIMbranchMIP.txt;
}
-----

```

**Model File: GFIMfast.mod**

```

-----
param Q>=0;
param J>=0;
param p>=0;
param N{1..Q};
param path{q in 1..Q,j in 1..N[q]};
param G{q in 1..Q,j in 1..N[q]};
var Y{1..J} binary;
var X{q in 1..Q,j in 1..N[q]}>=0,<=1;
maximize Z: sum {q in 1..Q} sum {j in 1..N[q]} G[q,j] *X[q,j];
subject to no_doubleflow {q in 1..Q}: sum {j in 1..N[q]} X[q,j] <= 1;
subject to only_at_facility {q in 1..Q,j in 1..N[q]}: Y[path[q,j]] >= X[q,j];
subject to total_facility: sum{j in 1..J} Y[j] = p;
-----

```

**Data file: GFIMfast.txt**

```

-----
4 7 2
4 1 3 5 7
3 2 3 6
3 4 5 6
2 4 7

12 8 2 0
3 2 0
3 2 0
4 0
-----

```

\* First line:  $Q, J, p$ ; lines 2-5:  $N$ , path; Lines 6-9: the matrix of  $G$ ; Data: the 7-node example.

**Appendix 3: Calculate the Shortest Paths in CPLEX****Appendix 3-A: ShortestPath55.mod**

# This is a 55-nodes example for calculating shortest paths by link matrix

```

param begin;
param end>begin;
set NODES:= begin .. end; # all intersections
param PathIn symbolic default 1;
param PathOut symbolic default 10;
param maxlink>=0; # the max link on a path
set LINKS within (NODES cross NODES);
param cost {LINKS} >= 0; # costs to travel roads
node Intersection{k in NODES};
arc Traff_In>=1,to Intersection[PathIn];
arc Traff_Out>=1,from Intersection[PathOut];
arc Shortpath{(i,j) in LINKS}>=0,
  from Intersection[i],to Intersection[j];
minimize Total_Cost: sum{(i,j) in LINKS} Shortpath[i,j]*cost[i,j];

```

```

data;
param begin :=1;
param end :=55;
param maxlink:=268; # the total links in the network
param: LINKS: cost :=

```

1	2	31623
1	5	20000
1	8	44721
1	13	22361
1	43	100000
1	44	41231
2	1	31623
2	3	44721
2	4	30000
2	8	31623
2	42	28284
3	2	44721
3	7	42426
3	8	31623
3	19	64031
3	30	50000
3	31	60828
3	34	53852
4	2	30000
4	5	30000
4	9	20000
4	42	22361

#Many links are omitted here.

```

55 54 101980;

```

**Appendix 3-B: ShortestPathBest.run**

```

# For calculating the shortest path by link and cost on each shortest path
# include ShortestPathBest.run;

model ShortestPathBest55.mod;
set CASES := 1..55;

param qnumber integer default 0;
set qlink:=1..maxlink; #total links in the network
printf "path from to cost :title \n" >shortpathcost.txt;# title

for {a in CASES}
{
  let PathIn:= a;
  for {b in CASES diff {a}}
  {
    let PathOut:= b;
    solve;
    #out put link shortestpath matrix: column is 1...links, the row is 1...600 path
    print {(i,j) in LINKS} Shortpath[i,j]>ShortestPathLinkMatrix.txt;
    let qnumber:=qnumber+1;
    printf "%4u %4s %4s %.0f\n",qnumber,a,b>Total_Cost>shortpathcost.txt;
  } # end for b
} # end for a
close all;

```



**Appendix 4: ShortestPath.cpp**

```

/*
Appendix ShortestPath.cpp
This program is to change the shortest paths by links from CPLEX into paths by nodes and for running
in CPLEX

```

```

3 input files:

```

1. shortpathcost.txt  
"path from to cost :title " at the end of the title must be ":title"
2. ShortestPathLinkMatrix.txt  
no title, the column are paths by unsorted links from 1...2212  
the rows are path 1...
3. linkid.txt  
no title linkid, link\_from, link\_to

```

4 output files:

```

1. outshortpathbynode.txt separate by comma
  2. outshortpathbynospace.txt separate by space
  3. outshortpathbylink.txt separate by space
  4. outlongestpathnode.txt
- ```

*/

```

```

#pragma warning(disable: 4786)
#include <iostream>
#include <algorithm>
#include <functional>
#include <vector>
#include <string>
#include <fstream>
#include <time.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
using namespace std;
void readwritefile()
{
int Q, TotalLink; //define Q as the total number of path
cout<<"Input files: shortpathcost.txt, ShortestPathLinkMatrix.txt and linkid.txt"<<endl;
cout<<"please enter the total number of paths and link, like 2970 268 "<<endl;
cin>>Q>>TotalLink;
int i,j; // i,j is for tempary
int k; //k is for each path q
int longestpath=0;
// to store the longest path node in the whole network
vector<string>title;
vector<string>q,source,sink,cost;
//define path number, path_from, path_to, and path cost
vector<string>linkid,linkf,linkt;
// to store linkid link from, link to from the linkid.txt
vector<string>pathlinkid,pathlinkf,pathlinkt;
// to store temp link from , link to in a path
vector<string>nodes,links;
//store each path nodes and links

```

```

string temps;
// to store temp string data

// start to read shortpathcost.txt
ifstream readfile("shortpathcost.txt",ios::in);
// there must be a tile,the end of the title must ":title"
if(!readfile)
{
cerr<<"open error! Check whether the file exist shortpathcost.txt!"<<endl;
exit (1);
}

readfile>>temps;// read by title by word
do
{
readfile>>temps;// read by title by word but not write it
}while(temps!=":title");

// start to read ShortestPathLinkMatrix.txt
ifstream readlink("ShortestPathLinkMatrix.txt ",ios::in);
// there are the link path by unsorted links from 1...2212
if(!readlink)
{
cerr<<"open error! Check whether the file exist ShortestPathLinkMatrix.txt !"<<endl;
exit (1);
}

//start to read linkid.txt
// there are the big path by unsorted links from 1...2212
ifstream readlinkid("linkid.txt",ios::in);
if(!readlinkid)
{
cerr<<"open error! Check whether the file exist linkid.txt!"<<endl;
exit (1);
}

for(i=0;i<TotalLink;i++)// linkid 1 is in 0
{
readlinkid>>temps;linkid.push_back(temps);
readlinkid>>temps;linkf.push_back(temps);
readlinkid>>temps;linkt.push_back(temps);
}

// start to write the title
ofstream writefile("outshortpathbynode.txt",ios::out);
writefile.precision(1);
writefile<<"q,source,sink,cost,nodemax,pathnode"<<endl;// write title

ofstream writefile2("outshortpathbynodespace.txt",ios::out);
writefile2<<"q source sink cost nodemax pathnode "<<endl;// write title
writefile2.precision(1);

ofstream writefile3("outshortpathbylink.txt",ios::out);

```

```

writefile3<<"q source sink cost linkmax pathlink "<<endl;// write title
writefile3.precision(1);

// start: each k read each path data,
// get the path when linkid =1 and then sort node, then write to file
cout<<"completed sorted path : "<<endl;// just to show the process
for(k=0;k<Q;k++)
{
//start to read each pathlink data
readfile>>temps; q.push_back(temps);// read and push path number
readfile>>temps; source.push_back(temps);//read and push from
readfile>>temps; sink.push_back(temps);//read and push to
readfile>>temps; cost.push_back(temps);//read and push cost
// is 0 because we delete each time.
writefile<<q[0]<<" "<<source[0]<<" "<<sink[0]<<" "<<cost[0]<<" ";
writefile2<<q[0]<<" "<<source[0]<<" "<<sink[0]<<" "<<cost[0]<<" ";
writefile3<<q[0]<<" "<<source[0]<<" "<<sink[0]<<" "<<cost[0]<<" ";

vector<string>::iterator Iter1,Iter2,Iter3;
Iter1=linkid.begin();
Iter2=linkf.begin();//point to link from begin
Iter3=linkt.begin();

int linksum=0;//define the number of links in each path
// read each pathlink get the link is 1 and push the node link_from link_to
for(i=1;i<=TotalLink;i++)
{
readlink>>temps;
// get each link is 1 and push the linkid and link from and link to
if(temps=="1")
{
//writefile<<i<<" ";
pathlinkid.push_back(*Iter1);
pathlinkf.push_back(*Iter2);
pathlinkt.push_back(*Iter3);
linksum++;
} //end if
Iter1++;Iter2++;Iter3++;
} // end for i

writefile<<linksum+1<<" "; //pathmaxnode
writefile2<<linksum+1<<" ";
writefile3<<linksum<<" ";
if(linksum>longestpath)longestpath=linksum;

//end read unsorted pathlink and change to unsorted path node

// begin to sort path node
vector<string>::iterator tempL,tempR;//define for vector
tempR=source.begin();
// tempR is the address point to source begin from 0 not 1
nodes.push_back(*tempR);//get the first node
vector<string>::iterator Iter4,Iter5,Iter6;

```

```

Iter4=pathlinkf.begin();//point to link from begin
Iter5=pathlinkt.begin();
Iter6=pathlinkid.begin();

do // until find the sink
{
tempL=Iter4;// begin to find
if (*tempL==*tempR)
{
*tempR=*Iter5;//give
nodes.push_back(*tempR);// get the sorted node
links.push_back(*Iter6);

Iter4=pathlinkf.begin();//point to path link from begin
Iter5=pathlinkt.begin();//point to path link to begin
Iter6=pathlinkid.begin();//point to path link id begin
linksum--; //find one
}
else
{ Iter4++;Iter5++;Iter6++;}; //not find then look for the next one
}while(linksum>0);

for (j=0;j<nodes.size();j++)// write the path node by comma
{
writefile<<nodes[j]<<" ";
}
writefile<<endl;

for (j=0;j<nodes.size();j++)// write the path node by space
{
writefile2<<nodes[j]<<" ";
}
writefile2<<endl;

for (j=0;j<links.size();j++)// write the path link by space
{
writefile3<<links[j]<<" ";
}
writefile3<<endl;

q.erase(q.begin(),q.end());// from 0
source.erase(source.begin(),source.end());
sink.erase(sink.begin(),sink.end());
cost.erase(cost.begin(),cost.end());
nodes.erase(nodes.begin(),nodes.end());
links.erase(links.begin(),links.end());
pathlinkid.erase(pathlinkid.begin(),pathlinkid.end());
pathlinkf.erase(pathlinkf.begin(),pathlinkf.end());
pathlinkt.erase(pathlinkt.begin(),pathlinkt.end());
//end sort path nodes

cout<<k<<" "; // completed sorted path

```

```
// end for k
ofstream writefile4("outlongestpathnode.txt",ios::out);
writefile4<<"the longest path has nodes: "<<longestpath+1<<endl;
} //end readwritefiles

void main()
{
  readwritefile();//read inpathlinks.txt
  cout<<"congrats, all done! results are in the 4 files!"<<endl;
} //end main
```

**Appendix 5: CalculateInterceptedFlows.cpp**

```

/*
Appendix CalculateInterceptedFlows.cpp
This program is to calculate flows at each node or link.
1 Input file
PathFlows.txt--No title, the first line is Q J or Q TotalLinkID, 2970 55 or 2970 268
From the second line: flow pathmax NodeOrLinkOfEachPath
Note the First column is flow
1 output file
OutFlow.txt--total flow of each node
*/

#pragma warning(disable: 4786)
#include <iostream>
#include <algorithm>
#include <functional>
#include <vector>
#include <string>
#include <fstream>
#include <time.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
using namespace std;
void main()
{
int i,j;
int Q, Totalid;
ifstream readfile("PathFlows.txt",ios::in);// there must be no title
if(!readfile)
{
cerr<<"open error! Check whether the file exist PathFlows.txt! "<<endl;
exit (1);
}

readfile>>Q>>Totalid;
double IDFlow[9999];
for(i=0;i<Totalid+1;i++)
{
IDFlow[i]=0;
}

double flow;
int IDMax,nodeid;
for(i=0;i<Q;i++) // in read each path data, and write the path by node to file
{
readfile>>flow>>IDMax;
for(j=0;j<IDMax;j++)// begin to calcualte dist from source
{
readfile>>nodeid;
IDFlow[nodeid]=IDFlow[nodeid]+flow;
}
}

```

```
// end for i
ofstream writefile("OutFlow.txt",ios::out);
writefile.precision(18);
writefile<<"IDs, flows"<<endl;// write title
for(i=0;i<Totalid+1;i++)
{
writefile<<i<<" "<<IDFlow[i]<<endl;
}
cout<<"Congrats From Weiping!"<<endl;

} //end main
```

**Appendix 6: UnionLink.cpp**

```

/*
Appendix UnionLinks.cpp
This program is to union links from paths by links

1 Input file
InPathLink.txt--No title, the first line is Q TotalLink, 2970 55
From the second line: pathmax linkOfEachPath

1 output file
OutLink.txt -- linkid, choose
*/

#pragma warning(disable: 4786)
#include <iostream>
#include <algorithm>
#include <functional>
#include <vector>
#include <string>
#include <fstream>
#include <time.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>

using namespace std;
void main()
{
int i,j;
int Q, TotalLink;
ifstream readfile("InPathLink.txt",ios::in);// there must be no title
if(!readfile)
{
cerr<<"open error! Check whether the file exist InPathLink.txt! "<<endl;
exit (1);
}
cout<<"Union links..."<<endl;
readfile>>Q>>TotalLink;
long int linkflow[9999];
for(i=0;i<TotalLink+1;i++)
{
linkflow[i]=0;
}
int linkmax,linkid;
for(i=0;i<Q;i++) //read each path and write path by link to file
{
readfile>>linkmax;
for(j=0;j<linkmax;j++)//
{
readfile>>linkid;
linkflow[linkid]=1;
}
}

```



```
}// end for i

ofstream writefile("OutLink.txt",ios::out);
writefile<<"linkid, choose"<<endl;// write title
for(i=0;i<TotalLink+1;i++)
{
writefile<<i<<","<<linkflow[i]<<endl;
}
cout<<"Congrats From Weiping!"<<endl;
} //end main
```

**Appendix 7: ChangeSolutionID.cpp**

```

/*
Appendix ChangeSolutionID.cpp
This program is to soluitons (p=1..2.0) newID to another solution oldID
2 Input files
SolutionIDBefore.txt-- p locationBynewid
oldID.txt--
only one column, the first line is total link or node
The rest is the old id sorted by the new id from 1 to total id !
1 output file
SolutionIDAfter.txt--p locationByoldid
*/

#pragma warning(disable: 4786)
#include <iostream>
#include <algorithm>
#include <functional>
#include <vector>
#include <string>
#include <fstream>
#include <time.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>

using namespace std;

void main()
{

ifstream ReadID("oldID.txt",ios::in);// there must be no title
if(!ReadID)
{
cerr<<"open error! Check whether the file exist oldID.txt! "<<endl;
exit (1);
}

ifstream ReadPath("SolutionIDBefore.txt",ios::in);// there must be no title
if(!ReadPath)
{
cerr<<"open error! Check whether the file exist SolutionIDBefore.txt! "<<endl;
exit (1);
}
ofstream writefile("SolutionIDAfter.txt",ios::out);
int i,j, getone;
int Totalid;
int oldid[9999];
for(i=0;i<9999;i++)
{
oldid[i]=0;
}

ReadID>>Totalid;

```

```
for(i=1;i<=Totalid;i++)
{
  ReadID>>oldid[i];
}

int p;
writefile<<"p,location"<<endl;
while(ReadPath>>p)
{
  writefile<<p<<" ";
  for(j=1;j<=p;j++)
  {
    ReadPath>>getone;
    writefile<<oldid[getone]<<" ";
  }
  writefile<<endl;
} //end while
cout<<"Congrats From Weiping!"<<endl;
} //end main
```

**Appendix 8: CalculateSolutionFlows.cpp**

```

/*
Appendix CalculateSolutionFlows.cpp
This program is to calculate the total flows (395 zones) intercepted by known solutions (p=1..20)

2 Input files:
location.txt--p location
PathFlows.txt is the same as the data in CPLEX except the first line
The first line is Q total paths
From the second line: flow pathmax Node

1 output file
OuttotalFLoW.txt-- p, totalflow

// system("pause"); writefile.precision(1);
*/

#pragma warning(disable: 4786)
#include <iostream>
#include <algorithm>
#include <functional>
#include <vector>
#include <string>
#include <fstream>
#include <time.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
using namespace std;
void main()
{
cout<<" Calculating total FILM flow at p=1..20 when facilities are known"<<endl;
int i,j,k;

//start to read location file to a matrix
int p;
int location[21][21];
for(i=0;i<21;i++)
{
for (j=0;j<21;j++){location[i][j]=0;}
}
ifstream readlocation("location.txt",ios::in);
if(!readlocation)
{
cerr<<"open error! Check whether the file exist location.txt! "<<endl;
exit (1);
}
readlocation>>p;
do{
for(j=0;j<p;j++){ readlocation>>location[p][j];}
}while(readlocation>>p);
// End read location file

```

```

//start to read and calculate path flow
int Q;
double flow;
double totalflow;
int flowflag;
int pathmax,nodeid;
int nowp;

ofstream writeflow("Outtotalflow.txt",ios::out);
writeflow.precision(18);
writeflow<<"p,totalflow"<<endl;

for(nowp=1;nowp<=20;nowp++)
{
totalflow=0;
ifstream readfile("PathFlows.txt",ios::in);
if(!readfile)
{
cerr<<"open error! Check whether the file exist PathFlows.txt! "<<endl;
exit (1);
}
readfile>>Q;

for(i=0;i<Q;i++)
{
flowflag=0;
readfile>>flow>>pathmax;
for(j=0;j<pathmax;j++)// begin to calcualte dist from source
{
readfile>>nodeid;
if(flowflag==0)
{
for(k=0;k<nowp;k++) {if(location[nowp][k]==nodeid) flowflag=1;}
} //end if flowflag=0
} // end for j=0
if (flowflag==1) totalflow=totalflow+flow;
} // end for i=0

writeflow<<nowp<<","<<totalflow<<endl;
cout<<"Done p="<<nowp<<endl;
} //end for nowp
} //end main

```

**Appendix 9: 1-opt interchange heuristic**

```

c This is M. John Hodgson's 1-opt interchange heuristic
C Last change: Zeng 23 February 2007 4:23 pm
c file Interchange.FOR Solves T&B Algorithm
  open(25,file='Edmonton_greedy.txt')
    open(26,file='Edmonton_T&B.txt')
    open(27,file='problemdata.txt')
    open(28,file='solutions.txt')
  call Solvers
  STOP
  END
    subroutine Dataread(nq,n,flow,kovcount,patnodes,nocov,kover)
      integer Patnodes(160000,085),q,n,nq,kovcount(160000),qq
      integer nocov(9999),kover(2250,35000)
      real*8 flow(160000)
      read(27,*)n,nq
      WRITE(*,*)n,nq
      do qq=1,nq
        read(27,*)q,flow(q),kovcount(q),(patnodes(q,j),j=1,kovcount(q))
        end do
100  format(i7.6,f15.0,i5.4,100i6.5)
*   create Kover matrix
      maxkovr=0
      maxcovr2=0
      do q=1,nq
        do k=1,kovcount(q)
          j=patnodes(q,k)
          if (j.gt.maxkovr)maxkovr=j
          nocov(j)=nocov(j)+1
          kover(j,nocov(j))=q
          if(nocov(j).gt.maxcovr2)maxcovr2=nocov(j)
        end do
      end do
      write(*,*)maxkovr,maxcovr2
    return
    end
  SUBROUTINE Solvers
c calculates ASYMMETRIC shortest path distances
c ie: both directions entered as data
  integer*4 node (2250)
  real*8 ROW(2250),sumlast,summax,sum,fl0,zbest,zopt
  INTEGER*4 NUM(2250)
  real*8 x(2250),y(2250),d
  real*8 pop(2250)
  real*8 flow(160000)
  integer*4 iden(160000,2),q
  integer Patnodes(160000,085),kovcount(160000)
c for T&B
  integer*4 try(2250),randopt,best(2250)
c for short T&B
  integer*4 kover(2250,35000),nocov(9999),set(2250)
  call Dataread(nq,n,flow,kovcount,patnodes,nocov,kover)

```

```

    Call GREEDY(flow,n,nq,nocov,kover)
    CALL TBSETUP(n,nofac,nq,flow,zbest,nocov,kover,set)
    return
  end
  SUBROUTINE TBSETUP(n,nofac,nq,flow,zbest,nocov,kover,set)
    integer*4 set(2250),best(2250),nocov(9999),kover(2250,35000),randopt
  INTEGER*4 try(2250)
    real*8 flow(160000),zopt,zbest
  c   set up T&B
  c   set is set of nodes in a solution
    WRITE(*,1002)
  1002 format(/ Range of p? ')
    read(*,*)nlow,nhigh
    if(nlow<2)nlow=2 !Don't do it for p=1
    WRITE(*,1001)
  1001 FORMAT(/ How many random starts? ')
    read(*,*)notry
    do nofac=nlow,nhigh
      call timer(iticks)
      zopt=0.0
      randopt=0
      do nrand=1,notry ! Random Starts
        WRITE(*,*)nofac,nrand

  C      INITIALIZE Starting Solution to ZERO
        DO I=1,N
          set(i)=0
          try(i)=0 ! Organizer for Repetition avoidance
        enddo
  C      Initialize Set - random with no repetitions
        KOUNT=0
        j=0
        nfac=nofac
          do while (kount<nfac)
            Call RANDOM_NUMBER(r)
            IRAND=R*1.0+(1.0-R)*N
            IF(try(IRAND).eq.0)then
              j=j+1
              set(j)=irand
              try(IRAND)=1
              KOUNT=KOUNT+1
            else
          endif
          enddo
          call tb(n,nofac,nq,flow,zbest,nocov,kover,set)
        call order(n,nofac,set)
        do maketry=1,n
          try(maketry)=0
        end do
        do maketry=1,nofac
          try(set(maketry))=1
        end do
        WRITE(28,105)nofac,zbest,(try(k),k=1,n)

```

```

105  format(i2.2,f15.0,1x,2250i1)
      if(zbest.gt.zopt)then
        zopt=zbest
        randopt=nrnd
        do ii=1,nofac  ! Use "best" to represent the solution
          best(ii)=set(ii)
        enddo
      else
      endif

      end do  ! random starts
      call timer(jticks)
      itime=jticks-iticks
      write(26,100)NOFAC,notry,itime,zopt,(Best(k),k=1,nofac)
      enddo  !nofac
100  format(i2.2,2i10,f15.0,1x,2000i5)
      RETURN
      END
      SUBROUTINE TB (n,nofac,nq,flow,zbest,nocov,kover,set)
      INTEGER*4 kounter(16000),nocov(9999)
      * ,kover(2250,35000),set(2250),inok(2250)
      real*8 FLOW(16000),sum,summax,zbest,z,sumin,SUMFIX
c     Effect interchange mechanism
      itstop=0
      iter=0
      do while (itstop.eq.0)
        z=0.0
c       create kounter
c       kounter(q) # of times a flow is covered in a particular solution
c       initialize to zero
c       inok -- tells what nodes are already in set so can't be inserted
        do j=1,n
          inok(j)=1
        end do
        do k=1,nofac
          inok(set(k))=0
        enddo
        do q=1,nq
          kounter(q)=0
        enddo
c       count them up
        do j=1,nofac
          jj=set(j)
          do k=1,nocov(jj)
            idq=kover(jj,k)
            kounter(idq)=kounter(idq)+1
            IF(kounter(idq).eq.1)z=z+flow(idq)
          enddo
        enddo
        SUM=z
        summax=z
        iter=iter+1
        SUMFIX=SUM

```



```

do kout=1,nofac
  SUM=SUMFIX
c   see impact on sum of removing each node
    jout=set(kout) !Node Removed
c   what flows does jout cover? Take out if only covered once
    nj=nocov(jout) !how many flows is it covering?
      do i=1,nj
        iqq=kover(jout,i) ! Rolling thru flows covered
        if (kounter(iqq).eq.1) then ! removal only affects sum if this is the only
coverage
          sum=sum-flow(iqq)
          kounter(iqq) = 0 ! Or else replacement can't function
        else
        endif
      enddo
      sumin=sum !sum before we start putting nodes in
c   systematically see impact of replacing it
      do jtry=1,n
        if(inok(jTRY).eq.1)then ! can't go to ones already in the solution
        jin=jtry ! Identify node to try inserting
        nj=nocov(jin) ! how many flows does it cover?
        sum=sumin !reset the sum for putting nodes in
          do j=1,nj
            jqj=kover(jin,j) ! Rolling thru flows covered
            if(kounter(jqj).eq.0)then !adding only effects non-covered
            sum=sum+flow(jqj)
            else
            endif
          enddo ! Have adjusted sum for this insertion

c   would this replacement improve total flows?

          if(sum.gt.summax)then
            summax=sum
            maxout=kout
            maxin=jin
          else
          endif
          else
          endif
          enddo

c   reset kounter if the node replaced covers it and it is now zero
          do k=1,nocov(jout)
            idq=kover(jout,k)
            if(kounter(idq).eq.0)kounter(idq)=1
          enddo

          enddo
          if(set(maxout).eq.maxin)itstop=1
          set(maxout)=maxin
          inok(jout)=1
          inok(maxin)=0
          z=summax
          enddo

```

```

ZBEST=SUMMAX
RETURN
END

SUBROUTINE GREEDY(flow,n,nq,nocov,kover)
c   for greedy
integer*4 insoln(2250),gry_sol(2250)
integer*4 nocov(9999),kover(2250,35000)
real*8 floadd(160000),zed(2250),sumlast,summax,sum,flow(160000)
do i=1,n
  gry_Sol(i)=0 ! Greedy solution
  insoln(i)=0
end do
c   set floadd -- flow matrix to be adjusted with facilities
do q=1,nq
  floadd(q)=flow(q)
enddo
c   facility loop
np=n
sumlast=0.0
do iter=1,min(50,np)
  summax=-1.0
  do j=1,n
    if(insoln(j).eq.0)then
      sum=0.0
      do k=1,nocov(j)
        idq=kover(j,k)
        sum=sum+floadd(idq)
      enddo
      if(sum.gt.summax)then
        summax=sum
        locmax=j
      else
        endif
      else
        endif
      enddo
    insoln(locmax)=iter
    sumlast=sumlast+summax
    zed(locmax)=sumlast
    gry_Sol(iter)=locmax
    do k=1,nocov(locmax)
      idq=kover(locmax,k)
      floadd(idq)=0.0
    enddo
  call order (n,iter,gry_Sol)
  write(25,100)iter,zed(locmax),(Gry_Sol(k),k=1,iter)
100 format(i2.2,f15.0,1x,2000i5)
enddo
RETURN
END
SUBROUTINE ORDER(N,NOFAC,IN)
INTEGER IN(2250),INTER(2250)

```

```
c      orders a solution vector
      do i=1,N
      INTER(I)=0
      ENDDO
      do i=1,nofac
      inter(in(i))=1
      enddo
      kount=0
      do i=1,n
      if(inter(i).eq.1)then
      kount=kount+1
      in(kount)=i
      else
      endif
      enddo
      return
      end
```

**Appendix 10: AggregatePaths.cpp**

```

/*
Appendix AggregatePaths.cpp
This program is to automatically aggregate the original (395 zones) flow paths
into paths with selected nodeID (e.g. 270 high flow nodes).

Two input files (not title in the first line):
NodeInPath.txt--source sink q flow pathmax path_by_node_or_link no title
NewIDFrom1.txt--
the first line is Q;
The rest is the new id sorted by the original node id from 1 to totalid!
Remember all the unselected node ID is 0.

Three output files:
OutPathNewID.txt--sourceNew, sinkNew, TripIDNew, PassingMax, pathByNode
OutFlowNewID.txt--sourceNew, sinkNew, TripIDNew, sumflow;
OutPath395ID.txt--source, sink, TripID395, TripIDNew, flow, PassingMax, pathByNode_NoSourceSink
The OutPath395ID.txt-- The path node ID has changed to NewNodeID but not combine paths
In this file we can find how many trips are interceded by the selected nodes.
*/

#pragma warning(disable: 4786)
#include <iostream>
#include <algorithm>
#include <functional>
#include <vector>
#include <string>
#include <fstream>
#include <time.h>
#include <math.h>
#include <stdio.h>
#include <stdlib.h>
using namespace std;
void main()
{
ifstream ReadID("NewIDFrom1.txt", ios::in); // there must be no title
if(!ReadID)
{
cerr<<"open error! Check whether the file exist NewIDFrom1.txt! "<<endl;
exit(1);
}

ifstream ReadPath("NodeInPath.txt", ios::in); // there must be no title
if(!ReadPath)
{
cerr<<"open error! Check whether the file exist NodeInPath.txt! "<<endl;
exit(1);
}

ofstream writefile("OutPath395ID.txt", ios::out);
ofstream writePathNewID("OutPathNewID.txt", ios::out);
ofstream writeFlowNewID("OutFlowNewID.txt", ios::out);

```

```

cout<<" Input files: NodeInPath.txt and NewIDFrom1.txt"<<endl;
cout<<" Output files: OutPathNewID.txt,OutFlowNewID.txt, and OutPath395ID.txt"<<endl;
int i,j,t, getone;
int Q, Totalid;
int newid[9999]; // the total ID (node or link) in the network must be <= 9999

for(i=0;i<9999;i++)
{
    newid[i]=0;
}

ReadID>>Totalid;

for(i=1;i<=Totalid;i++)
{
    ReadID>>newid[i];
}

ReadPath>>Q;

int source, sink, q,pathmax,PassingMax,qnew;
int NumS;// the nubmer of selected nodes for potential facility sites
NumS=270; //Change here
int sourceNew, sinkNew, Flag[270+1][270+1]; //Change here
double flow, SumFlow[270+1][270+1]; //Change here
for(i=0;i<=NumS;i++)
{
    for(j=1;j<=NumS;j++)
    {
        SumFlow[i][j]=0;
        Flag[i][j]=0;
    }
}

vector<int>PathNodes;
writefile<<"source395,sink395,TripID395,TripIDNew,flow,PassingMax,pathByNode"<<endl;
writefile.precision(0);
writePathNewID<<"sourceNew,sinkNew,TripID,PassingMax,pathByNode"<<endl;
writePathNewID.precision(0);
writeFlowNewID<<"sourceNew,sinkNew,TripIDNew,SumFlow"<<endl;
writeFlowNewID.precision(0);

sourceNew=0;
sinkNew=0;
qnew=0; // the order of the old trip395 where flow>0
for(i=1;i<=Q;i++) // in read each path data, and write the path to file
{
    ReadPath>>source>>sink>>q>>flow>>pathmax;
    PassingMax=0;
    for(j=1;j<=pathmax;j++)// for each path not read the source and sink
    {
        ReadPath>>getone; //getone reads one new node in a path
        if(newid[getone]>=1){

```

```

    PassingMax=PassingMax+1;
    PathNodes.push_back(newid[getone]); // get one node in selected nodes
  } // end for j=1
} // end for i=1

if (PassingMax>=1) // If passing node in selected nodes
{
  // begin for OutPath395ID.txt
  qnew=qnew+1;
  writefile<<source<<","<<sink<<","<<q<<","<<qnew<<","<<flow<<","<<PassingMax<<",";
  for(j=0;j<PassingMax;j++){ writefile<<PathNodes[j]<<","; }
  writefile<<endl;
  // end for OutPath395ID.txt

  //Begin OutPathNewID.txt
  sourceNew=PathNodes[0];
  sinkNew=PathNodes[PassingMax-1]; // minus 1
  Flag[sourceNew][sinkNew]=Flag[sourceNew][sinkNew]+1;
  SumFlow[sourceNew][sinkNew]=SumFlow[sourceNew][sinkNew]+flow;
  // calculate sum flow for the new path

  if(Flag[sourceNew][sinkNew]==1) // write pathmax and node for the first time
  {
    writePathNewID<<sourceNew<<","<<sinkNew<<",";
    writePathNewID <<NumS*(sourceNew-1)+sinkNew<<","<<PassingMax<<",";
    for(j=0;j<PassingMax;j++){ writePathNewID<<PathNodes[j]<<","; }
    writePathNewID<<endl;
  } //end if
} // end if PassingMax

PathNodes.erase(PathNodes.begin(),PathNodes.end()); // erase the changed path

} // end for i

t=0;
for(i=1;i<=NumS;i++)
{
  for(j=1;j<=NumS;j++)
  {
    if(SumFlow[i][j]>0)
    {
      t=t+1;
      writeFlowNewID<<i<<","<<j<<","<<t<<","<<SumFlow[i][j]<<","<<endl;
      // write sourceNew,sinkNew,TripIDNew,SumFlow;
    } //end if
  } // end for j
} //end for i
} //end main

```