A LOCATION MODEL FOR WEB SERVICES INTERMEDIARIES

By

YI SUN

A DISSERTATION PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

UNIVERSITY OF FLORIDA

2003

I would like to dedicate this work to my family, especially my wife Ziya and our daughter Emily. Their love has supported me throughout all the hesitations and frustrations of this dissertation.

ACKNOWLEDGMENTS

This dissertation could not have been completed without the support, guidance and patience of a multitude of people. One individual is significantly responsible for my success, Dr. Gary Koehler. His willingness to patiently share his expertise coupled with high expectations encouraged me to seek excellence. I would also like to express my appreciation to the other members of my committee, Drs. Hsing Kenneth Cheng, Asoo J. Vakharia, Haldun Aytug, and Joseph P. Geunes, for sharing their time and insights.

I also acknowledge my colleagues in the Ph.D. program for their unwavering moral support and friendship throughout my years in the program, especially Dr. Kutsal Dogan, Dr. Cheryl Aasheim, and Mr. Nihat Kasap.

I also thank my friends who have been patient, considerate and encouraging during my Ph.D program. Finally, I am forever indebted to my family, particularly my wife, Ziya Sun, for always believing in me.

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

LIST OF FIGURES

Abstract of Dissertation Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Doctor of Philosophy

A LOCATION MODEL FOR WEB SERVICES INTERMEDIARIES

By

Yi Sun

August 2003

Chair: Gary J. Koehler
Major Department: Decision and Information Sciences

In the recent past, Web services have entered the competition for distributed e-commerce platforms. By exploiting existing open standards such as XML and HTTP, Web services promise unprecedented levels of interoperation between programs on different frameworks. Questions remain on how to capitalize on this new technology.

This dissertation attempts to answer the questions from a service location perspective. We focus on the behaviors of Web services intermediaries that serve as common interfaces to their clients and obtain Web services from independent Web services providers on behalf of their clients. The distributed nature of Web services and the ever-increasing prominence of network latency in determining server performance justify the study of locating servers of Web services intermediaries on the Internet to optimize their performance and/or financial goals. We propose a mathematical integer programming model to help these intermediaries decide on the locations and usage rates of their servers.

As expected, the proposed model becomes computationally intractable as the number of participants increases. We therefore develop an efficient heuristic method named DAL to tackle the problem. For the problems tested, the DAL heuristics provide near optimal solutions in short computer times and with limited computer memory usage.

CHAPTER 1
INTRODUCTION

1.1 Background

In the recent past, Web services have entered the competition for distributed e-commerce platforms. They promise to deliver an unprecedented level of interoperation between programs written in different languages and running on different platforms by exploiting existing open standards, such as XML and HTTP. Because of modularization and open interfaces, Web services facilitate the development of highly customizable and adaptable applications to meet business demand. In addition, Web services offer a convenient service registration, search and discovery system. This system breeds a market of Web services intermediaries (WSIs) who, on behalf of their clients, search, assemble and customize various Web services at run time. It is this market on which we attempt to capitalize in this research.

Until recently, research on the performance of Internet services focused on increasing server processing speed, reducing transmission time, and shortening queuing delays. Network latency, which measures the time it takes for a signal to travel from one place to another on the network, was often ignored. Compared with other factors that influence service response time, especially on a slow network, network latency was often negligible. Nevertheless, because of recent changes in the networking environment, such as the prevalence of high-speed Internet communication, and because of the tremendous increase of computing power, studies (Johansson 2000; Jamin et al. 2001) suggest that network latency has begun to play a more important role in determining service response time. Minimizing service response time now requires decreasing the network latencies between servers and clients.

The network latency problem of Web services is even more prominent. Most Web services are offered on the Internet which, compared with a small and controlled environment such as an intranet, is famous for high latency. In addition, a WSI is involved to match his clients and Web services providers (WSPs). Therefore network latencies occur both between the WSI and his clients, and between the WSI and WSPs. Network latency can be treated as an operating cost incurred by the WSI. This research attempts to develop a location model that capture the intricacies to minimize a WSI's operating cost or maximize his profit. Furthermore, since the WSI can find and assemble Web services at run time, matchmakings are often done dynamically while minimizing latencies. This requires fast algorithms to solve the problem.

In this chapter, we provide an overview of our thesis, including a brief description of the model and its application environment, our motivation, and the research contribution. The research problem is presented in Section 1.2, the motivation for developing the particular model and the anticipated impact of this study are discussed in Section 1.3. The expected contributions and applications of this research are presented in Section 1.4. The organization of this dissertation is provided in Section 1.5.

1.2 Research Problem

A WSI serves as an interface to both his clients and WSPs. Since network latency is becoming an increasingly important factor that determines service performance, it is imperative for the WSI to find ideal locations that balance the expected loads between his clients and WSPs he may contact. The research problem is in designing a general model to describe the cost structures for a WSI with the understanding that network latency is treated as an operating cost. The WSI needs to determine where to locate his servers and how many servers to deploy. At the same time, the model asks the WSI to choose appropriate WSPs who offer Web services at different prices and with different performance and capacities. To capture the level of customization that Web services offer, the model allows the WSI to map each client request to a vector of the Web services needed to process the request.

This is a capacitated and fixed cost mixed integer mathematical programming problem. As expected, the proposed model becomes computationally intractable as the number of participants increases. We therefore address the question to ascertain if heuristics can be employed to solve the problem.

1.3 Motivation

The primary motivation for studying this problem is to acknowledge the need for models that solve WSI location problems. Web services have been touted as the "next big thing" by industry analysts. This technology promises to alleviate organizations' concerns for interoperability in areas such as Enterprise Application Integration (EAI) and Business-to-Business Integration (B2BI). The development managers seek to jump on board to cut development and integration cost.

There are many uncertainties around the deployment of Web services and the capitalization on this new technology. Questions include, for example, what is the appropriate pricing scheme for a WSI? How many servers should he set up to meet his objective? Where should he position his servers to be exposed to his clients and WSPs? Without definitive answers to these questions, businesses are hesitant to commit to Web services. In this paper, we attempt to answer some of these questions. The main purpose of this dissertation is to provide a framework that guides a WSI to set up his services.

Our model derives from the well-developed literature of facility location models. Nevertheless it differs from the latter in several aspects. First, as a three-tiered model, it studies interactions between clients, intermediaries and service providers, especially the relationship between servers of an intermediary. Second, the model studies the latency of delivering bundled information goods. The latency cost of such delivery remains the same regardless of the number of goods within a bundle. Lastly, the model requires that a client request, regardless of the number of Web services the request needs, goes to just one WSI server for the entire request, although the client may go to other WSI servers for additional requests.

1.4 Contribution

There are two major expected contributions from this research. First, we hope to develop a new and generalized model to describe WSI location problems. We expect the model to be applied with modifications to many business applications that take advantage of the capability and versatility that Web services offer. The other major result from this study is the development of a heuristic method to solve the problems. We hope the heuristics will provide optimal or near optimal solutions quickly. In addition, we expect the heuristics to be used for similar location problems.

1.5 Organization of Dissertation

Chapter 2 provides an overview of the basics of Web services and various attributes and contributions of this new technology. Chapter 3 presents a literature review of facility location problems in general and WSI location problems in particular. In addition, we review network latency and its measurement. Chapter 4 provides a discussion of the problem setting and details of the model. Chapter 5 presents a literature review of the solution techniques of facility location problems. Chapter 6 describes the heuristic method that we propose to solve WIS location problems. Details of the sources and the collections of the experiment data are given in Chapter 7. In Chapter 8 we outline an experimental design to test the efficacy of the heuristics and analyze the characteristics of the model. Chapter 9 gives a summary of our findings. Chapter 10 concludes the study and gives directions for future research.

CHAPTER 2
WEB SERVICES

2.1 Introduction

This chapter presents a general overview of Web services. The goal of this chapter is to review the basics of Web services and various features and contributions of this new technology. Section 2.2 provides definitions, standards, and protocols of Web services. Section 2.3 introduces the major players in Web services architecture and their interactions with one another. Section 2.4 discusses the advantages of Web services compared with previously distributed component technologies and browser-based online services. Section 2.5 looks at the business benefits and potentials of Web services.

2.2 Overview

2.2.1 Definition of Web Services

Web services describe a new category of applications. Although major players in defining and supporting Web services agree on the underlying standards and implementations of Web services, they have yet to concur on exactly what Web services refer to. It is important to look at the definitions from such players as IBM, Microsoft, and Sun.

IBM defines Web services as follows: “Web services are self-describing, self-contained, modular applications that can be mixed and matched with other Web services to create innovative products, processes, and value chains. Web services are Internet applications that fulfill a specific task or a set of tasks that work with many other web services in an interoperable manner to carry out their part of a complex work flow or a business transaction. In essence, they enable just-in-time application integration and these web applications can be dynamically changed by the creation of new web services" (IBM n.d.).

Microsoft also provides its definition for Web services: "A Web Service is a unit of application logic providing data and services to other applications. Applications access Web Services via ubiquitous Web protocols and data formats such as HTTP, XML, and SOAP, with no need to worry about how each Web Service is implemented. Web Services combine the best aspects of component-based development and the Web, and are a cornerstone of the Microsoft .NET programming model" (Microsoft n.d.).

Sun's definition of Web services is: "A coarse-grained function that is accessed over Internet protocols -- such as HTTP -- using XML to describe both the nature of the function itself and the data that flows to and from the function" (Sun Microsystems n.d. a).

Each of these definitions has its own emphasis. Some are more operational than others. Nevertheless, they all agree on the common denominators that support Web services. We define Web services as independent programmable application components that offer services to other applications through commonly defined interfaces and standard Web protocols across the Internet. The key in this definition is that Web services are modular and distributed component computing techniques that use open standards and common interfaces. As we will see in the following sections, these characteristics enable "Web Services-based applications to be loosely coupled, component-oriented, cross-technology implementations" (Kreger 2001) and therefore bring adopters various benefits and potentials.

2.2.2 Support and Framework

The World Wide Web Consortium (W3C) is responsible for the specifications of Web services. Microsoft, IBM, and Ariba submitted initial Web Services specifications to W3C in 2000. Currently, more than 100 companies work jointly to support and promote the standard (World Wide Web Consortium n.d. b).

It is fortunate for Web services to be widely endorsed by all the major players. With the market share and capacities of these sponsors, Web services are guaranteed to be anything but a fad. Previously distributed component technologies were doomed partially because of a lack of universal support. By now, several vendors have already rolled out Web services frameworks. These frameworks support Web services standards and protocols and provide comprehensive assistance to produce, deploy, scale, secure, and maintain Web services.

Microsoft's framework for Web services is its newly developed .Net platform. Although .Net is not all about Web services, Web services are one of its most important aspects. .Net provides developers on a previous Microsoft platform a familiar environment to deploy Web services. In addition, .Net supports writing Web services in multiple languages, such as C++, C#, J# and Visual Basic.Net.

IBM offers the WebSphere Studio Application Developer product as an environment for creation and deployment of Web services. It is J2EE[1]-compliant and takes advantage of the functionality in VisualAge for Java. Its WebSphere SDK[2] for Web

[1] Jave 2 Enterprise Edition

[2] Software Development Kit

Services provides a complete platform for designing, developing, and deploying Web services.

Sun advocates Open Net Environment (Sun ONE) to compete with Microsoft .Net. Sun ONE is a “standards-based software vision, architecture, platform, and expertise for building and deploying Services on Demand” (Sun Microsystems n.d. b). The Sun ONE platform fully supports Web services standards with some of its key products, such as the Sun ONE Application Server, Sun ONE Directory Server, and Java API[3] for XML[4]-based RPC.

Several other vendors have their own frameworks for Web services. For example, Oracle has Oracle9iAS Web Services as a complete infrastructure for developing, deploying, and managing Web Services. The Hewlett-Packard’s Web Services Platform is a standard-based Web services platform with a J2EE application server as a foundation. The BEA's WebLogic Workshop provides an environment for building Java-, XML-, and Web Services-based applications.

All these frameworks conform to the standards and protocols of Web services, though they differ in specific implementations. For example, some are biased toward using certain programming languages. They may also differ in run time handling, service discovery, terminology preference, and so forth. However, for Web services to be successful, services developed in either one of these platforms should interoperate with ones from other platforms without prior knowledge and arrangement. For example, Web services, written with IBM’s Java-based WebSphere and Web services developed within

[3] Application Programming Interface

[4] Extensible Markup Language

Microsoft's .NET, should be the same in terms of interface and accessibility. In fact, at the XML Web Services One conference in Boston in August 2002, IBM and Microsoft demonstrated that Web services developed in both platforms could work with each other (Farrell 2002).

2.2.3 Standard and Protocol

The key to the platform-independence of Web services architecture is the adoption of open standards and protocols. These standards and protocols allow distributed Web services applications to be loosely coupled on different platforms. While maintaining interoperability, they can be developed in parallel and run in different operating systems and on different hardware devices.

Among the protocols used for Web services, Extensible Markup Language (XML) occupies a special position as the underlying technology. It serves as a data representation layer for most Web services protocols. It is a "universal format for structured documents and data on the Web" (World Wide Web Consortium n.d. c). The XML is definition-driven and the contents it carries can be easily manipulated programmatically. Being syntax, its data structure is self-describing and needs no proprietary keys. Therefore, even applications of different systems written in different languages are able to communicate with each other using XML. Web services powered by XML do not need special prearranged networking, operating systems, or platform bindings.

Simple Object Access Protocol (SOAP) is an XML-based protocol used to transfer structured and typed information over the Internet. It defines rules to represent the transferred data and specifies formats for remote calling and invocation of Web services. Just like a message written in XML, a SOAP message is completely platform independent. It has been estimated that SOAP is implemented in more than 60 languages on more than 20 platforms (Clements 2001). In addition, SOAP works with various transport protocols including HTTP and can be used wherever the Internet exists. Since it is XML-based and therefore text-based, a message embedded in SOAP can go through firewalls with ease.

Web Services Description Language (WSDL) is "an XML format for describing network services as a set of end points operating on messages containing either document-oriented or procedure-oriented information" (World Wide Web Consortium n.d. a). The WSDL exposes the interface to Web services by describing objects and their methods. It presents all publicly available functions of the services, the domain of their parameters, transport protocols to be used for binding purpose, and the location of the service. Since all the protocols and standards are commonly available, WSDL allows virtually universal access. In addition, the interface that WSDL defines encapsulates implementation details of Web services and makes their usage transparent to users.

Universal Description, Discovery, and Integration (UDDI) is a Web service itself, and its main function is to maintain a directory of various Web services offered in the market (UDDI.org 2001). It offers an open architecture to describe, discover, and integrate Web services. As with using the business yellow pages, Web services providers put their Web services descriptions and links in a UDDI repository which handles chores, such as Web services register, advertising, and search. It helps business to expose their Web services and increase their visibility.

2.3 Players and Implementations

There are three major players in Web services architecture: service providers, service intermediaries, and service requesters. Service providers offer Web services and maintain a registry to make their services exposed and accessible. They may offer generic information, such as stock quotes, or exclusive information, such as stock price predictions. Service requestors are those who use Web services. They may invoke those services themselves or outsource by allowing service intermediaries to invoke various Web services on their behalf.

Service intermediaries act as matchmakers between service providers and service requestors. There are conceivable variations of the intermediaries' functionalities. Some intermediaries may be interested only in registering and categorizing various services from service providers. Some prefer to add synergetic values into their service. For example, an intermediary may assemble remote Web services and serve their clients with the aggregated services.

IBM published its Web services architecture that captures the infrastructure in which the service provider, service requestor, and service registry interact with one another to publish, find, and bind the Web services. Figure 2-1 (Gottschalk et al. 2002) shows that Web services providers develop Web services and publish their interface and location in a UDDI repository. Service requestors find suitable Web services in this repository and obtain the WSDL files to access these Web services at run time.

2.4 Comparison with Earlier Technologies

Transactions over the Internet currently rely mainly on Web browsers and distributed component technologies. Using a Web browser to access service has become popular in recent years. However, it suffers from its reliance on human initiations and interactions. Also, it requires the end-user device to be HTML compatible. Web services solve this problem by adopting XML as the data transport that allows its contents to be programmatically manipulated. It also enables any device that uses HTTP or similar Web services transportation protocols to access the services.

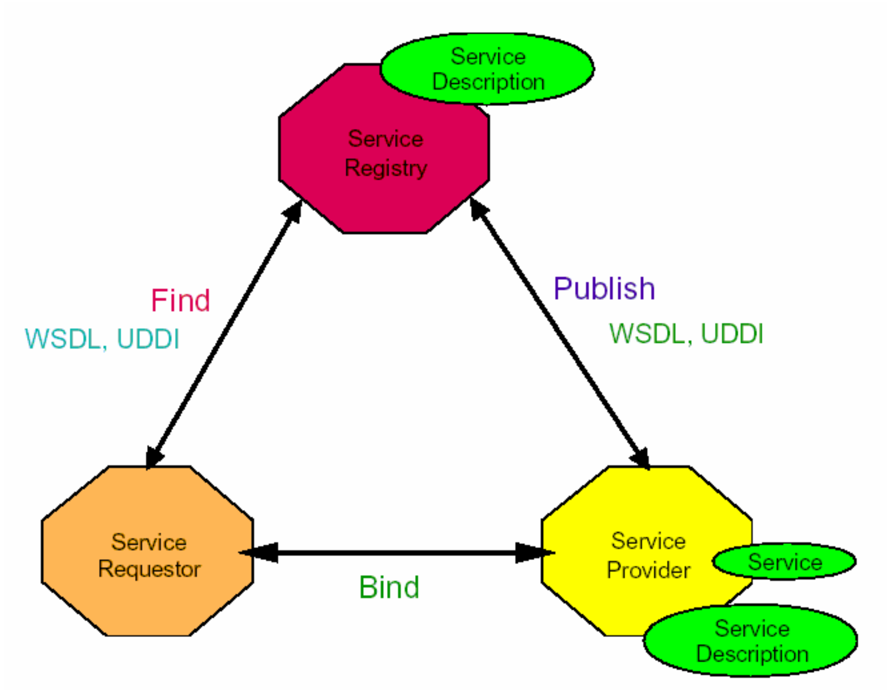

Figure 2-1. Web Services Roles and Operations.

Organizations have always faced the problem of reusing software code to shorten a development cycle. This problem becomes acute in distributed computing where programmers have less control over implementation of other distributed services. Distributed component technologies, such as Microsoft's Distributed Component Object Model (DCOM and COM+) scheme, the Common Object Request Broker Architecture (CORBA), and Sun's Remote Method Invocation RMI, Servlets and Enterprise Java Beans (EJB), address this problem by enabling remote invocation of methods.

However, these distributed component technologies still leave much to be desired. First, they are unable to interoperate over heterogamous systems, frameworks, and languages since their remote objects often have to be accessed via object-model-specific protocols. For example, DCOM and CORBA use proprietary standards and require complicated run time installed on all parties involved. Second, some technologies are language dependent. For example, Sun's Enterprise Java Bean is restricted to JAVA.

Web services become a natural evolution of these distributed component technologies. On the one hand, Web services enjoy many benefits of their predecessors, such as being closer topologically and/or geographically to end users and service functions remaining transparent to service requestors. On the other hand, Web services transcend them with interoperability and platform-independence. By adopting ubiquitous transportation protocols and data formats, Web services allow their applications to be accessed at run time without prior knowledge and special programming or device requirements. As long as all participants conform to Web services standards, their programs are guaranteed to work with each other. Furthermore, different Web services can be written in different languages on different platforms to put to good use their respective advantages. Overall, Web services make the framework open and highly interoperable (Lau and Ryman 2002).

In addition, Web services win over previous distributed component technologies by being more accessible to users on the Internet. This is because the Web services messages transferred between clients and servers are text-based and, as with HTML pages, will not be filtered by firewalls.

2.5 Business Benefits and Potentials

Web services not only are a breakthrough in technology, but have potential to transform the way that business is conducted on the Internet. For end users, Web services are more accessible over the Internet than previous technologies. In addition, Web services may lower entry cost for end users offering differential pricing through extensive service customization that accommodates businesses with different budgets. More importantly for our study, Web services bring new opportunities and challenges to Web services providers and service intermediaries in particular.

First, Web services providers enjoy low entry cost. Since Web services employ open standards, there are no special licensing fees. In addition, Web services do not lock developers into any special platform. They promote portability across different platforms and therefore reduce development cost. Also, Web services cut cost by encouraging software reuse and collaboration. Further, Web services harness the power of the Internet that is far-reaching and affordable compared with private or leased lines. Web services can also expose services offered by legacy systems at a low cost by wrapping them up in Web services standards. Because of the low entry cost, we can expect an increase in the number of Web services providers and intensified competitions between themselves.

Second, Web services offer an unparalleled level of customization. To maximize profit, businesses crave charging customizable services with differentiated prices. In the past, customizations are achieved at high cost. For example, traditional applications are often hardwired together and are expensive to change once the software is written. Even an object-oriented component application still has to deal with proprietary interfaces and incompatibilities across languages and platforms. Web services are more agile in working with one another to form solutions to various business problems. Service providers can easily form service packages by combining various Web services available on the Internet to meet individual needs in price and functionalities.

Third, Web services are also quick to adapt themselves to dynamic business environments. Business environments are often in flux and software life spans are short (Burbeck 2000), which is even more true for e-business. In the past, it is costly and timeconsuming to develop new software to cater to every new business need. In contrast, Web services can be assembled and disassembled relatively quickly at run time. Therefore, they can speed up development at lower cost (Ramakrishnan 2001). This enables a business to adapt to new environment quickly, at low cost, and gives him first-mover advantages.

Fourth, Web services breed a new market for Web services intermediaries. Integrations of Web services do not need collaboration in advance since their descriptions and interfaces use an open standard. In addition, automatic search and binding will give service intermediaries tools to assemble necessary Web services according to their clients' needs at run time. Furthermore, intermediaries can add synergistic values to the supply chains by exercising judgment and relying on experience to recommend suitable combinations of Web services to their clients. Admittedly, end users can contact service providers directly for service. However, if they decide to outsource, they can go through service intermediaries who categorize and dynamically bind appropriate Web services for them.

2.6 Summary

In this chapter, we presented a synopsis of basic concepts and functionalities of Web services and the status of their current development. By taking advantage of open standards and common protocols, Web services attempt to solve the problem of interoperability between distributed components. They allow client programs in any language, on any operating system, and on any device to programmatically invoke remote Web services methods. Various opportunities and challenges are therefore presented to business.

CHAPTER 3
FACILITY LOCATION MODELS AND WEB SERVICES

3.1 Introduction

This chapter presents a literature review of facility location models and their applications to the study of WSI location problems. This review will be grouped into two main topics. Section 2.2 provides a general review of network location models and discrete location models. Section 2.3 introduces the applications of these models to services on the Internet. This section will look at the prominence that network latency gained during the past several years in determining service response time. In particular, we will discuss why network latency makes facility location models appropriate for locating Web services servers. We will also discuss methods to measure network latency on the Internet.

3.2 Location Models

The literature on location problems focuses on modeling and solution strategies. There are basically three families of location models, namely, planar location models, network location models, and discrete location models (Tansel et al. 1983a, 1983b). They differ from one another in the level of complexity they capture, with planar models being the least expressive for practical problems and discrete location models the most expressive (Francis et al. 1992). Since network location models and discrete location models are more related to the study of locating Web services on the Internet, we will concentrate our effort on them with only a brief review of planar location models.

Planar location problems have been studied for a longer time than their network and discrete location counterparts. Planar models assume that travel happens on a plane and all points in the plane are potential locations. Distances are measured using the l_p metric. When $p = 1$, we have rectilinear distance and Euclidean distance when $p = 2$. Various models and solutions have been proposed.

3.2.1 Network Location Models

Network location models go one step beyond planar models by assuming that travel takes place on a network. This assumption makes models more realistic because many real-life problems, such as highway traffic control and Internet traffic control, can be mapped onto a network. Potential locations are restricted to nodes and/or edges of the network. The distance between two points is the shortest distance on the network and can be calculated by Dijkstra's algorithm (Dijkstra 1959).

Network location theory has been developing for the past 40 years. The seminal papers by Hakimi (1964, 1965) started interest in network location models. He introduced the *n*-median and *n*-center problems. *N*-Median models study how to select *n* number of vertices for facilities so as to minimize the sum of weighted distance between clients and their nearest facilities. The *N*-Center model studies how to select *n* number of vertices for facilities so as to minimize the maximum of weighted distances between clients and their nearest facilities. Another set of network models are called "covering problems," which study how to locate a minimum number of facilities with a maximum distance constraint between clients and their nearest facilities.

The complexities of these network problems depend on their network typologies. If the networks are general networks that allow cycles, then the problems are difficult to solve. The primary reason is that general network distance exhibits a piecewise linearity and concavity property (Hooker et al. 1991). Many problems on general networks are therefore NP-Hard, and various algorithms and heuristics to approximate optimal solutions are available (Garey and Johnson 1979; Kariv and Hakimi 1979a, 1979b; Kolen 1982; Megiddo et al. 1983; Church and Reveille 1994; Charikar and Guha 1999). In contrast, tree networks do not have cycles and enjoy the property of convexity. Their problems are tractable and admit straightforward solutions (Dearing et al. 1976). There are many well-solved special cases of tree-based networks, for example, 1-median problem on a tree network can be easily solved (Hua et al. 1962; Goldman 1971).

Another important property of network location models is the vertex and intersection point (VIP) optimality property. Hakimi (1964) showed that for 1-median problem, there is an optimal solution at one vertex, and for 1-center problem, there is an optimal solution at a vertex or an "intersection point." For the *n*-median problem, a set of optimal locations will always coincide with the vertices (Tansel et al. 1983a). Also the *n*-center problem exhibits the VIP optimality property, which suggests that "there exists at least one absolute *n*-center for which each center is at a vertex or an intersection point" (Minieka 1970; Francis et al. 1992). The VIP optimality property reduces solutions space from an infinite set to a finite set. Algorithms often take advantage of this property to solve network location problems.

3.2.2 Discrete Location Models

Discrete location models often apply to facility location problems. These problems include a set of potential locations with fixed costs for setting up facilities and transportation costs based on network distance. There is also a set of customers with demands to be satisfied by these facilities. The objective is to minimize the operating cost, including facility fixed costs, and transportation variable costs, by determining the number and locations of the facilities, and the assignment of clients to these facilities, while satisfying clients demands (Balinski 1965).

There are different subcategories within facility location problems depending on different assumptions. First, if a model assumes that facilities have specified service capacity (Christofides and Beasley 1983; Van Roy 1986), we have a Capacitated Plant/Facility Location Problem (CPLP). Second, the number of facilities can be limited to a range, in which case the model is named as the generalized *p*-median model (Francis et al. 1992). Third, the model may impose single source constraints by requiring that each client to be served by just one facility (Klincewicz and Luss 1986; Sridharan 1993). Fourth, multiple commodities can be introduced (Elson 1972; Geoffrion and Graves 1974; Laundy 1994). There are many applications for these models, including distribution planning and telecommunication network design.

A typical CPLP can be modeled in the following way (Khumawala 1974). Consider a set of customers, $i \in I$, with known demand for each product, a_i is to be served by a set of potential plant locations, $k \in K$. Each plant has a capacity of D_k with a fixed cost f_k. The variable transportation cost from plant to client is $t_{k,i}$.

Let the decision variables involved be

$X_{k,i}$=amount of product shipped from plant k to customer i;

$$P_k = \begin{cases} 1 & \text{if a plant is open at } k, \\ 0 & \text{otherwise.} \end{cases}$$

Problem SP

$$Min \sum_{k,i} t_{k,i} X_{k,i} + \sum_{k} f_k P_k$$

subject to

$$\sum_{k} X_{k,i} \geq a_i \qquad \text{for all } i \qquad (1)$$

$$\sum_{i} X_{k,i} \leq D_k P_k \qquad \text{for all } k \qquad (2)$$

$$X_{k,i} \geq 0 \qquad \text{for all } k \text{ and } i \qquad (3)$$

$$P_k = \{0,1\} \qquad \text{for all } k \qquad (4)$$

The objective is to minimize the sum of all transportation costs and fixed costs of opened plants. Constraints (1) are customer-demand constraints that ensure the clients' service level will be met. Constraints (2) are plant-capacity constraints that ensure that no facilities' capacities will be exceeded.

As a natural extension of CPLP, a multi-level/multi-echelon location model is especially relevant to our study. This model studies the number and the locations of the intermediate warehouses, as well as those of manufacturing plants. It can be applied to study the behaviors of any distribution systems that take advantage of economies of scale by seeing up intermediaries. One such example is the distribution system that Wal-Mart operates to secure products from various product suppliers to its warehouses and then supply them to their retail stores. In addition, this model can be adapted for multi-commodity scenarios.

Although relevant studies date back to the work of Geoffrion and Graves (1974), only recently have multi-level and/or multi-commodity facility location problems started to receive considerable attention (Hindi and Basta 1994; Klose 1999). Geoffrion and Graves (1974) set up a regional distribution center between several plants and customer zones to facilitate the flow of several commodities produced at plants to customers. However, in this model, the origin of each product quantity delivered was known.

Recently, several studies proposed more general models with selection of plants, as well as warehouses, as decision variables (Hindi and Basta 1994; Pirkul and Jayaraman 1998; Klose 1999, 2000). A typical multi-level and/or multi-commodity facility location problem can be formulated as the following mixed integer program (Pirkul and Jayaraman 1998). Consider a number of products, $l \in L$ and a set of customers, $i \in I$, with known demand for each product, a_{il} is to be served by a set of potential plant locations, $k \in K$, through a set of potential warehouses $j \in J$. Each plant and warehouse has a capacity of D_k and W_j, respectively. Each plant and warehouse site incurs a fixed cost f_k and g_j, respectively. Each product is related to plant capacity and warehouse capacity by the quantity q_l and s_l, respectively. Variable transportation costs from plant to warehouse is T_{jkl} and the cost from warehouse to client is C_{ijl}.

Let the decision variables involved be

X_{ijl} =amount of product l shipped from warehouse j to customer i;

Y_{jkl} =amount of product l shipped from plant k to warehouse i;

$$Z_j = \begin{cases} 1 & \text{if a warehouse is open at } j, \\ 0 & \text{otherwise.} \end{cases}$$

$$P_k = \begin{cases} 1 & \text{if a plant is open at } k, \\ 0 & \text{otherwise.} \end{cases}$$

Problem CP

$$Min \sum_{ijl} C_{ijl} X_{ijl} + \sum_{jkl} T_{jkl} Y_{jkl} + \sum_{k} f_k P_k + \sum_{j} g_j Z_j$$

subject to

$$\sum_{j} X_{ijl} = a_{il} \qquad \text{for all } i \text{ and } l \qquad (1)$$

$$\sum_{il} s_l X_{ijl} \le Z_j W_j \qquad \text{for all } j \qquad (2)$$

$$\sum_{i} X_{ijl} \le \sum_{k} Y_{jkl} \qquad \text{for all } k \qquad (3)$$

$$\sum_{jl} q_l Y_{jkl} \le D_k P_k \qquad \text{for all } k \qquad (4)$$

$$P_k, Z_j = \{0,1\} \qquad \text{for all } j \text{ and } k \qquad (5)$$

$$X_{ijl},\ Y_{jkl} \qquad \text{for all } i, j, k \text{ and } l \qquad (6)$$

This formulation is slightly different from that of (Pirkul and Jayaraman 1998), but it does capture the essence of the problem. The objective is to minimize the sum of all transportation costs and fixed costs of opened plants and warehouses. Constraints (1) are customer-demand constraints. Constraints (2) and (4) are plant-capacity constraints and warehouse-capacity constraints, respectively. Constraints (3) ensure the balance between clients' demand and plants' supply. There can be variations to this model. For example, the number of plants and warehouses can be limited to a range (Pirkul and Jayaraman 1998). Rather than just one level of warehouse, multiple levels can be set up (Klincewicz 1985). In addition, we can ask a client to be served by just one warehouse (Tragantalerngsaka et al. 1997).

Location problems are generally difficult to solve and therefore many researchers devote time to design solution methods. We will present a review of some solution methods in Chapter 5.

3.3 Online Services

Among other criteria, the performance of a distributed service on a network or on the Internet is often judged by its response time. For some applications, such as those used in the financial market, service response time is of paramount concern. Network latency refers to the amount of time that a packet of data takes to move from one location to another on the network. Only recently has network latency begun to constitute an increasingly larger percentage of service response time. For problems of locating Web services, network latency is closely related to the proximity between Web services and their clients. The placement of Web services on a network therefore becomes critical in improving service response time.

3.3.1 Network Latency and Response Time

Johansson (2000) suggested that service response time on the Internet included local processing time and network response time. Network response time consists of transmit time (the time taken to load data on to the network), queuing delay, and network latency. The former two are dependent on network capacity or bandwidth. Network latency is the inverse of signal speed, bounded by the speed of light, and it is ideally a function of network distance. In practice, it is also affected by transmission mediums (whether optical fiber, radio, or some other mediums), transportation protocols, and the number and types of routers used between two locations.

Until recently, studies of service response time on the network (Ram and Narasimhan 1994; Rho and March 1995) focused on increasing local processing speed and reducing transmission time and queuing delays. The factors under scrutiny were the server's computing power, data retrieval speed, network bandwidth, and so forth. In these studies, network latency was justifiably ignored because, compared with other factors, especially in a slow network, network latency accounted only for a very small percentage of service response time. Originally one of the selling points of service on the Internet was that the network latency was so insignificant as compared with other factors, that the locations of the servers were transparent to users. For example, when accessing a service on the Internet in the United States, a local user expected to experience similar service response time as a user from another continent.

However, because of recent changes in the networking environment, such as the prevalence of high-speed Internet communication, availability of super power servers, and improved database management, studies suggest that improving network latency plays an important role in reducing service response. Johansson's network response time model (2000) estimates that on a high-speed network, such as the T1 network, ignoring network latency will underestimate response time by more than 80 percent, though for a slow network, such as a 56 Kbps network, there was only a 20 percent difference.

Factors affecting local processing speed, transmit time, and queuing delays are being improved at an ever-decreasing cost. For example, it is much cheaper now to deploy powerful servers than it was just two years ago. In addition, fiber optic cables have already dominated the communication medium market. Furthermore, server clusters can be formed to solve the problem of undercapacited servers (Cardellini et al. 1999). In contrast, network latency is not easy to improve without moving servers nearer to their clients. Improvements on the transportation protocols, routers, and packet network overhead need orchestrated efforts by all equipment vendors, service providers, and users throughout the Internet community, and therefore are costly. One example is the delayed implementation of IP version 6. In conclusion, all factors except network latency are being greatly improved. This explains why network latency is increasingly becoming a major component in service response time, and the location choice of servers becomes a key in improving service performance.

Recent literature on efficiently redirecting clients' requests to the appropriate server (Basturk et al. 1997; Levine and Garcia-Luna-Aceves 1997; Shaikh et al. 2001) or on optimally placing servers (Crovella and Carter 1995; Guyton and Schwartz 1995; Carter and Crovella 1996; Fei et al. 1998; Li et al. 1999; Qiu et al. 2001) focuses on reducing network latency. Many of these studies of server placement on a network borrow analytical tools and algorithms from traditional facility location literature (Barford et al. 2001; Jamin et al. 2001).

Both general network models and tree models have been proposed to describe the interactions between players on the Internet. Some studies (Guyton and Schwartz 1995; Nussbaumer et al. 1995; Li et al. 1999; Krishnan et al. 2000; Cidon et al. 2001) use a tree-like hierarchical typology. Similar to traditional facility location problems, the main inspiration for the tree network is to have tractable solutions. Others argue that portraying the Internet with tree typology is not realistic, and the solutions thus obtained are not easily generalized (Kangasharju et al. 2001; Qiu et al. 2001). In their studies, they often have to find heuristics to solve their problems.

3.3.2 Network Latency and Web Services

Network latency is even more important in the study of Web services location problems. Web services are implemented in a loosely coupled network. An application may need the collaborations of many remotely located Web services and may generate many messages to discover, negotiate, and invoke these services. These messages greatly increase the overall network latency. In addition, network latency has a bigger impact on service response time when messages are short (Johansson 2000). Web services may often create short messages to discover and arrange services, and to pass parameters. This further increases the proportion of the network latency in the total service delay.

Besides, Web services are implemented on the Internet where network latency and its variance are usually higher than those of controlled networks. Many elements contribute to these disadvantages. For example, the TCP/IP protocol for unreliable networks, such as the Internet, needs to anticipate and correct transmission problems (Comer 2001). These actions create overhead that adds to the delays of transmission and therefore increase network latency. In addition, network latency has high variance because of nondeterministic delivery paths on the Internet.

The network latency problem becomes acute for certain types of Web services arrangements. For example, a service intermediary may provide services by accessing Web services remotely located in various places and offer its clients an integrated interface. In this case, there will be network latency not only between the intermediary and his clients but between the intermediary and various Web services providers. This problem is more apparent when each request from clients incurs multiple accesses to Web services providers. Therefore, placement of Web services servers is a crucial element to improve network latency and overall Web services response time.

3.3.3 Measuring Network Latency and Proximity

One way to improve the network latency of Web services is to place servers near clients. This is the underlying objective of commercial systems like stock brokerage applications. The measure of proximity therefore becomes a crucial subject in servers' selection and placement. This new issue and the attention it attracts result in several projects to serve the demand for proximity information. For example, SONAR provides a service which, given a list of IP addresses, orders the list by their proximity to the SONAR server (Moore et al. 1996). Francis et al. (1999, 2001) proposed IDMaps as an architecture to build a "distance map" of the Internet to measure the relative distances between points. This architecture constructed a virtual topology of the Internet by tracing paths on the Internet. Also, Network Weather Service (Wolski et al. 1999) measures end-to-end TCP/IP performance in terms of bandwidth and network latency. Ng and Zhang (2002) proposed a mechanism, called Global Network Positioning (GNP), to predict Internet network distance and measure transmission delay. They modeled the Internet as a geometric space and computed the absolute coordinates. They showed that GNP was more accurate and robust than IDMaps.

Many measures of proximity have been proposed, including return trip time (RTT), network or administrative system (AS) hops, geographic distance, domain name, HTTP request response time, HTTP request latency response time (with simple requests), and any combinations of these measures (Crovella and Carter 1995; Sayal et al. 1998; Qiu et al. 2001). There remains the question of which metrics are the best indication of the network latency.

The RTT is often not reliable since it may vary with the different routes a packet may take through the network. It is also subjected to the sporadic network traffic that results in high variance in the RTT. Nevertheless, some researchers (Obraczka and Silva 1999) concluded that RTT was a good metric.

Using the number of hops as a proximity measure is common. The value can be easily obtained from routing tables and it stays relatively stable (Sayal et al. 1998). However the validity of this method has been the focus of debate. Crovella and Carter (1995) suggested the number of hops in general was not a good indicator of network latency since it was not strongly correlated with RTT. In contrast, McManus (1999) later found that RTT was correlated with the AS hop count. Obraczka and Silva (1999) showed that, compared with the number of AS hops, the number of network hops was more meaningful as network latency metric. Shaikh et al. (2001) also found that network hops could be a good metric for determining client-server proximity. They used the standard trace route tool to learn the network path between clients and servers.

Geographical distance as a measurement has also been studied. Gwertzman and Seltzer (1995) showed that geographical information could be used to suggests the typology of network. Although not as good as network hops as an indicator of network latency, geographical distance is correlated with network latency, especially when hosts are on the same network backbone.

Using HTTP request response time as a criterion, Sayal et al. (1998) argued that network hops and RTT were poor indicators of proximity since they did not closely correlate with HTTP request response time. However, HTTP request response time itself is not a good measure of proximity since it also depends on the server's performance.

3.4 Summary

In this chapter, we presented a synopsis of basic models of facility location problems. There has been a recent surge of interest in applying these models to describe services on the Internet. This is because of the ever-increasing prominence of network latency in determining service response time, which is a result of server and network environment changes. As expected, applications on the Internet need to deal with the problems that traditional facility location models face. The right choice of underlying network typologies and algorithms often decides the successfulness of the applications.

CHAPTER 4
MODLE FORMULATION

4.1 Introduction

In this chapter we define a server location problem facing a WSI who provides services to his clients by accessing Web services offered by WSPs and adding value in some form. This WSI location problem departs from traditional facility location problems mainly in that the products provided are bundled information goods that have special cost structure and service delivery requirement. We give a general location model describing the cost structures for this WSI.

4.2 Assumption

As in any modeling task, a WSI location model is only as useful as the degree to which it describes the problem. Among all the factors that determine the success of such a model, reasonable assumptions are especially important. We assume there are three participants in our model: a WSI, his clients, and WSPs. The WSI may set up multiple identical servers at different locations to reduce service delays. Various WSPs offer (possibly different) Web services. These WSPs may be located remotely from WSI servers. Upon receiving service requests from his clients, the WSI will contacts one or more WSPs and possibly other WSI servers to process these requests. The WSI will then return the requested information to his clients. Part of the motivation for the clients to go through the WSI instead of contacting WSPs directly is to take advantage of the unified interfaces and services offered by the WSI. We presume that the WSI offers other value-added activities such as additional processing of information, collation, statistical quality control, etc. Our model studies optimal ways to locate the WSI servers, to assign clients to these servers, and allocate the usage of WSPs to minimize total operating costs, including fixed costs and variable communication costs, for the WSI subject to various quality of service and other constraints.

4.3 Notation

The following table summarizes the parameters and variables used in developing our model. We state the model as a mixed integer programming formulation.

Table 4-1. Summary of Notation.

General	
$G = N(V, E)$	Undirected graph G with node (vertex) set V and undirected arc (edge) set E.
$v = \lvert V \rvert$	Positive integer indicating the number of vertices and the index of their locations in the network G.
$e \in E$	An undirected arc in the network G
$l_{i,j}$	Nonnegative variable communication cost from vertex v_i to v_j (see text for more details).
$u_{i,j}$	Nonnegative integer decision variable representing the number of transfers of Web services vector from vertex i to j. This may be from a client to a WSI server or from a WSI server to a WSP.
z	Number of different types of Web services available.
Web Services Intermediaries	
$v_i \in \{0,1\}$	Binary variable that takes value 1 if vertex i is used as a WSI server location and 0 otherwise.
$\alpha_{i,j}$	Nonnegative integer decision vector valued variable representing the slack of Web services usage given from WSI server i to server j, a vector of z dimensions. Each dimension represents a unique Web service.
f_i	Nonnegative fixed cost per period to establish a WSI server at location i.
s_i	WSI server i's capacity per period. It refers to the number of requests a WSI server can process per period.
o	A z-dimensional sum vector with all elements equal to 1.
Clients	
$c \in C \subseteq V$	Index of client locations in the network G.
r_c	Nonnegative integer representing the service level requirement for client i per period. It refers to the number of requests a client submits per period.

Table 4-1. Continued.

h_c	A *z*-dimensional, nonnegative integer vector of Web services that client c uses per service. Each dimension represents a unique Web service.
Web Services Providers	
$w \in W \subseteq V$	Index of WSP locations in the network G.
q_w	A *z*-dimensional, nonnegative integer vector of Web services that WSP w provides. Each dimension represents a unique Web service.
t_w	WSP w's capacity per period. It refers to the number of requests a WSP can process per period.
$x_w \in \{0,1\}$	Binary variable that takes value 1 if a WSI uses service provided by WSP w, and 0 otherwise.
g_w	Nonnegative fixed fee that WSP w charges a WSI per period for accessing the WSP's service.

4.4 Model Setup

Consider a network $G = (V, E)$ with vertices $v \in V$, $v \equiv |V|$, and undirected arcs, $e \in E$. We use v to represent both the number of the vertices and the index of their locations letting context keep the particular meaning clear. There are n WSI's servers located at vertices $i \in V$ where n is a decision variable. These servers have the same interface, functionality and accessibility to all clients and WSPs. They differ from one another by their physical locations, fixed setup cost and service capacities.

Conceptually, the WSI's servers can be placed anywhere in the graph G, including on the arcs or at the vertices. Fortunately, a vertex-optimality property (Francis et al. 1992) shows that some vertex location is optimal for many location problems especially when one is minimizing distances or latencies. The problem we study has similar properties and we thus restrict our attention to locations on vertices.

The planning horizon of this WSI location problem consists of several consecutive periods. The length of each period is unspecified in our model but could be a week, a month, a year, etc. At the beginning of the planning horizon, the WSI makes initial decisions to minimize costs for the entire upcoming period. After the first period and at the beginning of each successive period, the WSI reviews and rebalances previous decisions to reflect changes that may have occurred during the previous period. These changes may be the changes in the customer base, changes in the number of and offerings by WSPs and the properties of WSI server locations. In the following model, we focus on the WSI's initial decision.

Different vertices may incur different fixed costs for the WSI. For example, locations with better infrastructure or better contract terms can charge higher fixed fees. Let f_i be the fixed fee to host a WSI server at location i. In practice, a location may also charge the WSI a variable cost based on the WSI's usage of the facility. This cost can be readily subsumed in the variable communication cost between WSI servers and their clients. In addition, we assume that each location posts a capacity limit on the number of requests a server can process per period. Let s_i be the capacity limit per period for WSI server i.

We assume a WSI knows the number of his clients, their locations $c \in C \subseteq V$, their requests, and their service requirement r_c**.** We assume that all requests from a client are the same. If a client has different requests, we divide this client into multiple sub-clients, each of which submit only the same requests and are located at the same physical location but the network is increased in size with a duplicate node. The WSI knows that there are z distinct Web services offered in the market by WSPs and knows how to use these Web services to process his client requests. He maps client c's request to a vector of Web services, denoted by a z-dimensional vector h_c. Any particular client request may not need all z Web services, in which case the unused Web services will have a zero value in h_c. We assume that a client request will be processed entirely by one WSI server, although the client can submit requests to multiple WSI servers at the same time.

Upon receiving requests from a client, the WSI will use the corresponding Web services vector for the client to obtain Web services from appropriate WSPs. A WSP publishes its location $w \in W \subseteq V$ in the graph G. It also publishes its services as Web services vector q_w with z dimensions. If a WSP offers different vectors, we break down this WSP into multiple sub-WSPs on a duplicate of the same vertex so each one offers just one vector. We assume that each request from a WSI server accesses either all the elements of a WSP's Web services vector or nothing. This assumption is justified by WSP's desire for bundling and/or for resource pooling. In addition, compared with proprietary technologies that are fine tuned for specific task, Web services often generate higher overhead per transaction. Therefore requesting multiple Web services at a time also helps the WSI to cut down communication cost by lowering the number of requests for WSP's services.

Like the WSI, a WSP is also subjected to a service capacity constraint limiting the number of requests it can process. Let t_w be the WSP w's capacity per period in serving requests. In addition, we allow WSP w to charge the WSI g_w as a fixed fee per period. In practice, the WSP may also charge a variable fee based on the usage of its service. This cost can be readily subsumed in the variable communication cost between WSI servers and WSPs.

Since a service request to a WSP activates its entire Web services vector, there are times when some Web services are not utilized by the original requesting WSI server but nevertheless available. We allow a WSI server to share its excess Web services obtained from WSP with other WSI servers. We use $\alpha_{j,i}$ as a decision vector to denote the slack of Web services that can be shared by WSI server j with WSI server i. In this model, we allow one Web service to be shared at a time. By enabling sharing, we trade-off the communication costs to access remote WSPs with the communication costs between WSI servers to obtain a, possibly, lower cost.

Let $u_{i,j}$ be a decision variable representing the frequency of communications between vertex i to j per period. It is a multiplier used with Web services vectors h_c and q_w to calculate the amount of Web services needed per period. For example, vector $h_c u_{ci}$ stands for the entire Web services needed by client c from WSI server i per period. Similarly, vector $q_w u_{i,w}$ stands for all the Web services that WSI server i obtains from WSP w per period.

Communication between vertices v_i to v_j incurs a variable communication cost $l_{i,j}$, which is a function of the network latency between a server and a client, and the servers' service time. There are different ways to interpret this cost in a business environment. For example, we can consider it as an opportunity cost expressed in the form of reimbursement from a WSI to his clients for the delays of service. For information goods, communication costs are less determined by the quantity of the information goods delivered per transaction than the network latencies they incur. This is because network bandwidth and congestions are constantly improving while network latency remains the same. In addition, although we decompose a service request into multiple Web services, the communication costs of the request may be independent of the number of Web services since WSI servers synthesize the answers they obtain from WSPs and forward the results to their clients. In the following, we treat $l_{i,j}$ as a linear costs based on the number of service requests.

4.5 Objective Function

In this model, we seek to determine the number and locations of WSI servers and the WSPs that the WSI contacts for services to minimize the overall operating costs for the WSI. We consider the variable communication costs between all participants. We also incorporate in the model the fixed costs to set up WSI servers and WSPs' fixed service fees for the WSI. There are four sets of decision variables: the locations of WSI servers v , the frequencies of communications between participants, u , the selections of WSPs as service providers, x , and the amount of Web services shared between WSI servers, α .

A WSI solves the following discrete optimization problem (4.1) for the first period of its planning horizon. The objective function is

$$\min_{v,u,x,\alpha} \sum_{c\in C}\sum_{i\in V} l_{c,i}u_{c,i} + \sum_{i\in V}\sum_{w\in W} l_{i,w}u_{i,w} + \sum_{i\in V}\sum_{\substack{i,j\in V\\ j\neq i}} l_{i,j}o'\alpha_{i,j} + \sum_{i\in V} f_i v_i + \sum_{w\in W} g_w x_w \tag{4.1}$$

The first term of this objective function, $\sum_{c\in C}\sum_{i\in V} l_{c,i}u_{c,i}$, is the communication costs between WSI servers and their clients. The second term, $\sum_{i\in V}\sum_{w\in W} l_{i,w}u_{i,w}$, is the communication costs between WSI servers and WSPs. The third term, $\sum_{i\in V}\sum_{\substack{i,j\in V\\ j\neq i}} l_{i,j}o'\alpha_{i,j}$, is the communication costs between WSI servers, where o is a z-dimensional sum vector with all elements equal to 1 and $o'\alpha_{i,j}$ is the communication frequency between WSI servers. The fourth term, $\sum_{i\in V} f_i v_i$, gives the fixed costs to setup all WSI servers. The last term, $\sum_{w\in W} g_w x_w$, refers to the fixed fees that WSPs charge the WSI.

4.6 Constraints

The WSI operates to minimize his operating costs subject to various constraints detailed in the following.

4.6.1 Web Services Usage Constraint

We model a market in which all clients of the WSI are fully served. Each WSI server should process all the incoming requests from his clients. Since we break down the client requests to Web services, each WSI server should obtain enough Web services from WSPs and/or from other WSI servers who enjoy excess Web services minus those he shares with other WSI servers. This requirement is captured by

$$\sum_{c\in C} h_c u_{c,i} \le \sum_{w\in W} q_w u_{i,w} + \sum_{j\in V} \alpha_{j,i} - \sum_{j\in V} \alpha_{i,j} \qquad \forall i \in V \tag{4.2}$$

Here the first term is the total client requests in terms of Web services to WSI server i. The second term is the amount of Web services that WSI server i obtains from WSPs. The third term is amount of the Web services that WSI server i obtains from other WSI servers. The fourth term is amount of the Web services that WSI server i shares with other WSI servers. By rearranging terms, we have

$$\sum_{j\in V} \alpha_{i,j} \le \sum_{w\in W} q_w u_{i,w} - \sum_{c\in C} h_c u_{c,i} + \sum_{j\in V} \alpha_{j,i} \qquad \forall i \in V \tag{4.3}$$

This expression suggests the number of Web services that a WSI server can share with other WSI servers is bounded by his excess Web services - the difference between what he obtains from WSPs and other WSI servers, and what he uses to serve his clients. Constraint 4.2 also implies

$$\sum_{i\in V}\sum_{c\in C} h_c u_{c,i} \le \sum_{i\in V}\sum_{w\in W} q_w u_{i,w}$$

which requires that the Web services the WSI uses for his clients should be fewer than or equal to the Web services he obtains from WSPs. It is equivalent to constraint 3 of problem CP in Chapter 3, which ensures the balance between clients' demands and plants' supplies.

The above implication can be seen as follows. By summing equation 4.3 over i and rearranging terms, we obtain

$$\sum_{i\in V}\sum_{c\in C} h_c u_{c,i} = \sum_{i\in V}\sum_{w\in W} q_w u_{i,w} - \sum_{i\in V}\sum_{j\in V}\alpha_{i,j} + \sum_{i\in V}\sum_{j\in V}\alpha_{j,i} \tag{4.4}$$

Since $\sum_{i\in V}\sum_{j\in V}\alpha_{i,j} - \sum_{i\in V}\sum_{j\in V}\alpha_{j,i} = 0$, we verify the claim.

Constraint 4.2 also insures that a client goes to just one WSI server for an entire request, although it may go to other WSI servers for other requests. A WSI functions as an interface to his clients and needs to process a client request as a whole rather than processing just part of it. A client is concerned only with submitting requests and getting responses rather than how the requests are serviced. In our model, a client request is mapped to a Web services vector. However the mapping takes place on WSI server side and should be completely transparent to clients.

The integer nature of frequency $u_{c,i}$ enables a WSI server to serve a client request as a whole. Since $u_{c,i}$ is integer, $h_c u_{c,i}$ in equation 4.2 refers to one or multiple sets of Web services requirement for each client request. This requires a WSI server to secure enough Web services from WSPs and/or from other WSI servers to fulfill each request as a whole.

4.6.2 WSI Server's Capacity and Client Assignment Constraint

Each WSI server location has a service capacity constraint limiting the amount of requests the server can process per period. This constraint is often a result of service contracts, company's policy, and/or technical limitations posed on the server by its location's infrastructure. This constraint requires that no WSI server can provide more services to its clients and other WSI servers than its capacity allows. In addition, this constraint insures that if a location is not selected as a WSI server location, no clients or WSI servers will request services from or give services to this location.

$$\sum_{c \in C} u_{c,i} + \sum_{j \in V} o'\alpha_{i,j} + \sum_{j \in V} o'\alpha_{j,i} \le s_i v_i \qquad \forall i \in V \qquad (4.5)$$

Here the first term is the number of service requests from all clients to WSI server i. The second term is the amount of sharing from WSI server i to other WSP servers. The third term is the amount of sharing from other WSI servers to WSI server i. The fourth term is the product of the WSI server's capacity and the binary decision variable of the selection of this server location.

4.6.3 WSP's Capacity and WSI Assignment Constraint

Often, multiple WSPs compete to provide Web services to a WSI. The WSI needs to choose those that help to his minimize operating costs while meeting his client demands. For example, if there are two WSPs providing the same Web services, the WSI compares their communication costs and fees and chooses the one with lower costs and sufficient capacity. This selection constraint insures the WSI does not contact any WSP that is not chosen for usage. In addition, a WSP also has a capacity constraint that limits its ability to process the WSI's demand. This limit may be the result of a contract between the WSI and the WSP as well as the maximum physical server capacity of the

WSP. This constraint therefore requires that no WSP can provide services more than its capacity permits.

$$\sum_{i \in V} u_{i,w} \leq x_w t_w \qquad \forall w \in W \tag{4.6}$$

Here the term on the left is the number of service requests from all WSI servers to WSP w. The term on the right is the product of the capacity of WSP w and the binary decision variable of the selection of WSP w.

4.6.4 Client Service Constraint

This constraint insures the summation of all requests from a client should be equal to its service requirement.

$$\sum_{i \in V} u_{c,i} = r_c \qquad \forall c \in C \tag{4.7}$$

Here the term on the left is the number of service requests from client c to all WSI servers. The term on the right is the service level requirement of client c.

4.7 Summary

This chapter defines a WSI location problem. We also detailed a mathematical model to describe the problem. The model captures the characteristics of the WSI and his interactions with his clients and WSPs. The problem objective is to minimize the operating costs of the WSI while satisfying all constraints imposed by the location infrastructures, client demand and WSPs characteristics. In the following chapters, we explore and develop solution procedures.

CHAPTER 5
SOLUTIONS TO FACILITY LOCATION PROBLEMS

5.1 Introduction

This chapter presents a literature review of solutions to facility location problems in general and to the Multilevel Facility-Warehouse (MLFW) location problems in specific. Since most facility location problems can be formulated as Mixed-Integer Linear Programming (MILP) problems, the first attempt is often the applications of well-known exact algorithms, such as branch and bound or cutting planes. Section 5.2 provides a general review of these exact algorithms.

Facility location problems are generally NP-Hard (Garey and Johnson 1979) and are therefore prohibitively difficult to solve when the problem size is big. As a result much research has been devoted to develop efficient heuristic algorithms as a bridge between solution speed and solution optimality. Section 5.3 introduces those heuristic methods that are relevant to our methods. Section 5.4 focuses on the current development of the solutions to the MLFW location problems, since we derive our model from similar models. For excellent reviews of both exact algorithms and heuristic algorithms, see (Francis et al. 1983; Sridharan 1995).

5.2 Exact Methods

Basic exact algorithms for facility location problems start with an enumeration of 2^n possible solutions, where n is the number of potential locations for facilities. For a review of the enumeration method, refer to (Geoffrion and Marsten 1972). Evaluating all potential solutions can be costly in terms of computing time and memory when n is big.

To refrain from assessing all possible solutions, candidates are fathomed through feasibility tests and bounding tests giving the well-known “branch and bound” approach. During the procedure, a series of linear programming (LP) problems are solved that give improved bounds (Land and Doig 1960).

Two ways to convert Integer Programs (IP) to LP problems are often employed. The first one is linear programming relaxation with integer constraint on potential locations being relaxed. The second one is Lagrangian relaxation with certain constraints moved into objective functions. The formulation of the relaxed problem directly influences the efficiency of these methods because the relaxed problem may be too difficult to solve or may not give good lower bounds (Francis et al. 1983).

5.2.1 Linear Programming Relaxation

The capacitated version of the facility location problem has been extensively studied. On the one hand, Sa (1969) developed efficient algorithms through Weak Linear Programming (WLP) relaxation formulation which allowed the assignment of clients to a closed location. Later Ellwein and Gray (1971), Akinc and Khumawala (1977), and Nauss (1978) refined and improved the algorithms. On the other hand, Davis and Ray (1969) and Akinc and Khumawala (1977) proposed algorithms with Strong Linear Programming (SLP) formulation which forbad the assignment of clients to a closed location. Although WLP and SLP are equivalent from the IP perspective, WLP has fewer constraints and is easier to solve (Francis et al. 1983). Nevertheless, Cornuejols et al. (1991) showed theoretically that SLP produced better lower bounds compared with WLP.

In addition, different algorithms have been proposed to solve variations of facility location problems. For example, Ross and Soland (1975) suggested an algorithm to solve problems with a single sourcing constraint and ReVelle and Swain (1970) developed an algorithm to solve problems with a constraint on the number of facilities to be opened.

5.2.2 Lagrangian Relaxation

Lagrangian relaxation also finds lower bounds used in the branch and bound method. It moves one or multiple constraints in the original problem to the objective function with associated multipliers. Given certain values of these multipliers, the newly formed problem is solved in an attempt to form an optimal solution to the original problem. Subgradient procedures are often employed to update and find optimal multipliers for the dual problem (Fisher 1981, 1985; Current et al. 2002). The rationale behind Lagrangian relaxation is that some problems are easier to solve without certain side constraints. Moving these constraints to the objective function with penalties simplifies the original problems.

Lagrangian relaxation has been widely used for facility location problems either independently or in combination with other heuristic methods. For example, Beasley (1993) developed a Lagrangian heuristics based on Lagrangian relaxation and subgradient optimization. The success of Lagrangian relaxation depends largely on the selections of the constraints to relax. The tradeoff between solution speed and good lower bounds needs to be carefully studied. Some select demand constraints to be relaxed (Geoffrion and McBride 1978; Nauss 1978; Christofides and Beasley 1983; Van Roy 1986). Others select capacity constraints (Van Roy 1986; Cornuejols et al. 1991).

5.2.3 Decomposition Method

Facility location problems formulated as MILP problems are also open to decomposition techniques, such as Benders decomposition (Benders 1962) and Cross Decomposition (Van Roy 1986). Benders decomposition converts a difficult IP problem into a collection of simple LP subproblems. To do so, the routine separates variables into two sets, including one set for integer variables. It successively fixes the values of integer variables and solves the resulting LP problem. At each step, the integer variables are updated by an IP Bender's master problem, which derives its constraints from the solutions of the dual of the LP problems in the previous iteration. Benders decomposition was so computationally efficient that it has been applied to large-scale MLFW location problems with multiple commodities (Geoffrion and Graves 1974).

The cross decomposition algorithm transforms a capacitated problem into uncapacitated problems by dualizing the capacity constraint (Van Roy 1986). This algorithm serves as a framework that unifies Benders decomposition and Lagrangian relaxation. The multiplier values used for the Lagrangian relaxation are the successive solutions to multiple transportation problems.

5.3 Heuristics

The computer time needed by exact algorithm is exponentially related to the number of integer variables (Khumawala 1974). These algorithms work well on small problems. However, for problems with large number of constraints, an exact algorithm consumes a prohibitive amount of computer power with no guarantee for optimal solutions. In contrast, heuristic algorithms often generate optimal or near optimal solutions quickly. For many problems of realistic sizes, heuristic methods are often the only practical option.

Two types of heuristic procedures are often used to solve facility location problems. The first one is called a constructive algorithm, which builds a solution from scratch. It features greedy ADD and DROP heuristics. The second one is called an improvement algorithm, which builds on an existing solution. It features neighborhood search, Alternate Location Allocation (ALA), Vertex Substitution Method (VSM), and tabu search. Both procedures have been employed to solve uncapacitated facility location problems (Rapp 1962; Kuehn and Hamburger 1963; Feldman et al. 1966; Teitz and Bart 1968). They have also been adapted for capacitated versions. For example, based on the heuristics for uncapacitated facility location problems, Jacobsen (1983) provided generalized heuristics, including ADD, DROP, SHIFT, ALA and VSM, for the capacitated version. The solution methods we propose for our problem derive from these heuristics, and therefore we devote more discussion detailing the specifics.

The following notations will be used in the following sections to illustrate heuristics for location problems

$K0$: subset of K for which $P_k = 0$,

$K1$: subset of K for which $P_k = 1$,

$C*(K, I)$: the optimal solution to a transportation model with source set K and sink set I and data in problem SP.

5.3.1 Construction Heuristics

Both construction heuristics ADD and DROP are greedy heuristics. In general, greedy heuristics assume the path to a global optimal solution is a sequence of locally optimal solutions. They iterate through the path and try to improve on the solution of a previous step. The process terminates when no more improvements can be made. Being heuristic, the procedure may result in a less-than-global-optimal solution because of its myopic nature. Nevertheless, greedy heuristics are generally fast and the results are reasonably good, especially when used with other techniques at the same time.

Kuehn and Hamburger (1963) described an ADD procedure for uncapacitated location problems. Their procedure located a facility that gave the greatest savings at each step and iterated through all potential locations until no further savings could be made. If there were n potential facility locations, the procedure would reduce the number of cost evaluations from 2^n, a complete enumeration, to a number lower than n^2 (Kuehn and Hamburger 1963). Although the solutions obtained might not be optimal, they are often good ones, and, more importantly, the procedure drastically reduces computer time and memory usage.

At the beginning of this procedure, there are no open facilities except a super source *SS* to have an initial feasible solution. That is only *SS* is in the set $K1$. All potential facility locations are in $K0$. The procedure is detailed as follows (Sridharan 1995):

1. For each $k \in K0$ compute

   $\sigma_k = C^*(K1, I) - C^*(K1 \cup \{k\}, I) - f_k$.

2. Find the plant k^* that maximizes σ_k from

   $\sigma_k{}^* = \max_{k \in K\phi} \{\sigma_k\}$.

3. If $\sigma_k{}^* > 0$, move k^* to $K1$ and go to Step 1. Otherwise, terminate the procedure.

For uncapacitated facility location problems, the cost evaluation of transportation models step 1 is rather simple. However, for the capacitated version, solving $|K0|$ capacitated transportation problems during each iteration can be costly (Jacobsen 1983). This is especially true for large-scale problems. To reduce computer time, easily calculable approximations to $C^*(K1 \cup \{k\}, I)$ are needed. Khumawala (1974) suggested an effective lower bound by solving a continuous knapsack problem.

Jacobsen (1983) also provided an upper bound, as well as a lower bound, on the savings. He proposed procedures termed ADD-LO and ADD-HI to use these bounds. Jacobsen's ADD-HI procedure is basically equivalent to Khumawala's Largest Omega Rule (1974). Domschke and Drexl (1985) improved the ADD-LO and ADD-HI procedure by providing priority rules to handle problems with different facility capabilities.

Feldman et al. (1966) designed the DROP procedure for uncapacitated location problems. At the beginning, all potential facility locations are in the set $K1$. The procedure then drops a location that gives the greatest savings at each step and expects to find optimal or near optimum solutions at the end of the iterative procedure. The procedure is detailed as follows (Sridharan 1995):

1. For each $k \in K1$ compute

   $\sigma_k = f_k + C^*(K1, I) - C^*(K1 \setminus \{k\}, I)$.

2. Find the plant k^* that maximizes σ_k from

   $\sigma_k{}^* = \max_{k \in K1} \{\sigma_k\}$.

3. If $\sigma_k{}^* > 0$, move k^* from $K1$ to $K0$ and go to Step 1. Otherwise, terminate the procedure.

Just like in ADD procedures, there are $|K1|$ transportation problems to be solved during each iteration (Jacobsen 1983). To reduce computer time, procedures to approximate the solutions are needed. For capacitated transportation problems, Akinc and

Khumawala (1977) and Jacobsen (1983) suggested both lower and upper bound on the savings.

5.3.2 Improvement Heuristics

One of the problems with greedy heuristics is that they do not allow revisions of earlier decisions. For example, the ADD procedure will keep a facility open for the final solution once it was selected. Similarly, the DROP procedure will not reopen a facility it closed in a previous step. However, there are cases where decisions made at intermediate steps are only local optima. Apparently, some revisions of past decisions may be called on for better global solutions. To compensate for this inherent "deficiency" of greedy heuristics, various improvement heuristics are designed. The basic idea of these heuristics is to improve the solution by perturbing the result of greedy heuristics to escape local optimality. Several procedures, including neighborhood search, ALA, VSM, and tabu search, are among the most popular ones. It should be noted that improvement heuristics do not necessarily require solutions from greedy heuristics. In general, they work well with any feasible solutions.

Neighborhood search was among the earliest improvement heuristics proposed to solve facility location problems (Maranzana 1964). Starting with a feasible solution, the procedure develops a neighborhood around each open facility based on customer assignment to the facility. Within each neighborhood, the procedure then chooses the optimal facility location for the neighborhood by solving a 1-median problem. Subsequently, new neighborhoods are defined according to new customer assignment. The procedure repeats itself until no new neighborhoods are formed.

The ALA was proposed by (Rapp 1962). To solve a p-median problem, the procedure closes a facility from a feasible solution set $K1$ and uses one ADD step to optimize the problem. It repeats itself until a sequence of $|K1|$ passed before an improvement is made. Teitz and Bart (1968) designed VSM, which is basically the opposite of ALA. Starting with a feasible solution, at each step, the procedure adds a facility from a closed set $K0$ and then optimizes the solution through one DROP iteration. The procedure repeats itself until at each location in $K0$ a facility has been added and DROPped. ALA and VSM, when used with ADD or DROP, may not dramatically improve solutions since the number of open facilities remains the same.

Various tabu search schemes were also developed to battle the trap of local optimality of greedy heuristics (Glover 1989, 1990). They impose restrictions on other heuristic procedures. The restrictions include making some movements illegal or some time-dependent. It also allows “aspiration criteria” for good solutions to overcome certain restrictions. Tabu schemes have been implemented to solve various facility location problems (Rolland et al. 1997; Gendron and Potvin 1999).

5.4 Multi-level Capacitated Facility-Warehouse Problem

Capacitated p-median problems are special cases of the MLFW problems. Since capacitated p-median problems are NP-Complete (Garey and Johnson 1979), the MLFW problems are also NP-Complete. The MLFW problems are sometimes modeled with a parameter for multiple commodities. Many current algorithms used to solve multi-commodity MLFW problems are either Benders decomposition-based algorithms (Geoffrion and Graves 1974) or branch and bound methods (Hindi and Basta 1994; Tragantalerngsaka et al. 1997).

With the Benders decomposition method, Geoffrion and Graves (1974) decomposed the multi-commodity MLFW problem by the number of commodities. For each commodity, an independent classical transportation problem was formed. This decomposition allowed the model to efficiently handle problems with a large number of commodities.

Hindi and Basta (1994) solved a multi-commodity MLFW problem with a branch and bound algorithm based on a weak LP relaxation of the problem. Their lower bounds were the solutions of a multi-commodity network flow problem with capacity constraints. Unfortunately, problems with a multi-commodity integral flow problem are NP-complete. They are difficult to solve even with two commodities (Skiena 1997). It should be noted that the fixed costs for plants were not considered in the model.

Klose solved two-stage capacitated facility location problems with single product using both LP relaxation (Klose 1999) and Lagrangian relaxation (Klose 2000). With Lagrangian relaxation, he relaxed the capacity constraints and produced a sub-problem of an uncapacitated facility location problem with aggregate capacity constraints.

Pirkul and Jayaraman (1998) introduced a formulation for a PLANWAR problem in which the selections of plants and warehouses, both with fixed costs, were decision variables. They solved the problem through Lagrangian relaxation and heuristics that use the result of the relaxation. For the Lagrangian relaxation, they introduced the customer-demand constraints and demand and supply balance constraints, which were constraints 1 and 3 of problem CP in Chapter 3 respectively, into the objective function. Their problem size was 20 warehouses, 10 plants, and 3 types of products.

5.5 Summary

In this chapter, we presented a review of solutions to various facility location problems. These solutions include both exact algorithms and heuristic algorithms. Many of the exact algorithms take advantage of IP relaxation or Lagrangian relaxation to calculate bounds on savings. Since most facility location problems are NP-hard, various heuristics are designed to solve problems of realistic size. We reviewed both constructive and improvement heuristics. When used properly, these heuristics provide good solutions within a reasonable amount of computer time.

CHAPTER 6
HEURISTICS DEVELOPMENT

6.1 Introduction

WSI location problem is a multi-service capacitated two-level server location problem with added constraints on bundling and sharing. As mentioned in Chapter 5, many exact algorithms have been developed for similar problems. None of them, however, can be readily adapted to solve our problem because of the extra constraints and the size of the problem. This chapter presents DAL (DROP, ALA and LP relaxation), a heuristic method to solve the problem. Section 6.2 outlines the overall procedure of the DAL heuristics. This procedure incorporates a mixture of greedy DROP heuristics, ALA improvement heuristics, and LP relaxations of Integer Transportation (IT) problems. Section 6.3 discusses the DROP heuristics and LP relaxations. It also introduces a method to solve the possible problem of infeasibility because of LP relaxations. Section 6.4 details the implementation of ALA heuristics.

6.2 Overview of the DAL Heuristics

Figure 6-1outlines the DAL heuristics employed to locate WSI servers and WSPs. The heuristics is iterative and consists of two major procedures. The first one derives from a DROP procedure and the second one from an ALA procedure. Both parts treat the locations of WSI servers and WSPs in the same way when deciding which one to drop or to add.

At each step of either procedure, we need to solve many IT problems. To reduce computer time, we relax the integer constraints on the communication frequencies. The resulting LP relaxed problems are solved for a lower bound, which is then used for selecting which location to drop or add. At the end of the DROP procedure, we solve an resulting IT problem for an integer solution. If an integer solution is not available, the procedure will backtrack and add one location at a time starting with the most recently dropped one. It then solves the resulting IT problem again. The backtracking and addition stop when an integer solution is obtained.

Once the DROP procedure determines the number and locations of WSI servers and WSPs, it will pass the solution set to the ALA procedure for further improvement. The ALA procedure will drop one location at a time from the solution set and replace it with a location that has been closed for possible lower cost. There will be many IT problems to be solved for each replacement. Again, the integer constraints on the communication frequencies will be relaxed and the resulting problems will be solved for lower bounds. The procedure uses the lower bound to determine which location comes into the solution set. In addition, it will also solve an IT problem each time a new location is to be added to the solution set to verify the availability of a feasible integer solution.

6.3 Settings

The following notations will be used in the following sections to elucidate the particulars of the DAL heuristics. Let there be a set of potential WSI servers and WSP locations $k \in K$ and a set of clients $c \in C$.

$$P_k = \begin{cases} 1 & \text{if a location is selected at } k, \\ 0 & \text{otherwise.} \end{cases}$$

$K1$: subset of K for which $P_k = 1$,

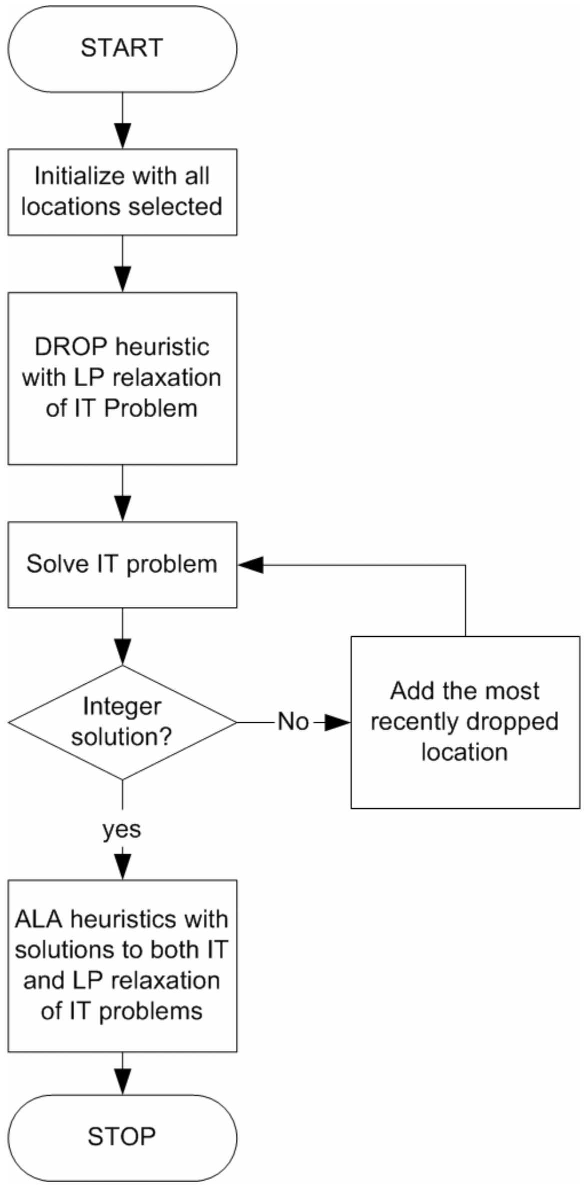

Figure 6-1. DAL Heuristics Flowcharts.

$K0$	: subset of K for which $P_k = 0$,
$C^*(K,I)$	: the optimal solution to an IT problem with source set K and sink set C.
$C(K,I)$	: the lower bound of an IT problem with LP relaxation of the integer constraints on communication frequencies with source set K and sink set C.

As discussed in Chapter 5, two of the predominant heuristics used to find cost-improving solutions to a single-product single-level facility location problem are the construction heuristics, including DROP and ADD, and improvement heuristics, including ALA. These two heuristics are often used together and, in practice, they perform well. With some modifications, we adapted the DROP and ALA procedures to form the DAL heuristics.

The common practice of DROP and ALA procedures is to select one potential facility location to drop or add at a step. If we limit our heuristic selections only to potential WSI server locations, we will need to solve, at each step, a multi-product transportation problem with the selection of WSPs as a decision variable. This is a difficult problem by itself, with integer constraints on both the selection of WSPs and the communication frequencies. This situation is especially dire when the number of potential WSPs is large. The LP relaxation of the integer constraints on the selection of WSPs' locations is not particular useful for DROP heuristics since it is not clear how to handle partial selections of a location.

The DAL heuristics treat both the potential WSI server locations and WSPs locations equally when choosing one to drop or add. The only selection criterion is the saving that dropping or adding a location brings. This equal treatment is viable because both types of locations incur fixed costs. For a WSI server location, it is the fixed location infrastructure cost for hosing this server. For a WSP location, it is the fixed service fee that it charges the WSI.

6.4 DROP Procedure

At the beginning of the DROP procedure, we open all the locations, including WSPs locations and WSI server locations, as an initial solution. This means all locations are in the set $K1$. The procedure is detailed as follows:

1. For each $k \in K1$ compute

   $\sigma_k = f_k + C(K1, I) - C(K1 \setminus \{k\}, I)$.

2. Find the location k^* that maximizes σ_k from

   $\sigma_k^* = \max_{k \in K1} \{\sigma_k\}$.

3. If $\sigma_k^* > 0$, move k^* from $K1$ to $K0$ and go to Step 1. Otherwise, terminate the procedure.

At step 1, there are $|K1|$ IT problems to be solved. As mentioned before, an IT problem is difficult to solve when the problem size is big. To reduce computer time, approximation to an optimal solution is needed. By relaxing the integer constraints on the communication frequencies, we simplify the original IT problem to one with fixed selections of WSI server and WSPs locations and without any integer constraints.

At the end of the DROP procedure, we will solve an IT problem to obtain $C^*(K, I)$. If the number of the locations in the solution set $K1$ becomes smaller after the procedure, then this IT problem will be easier to solve than the initial and most intermediary IT problems.

Because we solve all IT problems with LP relaxations during the procedure, there is a chance the final IT problem may not have a feasible integer solution. This infeasibility can happen especially when the capacity constraints are binding or near binding in the LP problem that determines the last drop of a location. This problem can be solved by allowing the procedure to backtrack to previous location decisions iteratively until we find a feasible integer solution. For example, for a particular problem, we dropped k_m, k_n and k_o sequentially. If there is no integer solution when the DROP procedure terminated, we will add k_o back to the solution set and solve the problem again. We will continue to add those dropped locations back to the solution set one at a time until we find an integer solution. If we returned to the initial solution to the problem and there is still no integer solution, then the problem does not have a feasible integer solution.

6.5 ALA Procedure

Once a feasible integer solution is found, the solution set $K1$ will be passed to the ALA procedure to serve as an initial solution. The ALA procedure is detailed as follows:

1. Drop a $k' \in K1$ that has not been dropped before. Terminate when every location has been dropped once.
2. For each $k \in K0$ compute

   $\sigma_k = C(K1, I) - C(K1 \cup \{k\}, I) - f_k$.
3. Find the location k^* that maximizes σ_k from

   $\sigma_k{}^* = \max_{k \in K0} \{\sigma_k\}$.
4. Solve $C^*(K1 \cup \{k^*\}, I)$. If not feasible, add k' to $K1$ and go to Step 1.
5. If $\sigma_k = C^*(K1, I) - C^*(K1 \cup \{k^*\}, I) - f_{k^*} > 0$, add k^* to $K1$ and go to Step 1. Otherwise, add k' to $K1$ and go to Step 1.

During the procedure, there are $|K0|$ IT problems to be solved at Step 2. To reduce computer time, we again approximate an optimal solution by relaxing the integer constraints on the communication frequencies. At Step 4, we verify that replacing k' with $k*$ gives an integer solution by solving an IT problem.

6.6 Summary

To deal with the size and complexity of the problem in real time, we developed the DAL heuristics, a combination of construction heuristics, improvement heuristics, and LP relaxation of integer constraints. The main feature of these heuristics is to treat both WSI server locations and WSP locations equally. Subsequent experimentations will verify the efficiency of this procedure.

CHAPTER 7
DATA COLLECTION AND ANALYSIS

7.1 Introduction

Our experiment consists of the executions of location problems with simulated data under various conditions. The data will test the effectiveness of the DAL heuristics and analyze the characteristics of the WSI location model. Since the model has the potential to be applied to many Web services applications, our goal is to gather and/or simulate the data in a way so that they are representative and general enough to be used in various situations.

7.2 Data Analysis

Model testing often benefits from using authentic data. This poses problems for our research. First, our model does not target at a specific application instance. It attempts to describe a general relationship that is evident in many Web services applications. Second, Web services technology is new and has not been widely implemented. Real life data therefore are not readily available. In our experiment, we use or generate reasonable and representative data and allow values of parameters to be modified to mimic different types of applications.

Our analysis consists of one-period problems of deciding the locations of WSI server locations and WSPS, and communication frequencies between all the participants, including WSPs, WSI and his clients,. The data include up to 40 unique clients with distinct demand vectors of Web services and service level requirements, up to 40 WSPs with distinct supply vectors, service capacities and fixed fees, and up to 40 potential locations of WSI servers with service capacities and fixed costs. All costs or fees tend to vary across applications. We are required in some instances to make assumptions about the values by observing similar real life problems.

7.3 Network Locations and Latencies

We populate a network with nodes/locations and arcs using the US Internet backbone networks. Some facility location problems deliberately demand that their plants and warehouses do not share the same locations. In our problems, all participants choose their locations independently. Depending on the application, though, we can limit each type of players to certain locations. The distances between the locations stand for network latencies, which are further converted into communication or opportunity costs for modeling purposes.

Some Web services applications may involve many clients, WSPs, and potential WSI server locations. For example, a stockbroker application may serve thousands of clients from various locations, contact hundreds of WSPs for Web services, and consider hundreds of potential locations to host its servers. Problems of this magnitude are challenging for any solution method. Some level of data aggregation is both desirable and necessary.

It has been observed that many participants of online services are clustered geographically. For example, locations such as New York or San Jose with good information technology infrastructures are attractive to many service providers and clients. Furthermore, participants near one another share similar characteristics in pricing, supply and demand. It is determined that aggregation of individual locations to city level is justified.

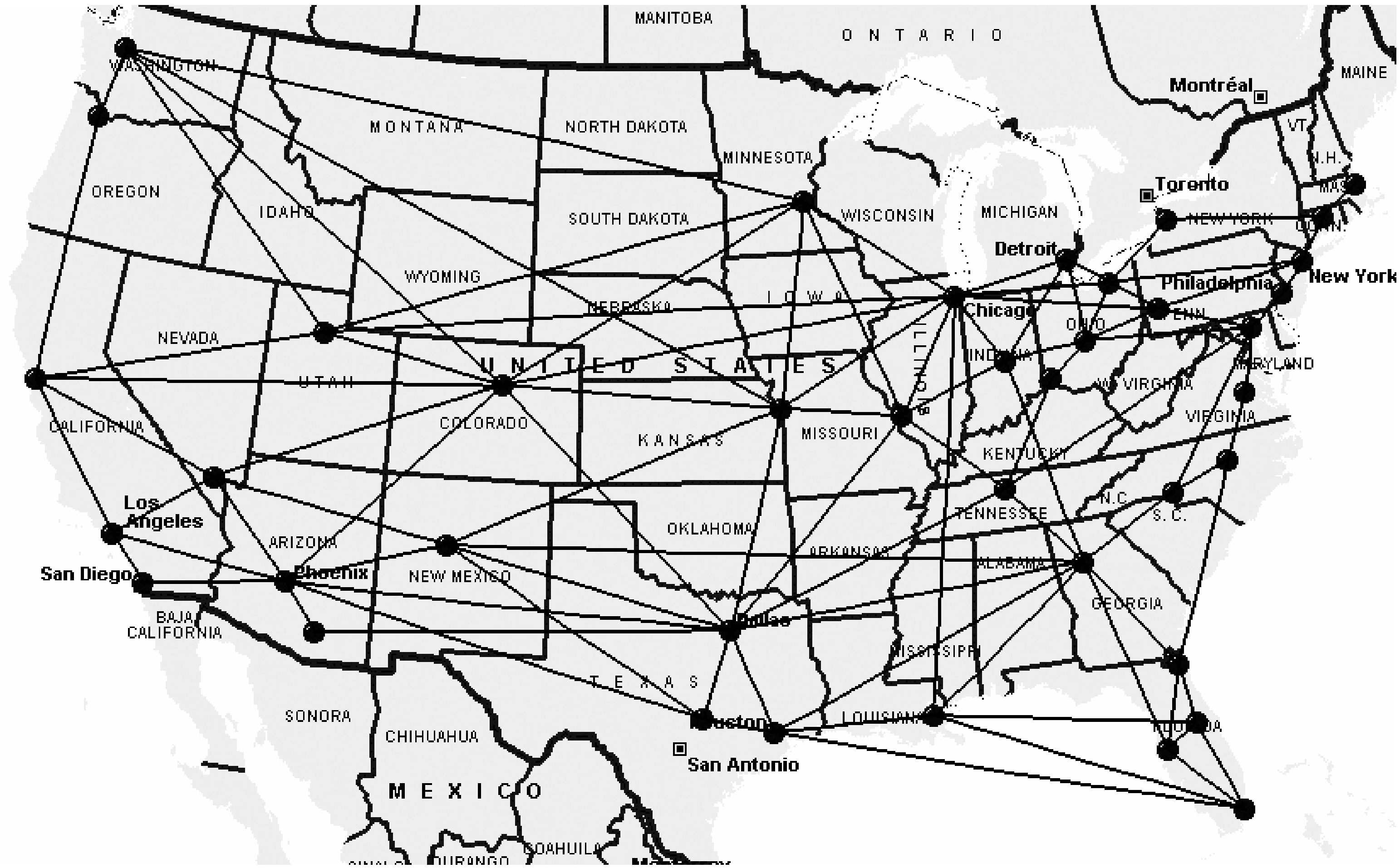

Figure 7-1. US Internet Backbone Networks with 40 City Locations and Connections.

Forty cities on the US Internet backbone networks are selected as the locations for the participants in our model. See Appendix C for a list of these cities. Figure 7-1 shows the entire network with 40 city locations and connections between them.

The reason that we choose the cities on the Internet backbones is that they are, because of their strategic locations, more populous with service participants than places off the backbones. They are therefore more representative of the network typology and conditions on the Internet. In addition, participants not located on the backbones but adjacent to these cities still connect to them for Internet connections. As a result, these cities may serve as the approximate centers of the clusters formed with their surrounding places.

We acquired 26 maps of the US Internet backbone networks operated by various companies and organizations (Internet Backbone Maps n.d.). These maps show more than 100 distinct cities on the backbones. Many of them serve as hubs connecting to more than one neighboring city. Forty of them are chosen as representatives to make the problem manageable. The following criteria are used to disqualify certain cities from our network. First, if a city connects to only one city, it will be disqualified since its neighbor can represent it without losing important typological characteristics of the network. However if the city is far from its neighbor, we will keep it on the network to preserve this characteristic. Another criterion is that if a city is a hub and is adjacent to another more connected hub, it may be disqualified since it can be approximated by its neighboring hub geographically.

7.4 Network Latency Measurement

The communication/opportunity cost is a linear function of the network latency. The direct measure of network latency is difficult, if not impossible. Accurate readings exist in labs where external factors, such as network congestions and Internet protocols in use, are controlled. Nevertheless, different measures to approximate the network latency are available. Most of them unavoidably add noise to the measurement. For example, network latency measured using RTT is sensitive to network conditions and is subject to change across time.

To escape from external influence, we choose to use geographical distance to approximate the network latency. There are several justifications for this choice. First, since these cities are on Internet backbone networks with tremendous bandwidth, delays due to the network congestions are minimized. There is a correlation between network latency and geographical distance, and this correlation is strong especially when hosts are on the same backbone network (Gwertzman and Seltzer 1995). On our network, although all cities are not on the same backbone network, the correlation should still be stronger than if they are not on the backbone network. Therefore, physical distance is a prominent factor in determining network latencies between those cities.

Except physical distance, all factors that have been used to approximate network latency are constantly improving. Measures based on these factors therefore need to be updated frequently. In contrast, physical distance remains relatively stable. Approximations of network latency based on physical distance can survive small Internet typological and protocol changes. After all, physical distance provides a lower bound on network latency. And it is expected that eventually physical distance will dominate the measure of network latency as other factors become progressively more irrelevant.

Exact measure of physical distance of the backbone connection between two cities is not readily available. Occasionally there may be multiple lines of different lengths operated by different companies linking two cities. Therefore Euclidean distance, the lower bound of the physical distances of the connections, is used. Distance between any two directly connected cities is obtained through commercial services. See Appendix B for a list of the distances of the arcs on the network. We then computed the shortest path between any two cities to serve as our latency measure.

Next, network latency is converted to communication/opportunity cost. This cost varies across different types of applications. For some applications, such as stock brokerage with time-sensitive clients, the opportunity cost can be high. We will linearly convert the Euclidean distance into opportunity cost and vary this parameter in our experiments to simulate different types of applications.

7.5 WSI Servers Fixed Cost

A WSI rents hosting services from Web hosting companies. The rental fees can be in the form of either variable fees based on usage, fixed fees, or both. Our model incorporates the variable fees into the communication cost between WSI servers and their clients.

There are several major factors affecting the cost of hosting a WSI server. The first one is the bandwidth amounts the server uses per period. Bandwidth usage often depends on the frequencies of clients' service requests if the requests are of similar size. The more service requests a WSI server needs to process, the more bandwidth a host will allocate for the WSI server and the more the host charges the WSI.

The second factor is the level of host computing supports for the WSI server. These supports include, among others, computing power, programming language support, and database connections. Quite often, a WSI application assimilates different services from various WSPs for a cohesive response to its clients. A database offers necessary intermediate storage of the data from WSPs before being processed by the WSI. In addition, the WSI may use the database to temporarily store data for sharing purpose.

The third factor is the quality of service that a host offers, including service response time and service reliability. Time-sensitive applications with high opportunity costs will appreciate the savings from renting from a host that enables the provider to respond to its client request quickly. Dedicated hosting, compared with shared ones, offers lower service response time and therefore commands higher fees. In addition, reliable services with data backup, server security, and quick recovery from system breakdowns are also pricey.

Hosting services are available with a wide range of prices. One exemplary service that compares hosting prices and features is UpTown Web Hosting (n.d.). Service prices range from as low as $10 a month to more than $5000 a month. A service plan from with a dedicated server suitable for demanding commercial service applications is shown in Table 7-1.

Table 7-1. A Service Plan from UpTown Web Hosting.

Monthly price: $4995.00 Setup fee: $500.00 Disk space: 60000 MB Data transfer: 30000 GB / month Server Operating System: Linux Database: MySQL Server software: Apache POP mailboxes: Unlimited Mailing lists: Unlimited Unlimited subdomains Rating: 1.76 out of 10 Based on 14 votes 24-hour phone support: No Daily backup: Yes	Server features: SSL SSH or telnet Control panel Usage stats Streaming audio support Streaming video support Own IP address Anonymous FTP Access to raw logs Shopping cart	Server programming support: Perl PHP Server C/C++ Python Miva Merchant Active Server Pages (ASP) FrontPage server extensions

7.6 WSP Fixed Fees

A WSP charges both fixed fees and variable fees for the services requested by a WSI. Our model incorporates the variable fees into the communication cost between WSI servers and WSPs. Several factors, including the quantity and the quality of the services rendered, influence the price. The quality may have several facets, including speed, reliability, and intrinsic values of the services. A WSI is willing to pay premium for fast services if his clients are time-sensitive. In addition, if the service is of high value, such as proprietary stock price prediction, it will cost more. Because of these factors, there can be a wide range of fees depending on the applications in question. For experimentation, we will vary these fees to see how they affect solutions.

7.7 Summary

In this chapter, we described the sources of our data. Our principal goal is to gather and/or simulate the data in such a way to make them reasonable and representative. With minor adaptation and modifications, the data can be flexible enough to simulate various problem instances.

CHAPTER 8
EXPERIMENTAL DESIGN

8.1 Introduction

This chapter presents an experimental design to test the efficacy of the DAL heuristics and the resulting design of the WSI location model under various conditions. Section 8.2 introduces the basic experiment environment and setup in which we conduct all our tests. Section 8.3 lays out plans to evaluate the performance of the DAL heuristics in terms of solution quality, computer time and robustness. Different test scenarios are set up by varying the size of problems, the values of parameters, including the fixed costs of WSI server locations and WSP fixed fees, communication costs and dimensions of the Web services vectors. Section 8.3 proposed six hypotheses to study the resulting design of the model. We are most interested in knowing how various factors affect the optimal solutions in terms of the number of WSI servers and WSPs, and the number of Web services being shared between WSI servers.

8.2 Experiment Environment and Setup

During the experimentation, we test the DAL heuristics against an exact algorithm as provided by CPLEX 8. The CPLEX 8 is a general-purpose commercial integer programming package. It will be used as a benchmark algorithm in this study. Because of its efficiency in solving IP/LP problems, CPLEX 8 is also used to solve the LP and IP sub-problems in the DAL heuristics.

By default, CPLEX 8 uses “Best Bound search,” a node selection strategy that may generate a long list of unexplored nodes for large problems (ILOG 2002). With this strategy, CPLEX needs exponential memory and often runs out of virtual memory when solving difficult problems.

All tests are conducted on the same computer to rule out any variability as a result from differing configurations. The machine configurations and software used are detailed in Table 8-1.

Table 8-1. Machine and Software Configurations

CPU Model and Speed	Pentium III, 933MHZ
Memory Size and Speed	512MB, 100MHZ
Hard Drive Speed	7200RPM
Operating System	Windows 2000 Professional
Compiler	Visual Studio .Net C++ 1.0
CPLEX Version	8

Since our model is new, there are no benchmark problems from the literature. In addition, Web services have not been widely implemented in practice and therefore direct access to all the real data necessary for our test is not available. As a result, we randomly generate the problems based on the real situation of Web services implementations using the network layout described in Chapter 7.

To facilitate the generation of the problems and the comparisons between results, we assign base values to the model's parameters. These base values are varied to simulate different scenarios and applications. They are derived from our previous data analysis to make them representative and realistic. We convert physical distance to communication cost in dollars by dividing the former one with a denominator. For example, Table 8-2 lists some of the base values for the parameters of a particular problem.

Table 8-2. An Example of Base Values for a Problem.

Parameters	Base Value
WSP Capacity	50000
WSP Fixed Charge	$3000
Value of Each WSP Service Vector Element	2
WSI Fixed Cost	$3000
WSI Capacity	50000
Client Demand	1000
Value of Each Client Service Vector Element	1
Communication Cost/ Distance to Cost Denominator	100

For every problem, the values of each parameter randomly vary within a 40% range with the parameter's base value as the medium for the efficacy test and a 4% range for the parameter test. For example, if the range is 40% and the base value of WSP capacity is 50000, then the value for each potential location will be uniformly distributed between 30000 and 70000.

8.3 Performance Measures

As the efficacy of the DAL heuristics is concerned, we are interested in answering the following questions:

1. Are the solutions found by the DAL heuristics good compared with optimal solutions?
2. Are the DAL heuristics fast enough to justify their usage?
3. Are the DAL heuristics robust enough to perform well over a wide range of test problems?

We answered these questions by benchmarking the DAL heuristics against CPLEX 8 with empirical testing. We considered eight classes of problems defined by the number of clients, WSPs and the potential WSI server locations. Table 8-3 shows the values of these classes.

Table 8-3. Performance Test Class

Class	a	b	c	d	e	f	g	h
Number of WSPs	5	10	15	20	25	30	35	40
Number of WSIs	5	10	15	20	25	30	35	40
Number of Clients	5	10	15	20	25	30	35	40

In each class, there are eight instances defined by the values of WSI and WSP fixed costs, communication costs, and the dimensions of Web services vectors. Table 8-4 shows the configurations of the eight tests for each class.

Table 8-4. Test Instance for Each Class.

Problem	1	2	3	4	5	6	7	8
WSI, WSP Fixed Cost/fee	1500	1500	1500	1500	6000	6000	6000	6000
Communication Cost	50	50	200	200	50	50	200	200
Dimensions of Services Vectors	2	8	2	8	2	8	2	8

8.3.1 Solution Quality

The DAL heuristics are only useful in practice if the quality of their solutions is near optimal. For some large problems, especially those with more than 25 locations for WSPs and WSI servers, CPLEX may take a long time to converge and may even run out of memory before providing an optimal solution. For our test instances, CPLEX can solve small-sized ones and thus give us the opportunity to compare DAL's solutions to the optimal solutions. We will report both the objective function values of the DAL heuristics and their relative standard deviations.

8.3.2 Solution Speed

Also of interest is whether the DAL heuristics are fast to terminate. We would like to examine empirically how solution times vary with problem size and different values of parameters. We will also study the computer times of the DAL heuristics for problems of different sizes against those of CPLEX 8. The number of location choices can be the biggest factor determining the size of a problem. In addition, the dimension of the Web services vector may also complicate a problem and therefore affects the solution speed. We will examine how these two factors affect DAL's speed.

In our experiment, the DAL heuristics and CPLEX require similar preprocessing, including computing the distance matrices, loading the test data into memory, and so forth. In addition, the time used for preprocessing is short and does not vary much across problems. Therefore, we do not record it.

8.3.3 Solution Robustness

We also examine if the DAL heuristics are robust for different problems. If they demonstrate good performance across a wide range of problem instances, we can be relatively confident that the heuristics are robust for the problems tested.

8.3.4 Performance of ALA Compared with DROP

The DAL heuristics consist of DROP and ALA procedures. The DROP procedure finds a good feasible solution quickly and the ALA procedure attempts to improve it. Both DROP and ALA solve a series of transportation problems and can be expensive in terms of computer times. In addition, ALA improves on the result of DROP without modifying the number of WSI servers and WSPs. We would like to see if ALA improves the solutions in our test and how much and at what expense in terms of computer time. During the experiment, we will record the computer times and solutions of both DROP and ALA procedures and determine if the extra computer times that ALA takes is justified by the improvement of solution quality it makes.

8.4 Hypotheses

We are also interested in knowing how various factors affect the design of the model. These factors include communication costs, the fixed costs of WSPs and WSI server locations, the number of client demands per period, and the number of Web services to be shared per sharing transaction between WSI servers. For each of these factors, we conducted 5 sets of 9 to 10 tests with 17 randomly generated locations for WSPs, WSI servers, and clients. The total number of tests conducted is 190. Within each set of tests, the value of the parameter under investigation is modified while others are being held constant. Table 8-5 shows the common base values for these tests. We artificially increase the capacities of WSP and WSI to a large number to avoid their influences on the test results. We set the number of Web services shared per sharing to 5 to facilitate the observations of sharing activities during the test.

Table 8-5. Common Base Values for Test Problem Parameters.

WSP Fixed Fee	1000
WSI Fixed Cost	1000
Communication Cost/ Distance to Cost Denominator	100
Dimension of Web Services Vectors	2
Client Demand	1000
WSP Capacity	5000000
WSI Capacity	5000000
Value of Each Element in WSP Web Services Vector	2
Value of Each Element in Client Web Services Vector	3
Amount of Web Services Shared per Sharing	5

Table 8-6 summarizes the modifications of the values of four experimental parameters. Up to 10 different values are assigned to each of them.

These tests are designed to verify the following expectations and test how sensitive the model is to altering the values of certain parameters. Communication costs or the latency costs may have impacts on solutions. One of the premises of our model is that network latency can be considered as a form of opportunity costs and therefore a WSI has incentive to minimize network latency to reduce his operating cost. We expect that for services with high latency cost, a WSI, facing time-sensitive clients, will increase the number of his servers and contact more WSPs to lower the overall network latencies.

Table 8-6. Experiment Parameters.

Denominator for Communication Cost	WSI and WSP Fixed Cost	Client Demand	Amount of Web Services Shared per Sharing
1	100	100	1
2	200	200	2
4	400	400	3
8	800	800	4
16	1600	1600	5
32	3200	3200	6
64	6400	6400	7
128	12800	12800	8
256	25600	25600	9
512			10

H1. *High latency costs and a large number of time-sensitive clients will increase the number of WSI servers and WSPs.*

In addition, modifying the values of WSI and WSP fixed costs and client demands may also influence the number of WSI servers and WSPs and/or the sharing between WSI servers. Reduced fixed cost will make latency cost to account for a larger portion of WSI operating cost. The WSI therefore has a bigger incentive to setup more servers and contact more WSPs to reduce the overall network latency. An increase in client demand will also increase the importance of latency cost versus fixed costs. For that reason, we again expect the WSI to setup more servers and contact more WSPs. Furthermore higher client demand may also proportionally increase sharing between WSI servers, if there are sharing at all.

H2. *High WSI and WSP fixed costs will decrease the number of WSI servers and WSPs.*

H3a. *High client demand will increase the number of WSI servers and WSPs.*

H3b. *High client demand will increase the amount of sharing between WSI servers proportionally.*

In our mathematical model, we allowed one Web service to be shared between WSI servers per sharing transaction. This may not be realistic for many real life applications. The WSI servers may increase the number of Web services to be transferred to other servers at a time. We expect this increase will lower the cost of sharing and thus result in more sharing activities between WSI servers.

H4a. *A high number of Web services to be shared between WSI servers per sharing transaction will increase the number of WSI servers.*

H4b. *A high number of Web services to be shared between WSI servers per sharing will increase the amount of sharing between WSI servers.*

We will use Pearson correlation to test if there are significant and strong correlations between these variables to support or to refute the associated hypothesis.

8.5 Summary

In this chapter, we have established comparative tests that examine, under various conditions, the solution quality and computer times of the DAL heuristics against that of CPLEX 8. We also set up a series of tests to study the influence of certain parameters on the design characteristics of the model. These tests will be carried out on generated problems. We realized that it is not possible to test all combinations of parameters in our model but attempt instead to supply some representative and realistic simulations to discover general trends.

CHAPTER 9
EXPERIMENTAL RESULTS

9.1 Introduction

The goal of this chapter is to present the results of the experiments conducted to determine the efficacy of the DAL heuristics in solving the WSI location problems and the characteristics of the layout design determined by the model under different parameter values. Section 9.2 introduces a comparative analysis of the performance of the DAL heuristics and CPLEX 8. It also studies the performance of the DROP and ALA procedures of the DAL heuristics. Section 8.3 presents a study of the impact of various parameters on the characteristics of the designed location model. The control parameters consist of communication costs, WSI and WSP fixed costs, client demand and amount of Web services shared per sharing between WSI servers.

9.2 DAL Performance

9.2.1 Solution Quality

A summary of the comparative study of the solution quality is presented in Table 9-1. For problems with 25 or more location choices, CPLEX 8 with default searching technique, often ran out of memory before finding optimal solutions or took excessive amount of time to converge. We therefore only tested problems from class *a* to class *e* with CPLEX. For each class, Table 9-1 shows the average percentage gap and maximum percentage gap between the solutions of DAL and CPLEX. The solutions of the DAL heuristics averaged a gap of about 0.38% of the optimum values for 34 problems (Appendix C). The maximum gap of these tests was about 2.88%.

Table 9-1. Objective Function Value Gap between DAL and CPLEX.

Problem Classes / Number of Locations	Average Percentage Gap between DAL and CPLEX	Maximum Percentage Gap between DAL and CPLEX
a / 5	0%	0%
b / 10	0.17977%	0.87819%
c / 15	0.24945%	1.00007%
d / 20	1.11267%	2.88675%
e / 25 *	0.30752%	0.61505%
Weighed Average	0.38089%	N/A

* only instance e1 and e2 are included.

Since CPLEX did not give optimal solutions to problems e-3 to e-8 within test time, we compared the anytime performance of the DAL heuristics and CPLEX using these tests. For every one of these problems, CPLEX did not produce better solutions at the time when DAL terminated. CPLEX ran out of memory when solving two problems and its best solutions were worse than those of the DAL heuristics. We terminated CPLEX when solving four other problems after it ran for no less than 10 times of the time that DAL used. CPLEX gave two worse solutions and two better solutions compared with those of DAL heuristics.

9.2.2 General Computer Times

The computer times of the DAL heuristics were compared with those of CPLEX. A summary of the results are shown in Tables 9-2 and 9-3 and Figure 9-1. If a problem needed less than one second to solve, we register the computer time as one second. We only compared the computer times of problems from class *a* to *d*, since CPLEX used excessive amount of times and often ran out of memory for problems of class *e* and beyond.

Table 9-2. DAL Computer Times.

Problem Classes / Number of Locations	DAL Average Time (Sec)	DAL Relative Standard Deviation
a / 5	2	0%
b / 10	7	34.99271%
c / 15	32.625	52.22145%
d / 20	116.625	58.99587%
Average	N/A	36.55250%

Table 9-3. CPLEX 8 Computer Times.

Problem Classes / Number of Locations	CPLEX Average Time (Sec)	CPLEX Relative Standard Deviation
a / 5	1	0%
b / 10	1.25	37.03280%
c / 15	28.25	106.62785%
d / 20	1358.875	140.46981%
Average	N/A	71.03261%

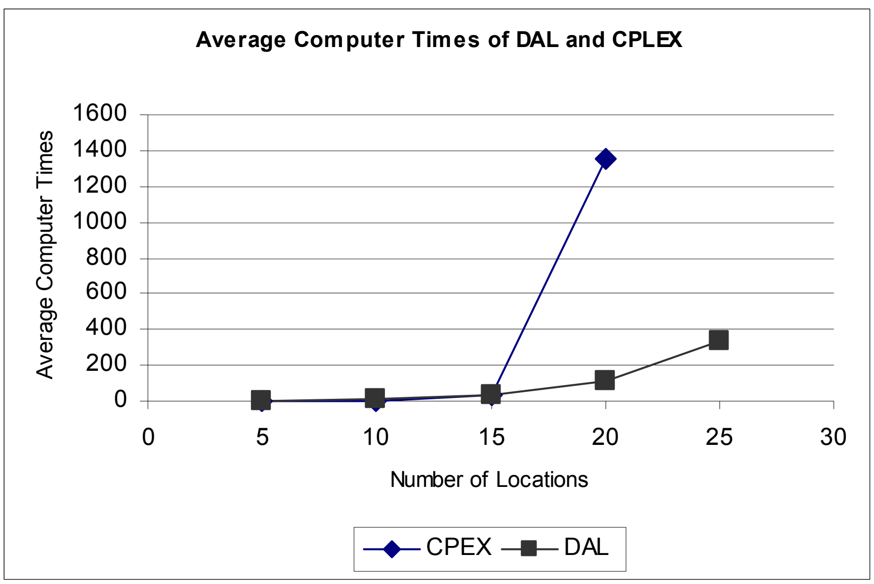

Figure 9-1. Average Computer Times of DAL and CPLEX.

For problems with fewer than 15 location choices, CPLEX 8 was faster than the DAL heuristics. However, for problems with 15 location choices or more, the computer times of CPLEX surpassed those of DAL. Figure 9-1 shows that, the computer times of CPLEX increased dramatically as the number of location choices continued to increase. It also shows the computer times of DAL, in contrast, increased at a slower rate. Table 9-2 and Table 9-3 show that for problems with 20 location choices, DAL averaged a computer time of 116.62 seconds while CPLEX averaged 1,358.87 seconds.

The results also demonstrate that on average the computer times of DAL varied much less across problems of similar size than CPLEX. We see in Table 9-2 and Table 9-3 the average relative standard deviation of DAL for the problems studied was about 36.55% while the one of CPLEX was 71.03%. Moreover, these tables show the relative standard deviations of computer times for DAL increased at a lower rate than those of CPLEX. This suggests the computer time of DAL is more predictable than CPLEX.

9.2.3 Computer Times by the Dimensions of Web Services Vectors

Tables 9-4 and 9-5 present a summary of 64 tests of DAL computer times decomposed into two categories by the dimensions of the Web services vectors. They show that the DAL heuristics were sensitive to the dimensions. In general, high dimensions complicated a problem and made it more difficult to solve.

Table 9-4 and 9-5 also illustrate the relative standard deviations of DAL computer times for problems of both dimensions were low. The average relative standard deviation was about 7.46% for two dimensions and 5.70% for eight dimensions.

Table 9-4. Computer Times with Two Dimensions of Web Services Vectors.

Problem Classes / Number of Locations	DAL Average Time	DAL Relative Standard Deviation
a / 5	2	0%
b / 10	4.75	10.52632%
c / 15	16.75	5.71598%
d / 20	52.5	10.37469%
e / 25	141.5	9.41990%
f / 30	292.25	6.62541%
g / 35	588.5	6.10543%
h / 40	1164.25	10.91707%
Average	N/A	7.46060%

Table 9-5. Computer Times with Eight Dimensions of Web Services Vectors.

Problem Classes / Number of Locations	DAL Average Time	DAL Relative Standard Deviation
a / 5	2	0%
b / 10	9.25	5.40540%
c / 15	48.5	4.29209%
d / 20	180.75	3.94776%
e / 25	536.75	6.72146%
f / 30	1329.5	7.29080%
g / 35	2846	13.67323%
h / 40	5520.5	4.24159%
Average	N/A	5.69655%

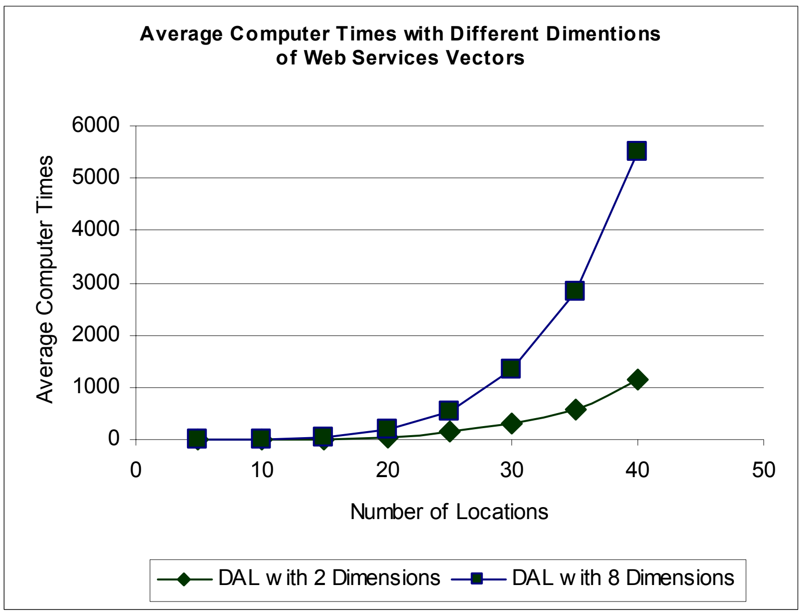

Figure 9-2. DAL Computer Times by Dimensions of Web Services Vectors.

9.2.4 ALA Performance

Table 9-6 summarizes the results of the comparative performance tests of the ALA procedure and DROP procedure. It shows the average solution quality improvement of ALA over DROP was about 0.51%. For one instance, the improvement was over 5%. The overall improvement was not insignificant considering the average gap between DAL solutions and optimal solutions was only about 0.38%. For some Web services projects, 0.51% of the total budget can be a worthwhile saving.

Table 9-6. Solution Quality Improvement of ALA over DROP.

Problem Classes / Number of Locations	Average Percentage Improvement of ALA over DROP	Maximum Percentage Improvement of ALA over DROP
a / 5	0.66107%	5.28862%
b / 10	0.00153%	0.01229%
c / 15	0.45566%	2.22578%
d / 20	0.16640%	1.33126%
e / 25	0.21911%	1.10705%
f / 30	0.52361%	3.00357%
g / 35	0.84189%	2.29438%
h / 40	1.20944%	4.23413%
Average	0.50984%	N/A

Table 9-7 and Figure 9-3 show that the average computer times of the ALA procedure was about 48.63% of those of the DROP procedure. Considering the improvement of solution quality it makes, ALA is a justified part of the DAL heuristics.

Table 9-7. Average DROP and ALA Computer Times.

Number of Locations	DROP Computer Time	ALA Computer Time	Average Percentage Computer Time of ALA over DROP
a / 5	1	1	100%
b / 10	4.875	2.125	43.5897%
c / 15	23	9.625	41.8478%
d / 20	84.5	32.125	38.0178%
e / 25	242.625	96.5	39.7733%
f / 30	569.375	241.5	42.4149%
g / 35	1200	517.25	43.1042%
h / 40	2383	959.375	40.2591%
Average	N/A	N/A	48.6259%

9.2.4 Test Results with Different Fixed Costs between WSI and WSP

The DAL heuristics treat WSI server locations and WSP locations equally when deciding which one to drop or add to minimize operating costs. In the previous tests, the

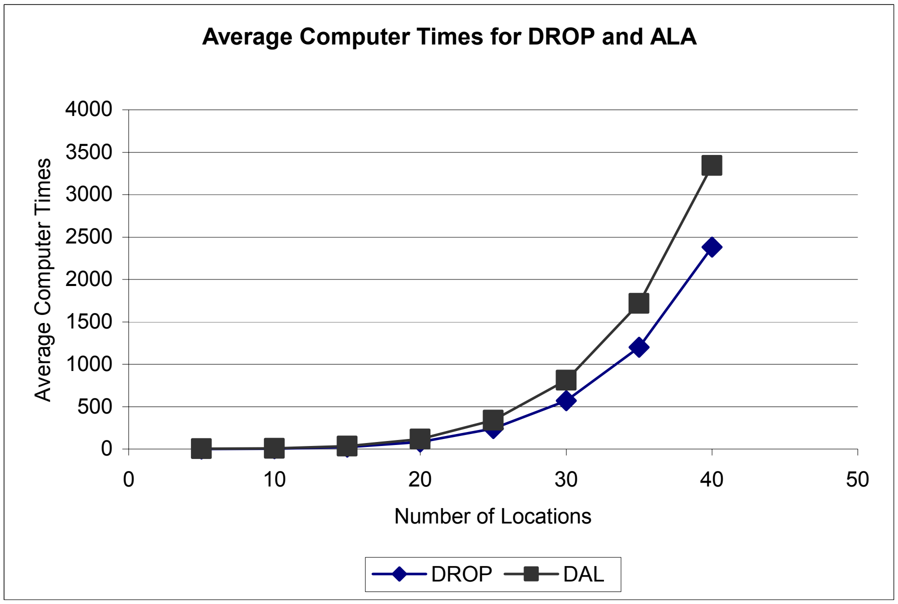

Figure 9-3. Average Computer Times for DROP, ALA and DAL Procedures (ALA Computer Times Are the Gap between the Times of DROP and DAL).

same values were assigned to the fixed costs of both WSI server locations and WSP locations. To confirm that the differences in fixed costs do not influence the performance of the DAL heuristics, we ran four more tests (i1, i2, i3 and i4 in Appendix C) with the fixed costs of WSI server locations differing from those of WSPs. These problems consisted of 20 random locations for the WSI, his clients and WSPs with two Web services. The parameters of the WSI and WSP fixed costs took different values of $6,000, $4000, $2000 or $1,000. Table 9-8 shows the results of these tests. For the problems tested, the solution gaps were no worse than those of the problems of similar size and with the same fixed costs for the WSI and the WSPs. For example, the average percentage gap between DAL and CPLEX in these tests was about 0.156% while the gap for the problems of similar size with same fixed costs (instance c1, c3, c5 and c7) was about 0.332%.

Table 9-8. Test Results of Problems with Different WSI and WSP Location Costs.

Problem Number	WSP Fixed Cost ($)	WSI Fixed Cost ($)	Average Percentage Gap between DAL and CPLEX
f-1	1000	6000	0 %
f-2	2000	4000	0.23818%
f-3	4000	2000	0.09728%
f-4	6000	1000	0.29188%
Average	N/A	N/A	0.15684%

9.2.5 Robustness of DAL

The DAL heuristics showed robustness across a wide range of problems tested. It obtained optimal or near optimal solutions for all the problems that were solved by CPLEX. While the solution times using DAL increased when the size of the problems increased, they did not vary much for problems of similar size. The low variance in computer times was particularly obvious when the test problems were categorized by the dimensions of their Web services vectors.

9.3 Hypotheses Testing

To study the impact of certain parameters on the solutions, we tested a series of problems (Appendix D) using CPLEX 8. From previous comparative studies in Section 9.2, we learned that using CPLEX to solve WSI location problems of small size is practical. More importantly, CPLEX provides optimal solutions and voids any influence of imperfections as a result of using DAL to solve the problems. We conducted 190 tests. For each parameter, there were either 45 or 50 tests with 9 or 10 different values assigned to the parameter. The hypotheses were tested using Pearson correlation. The results, in general, supported the hypotheses.

9.3.1 Hypothesis 1: Communication/Latency Cost

Our initial testing of the model concentrated on the influence of the communication/latency costs on the resulting design characteristics. Figure 9-4 shows that, for the problems tested, the latency costs were positively related to the number of WSI servers and WSPs.

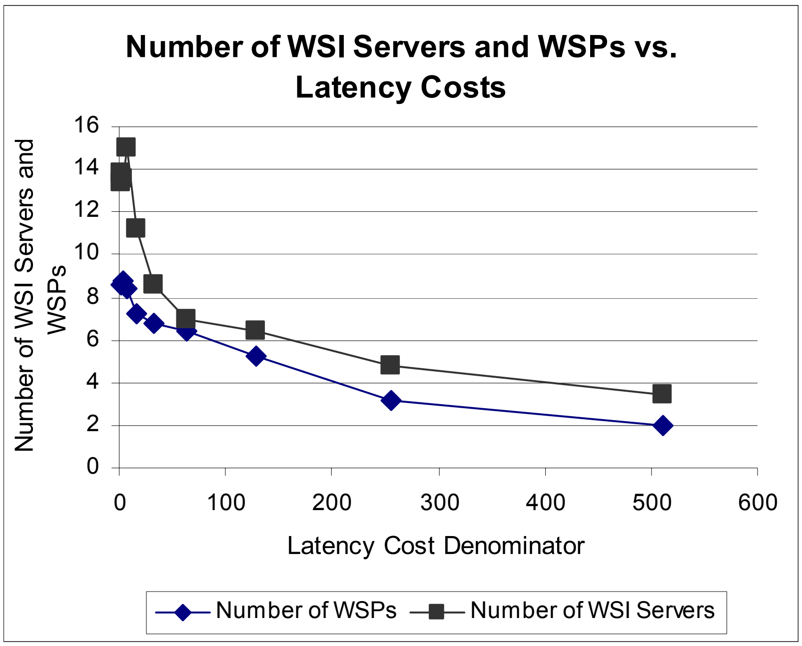

Figure 9-4. The Number of WSI Servers and WSPs for Each Latency Costs Decrement (Higher Values of the Denominator Mean Lower Latency Costs).

Hypothesis 1 states that high latency costs and a large number of time-sensitive clients will increase the number of WSI servers and WSPs. As showed in Table 9-9, this hypothesis was not refuted. The correlations between the denominators for communication costs and the number of WSI servers and WSPs were -.823 and -.657 respectively and were significant at the 0.01 level. This finding suggests that, facing clients who were sensitive to service delays, a WSI would alleviate the impact of the increased network latency costs by setting up more servers that were near to his clients and by contacting more WSPs near to his servers.

Table 9-9. Pearson Correlations: The Impact of Network Latency Cost on the Number of WSI Servers and WSPs.

		Number of WSPs	Number of WSI Servers
Denominator for Communication Cost	Pearson Correlation	-.823(**)	-.657(**)
	Sig. (2-tailed)	.000	.000
	N	50	50

** Correlation is significant at the 0.01 level (2-tailed).

9.3.2 Hypothesis 2: WSI Server and WSP Fixed Costs

We then conducted tests to determine whether an increase in the fixed costs of WSI server and WSPs would decrease the number of WSI servers and WSPs. Figure 9-5 supports this supposed negative relationship.

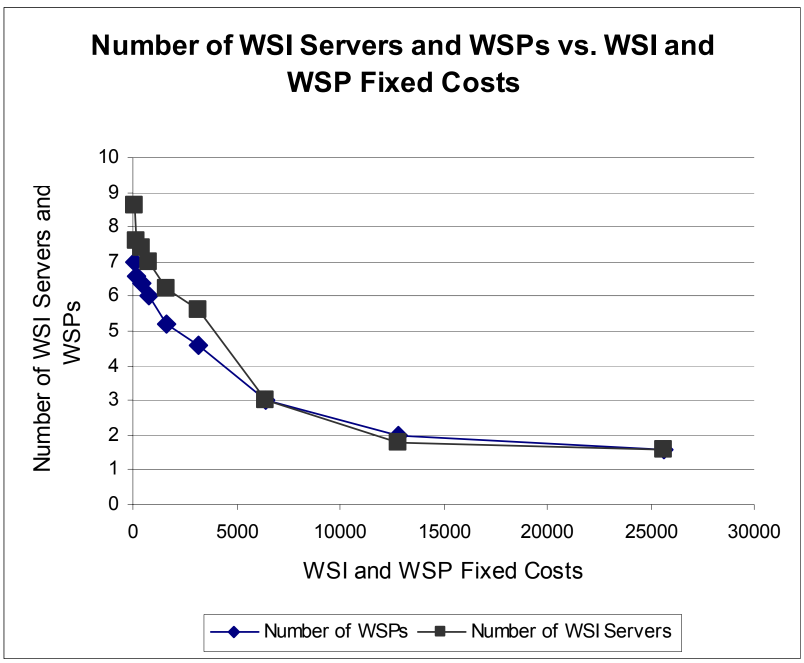

Figure 9-5. The Number of WSI Servers and WSPs for Each WSP and WSI Fixed Costs Increment.

Hypothesis 2 states that high WSI and WSP fixed costs will decrease the number of WSI servers and WSPs. Table 9-10 shows this was supported. The correlations between the WSI and WSP fixed costs and the number of WSI servers and WSPs were -.784 and -.693 respectively and were significant at the 0.01 level.

Table 9-10. Pearson Correlations: The Impact of WSI and WSP Fixed Cost on the Number of WSI Servers and WSPs.

		Number of WSPs	Number of WSI Servers
WSI and WSP Fixed Cost	Pearson Correlation	-.784(**)	-.693(**)
	Sig. (2-tailed)	.000	.000
	N	45	45

** Correlation is significant at the 0.01 level (2-tailed).

9.3.3 Hypothesis 3: Client Demand

Also of interest was whether an increase in client demand would change the optimal strategies of a WSI. Figures 9-6 and 9-7 indicate that client demand was positively related to both the number of WSI servers and WSPs and the sharing between WSI servers. The number of locations leveled off after the demand reached 6400. Since we set the capacities of WSI and WSP almost unlimited, the number of WSI servers and WSPs was not affected by their capacities. Figure 9-7 shows a near perfect linear relationship between the amount of client demand and the amount of sharing between WSI servers. This suggests an increase in the amount of sharing was proportional to an increase in the client demand.

Hypothesis 3a states that high client demand will increase the number of WSI servers and WSPs. As indicated in Table 9-11, this hypothesis was not refuted. The correlations between client demand and the number of WSI servers and WSPs were -.546 and -.779 respectively and were significant at the 0.01 level. An increase in client demand increased the overall communication costs and rendered the fixed costs less consequential. Apparently the WSI took advantage of these “decreased” fixed costs and expanded his server locations and their contacts with WSPs.

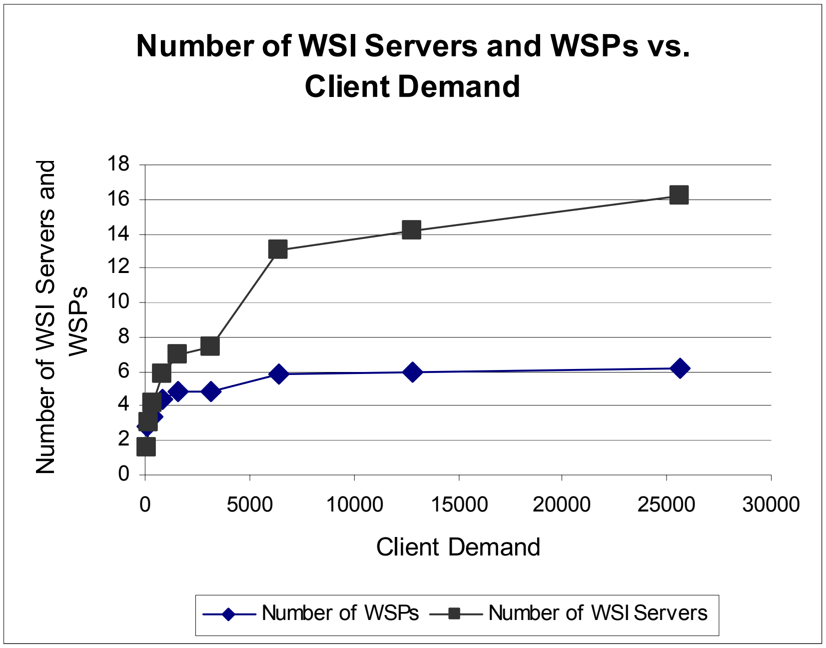

Figure 9-6. The Number of WSI Servers and WSPs for Each Client Demand Increment.

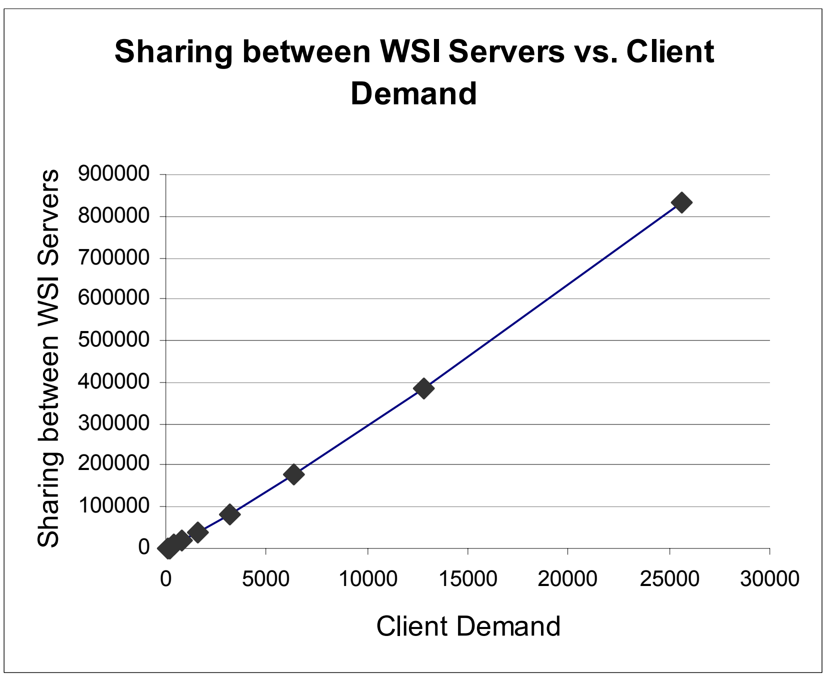

Figure 9-7. Sharing between WSI Servers for Each Client Demand Increment.

Hypothesis 3b states that high client demand will increase the amount of sharing between WSI servers proportionally. This too was not refuted. Table 9-11 shows the correlation between client demand and the amount of sharing between WSI servers was .526 and was significant at the 0.01 level.

Table 9-11. Pearson Correlations: The Impact of Client Demand on the Number of WSI Servers and WSPs.

		Number of WSPs	Number of WSI Servers	Sharing between WSI
Client Demand	Pearson Correlation	.546(**)	.779(**)	.526(**)
	Sig. (2-tailed)	.000	.000	.000
	N	45	45	45

** Correlation is significant at the 0.01 level (2-tailed).

9.3.4 Hypothesis 4: Number of Web Services Shared per Sharing

In our mathematical model, we allow one Web service to be shared between WSI servers per sharing transaction. Since sharing cost is a function of network latency rather than the number of Web services being shared per sharing, sharing multiple Web services at a time will lower the overall sharing costs. Figure 9-8 shows that the number of WSI servers increased gradually with an increase in the number of Web services shared per sharing. Figure 9-9 suggests that sharing between WSI servers increased dramatically as the number of Web services being shared per sharing increased.

Hypothesis 4a states that a high number of Web services to be shared between WSI servers per sharing will increase the number of WSI servers. As indicated in Table 9-12, this hypothesis was not refuted. The correlation between the number of Web services shared per sharing and the number of WSI servers was .525 and was significant at the 0.01 level. We suspect that the WSI set up more servers to take advantage this lowered sharing costs.

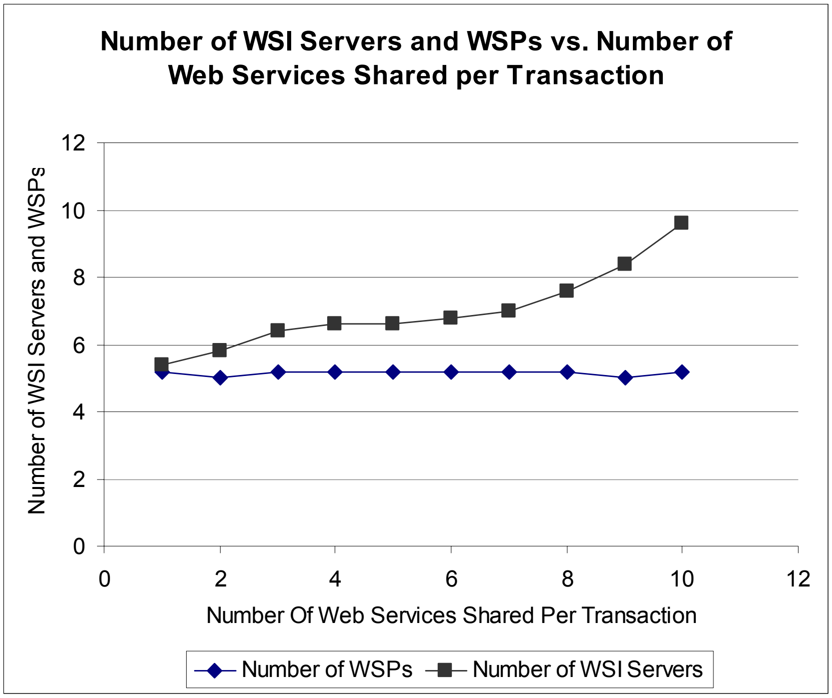

Figure 9-8. The Number of WSI Servers and WSPs for Each Number of Web Services Shared per Sharing Increment.

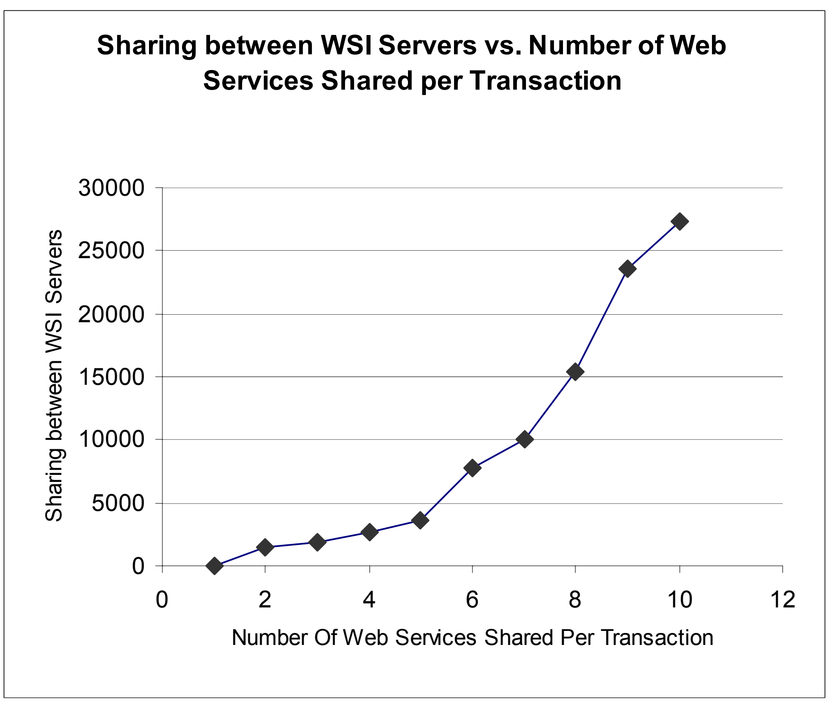

Figure 9-9. Sharing between WSI Servers for Each Number of Web Services Shared per Sharing Increment

Hypothesis 4b states that a high number of Web services to be shared between WSI servers per sharing will increase the amount of sharing between WSI servers. This too was not refuted. Table 9-11 shows the correlation was .569 and was significant at the 0.01 level.

Table 9-12. Pearson Correlations: The Impact of Number of Web Services Shared per Transaction on the Number of WSI Servers and WSPs.

		Number of WSPs	Number of WSI Servers
Web Services Shared per Sharing	Pearson Correlation	.525(**)	.569(**)
	Sig. (2-tailed)	.000	.000
	N	50	50

** Correlation is significant at the 0.01 level (2-tailed).

9.3.5 Number of WSI Servers and WSPs

The experiments show that when WSI server and WSP fixed costs are the same, a WSI often set up more servers than contacting more WSPs. This is probably because WSI server locations were more effective in lowering the objective function values. The locations of WSI servers directly affected both the latencies between WSI servers and WSPs and the latencies between WSI servers and their clients. In contrast, the locations of WSPs only directly affect the latencies between WSPs and WSI servers.

9.4 Summary

The results of the computational experiments recommended the DAL heuristics for solving WSI location problems. We feel the heuristics offered good solutions to the problems tested in reasonable times while being conservative on memory usage. In many practical applications of Web services, the solution time and quality of the DAL heuristics should be satisfactory or at least provide guidance or initial feasible solutions.

CHAPTER 10
CONCLUSION

10.1 Project Overview and Contribution

In this dissertation, we first defined a Web services intermediary location problem and presented a mathematical model to describe it. We also developed an efficient heuristic method named DAL to find good solutions in reasonable execution times. For the problems tested, the DAL heuristics provided near optimal solutions in short computer times and with limited computer memory.

Web services technology is a newly developed middle-tier platform. Armed with a common interface, Internet protocols, and distributed component technology, it promises to transform the current standard practice of business transactions. However, infrastructure problems need to be solved before businesses accept Web services. The WSI location problem with network latencies is one of them. This problem is pronounced for business applications facing time-sensitive clients.

A WSI can contact various service providers at run time and compile multiple Web services into a coherent presentation to his clients. In addition, services offered by Web services are information goods and have special cost structure. Our location model captures these features and therefore differs from traditional facility location models.

Because of the size and complexity of the problem, general-purpose linear/integer programming algorithms do not perform well. As a result, DAL heuristics were developed to undertake the problem. We conducted a series of tests to study the efficacy of the DAL heuristics compared with CPLEX. Test results indicated the DAL heuristics were competitive. Even though they did not converge to the optimal solutions for all problems tested, the DAL heuristics obtained solutions no further than 0.39% from the optimal ones on average. In addition, DAL computer times compared favorably with the times used by CPLEX 8 as the size of the problems increased. This is partially because that CPLEX is a general purpose program while the DAL heuristics were written specifically for the WSI location problem.

The DAL heuristics consist of two different procedures, DROP and ALA. The ALA capitalizes on the solutions provided by DROP. Test results show that ALA obtained improvements on the solutions of the DROP procedure at a modest cost of computer times. We feel the extra time ALA took is offset by improvements in the objective function value it made. In particular, ALA provided a 0.51% improvement in the final solution on average.

One of the major advantages of using DAL is evident when a previously solved problem has changes to the constraints on the locations of WSI and WSP and needs to be revolved. The DAL heuristics can be used to find good solutions quickly. For example, if WSI needs to set up one more server, or a selected WSP no longer provide services, the DAL heuristics, without resolving the entire problem, can build on the previous solution to the original problem and allow the DROP procedure to compute the best location to drop or to add. This is particularly advantageous in rebalancing or fine-tuning decisions when needed.

Another advantage of practical importance is that the DAL heuristics use less computer memory when compared with CPLEX 8 using “Best-bound search” technique.

This allows DAL to be implemented in small memory capacity computers in which CPLEX may run out of memory for the same problems.

A series of simulated tests were conducted to discover the characteristics of the WSI location model. From the experiment, we were able to extract patterns that are instrumental in learning a WSI's planning to minimize operating costs.

10.2 Limitations

There are important limitations in our investigation of WSI location problems. First, to study the efficacy of the DAL heuristics and the characteristics of the model, we used simulated data because of the scarcity of real life data. Despite our best effort to generate reasonable and representative data, we acknowledge the study may benefit from using real life data.

Second, to make the model manageable, we have certain assumptions that may or may not be universally true. For example, we assumed the cost of transporting information goods was independent of the quantities being moved. We also assumed that clients chose not to contact WSPs directly for services. In future research, some of these assumptions may be relaxed and revisited.

10.3 Direction for Future Research

This dissertation is a preliminary study of WSI location problems. Web services as a technology and WSI location problems are still in their infancy and, thus, a great deal remains for further development and refinement of their modeling and solution methodologies.

Our model minimizes the operating costs of a WSI. It assumes the buyers' market is fully covered and all client demands are met. A natural extension to this model is one that studies a monopoly market in which a WSI seeks profit maximization by deciding which client to serve and how much to serve. In this new model, clients' subscription fees could be parameterized. In addition, such a model can be extended to explore the strategic server placement of multiple competing WSIs, their pricing structures and their strategic relationships with one another, with their clients and with WSPs in a completive market.

The current model was designed for one-period planning of a WSI with the assumption that the values of all parameters would stay constant in future periods. However, this assumption can be easily violated by the dynamics of a customer base, the number of and offerings by WSPs and the properties of WSI server locations. New models are needed to assist the WSI to review and rebalance previous decisions to reflect changes.

In addition, our model uses deterministic parameters. In practice many parameters, including client demand, cost, and availability of servers, may be uncertain. A stochastic location model is necessary to study such problems.

One of the constraints of this research is the lack of real life data for experiments. Part of the reason is that Web services have not been widely implemented yet. An empirical study using data from real applications, when available, will supplement our understanding of the DAL heuristics and the characteristic of the WSI location model

APPENDIX A
CITY LISTS

Code	State	City		Code	State	City
1	NM	Albuquerque		21	CA	Los Angeles
2	GA	Atlanta		22	FL	Miami
3	TX	Austin		23	MN	Minneapolis
4	MD	Baltimore		24	TN	Nashville
5	MA	Boston		25	LA	New Orleans
6	NY	Buffalo		26	NY	New York
7	NC	Charlotte		27	FL	Orlando
8	IL	Chicago		28	PA	Philadelphia
9	OH	Cincinnati		29	AZ	Phoenix
10	OH	Cleveland		30	PA	Pittsburg
11	OH	Columbus		31	OR	Portland
12	TX	Dallas		32	NC	Raleigh
13	CO	Denver		33	VA	Richmond
14	MI	Detroit		34	UT	Salt Lake City
15	CT	Hartford		35	CA	San Diego
16	TX	Houston		36	CA	San Francisco
17	IN	Indianapolis		37	WA	Seattle
18	FL	Jacksonville		38	MO	St. Louis
19	KS	Kansas City		39	FL	Tampa
20	NV	Las Vegas		40	AZ	Tucson

APPENDIX B
CITY DISTANCE MATRIX

City Code	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
1	0	1270	615	0	0	0	0	0	0	0	0	588	0	0	0	0	0	0	716	483
2	1270	0	0	0	0	0	226	0	0	0	0	717	0	0	0	701	426	285	0	0
3	615	0	0	0	0	0	0	0	0	0	0	184	0	0	0	144	0	0	0	0
4	0	0	0	0	0	0	364	0	0	0	342	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	95.3	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	172	0	0	0	0	324	0	0	0	0	0
7	0	226	0	364	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	0	313	0	0	913	240	0	0	164	0	409	0
9	0	0	0	0	0	0	0	0	0	0	101	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	172	0	313	0	0	125	0	0	93.9	0	0	0	0	0	0
11	0	0	0	342	0	0	0	0	101	125	0	0	0	163	0	0	167	0	0	0
12	588	717	184	0	0	0	0	0	0	0	0	0	662	0	0	222	0	0	450	0
13	0	0	0	0	0	0	0	913	0	0	0	662	0	0	0	0	0	0	553	606
14	0	0	0	0	0	0	0	240	0	93.9	163	0	0	0	0	0	238	0	0	0
15	0	0	0	0	95.3	324	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	701	144	0	0	0	0	0	0	0	0	222	0	0	0	0	0	0	0	0
17	0	426	0	0	0	0	0	164	0	0	167	0	0	238	0	0	0	0	0	0
18	0	285	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
19	716	0	0	0	0	0	0	409	0	0	0	450	553	0	0	0	0	0	0	0
20	483	0	0	0	0	0	0	0	0	0	0	0	606	0	0	0	0	0	0	0

City Code	1	2	3	4	05	06	7	8	9	10	11	12	13	14	15	16	17	18	19	20
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	236
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	969	0	0	0	0
23	0	0	0	0	0	0	0	352	0	0	0	0	698	0	0	0	0	0	413	0
24	0	215	0	595	0	0	0	0	236	0	0	613	0	0	0	0	0	0	0	0
25	0	422	0	0	0	0	0	819	0	0	0	0	0	0	0	320	0	0	0	0
26	0	0	0	0	0	0	0	0	0	405	0	0	0	0	98.5	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	123	0	0
28	0	0	0	93.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	329	0	871	0	0	0	0	0	0	0	0	890	582	0	0	0	0	0	0	250
30	0	0	0	195	0	0	0	415	0	114	162	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	129	0	0	0	0	0	0	0	0	0	0	414	0	0
33	0	0	0	129	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	1250	0	0	0	0	371	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	946	0	0	0	0	0	0	415
37	0	0	0	0	0	0	0	0	0	0	0	0	1020	0	0	0	0	0	1500	0
38	0	0	0	0	0	0	0	257	0	0	0	544	0	0	0	0	232	0	240	0
39	0	418	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	172	0	0
40	0	0	0	0	0	0	0	0	0	0	0	829	0	0	0	0	0	0	0	0

City Code	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
1	0	0	0	0	0	0	0	0	329	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	215	422	0	0	0	0	0	0	0	0	0	0	0	0	0	418	0
3	0	0	0	0	0	0	0	0	871	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	595	0	0	0	93.5	0	195	0	0	129	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0	0	0	0	129	0	0	0	0	0	0	0	0
8	0	0	352	0	819	0	0	0	0	415	0	0	0	1250	0	0	0	257	0	0
9	0	0	0	236	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	405	0	0	0	114	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	162	0	0	0	0	0	0	0	0	0	0
12	0	0	0	613	0	0	0	0	890	0	0	0	0	0	0	0	0	544	0	829
13	0	0	698	0	0	0	0	0	582	0	0	0	0	371	0	946	1020	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	98.5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	969	0	0	320	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	232	0	0
18	0	0	0	0	0	0	123	0	0	0	0	414	0	0	0	0	0	0	172	0
19	0	0	413	0	0	0	0	0	0	0	0	0	0	0	0	0	1500	240	0	0
20	236	0	0	0	0	0	0	0	250	0	0	0	0	0	0	415	0	0	0	0

City Code	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
21	0	0	0	0	0	0	0	0	363	0	0	0	0	0	112	344	0	0	0	0
22	0	0	0	0	668	0	203	0	0	0	0	0	0	0	0	0	0	0	203	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	985	0	0	1390	466	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	252	0	0
25	0	668	0	0	0	0	532	0	0	0	0	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	77.9	0	315	0	0	0	0	0	0	0	0	0	0
27	0	203	0	0	532	0	0	0	0	0	0	0	0	0	0	0	0	0	78.5	0
28	0	0	0	0	0	77.9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
29	363	0	0	0	0	0	0	0	0	0	0	0	0	0	291	0	0	0	0	112
30	0	0	0	0	0	315	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	533	144	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	138	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	138	0	0	0	0	0	0	0	0
34	0	0	985	0	0	0	0	0	0	0	0	0	0	0	0	597	701	0	0	0
35	112	0	0	0	0	0	0	0	291	0	0	0	0	0	0	0	0	0	0	0
36	344	0	0	0	0	0	0	0	0	0	533	0	0	597	0	0	0	0	0	0
37	0	0	1390	0	0	0	0	0	0	0	144	0	0	701	0	0	0	0	0	0
38	0	0	466	252	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	203	0	0	0	0	78.5	0	0	0	0	0	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0	112	0	0	0	0	0	0	0	0	0	0	0

APPENDIX C
EXPERIMENTAL STUDY RESULTS FOR EFFICACY STUDY

Instance a-1: 5 locations

WSP location	14, 20, 22, 34, 26
WSI location	17, 9, 7, 13, 23
Client location	21, 29, 23, 33, 7
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$56234	$56234
Percent Above Optimal Solution	0.0%	N/A
Run Time	<2 Second	<1 Second
WSP location	22, 26	22, 26
WSI location	17, 9, 7, 23	17, 9, 7, 23

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$56234	$56234
Percent Above DROP Solution	N/A	0.0%
Run Time	1 Seconds	<1 Second
WSP location	22, 26	22, 26
WSI location	17, 9, 7, 23	17, 9, 7, 23

Instance a-2: 5 locations

WSP location	22, 26, 1, 7, 12
WSI location	30, 21, 14, 20, 1
Client location	33, 9, 38, 35, 15
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 90610	$ 90610
Percent Above Optimal Solution	0.0%	N/A
Run Time	<2 Second	<1 Second
WSP location	1, 12	1, 12
WSI location	20, 1	20, 1

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 90610	$ 90610
Percent Above DROP Solution	N/A	0.0%
Run Time	1 Seconds	<1 Second
WSP location	1, 12	26, 1, 7
WSI location	20, 1	21, 14, 20

Instance a-3: 5 locations

WSP location	33, 4, 26, 25, 38
WSI location	9, 10, 11, 29, 3
Client location	22, 35, 12, 18, 5
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 24922	$ 24922
Percent Above Optimal Solution	0.0%	N/A
Run Time	1 Second	<1 Second
WSP location	26, 25	26, 25
WSI location	9, 11	9, 11

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 25133	$ 24922
Percent Above DROP Solution	N/A	0.0%
Run Time	1 Seconds	1 Second
WSP location	4, 26	26, 25
WSI location	9, 11	9, 11

Instance a-4: 5 locations

WSP location	16, 34, 28, 17, 4
WSI location	19, 15, 13, 32, 39
Client location	22, 3, 29, 7, 25
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 13280	$ 13280
Percent Above Optimal Solution	0.0%	N/A
Run Time	1 Second	<1 Second
WSP location	16	16
WSI location	13	13

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 13280	$ 13280
Percent Above DROP Solution	N/A	0.0%
Run Time	1 Seconds	<1 Second
WSP location	16	16
WSI location	13	13

Instance a-5: 5 locations

WSP location	35, 36, 31, 22, 0
WSI location	39, 25, 15, 37, 31
Client location	5, 34, 8, 28, 20
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$78809	$78809
Percent Above Optimal Solution	0.0%	N/A
Run Time	1 Second	<1 Second
WSP location	22	22
WSI location	39, 37	39, 37

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$78809	$78809
Percent Above DROP Solution	N/A	0.0%
Run Time	1 Seconds	1 Second
WSP location	22	22
WSI location	39, 37	39, 37

Instance a-6: 5 locations

WSP location	12, 34, 1, 30, 35
WSI location	18, 11, 1, 20, 17
Client location	25, 20, 2, 19, 38
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$73809	$73809
Percent Above Optimal Solution	0.0%	N/A
Run Time	<2 Second	<1 Second
WSP location	34	34
WSI location	11, 20	11, 20

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$73809	$73809
Percent Above DROP Solution	N/A	0.0%
Run Time	1 Seconds	<1 Second
WSP location	34, 1	34
WSI location	1, 20	11, 20

Instance a-7: 5 locations

WSP location	22, 23, 11, 7, 38
WSI location	25, 11, 35, 22, 32
Client location	31, 20, 35, 21, 13
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$31010	$31010
Percent Above Optimal Solution	0.0%	N/A
Run Time	1 Second	<1 Second
WSP location	34	34
WSI location	11, 20	11, 20

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$32650	$31010
Percent Above DROP Solution	N/A	5.28862%
Run Time	1 Seconds	<1 Second
WSP location	12	34
WSI location	20, 17	11, 20

Instance a-8: 5 locations

WSP location	27, 28, 8, 34, 5
WSI location	34, 37, 8, 24, 26
Client location	4, 34, 25, 38, 2
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$37030	$37030
Percent Above Optimal Solution	0.0%	N/A
Run Time	1 Second	<1 Second
WSP location	8, 34	8, 34
WSI location	34, 8	34, 8

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$37030	$37030
Percent Above DROP Solution	N/A	0.0%
Run Time	1 Seconds	<1 Second
WSP location	8, 34	8, 34
WSI location	34, 8	34, 8

Instance b-1: 10 locations

WSP location	34, 13, 10, 2, 24, 26, 6, 19, 37, 32
WSI location	37, 1, 26, 25, 21, 31, 18, 39, 10, 8
Client location	0, 21, 10, 16, 37, 32, 34, 20, 19, 23
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 66000	$ 66000
Percent Above Optimal Solution	0.0%	N/A
Run Time	5 Seconds	<1 Second
WSP location	34, 10, 6, 37	34, 10, 6, 37
WSI location	37, 21, 31, 39, 10	37, 21, 31, 39, 10

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 66000	$ 66000
Percent Above DROP Solution	N/A	0.0%
Run Time	3 Seconds	2 Second
WSP location	34, 10, 6, 37	34, 10, 6, 37
WSI location	37, 21, 31, 39, 10	37, 21, 31, 39, 10

Instance b-2: 10 locations

WSP location	25, 10, 35, 15, 37, 1, 20, 8, 16, 2
WSI location	28, 38, 19, 32, 8, 12, 0, 35, 14, 34
Client location	35, 7, 19, 29, 13, 21, 34, 15, 38, 2
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 94793	$ 94793
Percent Above Optimal Solution	0.0%	N/A
Run Time	9 Seconds	<1 Second
WSP location	35, 1, 20, 8	35, 1, 20, 8
WSI location	38, 19, 8, 35, 34	38, 19, 8, 35, 34

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 94793	$ 94793
Percent Above DROP Solution	N/A	0.0%
Run Time	6 Seconds	3 Seconds
WSP location	35, 1, 20, 8	35, 1, 20, 8
WSI location	38, 19, 8, 35, 34	38, 19, 8, 35, 34

Instance b-3: 10 locations

WSP location	1, 34, 12, 15, 20, 5, 13, 7, 3, 38
WSI location	8, 10, 13, 6, 1, 18, 38, 32, 27, 22
Client location	14, 19, 13, 5, 30, 24, 15, 18, 33, 35
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 40664	$ 40310
Percent Above Optimal Solution	0.878194%	N/A
Run Time	5 Seconds	<1 Second
WSP location	7, 3	7
WSI location	10, 6, 18, 27	10, 18, 22

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 40669	$ 40664
Percent Above DROP Solution	N/A	0.0122959%
Run Time	3 Seconds	2 Second
WSP location	7, 3	7, 3
WSI location	10, 18, 27, 22	10, 6, 18, 27

Instance b-4: 10 locations

WSP location	39, 24, 11, 13, 23, 17, 27, 4, 32, 25
WSI location	5, 1, 11, 12, 22, 30, 7, 9, 18, 33
Client location	12, 10, 2, 33, 36, 29, 23, 22, 1, 19
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 29625	$ 29625
Percent Above Optimal Solution	0.0%	N/A
Run Time	10 Seconds	<1 Seconds
WSP location	11, 13, 23	11, 13, 23
WSI location	1, 11, 12, 9	1, 11, 12, 9

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 29625	$ 29625
Percent Above DROP Solution	N/A	0.0%
Run Time	7 Seconds	3 Seconds
WSP location	11, 13, 23	11, 13, 23
WSI location	1, 11, 12, 9	1, 11, 12, 9

Instance b-5: 10 locations

WSP location	1, 17, 31, 22, 23, 10, 36, 26, 9, 2
WSI location	4, 37, 15, 36, 30, 2, 29, 18, 3, 21
Client location	11, 14, 15, 27, 25, 36, 33, 35, 5, 21
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$102350	$101780
Percent Above Optimal Solution	0.560031%	N/A
Run Time	5 Seconds	<1 Second
WSP location	36, 9, 2	31, 36, 2
WSI location	36, 2, 29, 21	36, 2, 3, 21

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$102350	$102350
Percent Above DROP Solution	N/A	0.0%
Run Time	3 Seconds	2 Second
WSP location	36, 9, 2	36, 9, 2
WSI location	36, 2, 29, 21	36, 2, 29, 21

Instance b-6: 10 locations

WSP location	8, 1, 33, 30, 31, 4, 3, 19, 10, 12
WSI location	11, 22, 17, 5, 25, 35, 12, 33, 24, 6
Client location	18, 39, 17, 4, 7, 24, 20, 38, 8, 13
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$145344	$145344
Percent Above Optimal Solution	0.0%	N/A
Run Time	9 Seconds	1 Seconds
WSP location	1, 10, 12	1, 10, 12
WSI location	17, 5, 12, 24	17, 5, 12, 24

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$145344	$145344
Percent Above DROP Solution	N/A	0.0%
Run Time	7 Seconds	2 Seconds
WSP location	1, 10, 12	1, 10, 12
WSI location	17, 5, 12, 24	17, 5, 12, 24

Instance b-7: 10 locations

WSP location	36, 21, 17, 12, 31, 38, 32, 28, 33, 18,
WSI location	3, 30, 18, 10, 20, 12, 6, 7, 17, 26,
Client location	10, 7, 18, 1, 17, 31, 32, 22, 4, 35,
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$39910	$39910
Percent Above Optimal Solution	0.0%	N/A
Run Time	4 Seconds	1 Second
WSP location	31, 18	31, 18
WSI location	18, 6	18, 6

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$39910	$39910
Percent Above DROP Solution	N/A	0.0%
Run Time	3 Seconds	1 Second
WSP location	31, 18	31, 18
WSI location	18, 6	18, 6

Instance b-8: 10 locations

WSP location	17, 2, 26, 1, 8, 13, 3, 15, 24, 19
WSI location	27, 39, 2, 15, 23, 33, 21, 24, 37, 8
Client location	33, 8, 3, 14, 5, 7, 26, 0, 11, 39
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$45510	$45510
Percent Above Optimal Solution	0.0%	N/A
Run Time	9 Seconds	2 Second
WSP location	8, 15	8, 15
WSI location	2, 8	2, 8

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$45510	$45510
Percent Above DROP Solution	N/A	0.0%
Run Time	7 Seconds	2 Seconds
WSP location	8, 15	8, 15
WSI location	2, 8	2, 8

Instance c-1: 15 locations

WSP location	20, 39, 16, 14, 29, 34, 13, 6, 4, 36, 30, 19, 23, 7, 9
WSI location	37, 13, 15, 20, 11, 14, 31, 38, 27, 33, 24, 16, 18, 17, 5
Client location	3, 22, 16, 11, 32, 20, 24, 2, 25, 9, 8, 21, 19, 10, 34
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 77240	$ 77240
Percent Above Optimal Solution	0.0%	N/A
Run Time	17 Seconds	1 Second
WSP location	27, 38, 5	27, 38, 5
WSI location	13, 20, 11, 14, 31, 38, 24, 16	13, 20, 11, 14, 31, 38, 24, 16

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 77240	$ 77240
Percent Above DROP Solution	N/A	0.0%
Run Time	11 Seconds	6 Second
WSP location	27, 38, 5	27, 38, 5
WSI location	13, 20, 11, 14, 31, 38, 24, 16	13, 20, 11, 14, 31, 38, 24, 16

Instance c-2: 15 locations

WSP location	28, 14, 11, 5, 36, 18, 9, 33, 8, 17, 29, 32, 12, 34, 35
WSI location	32, 3, 27, 21, 30, 25, 2, 7, 17, 28, 23, 12, 24, 31, 4
Client location	38, 11, 28, 19, 26, 32, 12, 7, 29, 34, 33, 15, 13, 5, 23
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 88369	$ 87494
Percent Above Optimal Solution	1.00007%	N/A
Run Time	46 Seconds	2 Second
WSP location	28, 14, 11, 9, 8, 17, 32, 12	28, 11, 9, 8, 17, 32, 12
WSI location	32, 3, 2, 7, 17, 28, 23, 12, 4	32, 3, 2, 7, 17, 28, 23, 12

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 88369	$ 88369
Percent Above DROP Solution	N/A	0.0%
Run Time	29 Seconds	17 Seconds
WSP location	28, 14, 11, 9, 8, 17, 32, 12	28, 14, 11, 9, 8, 17, 32, 12
WSI location	32, 3, 2, 7, 17, 28, 23, 12, 4	32, 3, 2, 7, 17, 28, 23, 12, 4

Instance c-3: 15 locations

WSP location	6, 10, 32, 34, 27, 26, 8, 16, 33, 37, 21, 28, 19, 5, 20
WSI location	13, 19, 32, 25, 8, 0, 10, 21, 5, 36, 22, 17, 39, 6, 18
Client location	16, 8, 3, 6, 18, 2, 32, 33, 17, 0, 19, 31, 35, 10, 24
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 27940	$ 27940
Percent Above Optimal Solution	0.0%	N/A
Run Time	16 Seconds	23 Seconds
WSP location	6, 16, 19	6, 16, 19
WSI location	19, 0, 10, 6, 18	19, 0, 10, 6, 18

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 27940	$ 27940
Percent Above DROP Solution	N/A	0.0%
Run Time	11 Seconds	5 Second
WSP location	6, 16, 19	6, 16, 19
WSI location	19, 0, 10, 6, 18	19, 0, 10, 6, 18

Instance c-4: 15 locations

WSP location	39, 24, 20, 32, 4, 17, 21, 26, 30, 9, 3, 6, 12, 36, 34
WSI location	2, 4, 36, 15, 39, 24, 13, 14, 9, 21, 7, 17, 28, 12, 35
Client location	8, 21, 37, 6, 20, 29, 38, 39, 25, 26, 5, 15, 7, 3, 36
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 24839	$ 24839
Percent Above Optimal Solution	0.0%	N/A
Run Time	51 Seconds	65 Seconds
WSP location	19, 38, 29, 2	19, 38, 29, 2
WSI location	10, 39, 15, 3, 5, 17	10, 39, 15, 3, 5, 17

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 24839	$ 24839
Percent Above DROP Solution	N/A	0.0%
Run Time	36 Seconds	15 Seconds
WSP location	19, 38, 29, 2	19, 38, 29, 2
WSI location	10, 39, 15, 3, 5, 17	10, 39, 15, 3, 5, 17

Instance c-5: 15 locations

WSP location	23, 16, 9, 39, 32, 20, 3, 28, 7, 25, 1, 36, 22, 38, 26
WSI location	26, 5, 33, 14, 18, 27, 35, 21, 39, 37, 13, 20, 6, 8, 4
Client location	32, 14, 33, 13, 7, 20, 6, 21, 37, 10, 19, 23, 28, 2, 15
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$128260	$128260
Percent Above Optimal Solution	0.0%	N/A
Run Time	18 Seconds	16 Second
WSP location	16, 20	16, 20
WSI location	33, 21, 37, 13, 20, 6	33, 21, 37, 13, 20, 6

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$128260	$128260
Percent Above DROP Solution	N/A	0.0%
Run Time	14 Seconds	4 Seconds
WSP location	16, 20	16, 20
WSI location	33, 21, 37, 13, 20, 6	33, 21, 37, 13, 20, 6

Instance c-6: 15 locations

WSP location	18, 6, 29, 39, 3, 31, 13, 37, 38, 11, 14, 35, 25, 34, 23
WSI location	24, 23, 29, 38, 32, 4, 22, 13, 3, 25, 30, 10, 11, 0, 17
Client location	28, 11, 5, 13, 19, 30, 15, 36, 26, 16, 34, 22, 10, 2, 1
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$163446	$161976
Percent Above Optimal Solution	0.907542%	N/A
Run Time	48 Seconds	6 Seconds
WSP location	13, 38, 11, 35, 34	13, 38, 11, 35
WSI location	38, 13, 30, 11, 0,	38, 13, 30, 11

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$163446	$163446
Percent Above DROP Solution	N/A	0.088311014%
Run Time	33 Seconds	15 Seconds
WSP location	13, 38, 11, 35, 34	13, 38, 11, 35, 34
WSI location	38, 13, 30, 11, 0,	38, 13, 30, 11, 0,

instance c-7: 15 locations

WSP location	36, 2, 9, 28, 34, 39, 13, 38, 1, 10, 25, 21, 5, 7, 0
WSI location	15, 37, 10, 16, 25, 31, 39, 3, 0, 2, 1, 22, 19, 8, 4
Client location	22, 5, 11, 7, 15, 24, 29, 18, 26, 9, 12, 32, 14, 33, 19
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$53760	$53760
Percent Above Optimal Solution	0.088057051%	N/A
Run Time	16 Seconds	31 Seconds
WSP location	39, 10	39, 10
WSI location	15, 10, 19	15, 10, 19

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$53760	$53760
Percent Above DROP Solution	N/A	1.33126%
Run Time	12 Seconds	4 Second
WSP location	39, 10	39, 10
WSI location	15, 10, 19	15, 10, 19

Instance c-8: 15 locations

WSP location	12, 2, 21, 14, 36, 30, 32, 16, 38, 17, 19, 18, 22, 28, 10
WSI location	15, 30, 37, 4, 9, 3, 25, 0, 18, 29, 23, 8, 13, 34, 21
Client location	22, 39, 37, 35, 30, 16, 10, 15, 2, 13, 31, 19, 21, 17, 23
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$62450	$62450
Percent Above Optimal Solution	0.0%	N/A
Run Time	49 Seconds	82 Seconds
WSP location	2, 30, 10	2, 30, 10
WSI location	15, 30, 13	15, 30, 13

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$63840	$62450
Percent Above DROP Solution	N/A	2.22578%
Run Time	38 Seconds	11 Seconds
WSP location	2, 30, 10	2, 30, 10
WSI location	15, 30, 8	15, 30, 13

Instance d-1: 20 locations

WSP location	26, 19, 2, 7, 11, 20, 13, 15, 14, 21, 1, 17, 18, 3, 27, 8, 38, 5, 4, 36
WSI location	36, 16, 19, 13, 26, 1, 31, 33, 17, 37, 18, 4, 34, 38, 7, 10, 5, 0, 15, 29
Client location	39, 37, 3, 36, 13, 7, 23, 26, 5, 9, 30, 27, 4, 8, 32, 1, 2, 29, 38, 25
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 70913	$ 70913
Percent Above Optimal Solution	0.0%	N/A
Run Time	60 Seconds	69 Second
WSP location	19, 7, 13, 1, 38, 4, 36	19, 7, 13, 1, 38, 4, 36
WSI location	36, 19, 13, 1, 31, 37, 4, 38, 7, 5, 15, 29	36, 19, 13, 1, 31, 37, 4, 38, 7, 5, 15, 29

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 70913	$ 70913
Percent Above DROP Solution	N/A	0.0%
Run Time	42 Seconds	18 Second
WSP location	19, 7, 13, 1, 38, 4, 36	19, 7, 13, 1, 38, 4, 36
WSI location	36, 19, 13, 1, 31, 37, 4, 38, 7, 5, 15, 29	36, 19, 13, 1, 31, 37, 4, 38, 7, 5, 15, 29

Instance d-2: 20 locations

WSP location	17, 8, 38, 4, 7, 21, 11, 28, 23, 26, 24, 34, 5, 10, 6, 27, 16, 2, 31, 1
WSI location	23, 17, 38, 15, 26, 21, 6, 36, 3, 1, 16, 29, 34, 30, 27, 31, 22, 37, 24, 20
Client location	30, 34, 39, 14, 7, 26, 31, 22, 18, 24, 6, 19, 29, 12, 15, 20, 13, 35, 8, 25
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 124372	$ 123752
Percent Above Optimal Solution	0.501002%	N/A
Run Time	171 Seconds	55 Second
WSP location	26, 24, 34, 10, 6, 27, 16, 2, 31	26, 24, 34, 10, 6, 27, 16, 2
WSI location	3615, 26, 6, 16, 29, 34, 30, 27, 31, 24, 20	15, 26, 6, 16, 29, 34, 30, 27, 24, 20

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 124372	$ 124372
Percent Above DROP Solution	N/A	0.0%
Run Time	109 Seconds	62 Seconds
WSP location	26, 24, 34, 10, 6, 27, 16, 2, 31	26, 24, 34, 10, 6, 27, 16, 2, 31
WSI location	3615, 26, 6, 16, 29, 34, 30, 27, 31, 24, 20	3615, 26, 6, 16, 29, 34, 30, 27, 31, 24, 20

Instance d-3: 20 locations

WSP location	20, 4, 25, 15, 6, 26, 39, 23, 12, 5, 22, 38, 24, 33, 30, 3, 2, 1, 13, 18
WSI location	27, 21, 25, 14, 35, 39, 23, 8, 36, 29, 30, 13, 3, 32, 17, 11, 24, 15, 10, 4
Client location	33, 30, 25, 5, 16, 8, 34, 10, 28, 13, 38, 3, 9, 12, 35, 18, 7, 6, 11, 27
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 29333	$ 29333
Percent Above Optimal Solution	0.0%	N/A
Run Time	51 Seconds	236 Seconds
WSP location	25, 26, 39, 2, 13	25, 26, 39, 2, 13
WSI location	35, 39, 36, 3, 17, 11, 10	35, 39, 36, 3, 17, 11, 10

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 29333	$ 29333
Percent Above DROP Solution	N/A	0.0%
Run Time	37 Seconds	14 Second
WSP location	25, 26, 39, 2, 13	25, 26, 39, 2, 13
WSI location	35, 39, 36, 3, 17, 11, 10	35, 39, 36, 3, 17, 11, 10

Instance d-4: 20 locations

WSP location	36, 31, 9, 26, 23, 0, 7, 10, 22, 20, 16, 27, 25, 14, 24, 13, 15, 11, 17, 28
WSI location	3, 0, 10, 25, 4, 5, 32, 17, 6, 31, 2, 15, 36, 21, 34, 37, 39, 16, 35, 22
Client location	6, 28, 26, 0, 39, 12, 24, 10, 9, 18, 17, 8, 16, 14, 32, 27, 11, 34, 4, 2
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 27800	$ 27020
Percent Above Optimal Solution	2.88675%	N/A
Run Time	188 Seconds	5687 Seconds
WSP location	22, 16, 25, 15, 17, 28	16, 25, 15, 17, 28,
WSI location	3, 10, 4, 17, 2, 39, 22	3, 10, 4, 17, 2, 37, 39

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 27800	$ 27800
Percent Above DROP Solution	N/A	0.0%
Run Time	133 Seconds	55 Seconds
WSP location	22, 16, 25, 15, 17, 28	22, 16, 25, 15, 17, 28
WSI location	3, 10, 4, 17, 2, 39, 22	3, 10, 4, 17, 2, 39, 22

Instance d-5: 20 locations

WSP location	37, 4, 7, 6, 38, 15, 21, 20, 16, 5, 29, 24, 18, 22, 36, 33, 23, 13, 28, 32
WSI location	4, 13, 6, 36, 12, 0, 15, 35, 8, 29, 5, 18, 33, 1, 7, 2, 31, 9, 30, 16
Client location	14, 10, 21, 20, 11, 32, 17, 33, 2, 3, 25, 19, 16, 1, 5, 0, 6, 34, 4, 9
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$161145	$161145
Percent Above Optimal Solution	0.0%	N/A
Run Time	52 Seconds	721 Second
WSP location	15, 29, 33, 23, 28	15, 29, 33, 23, 28
WSI location	4, 0, 15, 35, 29, 33, 1,	4, 0, 15, 35, 29, 33, 1,

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$161145	$161145
Percent Above DROP Solution	N/A	0.0%
Run Time	37 Seconds	15 Seconds
WSP location	15, 29, 33, 23, 28	15, 29, 33, 23, 28
WSI location	4, 0, 15, 35, 29, 33, 1,	4, 0, 15, 35, 29, 33, 1,

Instance d-6: 20 locations

WSP location	12, 37, 26, 31, 28, 25, 9, 35, 24, 10, 1, 16, 6, 22, 2, 0, 8, 19, 33, 5
WSI location	16, 25, 10, 14, 15, 32, 2, 28, 22, 13, 21, 17, 1, 26, 18, 11, 20, 39, 8, 38
Client location	22, 34, 3, 5, 4, 6, 27, 13, 31, 37, 29, 7, 15, 20, 17, 0, 9, 33, 36, 25
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$184040	$182359
Percent Above Optimal Solution	0.921808%	N/A
Run Time	181 Seconds	1507 Seconds
WSP location	25, 10, 1, 22, 2, 19	25, 10, 1, 2, 19
WSI location	25, 10, 15, 28, 22, 1, 20	25, 10, 2, 1, 20

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$184040	$184040
Percent Above DROP Solution	N/A	0.0%
Run Time	127 Seconds	54 Seconds
WSP location	25, 10, 1, 22, 2, 19	25, 10, 1, 22, 2, 19
WSI location	25, 10, 15, 28, 22, 1, 20	25, 10, 15, 28, 22, 1, 20

instance d-7: 20 locations

WSP location	10, 2, 20, 39, 24, 9, 29, 23, 17, 31, 6, 32, 27, 11, 4, 7, 22, 33, 35, 5
WSI location	16, 11, 20, 30, 5, 23, 14, 8, 32, 15, 0, 24, 27, 12, 39, 7, 21, 4, 22, 1
Client location	26, 8, 37, 4, 20, 3, 31, 30, 19, 27, 12, 17, 15, 18, 29, 33, 6, 39, 34, 16
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$63630	$62030
Percent Above Optimal Solution	2.5794%	N/A
Run Time	47 Seconds	373 Seconds
WSP location	20, 23, 27	20, 23
WSI location	20, 32, 1	20, 1

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$63630	$63630
Percent Above DROP Solution	N/A	1.33126%
Run Time	39 Seconds	8 Second
WSP location	4, 7	4, 7
WSI location	16, 14, 8, 12	16, 14, 12, 1

Instance d-8: 20 locations

WSP location	25, 23, 27, 20, 31, 12, 11, 6, 15, 18, 9, 33, 32, 17, 1, 0, 14, 7, 37, 24
WSI location	29, 11, 3, 25, 4, 39, 6, 38, 21, 37, 28, 10, 35, 17, 2, 14, 34, 30, 31, 20
Client location	35, 20, 3, 34, 6, 24, 29, 32, 21, 22, 5, 18, 15, 27, 39, 19, 23, 12, 28, 26
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$67420	$66090
Percent Above Optimal Solution	2.01241%	N/A
Run Time	183 Seconds	2223 Seconds
WSP location	20, 31, 37	20, 9, 17
WSI location	37, 31, 20	29, 17, 20

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$67420	$67420
Percent Above DROP Solution	N/A	0.0%
Run Time	152 Seconds	31 Seconds
WSP location	20, 31, 37	20, 31, 37
WSI location	37, 31, 20	37, 31, 20

Instance e-1: 25 locations

WSP location	36, 30, 6, 16, 22, 8, 38, 17, 37, 2, 1, 0, 19, 11, 34, 14, 15, 9, 21, 32, 25, 28, 3, 10, 23
WSI location	3, 39, 6, 7, 17, 36, 33, 24, 21, 20, 16, 31, 2, 9, 28, 14, 25, 34, 0, 11, 5, 32, 26, 27, 38
Client location	6, 27, 30, 22, 11, 35, 25, 17, 16, 1, 32, 20, 2, 28, 39, 31, 4, 7, 24, 33, 29, 38, 14, 15, 34
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 83430	$ 82920
Percent Above Optimal Solution	0.615051%	N/A
Run Time	134 Seconds	1171 Seconds
WSP location	36, 6, 16, 38, 2, 11, 34, 32, 25, 28	36, 6, 16, 38, 2, 11, 34, 25, 28
WSI location	6, 7, 17, 36, 33, 24, 16, 31, 2, 9, 28, 14, 25, 34, 11, 32, 38	6, 7, 17, 36, 33, 24, 16, 31, 2, 9, 28, 14, 25, 34, 11, 38

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 83430	$ 83430
Percent Above DROP Solution	N/A	0.0%
Run Time	83 Seconds	51 Seconds
WSP location	36, 6, 16, 38, 2, 11, 34, 32, 25, 28	36, 6, 16, 38, 2, 11, 34, 32, 25, 28
WSI location	6, 7, 17, 36, 33, 24, 16, 31, 2, 9, 28, 14, 25, 34, 11, 32, 38	6, 7, 17, 36, 33, 24, 16, 31, 2, 9, 28, 14, 25, 34, 11, 32, 38

Instance e-2: 25 locations

WSP location	16, 37, 33, 39, 25, 28, 13, 19, 29, 38, 8, 31, 36, 4, 35, 5, 1, 15, 2, 11, 3, 0, 10, 14, 12
WSI location	13, 9, 8, 23, 31, 29, 26, 37, 27, 4, 18, 20, 25, 28, 22, 19, 0, 35, 24, 12, 21, 17, 30, 33, 1
Client location	6, 0, 8, 33, 9, 16, 4, 1, 22, 35, 2, 36, 31, 13, 37, 26, 27, 10, 11, 39, 19, 21, 38, 34, 15
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 108923	$ 108923
Percent Above Optimal Solution	0.0%	N/A
Run Time	483 Seconds	788 Second
WSP location	37, 33, 25, 28, 13, 19, 38, 8, 31, 36, 4, 35, 1, 0	37, 33, 25, 28, 13, 19, 38, 8, 31, 36, 4, 35, 1, 0
WSI location	13, 8, 31, 26, 37, 27, 4, 28, 22, 19, 0, 35, 24, 30, 33, 1	13, 8, 31, 26, 37, 27, 4, 28, 22, 19, 0, 35, 24, 30, 33, 1

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 108923	$ 108923
Percent Above DROP Solution	N/A	0.0%
Run Time	289 Seconds	194 Seconds
WSP location	37, 33, 25, 28, 13, 19, 38, 8, 31, 36, 4, 35, 1, 0	37, 33, 25, 28, 13, 19, 38, 8, 31, 36, 4, 35, 1, 0
WSI location	13, 8, 31, 26, 37, 27, 4, 28, 22, 19, 0, 35, 24, 30, 33, 1	13, 8, 31, 26, 37, 27, 4, 28, 22, 19, 0, 35, 24, 30, 33, 1

Instance e-3: 25 locations

WSP location	9, 16, 6, 1, 18, 8, 13, 5, 24, 7, 26, 17, 27, 37, 28, 22, 32, 34, 4, 25, 21, 19, 10, 29, 31
WSI location	13, 36, 30, 21, 1, 8, 2, 31, 25, 17, 28, 32, 33, 10, 27, 37, 5, 24, 29, 6, 3, 18, 22, 4, 11
Client location	16, 24, 7, 36, 27, 15, 35, 23, 37, 29, 32, 3, 11, 2, 5, 19, 39, 22, 4, 21, 14, 31, 20, 34, 9,
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 37550	N/A
Percent Above Optimal Solution	N/A	N/A
Run Time	161 Seconds	N/A
WSP location	13, 19	N/A
WSI location	2, 28, 10, 29, 6, 3, 18, 22, 4, 11	N/A

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 37550	$ 37550
Percent Above DROP Solution	N/A	0.0%
Run Time	117 Seconds	44 Seconds
WSP location	24, 37, 32, 34, 4, 21	13, 19
WSI location	30, 21, 28, 37, 24, 3, 4, 11	2, 28, 10, 29, 6, 3, 18, 22, 4, 11

CPLEX out of memory: feasible solution 40624 at 161 second. Feasible solution 38730 at 4243 second when run out of memory

Instance e-4: 25 locations

WSP location	0, 31, 8, 34, 20, 2, 19, 15, 7, 12, 18, 28, 1, 27, 9, 21, 23, 36, 13, 5, 17, 6, 22, 39, 37
WSI location	36, 3, 32, 19, 34, 27, 23, 0, 6, 24, 30, 22, 39, 11, 38, 28, 20, 10, 9, 26, 21, 16, 37, 31, 5
Client location	16, 13, 24, 11, 26, 23, 33, 6, 10, 12, 37, 38, 3, 19, 15, 18, 21, 5, 8, 20, 36, 0, 9, 27, 1
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 34285	N/A
Percent Above Optimal Solution	N/A	N/A
Run Time	556 Seconds	N/A
WSP location	2, 19, 27, 9, 21, 23, 36	N/A
WSI location	36, 19, 27, 23, 11, 10, 21	N/A

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 34285	$ 34285
Percent Above DROP Solution	N/A	0.0%
Run Time	402 Seconds	154 Seconds
WSP location	2, 19, 27, 9, 21, 23, 36	2, 19, 27, 9, 21, 23, 36
WSI location	36, 19, 27, 23, 11, 10, 21	36, 19, 27, 23, 11, 10, 21

CPLEX out of memory: feasible solution 36170 at 161 second. Feasible solution 35615 at 5538 second when run out of memory

Instance e-5: 25 locations

WSP location	22, 33, 7, 35, 16, 28, 20, 38, 21, 13, 0, 10, 25, 3, 2, 30, 23, 14, 29, 5, 4, 11, 39, 18, 26
WSI location	26, 22, 31, 10, 3, 35, 12, 21, 1, 25, 4, 2, 5, 38, 0, 17, 32, 33, 14, 39, 9, 7, 18, 15, 13
Client location	29, 10, 7, 25, 37, 2, 4, 14, 13, 12, 9, 21, 27, 15, 36, 5, 31, 28, 11, 32, 38, 26, 8, 35, 16
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 155877	N/A
Percent Above Optimal Solution	N/A	N/A
Run Time	139 Seconds	N/A
WSP location	33, 35, 38, 13, 3, 2	N/A
WSI location	35, 12, 25, 2, 38, 32, 39, 7, 13	N/A

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 155877	$ 155877
Percent Above DROP Solution	N/A	0.0%
Run Time	100 Seconds	39 Seconds
WSP location	33, 35, 38, 13, 3, 2	33, 35, 38, 13, 3, 2
WSI location	35, 12, 25, 2, 38, 32, 39, 7, 13	35, 12, 25, 2, 38, 32, 39, 7, 13

CPLEX stopped: feasible solution 160592 at 139 second. Feasible solution 155882 at 22246 second when being stopped.

Instance e-6: 25 locations

WSP location	10, 4, 16, 34, 30, 13, 38, 18, 31, 22, 1, 21, 9, 5, 28, 25, 11, 36, 32, 3, 6, 35, 26, 20, 24
WSI location	3, 35, 16, 1, 7, 14, 4, 37, 33, 31, 17, 27, 20, 38, 29, 8, 12, 32, 39, 10, 23, 13, 34, 21, 6
Client location	36, 19, 23, 4, 20, 34, 37, 28, 13, 33, 9, 12, 11, 32, 10, 26, 27, 14, 38, 8, 15, 0, 2, 30, 31
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$231170	N/A
Percent Above Optimal Solution	N/A	N/A
Run Time	549 Seconds	N/A
WSP location	4, 34, 13, 38, 1, 3, 35	N/A
WSI location	3, 35, 1, 4, 38, 13, 34	N/A

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$231170	$231170
Percent Above DROP Solution	N/A	0.0%
Run Time	393 Seconds	156 Seconds
WSP location	4, 34, 13, 38, 1, 3, 35	4, 34, 13, 38, 1, 3, 35
WSI location	3, 35, 1, 4, 38, 13, 34	3, 35, 1, 4, 38, 13, 34

CPLEX stopped: feasible solution 232346 at 549 second. Feasible solution 232346 at 6515 second when being stopped.

Instance e-7: 25 locations

WSP location	32, 30, 23, 9, 0, 37, 6, 24, 13, 33, 7, 11, 31, 25, 1, 12, 21, 14, 34, 27, 16, 26, 38, 35, 4
WSI location	39, 7, 24, 13, 14, 22, 32, 12, 8, 33, 21, 15, 27, 19, 35, 26, 34, 4, 3, 29, 17, 18, 2, 37, 30
Client location	5, 16, 24, 38, 34, 27, 7, 17, 0, 18, 29, 36, 8, 33, 35, 25, 22, 23, 11, 39, 2, 28, 10, 12, 26
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$77684	N/A
Percent Above Optimal Solution	N/A	N/A
Run Time	132 Seconds	N/A
WSP location	37, 13	N/A
WSI location	19, 29, 18	N/A

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$78544	$77684
Percent Above DROP Solution	N/A	1.10705%
Run Time	109 Seconds	23 Second
WSP location	23, 34	23, 34
WSI location	34, 29, 17, 18, 30	8, 34, 17, 18, 30

CPLEX stopped: feasible solution 86884 at 132 second. Feasible solution 76494 at 7659 second when being stopped.

Instance e-8: 25 locations

WSP location	33, 38, 39, 29, 25, 27, 4, 26, 3, 1, 30, 34, 22, 0, 36, 14, 37, 5, 23, 10, 35, 21, 9, 16, 7,
WSI location	27, 21, 39, 31, 4, 14, 19, 1, 29, 17, 9, 11, 34, 20, 6, 15, 12, 16, 3, 7, 38, 33, 35, 5, 8
Client location	23, 1, 15, 9, 7, 27, 8, 37, 5, 17, 30, 3, 26, 24, 2, 21, 16, 0, 10, 36, 4, 20, 32, 6, 22
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 82515	N/A
Percent Above Optimal Solution	N/A	N/A
Run Time	559 Seconds	N/A
WSP location	38, 10, 35	N/A
WSI location	17, 9, 15, 16, 35	N/A

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 84650	$ 82515
Percent Above DROP Solution	N/A	0.645875701%
Run Time	448 Seconds	111 Seconds
WSP location	38, 27, 10, 35	38, 10, 35
WSI location	17, 3, 35, 8	17, 9, 15, 16, 35

CPLEX stopped: feasible solution 83750 at 559 second. Feasible solution 80740 at 7576 second when being stopped.

Instance f-1: 30 locations

WSP location	7, 26, 37, 33, 25, 35, 27, 2, 17, 9, 8, 1, 10, 28, 36, 30, 13, 29, 11, 32, 14, 24, 38, 21, 16, 15, 3, 31, 6, 18
WSI location	17, 23, 31, 3, 9, 37, 2, 13, 26, 6, 21, 1, 24, 27, 8, 18, 28, 19, 22, 38, 14, 15, 35, 5, 34, 30, 11, 20, 0, 12
Client location	23, 32, 2, 30, 10, 27, 39, 1, 37, 29, 31, 4, 36, 6, 21, 7, 13, 20, 8, 12, 15, 19, 16, 17, 3, 25, 9, 18, 22, 14
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 84946
Run Time	274 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 84946	$ 84946
Percent Above DROP Solution	N/A	0.0%
Run Time	166 Seconds	108 Seconds
WSP location	37, 27, 17, 8, 1, 28, 30, 13, 14, 21, 15, 6, 18	37, 27, 17, 8, 1, 28, 30, 13, 14, 21, 15, 6, 18
WSI location	17, 31, 3, 9, 37, 2, 13, 6, 21, 1, 27, 8, 18, 28, 19, 22, 14, 15, 30, 20, 12	17, 31, 3, 9, 37, 2, 13, 6, 21, 1, 27, 8, 18, 28, 19, 22, 14, 15, 30, 20, 12

Instance f-2: 30 locations

WSP location	32, 25, 29, 28, 11, 5, 34, 36, 18, 39, 9, 31, 24, 7, 2, 30, 8, 12, 20, 4, 6, 35, 3, 21, 15, 27, 23, 14, 17, 16
WSI location	35, 13, 12, 38, 20, 9, 6, 8, 15, 7, 21, 2, 18, 19, 16, 14, 17, 32, 5, 31, 4, 37, 26, 33, 27, 0, 10, 25, 11, 36
Client location	38, 2, 29, 27, 32, 18, 12, 30, 1, 20, 19, 26, 39, 28, 33, 5, 8, 25, 34, 14, 22, 35, 36, 6, 7, 13, 16, 31, 37, 21
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 105854
Run Time	1196 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 105854	$ 105854
Percent Above DROP Solution	N/A	0.0%
Run Time	720 Seconds	476 Seconds
WSP location	32, 5, 36, 18, 9, 31, 7, 2, 8, 12, 20, 6, 35, 21, 27, 14, 16	32, 5, 36, 18, 9, 31, 7, 2, 8, 12, 20, 6, 35, 21, 27, 14, 16
WSI location	35, 12, 20, 9, 6, 8, 7, 21, 2, 18, 16, 14, 32, 5, 31, 27, 36	35, 12, 20, 9, 6, 8, 7, 21, 2, 18, 16, 14, 32, 5, 31, 27, 36

Instance f-3: 30 locations

WSP location	30, 9, 32, 19, 24, 4, 16, 13, 21, 38, 34, 12, 26, 33, 11, 23, 7, 5, 6, 35, 18, 25, 1, 0, 36, 8, 10, 39, 15, 2
WSI location	36, 17, 24, 31, 0, 29, 18, 5, 6, 2, 38, 28, 13, 35, 21, 1, 23, 25, 19, 15, 20, 10, 8, 22, 27, 39, 32, 12, 16, 33
Client location	3, 26, 25, 30, 21, 14, 6, 29, 32, 5, 1, 34, 17, 4, 35, 16, 27, 2, 37, 36, 39, 18, 13, 8, 20, 12, 10, 24, 15, 38
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 34459
Run Time	303 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 35494	$ 34459
Percent Above DROP Solution	N/A	3.00357%
Run Time	213 Seconds	90 Seconds
WSP location	30, 24, 4, 38, 34, 12, 6, 18, 8	30, 4, 38, 34, 12, 6, 18, 8, 15
WSI location	36, 31, 18, 38, 25, 15, 20, 10, 12	36, 31, 18, 38, 25, 15, 20, 10, 12

Instance f-4: 30 locations

WSP location	27, 1, 19, 39, 14, 2, 16, 17, 26, 13, 28, 20, 33, 3, 15, 31, 9, 23, 5, 32, 37, 18, 38, 21, 22, 4, 25, 6, 8, 12
WSI location	37, 39, 4, 13, 15, 31, 23, 20, 38, 19, 21, 33, 34, 3, 7, 32, 18, 30, 2, 17, 24, 26, 0, 9, 25, 35, 8, 29, 11, 10
Client location	7, 36, 20, 27, 30, 11, 0, 6, 22, 17, 19, 10, 14, 2, 21, 37, 31, 16, 5, 26, 32, 8, 15, 25, 24, 28, 18, 34, 12, 35
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 41940
Run Time	1396 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 41940	$ 41940
Percent Above DROP Solution	N/A	0.0%
Run Time	973 Seconds	423 Seconds
WSP location	19, 39, 2, 16, 20, 37, 38, 4, 6	19, 39, 2, 16, 20, 37, 38, 4, 6
WSI location	37, 39, 15, 31, 38, 19, 34, 7, 30, 25, 10	37, 39, 15, 31, 38, 19, 34, 7, 30, 25, 10

Instance f-5: 30 locations

WSP location	9, 3, 25, 20, 11, 16, 39, 28, 22, 13, 14, 8, 29, 32, 27, 19, 38, 36, 23, 21, 31, 5, 33, 2, 26, 7, 35, 6, 15, 30
WSI location	16, 12, 25, 19, 32, 30, 24, 14, 37, 5, 38, 4, 29, 17, 34, 35, 9, 18, 31, 20, 1, 13, 39, 27, 11, 3, 36, 0, 2, 22
Client location	22, 21, 26, 10, 3, 1, 39, 12, 29, 38, 33, 17, 37, 5, 30, 13, 25, 15, 6, 32, 19, 24, 27, 8, 31, 23, 20, 2, 36, 14
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 194190
Run Time	314 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$194190	$193470
Percent Above DROP Solution	N/A	0.372151%
Run Time	216 Seconds	98 Seconds
WSP location	20, 16, 22, 38, 31, 5, 33, 2, 30	20, 16, 22, 38, 31, 5, 33, 2, 30
WSI location	16, 12, 30, 5, 38, 29, 31, 20, 1, 27, 2, 22	16, 12, 25, 30, 5, 38, 29, 31, 20, 1, 2, 22

Instance f-6: 30 locations

WSP location	14, 0, 35, 24, 3, 34, 27, 22, 18, 11, 8, 37, 20, 23, 36, 38, 32, 39, 28, 25, 5, 33, 9, 30, 1, 10, 21, 19, 4, 31
WSI location	17, 28, 11, 8, 29, 1, 26, 20, 6, 30, 23, 12, 33, 15, 7, 0, 32, 39, 35, 38, 10, 2, 13, 16, 3, 19, 22, 9, 14, 24
Client location	24, 37, 11, 38, 19, 14, 5, 21, 22, 7, 20, 23, 34, 9, 10, 1, 31, 26, 35, 4, 15, 16, 28, 17, 3, 27, 18, 13, 8, 12
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 197710
Run Time	1406 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$198410	$197710
Percent Above DROP Solution	N/A	0.354054%
Run Time	981 Seconds	425 Seconds
WSP location	14, 24, 3, 11, 20, 38, 28, 9, 1, 10	14, 24, 3, 11, 20, 38, 28, 9, 1, 10
WSI location	28, 11, 1, 20, 38, 10, 3, 9, 14, 24	28, 11, 1, 20, 38, 16, 3, 9, 14, 24

Instance f-7: 30 locations

WSP location	29, 38, 26, 8, 2, 33, 27, 21, 7, 6, 23, 22, 14, 34, 0, 11, 24, 18, 13, 39, 15, 5, 9, 12, 4, 20, 10, 35, 28, 25
WSI location	32, 18, 2, 24, 37, 15, 26, 19, 33, 10, 34, 8, 38, 4, 12, 17, 11, 0, 30, 22, 16,35, 3, 29, 14, 27, 39, 23, 5, 6
Client location	39, 35, 3, 14, 18, 29, 11, 34, 16, 24, 13, 32, 0, 19, 5, 23, 7, 26, 15, 37, 4, 1, 6, 21, 22, 33, 12, 2, 38, 31
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 79870
Run Time	278 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$79870	$79870
Percent Above DROP Solution	N/A	0.0%
Run Time	227 Seconds	51 Seconds
WSP location	29, 26, 20	29, 26, 20
WSI location	15, 26, 33, 10, 34, 3	15, 26, 33, 10, 34, 3

Instance f-8: 30 locations

WSP location	10, 0, 13, 17, 31, 20, 23, 35, 1, 6, 24, 12, 14, 38, 37, 2, 11, 8, 16, 4, 3, 32, 34, 33, 9, 5, 26, 30, 7, 22
WSI location	16, 17, 13, 12, 33, 8, 20, 38, 36, 3, 14, 2, 22, 29, 32, 35, 5, 1, 4, 27, 15, 10, 24, 31, 30, 34, 0, 11, 28, 37
Client location	23, 26, 13, 6, 1, 39, 33, 32, 21, 28, 25, 8, 22, 30, 10, 16, 37, 17, 0, 14, 27, 38, 29, 34, 11, 12, 4, 5, 24, 3
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 82830
Run Time	1320 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$82830	$82830
Percent Above DROP Solution	N/A	0.459182959%
Run Time	1059 Seconds	261 Seconds
WSP location	0, 17, 37, 3, 33	0, 17, 37, 3, 33
WSI location	3, 38, 3, 0, 37	3, 38, 3, 0, 37

Instance g-1: 35 locations

WSP location	5, 39, 20, 23, 38, 0, 25, 18, 3, 4, 11, 17, 7, 27, 9, 29, 26, 32, 14, 1, 28, 8, 16, 37, 15, 36, 30, 33, 22, 24, 13, 10, 6, 2, 19
WSI location	3, 21, 37, 16, 6, 9, 28, 11, 18, 22, 17, 19, 7, 30, 26, 15, 10, 8, 24, 14, 0, 4, 39, 5, 13, 1, 20, 12, 34, 2, 32, 36, 33, 25, 31
Client location	16, 7, 37, 10, 18, 8, 28, 19, 4, 20, 34, 15, 1, 35, 26, 29, 13, 23, 21, 24, 27, 22, 11, 33, 5, 38, 0, 31, 12, 14, 17, 3, 36, 32, 39
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 94116
Run Time	546 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 94116	$ 94116
Percent Above DROP Solution	N/A	0.0%
Run Time	329 Seconds	217 Seconds
WSP location	5, 20, 38, 0, 18, 3, 11, 7, 14, 1, 28, 37, 33, 22, 24, 13, 10	5, 20, 38, 0, 18, 3, 11, 7, 14, 1, 28, 37, 33, 22, 24, 13, 10
WSI location	3, 21, 37, 28, 11, 18, 22, 17, 19, 7, 26, 15, 10, 24, 14, 0, 5, 13, 1, 20, 12, 32, 36, 33	3, 21, 37, 28, 11, 18, 22, 17, 19, 7, 26, 15, 10, 24, 14, 0, 5, 13, 1, 20, 12, 32, 36, 33

Instance g-2: 35 locations

WSP location	4, 33, 8, 30, 16, 38, 23, 6, 25, 22, 2, 12, 39, 21, 19, 27, 0, 17, 36, 37, 20, 29, 15, 26, 3, 28, 1, 10, 24, 31, 18, 14, 32, 11, 9
WSI location	10, 2, 8, 23, 19, 29, 15, 9, 21, 6, 38, 17, 16, 39, 30, 22, 34, 11, 33, 35, 3, 32, 12, 20, 4, 37, 0, 7, 14, 13, 24, 25, 18, 27, 28
Client location	17, 11, 8, 22, 0, 35, 34, 36, 30, 6, 13, 32, 37, 27, 3, 5, 23, 7, 33, 20, 4, 10, 12, 26, 38, 28, 18, 39, 1, 2, 16, 19, 29, 14, 31
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 102101
Run Time	2274 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 102101	$ 102101
Percent Above DROP Solution	N/A	0.0%
Run Time	1308 Seconds	966 Seconds
WSP location	33, 8, 30, 16, 38, 23, 6, 22, 2, 12, 39, 27, 0, 17, 37, 20, 3, 28, 18, 14, 32, 11, 9	33, 8, 30, 16, 38, 23, 6, 22, 2, 12, 39, 27, 0, 17, 37, 20, 3, 28, 18, 14, 32, 11, 9
WSI location	2, 8, 23, 19, 9, 6, 38, 17, 16, 39, 30, 22, 11, 33, 3, 32, 12, 20, 37, 0, 14, 18, 27, 28	2, 8, 23, 19, 9, 6, 38, 17, 16, 39, 30, 22, 11, 33, 3, 32, 12, 20, 37, 0, 14, 18, 27, 28

Instance g-3: 35 locations

WSP location	29, 1, 35, 4, 27, 33, 10, 39, 31, 32, 6, 26, 16, 9, 24, 38, 25, 13, 14, 21, 28, 12, 7, 8, 19, 0, 36, 20, 34, 15, 18, 22, 30, 17, 23
WSI location	2, 27, 35, 34, 29, 31, 3, 21, 0, 6, 39, 8, 18, 10, 11, 32, 17, 28, 1, 20, 15, 24, 7, 26, 30, 38, 16, 19, 4, 37, 33, 25, 12, 14, 23
Client location	12, 24, 8, 5, 20, 39, 10, 36, 21, 18, 1, 7, 16, 13, 25, 37, 35, 11, 23, 14, 3, 27, 30, 33, 2, 19, 9, 29, 22, 15, 4, 31, 34, 32, 28
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 374972
Run Time	609 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 374972	$ 374972
Percent Above DROP Solution	N/A	0.0%
Run Time	435 Seconds	174 Seconds
WSP location	35, 10, 39, 32, 26, 12, 15	35, 10, 39, 32, 26, 12, 15
WSI location	2, 35, 3, 6, 39, 18, 10, 20, 24, 26, 30, 33, 12	2, 35, 3, 6, 39, 18, 10, 20, 24, 26, 30, 33, 12

Instance g-4: 35 locations

WSP location	20, 39, 25, 37, 35, 1, 32, 27, 28, 12, 2, 10, 24, 0, 16, 7, 9, 26, 4, 38, 30, 21, 18, 14, 5, 11, 6, 15, 33, 8, 23, 22, 36, 17, 3
WSI location	30, 36, 9, 11, 10, 21, 5, 27, 8, 39, 22, 25, 31, 16, 13, 35, 37, 38, 3, 33, 26, 17, 29, 15, 0, 6, 28, 19, 23, 1, 14, 18, 24, 32, 34
Client location	3, 22, 2, 0, 21, 19, 24, 17, 7, 38, 36, 1, 30, 35, 34, 25, 27, 6, 8, 11, 28, 39, 18, 31, 13, 23, 33, 4, 37, 5, 32, 14, 16, 15, 12
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 43149
Run Time	3093 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 44139	$ 43149
Percent Above DROP Solution	N/A	2.29438%
Run Time	2159 Seconds	934 Seconds
WSP location	20, 28, 7, 9, 38, 14, 6, 15, 33, 23, 36	20, 28, 7, 9, 38, 14, 6, 15, 33, 23, 36
WSI location	36, 10, 27, 31, 16, 33, 26, 15, 28, 23, 34	36, 10, 27, 31, 16, 38, 33, 15, 28, 23, 34

Instance g-5: 35 locations

WSP location	25, 10, 13, 37, 15, 31, 22, 18, 29, 26, 4, 38, 11, 36, 7, 16, 17, 6, 2, 24, 9, 23, 21, 1, 27, 5, 12, 39, 8, 0, 3, 33, 35, 28, 30
WSI location	8, 24, 30, 0, 1, 9, 15, 26, 37, 25, 10, 3, 29, 13, 38, 23, 5, 4, 7, 12, 32, 35, 14, 33, 21, 17, 19, 16, 31, 18, 11, 28, 2, 20, 6
Client location	14, 1, 30, 31, 23, 3, 26, 2, 21, 33, 0, 7, 38, 18, 4, 27, 19, 12, 15, 28, 22, 37, 24, 9, 17, 34, 11, 5, 8, 20, 10, 32, 16, 13, 39
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 554092
Run Time	626 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$214719	$211019
Percent Above DROP Solution	N/A	1.7534%
Run Time	455 Seconds	171 Seconds
WSP location	26, 16, 2, 3, 33, 28, 30	26, 16, 2, 3, 33, 28, 30
WSI location	8, 30, 26, 37, 3, 5, 7, 33, 31, 28, 2	8, 30, 26, 37, 3, 13, 7, 33, 31, 28, 2

Instance g-6: 35 locations

WSP location	10, 31, 2, 14, 36, 32, 16, 6, 26, 5, 17, 18, 23, 0, 15, 11, 22, 3, 28, 21, 29, 13, 39, 9, 12, 35, 19, 4, 7, 34, 24, 30, 1, 8, 33
WSI location	16, 39, 3, 30, 23, 27, 21, 35, 7, 0, 15, 14, 10, 26, 6, 4, 19, 8, 33, 38, 28, 25, 22, 18, 34, 17, 36, 12, 13, 5, 1, 29, 31, 37, 11
Client location	23, 16, 3, 20, 12, 1, 6, 21, 24, 0, 22, 37, 14, 2, 26, 25, 13, 15, 27, 28, 30, 33, 9, 17, 18, 5, 10, 34, 31, 4, 35, 29, 38, 7, 11
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 22468
Run Time	3090 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$22468	$22468
Percent Above DROP Solution	N/A	0.0%
Run Time	2136 Seconds	954 Seconds
WSP location	14, 36, 16, 26, 23, 0, 15, 3, 13, 12, 35, 34	14, 36, 16, 26, 23, 0, 15, 3, 13, 12, 35, 34
WSI location	16, 3, 23, 35, 0, 15, 14, 26, 34, 36, 12, 13	16, 3, 23, 35, 0, 15, 14, 26, 34, 36, 12, 13

Instance g-7: 35 locations

WSP location	21, 10, 31, 29, 12, 17, 11, 26, 13, 1, 14, 6, 3, 25, 0, 35, 20, 22, 2, 9, 7, 18, 33, 4, 27, 15, 24, 38, 8, 16, 32, 37, 30, 23, 28
WSI location	27, 19, 31, 20, 33, 22, 36, 12, 32, 37, 1, 9, 11, 17, 30, 34, 5, 7, 29, 2, 38, 15, 25, 8, 14, 13, 3, 39, 24, 4, 18, 10, 35, 23, 16
Client location	34, 36, 31, 19, 14, 35, 20, 37, 7, 21, 26, 17, 1, 39, 22, 29, 28, 24, 23, 0, 4, 12, 9, 5, 10, 13, 32, 16, 30, 15, 2, 18, 11, 38, 8
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 95323
Run Time	573 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 97447	$ 95323
Percent Above DROP Solution	N/A	2.22821 %
Run Time	475 Seconds	98 Seconds
WSP location	18, 15, 28	18, 15, 28
WSI location	20, 36, 38, 15, 18, 10	19, 36, 38, 15, 18, 10

Instance g-8: 35 locations

WSP location	26, 37, 27, 4, 39, 8, 38, 13, 14, 3, 12, 16, 33, 1, 18, 17, 21, 29, 2, 34, 10, 0, 36, 23, 11, 22, 24, 28, 31, 7, 35, 9, 30, 19, 6
WSI location	32, 13, 27, 34, 28, 23, 7, 29, 36, 35, 38, 37, 3, 21, 11, 12, 19, 31, 17, 1, 15, 8, 6, 30, 26, 5, 2, 9, 4, 10, 25, 0, 39, 33, 16
Client location	36, 2, 4, 9, 22, 20, 16, 39, 13, 8, 17, 24, 27, 3, 31, 6, 28, 7, 30, 18, 37, 10, 38, 5, 29, 26, 33, 0, 34, 35, 1, 14, 15, 32, 21
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 93880
Run Time	2927 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$94260	$93880
Percent Above DROP Solution	N/A	0.459182959%
Run Time	2303 Seconds	624 Seconds
WSP location	14, 17, 2, 28, 7, 35	014, 17, 2, 28, 7, 35
WSI location	7, 35, 15, 26, 25, 39	27, 7, 35, 15, 26, 39

Instance h-1: 40 locations

WSP location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSI location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
Client location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 89089
Run Time	1006 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 89089	$ 89089
Percent Above DROP Solution	N/A	0.0%
Run Time	605 Seconds	401 Seconds
WSP location	0, 2, 5, 7, 8, 12, 13, 15, 16, 17, 19, 20, 21, 22, 26, 27, 28, 30, 33, 35, 37	0, 2, 5, 7, 8, 12, 13, 15, 16, 17, 19, 20, 21, 22, 26, 27, 28, 30, 33, 35, 37
WSI location	0, 1, 2, 5, 6, 7, 8, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 37, 39	0, 1, 2, 5, 6, 7, 8, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 37, 39

Instance h-2: 40 locations

WSP location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSI location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
Client location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 102029
Run Time	5409 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 102029	$ 102029
Percent Above DROP Solution	N/A	0.0%
Run Time	3377 Seconds	2032 Seconds
WSP location	32, 5, 36, 18, 9, 31, 7, 2, 8, 12, 20, 6, 35, 21, 27, 14, 17, 16, 0	32, 5, 36, 18, 9, 31, 7, 2, 8, 12, 20, 6, 35, 21, 27, 14, 17, 16, 0
WSI location	35, 12, 20, 9, 6, 8, 7, 21, 2, 18, 16, 14, 32, 5, 31, 26, 27, 36, 0	35, 12, 20, 9, 6, 8, 7, 21, 2, 18, 16, 14, 32, 5, 31, 26, 27, 36, 0

Instance h-3: 40 locations

WSP location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSI location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
Client location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 41349
Run Time	1240 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 42385	$ 41349
Percent Above DROP Solution	N/A	2.5055 %
Run Time	898 Seconds	342 Seconds
WSP location	3, 9, 11, 18, 24, 26, 30, 33, 34	3, 11, 16, 18, 24, 26, 30, 33, 34
WSI location	0, 2, 10, 12, 17, 18, 24, 25, 32, 33, 34, 36	2, 10, 12, 17, 18, 24, 25, 28, 32, 33, 34, 36

Instance h-4: 40 locations

WSP location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSI location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
Client location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSP Fixed Charge	1500
WSI Fixed Cost	1500
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 43449
Run Time	5762 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 43694	$ 43449
Percent Above DROP Solution	N/A	0.563879 %
Run Time	4114 Seconds	1648 Seconds
WSP location	1, 2, 12, 16, 17, 19, 27, 28, 30, 32, 34	1, 2, 12, 16, 17, 19, 27, 28, 30, 32, 34
WSI location	1, 7, 10, 11, 12, 14, 19, 26, 28, 31, 34, 36	1, 7, 10, 12, 14, 15, 19, 26, 28, 31, 34, 36

Instance h-5: 40 locations

WSP location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSI location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
Client location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 23733
Run Time	1290 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$241876	$23733
Percent Above DROP Solution	N/A	1.9129 %
Run Time	884 Seconds	406 Seconds
WSP location	0, 7, 9, 15, 17, 18, 22, 25, 32, 34, 35, 36	7, 9, 15, 17, 18, 25, 28, 32, 34, 35, 36
WSI location	0, 7, 9, 15, 17, 18, 22, 25, 32, 33, 34, 35, 36, 39	0, 1, 7, 9, 15, 17, 18, 25, 32, 33, 34, 35, 36, 38, 39

Instance h-6: 40 locations

WSP location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSI location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
Client location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 228954
Run Time	5662 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$228954	$228954
Percent Above DROP Solution	N/A	0.0%
Run Time	3964 Seconds	1698 Seconds
WSP location	2, 13, 20, 22, 23, 24, 25, 26, 28, 30, 31, 33	2, 13, 20, 22, 23, 24, 25, 26, 28, 30, 31, 33
WSI location	2, 13, 20, 22, 23, 24, 25, 26, 28, 30, 31, 33	2, 13, 20, 22, 23, 24, 25, 26, 28, 30, 31, 33

Instance h-7: 40 locations

WSP location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSI location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
Client location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 93880
Run Time	1121 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$97855	$93880
Percent Above DROP Solution	N/A	4.23413 %
Run Time	932 Seconds	189 Seconds
WSP location	5, 31, 34, 36	5, 15, 26, 34, 36
WSI location	9, 11, 17, 20, 27, 30	9, 11, 17, 20, 30

Instance h-8: 40 locations

WSP location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSI location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
Client location	0,1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39
WSP Fixed Charge	6000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	8
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

DAL Performance Result:

	Heuristic
Objective Function Value	$ 95555
Run Time	5249 Seconds

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$95555	$95555
Percent Above DROP Solution	N/A	0.459182959%
Run Time	4290 Seconds	959 Seconds
WSP location	15, 17, 25, 28, 36	15, 17, 25, 28, 36
WSI location	10, 15, 17, 27, 28, 36	10, 15, 17, 27, 28, 36

Instance i-1: 20 locations

WSP location	26, 19, 2, 7, 11, 20, 13, 15, 14, 21, 1, 17, 18, 3, 27, 8, 38, 5, 4, 36
WSI location	36, 16, 19, 13, 26, 1, 31, 33, 17, 37, 18, 4, 34, 38, 7, 10, 5, 0, 15, 29
Client location	39, 37, 3, 36, 13, 7, 23, 26, 5, 9, 30, 27, 4, 8, 32, 1, 2, 29, 38, 25
WSP Fixed Charge	1000
WSI Fixed Cost	6000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 436499	$ 436499
Percent Above Optimal Solution	0.0%	N/A
Run Time	54 Seconds	1 Second
WSP location	27, 38, 5	27, 38, 5
WSI location	36, 16, 13, 26, 31, 33, 17, 4, 10, 0, 15	36, 16, 13, 26, 31, 33, 17, 4, 10, 0, 15

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 436499	$ 436499
Percent Above DROP Solution	N/A	0.0%
Run Time	38 Seconds	16 Second
WSP location	27, 38, 5	27, 38, 5
WSI location	36, 16, 13, 26, 31, 33, 17, 4, 10, 0, 15	36, 16, 13, 26, 31, 33, 17, 4, 10, 0, 1510, 8

Instance i-2: 20 locations

WSP location	11, 27, 36, 0, 12, 6, 21, 24, 30, 8, 32, 9, 29, 19, 3, 37, 25, 2, 15, 39
WSI location	14, 15, 12, 23, 7, 13, 5, 16, 2, 20, 28, 11, 35, 21, 29, 30, 17, 8, 1, 37
Client location	21, 24, 12, 14, 36, 26, 38, 31, 4, 32, 0, 29, 17, 33, 9, 15, 28, 11, 2, 6
WSP Fixed Charge	2000
WSI Fixed Cost	4000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$ 292710	$ 292710
Percent Above Optimal Solution	0.0%	N/A
Run Time	60 Seconds	2 Seconds
WSP location	12, 21, 10	12, 21, 10
WSI location	7, 11, 17, 34, 38, 14, 23	7, 11, 17, 34, 38, 14, 23

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$ 292710	$ 292710
Percent Above DROP Solution	N/A	0.0%
Run Time	43 Seconds	17 Second
WSP location	12, 21, 10	12, 21, 10
WSI location	7, 11, 17, 34, 38, 14, 23	7, 11, 17, 34, 38, 14, 23

Instance i-3: 20 locations

WSP location	37, 4, 7, 6, 38, 15, 21, 20, 16, 5, 29, 24, 18, 22, 36, 33, 23, 13, 28, 32
WSI location	4, 13, 6, 36, 12, 0, 15, 35, 8, 29, 5, 18, 33, 1, 7, 2, 31, 9, 30, 16
Client location	14, 10, 21, 20, 11, 32, 17, 33, 2, 3, 25, 19, 16, 1, 5, 0, 6, 34, 4, 9
WSP Fixed Charge	4000
WSI Fixed Cost	2000
Communication cost/ Distance to cost devisor	50
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$390179	$390179
Percent Above Optimal Solution	0.0%	N/A
Run Time	52 Seconds	1 Second
WSP location	33, 32	33, 32
WSI location	4, 15, 35, 29, 2, 31	4, 15, 35, 29, 2, 31

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$390179	$390179
Percent Above DROP Solution	N/A	0.0%
Run Time	39 Seconds	13 Seconds
WSP location	33, 32	33, 32
WSI location	4, 15, 35, 29, 2, 31	4, 15, 35, 29, 2, 31

Instance i-4: 20 locations

WSP location	10, 2, 20, 39, 24, 9, 29, 23, 17, 31, 6, 32, 27, 11, 4, 7, 22, 33, 35, 5
WSI location	16, 11, 20, 30, 5, 23, 14, 8, 32, 15, 0, 24, 27, 12, 39, 7, 21, 4, 22, 1
Client location	26, 8, 37, 4, 20, 3, 31, 30, 19, 27, 12, 17, 15, 18, 29, 33, 6, 39, 34, 16
WSP Fixed Charge	6000
WSI Fixed Cost	1000
Communication cost/ Distance to cost devisor	200
Dimensions of Service Vectors	2
Client Demand	1000
WSP Capacity	5000
WSI Capacity	5000
Value of each WSP Service Vector element	2
Value of each client Vector element	1
# of component transferred of each sharing	1
Percentage of changes	20%

Comparative Result between Heuristic and CPLEX:

	Heuristic	CPLEX
Objective Function Value	$414869	$414504
Percent Above Optimal Solution	0.088057051%	N/A
Run Time	49 Seconds	5 Seconds
WSP location	4, 7	4, 7
WSI location	16, 14, 12, 1	11, 14, 12, 1

Comparative Result between DROP and ALA:

	DROP	ALA
Objective Function Value	$420392	$414869
Percent Above DROP Solution	N/A	1.33126%
Run Time	39 Seconds	10 Second
WSP location	4, 7	4, 7
WSI location	16, 14, 8, 12	16, 14, 12, 1

APPENDIX D
EXPERIMENTAL STUDY RESULTS FOR PARAMETOR STUDY

Common Setup for All Parameter Test:

WSP Fixed Fee	1000
WSI Fixed Cost	1000
Communication Cost/ Distance to Cost Denominator	100
Dimension of Web Services Vectors	2
Client Demand	1000
WSP Capacity	5000000
WSI Capacity	5000000
Value of Each Element in WSP Web Services Vector	2
Value of Each Element in Client Web Services Vector	3
Amount of Web Services Shared per Sharing	5

Communication cost Test 1.

Problem Set up

WSP location	37, 6, 11, 27, 24, 14, 38, 20, 39, 8, 17, 18, 22, 13, 21, 29, 36
WSI location	27, 38, 29, 21, 39, 37, 8, 35, 16, 0, 13, 20, 36, 4, 11, 19, 28
Client location	33, 15, 19, 22, 18, 25, 24, 30, 17, 8, 7, 32, 2, 23, 9, 10, 3

Solutions

Denominator for communication cost	WSP locations	WSI locations	Sharing among WSI
1	12	20	3704
2	12	20	8355
4	12	20	8172
8	11	20	12144
16	7	10	1670
32	7	10	1684
64	7	7	0
128	7	7	0
256	4	5	50
512	2	4	8909

Communication cost Test 2.

Problem Set up

WSP location	18, 35, 37, 22, 32, 3, 16, 26, 25, 17, 30, 9, 21, 2, 24, 29, 13
WSI location	21, 24, 13, 28, 8, 39, 35, 11, 0, 38, 37, 14, 2, 12, 17, 36, 23
Client location	27, 32, 19, 37, 12, 36, 15, 29, 35, 10, 30, 5, 34, 20, 8, 17, 25

Solutions

Denominator for communication cost	WSP locations	WSI locations	Sharing among WSI
1	8	9	2485
2	8	9	2485
4	8	9	2483
8	9	15	7268
16	7	15	6288
32	7	9	4309
64	7	9	4308
128	5	6	2524
256	3	5	9997
512	2	3	40

Communication Cost Test 3.

Problem Set up

WSP location	25, 7, 39, 14, 26, 32, 21, 2, 11, 4, 12, 1, 6, 29, 36, 10, 27
WSI location	32, 16, 29, 30, 35, 0, 10, 31, 7, 13, 26, 20, 34, 9, 14, 21, 12
Client location	38, 33, 29, 28, 17, 5, 26, 16, 23, 36, 10, 13, 37, 2, 9, 30, 3

Solutions

Denominator for communication cost	WSP locations	WSI locations	Sharing among WSI
1	8	8	0
2	8	10	4950
4	8	16	6950
8	7	13	7445
16	7	9	6929
32	5	6	4001
64	5	6	3999
128	4	5	3332
256	3	5	3334
512	2	4	2020

Communication Cost Test 4.

Problem Set up

WSP location	19, 33, 36, 37, 6, 18, 8, 30, 35, 24, 2, 20, 11, 4, 5, 22, 1
WSI location	26, 19, 34, 18, 7, 22, 39, 33, 2, 38, 25, 14, 12, 17, 27, 21, 3
Client location	32, 25, 39, 12, 6, 8, 23, 26, 7, 34, 0, 10, 31, 14, 3, 9, 17

Solutions

Denominator for communication cost	WSP locations	WSI locations	Sharing among WSI
1	8	16	10950
2	8	16	12599
4	9	16	10280
8	8	16	12600
16	8	11	4664
32	8	11	5674
64	7	7	1970
128	5	9	1159
256	3	6	1020
512	2	3	0

Communication Cost Test 5.

Problem Set up

WSP location	6, 28, 14, 4, 5, 27, 30, 2, 9, 18, 35, 12, 15, 34, 7, 33, 10
WSI location	9, 16, 30, 38, 11, 19, 23, 34, 29, 33, 20, 4, 25, 36, 22, 26, 31
Client location	19, 14, 7, 32, 36, 29, 20, 27, 28, 23, 39, 8, 1, 21, 13, 30, 17

Solutions

Denominator for communication cost	WSP locations	WSI locations	Sharing among WSI
1	7	14	1990
2	7	14	1990
4	7	7	1980
8	7	11	3999
16	7	11	3998
32	7	7	1990
64	6	6	1035
128	5	5	1050
256	3	3	0
512	2	3	0

WSI and WSP Fixed Cost Test 1.

Problem Set up

WSP location	37, 6, 11, 27, 24, 14, 38, 20, 39, 8, 17, 18, 22, 13, 21, 29, 36
WSI location	27, 38, 29, 21, 39, 37, 8, 35, 16, 0, 13, 20, 36, 4, 11, 19, 28
Client location	33, 15, 19, 22, 18, 25, 24, 30, 17, 8, 7, 32, 2, 23, 9, 10, 3

Solutions

WSI and WSP fixed cost	WSP locations	WSI locations	Sharing among WSI
100	8	8	0
200	8	8	0
400	8	8	0
800	8	8	0
1600	7	7	0
3200	5	5	0
6400	3	3	0
12800	2	2	0
25600	1	1	0

WSI and WSP Fixed Cost Test 2.

Problem Set up

WSP location	18, 35, 37, 22, 32, 3, 16, 26, 25, 17, 30, 9, 21, 2, 24, 29, 13
WSI location	21, 24, 13, 28, 8, 39, 35, 11, 0, 38, 37, 14, 2, 12, 17, 36, 23
Client location	27, 32, 19, 37, 12, 36, 15, 29, 35, 10, 30, 5, 34, 20, 8, 17, 25

Solutions

WSI and WSP fixed cost	WSP locations	WSI locations	Sharing among WSI
100	7	7	0
200	7	7	0
400	7	7	0
800	7	7	0
1600	7	7	0
3200	6	6	0
6400	3	3	0
12800	2	2	0
25600	2	2	0

WSI and WSP Fixed Cost Test 3.

Problem Set up

WSP location	25, 7, 39, 14, 26, 32, 21, 2, 11, 4, 12, 1, 6, 29, 36, 10, 27
WSI location	32, 16, 29, 30, 35, 0, 10, 31, 7, 13, 26, 20, 34, 9, 14, 21, 12
Client location	38, 33, 29, 28, 17, 5, 26, 16, 23, 36, 10, 13, 37, 2, 9, 30, 3

Solutions

WSI and WSP fixed cost	WSP locations	WSI locations	Sharing among WSI
100	6	6	0
200	6	6	0
400	6	6	0
800	5	5	0
1600	4	4	0
3200	4	4	0
6400	4	4	0
12800	2	1	0
25600	1	1	0

WSI and WSP Fixed Cost Test 4.

Problem Set up

WSP location	19, 33, 36, 37, 6, 18, 8, 30, 35, 24, 2, 20, 11, 4, 5, 22, 1
WSI location	26, 19, 34, 18, 7, 22, 39, 33, 2, 38, 25, 14, 12, 17, 27, 21, 3
Client location	32, 25, 39, 12, 6, 8, 23, 26, 7, 34, 0, 10, 31, 14, 3, 9, 17

Solutions

WSI and WSP fixed cost	WSP locations	WSI locations	Sharing among WSI
100	8	16	113334
200	6	11	110820
400	6	11	110820
800	5	10	125820
1600	3	8	140820
3200	3	8	140820
6400	2	2	0
12800	2	2	0
25600	2	2	0

WSI and WSP Fixed Cost Test 5.

Problem Set up

WSP location	6, 28, 14, 4, 5, 27, 30, 2, 9, 18, 35, 12, 15, 34, 7, 33, 10
WSI location	9, 16, 30, 38, 11, 19, 23, 34, 29, 33, 20, 4, 25, 36, 22, 26, 31
Client location	19, 14, 7, 32, 36, 29, 20, 27, 28, 23, 39, 8, 1, 21, 13, 30, 17

Solutions

WSI and WSP fixed cost	WSP locations	WSI locations	Sharing among WSI
100	6	6	0
200	6	6	0
400	5	5	0
800	5	5	0
1600	5	5	0
3200	5	5	0
6400	3	3	0
12800	2	2	0
25600	2	2	0

Client Demand Test 1.

Problem Set up

WSP location	37, 6, 11, 27, 24, 14, 38, 20, 39, 8, 17, 18, 22, 13, 21, 29, 36
WSI location	27, 38, 29, 21, 39, 37, 8, 35, 16, 0, 13, 20, 36, 4, 11, 19, 28
Client location	33, 15, 19, 22, 18, 25, 24, 30, 17, 8, 7, 32, 2, 23, 9, 10, 3

Solutions

Client Demand	WSP locations	WSI locations	Sharing among WSI
100	2	1	0
200	4	3	200
400	4	3	0
800	5	5	0
1600	6	8	4797
3200	6	9	11734
6400	6	17	68264
12800	6	17	136536
25600	7	17	273062

Client Demand Test 2.

Problem Set up

WSP location	18, 35, 37, 22, 32, 3, 16, 26, 25, 17, 30, 9, 21, 2, 24, 29, 13
WSI location	21, 24, 13, 28, 8, 39, 35, 11, 0, 38, 37, 14, 2, 12, 17, 36, 23
Client location	27, 32, 19, 37, 12, 36, 15, 29, 35, 10, 30, 5, 34, 20, 8, 17, 25

Solutions

Client Demand	WSP locations	WSI locations	Sharing among WSI
100	4	1	0
200	3	3	0
400	3	3	0
800	5	6	1335
1600	6	7	5334
3200	6	7	10668
6400	8	16	23469
12800	8	16	46938
25600	8	16	93862

Client Demand Test 3.

Problem Set up

WSP location	25, 7, 39, 14, 26, 32, 21, 2, 11, 4, 12, 1, 6, 29, 36, 10, 27
WSI location	32, 16, 29, 30, 35, 0, 10, 31, 7, 13, 26, 20, 34, 9, 14, 21, 12
Client location	38, 33, 29, 28, 17, 5, 26, 16, 23, 36, 10, 13, 37, 2, 9, 30, 3

Solutions

Client Demand	WSP locations	WSI locations	Sharing among WSI
100	2	2	0
200	2	2	0
400	2	2	0
800	3	3	0
1600	3	4	7999
3200	3	5	15999
6400	4	6	31998
12800	4	6	63998
25600	4	16	435200

Client Demand Test 4.

Problem Set up

WSP location	19, 33, 36, 37, 6, 18, 8, 30, 35, 24, 2, 20, 11, 4, 5, 22, 1
WSI location	26, 19, 34, 18, 7, 22, 39, 33, 2, 38, 25, 14, 12, 17, 27, 21, 3
Client location	32, 25, 39, 12, 6, 8, 23, 26, 7, 34, 0, 10, 31, 14, 3, 9, 17

Solutions

Client Demand	WSP locations	WSI locations	Sharing among WSI
100	4	2	1499
200	4	4	2998
400	4	9	42998
800	4	9	85999
1600	4	9	171999
3200	4	9	343999
6400	4	9	687998
12800	5	15	1501864
25600	5	15	3003739

Client Demand Test 5.

Problem Set up

WSP location	6, 28, 14, 4, 5, 27, 30, 2, 9, 18, 35, 12, 15, 34, 7, 33, 10
WSI location	9, 16, 30, 38, 11, 19, 23, 34, 29, 33, 20, 4, 25, 36, 22, 26, 31
Client location	19, 14, 7, 32, 36, 29, 20, 27, 28, 23, 39, 8, 1, 21, 13, 30, 17

Solutions

Client Demand	WSP locations	WSI locations	Sharing among WSI
100	2	2	499
200	4	3	598
400	4	4	800
800	5	6	4798
1600	5	7	9600
3200	5	7	19199
6400	7	17	86399
12800	7	17	172799
25600	7	17	345599

Amount of Component Shared in One Transaction among WSI Servers Test 1.

Problem Set up

WSP location	37, 6, 11, 27, 24, 14, 38, 20, 39, 8, 17, 18, 22, 13, 21, 29, 36
WSI location	27, 38, 29, 21, 39, 37, 8, 35, 16, 0, 13, 20, 36, 4, 11, 19, 28
Client location	33, 15, 19, 22, 18, 25, 24, 30, 17, 8, 7, 32, 2, 23, 9, 10, 3

Solutions

Amount of Component Shared	WSP locations	WSI locations	Sharing among WSI
1	7	7	0
2	7	7	0
3	7	8	1009
4	7	8	1022
5	7	8	1023
6	7	8	1021
7	7	8	1021
8	7	8	4986
9	7	8	4988
10	7	8	4983

Amount of Component Shared in One Transaction among WSI Servers Test 2.

Problem Set up

WSP location	18, 35, 37, 22, 32, 3, 16, 26, 25, 17, 30, 9, 21, 2, 24, 29, 13
WSI location	21, 24, 13, 28, 8, 39, 35, 11, 0, 38, 37, 14, 2, 12, 17, 36, 23
Client location	27, 32, 19, 37, 12, 36, 15, 29, 35, 10, 30, 5, 34, 20, 8, 17, 25

Solutions

Amount of Component Shared	WSP locations	WSI locations	Sharing among WSI
1	5	5	0
2	5	5	0
3	6	7	1514
4	6	7	1531
5	6	7	2039
6	6	8	6514
7	6	9	16081
8	6	10	22050
9	6	10	27000
10	6	10	27000

Amount of Component Shared in One Transaction among WSI Servers Test 3.

Problem Set up

WSP location	25, 7, 39, 14, 26, 32, 21, 2, 11, 4, 12, 1, 6, 29, 36, 10, 27
WSI location	32, 16, 29, 30, 35, 0, 10, 31, 7, 13, 26, 20, 34, 9, 14, 21, 12
Client location	38, 33, 29, 28, 17, 5, 26, 16, 23, 36, 10, 13, 37, 2, 9, 30, 3

Solutions

Amount of Component Shared	WSP locations	WSI locations	Sharing among WSI
1	5	5	0
2	5	7	5010
3	5	7	5024
4	5	7	7058
5	5	7	10956
6	5	7	17017
7	5	7	17016
8	5	7	23018
9	5	7	23017
10	5	7	28016

Amount of Component Shared in One Transaction among WSI Servers Test 4.

Problem Set up

WSP location	19, 33, 36, 37, 6, 18, 8, 30, 35, 24, 2, 20, 11, 4, 5, 22, 1
WSI location	26, 19, 34, 18, 7, 22, 39, 33, 2, 38, 25, 14, 12, 17, 27, 21, 3
Client location	32, 25, 39, 12, 6, 8, 23, 26, 7, 34, 0, 10, 31, 14, 3, 9, 17

Solutions

Amount of Component Shared	WSP locations	WSI locations	Sharing among WSI
1	5	6	25
2	5	6	26
3	5	6	23
4	5	6	23
5	5	6	48
6	5	6	18
7	5	6	59
8	5	6	20
9	5	6	60
10	6	6	1999

Amount of Component Shared in One Transaction among WSI Servers Test 5.

Problem Set up

WSP location	6, 28, 14, 4, 5, 27, 30, 2, 9, 18, 35, 12, 15, 34, 7, 33, 10
WSI location	9, 16, 30, 38, 11, 19, 23, 34, 29, 33, 20, 4, 25, 36, 22, 26, 31
Client location	19, 14, 7, 32, 36, 29, 20, 27, 28, 23, 39, 8, 1, 21, 13, 30, 17

Solutions

Amount of Component Shared	WSP locations	WSI locations	Sharing among WSI
1	4	4	0
2	3	4	2029
3	3	4	2027
4	3	5	4039
5	3	5	4039
6	3	5	13970
7	3	5	15948
8	3	7	26889
9	2	11	62828
10	2	17	76780

LIST OF REFERENCES

Akinc, U. and B. M. Khumawala (1977). "An Efficient Branch and Bound Algorithm for the Capacitated Warehouse Location Problem." Management Science **23**(6): 585-594.

Balinski, M. L. (1965). "Integer Programming: Methods, Uses, Computation." Management Science **12**(3): 253-313.

Barford, P., J.-Y. Cai and J. Gast (2001). Cache Placement Methods Based on Client Demand Clustering. Technical report TR1437, University of Wisconsin at Madison, Madison, WI.

Basturk, E., R. Engel, R. Haas, D. Kandlur, V. Peris and D. Saha (1997). Using Network Layer Anycast for Load Distribution in the Internet. Technical report, IBM T.J. Watson Research Center, Yorktown Heights, NY.

Beasley, J. E. (1993). "Lagrangian Heuristics for Location Problem." European Journal of Operational Research **65**(3): 383-399.

Benders, J. F. (1962). "Partitioning Procedures for Solving Mixed-Variables Programming Problems." Numerische Mathematik **4**: 238-252.

Burbeck, S. (2000). The Tao of E-Business Services: the Evolution of Web Applications into Service-Oriented Components with Web Services. White paper, http://www-106.ibm.com/developerworks/webservices/library/ws-tao/index.html?dwzone=webservices (accessed June 2002).

Cardellini, V., M. Colajanni and P. S. Yu (1999). "Dynamic Load Balancing on Web Server Systems." IEEE Internet Computing **3**(3): 28-39.

Carter, R. L. and M. E. Crovella (1996). Dynamic Server Selection Using Bandwidth Probing in Wide-Area Networks. Technical report BU-CS-96-007, Boston University, Boston, MA.

Charikar, M. and S. Guha (1999). Improved Combinatorial Algorithms for the Facility Location and K-Median Problems. 40th Annual IEEE Conference on Foundations of Computer Science, New York, NY.

Christofides, N. and J. E. Beasley (1983). "Extensions to a Lagrangean Relaxation Approach for the Capacitated Warehouse Location Problem." European Journal of Operational Research **12**(1): 19-28.

Church, R. and C. Reveille (1994). "The Maximal Coverage Location Problem." Papers of the Regional Science Association **32**: 101-118.

Cidon, I., S. Kutten and R. Soffer (2001). Optimal Allocation of Electronic Content. IEEE INFOCOM, Anchorage, AK.

Clements, T. (2001). Best Practices Web Services-Technical Overviews, Overview of SOAP. White paper, http://dcb.sun.com/practices/webservices/overviews/overview_soap.jsp (accessed July 2002).

Comer, D. E. (2001). Computer Networks and Internets (with Internet Applications). Prentice-Hall, Inc., Upper Saddle River, NJ.

Cooper, L. (1964). "Heuristic Methods for Location-Allocation Problems." SIAM Review **6**(1): 37-53.

Cornuejols, G., R. Sridharan and J. M. Thizy (1991). "A Comparison of Heuristics and Relaxations for the Capacitated Plant Location Problem." European Journal of Operational Research **50**(3): 280-297.

Crovella, M. E. and R. L. Carter (1995). Dynamic Server Selection in the Internet. 3rd. IEEE Workshop on the Architecture and Implementation of High Performance Communication Subsystems (HPCS' 95), Mystic, CT.

Current, J., M. Daskin and D. Schilling (2002). Discrete Network Location Models. Facility Location: Applications and Theory. Z. Drezner and H. Hamacher. Berlin, Germany, Springer Verlag**:** 81-118.

Davis, P. S. and T. L. Ray (1969). "A Branch-and-Bound Algorithm for Capacitated Facilities Location Problem." Naval Research Logistics Quarterly **16**(3): 331-344.

Dearing, P. M., R. L. Francis and T. J. Lowe (1976). "Convex Location Problems on the Networks." Operations Research **24**(4): 628-642.

Dijkstra, E. (1959). "A Note on Two Problems in Connexion with Graphs." Numerische Mathematik **1**: 269-271.

Domschke, W. and A. Drexl (1985). "ADD-Heuristics' Starting Procedures for Capacitated Plant Location Models." European Journal of Operational Research **21**(1): 47-53.

Ellwein, L. B. and P. Gray (1971). "Solving Fixed Charge Location-Allocation Problems with Capacity and Configuration Constraints." AIIE Transactions **3**(4): 290-299.

Elson, D. G. (1972). "Site Location via Mixed-Integer Programming." Operational Research Quarterly **23**: 31-43.

Farrell, J. (2002). Web Services Interoperability between the WebSphere and .Net platforms. White paper, http://www-106.ibm.com/developerworks/webservices/library/i-wasnet/ (accessed September 2002).

Fei, Z., S. Bhattacharjee, E. W. Zegura and M. H. Ammar (1998). A Novel Server Selection Technique for Improving the Response Time of a Replicated Service. IEEE INFOCOM, San Francisco, CA.

Feldman, E., F. A. Lehrer and T. L. Ray (1966). "Warehouse Location under Continuous Economies of Scale." Management Science **12**(9): 670-684.

Fisher, M. L. (1981). "The Lagrangian Relaxation Method for Solving Integer Programming Problems." Management Science **27**(1): 1-18.

Fisher, M. L. (1985). "An Applications Oriented Guide to Lagrangian Relaxation." Interfaces **15**(2): 2-21.

Francis, P., S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt and L. Zhang (2001). "IDMaps: A Global Internet Host Distance Estimation Service." IEEE/ACM Transactions on Networking **9**(5): 525-540.

Francis, P., S. Jamin, V. Paxson, L. Zhang, D. F. Gryniewicz and Y. Jin (1999). An Architecture for a Global Internet Host Distance Estimation Service. IEEE INFOCOM, New York, NY.

Francis, R. L., L. F. McGinnis and J. A. White (1992). Facilities Layout and Location: An Analytical Approach. Prentice-Hall, Englewood Cliffs, NJ.

Francis, R. L., L. F. J. McGinnis and J. A. White (1983). "Locational Analysis." European Journal of Operational Research **12**(2): 220-252.

Garey, M. R. and D. S. Johnson (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York, NY.

Gendron, B. and J. Y. Potvin (1999). "Tabu Search with Exact Neighborhood Evaluation for Multicommodity Location with Balancing Requirements." INFOR **37**: 255-269.

Geoffrion, A. M. and G. W. Graves (1974). "Multicommodity Distribution System Design by Benders Decomposition." Management Science **20**(5): 822-844.

Geoffrion, A. M. and R. E. Marsten (1972). "Integer Programming Algorithms a Framework and State-of-the-Art Survey." Management Science **18**(7): 465-491.

Geoffrion, A. M. and R. McBride (1978). "Lagrangean Relaxation Applied to Capacitated Facility Location Problems." AIIE Transactions **14**(1): 40-47.

Glover, F. (1989). "Tabu Search Part I." ORSA Journal on Computing **1**: 190 -206.

Glover, F. (1990). "Tabu Search Part II." ORSA Journal on Computing **2**: 4-32.

Goldman, A. J. (1971). "Optimal Center Location in Simple Networks." Transportation Science **5**: 212-221.

Gottschalk, K., S. Graham, H. Kreger and J. Snell (2002). "Introduction to Web Services Architecture." IBM Systems Journal **41**(2): 170-177.

Guyton, J. D. and M. F. Schwartz (1995). Locating Nearby Copies of Replicated Internet Servers. Technical report CU-CS-762-95, University of Colorado**,** Boulder, CO.

Gwertzman, J. and M. Seltzer (1995). An Analysis of Geographical Push-Caching. In Proceedings of the 5th IEEE Workshop on Hot Topics in Operating Systems, Orcas Island, WA.

Hakimi, S. L. (1964). "Optimum Locations of Switching Centers and the Absolute Centers and Median of a Graph." Operations Research **12**(3): 450-459.

Hakimi, S. L. (1965). "Optimum Distribution of Switching Centers in a Communication Network and Some Related Graph Theoretic Problems." Operations Research **13**(3): 462-475.

Hindi, K. S. and T. Basta (1994). "Computationally Efficient Solution of a Multiproduct, Two-Stage Distribution-Location Problem." The Journal of the Operational Research Society **45**(11): 1316-1323.

Hooker, J. N., R. S. Garfinkel and C. K. Chen (1991). "Finite Dominating Sets For Network Location Models." Operations Research **39**(1): 100-118.

Hua, L. K. and Others (1962). "Applications of Mathematical Methods to Wheat Harvesting." Chinese Math **2**: 77-91.

IBM (n.d.). http://alphaworks.ibm.com/tech/webservicestoolkit (accessed May 2002).

ILOG (2002). ILOG CPLEX 6.5 User's Manual.

Internet Backbone Maps (n.d.). http://www.nthelp.com/maps.htm (accessed July 2002).

Jacobsen, S. K. (1983). "Heuristics for the Capacitated Plant Location Model." European Journal of Operational Research **12**(2): 253-261.

Jamin, S., C. Jin, Y. Jin, D. Raz, Y. Shavitt and L. Zhang (2000). On the Placement of Internet Instrumentation. IEEE INFOCOM, Tel Aviv, Israel.

Jamin, S., C. Jin, A. R. Kurc, D. Raz and Y. Shavitt (2001). Constrained Mirror Placement on the Internet. IEEE INFOCOM, Anchorage, AK.

Johansson, J. M. (2000). "On the Impact of Network Latency on Distributed Systems Design." Information Technology Management **1**: 183-194.

Kangasharju, J., J. Roberts and K. W. Ross (2001). Object Replication Strategies in Content Distribution Networks. Web Caching and Content Distribution Workshop (WCW'01), Boston, MA.

Kariv, O. and S. L. Hakimi (1979a). "An Algorithmic Approach to Network LocationProblems, I: The *p*-Centers." SIAM Journal on Applied Mathematics **37**: 513-538.

Kariv, O. and S. L. Hakimi (1979b). "An Algorithmic Approach to Network LocationProblems, II: The *p*-Medians." SIAM Journal on Applied Mathematics **37**: 539-560.

Khumawala, B. M. (1974). "An Effcient Heuristic Procedure for the Capacitated Warehouse Location Problem." Naval Research Logistics Quarterly **21**: 609-623.

Klincewicz, J. G. (1985). "A Large-Scale Distribution and Location Model." AT&T Technical Journal **64**: 1705-1730.

Klincewicz, J. G. and H. Luss (1986). "A Lagrangean Relaxation Heuristic for Capacitated Facility Location with Single-Source Constraints." Journal of the Operational Research Society **37**(5): 495–500.

Klose, A. (1999). "An LP-Based Heuristic for Two-Stage Capacitated Facility Location Problems." Journal of the Operational Research Society **50**(2): 157-166.

Klose, A. (2000). "A Lagrangean Relax-and-Cut Approach for the Two-Stage Capacitated Facility Location Problem." European Journal of Operational Research **126**(2): 408-421.

Kolen, A. J. W. (1982). Location Problems on Trees and the Rectilinear Plane. Stitchting Mathematisch Centrum, Amsterdam, The Netherlands.

Kreger, H. (2001). Web Services Conceptual Architecture (WSCA 1.0). White paper, http://www-3.ibm.com/software/solutions/webservices/pdf/WSCA.pdf (accessed May 2002).

Krishnan, P., D. Raz and Y. Shavitt (2000). "The Cache Location Problem." IEEE/ACM Transactions on Networking **8**(2): 568 - 582.

Kuehn, A. A. and M. J. Hamburger (1963). "A Heuristic Program for Locating Warehouses." Management Science **9**(4): 643-666.

Land, A. and A. Doig (1960). "An Automatic Method of Solving Discrete Programming Problems." Econometrica **28**: 497-520.

Lau, C. and A. Ryman (2002). "Developing XML Web Services with WebSphere Studio Application Developer." IBM Systems Journal **41**(2): 178-197.

Laundy, R. S. (1994). "A Tree Search Algorithm for the Multi-Commodity Location Problem." European Journal of Operational Research **20**(3): 344-351.

Levine, B. N. and J. J. Garcia-Luna-Aceves (1997). Improving Internet Multicast with Routing Labels. International Conference on Network Protocols, Atlanta, GA.

Li, B., M. J. Golin, G. F. Italiano and X. Deng (1999). On the Optimal Placement of Web Proxies in the Internet. IEEE INFOCOM, New York, NY.

Maranzana, F. E. (1964). "On the Location of Supply Points to Minimize Transport Costs." Operational Research Quarterly **15**: 261-270.

McManus, P. R. (1999). A Passive System for Server Selection within Mirrored Resource Environments Using AS Path Length Heuristics. Technical report, AppliedTheory Corp.**,** New York, NY.

Megiddo, N., E. Zemel and S. L. Hakimi (1983). "The Maximum Coverage Location Problem." SIAM Journal on Algebraic and Discrete Methods **4**(2): 253-261.

Microsoft (n.d.). http://www.microsoft.com/india/msdn/dotnet/webservices.asp (accessed July 2002).

Minieka, E. (1970). "The *m*-Center Problem." SIAM Review **12**: 138-139.

Moore, K., J. Cox and S. Green (1996). SONAR-A Network Proximity Service. Internet draft, http://www.netlib.org/utk/projects/sonar/draft-moore-sonar-01.txt (accessed August 2002).

Nauss, R. M. (1978). "An Improved Algorithm for the Capacitated Facility Location Problem." Journal of the Operational Research Society **29**(12): 1195-1201.

Ng, T. S. E. and H. Zhang (2002). Predicting Internet Network Distance with Coordinates Based Approaches. IEEE INFOCOM, New York, NY.

Nussbaumer, J.-P., B. V. Patel, F. Schaffa and J. P. G. Sterbenz (1995). "Networking Requirements for Interactive Video on Demand." IEEE Journal on Selected Areas in Communications **13**(5): 779-787.

Obraczka, K. and F. Silva (1999). Looking at Network Latency for Server Proximity. Technical report USC-CS-99-714, Department of Computer Science, University of Southern California**,** Los Angeles, CA.

Pirkul, H. and V. Jayaraman (1998). "A Multi-Commodity, Multi-Plant, Capacitated Facility Location Problem: Formulation and Efficient Heuristic Solution." Computers & Operations Research **25**(10): 869-878.

Qiu, L., V. N. Padmanabhan and G. M. Voelker (2001). On the Placement of Web Server Replicas. IEEE INFOCOM, Anchorage, AK.

Ram, S. and S. Narasimhan (1994). "Database Allocation in a Distributed Environment: Incorporating a Concurrency Control Mechanism and Queuing Costs." Management Science **40**(8): 969-983.

Ramakrishnan, V. S. R. (2001). Web Services: OR's Newest Ally? OR/MS today. **28**(6): 37-39.

Rapp, Y. (1962). "Planning of Exchange Locations and Boundaries." Ericson Technics **2**: 1-22.

ReVelle, C. S. and R. Swain (1970). "Central Facilities Location." Geographical Analysis **2**: 30-42.

Rho, S. and S. T. March (1995). Designing Distributed Database Systems for Efficient Operation. Sixteenth International Conference on Information Systems, Amsterdam, The Netherlands.

Rolland, E., D. A. Schilling and J. R. Current (1997). "An Efficient Tabu Search Heuristic for the *p*-Median Problem." European Journal of Operational Research **96**(2): 329-342.

Ross, G. T. and R. M. Soland (1975). "A Branch and Bound Algorithm for the Generalized Assignment Problem." Mathematical Programming Studies **8**: 91-103.

Sa, G. (1969). "Branch and Bound and Approximate Solutions to the Capacitated Plant-Location Problem." Operations Research **17**(6): 1005-1016.

Sayal, M., Y. Breitbart, P. Scheuermann and R. Vingralek (1998). Selection Algorithms for Replicated Web Servers. Workshop on Internet Server Performance (WISP98), Madison, WI.

Shaikh, A., R. Tewari and M. Agrawal (2001). On the Effectiveness of DNS-Based Server Selection. IEEE INFOCOM, Anchorage, AK.

Skiena, S. S. (1997). The Algorithm Design Manual. Springer-Verlag, New York, NY.

Sridharan, R. (1993). "A Lagrangian Heuristic for the Capacitated Plant Location Problem with Single Source Constraints." European Journal of Operational Research **66**(3): 305-312.

Sridharan, R. (1995). "The Capacitated Plant Location Problem." European Journal of Operational Research **87**(2): 203-213.

Sun Microsystems (n.d. a). http://dcb.sun.com/practices/webservices/webserv_glossary.html (accessed August 2002).

Sun Microsystems (n.d. b). http://wwws.sun.com/software/sunone/ (accessed February 2003).

Tansel, B. C., R. L. Francis and T. J. Lowe (1983a). "Location of Networks: A Survey, Part I." Management Science **29**(4): 482-497.

Tansel, B. C., R. L. Francis and T. J. Lowe (1983b). "Location of Networks: A Survey, Part II." Management Science **29**(4): 498-511.

Teitz, M. B. and P. Bart (1968). "Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph." Operations Research **16**(5): 955-961.

Tragantalerngsaka, S., J. Holta and M. Rönnqvistb (1997). "Lagrangian Heuristics for the Two-Echelon, Single-Source, Capacitated Facility Location Problem." European Journal of Operational Research **102**(3): 611-625.

UDDI.org (2001). UDDI Executive White Paper. White paper, http://uddi.org/pubs/UDDI_Executive_White_Paper.pdf (accessed August 2002).

UpTown Web Hosting (n.d.). http://www.uptownwebhosting.com (accessed February 2003).

Van Roy, T. J. (1986). "A Cross Decomposition Algorithm for Capacitated Facility Location." Operations Research **34**(1): 145-163.

Wolski, R., N. Spring and J. Hayes (1999). "The Network Weather Service: A Distributed Resource Performance Forecasting Service for Metacomputing." Journal of Future Generation Computing Systems **15**(5-6): 757-768.

World Wide Web Consortium (n.d. a). Web Services Description Language (WSDL) 1.1. White paper, http://www.w3.org/TR/wsdl (accessed August 2002).

World Wide Web Consortium (n.d. b). http://www.w3.org/2002/ws/ (accessed August 2002).

World Wide Web Consortium (n.d. c). http://www.w3.org/XML/ (accessed February 2003).

BIOGRAPHICAL SKETCH

Yi Sun was born in Beijing, China, in March 1972. He received his Bachelor of Arts degree in government from Foreign Affairs College in Beijing in 1995. After completing his undergraduate studies he worked as a junior analyst for the Foreign Affairs Committee of the Chinese National People's Congress from 1995 to 1996. He received the Master of Arts degree in political science from the University of Florida in 1998. He received the Master of Science degree in business administration with a concentration in decision and information sciences from the University of Florida in 2000.

He intends to pursue an academic research and teaching career following the completion of his doctoral degree.