MULTIOBJECTIVE HUB LOCATION PROBLEM

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

ARAS BARUTÇUOĞLU

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR
THE DEGREE OF MASTER OF SCIENCE
IN
INDUSTRIAL ENGINEERING

JULY 2009

Approval of the thesis:

MULTIOBJECTIVE HUB LOCATION PROBLEM

submitted by **ARAS BARUTÇUOĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Industrial Engineering Department, Middle East Technical University** by,

Prof. Dr. Canan Özgen ________________
Dean, Graduate School of **Natural and Applied Sciences**

Prof. Dr. Nur Evin Özdemirel ________________
Head of Department, **Industrial Engineering**

Prof. Dr. Murat Köksalan ________________
Supervisor, **Industrial Engineering Dept., METU**

Examining Committee Members:

Asst. Prof. Müge Avşar ________________
Industrial Engineering Dept, METU

Prof. Dr. Murat Köksalan ________________
Industrial Engineering Dept, METU

Prof. Dr. Sencer Yeralan ________________
Agricultural and Biological Engineering, University of Florida

Asst. Prof. İsmail Bakal ________________
Industrial Engineering Dept, METU

Asst. Prof. Seçil Savaşaneril ________________
Industrial Engineering Dept, METU

Date: ________________

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name : Aras Barutçuoğlu

Signature :

ABSTRACT

MULTIOBJECTIVE HUB LOCATION PROBLEM

Barutçuoğlu, Aras
M.S. Department of Industrial Engineering
Supervisor: Prof. Dr. Murat Köksalan

July 2009, 80 pages

In this study, we propose a two-phase solution approach for approximating the efficient frontier of a bicriteria hub location problem. We develop an evolutionary algorithm to locate the hubs on the network as the first phase. In the second phase, we develop a bounding procedure based on dominance relations and using the determined bounds, we solve the allocation subproblem for each located hub set. The two-phase approach is tested on the Australian Post data set and it is observed that our approach approximates the entire efficient frontier well. In addition, we suggest an interactive procedure to find the solutions that are in the decision maker's preferred region of the solution space. In this procedure, we progressively incorporate the preferences of the decision maker and direct the search towards the preferred regions. Based on some computational experiments, it is observed that the interactive procedure converges to the preferred regions.

Keywords: Multiobjective Evolutionary Algorithms, Hub Location, Multiobjective Hub Location

ÖZ

ÇOK AMAÇLI MERKEZ ÜSSÜ YER SEÇİMİ PROBLEMİ

Barutçuoğlu, Aras
Yüksek Lisans, Endüstri Mühendisliği Bölümü
Tez Yöneticisi: Prof. Dr. Murat Köksalan

Temmuz 2009, 80 Sayfa

Bu çalışmada iki amaçlı merkez üssü yer seçimi probleminin etkin yüzeyine yaklaşmayı amaçlayan iki aşamalı bir çözüm yaklaşımı önerilmiştir. Birinci aşamada merkez üslerinin ağ üzerindeki yerlerini belirlemek için bir evrimci algoritma geliştirilmiştir. İkinci aşamada ise, baskınlık ilişkilerine dayalı bir sınırlama prosedürü geliştirilmiş ve yerleri belirlenen merkez üslerinin atama alt problemleri belirlenen sınırlara göre çözdürülmüştür. İki aşamalı yaklaşım Avusturalya Postası verileriyle denenmiş ve yaklaşımın etkin yüzeylere başarıyla yaklaştığı görülmüştür. Buna ek olarak, karar vericinin ilgilendiği alanlardaki çözümleri bulmak için etkileşimli bir prosedür önerilmiştir. Bu prosedürde, karar vericinin tercihleri kademeli olarak kullanılmakta ve arama ilgilenilen alanlara doğru yönlendirilmektedir. Yapılan bazı testler sonucunda, etkileşimli prosedürün karar vericinin ilgilendiği alanlara yakınsadığı gözlenmiştir.

Anahtar Kelimeler: Çok Amaçlı Evrimci Algoritmalar, Merkez Üssü Yer Seçimi, Çok Amaçlı Merkez Üssü Yer Seçimi

To my mom and dad

ACKNOWLEDGMENTS

First, I would like to express my gratitude to my supervisor, Prof. Dr. Murat Köksalan for guiding and encouraging me throughout this study. I am very happy that I had the chance to work with such an excellent research person.

I am also grateful to Prof. Dr. Meral Azizoğlu for her sincere support, encouragement, and friendliness. I also owe thanks to Prof. Dr. Nur Evin Özdemirel and Assoc. Prof. Dr. Canan Sepil for their smiling face and understanding. I would also like to thank jury members for their valuable comments and contributions on this thesis.

I would like to thank to my friends Melih Çelik, Büşra Atamer, Banu Lokman and Tülin İnkaya for their sympathy. Although their names do not appear individually here, I am grateful to all my professors for their valuable contributions to my academic life.

I am grateful to my parents Zeynep Barutçuoğlu and Ömer Barutçuoğlu for their endless love, morale support, and encouragement. This work would not be possible without them.

Finally, my deepest thanks go to my lovely wife, Şirin, for all her love, patience, and tireless efforts.

TABLE OF CONTENTS

LIST OF TABLES

TABLES

LIST OF FIGURES

FIGURES

CHAPTER 1

INTRODUCTION

Hubs are central facilities that act as consolidation, sorting and switching points in many-to-many distribution systems, where flows of mail, passenger, or information are directed from origins to destinations. The set of hubs that are located on the network is called hub set. Hub networks offer economies of scale via concentration of flows between fully-interconnected hub facilities in the hub set, eliminating the need for establishing a direct link between each origin-destination pair. Concentration of flow on a small number of hubs is enabled with larger, faster and high capacity transportation media available in inter-hub transfers. In hub networks, flows of origins (collection flows) are first consolidated at the hubs assigned to these origins, then the flows are directed to hubs assigned to destinations of these flows (transfer flows), and lastly they are distributed to the destinations (distribution flows). Thus, all flows have to be directed via either one hub or two hubs (depending on the identicalness of the two hubs assigned to origin and destination points), not allowing any direct link between non-hub nodes in the network.

The two main decisions in hub location problems (HLP) are concerned with locating the hubs on the network (location subproblem) and allocating non-hub nodes to the located hubs (allocation subproblem). However, these two subproblems are not independent since the two decisions heavily interact with each other when designing a hub network. This interaction adds more difficulty to the combinatorial nature of HLPs.

HLPs have many applications in businesses such as postal/cargo delivery systems, air transportation, telecommunication systems, and emergency services (Ebery et al., 2000). In postal/cargo delivery systems, the speed of delivery is at least as important as the cost of delivery (Tan and Kara, 2007). Thus, time-efficiency should also be considered as another criterion besides the traditional total cost minimization objective in hub network design of these systems. In these systems, sorting of the incoming mail from non-hub nodes to hub nodes is the main time consuming operation that is done at hub facilities and hence, this operation accounts for the main capacity restriction at hubs (Ernst and Krishnamoorthy, 1999). Then, designing the network considering the total time spent for sorting operation besides total cost minimization would yield promising results for postal/cargo delivery systems.

In this study, we consider a bicriteria hub location problem for a postal/cargo delivery system. The first objective is the minimization of the total cost to serve the network (total transportation cost and fixed hubbing cost) and the second objective is the minimization of the total time to process the flows at hubs due to sorting operations as in Costa et al. (2008). We assume that non-hub nodes can only be assigned to a single hub and the number of hubs to be opened is a decision variable to be determined using fixed costs of opening a hub. Different than Costa et al. (2008), we try to approximate the entire set of efficient solutions for relatively more difficult problem instances. We propose an evolutionary algorithm that locates the hubs on the network using representative allocation structures of each hub set similar to Soylu and Köksalan (2006). Then for each located hub set, we find an efficient way to solve the allocation problem to avoid overlaps in generating nondominated solutions. We solve the allocation subproblem with Cplex using bounds determined based on dominance relationships.

In Chapter 2, we review the related literature. In Chapter 3, we give the definition and the mathematical formulation of the problem. In Chapter 4, we give the solution approach, which tries to approximate the entire efficient frontier. In Chapter 5, we propose an interactive procedure to converge to the preferred region of the decision maker. Lastly, we give the conclusions and the future research directions in Chapter 6.

CHAPTER 2

LITERATURE REVIEW

HLPs have been the focus of many researchers because there are many applications of these problems to distribution networks and implementable results can be obtained, which promise economies of scale via inter-hub transfers. HLPs can be categorized into two main headings with respect to allocation restrictions. Single allocation problems limit assignment of a non-hub node only to one hub, whereas multiple allocation problems do not, so that each non-hub node can be assigned to more than one hub. Another classification of HLPs is done according to the existence of capacity restrictions. There can be capacities on the flows that flow through a hub or on the links (arcs) in the hub network. In addition, the number of hubs to be opened (p) can be known in advance -making the problem a p-hub location problem- or not. If p is not known a priori, then the solution procedure determines how many hubs to open using fixed costs of opening a hub without constraining the number of hubs to any predetermined number. A recent and comprehensive literature review on network hub location problems (Alumur and Kara, 2008) gives different types of HLPs and discusses the salient issues considered when solving these problems.

2.1 Single Objective HLPs

O'Kelly (1987) is the first to give a mathematical formulation of an HLP. He introduced a quadratic integer programming formulation of the single allocation p-hub median problem that tries to minimize the total transportation cost. In this study, he also provided the Civil Aeronautics Board (CAB) data set, which is used by nearly all of the researchers who studied HLPs (Alumur and Kara, 2008).

O'Kelly (1992) introduced the uncapacitated single allocation hub location problem (USAHLP), where number of hubs to be opened is a decision variable. He formulated the problem as a quadratic integer program and used fixed costs for hub openings. Then Campbell (1994) made a linearization of this formulation, which happened to be the first linear programming formulation for USAHLP. Campbell (1994) also provided the capacitated version of this problem with additional capacity constraints on the total flow that is directed through hubs.

An efficient mixed integer linear programming formulation of the capacitated single allocation hub location problem (CSAHLP) is presented by Ernst and Krishnamoorthy (1999). Their formulation is based on the idea formerly used in the p-hub median formulations of the same authors. This formulation handles the inter-hub transfers as multi-commodity flows and requires fewer variables and constraints than the one suggested by Campbell (1994). Moreover, the authors change the capacity constraints such that hubs are capacitated only in terms of the collection flows, that is, the flows from non-hub nodes to hub nodes. They offer two heuristics based on simulated annealing (SA) and random descent (RDH) and test these approaches on the Australian Post (AP) data set, consisting of instances up to 200 nodes.

Labbe et al. (2005) consider the single assignment hub location problems with fixed capacities on the amount of traffic that flows through hubs in telecommunication networks. The authors make a polyhedral analysis of the problem and define some valid inequalities. They also develop a branch and cut algorithm based on their findings.

Randall (2008) proposes an ant colony optimization algorithm to solve the CSAHLP. He develops four modeling variations and conducts computational experiments on the AP data set with up to 50 nodes. He obtains the optimal solutions in almost all of the runs within short computation times.

Chen (2008) suggests an effective heuristic for the CSAHLP. The heuristic performs in three levels. First, the number of hubs to be opened, p, is determined using upper bounds on p. Second, p hubs having the highest capacity are located on the network and lastly, the hub locations are improved by the so called restricted single location exchange procedure. He makes the experiments using the AP data set with up to 200 nodes and reports that the proposed heuristic outperforms the SA method of Ernst and Krishnamoorthy (1999) in 100 and 200 node problem instances.

2.2 Evolutionary Algorithms for HLPs

Having been used to solve numerous combinatorial optimization problems, evolutionary algorithms are also employed as a solution approach to HLPs. Abdinnour-Helm and Venkataramanan (1998) is one of the earliest studies that provide an evolutionary algorithm for an HLP. The authors solve the single allocation uncapacitated hub location problem with fixed costs via a genetic algorithm (GA) and find near-optimal solutions quickly and efficiently. Another GA is proposed by Abdinnour-Helm (1998) combined with a tabu search heuristic (GATS) for the same problem. She uses the GA to determine the number and location of the hubs, then she assigns non-hub nodes to its nearest hub to construct a starting solution to tabu search.

Topcuoglu et al. (2005) is another study that proposes a GA to solve the uncapacitated single allocation hub location problem. In the GA, each individual has a hub array and an assign array both of size n, that is, number of nodes in the network. Element j of the hub array takes a value of 1 or 0, indicating either j^{th} node is a hub or not. On the other hand, the j^{th} element of the assign array takes value k if this node is assigned to hub k, and $k = j$ holds if node j is a hub. They provide a procedure to generate the initial population, which suggests locating the nodes that have the highest total flow as hubs, and assignment of non-hub nodes to the nearest hubs. In our study, we also make use of a variation of their procedure to generate the initial population. The authors report that their GA outperforms GATS heuristic of Abdinnour-Helm (1998).

Cunha and Silva (2007) also propose a GA that is combined with a SA heuristic which outperforms the ones proposed in Abdinnour-Helm (1998) and Abdinnour-Helm and Venkataramanan (1998). However, they do not compare their hybrid heuristic with the GA of Topcuoglu et al. (2005).

2.3 Multiobjective HLPs

Almost all of the literature on HLPs considers a single objective, except for several studies. One study that considers multiple objectives in an HLP is due to Çamlar (2005). He considers the uncapacitated single assignment hub location problem without fixed charges for opening a hub. The objectives are to minimize the overall network cost and to minimize the maximum delivery time between origin/destination pairs. To find good solutions regarding both objectives, he uses two well known Multi-objective Evolutionary Algorithms (MOEA), namely SPEA2 (Zitzler et al., 2002) and NSGA-II (Deb et al., 2002).

Another study that respects multiple objectives in HLPs is Soylu and Köksalan (2006). In this study, two uncapacitated multiple allocation p-hub median problems are considered. In the first problem, total transportation cost minimization (p-hub median) is the first objective and total collection and distribution cost minimization (p-median) is the second objective. In the second problem, they consider minimization of maximum delay over all hubs as the second objective in addition to the first objective in the first problem. Here, delay for a hub refers to demand (collection flow to the hub)/capacity ratio for that hub. To solve the problems, they propose the Favorable Weights Evolutionary Algorithm (FWEA). The idea in this algorithm is to make each solution compete with the others through their relative strengths. Relative strength of a solution is defined by measuring the closeness of the solution (relative to others) to an ideal point in its favorite direction. This direction is defined by the favorable weights of the solution that minimize the weighted Tchebycheff distance (L_∞ norm) to the ideal point. Moreover, eliminating the need for an explicit diversity preserving operator, nearest neighbor information is also added to the relative strength of solutions. The algorithm also uses the nondominated sorting concept in NSGA-II.

Costa et al. (2008) propose two interactive solution approaches for two bicriteria single allocation hub location problems (BSAHLP). In the first problem, they convert CSAHLP into BSAHLP1 by treating the hard capacity constraints as a soft constraint via taking it to the objective. In addition to the total cost minimization objective (including fixed hubbing costs), they try to minimize the time to process total collection flow (service time) entering a hub since these flows account for the capacity usage due to sorting operation. In the second problem, with the same first objective, they try to minimize the maximum service time spent at any hub. The authors test their approaches on the AP data set.

CHAPTER 3

PROBLEM DEFINITION

Recall that we consider a bicriteria hub location problem for a postal/cargo delivery system. One of the objectives is the minimization of the total cost, which consists of total transportation cost and fixed hubbing cost and the other objective is the minimization of the total time to process the flows at hubs due to sorting operations. We assume that non-hub nodes can be assigned to a single hub and the number of hubs to be opened is not known a priori. The problem can be regarded as a combination of two subproblems: location of hubs on the network and allocation of the located hubs to non-hub nodes.

In this chapter, we first state the motivation for the study, then define the problem, and finally give the mathematical formulation of the problem.

3.1 Motivation

When making location and allocation decisions in hub location problems, an important consideration is the capacity restriction on the hubs. For postal service applications, these restrictions are due to the processing times of the flows directed from non-hub nodes to hub nodes. There are similar capacity restrictions in other systems. Sorting of cargos/packages is the major activity that constitute the flow processing times (service times) at hubs of postal delivery systems. (Alumur and Kara (2008)). Thus, sorting capacity of a hub determines the maximum number of cargos/packages that can be directed from non-hub nodes to that hub in a day.

Until recent years, the common approach has been to solve the Capacitated Single Allocation Hub Location Problem (CSAHLP) using hard capacity constraints (see Campbell 1994, Ernst and Krishnamoorthy 1999, Labbé et al.

2005). However, the hub capacities may be flexible and it may be possible, although undesirable, to use larger capacities. In real life applications, there is the possibility of excess utilization of hubs via overtime of workers and the machinery. A different approach is suggested in Costa et al. (2008) that points out the possible benefits of treating the capacity constraint as a soft constraint. They regard the capacity as the second objective that measures the time hubs take to process collected flows in addition to the usual cost minimization objective. They propose two interactive procedures to converge to the preferred region of the decision maker (DM) in the criterion space. First procedure uses the idea of minimizing the weighted sum of objective functions, whereas the second procedure employs the minimization of the Tchebycheff distance to a reference point. In both procedures, they iteratively find new nondominated solutions and ask the DM for the reservation and reference points to clarify the search direction and continue the procedure until s/he is satisfied with the results.

Our study differs from Costa et al. (2008) in the sense that we try to approximate the entire set of efficient solutions for the problem. We use an evolutionary algorithm that locates the hubs on the network. Then for each located hub set, we find an efficient way to solve the allocation problem to avoid overlaps in generating nondominated solutions. Moreover, we generate problem instances having more efficient solutions from the AP data set. We also propose a different interactive procedure to converge to the preferred region of the decision maker.

By handling the CSAHLP in multi-objective context, compared to the optimal solution obtained from capacitated single objective version of the problem, we may have the opportunity to find solutions that are far better in cost objective at the expense of a little excess utilization of the hubs regarding their capacity. Thus, with this approach, we can highlight the improvement potentials and provide more information as opposed to the common approach for solving the CSAHLP.

3.2 Definitions

In this study, we address the Bi-criteria Single Allocation Hub Location Problem (BSAHLP). Minimization of total cost is the first objective and minimization of flow process times summed over all hubs is the second objective. These are the same two objectives considered in BSAHLP-1 of Costa et al. (2008).

In combinatorial optimization problems with multiple criteria, the optimality of any solution is difficult to state since it is hardly possible to find a unique solution that gives the best criterion values in all of the conflicting criteria. The following definitions are given to clarify the nature of combinatorial optimization problems with multiple criteria. These problems can be formulated as:

$$\begin{aligned} &\text{"Min"}\{z_1(x), z_2(x), ..., z_k(x)\} \\ &\text{subject to} \\ &x \in X \end{aligned} \tag{3.1}$$

where $z_j(x)$ denotes the jth objective function value corresponding to the decision variable vector x, X is the feasible solution space, and k (≥2) is the number of objectives. If there exists a criterion vector $z_j(x')$ such that $z_j(x') \leq z_j(x)$ for j=1,2,…,k and $z_j(x') < z_j(x)$ holds for at least one j, then $z_j(x')$ *dominates* $z_j(x)$. If no x' exists satisfying these conditions, then $z_j(x)$ is a *nondominated* solution and x is an *efficient* solution. Provided that $z_j(x') < z_j(x)$ holds for all j, then $z_j(x')$ *strongly dominates* $z_j(x)$. If there does not exist any solution x' such that $z_j(x')$ strongly dominates $z_j(x)$, then the criterion vector $z(x)$ is said to be *weakly nondominated*, and the decision vector x is said to be a *weakly efficient* solution. Hence, there may be dominated solutions as well as nondominated solutions in the set of weakly nondominated solutions.

All nondominated solutions of a problem form the *Pareto-optimal (efficient) frontier* for that problem. *Ideal point* (z^{**}) is a vector of size k (i.e., $z^{**} \in \Re^k$), which is obtained by combining the optimal objective function values of all criteria. *Nadir point*, on the other hand, is a vector again of size k, which is obtained by the combination of the worst objective function values of each criterion j among the nondominated solutions. Ideal and nadir point vectors and the efficient frontier are illustrated in Figure 3.1.

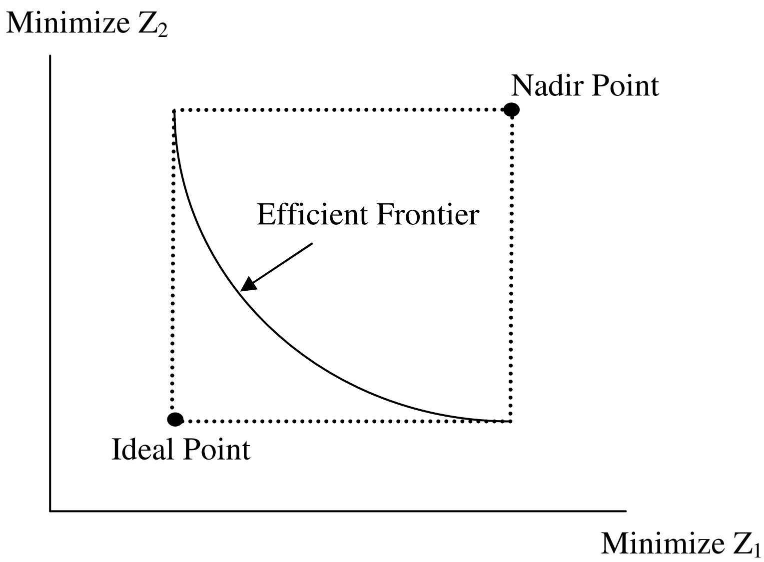

Figure 3.1 Illustration of ideal and nadir points

L_q-Metrics

Metrics are used for measuring the distance between two points x and y in the space. L_q-metric measures the distance between x and y as follows:

$$\|x-y\|_q = \left[\sum_{j=1}^{k} \left| x_j - y_j \right|^q \right]^{\frac{1}{q}} \tag{3.2}$$

A weighted L_q-metric is computed as follows:

$$\|x-y\|_q^w = \left[\sum_{j=1}^{k} (w_j \left| x_j - y_j \right|)^q \right]^{\frac{1}{q}} \tag{3.3}$$

where $w_j \geq 0$ for all j and $\sum_{j=1}^{k} w_j = 1$.

3.3 The Problem

On the network, we have a set of n demand points that act as origin/destination points, which also exhaust the possible hub location alternatives, assuring that no hub can be located at any point other than these n points. There is flow between each origin-destination pair, all of which needs to be directed through the located hub nodes. When hubs are allowed to interchange the flows of demand nodes, there is a potential of decrease in transportation costs due to economies of scale. According to Campbell (1996), the cost of transportation using hubs is less than it is with direct transportation between each pair of non-hub nodes, since it reduces the number of links and thus fixed costs of establishing a system allowing a smaller fleet of vehicles. Besides, the availability of specialized and high capacity transportation facilities (i.e., trucks, airplanes) motivates companies to use hubs in many-to-many distribution systems to enjoy economies of scale through inter-hub transfers. An example illustration of a hub network is given in Figure 3.2. The thick lines depict the concentrated flows between hub nodes (transfer flows), whereas the thin ones depict collection and distribution flows from non-hub nodes to hub nodes and from hub nodes to non-hub nodes, respectively. These advantages all contribute to the need for solving hub location problems for the practitioners of the problem.

The problem in this study is to locate the hub(s) in the network such that the total cost and the total flow process time objectives are minimized. The number of hubs to be opened is not known a priori. Thus, the model determines that number with respect to fixed costs of opening hubs. Fixed costs incurred for opening hubs added to total transportation cost defines the total cost for the problem. Total flow process time represents the service times spent at hubs to sort the flows collected from non-hub nodes. Since the problem is of single allocation type, any non-hub node is assumed to be assigned to only one hub node and multiple assignments are only possible for hub nodes.

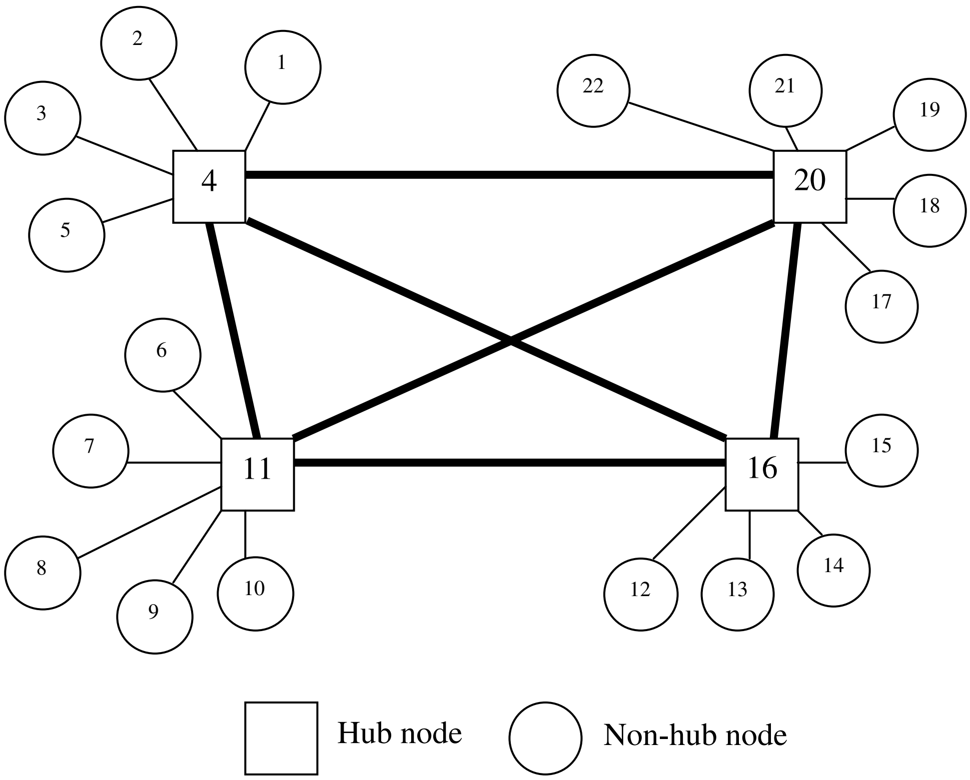

Figure 3.2 Hub network illustration

Mathematical Formulation

We give the mathematical formulation with hard capacity constraints first, and then the bi-objective model based on Costa et al.'s (2008) formulation, each using the notation of Ernst and Krishnamoorthy (1999).

The assumptions of the model are stated below.

- All nodes are fully interconnected, so that a direct link between any two demand points can be exercised.

- There is capacity on the hubs that restricts the total amount of flow that can be collected from the non-hub nodes.

- The inter-hub transfers are discounted by a factor that is between zero and one.

- A non-hub node can only be assigned to a single hub node.

Sets

$i, j, k, l \in N$,

where N is the set of nodes in the network and $N = \{1, 2, \ldots, n)$

Parameters

F_k: fixed cost of opening a hub at node k

Γ_k: capacity of hub k on the total amount of collected flow

α: coefficient of the transfer cost between any two hubs (per unit flow)

χ: coefficient of the collection cost from any non-hub node to any hub node (per unit flow)

δ: coefficient of the distribution cost from any hub node to any non-hub node (per unit flow)

d_{ij}: distance between nodes i and j

W_{ij}: total flow from location i to location j

O_i: total flow originating from node i

D_i: total flow destined to node i

($O_i = \sum_j W_{ij}$ and $D_i = \sum_j W_{ji}$)

Decision Variables

Z_{ik}: takes value 1 if i^{th} node is assigned to hub at node k, 0 o/w

Y^i_{kl}: total amount of flow routed from location i (origin) through hubs k and l

CSAHLP:

$$\text{Min} \sum_{i\in N}\sum_{k\in N} d_{ik}Z_{ik}(\chi O_i + \delta D_i) + \sum_{i\in N}\sum_{k\in N}\sum_{l\in N}\alpha d_{kl}Y^i_{kl} + \sum_{k\in N} F_k Z_{kk}$$

s.t.

$$\sum_{k\in N} Z_{ik} = 1 \qquad i = 1,\dots,n \tag{3.4}$$

$$Z_{ik} \le Z_{kk} \qquad i = 1,\dots,n \quad k = 1,\dots,n \tag{3.5}$$

$$\sum_{i\in N} O_i Z_{ik} \le \Gamma_k Z_{kk} \qquad k = 1,\dots,n \tag{3.6}$$

$$\sum_l Y^i_{kl} - \sum_l Y^i_{lk} = O_i Z_{ik} - \sum_j W_{ij} Z_{jk} \qquad i = 1,\dots,n \quad k = 1,\dots,n \tag{3.7}$$

$$Z_{ik} \in \{0,1\} \qquad i = 1,\dots,n \quad k = 1,\dots,n \tag{3.8}$$

$$Y^i_{kl} \ge 0 \qquad i = 1,\dots,n \quad k = 1,\dots,n \quad l = 1,\dots,n \tag{3.9}$$

The objective function tries to minimize the total cost, which includes transportation costs and fixed costs of hubbing. Constraints (3.4) allocate each demand node to a single hub node. Constraints (3.5) ensure that the allocation is done to a node if there is a hub at that node. Constraints (3.6) are to ensure that collected flows that are directed to any hub cannot exceed that hub's capacity. Constraints (3.7) are the divergence equations and relate flow variables to binary variables. Constraints (3.8) are to assure the values of Z_{ik} variables to be either zero or one, and finally Constraints (3.9) assure the non-negativity of Y^i_{kl} variables.

This formulation can be modified so that the capacity constraint is treated as a soft constraint rather than a hard constraint by regarding it as the second objective of the problem. The modified model as suggested in Costa et al. (2008) can be given as follows:

Additional Parameters

T_k: the time for hub k to process one unit of flow

P_k: fixed time to initiate the service at hub k

BSAHLP:

$$\text{Min} \sum_{i\in N}\sum_{k\in N} d_{ik} Z_{ik}(\chi O_i + \delta D_i) + \sum_{i\in N}\sum_{k\in N}\sum_{l\in N} \alpha d_{kl} Y^i_{kl} + \sum_{k\in N} F_k Z_{kk}$$

$$\text{Min} \sum_{i\in N}\sum_{k\in N} O_i T_k Z_{ik} + \sum_{k\in N} P_k Z_{kk}$$

s.t.

(3.4), (3.5), (3.7), (3.8), and (3.9).

The second objective measures the total service (flow process) time hubs spend to process collected flows. Note that the capacity of a hub is defined as the amount of collected flow the hub can process in CSAHLP, whereas in BSAHLP, it is defined as the time the hub takes to process all the collected flow. To make this conversion, Costa et al. (2008) define parameters T_k and P_k, which are generated as a function of the capacity values of hubs and assuming that hub capacities are expressed in units of flow for one day comprised of eight hours of work. The details of how these parameters are found will be given in the computational results part in Chapter 4.

Uncapacitated Single Allocation Hub Location Problem (USAHLP) is proven to be NP-Hard in Kara and Tansel (1998). Moreover, Stanimirović (2008) states that CSAHLP is NP-Complete.

CHAPTER 4

SOLUTION APPROACH

Location and allocation decisions are the two components that need to be considered together to solve the hub location problems. The main reason is that to locate the hubs, we need to evaluate the allocation decisions, which makes location decisions strictly based on the allocation decisions. This adds to the complexity of these problems. Research shows that even with fixed hub locations and a single objective, the allocation subproblem is difficult (Alumur and Kara, 2008). In our case, we try to solve a biobjective problem which does not have a single optimal solution. To overcome the difficulty, we suggest a two-phase solution approach to BSAHLP. In the first phase, the location of hubs, i.e., hub sets, are determined with an evolutionary algorithm heuristically. Then in the second phase, for each fixed hub set, we solve the allocation subproblem to generate the efficient solutions of that hub set. To do this, we put one of the objectives to the constraint, solve the problem exactly with Cplex, find an efficient solution and continue changing the right hand side of the constraint iteratively to find all the efficient solutions. By doing so, we eliminate the difficulty of simultaneously making the location-allocation decisions and only solve the allocation subproblem under fixed hub locations, which significantly reduces the computation time.

In this chapter, a solution approach is suggested and explained for the BSAHLP. This approach tries to generate all the solutions on the efficient frontier. One of the reasons for approximating all the efficient solutions is to present the decision maker a visual representation of the available solutions, which may be helpful to give an idea in some cases. Another reason is to test the performance of the evolutionary algorithm.

4.1 Solving the Location Subproblem

In the first phase, to determine the set of located hubs for BSAHLP, we make modifications to FWEA_loc, an evolutionary algorithm developed in Soylu and Köksalan (2006) for solving bicriteria uncapacitated multiple allocation p-hub location problems. Our aim is to approximate the efficient hub sets (hub sets with at least one allocation which is an efficient solution) via the evolutionary algorithm in the first phase, and then using these hub sets, solve the allocation subproblem in the second phase. As a result of the two phases, we intend to approximate the efficient frontier of the problem.

In general, evolutionary algorithms maintain a population of solutions instead of a single solution during the search. Each of these solutions is referred to as a member of the population. The evolutionary algorithm we suggest uses the idea in Köksalan (1999), which is to fit some Lq curves in the criteria space that intends to represent the possible locations of nondominated solutions. Borrowing this idea, we fit Lq curves to the hub sets to represent their efficient allocations in the criteria space. Since location decisions are highly dependent on the allocation decisions, if we can well represent the efficient allocations of the hub sets, then we can find desirable locations of the hubs on the network. The structure and details of our evolutionary algorithm are given below.

Representation

In the evolutionary algorithm, we use binary representation to represent the hub nodes. Each chromosome consists of an array of size n (number of nodes in the network), and each element of the array -corresponding to a node- takes value 1 if the node is a hub node and takes value 0, otherwise. An example representation is shown in Figure 4.1 for n=10. Here nodes 3 and 7 are the hub nodes, and the remaining nodes are non-hub nodes

binary var.	0	0	1	0	0	0	1	0	0	0
node i	1	2	3	4	5	6	7	8	9	10

Figure 4.1 Binary representation of located hubs

Representing the Allocations of Hub Sets via Lq Curves

To represent the efficient allocations of hub sets, we use Lq curves. In this procedure, we fit Lq curves to hub sets as in Soylu and Köksalan (2006). We need two extreme efficient (or approximately efficient) points (z_1^* = (f_1^{lower} , f_2^{upper}) and (z_2^* = (f_1^{upper} , f_2^{lower}) and a midpoint (a,b) of the hub set to fit an Lq curve. We develop heuristic procedures to find these points, which are discussed later in detail. After these three points are obtained, they are scaled to the interval [0, 1] with respect to hub set's ideal point (f_1^{lower} , f_2^{lower}) and nadir point (f_1^{upper} , f_2^{upper}) through the following equation:

$$(a',b') = \left\{ \frac{(a - f_1^{lower})}{(f_1^{upper} - f_1^{lower})}, \frac{(b - f_2^{lower})}{(f_2^{upper} - f_2^{lower})} \right\} \tag{4.1}$$

After scaling, we have three scaled points as illustrated in Figure 4.2 below, which is similar to the figure in Soylu (2007, p.60).

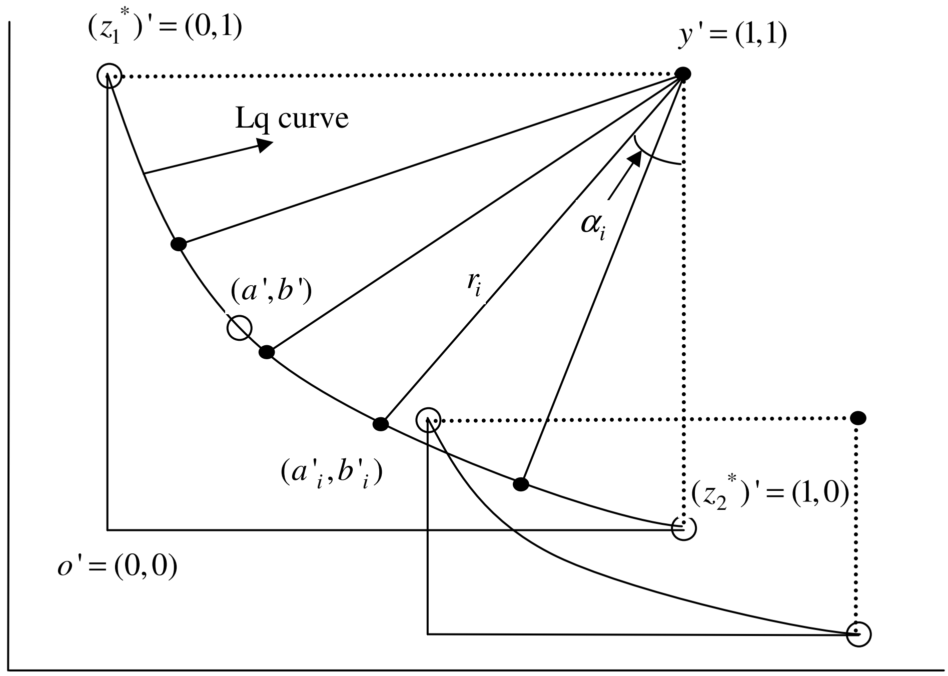

Figure 4.2 Demonstration of the Lq Curve

The following equation can be used to define an Lq distance function:

$$(1-a')^q + (1-b')^q = 1 \tag{4.2}$$

We try to fit a curve that passes through a known point (a',b') on the curve. In doing so, the equation is solved by trial and error to find the value of q. The fitted Lq curve is used to symbolize the efficient frontier of the hub set. On the fitted curve, we take equally spaced hypothetical points to represent efficient allocations of the hub set as in Köksalan (1999). To take k+1 equally spaced points, the 90° angle $\widehat{(z_1^*)'y'(z_2^*)'}$ is divided into k equal angles α_i, where $\alpha_i = 90i/k$ for i=0,1,..,k. A point (a'_i, b'_i) at α_i degree from $\overline{y'(z_2^*)'}$ can be generated by

$$a'_i = 1 - r_i \sin\alpha_i \quad \text{and} \quad b'_i = 1 - r_i \cos\alpha_i \tag{4.3}$$

where r_i is the Euclidean distance from y' to (a'_i, b'_i). The normalized coordinates of these hypothetical points (a'_i, b'_i) should be backtransformed to the original range of criteria space so as to find (a_i, b_i). This can be done easily if we solve equation (4.1) for a_i and b_i.

In Figure 4.3, we give an illustration of the fitted Lq curves and the representative points (circular points) taken for three efficient hub sets of an instance of the AP data set. This figure shows that the Lq distance functions represent the solutions well.

After approximating the efficient frontier of a member's hub set with hypothetical points on the Lq curve, we select one of these points deterministically and mark as member_ideal (MI), which is then used to represent the overall performance of the member. The MI is selected among the least dominated points of the member, where the degree of dominance is indicated by the front identification numbers (front IDs) assigned to points. The least dominated point of a member has the smallest front ID. The determination of front IDs is described under "Fitness of Members". If there are more than one point of the member having the best front ID, then the point that has the minimum Tchebycheff distance to the ideal point is selected as the MI to break the tie. The fitness calculations and fitness value related functions operate on the representative MIs.

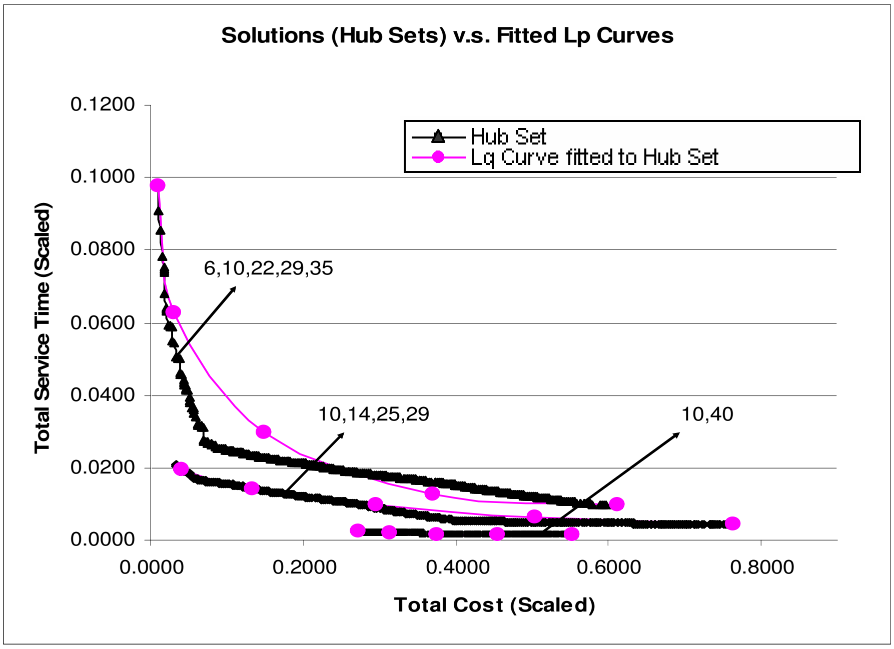

Figure 4.3 Illustration of solutions and fitted Lq curves for the AP data set.

Fitness of Members

In the proposed EA, the fitness assignment strategy in Soylu and Köksalan (2006) is used. According to this strategy, a raw-fitness value is assigned to members according to their relative strength over the other members in the population. Authors compute the relative strength $\Delta(x^i)$ of a member $\hat{x}$ over any other member $x^i \in X$ using weighted Tchebycheff distances, $\phi(w,x)$, of these members' MIs to an ideal point by

$$\Delta(x^i) = \phi(\hat{w}, x^i) - \phi(\hat{w}, \hat{x}) \tag{4.4}$$

where

$$\phi(w,x) = \max_{j=1,2,\ldots,k} \left\{ w_j (z_j(x) - z_j^*) \right\} \tag{4.5}$$

, X is the set of feasible solutions and z_j^* is an ideal point in objective j.

Here the weight ($\hat{w}$) is the favorable weight of member $\hat{x}$, meaning that this weight minimizes the weighted Tchebycheff distance of $\hat{x}$ to the ideal point. Thereby, finding favorable weights of a member and using these weights to compare this member with another member help us find out if the latter member is better (closer to ideal point) with the direction that favors the former member. If it is not, this means the former member performs better in its favorite direction. With this approach we can have members contend with each other on the field they are good at, and assign raw-fitness values accordingly. We can find the favorable weights of a solution to minimize a weighted Tchebycheff distance function with the following closed form solution (Steuer, 1986 p.425):

$$w_j = \begin{cases} \dfrac{1}{(z_j - z_j^{**})}\left[\sum_{i=1}^{k} \dfrac{1}{(z_i - z_i^{**})}\right]^{-1} & if\ z_i \neq z_i^{**}\ for\ all\ i \\ 1 & if\ z_j = z_j^{**} \\ 0 & if\ z_j \neq z_j^{**}\ but\ \exists i \ni z_i = z_i^{**} \end{cases} \tag{4.6}$$

Here z_j is the j^{th} objective function value of the (j^{th} element of criterion vector of solution z) and z_j^{**} is an ideal point in criterion j.

In other words, the relative strength of $\hat{x}$ over x^i shows how much member $\hat{x}$ is closer to ideal point than member x^i with its own favorable weights. Note that $\Delta(x^i)$ can take a negative value implying that even with favorable weights of $\hat{x}$, x^i is the favored member considering the weighted Tchebycheff distances. The raw-fitness values of members are assigned with the following equation:

$$\text{rawfitness}(\hat{x}) = \alpha\overline{\Delta} + (1-\alpha)\Delta_{\min} \tag{4.7}$$

Here $\overline{\Delta}$ denotes the average relative strength and is computed by $\overline{\Delta} = \frac{\sum_{x^i \in X} \Delta(x^i)}{|X|}$ and $\Delta_{\min}$ is a worst case measure which can be found by $\Delta_{\min} = \min_{x^i \in X} \left\{ \Delta(x^i) \right\}$. The parameter α defines the weights (importance) of average relative strength and minimum relative strength in determining the raw-fitness of the member. By means of the minimum strength measure, a point far from its neighbors will be favored simply by being promoted with a larger $\Delta_{\min}$ value, attempting to preserve diversity during the search of the EA.

We also use the nondominated sorting based fitness assignment to adjust the raw fitness values of members as in FWEA_loc (Soylu and Köksalan, 2006). Nondominated sorting principle is first proposed by Goldberg (1989) and it has been used in some well known multi-objective evolutionary algorithms such as NSGA I-II (Srinivas and Deb, 1994, Deb et al., 2002). The idea in nondominated sorting is to put the nondominated members of the population on the first front (assign their front ID as 1), then temporarily take out these members and put the nondominated members in the remaining population on the second front and continue this classification until all members are assigned a front ID.

After all members are classified onto a front, the raw-fitness values of members are adjusted in order not to allow a member in a worse front have a better fitness value than a member in a better front. To prevent this, the raw-fitness values of the members in the first front are left as they are and the following adjustment is done for members in the remaining fronts:

$$\text{fitness}(\hat{x}) = \min_{x \in X^{front\_j-1}} \left\{ \text{fitness}(x) \right\} - \max_{x \in X^{front\_j}} \left\{ \text{rawfitness}(x) \right\} + \text{rawfitness}(\hat{x}) - \varepsilon$$

where $X^{front-j}$ denotes the set of solutions on front j, $j = 2,3,...,N$ and ε is a small positive constant. This adjustment assures that the minimum fitness value on a better front is at least an amount ε better than the maximum fitness value in a worse front.

Scaling of Objectives

As stated before, we use distance functions in the EA to assign fitness values to individuals. If we use the real criterion values for calculating the distances to the ideal point instead of scaled values, then we cannot escape favoring the objective having larger range. This is illustrated in Figure 4.4.

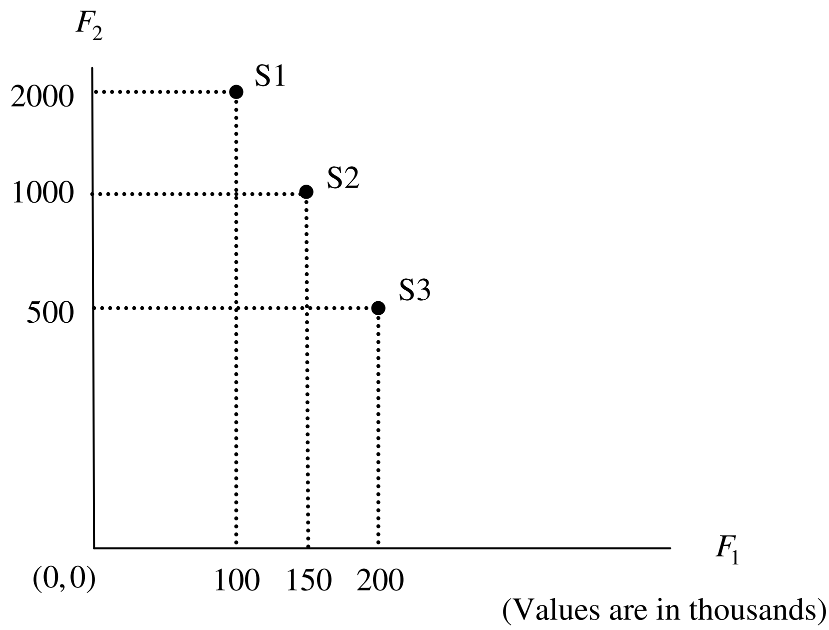

Figure 4.4 A hypothetical example to illustrate scaling of objectives

Suppose F_1 is in the range (100000, 200000) and F_2 is in range (1000000, 2000000) and suppose we want to select the solution having the minimum weighted Tchebycheff distance to the ideal point (0, 0) using weights (0.5, 0.5). With using equal weights our intention is to make an unbiased selection (do not favor either of the objectives for minimization) with respect to the two criteria. Using equation (4.5), the resulting distance values of S1, S2, and S3 are 1000000, 750000, and 500000, respectively. So, our selection would be S3, which has the minimum value in the objective that has the larger range, namely F_2. However, we would expect to select S2 under equal importance of criteria, since it tries to minimize both objectives. To remove such biases, we can scale down the objectives in range zero-one. Let $z^j = (z_1{}^j, z_2{}^j)$ be a solution in the objective space. The scaled values can be found as:

$$(z_i{}^j)' = \frac{z_i{}^j - z_i{}^{lower}}{(z_i{}^{upper} - z_i{}^{lower})}, \quad i = 1,2 \tag{4.8}$$

where $(z_i^{lower}, z_i^{upper})$ defines the range for objective i. To find z_i^{lower}, we can solve LP relaxation for objective *i*, and for z_i^{upper}, we can take the worst value in the initial population (Soylu and Köksalan, 2006). After scaling the objectives, the weighted Tchebycheff distances for S1, S2, and S3 with the same weight set become 0.5, 0.25, and 0.5, respectively, thereby leading us to select the expected solution, which is S2.

Initial Population

We generate a predetermined number of chromosomes for the initial population. Recall that the members of our EA represent the location of hubs. Since number of hubs to be opened is unknown beforehand in BSAHLP, we first need to determine the number of hubs and then locate them. To generate the initial population, we apply the strategy suggested in Topçuoğlu et al. (2005) with the required modifications. For 75% of the initial population, the number of hubs is selected at random from the interval $[1,2,...,n/4]$, n being the node size of the problem. The number of nodes for the remaining 25% of the initial population is selected from the interval $[n/4,...,n/2]$, leading to generation of chromosomes having more hubs. This strategy states that number of hubs of a member can be at most half of the node size *n*.

What comes next is to divide the initial population into three portions and make hub location decisions differently in each of the portions. In the first portion, corresponding to three eights of the initial population, hubs are located so as to favor the total cost criterion. To do this, we sort the nodes in decreasing order of total flow of each node. Total flow for a node *i* is calculated by summing total flow that originates from node *i* and total flow that is destined to node *i*. Then for this portion, the nodes from the top two thirds of the sorted list are selected as hubs, starting with the node that has the highest total flow value. This selection rules gives priority to nodes that have higher total flow to become a hub, and by doing so, we have more chance to benefit economies of scale via the inter-hub transfer in favor of total cost objective.

For the second portion, for another three eights of the initial population, hubs are located to favor the total flow process time objective. For selection of hubs, again a list of nodes is formed, this time in increasing order of T_k values of the nodes. Remember that T_k denotes the time node k (if selected as hub) takes to process one unit of flow. In the same manner as in first portion, nodes that reside in the top two thirds of the sorted list (starting with the lowest T_k value) are selected as hubs in favor of the total flow process time objective.

In the last portion, consisting of the remaining two eights of the initial population, we assign the hubs on the chromosome totally randomly, that is, favoring neither of the objectives. When selecting the hub nodes, we do not allow any two chromosomes to appear in the initial population. The reason for this precaution is to initialize the EA with as many different hub sets as possible and not let any node be overrepresented when there are nodes not selected as hub even once in the initial population. Note that if any node does not exist in the initial population, we may never see that node as a hub during the search, since the search is mainly directed via crossover of chromosomes.

Crossover

We apply single point crossover in the EA with probability 1. According to this procedure, a single random point is selected on the strings of the chromosomes and the parts on the same side (right side in our case) are interchanged between the two chromosomes as shown in Figure 4.5.

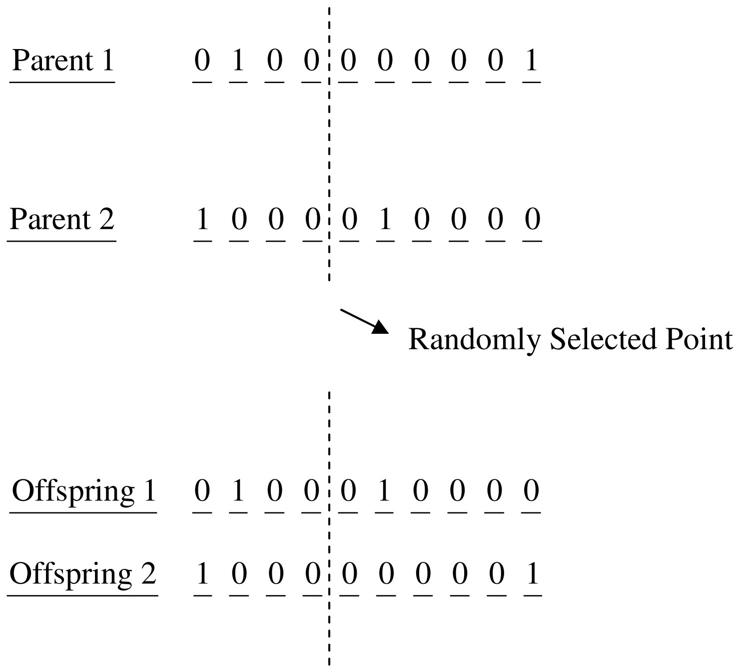

Figure 4.5 An example crossover for n=10

Note that there is the possibility of obtaining an offspring without a hub after the crossover operation, which is infeasible. In such a case, we discard the offspring that has no hubs to preserve feasibility of solutions.

Mutation

After the population starts to converge over as number of generations gets larger, only crossover itself becomes incapable of diversifying the search. At this point, mutating the offspring becomes more dominant at diversity preservation. In this study, for the mutation of offspring, the value of a randomly chosen gene on the offspring chromosome is changed to 1 if the value is 0 at then or it is changed to 0, otherwise.

Selection

In selection of parents for performing crossover, we apply binary tournament selection. In this method, two members of the population are selected at random and a tournament is played between them. The member with higher fitness becomes the first parent for crossover. Then again a couple of members are selected at random and the winner is promoted as the second parent unless it is the same member with the first parent (has the same hub set). If two parents are the same, a different couple is selected to play a tournament until the parents are distinct. The winners are then recombined for offspring generation.

Insertion and Replacement

We use the insertion strategy suggested in Köksalan and Phelps (2006). According to their suggestion, insertion of an offspring into the population is performed until a preset upper limit on population cardinality is reached and if the offspring is neither "duplicate" (in the decision space) nor "stillborn". The "stillborn" status in our study applies to an offspring if it is dominated by any of the members in the worst frontier. The "duplicate" status, on the other hand, applies to offspring which has the same hub set with a member in the population. If the predetermined upper limit is reached and if the offspring is neither a "duplicate" nor a "stillborn", it replaces a member on the worst frontier to keep the population size constant.

We use steady-state replacement strategy; thereby we produce two offspring at each generation and replace each of them with a randomly selected member amongst the ones on the worst frontier.

Finding Upper and Lower Bounds (for Members) in the Criteria Space

To be able to fit the Lq curves to members having more than one hub, we need three allocations (points) on the efficient frontier of these members as represented with empty circles in Figure 4.2. These points correspond to the extreme efficient (or almost efficient) solution in the first objective, the extreme efficient (or almost efficient) solution in the second objective, and a midpoint in between these two extremes. Below heuristic procedures are developed to find these points. Note that if the member has only one hub, we cannot fit an Lq curve. Thus, single hub members are represented by a single point in the criteria space.

To find the minimum total flow process time (referred to as minservt hereby) allocation of a hub set, we use the following proposition to make the allocation decision.

Proposition. Given a set S of fixed hub locations of a member, assigning all the non-hub nodes to the hub having the smallest T_k value is an optimal solution to minimize total flow process time objective for that member.

Proof. Let $T_{[1]} \leq \ldots \leq T_{[i]} \leq T_{[i+1]} \leq \ldots \leq T_{[|S|]}$ denote the list of times hubs take to process one unit of flow sorted in the increasing order and PT* denote the optimal value of the total flow process time objective. Assume that we assign all non-hub nodes to the hub in S having the smallest T_k value, namely $T_{[1]}$. If we can show that assigning non-hub nodes to another hub i in S with larger unit flow processing time, such as replacing $T_{[1]}$ with $T_{[i]}$, cannot reduce PT* under constant total flow, then the optimality of PT* will be proven. Let PT^{new} be the new value of the total flow process time objective after replacement. $PT^{new} \leq PT^*$ is true if and only if $(\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij}) * T_{[i]} \leq (\sum_{i=1}^{N}\sum_{j=1}^{N} W_{ij}) * T_{[1]}$ is true. Since $W_{ij} \geq 0$ for all i and for all j, and $T_{[i]} \geq T_{[1]}$, $PT^* \leq PT^{new}$.

This proves our proposition.

□

Thus, the minservt allocation of the member can be found simply by assigning all non-hub nodes to the hub node with the smallest T_k value (minservt_hub) in the hub set of the member.

The solution found above is inputted to the heuristic procedure as a starting solution to find the minimum total cost (referred to as mincost hereby) allocation. Since the number of hubs is not restricted in our problem, single hub members can also exist, which do not need any of the treatments suggested in the heuristic. Thus, there are two cases to be considered when finding the mincost allocation:

Case 1. Number of hubs of the member is equal to one.

When this is the case, we only have one allocation possibility, thus, no need for calculation of mincost extreme since mincost and minservt allocations are the same for single hub solutions.

Case 2. Number of hubs of the member is more than one.

Under this condition we have as many as p^{N-p} allocation alternatives, where p is the number of fixed hub locations. To find the mincost allocation, for each non-hub node j, we first find the hub that is at minimum distance to node j, min_dist^j, and then calculate a savings measure (Savings^j) for node j. This measure is an indicator to the expected improvement in total cost objective if node j were assigned to min_dist^j instead of minservt_hub assuming the two hubs are distinct. Savings for node j can be calculated as follows:

$$\text{Savings}^j = \frac{d_{j\text{minservt\_hub}} - d_{j\text{min\_dist}^j}}{T_{\text{min\_dist}^j} - T_{\text{minservt\_hub}}} \tag{4.9}$$

The numerator gives the difference of distances from node j to minservt_hub and from node j to its closest hub, showing how much closer node j will get to its hub if it its assignment is changed from minservt_hub to min_dist^j. The larger this difference the more the possibility of reducing total transportation cost since non-hub to hub transfers have a bigger cost factor (see 4.1.13 for cost parameters). The denominator, on the other hand, gives the difference of unit flow processing times between min_dist^j and minservt_hub. This measure intends to indicate the potential of improvement in total cost objective with a unit sacrifice from the total flow process time objective if we change the assignment of a node j from the minservt_hub to min_dist^j. Therefore, positive savings lead to changing of the assignments.

After finding Savings^j for each non-hub node j, savings are sorted in decreasing order. Starting with the biggest savings giving node and for all j such that $\text{Savings}^j > 0$, the assignments are changed from minservt_hub to min_dist^j step by step until a non-improving step is encountered. Here, a non-improving step refers to an assignment change that leads to an increase in total cost. At the end of improving steps, the solution we find becomes the mincost allocation for the member. Note that there is the possibility of obtaining no improvement in the first step. In that case, we regard the member as a single hub member and no Lq curve is fitted to such members. Another possibility is the existence of only one improving step. If this is the case, with two allocations at hand, we do not find a midpoint but join the two extremes with a straight line (take p = 1.0 in Lq equation).

In processing the above heuristic, starting with the minservt allocation, we find a sequence of solutions until the mincost allocation is reached. This sequence of solutions is also used for finding a midpoint between the two extremes. The solution with least difference in scaled objective values (the one that minimizes weighted Tchebycheff distance to the ideal point (0, 0) with equal weights) is marked as the midpoint to be used in Lq curve fitting.

The Steps of the Evolutionary Algorithm

The parameters used in the algorithm are as follows:

- PopSize1: cardinality of the initial population.
- PopSize2: upper limit on the cardinality of the population.
- p_c: crossover probability
- p_m: mutation probability
- Maxgens: maximum number of generations

Below is the sketch of the algorithm:

0. Generate an initial population of size PopSize1.
1. For each member of the population,
   - 1.1 Solve the allocation problem for each criterion and find extreme efficient solutions (allocations). Also find another solution (a, b) of this member.
   - 1.2 Scale the objectives.
   - 1.3 Find an Lq distance function passing from the three points of the member found in Step 1.1.
   - 1.4 Take k equally spaced representative points on the fitted Lq curve.
   - 1.5 Determine the frontier of each point on the Lq curve.
   - 1.6 Determine the member_ideal for each member.

   For each MI in the initial population,
   - 1.7 Compute the favorable weights and the raw-fitness scores.
   - 1.8 Adjust the raw-fitness scores according to frontiers.
2. Select two parents according to the selection operator.
3. Apply crossover with probability p_c to generate two offspring at each generation and apply mutation to the offspring with probability p_m.
4. Check if the offspring is in "duplicate" status, discard the duplicate one. If both offspring are duplicates, then discard both and go to Step 9.
5. Repeat Steps 1.1-1.4.

6. Check if the offspring is dominated by any of the member_ideals on the worst frontier ("stillborn"), discard the stillborn offspring. If both offspring are stillborn, then discard both and go to Step 9.
7. Insert the offspring one by one into the population until the population cardinality reaches PopSize2. Then, to keep the population size constant, replace the offspring with one of the members on the worst frontier.
8. Repeat Steps 1.5-1.8.
9. If Maxgens is not reached, go to Step 2, terminate the simulation, otherwise.

4.2 Solving the Allocation Subproblem

At the end of the simulation run, the set of members having front ID 1 in the final population corresponds to the set of efficient or approximately efficient hub sets for the problem. Having solved the location problem, the next step is to find the allocations of each of these hub sets to approximate the entire Pareto-optimal frontier of the problem. Note that solving the allocation problem for a hub set, we can find the efficient allocations (frontier) for that specific hub set. However, efficient allocations of a hub set may be inefficient in the presence of other hub sets' efficient solutions. This is illustrated in Figure 4.6. Each of the three hub sets has its own efficient frontier consisting of their (locally) efficient allocations, and some of these locally efficient allocations are globally inefficient, which are represented with dashed lines.

Knowing this fact, we can argue that solving the allocation problem for each hub set and finding all the locally efficient allocations would be ineffective in two senses. First, it needs too much computational effort to find all allocations for a given hub set. Second, some of these allocations will be inefficient in the global sense. To overcome these shortcomings, we propose a bounding procedure and try to find only the likely to be globally efficient allocations for each hub set.

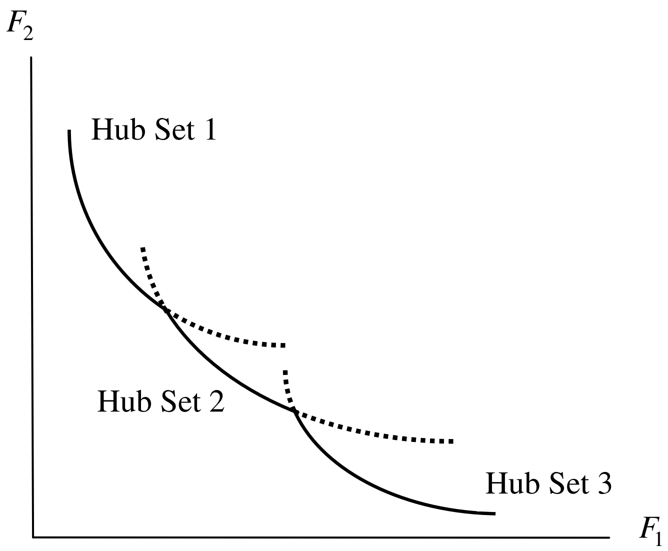

Figure 4.6 Efficiency of hub sets' allocations

In the suggested procedure, we make use of the generated representative (hypothetical) points on the formerly fitted Lq curves for the hub sets. For an efficient hub set, using all efficient hub sets' representative points, we find the dominated portions of the (locally) efficient frontier for that hub set. Then, we eliminate these portions, determine bounds, and solve the allocation problem for each hub set within the determined bounds. These bounds restrict the criteria space in an attempt to help us find only the allocations that are globally efficient.

The details of the procedure are given using the following notation:

- $H(p)$:efficient hub set of the representative point p.
- S : set of representative points sorted in decreasing order of F_2 values.
- $p_{[s]}$: representative point at sequence s in list S.
- $(p)^{F_i}$: i^{th} objective function value of representative point *p*.
- *a*, *b*: the sequence indices of the two points compared in list *S*.

- $H(p)^{F_{i\_lower(upper)}}$: the minimum (maximum) value of i^{th} objective on the efficient frontier of hub set *H*.
- SIZE: number of nondominated representative points of all efficient hub sets.

Before implementing the bounding procedure, we do some preprocessing to eliminate the representative points that are nondominated among the set of all representative points but that are dominated by the Lq curves of the efficient hub sets. The outline of elimination procedure is given below.

1. Set $a = 0$ and $b = 1$. Form a list *S* of representative points $p_{[s]}$ that are nondominated among all $p_{[s]}$ and sorted in decreasing order of F_2 values. Set $SIZE = |S|$.
2. Set $a = b$ and $b = b + 1$.
3. If $b >$ SIZE, go to Step 6; otherwise, test if the hub sets of points $p_{[a]}$ and $p_{[b]}$ are the same. If $H(p_{[a]}) = H(p_{[b]})$, return to Step 2; otherwise, go to Step 4.
4. Take horizontal projection of $p_{[a]}$ onto the Lq curve of $H(p_{[b]})$. If there is an intersection point and F_1 value of the projection (intersection point) is less than $p_{[a]}^{F_1}$, then eliminate $p_{[a]}$ and return to Step 2; otherwise, go to Step 5.
5. Take vertical projection of $p_{[b]}$ onto the Lq curve of $H(p_{[a]})$. If there is an intersection point and F_2 value of the projection (intersection point) is less than $p_{[b]}^{F_2}$, eliminate $p_{[b]}$, set $b = b + 1$, and return to Step 3; otherwise, return to Step 2.
6. End of elimination procedure. Update list *S* by taking out the eliminated points and form the list S'.

After obtaining the updated list using the elimination procedure, we implement the steps below to determine the bounds on efficient hub sets.

1. Set $s = 0$. Use the updated list S' of points $p_{[s]}$ obtained from the elimination procedure. Set $SIZE = |S'|$.
2. Set $s = s + 1$.
3. Test if the hub sets of the successive points are the same. If $H(p_{[s]}) = H(p_{[s+1]})$, return to Step 2; otherwise, continue.
4. Test if $H(p_{[s]})^{F_1\_lower} < (p_{[s+1]})^{F_1} < H(p_{[s]})^{F_1\_upper}$. If true, go to Step 5; otherwise, go to Step 6.
5. Find $H(p_{[s]})^{LB}$ by projecting $p_{[s+1]}$ onto the Lq curve of $H(p_{[s]})$ vertically. The point of intersection gives $H(p_{[s]})^{LB}$ in F_2.
6. Set $H(p_{[s+1]})^{UB} = (p_{[s]})^{F_2}$. If $s+1 <$ SIZE, return to Step 2; otherwise, end the procedure.

A hypothetical example of the bounding procedure is illustrated in Figure 4.7 below.

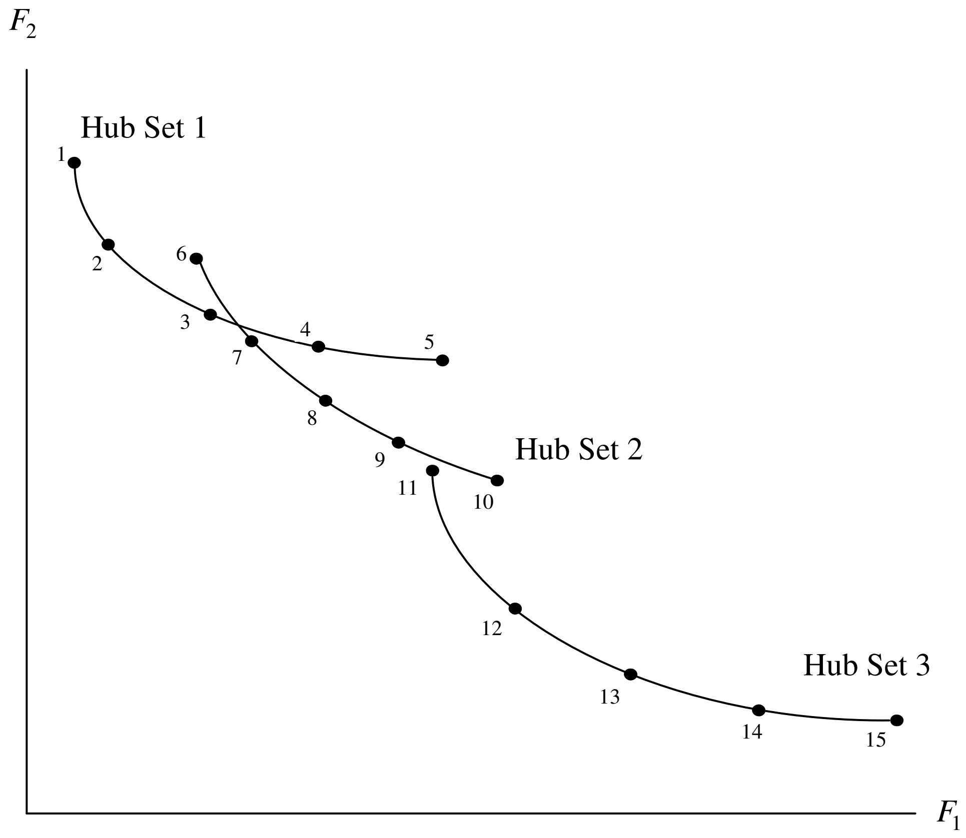

Figure 4.7 A hypothetical example for the bounding procedure

Point 5 of Hub Set 1 is already dominated by points 8 and 9, thus it is discarded even before the preprocessing. The preprocessing part would eliminate points 6, 4, and 10 since these points are not dominated by other representative points but are dominated by the fitted Lq curves. Eliminating these points is logical because they do not dominate any portion on the Lq curves, hence, they have no help in setting bounds on hub sets' allocations. Next, we determine the bounds by performing the steps of the bounding procedure. After iterating the steps, we determine the following bounds on the hub sets (Figure 4.8):

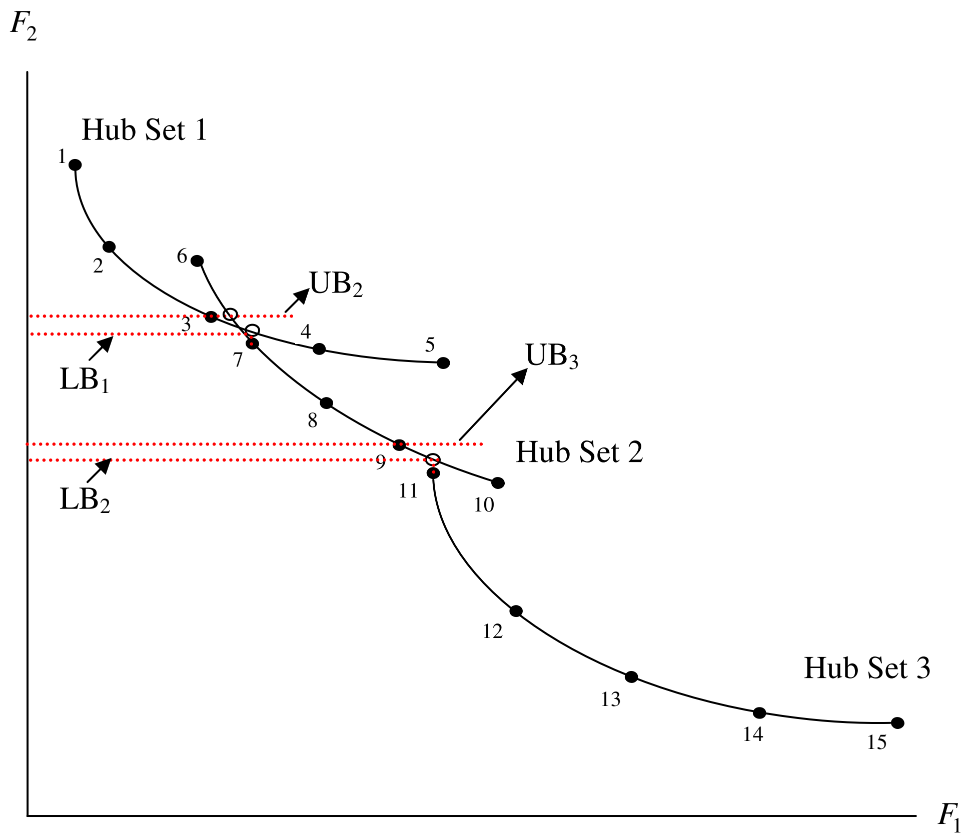

Figure 4.8 Demonstration of the bounds determined

- LB on Hub Set 1 (LB_1) is the F_2 value of the intersection point found by vertical projection of point 7 onto the Lq curve of this hub set.
- UB on Hub Set 2 (UB_2) is the F_2 value of point 3.
- LB on Hub Set 2 (LB_2) is the F_2 value of the intersection point found by vertical projection of point 11 onto the Lq curve of this hub set.
- UB on Hub Set 3 (UB_3) is the F_2 value of point 9, which in this case does not bound allocations of this hub set.

To test whether this procedure brings savings or not, we selected the problem instance 40TT_0.1FC (what the abbreviation stands for is described in section 4.4) and did some computations. First, we ran the EA to find the efficient or approximately efficient hub sets and found 71 such hub sets. Then, we tried to solve (with Cplex) the allocation problem for each of these hub sets without bounds. As expected, finding all allocations of hub sets consumed too much time. In fact, we could only find the allocations of 47 hub sets in 77.4 hours, and the model stopped running due to memory requirements. The number of allocations found corresponding to these hub sets was 25,562. However, the problem has 1135 efficient allocations in total, which points out that much of the allocations found are inefficient in the global sense. Finally, we used the bounding procedure and solved the allocation problem for each hub set within the determined bounds. As a result, we obtained 1549 allocations in about 2 hours, and 903 of these allocations appeared to be globally efficient. Thus, based on the results, we concluded that the bounding procedure brings much savings.

4.3 Performance Evaluation

To evaluate the performance of our solution procedure, we generated the efficient frontier (ie., all efficient hub sets and their efficient allocations) using the $\varepsilon-$constraint method suggested in Haimes et al. (1971). In this method, one of the objectives is regarded as a constraint and by parametrically restricting the value of this objective, the resulting single objective problem is solved repeatedly until all efficient solutions are obtained. The constrained objective is also augmented to the optimized objective with a sufficiently small positive constant (γ) in order to eliminate weakly nondominated but dominated solutions. The resulting augmented single objective model is as follows:

$\varepsilon-$constraint model:

Min $f_1(x)+\gamma.f_2(x)$

s.t.

(3.4), (3.5), (3.7), (3.8), and (3.9)

$$f_2(x) \leq \varepsilon \tag{4.10}$$

$$f_1(x) = \sum_{i \in N} \sum_{k \in N} d_{ik} Z_{ik} (\chi O_i + \delta D_i) + \sum_{i \in N} \sum_{k \in N} \sum_{l \in N} \alpha d_{kl} Y_{kl}^i + \sum_{k \in N} F_k Z_{kk} \tag{4.11}$$

$$f_2(x) = \sum_{i \in N} \sum_{k \in N} O_i T_k Z_{ik} + \sum_{k \in N} P_k Z_{kk} \tag{4.12}$$

ε is the parameter that is used to set an upper bound on the second objective. We find ε by subtracting a sufficiently small positive constant (*c*) - we took *c* as 0.1 in the experiments - from the second objective value of previously found efficient solution. Thus, ε is not constant and takes a smaller value at each iteration. Suppose that we first solve the problem without bounding $f_2(x)$. Let $z = (z_1^{lower}, z_2^{upper})$ be the resulting optimal (extreme efficient) solution that minimizes $f_1(x)$. Then this gives the worst $f_2(x)$ value, z_2^{upper}, due to the confliction between the two objectives. Next, we solve the problem this time by setting $\varepsilon = z_2^{upper} - c$. ε in this case forces the model to find the adjacent efficient solution having smaller $f_2(x)$ value through sacrificing from (worsening) the total cost. Using this method, we guarantee generation of all efficient allocations.

For the evaluation of our two-phase approach, we used a performance indicator based on the Hypervolume metric (Zitzler and Thiele, 1998). This metric measures the volume of the objective space that is dominated by a set of solutions with respect to a reference point. An illustration of the Hypervolume (HV) measure is given in Figure 4.9. The shaded region shows the total volume dominated by the four inferior solutions on the figure. This volume would increase if we had more converging and diverse set of efficient solutions as represented by the points that are more distant to the reference point, W. We can select the reference point as the nadir point. Since we scale our objectives to range zero-one, we use W= (1, 1). To measure how close is our set of solutions *S* to the set of Pareto-optimal solutions *P*, we use HVR (Veldhuizen, 1999), which is the ratio of HVs for the two sets as given below:

$$HVR = \frac{HV\ of\ S}{HV\ of\ P} \tag{4.13}$$

Then, we want bigger values of HVR, which can be at most 1 in the ideal case.

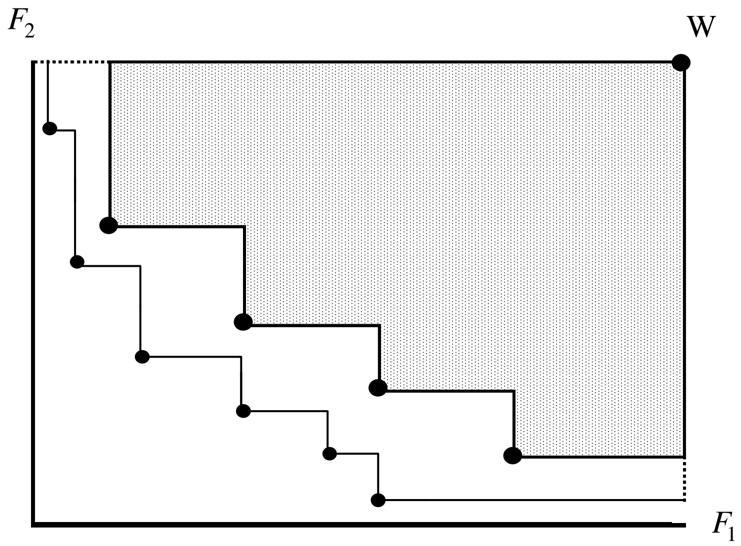

Figure 4.9 Hypervolume demonstration

4.4 Computational Results

We test our two-phase solution approach with the Australian Post (AP) data set, which is a commonly used data set by the hub location researchers (Alumur and Kara, 2008). This data set was first used in Ernst and Krishnamoorthy (1996). Different than the CAB data set, another commonly used data set on this field, the AP data set includes fixed hubbing costs and capacities for hubs, which is the reason why we prefer the AP data set. Another data set that has been recently brought into the literature is the Turkish PTT data set (TPDS), however, it does not involve the capacity information for hubs. All three data sets can be accessed via OR_Library, (http://people.brunel.ac.uk/~mastjjb/jeb/orlib/phubinfo.html).

The AP data set consists of 200 nodes representing the postal districts in Sydney, Australia. Smaller-sized instances can be generated via a C code available in the OR_Library. In the data set, the mail flows are not symmetric ($W_{ij} \neq W_{ji}$) and there is also flow within a city ($W_{ii} \neq 0$). The data set involves the coordinates of the districts and we used them to compute the Euclidean distances between each pair of districts. The costs per unit flow for collection, transfer, and distribution are 3, 0.75, and 2, respectively. Two types of fixed hubbing costs and capacities are present in the data set, namely tight (T) and loose (L). To compute parameters T_k and P_k to be able to build the second objective function, Costa et al. (2008) use the capacity values in Australian Post (AP) data set. For example, if a hub has a capacity value of 2500, they assume that 2500 mails/cargos can be sorted in a day comprised of 8 hours (28800 seconds). Then they generate T_k values, the time hub *k* takes to process unit flow, simply by dividing 28800 by 2500. Thereby, T_k values denote how many seconds hub *k* needs to process one unit of flow. Note that smaller the T_k larger the capacity of hub *k*. Using the computed T_k values, they generate the P_k values, fixed time to initiate service at hub *k*, by fitting an increasing (concave) function on the capacity of the nodes.

We attempted to apply our solution approach on a 40 node problem instance with tight fixed costs (FC) and capacity parameters (BSAHLP_40TT). We first used the $\varepsilon-$constraint method to generate the efficient frontier. However, we noticed that at most three hubs were opened in the efficient solutions to this problem. This observation led us to the conclusion that opening more than three hubs, which would reduce the transportation cost via increased volume of inter-hub transfers, is not preferred due to high FC of opening a hub. Hence, to force the model open more hubs and to make the problem more conflicting, we deliberately decreased the fixed hubbing costs and solved the problem with new cost parameters. This manipulation added to the difficulty of the problem significantly. Nevertheless, with increased number of efficient solutions and increased number of hubs opened, the problem became more challenging, thus more suitable to test the performance of the EA. We sequentially applied 60%

(0.4FC), 70% (0.3FC), 80% (0.2FC), and 90% (0.1FC) reduction to fixed hubbing costs of the 40 node problem and did the computations for these four problems. Note that more reduction brought more difficulty to the problem. Moreover, five replications are done for each problem instance using different seeds for the random number generator and the performance analysis is done for one randomly selected replication and for the merged results of the five replications.

In the evolutionary algorithm, we set crossover probability to 1.0 and mutation probability to 0.05. In the raw-fitness function, we use the same β level set in FWEA_loc. The number of hypothetical points on the fitted Lq curves to represent the allocations of a hub set is selected as 5. Initial population size and the upper limit (PopSize1, PopSize2) are selected differently for each instance of the problem. We set maximum number of generations to 20,000 as the termination condition. All the computations are done on a Pentium IV 2.8 GHz PC with 2GB RAM.

BSAHLP_40TT_0.4FC

The fixed costs of hubbing are reduced by the factor 0.6 for this problem. When we generate the entire efficient frontier for the problem using the $\varepsilon-$constraint method, we observe 15 efficient hub sets. The efficient solutions are plotted in Figure 4.10. Since the problem is the easiest one among the four, we took the initial population size and the cardinality of the final population as 20 and 25, respectively.

Table 4.1 gives the hub sets that are found by each replication of the evolutionary algorithm. Note that all replications were capable of finding all efficient hub sets of the problem, promising a good approximation of the entire efficient frontier via the use of any single replication and solving the corresponding allocation problems. We also observe that each replication has the same three inefficient hub sets (marked with *) that are put in the first front by the EA. In the presence of all efficient hub sets, one would expect these hub sets to have higher front ID. However, since we find the mincost extreme heuristically, the mincost extreme solution of an efficient hub set can fail to dominate the mincost extreme of the ineffcient solution due to the suboptimality of the heuristic solution.

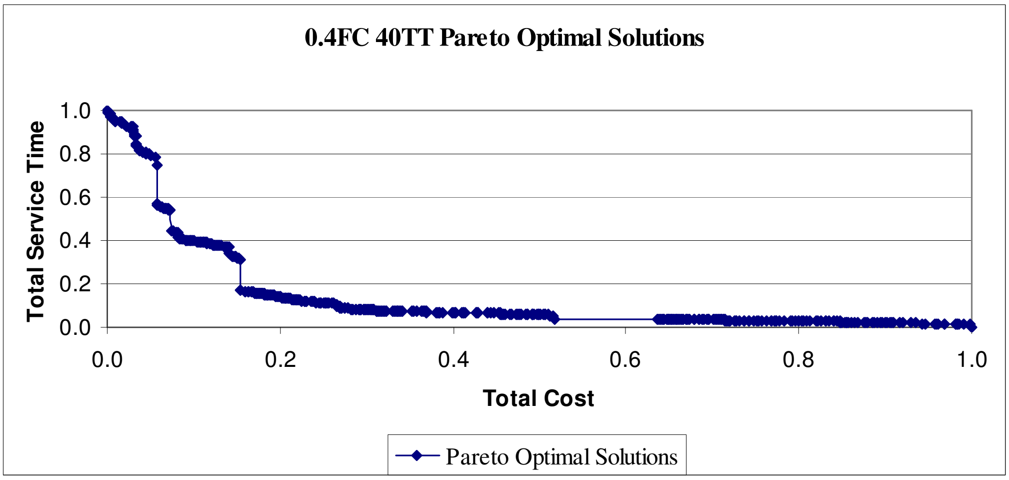

Figure 4.10 Efficient solutions of 40TT_0.4FC

Due to the fact that all replications have the same hub sets in the final population, we only solve the allocation problem for a selected replication since merging the hub sets would make no difference. Replication 5 is selected for computation. After the bounding procedure is applied, the allocations found are plotted in Figure 4.11.

Table 4.1 Efficient hub sets vs. the algorithm's hub sets for 40TT_0.4FC

BSAHLP_40TT_0.4FC	**Hub Sets found by the EA**				
Efficient Hub Sets	**Replication1**	**Replication2**	**Replication3**	**Replication4**	**Replication5**
32,40	32,40	32,40	32,40	32,40	32,40
25,40	25,40	25,40	25,40	25,40	25,40
25,32	25,32	25,32	25,32	25,32	25,32
14,29	14,29	14,29	14,29	14,29	14,29
14,25,40	14,25,40	14,25,40	14,25,40	14,25,40	14,25,40
14,25,38	14,25,38	14,25,38	14,25,38	14,25,38	14,25,38
14,25,29	14,25,29	14,25,29	14,25,29	14,25,29	14,25,29
14,19,40	14,19,40	14,19,40	14,19,40	14,19,40	14,19,40
14,19,38	14,19,38	14,19,38	14,19,38	14,19,38	14,19,38
10,40	14,19,29*	14,19,29*	14,19,29*	14,19,29*	14,19,29*
10,25,40	10,40	10,40	10,40	10,40	10,40
10,14,40	10,32*	10,32*	10,32*	10,32*	10,32*
10,14,29	10,25,40	10,25,40	10,25,40	10,25,40	10,25,40
40	10,14,40	10,14,40	10,14,40	10,14,40	10,14,40
32	10,14,38*	10,14,38*	10,14,38*	10,14,38*	10,14,38*
	10,14,29	10,14,29	10,14,29	10,14,29	10,14,29
	40	40	40	40	40
	32	32	32	32	32
# of Efficient Hub Sets found by EA / Total # of Efficient Hub Sets	15/15	15/15	15/15	15/15	15/15
CPU (sec)	28.7	20.8	30.2	18.3	18.9
*Almost efficient but inefficient hub sets with front ID 1 as set by the EA					

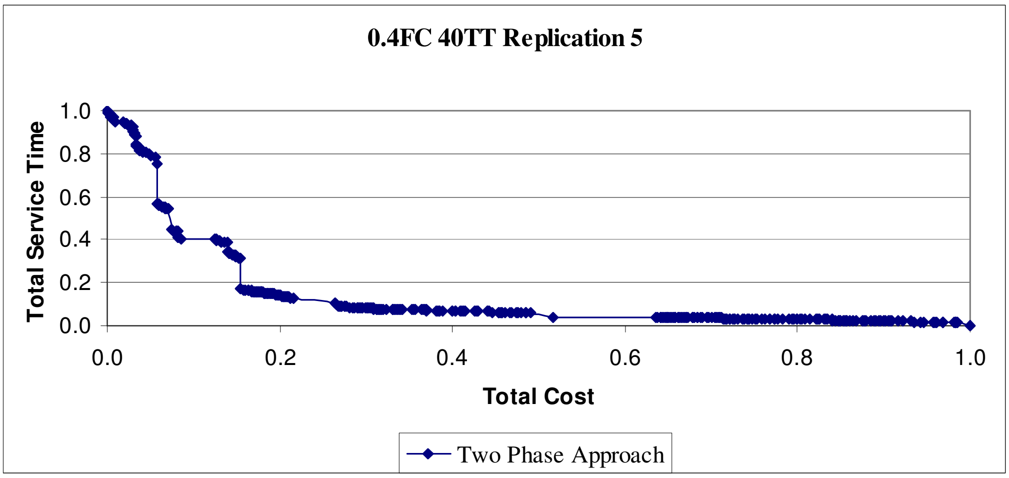

Figure 4.11 Allocations of Replication 5's hub sets for 40TT_0.4FC

Although we use all the efficient hub sets and solve the allocation problem for each, we cannot find all the efficient solutions on the Pareto-optimal front. The reason for that is the bounding procedure. Remember that we determine the bounds based on dominance with respect to the hypothetical points on the Lq curve. Thus, exclusion of some portion of a hub set using bounds may result in loss of efficient points on that portion since the assumption that there is an efficient solution very close to the hypothetical point may not hold. But, since this is a heuristic procedure, these losses are tolerable and actually, they do not have a significant effect on the performance of the suggested procedure.

In Table 4.2, we show the performance of our solution approach based on how good our approach approximated the efficient frontier. Looking at the dominated Hypervolume, the efficient solutions found dominate 99.83% of the objective space that is dominated by the true Pareto-optimal solutions. Moreover, the time spent for computation is about 5 times less with our procedure.

Table 4.2 Performance of the two-phase approach for 40TT_0.4FC

	H.V.R (%)	CPU Time (hr)
Finding all efficient solutions with Cplex	100.00	4.95
Replication 5 + Bounded allocations	99.83	1.03

BSAHLP_40TT_0.3FC

This problem has 24 efficient hub sets in total as found by the $\varepsilon-$constraint method. The Pareto-optimal solutions are given on Figure 4.12. In the EA, we took the initial population size and the upper limit for the final population as 40 and 50, respectively. Table 4.3 gives the hub sets found by EA in each of the five replications. According to the results, we are able to find most of the efficient hub sets in all replications. Moreover, the hub sets which are actually inefficient but put in the first front by EA contribute also to converging and diversity objectives in the absence of and as alternatives to efficient hub sets that could not be found by the algorithm.

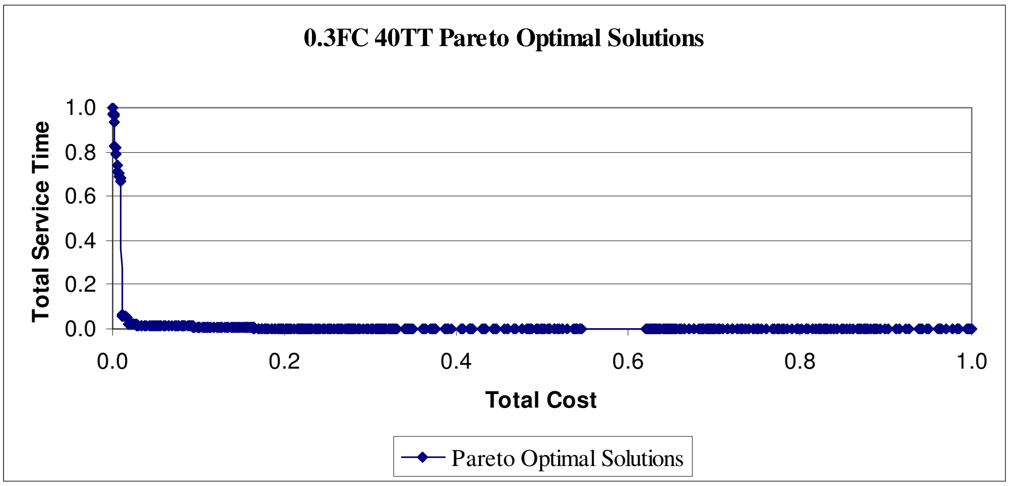

Figure 4.12 Efficient Solutions of 40TT_0.3FC

We first inputted the hub sets found by Replication 1 to the bounding procedure and solved the allocation problem within the determined bounds for each hub set, which generated the solutions on Figure 4.13. Then we did the same computations for the hub sets obtained by merging all five replications' hub sets. Figure 4.14 shows the solutions generated via merging of hub sets.

Table 4.3 Efficient hub sets vs. the algorithm's hub sets for 40TT_0.3FC

BSAHLP_40TT_0.3FC	Hub Sets found by the EA				
Efficient Hub Sets	**Replication1**	**Replication2**	**Replication3**	**Replication4**	**Replication5**
6,19,22,29	6,10,22,29*	32,40	6,11,22,29	6,11,22,29	6,11,22,29
6,11,22,29	32,40	25,40	6,10,22,29*	32,40	6,10,22,29*
32,40	25,40	25,32	32,40	25,40	32,40
25,40	25,32	14,25,40	25,40	25,32	25,40
25,32	14,26,38*	14,25,38	25,32	14,26,38*	25,32
14,25,40	14,25,40	14,25,30	14,26,38*	14,25,40	14,25,40
14,25,38	14,25,38	14,25,29	14,25,40	14,25,38	14,25,38
14,25,30	14,25,30	14,19,40	14,25,38	14,25,30	14,25,30
14,25,29	14,25,29	14,19,38	14,25,30	14,25,29	14,25,29
14,19,40	14,22,29*	14,19,30*	14,25,29	14,22,29*	14,22,29*
14,19,38	14,19,40	14,19,29	14,22,29*	14,19,40	14,19,40
14,19,29	14,19,38	10,40	14,19,40	14,19,38	14,19,38
11,22,29	14,19,30*	10,32*	14,19,38	14,19,30*	14,19,30*
10,40	14,19,29	10,25,40	14,19,30*	14,19,29	14,19,29
10,25,40	14,19,22,29*	10,14,40	14,19,29	14,19,22,29*	14,19,22,29*
10,14,40	11,14,22,29*	10,14,38	14,19,22,29*	11,22,29	11,22,29
10,14,38	10,40	10,14,30	11,22,29	11,14,29*	10,40
10,14,30	10,32*	10,14,29	10,40	10,40	10,32*
10,14,29	10,25,40	10,14,25,40	10,32*	10,32*	10,25,40
10,14,25,40	10,14,40	10,14,25,29	10,25,40	10,25,40	10,14,40
10,14,25,38	10,14,38	40	10,14,40	10,14,40	10,14,38
10,14,25,29	10,14,30	32	10,14,38	10,14,38	10,14,30
40	10,14,29		10,14,30	10,14,30	10,14,29
32	10,14,25,40		10,14,29	10,14,29	10,14,25,40
	10,14,25,29		10,14,25,40	10,14,25,40	10,14,25,29
	40		10,14,25,29	10,14,25,29	40
	32		40	40	32
			32	32	
# of Efficient Hub Sets found by EA / Total # of Efficient Hub Sets	20/24	20/24	22/24	22/24	22/24
CPU (sec)	128.8	96.1	156.9	97.1	115.8
*Almost efficient but inefficient hub sets with front ID 1 as set by the EA					

Figure 4.13 indicates that solutions obtained via Replication 1 well represent the efficient frontier except for those solutions on the minimum cost extreme. However, note that these solutions are not expected to be of much interest to the decision maker since they worsen the total service time objective too much to save only a little from the total cost. Thus, they are almost dominated solutions.

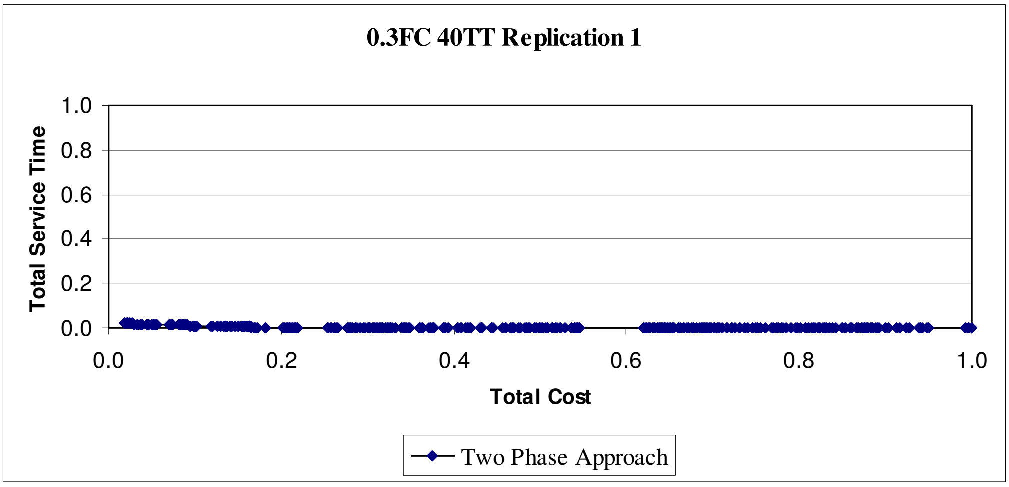

Figure 4.13 Allocations of Replication 1's hub sets for 40TT_0.3FC

We are able to find even the solutions on the minimum cost extreme (Figure 4.14) with the allocations of merged hub sets of replications.

In Table 4.4, we present the performance of our solution approach in finding the efficient allocations. Looking at the dominated Hypervolume, our procedure performs very well in approximating the efficient frontier. Moreover, the computation is about 5 times faster with our procedure.

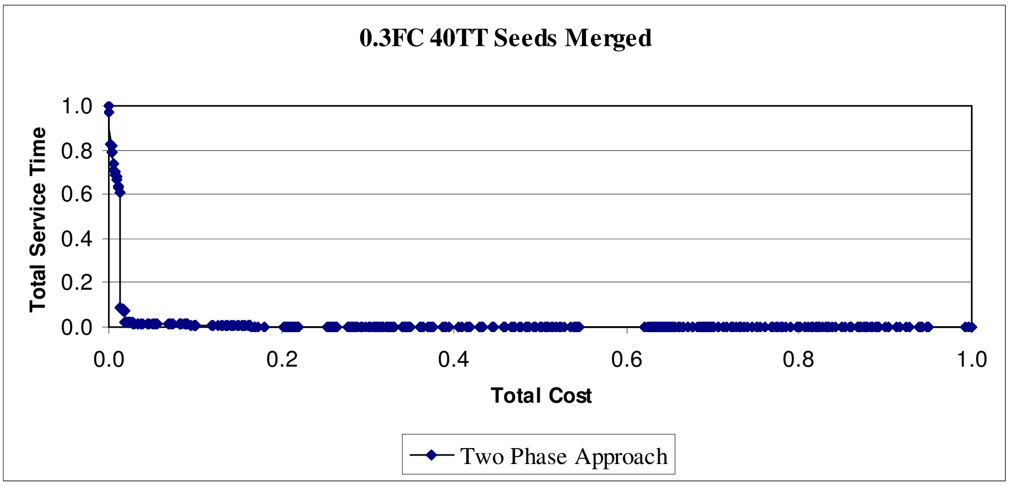

Figure 4.14 Allocations of all replication's hub sets for 40TT_0.3FC

Table 4.4 Performance of the two-phase approach for 40TT_0.3FC

	H.V.R (%)	CPU Time (hr)
Finding all efficient solutions with Cplex	100	5.97
Replication 1 + Bounded allocations	98.8	1.03
Merged Seeds + Bounded allocations	99.8	1.22

BSAHLP_40TT_0.2FC

We observe that the number of efficient hub sets is increased to 30 (Table 4.5) when fixed hubbing costs are decreased by 80%. The number of inefficient hub sets having front ID 1 also increases, which is expected due to the increased cardinality of the efficient hub sets. That is, one may find a hub set that behaves similar to an efficient hub set (consisting of relatively more hubs) by changing or deleting (adding) a single hub of (to) the efficient hub set. An example to these hub sets would be 6,10,14,26,29* in Replication 5, which might represent the region of Pareto front in the absence of the efficient hub set 6,10,14,29,35.

The spread of solutions found via Replication 2 is good (Figure 4.16) but as expected, not as good as of the solutions found via merging of hub sets (Figure 4.17).

Table 4.5 Efficient hub sets vs. the algorithm's hub sets for 40TT_0.2FC

BSAHLP_40TT_0.2FC	Hub Sets found by the EA				
Efficient Hub Sets	**Replication1**	**Replication2**	**Replication3**	**Replication4**	**Replication5**
6,11,22,25,29	6,10,22,29,35	6,14,19,29*	6,10,22,29,35	6,10,22,29,35	6,14,19,22,29*
6,10,22,29,35	6,10,22,29	6,14,19,24,29*	6,10,22,29	6,10,22,29	6,14,19,22,25,29*
6,10,22,29	6,10,14,29,35	6,14,19,24,25,38*	6,10,14,29,35	6,10,14,29,35	6,10,22,29
6,10,22,25,29	6,10,14,29	6,14,19,22,29*	6,10,14,29	6,10,14,29	6,10,22,25,29
6,10,14,29,35	6,10,14,26,29*	6,14,19,22,29,35*	6,10,14,26,29*	6,10,14,26,29*	6,10,14,29
6,10,14,29	6,10,14,25,40	6,14,19,22,25,29*	6,10,14,25,40	6,10,14,25,40	6,10,14,26,29*
6,10,14,25,40	6,10,14,25,38,40*	6,10,14,29,35	6,10,14,25,38	6,10,14,25,38	6,10,14,25,40
6,10,14,25,38	6,10,14,25,38	6,10,14,29	6,10,14,25,30	6,10,14,25,30	6,10,14,25,38
6,10,14,25,30	6,10,14,25,29,40	6,10,14,26,29*	6,10,14,25,29,40	6,10,14,25,29,40	6,10,14,25,30
6,10,14,25,29,40	6,10,14,25,29	6,10,14,25,40	6,10,14,25,29	6,10,14,25,29	6,10,14,25,29,40
6,10,14,25,29	6,10,14,22,29,35	6,10,14,25,38	6,10,14,24,29,35*	6,10,14,24,29,35*	6,10,14,25,29
6,10,14,24,25,38	6,10,14,22,29	6,10,14,25,30	6,10,14,24,29*	6,10,14,24,29*	6,10,14,24,29*
6,10,14,22,29,35	6,10,14,22,25,29	6,10,14,25,29,40	6,10,14,24,25,38	6,10,14,24,25,38	5,25,40
6,10,14,22,29	5,25,40	6,10,14,25,29	6,10,14,22,29,35	6,10,14,22,29,35	5,14,25,40*
6,10,14,22,25,29	5,14,26,38*	6,10,14,24,29*	6,10,14,22,29	6,10,14,22,29	5,14,25,30*
5,25,40	5,14,25,40*	6,10,14,24,29,35*	6,10,14,22,25,29	6,10,14,22,25,29	5,10,25,40*
32,40	5,10,25,40*	6,10,14,24,25,38	5,25,40	5,25,40	32,40
25,40	32,40	6,10,14,22,29,35	5,10,25,40*	5,14,25,40*	25,40
25,32,40	25,40	6,10,14,22,29	32,40	5,14,25,30*	25,32
25,32	25,32	5,25,40	25,40	5,10,25,40*	10,40
10,40	10,40	5,14,25,40*	25,32	32,40	10,32*
10,25,40	10,32*	5,14,25,30*	10,40	25,40	10,25,40
10,14,40	10,25,40	5,10,25,40*	10,32*	25,32	10,14,40
10,14,25,40	10,14,40	32,40	10,25,40	10,40	10,14,26,29*
10,14,25,38	10,14,38*	25,40	10,14,40	10,32*	10,14,25,40
10,14,25,30	10,14,26,38*	25,32	10,14,26,30*	10,25,40	10,14,25,38,40*
10,14,25,29,40	10,14,26,29*	10,40	10,14,26,29*	10,14,40	10,14,25,38
10,14,25,29	10,14,25,40	10,32*	10,14,25,40	10,14,26,30*	10,14,25,30
40	10,14,25,38,40*	10,25,40	10,14,25,38,40*	10,14,26,29*	10,14,25,29,40
32	10,14,25,38	10,14,40	10,14,25,38	10,14,25,40	10,14,25,29,32*
	10,14,25,29,40	10,14,26,30*	10,14,25,30	10,14,25,38,40*	10,14,25,29
	10,14,25,29,32*	10,14,26,29*	10,14,25,29,40	10,14,25,38	1,32*
	10,14,25,29	10,14,25,40	10,14,25,29,32*	10,14,25,30	1,14,19,22,25,29*
	10,14,22,29,35*	10,14,25,38,40*	10,14,25,29	10,14,25,29,40	1,10,14,26,29*
	10,14,22,29*	10,14,25,38	10,14,24,25,38*	10,14,25,29,32*	40
	10,14,22,25,29*	10,14,25,30	10,14,22,29,35*	10,14,25,29	32
	1,32*	10,14,25,29,40	10,14,22,29*	10,14,24,25,38*	
	1,10,14,26,29*	10,14,25,29,32*	10,14,22,25,29*	10,14,22,29,35*	
	40	10,14,25,29	1,32*	10,14,22,29*	
	32	10,14,24,25,38*	1,10,14,26,29*	10,14,22,25,29*	
		1,32*	40	1,32*	
		1,10,14,26,29*	32	1,10,14,26,29*	
		40		40	
		32		32	
# of Efficient Hub Sets found by EA / Total # of Efficient Hub Sets	24/30	24/30	27/30	27/30	22/30
CPU (sec)	91.8	104.1	106.3	66.1	58.4
*Almost efficient but inefficient hub sets with front ID 1 as set by the EA					

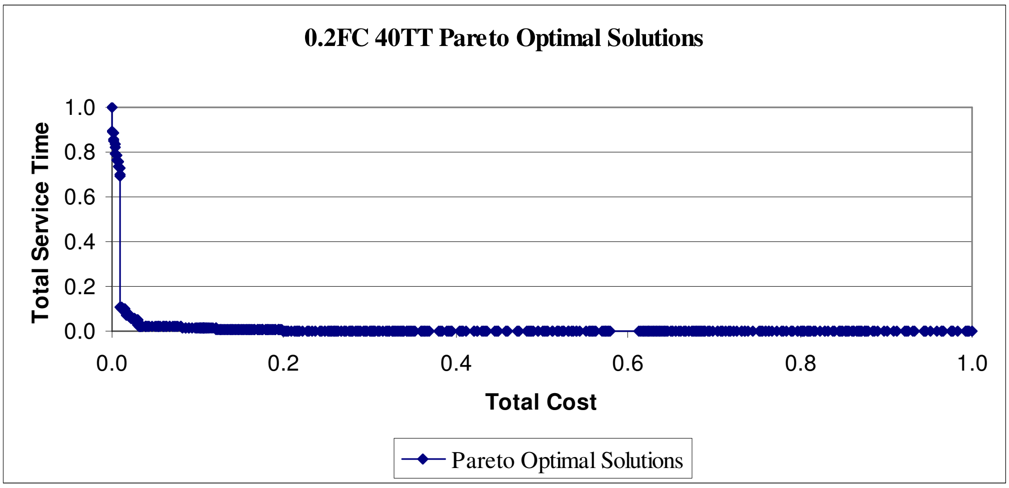

Figure 4.15 Efficient Solutions of 40TT_0.2FC

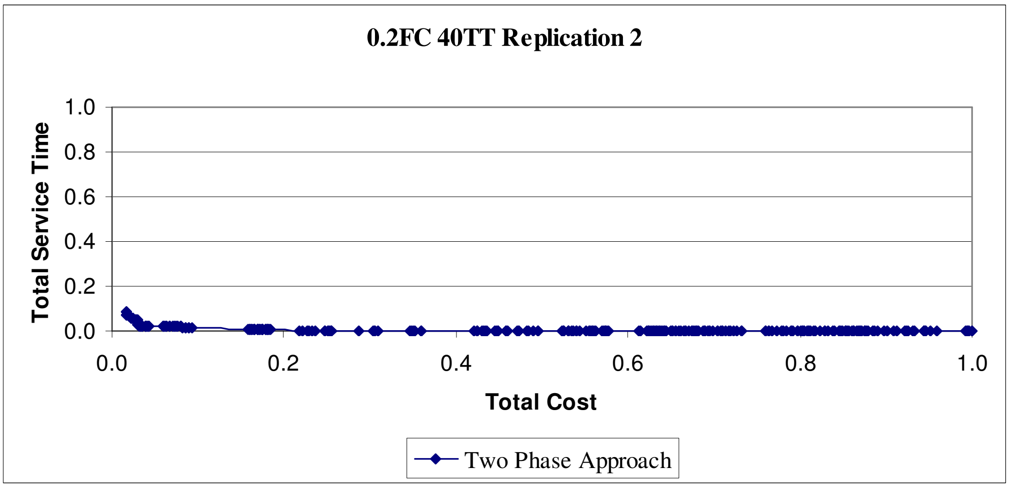

Figure 4.16 Allocations of Replication 2's hub sets for 40TT_0.2FC

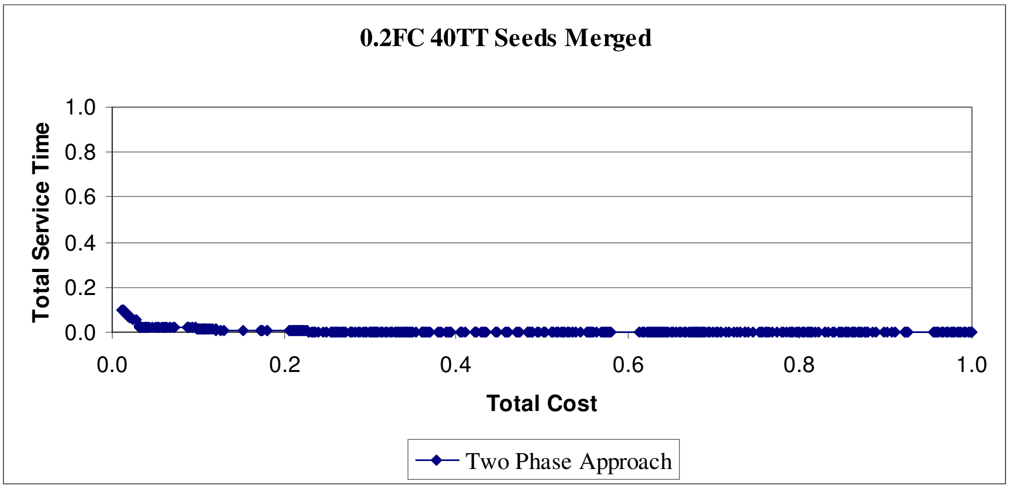

Figure 4.17 Allocations of all replication's hub sets for 40TT_0.2FC

Table 4.6 shows that the Hypervolume ratio is very close to 100% in both cases indicating that the solutions obtained are converging to the Pareto front and have good diversity in the criteria space. Moreover, the computation time with our procedure is about one sevenths of the time required for generating all efficient solutions with Cplex MIP solver.

Table 4.6 Performance of the two-phase approach for 40TT_0.2FC

	H.V.R (%)	CPU Time (hr)
Finding all efficient solutions with Cplex	100.00	14.55
Replication 2 + Bounded allocations	99.26	1.97
Merged Seeds + Bounded allocations	99.70	1.87

BSAHLP_40TT_0.1FC

As mentioned before, this is the most difficult one of the four problem instances. Thereby, we took the initial population size as 60 and the upper limit as 75. In this problem, the number of efficient hub sets increased as expected and amounted to 59 grounding to a significant increase in the number of efficient solutions. The plot of the efficient solutions is presented in Figure 4.18. Moreover, the efficient solutions to this problem involve hub sets with as many as 10 hubs (see Table A.1 in Appendix), pointing out the increased conflict in the problem nature, which also creates a handicap for the search of the EA.

The solutions obtained via Replication 4 only (Figure 4.19) have a good approximation of the efficient frontier, if not as good as those obtained from merging of all replications (Figure 4.20).

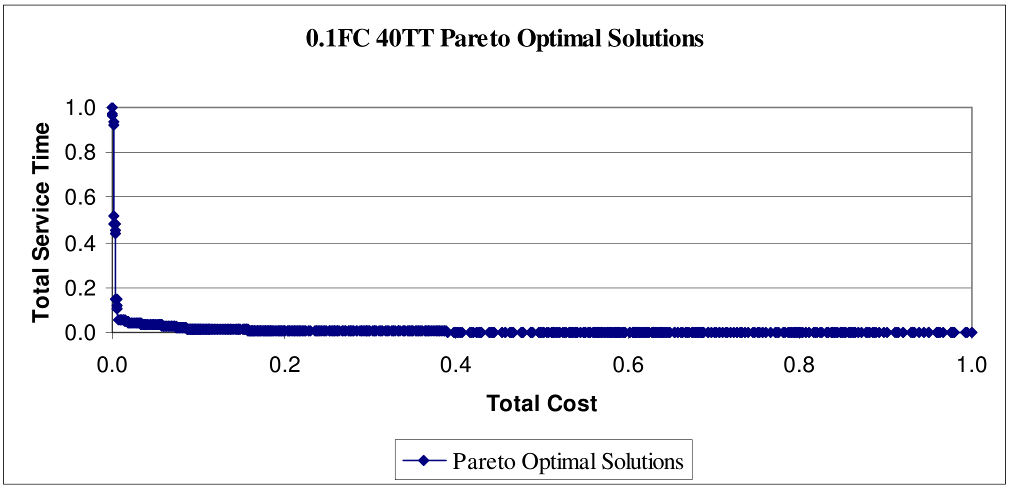

Figure 4.18 Efficient solutions of 40TT_0.1FC

Referring to Table 4.7, the performance of our approach is good in both of the single replication and the merged replications cases. Using the hub sets of Replication 4 only, we are able to generate efficient solutions that dominate 97.67% of the unit square in 7.6% of the time required for generating all efficient solutions via the $\varepsilon-$constraint method. If we can tolerate 15.8% of 31.23 hours, then we can generate efficient solutions that dominate 99.59% of the unit square, which may be preferred since location decisions are strategic level decisions.

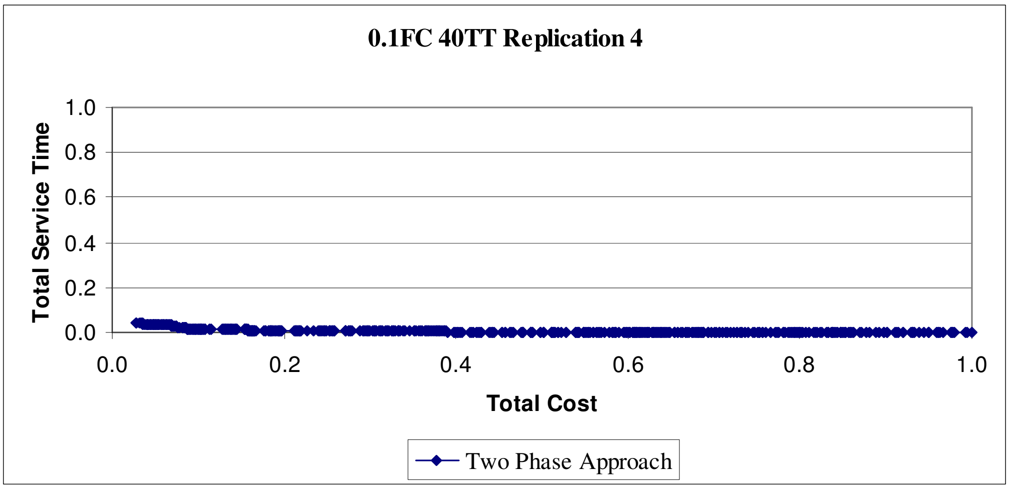

Figure 4.19 Allocations of Replication 4's hub sets for 40TT_0.1FC

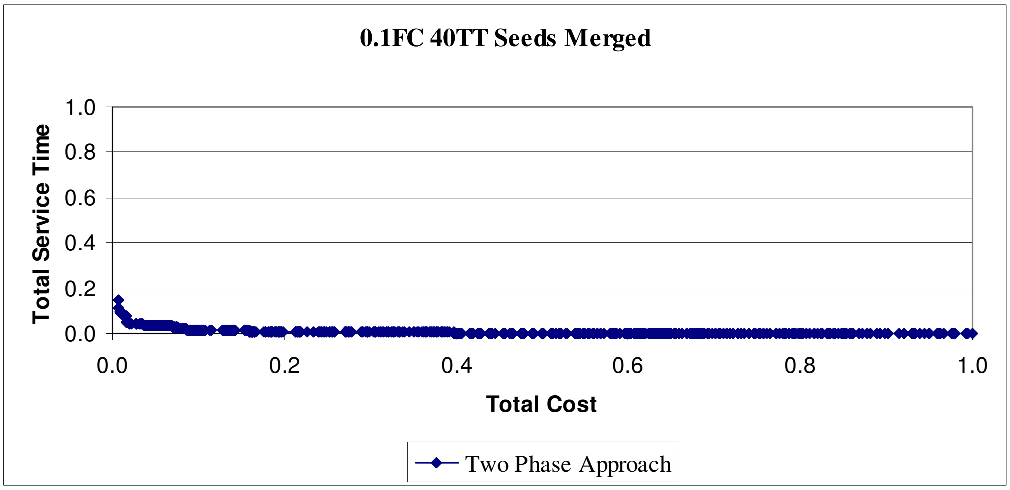

Figure 4.20 Allocations of all replication's hub sets for 40TT_0.1FC

Table 4.7 Performance of the two-phase approach for 40TT_0.1FC

	H.V.R (%)	CPU Time (hr)
Finding all efficient solutions with Cplex	100.00	31.23
Replication 4 + Bounded allocations	97.67	2.37
Merged Seeds + Bounded allocations	99.59	4.92

Discussion

In this chapter, a two-phase solution approach is proposed to approximate the efficient frontier. The set of located hubs are determined with an EA in the first phase. The members having front ID 1 in the final population of the EA are regarded as the efficient hub sets of the problem. However, at the end of the run, it is possible that hub sets with a higher front ID (i.e., with front ID 2) may also yield solutions on the global efficient frontier, since front IDs are determined based on hypothetical points on the representative Lq curves. But, this is seldom encountered in the computational experiments. In fact, we observed only one such hub set, which shows that the fitted Lq curves effectively represent the allocations of hub sets.

In the formulation of the problem, we ensure that each non-hub node is assigned to a hub node. This constraint prevents the no-hub case to be considered as an efficient solution of the problem. Note that comparing the no-hub case with a single-hub case would be meaningful since we do not enjoy economies of scale via inter-hub transfers in single hub case although we incur the fixed cost of hubbing.

There are different aspects that affect the performance of the two-phase approach. One of them is related with how well we fit the Lq curves to hub sets. The graphical analysis showed that the curves represent the allocations of hub sets well implying that the procedures we suggest to find the three solutions used to fit the curves proved effective. Thus, the suggested procedures are among the major contributions of this study. The evolutionary algorithm itself is another aspect that affects the performance since it constructs a basis via the located hub sets for the generation of efficient allocations. Thus, the higher the number of efficient hub sets found is with the EA, the larger the Hypervolume ratio is. Looking at the computational results, we can find most of the efficient hub sets with the EA, indicating a high performance for the approach. The bounding procedure is another major contribution of this study, which has important computational effects on the performance of the two-phase approach. Although we can find most of the efficient hub sets using the EA, finding all the allocations of these hub sets is neither effective nor efficient as discussed at the end of section 4.2. Determination of bounds for the efficient hub sets and solving the allocation problems within these bounds brings many savings by avoiding generation of globally inefficient allocations and thus, reducing the computation time. One other aspect would be the initial population generation. The procedure used in this work may be compared with others as a future research to find out how different ways of generating the initial population affect the overall performance.

CHAPTER 5

INTERACTIVE PROCEDURE

5.1 Introduction

Interactive procedures are iterative approaches, where we incorporate the preferences of the decision maker (DM) to the solution procedure progressively through the course of finding satisfactory solutions. In each iteration of an interactive procedure, a set of solutions are found and the decision maker is asked to select one of these solutions that represents his/her preference most. Then, using this information as an indicator of the DM's utility function and assuming the DM is consistent with his decisions, the search is directed through the preferred region of the DM, and the procedure continues until the DM is satisfied with the presented solutions. In this sense, interactive procedures are good for allowing the decision maker to make improved judgments using new information and for allowing corrections to the search process (Steuer, 1986).

With the availability of an interactive procedure, instead of spending much computational effort to generate all efficient solutions, we can just try to converge to the solutions that are of interest to the DM by directing the search towards his/her preferred region. Thus, the interactive procedures save us the time spent for generating the efficient solutions that do not have much utility to the DM.

In this chapter, we first explain the details of the interactive procedure and then give some computational examples.

5.2 The Interactive Procedure

We implement a variation of the interactive procedure suggested in Steuer and Choo (1983, pp.326-344), namely the Interactive Weighted Tchebycheff Procedure (IWTP). The IWTP is a weight space reduction based algorithm, which consists of sampling a number of λ-weighting vectors, solving the corresponding augmented weighted Tchebycheff programs, presenting the resulting diverse solutions to the DM and reducing the weight space around the preferred solutions of the DM. Different than IWTP, we run our EA for a short while and present the most diverse 5 nondominated solutions from the resulting criterion vectors to the DM. Thus, we do not select and utilize the λ-weights to find diverse solutions that are to be presented to the DM. After the DM chooses his/her preferred solution $z^{(h)}$ from diverse set of solutions, the $\lambda^{(h)}$ vector corresponding to this solution is given by

$$\lambda_i^{(h)} = \begin{cases} \dfrac{1}{(z_i^{(h)} - z_i^{**})}\left[\displaystyle\sum_{i=1}^{k} \dfrac{1}{(z_i^{(h)} - z_i^{**})}\right]^{-1} & if\ z_i^{(h)} \neq z_i^{**}\ for\ all\ i \\ 1 & if\ z_i^{(h)} = z_i^{**} \\ 0 & if\ z_i^{(h)} \neq z_i^{**}\ but\ \exists\, j \ni z_j^{(h)} = z_j^{**} \end{cases} \tag{5.1}$$

where z_i^{**} denotes the ideal point in the i^{th} objective and h denotes the iteration number. Using $\lambda^{(h)}$ weight vector, we can find the weight interval $\left[l_i^{(h+1)}, u_i^{h+1}\right]$ as follows:

$$\left[l_i^{(h+1)}, u_i^{(h+1)}\right] = \begin{cases} \left[0, r^h\right] & if\ \lambda_i^{(h)} - \dfrac{r^h}{2} \leq 0 \\ \left[1 - r^h, 1\right] & if\ \lambda_i^{(h)} + \dfrac{r^h}{2} \geq 1 \\ \left[\lambda_i^{(h)} - \dfrac{r^h}{2}, \lambda_i^{(h)} + \dfrac{r^h}{2}\right] & otherwise \end{cases} \tag{5.2}$$

where r^h is the weight space reduction factor that gets smaller as the number of iterations increase.

This weight interval is an approximate representation of the DM's preferred region in the solution space. Thus, we employ the weight interval in the evolutionary algorithm to favor the solutions that reside in the correspondent solution space of this weight interval as in Karahan (2008). Iteratively, this procedure allows us to get closer to the solutions that are most valuable to the DM.

The evolutionary algorithm is modified so as to keep those solutions in the final population, which are in the preferred weight space of the DM. To assure this, we simply increment the front ID of the solutions by 1 that reside out of the weight space. By doing so, only the nondominated solutions that belong to this weight space will have front ID 1, which will keep them in the final population.

At the end of each iteration, if the DM desires to continue the search by selecting a solution among those we present, then via narrowing down the weight space further we aim to converge to the solution or solutions that maximize the DM's utility. The procedure continues until the DM is happy with the results.

5.3 Some Computational Experiments

To illustrate the implementation of the approach, we give three example solutions on the efficient frontier of the 0.3FC 40TT problem (Figure 5.1), each of which is assumed to be the one that maximizes the DM's utility function. Hence, each solution is the one we want the procedure to converge to at the end of iterations. The DM is assumed to have an underlying Tchebycheff utility function and assumed to select consistently the solution maximizing this utility function each time we ask to choose one.

Example 1: Suppose that solution S1 is the most appealing solution to the DM. Then the underlying utility function of the DM is defined by the favorable weights of this solution, which we can find with equation (5.1). S1 has the scaled

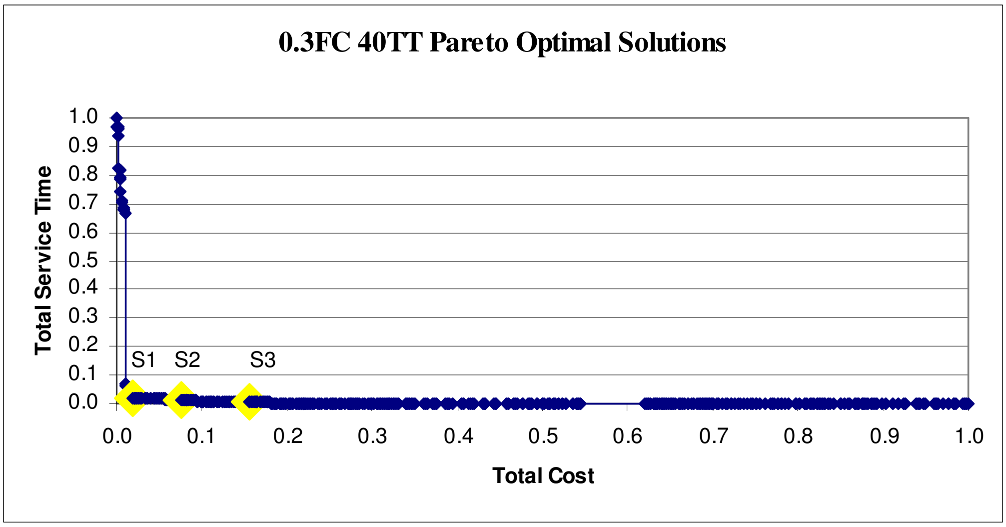

Figure 5.1 Example efficient solutions of 40TT_0.3FC

objective function values (0.016409, 0.018423) with favorable weights (0.528910, 0.471090) and it is an allocation of the hub set (14, 19, 29). After running the algorithm for a short time (1.5 seconds), the diverse set of five solutions selected among the nondominated solutions in the final population are shown in Table 5.1 with borders. We made the selection based on dividing the total cost range into equal intervals. Among these solutions (1, 9, 14, 17, 19), the DM is expected to choose the first one since it has the closest weights to the favorable weights of S1. Note that we only compare the first weight in the weight vector since the second weight (λ_2) is equal to 1-λ_1.

Next, we narrow down the weight space around the favorable weights of the selected solution using equation (5.2) and the weight interval becomes

$\left[l_1^{(2)}, u_1^{(2)}\right] = (0.220862, 0.668075)$. Then, we run the algorithm for 5000 generations (40 sec) giving this weight space as an input. The obtained nondominated solutions are presented in Table 5.2.

Table 5.1 Diverse set of solutions presented to the DM

No	Total Cost	Total Service Time	Tchebycheff Weight (λ_1)	Hub Sets
1	0.023219	0.018577	0.444468	14,29
2	0.052511	0.018057	0.255881	14,19,29,34
3	0.077455	0.017673	0.185781	14,23,25,29
4	0.088817	0.016422	0.156045	14,19,29,40
5	0.102687	0.014018	0.120115	14,25,29,40
6	0.135544	0.011145	0.075977	14,19,29,40
7	0.139130	0.009548	0.064219	14,25,29,40
8	0.217641	0.003040	0.013776	14,40
9	0.283596	0.002381	0.008326	14,40
10	0.344972	0.002348	0.006760	1,14,40
11	0.349408	0.001895	0.005394	14,40
12	0.421084	0.001450	0.003432	14,40
13	0.486555	0.001379	0.002826	1,40
14	0.492373	0.001270	0.002573	1,40
15	0.495548	0.000770	0.001551	40
16	0.650371	0.000632	0.000971	32,40
17	0.743586	0.000487	0.000655	32,40
18	0.844384	0.000354	0.000419	32,40
19	1.000000	0.000000	0.000000	32

Table 5.2 Solutions presented to the DM at Iteration 2

	Total Cost	Total Service Time	Tchebycheff Weight (λ_1)	Hub Set
1	0.019557	0.018018	0.479521	14,19,29
2	0.026026	0.016448	0.387249	10,14,29
3	0.033565	0.015615	0.317507	14,25,29
4	0.047816	0.015281	0.242183	10,14,25,29

Here, we observe that the first solution, having the same hub set with S1 and being the closest hypothetical point to S1, is the one that the DM would like to achieve. This shows that at Iteration 2, we converged to the solution that represents the utility maximizing real solution, S1, well. However, for the sake of completeness, we iterate once more to be sure that this solution keeps standing in the final population.

As we expect, the DM chooses the first solution in Table 5.2. We restrict the weight space further around this solution using equation (5.2) again. The new weight interval becomes $\left[l_1^{(3)}, u_1^{(3)}\right] = (0.379521, 0.579521)$. After we run the algorithm for 5000 generations, the resulting nondominated solutions are as shown in Table 5.3.

Table 5.3 Solutions Presented to the DM at Iteration 3

	Total Cost	Total Service Time	Tchebycheff Weight (λ_1)	Hub Set
1	0.019557	0.018018	0.479520958	14,19,29
2	0.026026	0.016448	0.38724867	10,14,29

We finalize the procedure with this iteration since we converged to the desired solution of the DM.

Example 2: This time suppose that the DM's underlying Tchebycheff utility function is maximized with S2. It is an allocation of hub set (14, 19, 38) and the solution has the criterion vector (0.074166, 0.010706) with favorable weights (0.126145, 0.873855). Suppose also that we present the same diverse set of solutions (Table 5.1) initially. The DM will select solution number 9 in Table 5.1 since its favorable weight (0.008326) is the closest one to that of S2 (0.074166). The weight space is now reduced around solution 9, and the algorithm is run. The solutions that are within this weight space and the most diverse five are shown in Table 5.4. Among the diverse set of five solutions (1, 14, 21, 23, 27), the DM will select solution number 14. After reducing the weight space around this solution, the solutions obtained and the most diverse five are shown in Table 5.5. Now, the DM's preference will be the solution having number 1 and note that this solution is actually the one the DM is willing to reach based on his/her Tchebycheff utility function. Since the weight space will be reduced around this solution's weight, it is guaranteed that this solution will remain in the population, thus convergence to the representative solution of S2 (solution 1) is assured in this example.

Table 5.4 Solutions Presented to the DM at Iteration 2

No	Total Cost	Total Service Time	Tchebycheff Weight (λ1)	Hub Set
1	0.026026	0.016448	0.387249	10,14,29
2	0.033565	0.015615	0.317507	14,25,29
3	0.047816	0.015281	0.242183	10,14,25,29
4	0.075415	0.010620	0.123438	14,19,38
5	0.086393	0.008357	0.088201	14,25,38
6	0.092934	0.008106	0.080226	10,14,38
7	0.137932	0.007729	0.053062	14,19,30
8	0.140993	0.006422	0.043564	14,19,40
9	0.149531	0.005535	0.035694	10,14,30
10	0.150084	0.003277	0.021368	14,25,40
11	0.164510	0.003117	0.018595	10,14,25,40
12	0.183140	0.002916	0.015673	10,14,40
13	0.230573	0.001995	0.008578	10,25,40
14	0.251246	0.001795	0.007094	10,40
15	0.293816	0.001509	0.005110	10,40
16	0.360661	0.001286	0.003553	10,40
17	0.403150	0.001243	0.003074	25,40
18	0.436943	0.001124	0.002566	25,40
19	0.443466	0.001106	0.002488	10,40
20	0.480724	0.001032	0.002142	25,40
21	0.495548	0.000770	0.001551	40
22	0.650371	0.000632	0.000971	32,40
23	0.743586	0.000487	0.000655	32,40
24	0.794377	0.000477	0.000600	10,32
25	0.810237	0.000364	0.000449	25,32
26	0.844384	0.000354	0.000419	32,40
27	1.000000	0.000000	0.000000	32

Example 3: In this last example, solution S3 is assumed to be the one the DM is willing to reach. S3 is an allocation of hub set (14, 25, 40). The criterion vector of this solution is (0.153911, 0.003144) and it has the favorable weights

(0.020021, 0.979979). We continued the iterations until convergence in this example. Based on the results, the representative solution that is closest to S3 is observed to remain in the population at each iteration. Hence, we could converge to the ideal solution of the DM under the assumed utility function in this example as well. All the tables of this example are included in the Appendix B.

In conclusion, we can state that converging to the preferred region of the DM through employing his/her preferences during the search is an effective alternative to approximating the entire efficient frontier. However, one should pay attention to the assumption the interactive procedure works under, which states that the DM is consistent with his decisions. In reality, this assumption may not hold with all the DMs. Hence, developing a mechanism to allow the DM change his decisions during the search could make the interactive procedure more realistic.

Table 5.5 Solutions Presented to the DM at Iteration

No	Total Cost	Total Service Time	Tchebycheff Weight (λ1)	Hub Set
1	0.075415	0.010620	0.123438	14,19,38
2	0.086393	0.008357	0.088201	14,25,38
3	0.092934	0.008106	0.080226	10,14,38
4	0.140993	0.006422	0.043564	14,19,40
5	0.150084	0.003277	0.021368	14,25,40
6	0.164510	0.003117	0.018595	10,14,25,40
7	0.183140	0.002916	0.015673	10,14,40
8	0.230573	0.001995	0.008578	10,14,38
9	0.251246	0.001795	0.007094	10,40
10	0.293816	0.001509	0.005110	10,40
11	0.360661	0.001286	0.003553	10,40
12	0.403150	0.001243	0.003074	25,40
13	0.436943	0.001124	0.002566	25,40
14	0.443466	0.001106	0.002488	10,40
15	0.480724	0.001032	0.002142	25,40
16	0.495548	0.000770	0.001551	40
17	0.650371	0.000632	0.000971	32,40
18	0.743586	0.000487	0.000655	32,40
19	0.794377	0.000477	0.000600	10,32
20	0.810237	0.000364	0.000449	25,32
21	0.844384	0.000354	0.000419	32,40
22	1.000000	0.000000	0.000000	32

CHAPTER 6

CONCLUSIONS

In this study, we propose a two-phase solution approach for solving a bicriteria single allocation hub location problem with time and cost related objectives. This problem is derived from the capacitated single allocation hub location problem, which is a well known problem having many applications in postal delivery systems. We develop an evolutionary algorithm to locate the hubs on the network as the first phase and then in the second phase, we employ a bounding procedure based on dominance relations. Using the determined bounds, we solve the allocation subproblem for each located hub set with Cplex MIP solver.

Involving both location and allocation decision components, the problem is combinatorial and gets more difficult to solve when the number of nodes increase. However, the results show that the evolutionary algorithm solves the location subproblem effectively in a small amount of computational time, and solving the corresponding allocation problems takes much less time than finding all efficient solutions just by using Cplex. Thus, solving large scale realistic problems becomes possible with our solution approach.

We also propose an interactive procedure that incorporates the preferences of the decision maker progressively and tries to converge to the solutions that are in the preferred region of the decision maker. The procedure worked well based on the computational results of three illustrative examples.

When solving the problem, in fact we seek the solutions that may reduce the total cost via excess utilization of some hub facilities or total service time via sacrificing from the total cost objective. These solutions are not observable when we use hard capacity constraints. However, our problem is more valid for designing a network from scratch, because we also find solutions that slightly improve one objective while significantly worsening the other. These solutions may be impractical for the networks that are not subject to change.

As a future research, the two-phase approach can be tested with larger scale problems (ie., problems up to 100 nodes). Since the number of efficient hub sets is not expected to increase sharply with a larger problem size, the corresponding allocation problems may be solved in a reasonable amount of time.

Instead of solving the allocation problems with Cplex, developing another evolutionary algorithm to consider both the locations of hubs and the corresponding efficient allocations is another future research direction. Such an algorithm would have a significant potential in reducing the total computation time.

For the hub networks that are already set up, minimization of the maximum service time at any hub may be worth being considered as the second objective instead of minimizing the total service time and the results can be compared with those of the existing work. In fact, this minmax type objective may even be considered as a third objective besides the two criteria handled in this study.

REFERENCES

Abdinnour-Helm, S., 1998. A hybrid heuristic for the uncapacitated hub location problem. European Journal of Operational Research 106 (2–3), 489–499.

Abdinnour-Helm, S., Venkataramanan, M.A., 1998. Solution approaches to hub location problems. Annals of Operations Research 78, 31–50.

Alumur S., Kara B.Y., 2008. Network hub location problems: The state of the art. European Journal of Operational Research, 190(1), 1-21.

Campbell, J.F., 1994. Integer programming formulations of discrete hub location problems. European Journal of Operational Research 72, 387–405.

Campbell, J.F., 1996. Hub location and the p-hub median problem. Operations Research 44 (6), 1–13.

Chen, J.F., 2008. A Heuristic for the Capacitated Single Allocation Hub Location Problem. In Alan H. S. Chan and Sio-Iong Ao (Eds.), Advances in industrial engineering and operations research, Volume 5 of Lecture Notes in Electrical Engineering, pp. 185-196, Springer.

Costa, M.G., Captivo, M.E., Climaco, J., 2008. Capacitated single allocation hub location problem – a bi-criteria approach. Computers & Operations Research 35, 3671 – 3695.

Cunha, C.B., Silva, M.R., 2007. A genetic algorithm for the problem of configuring a hub-and-spoke network for a LTL trucking company in Brazil. European Journal of Operational Research 179, 747–758.

Çamlar, O., 2005. Solution to Multi-Objective Hub Location Problem Using Evolutionary Algorithm – An Application to PTT Network, MSc. Thesis, Department of Industrial Engineering, Middle East Technical University, Ankara, Turkey.

Deb, K., A. Pratap, S. Agarwal, and T. Meyarivan, 2002. A Fast and Elitist Multiobjective Genetic Algorithm: NSGA–II. IEEE Transactions on Evolutionary Computation 6(2), 182–197.

Ebery, J., Krishnamoorthy, M., Ernst, A., Boland, N., 2000. The capacitated multiple allocation hub location problem: Formulations and algorithms. European Journal of Operational Research 120, 614–631.

Ernst, A.T., Krishnamoorthy, M., 1996. Efficient algorithms for the uncapacitated single allocation p-hub median problem. Location Science 4 (3), 139–154.

Ernst, A.T., Krishnamoorthy, M., 1999. Solution algorithms for the capacitated single allocation hub location problem. Annals of Operations Research 86, 141–159.

Goldberg, D.E., 1989. Genetic algorithms in search, optimization and machine learning. Addison-Wesley, Reading, Massachusetts.

Haimes, Y.Y., L.S. Lasdon and D.A. Wismer, 1971. On a bicriterion formulation of the problems of integrated system identification and system optimization. IEEE Transactions on Systems, Man and Cybernetics, Vol.1, No.3, pp.296-297.

Kara, B.Y., Tansel, B.C., 1998. On the allocation phase of the p-hub location problem. Technical Report, Bilkent University.

Karahan, İ., 2008. Preference-based flexible multiobjective evolutionary algorithms, MS Thesis, Middle East Technical University, Industrial Engineering Department, Ankara-Turkey.

Köksalan, M., 1999. A heuristic approach to bicriteria scheduling, Naval Research Logistics, 46, 777-789.

Köksalan, M., S. P. Phelps, 2007. An evolutionary metaheuristic for approximating preference-nondominated solutions. Informs Journal on Computing 19(2), 291–301.

Labbé, M., Yaman, H., Gourdin, E., 2005. A branch and cut algorithm for the hub location problems with single assignment. Mathematical Programming 102, 371–405.

O'Kelly, M.E., 1987. A quadratic integer program for the location of interacting hub facilities. European Journal of Operational Research 32, 393–404.

O'Kelly, M.E., 1992. Hub facility location with fixed costs. Papers in Regional Science 71 (3), 293–306.

Randall M., 2008. Solution approaches for the capacitated single allocation hub location problem using ant colony optimisation. Comput Optim Appl, 39: 239–261.

Soylu, B., 2007. An evolutionary algorithm for multiple criteria problems, MS Thesis, Middle East Technical University, Industrial Engineering Department, Ankara-Turkey.

Soylu, B. and M. Köksalan, 2006. A favorable weight based evolutionary algorithm for multiple criteria problems. Technical report, Middle East Technical University.

Srinivas, N. and K. Deb, 1994, Multiobjective function optimization using non-dominated sorting genetic algorithm. Evolutionary Computation Journal, Vol.2, No.3, pp. 221-248.

Stanimirović, Z., 2008. Solving the capacitated single allocation hub location problem using genetic algorithm, Computing and Informatics, Vol. 27, in press.

Steuer RE., 1986. Multiple criteria optimization. New York: Wiley.

Steuer R.E., Choo E.U., 1983. An interactive weighted Tchebycheff procedure for multiple objective programming. Mathematical Programming, (26), 326-344.

Tan, P.Z., Kara, B.Y., 2007. A hub covering model for cargo delivery systems. Networks 49 (1), 28–39.

Topcuoglu, H., Corut, F., Ermis, M., Yilmaz, G., 2005. Solving the uncapacitated hub location problem using genetic algorithms. Computers & OR 32 (4), 967–984.

Veldhuizen, D.Van, 1999. Multiobjective evolutionary algorithms: Classifications, analyzes, and new innovations," Air Force Inst. Technol., Dayton, OH, Tech. Rep. AFIT/DS/ENG/99-01.

Zitzler, E., L. Thiele, 1998. Multiobjective optimization using evolutionary algorithms - A comparative case study. In A. E. Eiben, T. Bäck, M. Schoenauer, and H.-P. Schwefel (Eds.), Parallel problem solving from nature - PPSN V, Volume 1498 of Lecture Notes in Computer Science, pp. 292–301. Springer, 96.

Zitzler, E., M. Laumanns, L. Thiele, 2002. SPEA2: Improving the strength pareto evolutionary algorithm. In K. Giannakoglou, D. Tsahalis, J. Periaux, P. Papailou, and T. Fogarty (Eds.), EUROGEN 2001. Evolutionary methods for design, optimization and control with applications to industrial problems, Athens, Greece, pp. 95–100.

APPENDIX A

RESULTS FOR 40TT_0.1FC PROBLEM

Table A.1 Efficient hub sets vs. the algorithm's hub sets for 40TT_0.1FC

BSAHLP_40TT_0.1FC	Hub Sets found by the EA				
Efficient Hub Sets	**Replication1**	**Replication2**	**Replication3**	**Replication4**	**Replication5**
5,8,10,14,26,40	9,32*	5,8,10,25,40*	5,8,10,14,26,40	5,8,10,14,26,40	5,8,10,14,26,40
5,8,10,14,26,30,32	5,8,10,14,26,40	5,8,10,14,26,40	5,8,10,14,26,32,40*	5,8,10,14,26,32,40*	5,8,10,14,26,30,32
5,8,10,14,26,30	5,8,10,14,26,32,40*	5,8,10,14,26,32,40*	5,8,10,14,26,30,32	5,8,10,14,26,30,32	5,8,10,14,26,30
5,8,10,14,26,29,40	5,8,10,14,26,30,32	5,8,10,14,26,30,32	5,8,10,14,26,30	5,8,10,14,26,30	5,8,10,14,26,29,40
5,8,10,14,26,29,32	5,8,10,14,26,30	5,8,10,14,26,30	5,8,10,14,26,29,40	5,8,10,14,26,29,40	5,8,10,14,26,29,32
5,8,10,14,24,26,30	5,8,10,14,26,29,40	5,8,10,14,26,29,40	5,8,10,14,26,29,32	5,8,10,14,26,29,32	5,8,10,14,25,40*
5,8,10,14,24,26,29	5,8,10,14,26,29,32	5,8,10,14,26,29,32	5,8,10,14,25,40*	5,8,10,14,25,40*	5,8,10,14,25,29,32*
5,8,10,14,24,25,29	5,8,10,14,25,40*	5,8,10,14,25,40*	5,8,10,14,25,29,32*	5,8,10,14,25,29,32*	5,8,10,14,24,26,30
5,10,25,40	5,8,10,14,25,29,32*	5,8,10,14,25,29,32*	5,8,10,14,24,26,30	5,8,10,14,24,26,30	5,8,10,14,24,26,29,40*
5,10,25,32,40	5,8,10,14,24,26,30	5,8,10,14,24,26,30	5,8,10,14,24,26,29,40*	5,8,10,14,24,26,29,40*	5,8,10,14,24,26,29
5,10,25,30,32	5,8,10,14,24,26,29,40*	5,8,10,14,24,26,29,40*	5,8,10,14,24,26,29	5,8,10,14,24,26,29	5,8,10,14,24,25,29
5,10,14,26,40	5,8,10,14,24,26,29	5,8,10,14,24,26,29	5,8,10,14,24,25,29	5,8,10,14,23,26,30*	5,8,10,14,23,26,30*
5,10,14,26,30,32	5,8,10,14,23,26,30*	5,8,10,14,24,25,29	5,8,10,14,23,26,30*	5,8,10,14,23,26,29,40*	5,8,10,14,23,26,29,40*
5,10,14,26,30	5,8,10,14,23,26,29,40*	5,8,10,14,23,26,30*	5,8,10,14,23,25,29*	5,8,10,14,21,26,29,40*	5,8,10,14,23,25,29*
5,10,14,26,29,40	5,8,10,14,21,26,29,40*	5,8,10,14,23,26,29*	5,8,10,14,22,24,25,29*	5,6,10,14,26,40*	5,8,10,14,22,24,25,29*
5,10,14,26,29,32	5,6,10,14,26,40*	5,8,10,14,23,26,29,40*	5,6,10,14,26,40*	5,6,10,14,26,29,32*	5,6,10,14,26,40*
5,10,14,25,40	5,6,10,14,26,29,32*	5,8,10,14,23,25,29*	5,6,10,14,26,29,32*	5,6,10,14,25,40*	5,6,10,14,26,29,32*
5,10,14,25,30,40	5,6,10,14,25,40*	5,8,10,14,22,25,29*	5,6,10,14,25,29,32*	5,6,10,14,25,29,32*	5,6,10,14,25,40*
5,10,14,25,30,32	5,6,10,14,25,29,32*	5,8,10,14,21,26,29,40*	5,6,10,14,24,25,29*	5,26,30,32*	5,6,10,14,25,29,32*
5,10,14,25,30	5,25,40*	5,25,40*	5,14,26,32,40*	5,25,40*	5,6,10,14,24,25,29*
5,10,14,24,26,29	5,14,26,40*	5,25,32,40*	5,14,26,30*	5,25,32,40*	5,26,30,32*
32,40	5,10,26,40*	5,14,26,32,40*	5,10,26,40*	5,14,26,40*	5,25,40*
25,32,40	5,10,26,30,32*	5,14,26,30,32*	5,10,26,30*	5,14,26,32,40*	5,25,32,40*
25,32	5,10,26,30*	5,14,26,30,32,40*	5,10,26,30,32*	5,14,26,30,40*	5,14,26,40*

Table A.1 (cont'd) Efficient hub sets vs. the algorithm's hub sets for 40TT_0.1FC

BSAHLP_40TT_0.1FC	**Hub Sets found by the EA**				
Efficient Hub Sets	**Replication1**	**Replication2**	**Replication3**	**Replication4**	**Replication5**
10,40	5,10,25,40	5,14,25,40*	5,10,25,40	5,14,26,30,32*	5,14,26,30,32*
10,32,40	5,10,25,32,40	5,10,26,30*	5,10,25,32,40	5,14,26,30,32,40*	5,14,26,30,32,40*
10,25,40	5,10,14,26,40	5,10,25,40	5,10,14,26,40	5,14,25,40*	5,10,26,30*
10,25,32,40	5,10,14,26,38,40*	5,10,25,32,40	5,10,14,26,32,40*	5,10,26,40*	5,10,25,40
10,14,25,40	5,10,14,26,32,40*	5,10,14,26,40	5,10,14,26,30,32	5,10,26,30*	5,10,25,32,40
10,14,25,30,32	5,10,14,26,30,32	5,10,14,26,38,40*	5,10,14,26,30	5,10,26,30,32*	5,10,25,30,32
1,5,8,14,19,24,25,30	5,10,14,26,30	5,10,14,26,32,40*	5,10,14,26,29,40	5,10,25,40	5,10,14,26,40
1,5,8,14,19,24,25,29	5,10,14,26,29,40	5,10,14,26,30,32	5,10,14,26,29,32	5,10,25,32,40	5,10,14,26,32,40*
1,5,8,10,14,26,30	5,10,14,26,29,32	5,10,14,26,30	5,10,14,25,32,40*	5,10,14,26,40	5,10,14,26,30,32
1,5,8,10,14,26,29,40	5,10,14,25,40	5,10,14,26,29,40	5,10,14,23,26,30*	5,10,14,26,32,40*	5,10,14,26,30
1,5,8,10,14,26,29,32	5,10,14,25,38,40*	5,10,14,26,29,32	40*	5,10,14,26,30,32	5,10,14,26,29,40
1,5,8,10,14,24,26,30	5,10,14,23,26,30*	5,10,14,25,40	32,40	5,10,14,26,30	5,10,14,26,29,32
1,5,8,10,14,24,26,29	40*	5,10,14,25,38,40*	25,40*	5,10,14,26,29,40	5,10,14,25,40
1,5,8,10,14,24,25,29	32,40	5,10,14,23,26,30*	25,32	5,10,14,26,29,32	5,10,14,23,26,30*
1,5,8,10,14,21,24,29,35	25,40*	40*	10,40	5,10,14,25,40	40*
1,5,8,10,14,21,24,25,29	25,32	32,40	10,32*	5,10,14,23,26,30*	32,40
1,5,8,10,14,19,24,25,30	10,40	25,40*	10,32,40	40*	25,40*
1,5,8,10,14,19,24,25,29	10,32,40	25,32	10,25,40	32,40	25,32
1,5,8,10,14,16,26,30	10,32*	10,40	10,14,26,30*	25,40*	10,40
1,5,10,14,26,40	10,25,40	10,32*	10,14,25,40	25,32	10,32*
1,5,10,14,26,30	10,25,32,40	10,32,40	1,5,8,10,14,26,40*	10,40	10,32,40
1,5,10,14,26,29,40	10,14,26,30*	10,25,40	1,5,8,10,14,26,30,32*	10,32*	10,25,40
1,5,10,14,26,29,32	10,14,25,40	10,25,32,40	1,5,8,10,14,26,30	10,32,40	10,25,32,40
1,5,10,14,24,26,29	1,5,8,10,14,26,40*	10,14,26,30*	1,5,8,10,14,26,29,40	10,25,40	10,14,26,30*
1,3,8,14,19,22,25,29	1,5,8,10,14,26,30,32*	1,5,8,10,14,26,30,32*	1,5,8,10,14,26,29,32	10,25,32,40	1,5,8,10,14,26,40*
1,3,8,14,19,22,24,25,29	1,5,8,10,14,26,30	1,5,8,10,14,26,30	1,5,8,10,14,25,29,32*	10,14,26,30*	1,5,8,10,14,26,30,32*
1,3,8,14,18,22,29,35	1,5,8,10,14,26,29,40	1,5,8,10,14,26,29,40	1,5,8,10,14,24,26,30	1,5,8,10,14,26,40*	1,5,8,10,14,26,30

Table A.1 (cont'd) Efficient hub sets vs. the algorithm's hub sets for 40TT_0.1FC

BSAHLP_40TT_0.1FC	Hub Sets found by the EA				
Efficient Hub Sets	**Replication1**	**Replication2**	**Replication3**	**Replication4**	**Replication5**
1,3,8,14,18,22,25,29 1,3,8,14,18,22,24,29,35 1,3,8,14,18,22,24,25,29 1,3,8,11,14,18,22,29,35 1,3,8,11,14,18,22,25,29 1,3,8,11,14,18,22,24,29,3 1,3,8,11,14,18,22,24,25,2 32	1,5,8,10,14,26,29,32 1,5,8,10,14,25,29,32* 1,5,8,10,14,24,26,30 1,5,8,10,14,24,26,29,40* 1,5,8,10,14,24,26,29 1,5,8,10,14,23,26,30* 1,5,8,10,14,23,26,29,40* 1,5,8,10,14,23,26,29* 1,5,8,10,14,21,23,26,29* 1,5,10,26,30,32* 1,5,10,25,32,40* 1,5,10,14,26,40 1,5,10,14,26,30,32* 1,5,10,14,26,30 1,5,10,14,26,29,40 1,5,10,14,26,29,32 1,5,10,14,25,40* 1,5,10,14,25,29,32* 1,5,10,14,23,26,30* 1,32* 32	1,5,8,10,14,26,29,32 1,5,8,10,14,24,26,30 1,5,8,10,14,24,26,29,40* 1,5,8,10,14,24,26,29 1,5,8,10,14,24,25,29 1,5,8,10,14,23,26,30* 1,5,8,10,14,23,26,29* 1,5,8,10,14,23,26,29,40* 1,5,8,10,14,23,25,29* 1,5,8,10,14,22,25,29* 1,5,8,10,14,21,24,26,29,40* 1,5,8,10,14,21,23,25,29* 1,5,8,10,14,21,22,25,29* 1,5,10,25,32,40* 1,5,10,14,26,40 1,5,10,14,26,30,32* 1,5,10,14,26,30 1,5,10,14,26,29,40 1,5,10,14,26,29,32 1,5,10,14,25,40* 1,5,10,14,25,29,32* 1,32* 32	1,5,8,10,14,24,26,29,40* 1,5,8,10,14,24,26,29 1,5,8,10,14,24,25,29 1,5,8,10,14,23,26,30* 1,5,8,10,14,23,26,29* 1,5,8,10,14,23,26,29,40* 1,5,8,10,14,23,25,29* 1,5,8,10,14,22,25,29* 1,5,8,10,14,22,24,25,29* 1,5,8,10,14,21,23,25,29* 1,5,8,10,14,21,22,24,25,29* 1,5,10,26,30,32* 1,5,10,25,32,40* 1,5,10,14,26,40 1,5,10,14,26,30,32* 1,5,10,14,26,30 1,5,10,14,26,29,40 1,5,10,14,26,29,32 1,5,10,14,25,40* 1,5,10,14,25,29,32* 1,5,10,14,23,26,30* 1,32* 32	1,5,8,10,14,26,30,32* 1,5,8,10,14,26,30 1,5,8,10,14,26,29,40 1,5,8,10,14,26,29,32 1,5,8,10,14,25,29,32* 1,5,8,10,14,24,26,30 1,5,8,10,14,24,26,29,40* 1,5,8,10,14,24,26,29 1,5,8,10,14,23,26,30* 1,5,8,10,14,23,26,29* 1,5,8,10,14,23,26,29,40* 1,5,10,25,32,40* 1,5,10,14,26,40 1,5,10,14,26,30,32* 1,5,10,14,26,30 1,5,10,14,26,29,40 1,5,10,14,26,29,32 1,5,10,14,25,40* 1,5,10,14,25,29,32* 1,5,10,14,23,26,30* 1,32* 32	1,5,8,10,14,26,29,40 1,5,8,10,14,26,29,32 1,5,8,10,14,25,29,32* 1,5,8,10,14,24,26,30 1,5,8,10,14,24,26,29,40* 1,5,8,10,14,24,26,29 1,5,8,10,14,24,25,29 1,5,8,10,14,23,26,29* 1,5,8,10,14,23,26,29,40* 1,5,8,10,14,23,25,29* 1,5,8,10,14,22,24,25,29* 1,5,8,10,14,19,24,29,34* 1,5,10,25,32,40* 1,5,10,14,26,40 1,5,10,14,26,30,32* 1,5,10,14,26,30 1,5,10,14,26,29,40 1,5,10,14,26,29,32 1,5,10,14,25,40* 1,5,10,14,25,29,32* 1,5,10,14,23,26,30* 1,32* 32
# of Efficient Hub Sets found by EA / Total # of Efficient Hub Sets	32/59	33/59	32/59	31/59	34/59
CPU (sec)	278.4	252.1	361.4	349.1	326.2
*Almost efficient but inefficient hub sets with front ID 1 as set by the EA					

APPENDIX B

RESULTS FOR INTERACTIVE PROCEDURE

Table B.1 Diverse set of solutions presented to the DM at iteration1

No	Total Cost	Total Service Time	Tchebycheff Weight (λ_1)	Hub Sets
1	0.023219	0.018577	0.444468	14,29
2	0.052511	0.018057	0.255881	14,19,29,34
3	0.077455	0.017673	0.185781	14,23,25,29
4	0.088817	0.016422	0.156045	14,19,29,40
5	0.102687	0.014018	0.120115	14,25,29,40
6	0.135544	0.011145	0.075977	14,19,29,40
7	0.139130	0.009548	0.064219	14,25,29,40
8	0.217641	0.003040	0.013776	14,40
9	0.283596	0.002381	0.008326	14,40
10	0.344972	0.002348	0.006760	1,14,40
11	0.349408	0.001895	0.005394	14,40
12	0.421084	0.001450	0.003432	14,40
13	0.486555	0.001379	0.002826	1,40
14	0.492373	0.001270	0.002573	1,40
15	0.495548	0.000770	0.001551	40
16	0.650371	0.000632	0.000971	32,40
17	0.743586	0.000487	0.000655	32,40
18	0.844384	0.000354	0.000419	32,40
19	1.000000	0.000000	0.000000	32

Table B.2 Diverse set of solutions presented to the DM at iteration 2

No	Total Cost	Total Service Time	Tchebycheff Weight (λ1)	Hub Set
1	0.026026	0.016448	0.387249	10,14,29
2	0.033565	0.015615	0.317507	14,25,29
3	0.047816	0.015281	0.242183	10,14,25,29
4	0.075415	0.010620	0.123438	14,19,38
5	0.086393	0.008357	0.088201	14,25,38
6	0.092934	0.008106	0.080226	10,14,38
7	0.137932	0.007729	0.053062	14,19,30
8	0.140993	0.006422	0.043564	14,19,40
9	0.149531	0.005535	0.035694	10,14,30
10	0.150084	0.003277	0.021368	14,25,40
11	0.164510	0.003117	0.018595	10,14,25,40
12	0.183140	0.002916	0.015673	10,14,40
13	0.230573	0.001995	0.008578	10,25,40
14	0.251246	0.001795	0.007094	10,40
15	0.293816	0.001509	0.005110	10,40
16	0.360661	0.001286	0.003553	10,40
17	0.403150	0.001243	0.003074	25,40
18	0.436943	0.001124	0.002566	25,40
19	0.443466	0.001106	0.002488	10,40
20	0.480724	0.001032	0.002142	25,40
21	0.495548	0.000770	0.001551	40
22	0.650371	0.000632	0.000971	32,40
23	0.743586	0.000487	0.000655	32,40
24	0.794377	0.000477	0.000600	10,32
25	0.810237	0.000364	0.000449	25,32
26	0.844384	0.000354	0.000419	32,40
27	1.000000	0.000000	0.000000	32

The solution the DM is willing to reach resides in the above population. It is solution number 10 with hub set (14, 25, 40). The DM prefers solution number 14, which has the closest favorable weight to S3.

Table B.3 Diverse set of solutions presented to the DM at iteration 3

No	Total Cost	Total Service Time	Tchebycheff Weight (λ1)	Hub Set
1	0.075415	0.010620	0.123438	14,19,38
2	0.086393	0.008357	0.088201	14,25,38
3	0.092934	0.008106	0.080226	10,14,38
4	0.140993	0.006422	0.043564	14,19,40
5	0.150084	0.003277	0.021368	14,25,40
6	0.164510	0.003117	0.018595	10,14,25,40
7	0.183140	0.002916	0.015673	10,14,40
8	0.230573	0.001995	0.008578	10,14,38
9	0.251246	0.001795	0.007094	10,40
10	0.293816	0.001509	0.005110	10,40
11	0.360661	0.001286	0.003553	10,40
12	0.403150	0.001243	0.003074	25,40
13	0.436943	0.001124	0.002566	25,40
14	0.443466	0.001106	0.002488	10,40
15	0.480724	0.001032	0.002142	25,40
16	0.495548	0.000770	0.001551	40
17	0.650371	0.000632	0.000971	32,40
18	0.743586	0.000487	0.000655	32,40
19	0.794377	0.000477	0.000600	10,32
20	0.810237	0.000364	0.000449	25,32
21	0.844384	0.000354	0.000419	32,40
22	1.000000	0.000000	0.000000	32

The solution we aim to reach is still in the population, this time with number 5. The DM prefers solution number 10 in this iteration and the weight space is reduced accordingly.

Table B.4 Diverse set of solutions presented to the DM at iteration 4

No	Total Cost	Total Service Time	Tchebycheff Weight (λ1)	Hub Set
1	0.086393	0.008357	0.088201	14,25,38
2	0.092934	0.008106	0.080226	10,14,38
3	0.135360	0.007911	0.055217	14,26,38
4	0.140993	0.006422	0.043564	14,19,40
5	0.150084	0.003277	0.021368	14,25,40
6	0.164510	0.003117	0.018595	10,14,25,40
7	0.183140	0.002916	0.015673	10,14,40
8	0.230573	0.001995	0.008578	10,25,40
9	0.251246	0.001795	0.007094	10,40
10	0.293816	0.001509	0.005110	10,40
11	0.360661	0.001286	0.003553	10,40
12	0.403150	0.001243	0.003074	25,40
13	0.436943	0.001124	0.002566	25,40
14	0.443466	0.001106	0.002488	10,40
15	0.480724	0.001032	0.002142	25,40
16	0.495548	0.000770	0.001551	40
17	0.650371	0.000632	0.000971	32,40
18	0.743586	0.000487	0.000655	32,40
19	0.794377	0.000477	0.000600	10,32
20	0.810237	0.000364	0.000449	25,32
21	0.844384	0.000354	0.000419	32,40
22	0.909091	0.000000	0.000000	32

The reduced weight space could only eliminate the first solution in Table B.3 and the algorithm found an additional solution (No 3 in Table B.4) besides those found at iteration 3. Coincidentally, the comments are exactly the same with those made in the above iteration.

Table B.5 Diverse set of solutions presented to the DM at iteration 5

No	Total Cost	Total Service Time	Tchebycheff Weight (λ1)	Hub Set
1	0.150084	0.003277	0.021368	14,25,40
2	0.164510	0.003117	0.018595	10,14,25,40
3	0.183140	0.002916	0.015673	10,14,40
4	0.230573	0.001995	0.008578	10,25,40
5	0.251246	0.001795	0.007094	10,40
6	0.293816	0.001509	0.005110	10,40
7	0.360661	0.001286	0.003553	10,40
8	0.403150	0.001243	0.003074	25,40
9	0.436943	0.001124	0.002566	25,40
10	0.443466	0.001106	0.002488	10,40
11	0.480724	0.001032	0.002142	25,40
12	0.495548	0.000770	0.001551	40
13	0.650371	0.000632	0.000971	32,40
14	0.743586	0.000487	0.000655	32,40
15	0.794377	0.000477	0.000600	10,32
16	0.810237	0.000364	0.000449	25,32
17	0.844384	0.000354	0.000419	32,40
18	1.000000	0.000000	0.000000	32

At this iteration, the number of solutions in the population is reduced to 18 from 22. The solution representing S3 is still in the population with number 1. This time the DM will select this solution, which is presented to him/her as a diverse solution. Continuing the iterations, the solutions in Table B.6 are obtained.

Table B.6 Diverse set of solutions presented to the DM at iteration 6

No	Total Cost	Total Service Time	Tchebycheff Weight (λ1)	Hub Set
1	0.183140	0.002916	0.015673	10,14,40
2	0.164510	0.003117	0.018595	10,14,25,40
3	0.150084	0.003277	0.021368	14,25,40

This finalizes the iterations since we converged to the solution (No 3 in Table B.6) that best represents solution S3.