PERSONNEL BUS ROUTING PROBLEM: FORMULATION AND SOLUTION METHOD

A THESIS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF MASTER OF SCIENCE

By

Fatma Gzara

June, 1997

PERSONNEL BUS ROUTING PROBLEM: FORMULATION AND SOLUTION METHOD

A THESIS

SUBMITTED TO THE DEPARTMENT OF INDUSTRIAL ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCES

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Fatma Gzara

June, 1997

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assist. Prof. Mustafa Ç. Pınar(Principal Advisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Barbaros Ç. Tansel

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Osman Oğuz

Approved for the Institute of Engineering and Sciences:

Prof. Mehmet Baray
Director of Institute of Engineering and Sciences

ABSTRACT

PERSONNEL BUS ROUTING PROBLEM: FORMULATION AND SOLUTION METHOD

Fatma Gzara
M.S. in Industrial Engineering
Supervisor: Assist. Prof. Mustafa Ç. Pınar
June, 1997

In this thesis, we tackle the problem faced by many companies who offer transportation services to their personnel. We would refer to it as the Personnel Bus Routing Problem. The transportation services offered require a fleet of vehicles and a set of routes to daily transport personnel back and forth from the company to their residences. Hence, the problem of designing a transportation service system consists of three related subproblems: determine bus stops, assign residents to these stops and generate routes that visit the bus stops. The problem is significantly more complicated than conventional vehicle routing problems. It is compounded by several factors such as the heterogeneity of the fleet of vehicles, and the system efficiency that is measured by the transportation costs as well as by the level of personnel satisfaction. Moreover, the problem size is large because of the number of personnel to be serviced at a time and the dispersion of their residences on a large geographical area. We present in this thesis a multi-objective formulation of the problem and develop a heuristic method to generate solutions to it. The heuristic solution method is composed of two parts. A clustering part where clusters are generated each of which is to be serviced by one vehicle. In the second part, bus stops are located, residents are assigned to these bus stops, and routes are constructed simultaneously.

Key words: Bus Routing, Clustering, Branch and Bound, Heuristics.

ÖZET

PERSONEL OTOBÜS ÇIZELGELEME PROBLEMI: FORMÜLASYON VE ÇÖZÜM YÖNTEMI

Fatma Gzara
Endüstri Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Yrd. Doç. Mustafa Ç. Pınar
Haziran, 1997

Bu tez çalışmasında birçok firmanın personeline sunduğu ulaşım servislerinin optimum çalıştırılması problemi ele alınmıştır. Bu probleme Personel Otobüs Çizelgeleme Problemi adı verilmiştir. Ulaşım servisleri bir otobüs filosu ile belli güzergahlar üzerinde şehir ile firma arasinda çalışırlar. Sistemin tasarımında üç karar aşaması bulunur: otobüs duraklarının belirlenmesi, personelin bu duraklara dağıtımı ve güzergah seçimi. Bu özellikler problemi literatürde çok çalışılmış araç çizelgeleme probleminden daha karmaşık hale getirir. Ayrıca karar mekanizmaları otobüslerin ekonomik işletimine ek olarak sistemi kullanan çalışanların da memnuniyetini dikkate almak zorundadır. Bir diğer zorluk ise otobüs filosunda değişik tipte araçlar bulunmasıdır. Bütün bu kısıtlar altında büyük ölçekli bir karar problemine varılır. Bu tez çalışmasında problemin çok amaçlı bir formülasyonu ve sezgisel bir çözüm yöntemi önerilmiştir. Çözüm yöntemi iki aşamalıdır. Birinci aşama bir gruplama aşamasıdır. Bu aşamada, duraklar her biri bir araç tarafından ziyaret edilecek şekilde gruplanır. Ikininci aşamada, durakların içinden kullanıma açılacak olanlar seçilir, çalışanlar bu duraklara dağıtılır ve en iyi güzergah seçimi yapılır.

Anahtar sözcükler: Araç Çizelgeleme, Gruplama, Dal ve Sınır Metodu, Sezgisel Yöntemler.

To my mother,
and to the memory
of my father

ACKNOWLEDGEMENT

I am indebted to Assist. Prof. Mustafa Ç. Pınar for his invaluable guidance, encouragement and the enthusiasm which he inspired on me during this study.

I am also indebted to Assoc. Prof. Barbaros Ç. Tansel, and Assoc. Prof. Osman Oğuz for showing keen interest on the subject matter and accepting to read and review this thesis.

I would like to thank Alev Kaya, Mohamed Riahi, Souheyl Touhami, and Wail Gueaieb for their help during the preparartion of this thesis.

I would like also to thank my graduate collegues, and the technical and academic staff of Bilkent University.

Finally, I would like to thank Samir Elhedhli for everything.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Inspired from a real-life problem proposed by an electronics company in the city of Ankara, we define the Personnel Bus Routing Problem (PBRP), propose a comprehensive multi-objective mathematical formulation, and devise a heuristic solution method to generate solutions to the mathematical model.

Many companies offer transportation services to their personnel. The service can be described as follows: At the beginning of each working day, vehicles start at the company, traverse a specified itinerary, pick up employees at the assigned bus stops, and go back to the company. Similarly, at the end of the working day, vehicles pick up employees from the firm and following the reverse itinerary, drop them at the assigned bus stops. Offering such a service requires a fleet of vehicles and a set of routes to daily transport personnel back and forth from the company to their residences. The problem is then to determine a set of bus stops, assign employees to these bus stops, and generate routes that visit the located bus stops. We call this problem the "personnel bus routing problem".

We classify the PBRP as a class of the family of vehicle routing problems (VRP), that belongs to the category of location routing problems (LRP). Location routing problems are vehicle routing problems that require both routing and location decisions where location decisions affect the routing decisions and vice-versa. Vehicle routing problems arise in a large number of practical situations concerning the distribution of commodities. Both vehicle routing problems and location routing problems are difficult problems and are in most of the instances NP-hard problems except for some well-defined cases. As a result, most of the work done on VRP's and LRP's has been concerned with building heuristics.

The PBRP is a large-scale problem and is defined on a large geographical residential area. Moreover, it has a multi-objective nature. The main objectives of interest are employees' satisfaction with the transportation services they get and total routing costs. In addition, one main distinction of the problem we consider is the heterogeneity of the fleet of vehicles. Very few studies in the literature of vehicle routing and location routing problems assumed heterogeneous fleet.

In this thesis, we present a mathematical formulation that encompasses all the objectives of the PBRP and that also assumes a heterogeneous fleet of vehicles. Moreover, the mathematical formulation is characterised by its generality for it can easily be adapted to cover other variants of the problem. We then develop a two-phase heuristic algorithm based on this formulation. The solution methodology is of the type cluster-first route-second. In phase one, clusters of bus stops are generated via a branch and bound heuristic algorithm developed to solve a structured generalized assignment problem. The branch and bound algorithm has the following main characteristics.

- It uses the ideas of the tree search algorithms for the set partitioning problem.
- It uses column generation iteratively to form potential clusters.
- It is powerful in the sense that many realistic route constraints and cost functions can be handled easily.
- It succeded in producing cluster shapes that approximate routes by using the different distance measures that we propose.

- It handles fleet homogeneity as well as heterogeneity equally.
- It can easily be adapted to solve location problems such as the capacitated facility location problem.

Phase two concerns locating bus stops, allocating employees to these bus stops, and constructing bus routes on each of the clusters independently. The main part of the routing algorithm is an improvement scheme that we propose to improve the quality of the routing solution. The proposed solution methodology is implemented and tested with data from a real-life problem. The tests are conducted to validate the solution methodology, to test the performance of the clustering and routing algorithms, and to test the impact of different algorithmic strategies on the solutions obtained.

The organization of the thesis is as follows. In Chapter 2, the related literature on the vehicle routing problem is reviewed. Then, the characteristics of the PBRP are discussed through a comparison with vehicle routing problems, and the mathematical formulation is described in Chapter 3. Chapter 4 is devoted to the heuristic solution method we propose to solve the PBRP. Numerical testing of both phases of the algorithm is provided in Chapter 5. The thesis concludes with some remarks and suggestions for future research.

Chapter 2

Literature Review

The vehicle routing problem can be defined as the problem of designing a set of pickup and/or delivery routes optimal with respect to some specified criteria subject to a set of side constraints [18]. Routes initiate at one or several depots and are traversed by a fleet of vehicles with fixed characteristics to cover a set of customers each with a known location and a known demand for some commodity. There are many variations to the vehicle routing problem such as the location routing problem, the school bus routing problem (SBRP), the vehicle routing problem with time windows (VRPTW) and the dispatching problem [20]. The Vehicle routing problem arises in a large number of practical situations concerning the distribution of commodities. e.g. retail distribution, school bus routing, mail and newspaper delivery, municipal waste collection, and fuel oil delivery. The vehicle routing problem is characterized by its basic components, the optimization criteria and the side constraints [8].

2.1 Characteristics of the VRP

2.1.1 Basic Components of the VRP

The basic components of the VRP are: the nature of demand, the type of commodities to be transported, the pickup and delivery points, and the fleet of vehicles.

The Nature of Demand. In most vehicle routing problems, demand is known at the time the routes are to be constructed: this is the static case. In the dynamic case, some of the demands are known at the time of route construction and others become available in real time during execution of the routes [24]. In the static situation, routes are final once built up, while in the dynamic situation routes usually change in the process of execution to satisfy newly received demands. Dynamic situations arise mostly in pickup and delivery problems.

The Type of Commodities. In the literature, there is a big variety in the commodities transported. Commodities range from human beings, as in the case of school bus routing where school children are serviced, to hazardous materials. Other types of commodities are consumer goods as in the case of distribution of consumer goods from factories to warehouses and/or customers, newspaper delivery and garbage collection. These are "simple packages" [3] that in most cases do not cause any additional complications to the vehicle routing problem. On the other hand, in the case of school children, the school bus routing problem is more complicated because of the several additional goals, e.g. the efficiency and equity of service [23], that may be of concern. Moreover, transportation of hazardous materials adds significant complications to the problem since more attention is paid to the geographical characteristics of the route and the demographic distribution of the area it crosses rather than its length.

Pickup and Delivery Points. In most vehicle routing problems, the pickup points are located at the depot and the delivery points correspond to demand points or customer locations. The distribution of consumer goods from factories (depots) to warehouses (delivery points) is an example of such problems.

The depot is usually the place where vehicles start and end their routes [24]. Depending on the number of depots, the problem can be classified as a single-depot or a multi-depot problem. In multi-depot problems, depots may be independent, meaning that each depot has its own fleet of vehicles and geographical customer area to serve. Here, the problem reduces to several single-depot vehicle routing problems. In other cases depots are interdependent, that is, vehicles starting at a certain depot may end up at a different depot to load and restart. Consequently, depots can not be dealt with in isolation and the resulting vehicle routing problem should be solved as a whole.

Delivery points may be known and fixed so that the routing decision involves determining which of these points will be serviced by which vehicle and what route each vehicle will follow. In other situations, delivery points are to be chosen from a set of potential positions resulting in an additional location decision. In this case the problem is known as the location routing problem.

In some vehicle routing problems, the pickup and delivery points are the same, that is pickup and delivery occur at the same point. As an example, in school bus routing, the school acts as the delivery point or depot while bus stops are pickup points in the process of transporting students to school. This is reversed when students are transported back to their houses: the pickup points reduce to a single point at the school and bus stops serve as delivery points.

The Fleet of Vehicles. In all vehicle routing problems, vehicles are supposed to have a known capacity. The fleet of vehicles is most of the time supposed to be homogeneous. That is, all vehicles in the fleet have the same capacity. However, it may be the case that the fleet is heterogeneous, meaning that there are different vehicle types with different carrying capacities. This results in a richer problem with the additional decision of which vehicle type to serve which route. Other characteristics of the vehicles may include speed, fuel consumption, appropriateness to transport the required commodities etc. These characteristics do not have much relevance to the routing decision except in terms of the fixed cost incurred when a vehicle is used and the fuel usage costs which are directly proportional to the length of the route.

2.1.2 Optimization Criteria of the VRP

A wide variety of optimization criteria is found in the VRP literature. The most common ones are discussed below.

1. **Number of Routes.** Generally, each route is supposed to be serviced by one vehicle. So, minimizing the number of routes is equivalent to minimizing the number of vehicles to be used. The number of vehicles to be used is held to a minimum because the capital cost per bus is significantly larger than the incremental cost per year per bus as reported by Bowerman et al. [3] .

2. **Total Route Length.** This is the total length of all routes generated. The length of a route is the total distance traveled between different pickup and delivery points on the route. The length of the route contributes to the incremental cost of traveling as well as to the time duration of the trip along the route.

3. **Route Duration.** This includes travel times, loading and unloading times, and break times. This is relevant in cases where vehicles are used continuously during some time period, so that minimal duration of the route allows a higher number of trips done by the corresponding vehicle.

4. **Level of Customer Service.** In the case of consumer commodities, the level of customer service corresponds to the customer receiving the demand in time and as expected. In the case of transportation of people,

level of customer service includes service equity [23], duration of the trip and distance that customers need to walk to reach the pickup points.

5. **Load Balancing.** This involves minimizing the variation in the load of commodities on each vehicle. This arises when demand changes occasionally so that minor changes in demand are fulfilled without resolving the VRP.

The first two criteria are the most common criteria included in the objective function of vehicle routing problems, while the others arise in specific real-life problems.

2.1.3 Routing Constraints

Apart from the vehicle capacity constraints and the assignment constraints that guarantee all demand requirements to be fulfilled, some side constraints related to practical characteristics of the VRP are imposed. Such constraints are:

1. Upper bound on the number of delivery/pickup points on a single route.
2. Upper bound on the route length.
3. Upper bound on route duration.

2.2 VRP Models and Solution Methods

2.2.1 VRP Basic formulations

By considering various practical features that arise in vehicle routing problems, we identify a large number of models in the literature. Generally, vehicle routing models are interrelated in terms of the objective function terms and constraints basic to the conventional VRP, but differ in one or several additional components that are relevant to some practical characteristics of the real-life problem under consideration. Kulkarni and Bhave [17] discuss various mathematical formulations for the VRP. As classified by Magnanti [20] the basic formulations of the conventional VRP are of three types:

- set-covering formulation,
- commodity flow based formulation, and
- vehicle flow based formulation.

In the set-covering formulation, first demand points are assigned to vehicles then a route is constructed for each vehicle to cover the demand points assigned to it. Hence, basically we want to find the best assignment of demand points to vehicles according to some criterion which is usually minimizing the total routing costs. This represents the idea behind the cluster-first route-second approach to solving vehicle routing problems.

The commodity flow based formulation, as the name implies, controls both the flow of vehicles as well as the flow of commodities transported by these vehicles through Traveling Salesman Problem (TSP)-like assignment constraints. The formulation makes sure that exactly one vehicle enters and leaves each demand point, that no transshipment of commodities between any two demand points exceeds the vehicle capacity, and that commodities flow between two demand points only if there is a vehicle traveling between them.

The vehicle flow based formulation is basically an extension of the traveling salesman problem formulation. In addition to the assignment constraints that ensure that exactly one vehicle enters and leaves each demand point and the vehicle capacity constraints, the formulation uses sub-tour elimination constraints that prohibit the formation of sub-tours that do not include the depot.

The above formulations are integer programming ones, a dynamic programming formulation of the basic VRP was proposed by Christofides, Mingozzi and Toth [6]. The dynamic programming function represents the minimum cost of supplying a subset of demand points by using only a subset of the available vehicles. The dynamic programming recursion is then initialized for one vehicle and solved by increasing the number of vehicles at each step.

2.2.2 VRP Solution Methods

A great deal of the work done on the VRP has been concerned with building heuristics. Exact solution methods are devised only for some well-defined VRP's because of the nature of the problem itself [8]:

- Almost all formulations use assignment and TSP sub-tour elimination constraints which make the problem difficult to solve.
- In many cases the objective function turns to be nonlinear and/or multi-objective which adds one more level of complication to the problem.
- Usually real life problems are relatively large problems which makes solving the problem to optimality time consuming and in most of the cases beyond the capacity of most sophisticated available software packages.

The heuristics developed in past work are based on the integer programming formulations discussed above. According to Fisher and Jaikumar [12] the existing heuristics for the VRP can be classified into four classes:

1. tour building heuristics,
2. tour improvement heuristics,
3. two-phase methods, and
4. incomplete optimization methods.

Tour building heuristics begin with infeasible assignments, then feasible routes are built by adding a link at a time between two customers, every time the vehicle capacity constraints are checked for violation. The link to be added is chosen according to some measure of cost savings. The algorithm developed by Clarke and Wright [9] is an example of tour building heuristics. The main steps of the algorithm are:

1. initially every demand point is assigned to a different vehicle, consequently as many vehicles as demand points are used,

2. combine two demand points by assigning them to the same vehicle, if this would result in using only one of the two vehicles and also reduce the solution cost. Demand points are chosen if their combination results in a maximum cost saving and is feasible with respect to vehicle capacity constraints,

3. the combined demand points are now regarded as a single point and Step 2 is performed again until no more combinations are feasible.

The routes can be constructed either sequentially or in parallel. Routes are formed sequentially when demand points are added to the current route until vehicle capacity is exhausted. Then, a new route is constructed. Routes are formed in parallel when partial routes are constructed for every vehicle simultaneously. The Clarke and Wright algorithm was modified by defining different measures of cost savings.

Tour improvement heuristics begin with a feasible assignment of demand points to vehicles. Then, at each iteration some combination of links are exchanged and a check is made to verify if the exchange is feasible and if it results in cost reduction. Tour improvement heuristics are based on the Lin and Lin-Kernighan heuristic for the traveling salesman problem.

In phase one of the two-phase methods, demand points are assigned to vehicles so that vehicle capacity constraints are satisfied. In phase two, routes are constructed for each vehicle using TSP heuristics [8]. Cluster-first route-second algorithms are examples of two-phase methods. Others are algorithms developed by Gillett and Miller [15], and Fisher and Jaikumar [12]. Gillett and Miller use a "sweep" algorithm in phase one with customers represented in a polar coordinate system and distance between customers as cost. The first phase of the algorithm developed by Fisher and Jaikumar performs a parallel clustering by solving optimally a generalized assignment problem.

Incomplete optimization methods are essentially optimization algorithms such as branch and bound that are constructed to terminate prior to optimality.

Chapter 3

PBRP Characteristics and Comparison to VRP's

The personnel bus routing problem is essentially a vehicle routing problem compounded by several practical issues that bring additional complexity to the conventional vehicle routing problem. In vehicle routing terminology, the problem can be described as follows. A set of employees are to be transported back and forth from the firm (or, factory) to their residences. For this purpose a set of bus stops is located, a set of routes is constructed, and a fleet of vehicles is put in operation on the routes to transport employees (or, residents).

The problem has two symmetric instances. At the beginning of each working day, vehicles start at the firm, traverse a specified itinerary, pick up residents (or, employees) at the assigned bus stops, and go back to the firm. In this situation, the firm represents the depot and the delivery point, and the bus stops represent the pickup points. Similarly, at the end of the working day, vehicles pick up employees from the firm and following the inverse itinerary drop them at the assigned bus stops. In this situation, the bus stops represent the delivery points, and the firm represents the depot and the only pickup point. Hence, solving one instance of the problem is sufficient to construct the solution for both instances by using the same routes and the symmetric (with respect to both sides of a route) bus stops to the ones located in the solved instance.

In both instances of the problem, we find a distinction between the demand points and the pickup and/or delivery points. Demand points are employees' residences (or, home addresses), while pickup and delivery points are the bus stops and/or the firm. This results in an additional decision level to the vehicle routing problem. The decision concerns which residents are to be assigned to which bus stops in a way that minimizes both transportation costs and residents inconvenience. Consequently, the personnel bus routing problem consists of three interrelated subproblems:

- allocate residents to bus stops,
- locate a set of bus stops among the potential sites, and
- find a series of routes to be traversed by a fleet of vehicles.

Problems with these characteristics are known as location routing problems (LRP).

3.1 Location Routing Problems

Location routing problems are those problems that require both routing and location decisions and where location decisions affect the routing decisions and vice-versa. Laporte [19] classifies location routing problems according to

- the number of layers involved in the problem,
- the type of decision at each layer, and
- the interaction between the layers.

Layers represent the physical components of the problem, such as the depot, the pickup and delivery points and demand points. The decision is either a location or a routing decision. And the interaction between layers can be described by the distribution mode used by vehicles flowing between different layers or within the same layer. The distribution mode is either a round trip (a trip to and from a single element in a layer) or a tour (a round trip through several elements that may belong to more than one layer). Location-routing problems are further discussed in Laporte [19].

According to the above classification scheme, the PBRP has three layers:

1. layer one is the firm or depot,
2. layer two includes the bus stops or pickup and delivery points, and
3. layer three is composed of the set of residences or demand points.

Moreover, trips are made by the residents at the third layer to reach their assigned bus stops at layer two. Residents are then picked up in round-trips and brought to the company by vehicles. Routes interact between layers one and two, while residents walking to the bus stops cause the interaction between layers two and three. The location decisions are made at layer two: a set of bus stops is located among all potential bus stop sites at layer two. For an illustration see Figure 3.1 below.

In most cases, location routing problems are NP-hard problems [19]. As a result, most of the past work done on LRP's concentrated on developing heuristics that exploited the special structure of LRP's by decomposing the problem into its three subproblems and solving the resulting problems sequentially or simultaneously (in parallel). Most of the existing heuristics are one of the two types:

- location allocation routing (LAR), and
- allocation routing location (ARL).

Location allocation routing methods consist of three steps:

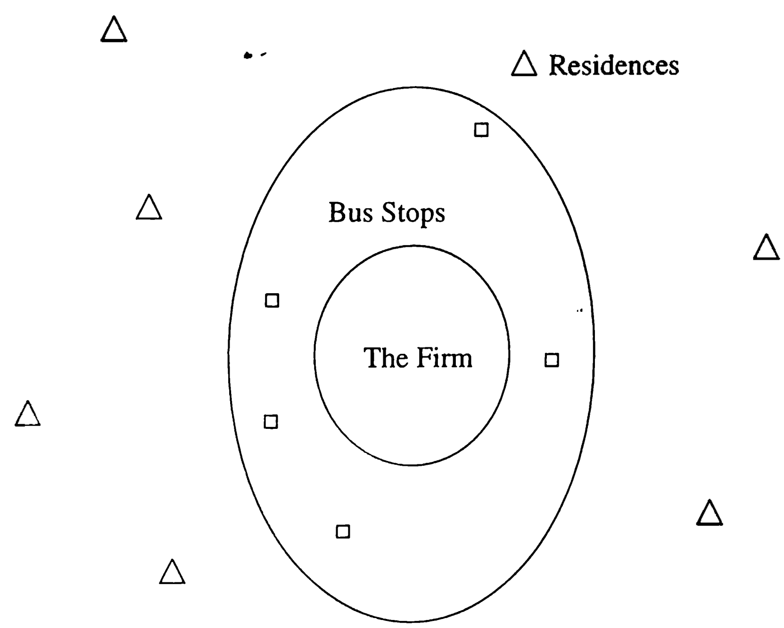

Figure 3.1: Layer Diagram for Location Routing Problems

- pickup and/or delivery points (in our case, bus stops) are located,
- demand points (in our case, residents) are allocated to pickup and/or delivery points, and
- routes are constructed to cover the pickup and/or delivery points located.

The three steps are either performed sequentially or combined. LAR methods are of the cluster-first route-second type algorithms. Usually demand points are allocated to pickup points and/or delivery points such that each group of delivery and/or pickup points corresponds to a route.

Allocation routing location methods are composed of two steps:

- routes are constructed with the assumption that all potential sites for pickup and/or delivery points are available to use, and
- a nonempty subset of potential sites is dropped and step one is performed again with the new set of potential sites.

The algorithm stops when dropping any nonempty subset of sites results in an infeasible problem or when a stable solution is reached. A stable solution is found when removing any subset of the available sites does not result in any cost savings. Generally, ARL methods are of the route-first cluster-second type of heuristics. In the first step, long routes are constructed to visit all potential sites. In the second step, pickup and/or delivery are located and vehicle routes constructed.

Generally, the criteria of optimization in LRP's encompass routing costs, vehicle capital costs, and depot operating costs. However, the problem we are concerned with has a more complicated objective. The complication comes from the term that measures residents' inconvenience. Residents represent both the customers and the commodities, hence their satisfaction of the system both includes the level of service as well as the distance they have to walk to reach for the bus stops they are assigned to. System efficiency involves such criteria as load-balancing of vehicles, ability to absorb small changes in residences or in number of employees to be transported. Here, we note the similarity of the personnel bus routing problem to the school bus routing problem (SBRP) which we discuss below.

3.2 The School Bus Routing Problem

The school bus routing problem is a location routing problem with the distinction that the commodities to be transported are school children. School bus routing problems are of two types: urban school bus routing, and rural school bus routing. While a lot of research was conducted on the first type of problem, the second attracted little attention.

In urban school bus routing problems, the number of students to be served may be relatively high. However, the geographical area under consideration is not large resulting in a relatively high ratio to the population density compared to the case of personnel transportation. The reason behind this is the policy of assigning children to schools in the same district as or in the surrounding areas of their residences. In contrast, it's not common practice to hire employees according to the area they live in. Moreover, it's not customary of employees to live in areas around the company since companies are usually built out of residential areas. Hence the difference in geographical customer area for both problems shows that the personnel bus routing problem is more complicated than school bus routing.

3.2.1 Solution Methods for SBRP

Because of the similarity between the SBRP and the problem of designing a personnel transportation service system, we find it useful to review the existing solution methods for the SBRP. All the existing procedures of the SBRP that we are aware of are heuristic methods and are of two types: cluster-first route-second, and route-first cluster-second .

Cluster-First Route-Second Procedures

In the cluster-first route-second procedures, first clusters of bus stops are formed. These clusters are built to satisfy some side constraints such as vehicle capacity constraints. Then, routing is done over the set of bus stops in each cluster to find the minimum total route length. The algorithm developed by Dulac, Ferland and Fogues [10] is a cluster-first route-second heuristic. The main steps of the algorithm are:

1. locate students on street segments on which they live, then assign each student to an incident street intersection or equivalently street node,
2. choose a subset of the incident street nodes to be potential bus stop sites,
3. select the bus stop with maximum number of students within walking distance and assign to it all students within walking distance. The

walking distance is a quantity judged to be the maximum distance a student can walk to reach the bus stop he/she is assigned to,

4. repeat step three until all students are assigned to some bus stop.

At this level, all students are allocated to some bus stops, so the problem reduces to a vehicle routing problem with the selected bus stops as demand points. Then, the next step is to solve the resulting VRP using the Clarke and Wright method [9].

In the first part of the algorithm by Dulac et al., students are assigned to bus stops by considering the student walking distance only, and no attention is paid to the effect of these assignments on the routing decision. As a result, the solution may contain more routes than necessary, this in turn would result in a higher total route length than in the case of considering the routing decision at the allocation step. In order to overcome this deficiency, Chapleau et al. [4] introduced a new distance measure and used it to group incident nodes into clusters that estimated the route length. Their clustering approach is as follows:

1. each student is assigned to an incident node as in the Dulac et al. algorithm,
2. determine the minimum necessary number of clusters, where each cluster size is approximately equal to the vehicle capacity. The minimum necessary number of clusters is given by the total number of residents divided by vehicle capacity,
3. each cluster specifies a one-route problem for which stops must be located and route generated independently of the others.

The clustering approach of Chapleau et al. is based on a distance measure that estimates the potential detour induced by the inclusion of a new node in a cluster. Different shapes of clusters are obtained through the variation of a penalty factor included in the definition of distance. The only constraint on the clusters formed is the vehicle capacity. The clusters generated at the first step are allowed to violate the vehicle capacity constraints. Exchange heuristics are then used to eliminate the violation and to improve the compactness of clusters.

Bowerman, Hall and Calamai [3] use a cluster-first route-second heuristic to solve a multi-objective mathematical formulation of the USBRP. The formulation has the total route length and the number of routes as objectives. It also includes minimizing the student walking distance, and load and length balancing of routes. The heuristic of Bowerman et al. is divided in two parts: a districting or equivalently clustering algorithm and a routing algorithm. In the districting algorithm, minimizing the number of routes and load-balancing are the objectives dealt with. To form clusters, a multi-objective VRP is solved. The VRP is defined on the school as the depot, street intersections or nodes as demand points and the number of students assigned to each node as the level of demand at that node. The solution of the VRP gives the assignment of nodes to routes and these give the clusters of students.

Then the routing algorithm is performed on each cluster to locate bus stops and generate routes. In this second part, both the route length and the walking distance criteria are minimized. A weighting procedure is used in the objective function to reflect the relative importance of each of the terms. The algorithm is performed in three main steps:

1. find a set of bus stops among the potential sites in a cluster and assign all students in the cluster to the bus stops in that set, such that every student is within walking distance from the bus stop he/she is assigned to,

2. on each set of bus stops, generate a school bus route,

3. find the set of stops that has the least total weighted distance, then add bus stops with no assignments to the solution in order to reduce the objective function value.

Route-First Cluster-Second Procedures

The route-first cluster-second heuristics find the shortest route that visits all the demand points through solving a traveling salesman problem, then break this route into a set of shorter routes, each to be traversed by one vehicle. Then, the routes are found so that they are feasible with respect to the vehicle capacity constraints and any other side constraints on the route structure or length. The algorithm developed by Bodin and Berman [2] is a route-first cluster-second heuristic. The algorithm can be described as follows:

1. assign students to potential bus stop locations which are termed ministops by Bodin and Berman,
2. among the set of ministops, find the set of bus stops that serve the school,
3. assign each student to the nearest bus stop from the ministop,
4. find a tour on the bus stops that have some student assignments,
5. break the tour into a set of feasible routes.

3.3 PBRP Formulation

In the light of the discussion above, we would like to highlight some salient characteristics of the PBRP. Generally, the number of employees that get such services counts for over 1000 employees. Employees residences are usually spread in the city so that all the city map will be under study in solving the problem. Hence, the problem size is relatively large compared both to most of the VRP sizes solved in past work and to the capacity of most sophisticated available solution tools.

Second, we consider a heterogeneous fleet of vehicles with respect to vehicle capacity. In most of the work done in the field of VRP, it is assumed that the fleet of vehicles is homogeneous. However, in the real application that motivated this study, there is a fleet of existing vehicles of different capacities which we refer to as vehicle types. Consequently, the type of vehicle to use becomes a decision of the problem. Deciding on the vehicle depends on the capital cost as well as the transportation costs incurred by a vehicle. This distinction to conventional VRP's adds another level of complexity to the problem.

As we stated previously, one of the main objectives of the problem is to minimize residents' inconvenience with the system. Most of the past work on problems where customers themselves are the commodities to be transported concentrated more on objective terms related to transportation costs, e.g. route length and capital costs, than on those related to customers' satisfaction with the system. In the school bus routing algorithms discussed above, students were assigned to bus stops that are within walking distance. Walking distance is a quantity judged to be the maximum distance a student can walk to reach the bus stop from which he/she is picked up and dropped at. In other words, the solution is constructed to have the minimum necessary number of bus stops to transport all students with the aim of minimizing the total route length and the number of routes while no student has to walk more than a fixed maximum walking distance. Hence users' satisfaction is not minimized explicitly, instead it is kept feasible with respect to some undesirable level, except in Bowerman et al. [3].

Unlike students, employees are to use the system for a longer time period. Moreover, job performance is found to depend a lot on employees psychology and social life. As a result many companies provide their employees with many services such as social activities and job-related discussions to enhance job performance. Consequently, we judge employees' satisfaction to be an important factor in the construction of the transportation service system.

Residents' satisfaction with the transportation system is a subjective measure. However, a resident is found to be more satisfied with the system when he/she is assigned to a bus stop nearest to his/her residence. Hence, we use resident walking distance as a measure of residents' satisfaction with the transportation service. Maximizing residents satisfaction or minimizing their inconvenience is then equivalent to minimizing the total walking distance or the maximum of individual walking distances.

In what follows we present the optimization criteria associated with the PBRP. Then we propose a comprehensive mathematical formulation of the problem.

3.3.1 Model Objectives

We define here the optimization criteria used to evaluate the transportation system.

1. Total Route Length. Since route length is directly proportional to the variable transportation costs and trips duration, total route length is minimized.
2. Total Walking Distance. From the above discussion, total walking distance is minimized to minimize residents' inconvenience.
3. Number of routes. This criterion reduces the number of vehicles in use in order to minimize vehicles capital costs.

3.3.2 Zero-one Integer Program Formulation

The mathematical formulation we propose is defined on the graphical representation of the general problem depicted in Figure 3.2.

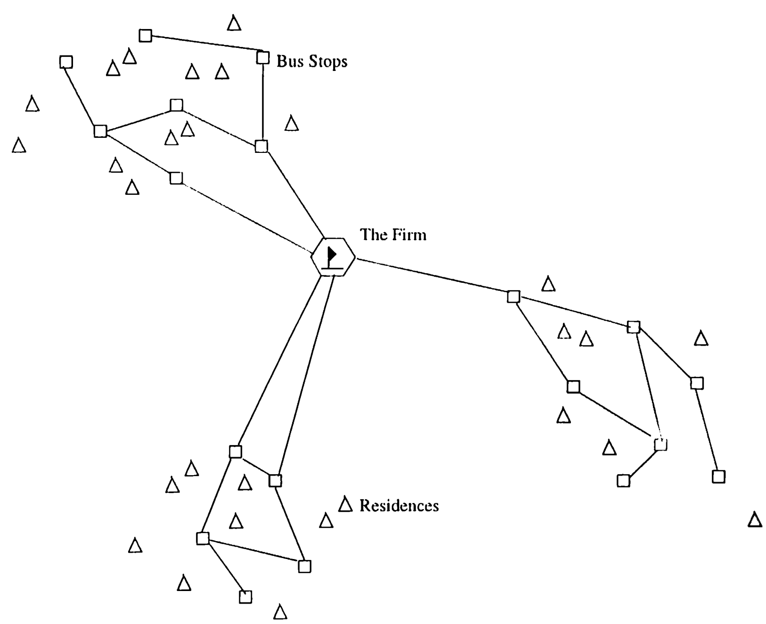

Figure 3.2: Network Representation of the PBRP

Indices

In order to define the PBRP mathematical model we use.

i = index for residences

j = index for bus stops

p, q = indices for routing points

k = index for bus types

l_k = index for vehicles of type k

Sets

We use the following sets in the model.

F = set of cardinality one representing the firm

B = set of all potential bus stop sites

$P = F \cup B$, all potential routing points

I = set of all residences

L = set of buses, each bus is of type k , where $k = 1, \ldots, K$

K is the number of bus types available. For each bus type $k = 1, \ldots, K$, there's a set of buses. We assume that there's no limit on the availability of buses of each type.

Parameters

The following parameters are used in the model.

b_k = capacity of bus of type k

d_{ij} = walking distance from residence $i \in I$ to routing point $j \in J$

d_{pq} = distance on street network between routing points $p, q \in P$

a_i = load of residence $i \in I$, if more than one employee live at the same address, we associate a weight to the residence. The weight is equal to the number of employees at that residence.

c_k = fixed cost of using a vehicle of type k

Model Variables

We define the following decision variables.

$$x_{ijl_k} = \begin{cases} 1 & \text{if residence } i \text{ is assigned to bus stop } j \text{ and serviced by vehicle } l_k \\ 0 & \text{otherwise,} \end{cases}$$

$$y_{pql_k} = \begin{cases} 1 & \text{if routing point } p \text{ precedes routing point } q \text{ on route of vehicle } l_k \\ 0 & \text{otherwise,} \end{cases}$$

$$z_{l_k} = \begin{cases} 1 & \text{if vehicle } l_k \text{ is used} \\ 0 & \text{otherwise.} \end{cases}$$

Objective Functions

$$\begin{aligned}
f_1 &= \sum_{i \in I, j \in B, l_k \in L} a_i d_{ij} x_{ijl_k} && \text{Total Resident Walking distance} \\
f_2 &= \sum_{p,q \in P, l_k \in L} d_{pq} y_{pql_k} && \text{Total Bus Route Length} \\
f_3 &= \sum_{l_k \in L} c_k z_{l_k} && \text{Total Bus Fixed Cost} \\
f_4 &= \sum_{l_k \in L} \left(\sum_{i \in I, j \in B} a_i x_{ijl_k} - b_k z_{l_k} \right)^2 && \text{Load Balancing}
\end{aligned}$$

We propose the fourth objective term to load-balance the buses used. The objective function f_4 is the sum of squares of the excess capacity in the used buses. Minimizing f_4 leads to comparable levels of unused capacity for each used bus. The objective function f_4 takes into account the heterogeneity of the fleet of buses.

The Model

$$
\begin{array}{llll}
\text{minimize} & (f_1, f_2, f_3, f_4) & & (1)\\
\text{subject to} & \sum_{l_k \in L, j \in B} x_{ijl_k} = 1 & i \in I, & (2)\\
& \sum_{i \in I, j \in B} a_i x_{ijl_k} \leq b_k & l_k \in L, & (3)\\
& z_{l_k} \geq x_{ijl_k} & i \in I, j \in B, l_k \in L, & (4)\\
& \sum_{p \in P} y_{pql_k} \geq x_{ijl_k} & i \in I, j \in B, l_k \in L, q \in P, q \equiv j, & (5)\\
& \sum_{p \in P} y_{qpl_k} \geq x_{ijl_k} & i \in I, j \in B, l_k \in L, q \in P, q \equiv j, & (6)\\
& \sum_{p \in B} y_{pql_k} = z_{l_k} & l_k \in L, q \in F, & (7)\\
& \sum_{p \in B} y_{qpl_k} = z_{l_k} & l_k \in L, q \in F, & (8)\\
& \sum_{(p,q) \in (S,S)} y_{pql_k} - n^2 u_{sl_k} \leq |S| - 1 & l_k \in L, S \subset B, & (9)\\
& \sum_{(p,q) \in (S,\overline{S})} y_{pql_k} + v_{sl_k} \geq 1 & l_k \in L, S \subset B, & (10)\\
& u_{sl_k} + v_{sl_k} \leq 1 & l_k \in L, S \subset B, & (11)\\
& u_{sl_k}, v_{sl_k} \in \{0,1\} & l_k \in L, S \subset B, & (12)\\
& x_{ijl_k} \in \{0,1\} & i \in I, j \in B, l_k \in L, & (13)\\
& y_{pql_k} \in \{0,1\} & p, q \in P, l_k \in L, & (14)\\
& z_{l_k} \in \{0,1\} & l_k \in L. & (15)
\end{array}
$$

Constraints (2) ensure that every resident is assigned to exactly one bus stop and to only one bus. Constraints (3) guarantee that bus capacities are not exceeded. Constraints (4) defines the bus status, if it has any residents assigned to it or no. Constraints (5) and (6) ensure that if a resident is assigned to a bus stop and a bus, then the bus visits the bus stop in its tour. Constraints (7) and (8) ensure that all bus routes initiate and end at the firm. Constraints (9) through (12) eliminate disconnected sub-tours but allow cycles. This type of sub-tour elimination constraints are used in the rural postman problem mathematical formulations [1]. Constraints (13) through (15) are the integrality restrictions on the decision variables. The objective function (1) is a function of the four optimizing criteria defined previously. One advantage of the PBRP formulation is its generality. If $K = 1$, the formulation becomes valid for problems with homogeneous fleet of vehicles. The formulation is also valid if there are many firms or generally depots. If the firms are serviced independently, increase the size of the set F of firms which currently has cardinality one to include all firms. The set of routing points P is updated accordingly.

The formulation presented above cannot be used directly to solve the PBRP for three reasons. First, the problem formulation cannot be solved directly in a reasonable amount of time since the LRP is a NP-hard problem [19]. Second, the formulation is multi-objective with one of the objective terms being nonlinear. Third, the formulation generates a model with a very large number of variables and constraints. For example, for a PBRP with 20 potential bus stop sites, 100 residences, 2 bus types, and 4 buses, the number of variables would be over 12200 and the number of constraints over 33870. Hence, the PBRP cannot be solved in a reasonable amount of time using available optimization tools. Therefore, we develop a heuristic approach in order to generate solutions to the PBRP. The heuristic we propose is a two-phase approach of the type cluster-first route second. In the clustering part, we develop a heuristic branch and bound algorithm for the generalized assignment problem to generate clusters. In the routing part, we use an algorithm that finds the optimal routes on each cluster by taking the allocation, location and routing decisions simultaneously. The heuristic methodology proposed is presented in the next chapter.

Chapter 4

A Heuristic Solution Method for the Multi-objective Model

The PBRP is a large-scale problem which cannot be solved as a whole with the available optimization tools. Consequently, we develop a heuristic solution method of the cluster-first route-second type in order to reduce the PBRP into a set of independent smaller problems that can be solved in reasonable time while explicitly considering the multi-objective nature of the problem and the heterogeneity of the fleet of buses. The solution heuristic we devise to solve the multi-objective model exploits the special structure of the problem and considers explicitly its objectives. The heuristic is performed in two phases. First, a clustering heuristic algorithm is used to assign residents to bus stops and group bus stops into clusters each of which is serviced by one bus. The main objective considered in the clustering phase is minimizing the number of buses in use which is equivalent to minimizing the number of clusters generated. The objective of minimizing the total walking distance is dealt with by keeping the total walking distance at the lowest feasible level. Moreover, the clusters are formed with cost functions devised to approximate the length of routes resulting once the cluster is generated. Hence, the objective of minimizing the total route length is also considered implicitly. Second, bus routes are constructed in each cluster. In the routing phase, the allocation, location and routing decisions are taken simultaneously in each cluster with the aim of minimizing the total residents walking distance and the total bus route length.

4.1 The Clustering Algorithm

The problem we are dealing with is a large-scale problem defined on a vast geographical residential area. Consequently, we first devise a clustering algorithm with the purpose of reducing the whole problem into a set of smaller problems. This is achieved through grouping residents into clusters each of which can be serviced by a unique bus route. The algorithm can be described as follows: assign each employee to the bus stop nearest to his residence, group the bus stops into clusters each to be serviced by one vehicle. To generate clusters, we define a $0-1$ integer programming model and develop a heuristic method to generate solutions to this model.

Not all objectives of the PBRP can be dealt with during the clustering phase since bus routes are not yet defined. The two goals explicitly considered during the clustering algorithm are:

- minimizing the number of routes or vehicles which is equivalent to minimizing the number of routes or vehicles since in each cluster a unique route is formed and it is traversed by one vehicle.
- keeping residents' satisfaction to a maximal level by using a minimum distance assignment criterion.

The objective of minimizing the number of routes is independent of the total route length and total walking distance objectives. Meanwhile, it depends on the number of residents to be serviced and on the available bus capacities. Moreover, the decision on which type of buses to use is interrelated with the decision on the number of buses. Hence, it is in the clustering part that the types and number of buses is fixed. As a consequence of the fact that the allocation-location decisions and the routing decision are interdependent, we implicitly consider the total route length objective in the second step of the clustering algorithm. We define appropriate cost measures in the clustering model to approximate the length of the route in the cluster once generated.

4.1.1 Assign Residents to Bus Stops

In the first step, each employee is assigned to a bus stop based on a minimum-distance assignment criterion. The distance measure used here is the Euclidean distance between residences and bus stops on the network and termed as the employee (or, resident) walking distance. The procedure is to assign the next unassigned resident to the nearest bus stop to his residence. If there is more than one potential bus stop for assignment, then the resident is assigned to a bus stop with nonzero load. If there is more than one nonzero-loaded bus stop, the resident is assigned to the minimum-loaded bus stop in order to load-balance bus stops. The load of a bus stop is defined as the number of residents assigned to the bus stop thus far. We use the terms load and weight of a bus stop interchangeably. Figure 4.1 gives a flow chart of the assignment procedure used at this level. We use a minimum distance assignment scheme to keep residents inconvenience at its lowest level since we judge it to be a dominant objective in the PBRP.

We note that if the route length objective is judged to be more important than residents' inconvenience, then other schemes can be devised in order to obtain the minimum necessary number of stops to use. Such a scheme may be: assign the next unassigned resident to a nonzero-loaded bus stop such that a maximum walking distance is not exceeded else assign the next unassigned resident to the bus stop with the highest number of unassigned residents within walking distance. This scheme is used in Chapleau et al. [4].

At the end of the first step, a set of bus stops with nonzero loads is obtained. These bus stops are grouped into clusters in the second part of the clustering algorithm.

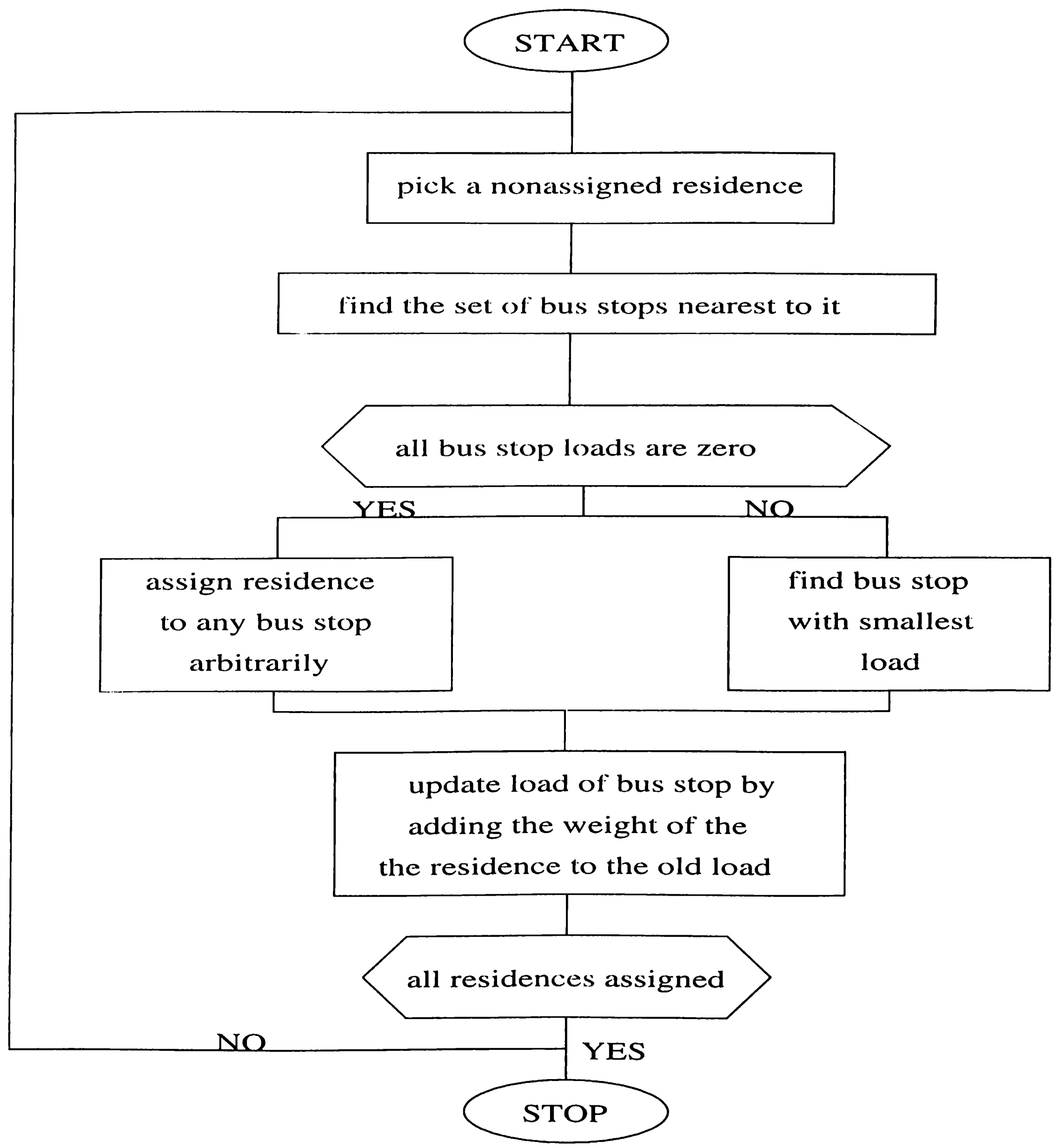

Figure 4.1: Minimum Distance Assignment Flow Chart

4.1.2 Generate Clusters

At this stage, bus stops, each with a known weight, are grouped into clusters such that each cluster is serviced by one vehicle. In other words, each bus stop is assigned to exactly one cluster, and the total weight assigned to a used cluster does not exceed its capacity. Referring to Figure 4.2, clusters can be

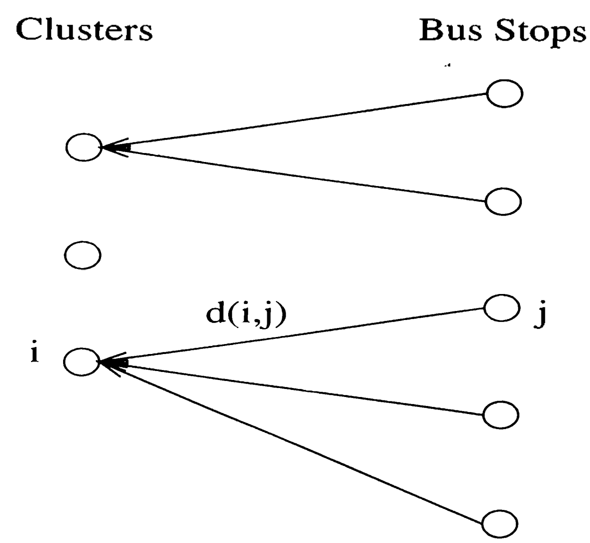

Figure 4.2: Network Representation of the Clustering Problem

considered as facilities that provide service to bus stops. Bus stop weights represent demand requirements, and clusters are the facilities to be located to satisfy total demand requirements. We take bus stop locations as the set of potential location sites for clusters. Moreover, if there are K different vehicle capacities then we consider K potential clusters each with a different capacity at each bus stop location. In what follows, bus types and cluster types are equivalent and are used interchangeably. We now present a formulation of the clustering model. We then describe the similarity of the clustering model to the capacitated facility location problem with choice of facility type (CFLP). Based on this similarity we transform the clustering model into a generalized assignment problem (GAP) and propose a heuristic algorithm to solve the resulting GAP.

Mathematical Formulation for Clustering

The formulation described below is a $0-1$ integer program model for generating clusters of bus stops.

$$
\begin{array}{llll}
\text{minimize} & \sum_{i\in I}\sum_{j\in J} d(i,j)\,x_{ij} + \sum_{i\in I} c_i y_i & & (1)\\
\text{subject to} & \sum_{i\in I} x_{ij} = 1 & j\in J, & (2)\\
& \sum_{j\in J} a_{ij}x_{ij} \le b_i & i\in I, & (3)\\
& \sum_{k=1}^{K} y_{i+k} \le 1 & i\in J, & (4)\\
& x_{ij}\in\{0,1\} & i\in I, j\in J, & (5)\\
& y_i\in\{0,1\} & i\in I. & (6)
\end{array}
$$

Here $I = \{1,\ldots,M\}$ is the set of potential clusters, $J = \{1,\ldots,N\}$ is the set of bus stops to be grouped. Each bus stop location represents a potential site for a cluster. Hence, the number of potential clusters is the number of bus stops multiplied by the number of cluster types: $M = N \times K$, where cluster at site p and of type q is cluster $i = (p-1)\times K + q$. The function $d(i,j)$ is a measure of the cost incurred when bus stop j is assigned to cluster i, c_i is a fixed cost term associated with generating cluster i,a_{ij} is the load of bus stop j, and b_i is the capacity of cluster i. Variables x_{ij} and y_i are defined as follows.

$$
x_{ij} = \begin{cases} 1 & \text{if bus stop } j\in J \text{ is assigned to cluster } i\in I\\ 0 & \text{otherwise,}\end{cases}
$$

$$
y_i = \begin{cases} 1 & \text{if cluster } i\in I \text{ is used}\\ 0 & \text{otherwise.}\end{cases}
$$

The objective function (1) measures the total cost of assignments. The first objective term minimizes the total incremental cost of a cluster incurred by including some set of bus stops into the cluster. The second term minimizes the number of clusters generated. Constraints (2) ensure that the requirement of each bus stop is totally fulfilled. Constraints (3) guarantee that vehicle capacities are not violated. Constraints (4) guarantee that at most one cluster is located at each site, or equivalently at most one type of each cluster is used. Constraints (5) and (6) are the integrality restrictions on variables x_{ij} and y_i.

The clustering model we describe above and the capacitated facility location model with choice of facility type are two similar mathematical models. The CFLP with choice of facility type is the problem of locating a number of facilities which have to service a set of customers, at minimum cost, where each customer has an associated demand and there are constraints on the total demand that can be met from a facility [7]. Moreover, there's the possibility of choosing the type or size of the facility as well as the location sites themselves. Similar to the clustering problem we deal with, the CFLP associates a cost to the assignment of demand points to facilities and a fixed or set up cost to establishing a facility. However, in the CFLP, fractional assignments are allowed. Hence, the demand of a customer can be satisfied from different facilities. This represents the main distinction of the CFLP model to the clustering model because we require that every bus stop is assigned solely to a unique cluster. This requirement causes the clusters generated to be disjoint. Disjoint clusters are preferred for two reasons. First, each cluster is to be serviced by one bus route. Hence, if there is a bus stop belonging to more than one cluster, the bus stop is serviced by more than one bus so the total route length is expected to be higher. Second, from a managerial point of view, disjoint clusters are preferred since they result in disjoint routes. The capacitated facility location problem with choice of facility type can be used to generate clusters. However, the clusters generated may not be disjoint. As a result, exchange algorithms must be used to modify the clusters generated until disjoint clusters are found. On the other hand, Ross and Soland [21] show that the CFLP with choice of facility type and a number of other facility location problems can be formulated and solved as generalized assignment problems. We use the ideas of Ross and Soland to transform the $0-1$ clustering model into a generalized assignment problem.

Transform Clustering Model into GAP

We first define what is termed a generalized assignment problem. As defined by Ross and Soland [22] the generalized assignment problem minimizes the cost of assigning n tasks to a subset of m agents. Each task is assigned to a unique agent, while each agent is limited by some resource availability. A mathematical formulation of the GAP is

$$\begin{array}{llll} \text{minimize} & \sum\limits_{i\in I}\sum\limits_{j\in J} c_{ij}x_{ij} & & (1) \\ \text{subject to} & \sum\limits_{i\in I} x_{ij} = 1 & j \in J, & (2) \\ & \sum\limits_{j\in J} a_{ij}x_{ij} \leq b_i y_i & i \in I, & (3) \\ & x_{ij} \in \{0,1\} & i \in I, j \in J. & (4) \end{array}$$

In the context of clustering, $I = \{1, \ldots, M\}$ represents the set of potential clusters, $J = \{1, \ldots, N\}$ is the set of bus stops,c_{ij} is the cost of assignment of bus stop j to cluster i, a_{ij} is the load of bus stop $j \in J$, b_i is the capacity of cluster $i \in I$, and variable x_{ij} takes value one if bus stop j is assigned to cluster i. The objective function term (1) minimizes the cost of assignments of bus stops to clusters. Constraints (2) guarantee that every bus stop is assigned to exactly one cluster. Constraints (3) are the capacity constraints. Constraints (4) are the integrality constraints on variables x_{ij}.

The GAP [11] is NP-hard since the NP-complete 2-partition problem is reducible to the GAP. Given n real numbers $A_1, \ldots, A_n$, the 2-partition problem asks if there is a set $S \subseteq \{1, \ldots, n\}$ such that $\sum_{j\in S} A_j = \sum_{j\notin S} A_j$. This problem is equivalent to a GAP with $m = 2$, $a_{1j} = a_{2j} = A_j$ for all j, $b_1 = b_2 = \sum_{j\in N} A_j/2$ and c_{ij} arbitrary.

In what follows we explain the essence of the transformation from the clustering model to the generalized assignment problem. Then, we illustrate it based on a hypothetical example.

In the new problem, the x_{ij} variables are defined as above for $i = 1, \ldots, M$, $j = 1, \ldots, N$. To specify whether or not each of the M potential clusters is to be used, M "bus stops" to be assigned are added. It is also necessary to impose the use of at most one cluster at each potential site, this is achieved by adding N "potential clusters". The dimensions of the new problem are $m = M + N$ and $n = N + M$. For $i \leq M$ and $j > N$ variables x_{ij} are defined by

$$x_{ij} = \begin{cases} 1 & \text{if cluster } i = j - N \text{ is not used} \\ 0 & \text{otherwise.} \end{cases}$$

For $i > M$, $j > N$ variables x_{ij} are defined by

$$x_{ij} = \begin{cases} 1 & \text{if cluster at site } \hat{\imath} = i - M \text{ of type } \kappa = j - M - (\hat{\imath} - 1) \times K \text{ is used} \\ 0 & \text{otherwise.} \end{cases}$$

Cost and coefficient matrices terms, and RHS terms corresponding to the added bus stops and clusters are defined as follows.

$$\begin{aligned}
&a_{ij} = 0,\ d(i,j) = b_i, && i \leq M,\ j > N \text{ and } i = j, && (1)\\
&a_{ij} = 1,\ d(i,j) = F_i, && i > M,\ j = N + (i - M) + k \text{ and } k = 1, \ldots, K - 1, && (2)\\
&b_i = 1, && i > M, && (3)\\
&a_{ij} = d(i,j) = \infty && \text{otherwise.} && (4)
\end{aligned}$$

In (1) we associate a weight equal to the capacity of the cluster for each additional "bus stop" so that no stops are assigned to a cluster if it is not used. This assignment has cost 0. (2) associates a fixed cost to establishing a cluster and a weight 1 so that if one cluster of some type is opened at a site no other cluster is opened at the same site. (3) ensures that at most one cluster is opened at each potential site. In (4), all remaining terms are set to infinity to make corresponding assignments infeasible.

Example: Consider the following hypothetical example. Suppose there are three bus stops to be clustered with two cluster types. Then $N = 3, K = 2, M = 6, m = n = 9$. Matrices C and A below give the cost and coefficient matrices of the clustering model respectively. The right hand side vector is denoted by b.

$$C^T = \begin{bmatrix} 0 & 0 & 2 & 2 & 3 & 3 \\ 2 & 2 & 0 & 0 & 4 & 4 \\ 3 & 3 & 4 & 4 & 0 & 0 \end{bmatrix}$$

$$A^T = \begin{bmatrix} 5 & 5 & 5 & 5 & 5 & 5 \\ 10 & 10 & 10 & 10 & 10 & 10 \\ 3 & 3 & 3 & 3 & 3 & 3 \end{bmatrix} \quad b^T = \begin{bmatrix} 10 & 15 & 10 & 15 & 10 & 15 \end{bmatrix}.$$

Suppose a fixed cost of 20 units is incurred when a cluster is used. The resulting GAP can be defined by the following tableaus which give the coefficient matrix A', the cost matrix C', and the right hand side vector b'.

$$C' = \begin{bmatrix} 0 & 2 & 3 & 0 & \infty & \infty & \infty & \infty & \infty \\ 0 & 2 & 3 & \infty & 0 & \infty & \infty & \infty & \infty \\ 2 & 0 & 4 & \infty & \infty & 0 & \infty & \infty & \infty \\ 2 & 0 & 4 & \infty & \infty & \infty & 0 & \infty & \infty \\ 3 & 4 & 0 & \infty & \infty & \infty & \infty & 0 & \infty \\ 3 & 4 & 0 & \infty & \infty & \infty & \infty & \infty & 0 \\ \infty & \infty & \infty & 20 & 20 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 20 & 20 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 20 & 20 \end{bmatrix}$$

$$A' = \begin{bmatrix} 5 & 10 & 3 & 10 & \infty & \infty & \infty & \infty & \infty \\ 5 & 10 & 3 & \infty & 15 & \infty & \infty & \infty & \infty \\ 5 & 10 & 3 & \infty & \infty & 10 & \infty & \infty & \infty \\ 5 & 10 & 3 & \infty & \infty & \infty & 15 & \infty & \infty \\ 5 & 10 & 3 & \infty & \infty & \infty & \infty & 10 & \infty \\ 5 & 10 & 3 & \infty & \infty & \infty & \infty & \infty & 15 \\ \infty & \infty & \infty & 1 & 1 & \infty & \infty & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & 1 & 1 & \infty & \infty \\ \infty & \infty & \infty & \infty & \infty & \infty & \infty & 1 & 1 \end{bmatrix} \quad b' = \begin{bmatrix} 10 \\ 15 \\ 10 \\ 15 \\ 10 \\ 15 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

It is possible to add further constraints to the problem: lower bounds on the capacity of clusters, upper limit on the number of clusters to be used in total. Refer to Ross and Soland [22] for a more detailed description of the transformation.

Even though the transformation causes a three-fold increase in the dimension of the matrices defining the problem, the resulting matrices reveal a special structure and the GAP can be solved by just storing the information provided in the defining matrices of the clustering model.

4.1.3 Branch and Bound Heuristic Algorithm for the GAP

We now propose a branch and bound heuristic algorithm to solve the GAP described above. The algorithm exploits the special structure of the matrices defining the GAP. The proposed algorithm is based on the ideas of the branch and bound algorithm that was proposed by Garfinkel and Nemhauser [14] for solving the set partitioning and covering problems. The set covering problem can be defined as follows [5]. Given a set of objects $J = \{j_1, \ldots, j_n\}$ and a family $\mathcal{S} = \{S_1, \ldots, S_m\}$ of sets $S_i \subset J$, the set covering problem is to find a minimal cost sub-family $\mathcal{S}' = \{S_{i_1}, \ldots, S_{i_k}\}$ of $\mathcal{S}$ such that $\cup_{l=1}^{k} S_{i_l} = R$. Moreover if we require $S_{j_h} \cap S_{j_l} = \emptyset$, for all $h, l \in \{1, \ldots, k\}$ then the resulting problem is the set partitioning problem. The sets S_i are called covering sets and the sub-family $\mathcal{S}'$ is called a set covering. For a given family $\mathcal{S}$, the tree search method of Garfinkel and Nemhauser solves the set covering/partitioning problem defined on J optimally. The branch and bound algorithm for the set covering problem starts by forming for each object $j_l \in J$ a list $L_l = \{S_i : j_l \in S_i\}$. The algorithm then searches the lists sequentially, starting from the list corresponding to the first object and picking at each step the next list for which the corresponding object is not covered yet, until a set covering is found. Then through back search, all set coverings are checked until the first list is exhausted, then the optimal solution is found. The algorithm for set partitioning is essentially the same except in the definition of lists L_l. In the context of the clustering problem, the bus stops are the objects to be covered and the clusters are the covering sets. We want to find a set partitioning of the set of bus stops in order to generate disjoint clusters. However, the branch and bound algorithm described above cannot be used directly to solve the clustering problem because the clusters are not yet defined. As a result, we modify the branch and bound algorithm so as to match with the characteristics of the problem. First, to generate the covering sets we solve a set of knapsack problems at each iteration of the algorithm. Consequently, the covering sets and the lists are formed during the course of the algorithm. In other words, this is equivalent to generating new columns of the coefficient matrix at each iteration. The solutions of knapsack problems correspond to the columns generated. Moreover, whenever a covering set (or, cluster) is added to the set covering (or, clusters to be used), the set of objects (or, unassigned bus stops) and the set of covering sets (or, potential clusters) is updated. The branch and bound algorithm is a depth-first breadth-second algorithm. In what follows we give a detailed description of the algorithm.

Notation

The following notation is used through the algorithm.

$\hat{B}$: a family of clusters to which all bus stops are assigned exactly once. This represents the best solution found so far.

$\hat{Z}$: cost associated with solution $\hat{B}$. This gives an upper bound on the optimal objective value.

B: a subset of clusters that form a partial solution at the current stage of the algorithm with associated cost Z.

E: set of bus stops assigned to some cluster in B at the current stage of the algorithm.

$\overline{E} = J - E$: set of bus stops not assigned yet to any cluster in B.

$\hat{I}$: the set of potential clusters at the current stage of the algorithm.

S_i : the set of bus stops that are assigned to cluster i once it is opened. Call it assignment set. Sets S_i are found by solving knapsack problems defined on the set of unassigned bus stops $\overline{E}$ for each $i \in \hat{I}$. The cost associated with S_i is $C(S_i)$.

$L_j = \{i \in \hat{I} : j \in S_i\}$, list of potential clusters i to which bus stop j can be assigned. Call it list of potential assignments.

At each stage of the algorithm, the following steps are performed. First, solve a set of knapsack problems. Each knapsack corresponds to a potential cluster $i \in \hat{I}$

and defined on the set of unassigned bus stops $\overline{E}$. The solution of knapsack problem i, S_i, gives the maximum feasible number of assignments to cluster i. Then, relative to some $j \in \overline{E}$, which we call the seed for the next set of assignments, form list L_j which includes all sets S_i that contain (or, cover) j. Second, find a feasible S_i from list L_j and perform necessary updates in the partial solution B, the corresponding objective value Z, and sets E and $\overline{E}$. We say that S_i is cost feasible or simply feasible if $Z + C(S_i) \prec \hat{Z}$ is satisfied, otherwise S_i is said to be cost infeasible or simply infeasible. Then, check if all bus stops are assigned then an improved solution is found, record it and continue to inspect other possible assignments. Otherwise, continue the algorithm by solving a new set of knapsack problems. The lists formed during the course of the algorithm are searched sequentially. The algorithm stops when the first list formed is exhausted.

The algorithm can be described as follows.

B & B Algorithm

Initialize step 1: Initialize $B = \emptyset$, $E = \emptyset$, $\overline{E} = J$, $Z = 0$, and $\hat{Z} = \infty$. No assignments yet.

Augment step 2:

1. Select a $j \in \overline{E}$. Call it j.
2. Define $\hat{I}$ as the set of potential clusters. $\left|\hat{I}\right| = \left|\overline{E}\right| \times K$.
3. For each $i \in \hat{I}$ solve a knapsack problem defined on all $j \in \overline{E}$.
4. Construct sets $S_i = \{j \in J : x_{ij} = 1\}$.
5. Form list $L_j = \{i \in \hat{I} : j \in S_i\}$. Sort L_j in ascending order of costs, and go to step 3.

step 3: Find i in L_j such that $Z + C(S_i) \prec \hat{Z}$ and S_i not inspected yet. Call it $\hat{\imath}$.

Test: If such an S_i exists then go to step4. Else list L_j has been exhausted then go to step 5.

Test for New Solution step 4: Perform the following updates: $B = B \cup \{S_i\}$, $E = E \cup S_i$, $\overline{E} = \overline{E} - S_i$, $Z = Z + C(S_i)$.

Test: If $\overline{E} = \emptyset$ a better solution is found, set $\hat{B} = B$, $\hat{Z} = Z$, and go to step 5. Else go to step 2.

Backtrack step 5: B can't lead to a better solution.

Test: If $B = \emptyset$ then list L_1 has been exhausted, STOP. Else remove $\{S_i\}$ from B, S_i from E, and add S_i to $\overline{E}$. Check if L_j is exhausted then go to step 5. Else go to step 3 .

We note that in the problem we solve, the list L_i contains at least K S_j's, each associated with one knapsack defined on one of the K potential clusters at site i. Thus, the algorithm always ends with a feasible solution. However the algorithm does not guarantee to end up with the optimal solution for two reasons. First, the algorithm performs partial search since the algorithm is performed for only one bus stop as the initial bus stop. However, it can be extended to perform total search by performing the algorithm each time for the next bus stop as the one that starts the algorithm, starting from the bus stop with the smallest index. Second, the solutions generated from knapsack problems at some stage of the algorithm affect the following steps. Moreover, to solve knapsack problems we use a greedy heuristic algorithm. The greedy algorithm gives the possibility of using different distance measures in evaluating the assignments. The algorithm is described in what follows. Then we discuss the criteria for seed selection we use and define the distance measures and cost functions we propose. We also discuss the complexity of the algorithm and its worst and average-case behavior.

Greedy Heuristic Algorithm for Solving the Knapsack Problem. The knapsack problems generated during the course of the branch and bound algorithm are solved to form sets of assignments associated with each of the potential clusters. When a cluster is established, all assignments found by solving the corresponding knapsack are made. The objective function value of the knapsack problem is then taken as the incremental cost part of the objective function of the clustering model. A certain distance measure is used in solving knapsack problems. With the different distance measures we propose we find it useful to use a greedy heuristic to solve the knapsack problems since it allows for cost updates at every iteration. Moreover, to have compact clusters bus stops nearest to the cluster site with respect to the distance measure in use have the priority for assignment even though farther bus stops may lead to higher number of assignments. The steps of the greedy algorithm are described below.

1. find the unassigned bus stop with minimum distance to the cluster,
2. check if the bus stop is feasible: the bus stop weight less than the remaining unused capacity in the cluster. If the bus stop is feasible then go to step 3. Else go to step 4,
3. update the knapsack solution set by adding the bus stop and remove it from the list of unassigned bus stops. Update the available capacity by subtracting the assigned bus stop weight. Update unassigned bus stop assignment costs according to the distance measure in use. Go to step 1,
4. if the list of unassigned bus stops is exhausted, then the knapsack solution is found. Stop the algorithm. Else, remove the last inspected bus stop from list of unassigned bus stops. Go to step 1.

Criteria for Seed Selection. At Step 2 of the branch and bound algorithm, a seed is selected among the set of unassigned bus stops to form a list of potential assignments from which feasible assignment sets are inspected sequentially in subsequent stages of the algorithm. The seed selected at the current stage of the algorithm affects the assignments made, the cost attributed and the remaining set of unassigned bus stops. Consequently, the quality of solution found is affected with respect to the total cost and cluster forms which in turn affect the routing decision. In addition, the choice of the seed affects the behavior of algorithm in terms of number of tree branches searched and running time. The following are three criteria for seed selection. Let the seed be:

- the unassigned bus stop with smallest index,
- the unassigned bus stop with maximum weight, or

- the unassigned bus stop with minimum number of assignment sets in the corresponding list of potential assignments.

The first criterion is random. The second one bases the selection on the characteristics of the candidate bus stops. The third criterion is chosen to fit with the branch and bound algorithm. A bus stop with a big load is a determinant factor in forming clusters because it cannot be fitted easily into generated clusters as in the case of bus stops with smaller loads. Hence, selecting the unassigned bus stop with maximum load as the seed for the next search increases the quality of the solution obtained since the assignments inspected are more likely to occur in the optimal solution. The third criterion results from the fact that the search in the algorithm is depth-first breadth-second. A new portion of the search tree is generated for each feasible set in the assignment lists. As a consequence, it is better to process smaller lists first in order to decrease the size of the search tree. Moreover, at the late branches cost infeasibility becomes dominant in eliminating assignment sets for inspection. Hence, keeping large lists to the later stages of the algorithm discards large portions of the potential tree search.

Definition of costs. There are two types of costs contributing to the objective function value of the GAP: Incremental cost comes from individual assignments of bus stops to clusters. Fixed cost is incurred when a cluster is set up. We introduce the definition of the incremental cost term represented in the objective function of the GAP by $d(i,j)$. The cost measure $d(i,j)$ is defined as a measure of distance between cluster i and bus stop j. In what follows we introduce four different distance measures used to generate different cluster shapes. Referring to Figure 4.3, we denote by $r(l)$ the reference point of cluster l, it is the first bus stop assigned to cluster l and so it is the bus stop located at the same site as the cluster itself. We also denote by $\overline{pq}$ the Euclidean distance between points p and q on the network. A point can be a bus stop or a cluster.

1. $d(p,q) = \overline{pq}$, here the distance measure is the Euclidean distance between cluster p and bus stop q. The incremental cost term then reduces to the total sum of the Euclidean distance between bus stop j and cluster i over all assignments; see Figure 4.3.a.

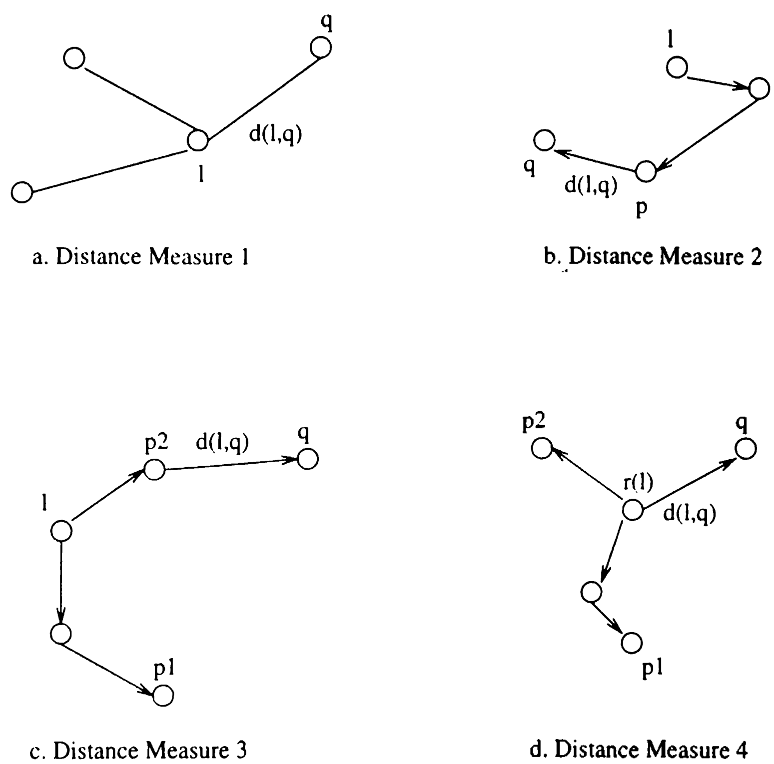

Figure 4.3: Four Different Distance Measures for Clustering

2. $d(l,q) = \overline{pq}$, here the distance measure between cluster l and bus stop q gives the Euclidean distance between bus stop q and p, the last bus stop assigned to cluster l. With this distance measure the cluster takes the form of a route that visits the assigned points to the cluster however it grows in only one direction; see Figure 4.3.*b*.

3. $d(l,q) = \min_{\forall p_i} \overline{p_i q}$, where p_i's are the bus stops assigned to cluster l which do not precede any of the other bus stops assigned to cluster l. The cluster l formed in this case is a route built by assigning the nearest feasible bus stop to its endpoints starting with $r(l)$. The route grows in two directions only; see Figure 4.3.*c*.

4. $d(l,q) = \min\{\min_{\forall p_i} \overline{p_i q}\ , \overline{r(l)q}\}$, here the next bus stop assigned to cluster l is the bus stop that is nearest to any of the endpoints or to the reference node of the cluster l. This distance measure approximates the detour in the route induced by the next assignment. The cluster configuration shows multiple route segments that intersect at a single point which is the reference point of the cluster; see Figure 4.3.*d*.

Algorithm Complexity. In the worst case, one bus stop is assigned whenever the algorithm performs Step 2. Hence each time all bus stops are assigned we have constructed at most n lists to be inspected by backtracking. When constructing the i^{th} list, we solve at most $K \times (n - i + 1)$ knapsack problems which means we have at most $K \times (n - i + 1)$ assignment sets to be inspected. Each time we go back to the previous list we explore a new portion of the search tree. If we denote by $\Delta(n)$ the complexity of the algorithm we use to solve one knapsack problem in terms of n the total number of bus stops to be assigned, then the worst case behavior of the algorithm is in the order of $n!K^n\Delta(n)$.

The average complexity of the algorithm depends on three factors: cost terms $d(i,j)$ and F_i, load of bus stops, and cluster capacities. First, costs affect the behavior of the algorithm through cost feasibility. An important portion of the search tree can be discarded when the inspected assignment sets are infeasible in terms of cost. Second, with higher cluster capacities and lower bus stop loads more assignments are made. Consequently, the lists constructed in the algorithm are fewer and shorter. This reduces the search tree both in depth and in breadth. Because of the special structure of the problem we consider, the three stated factors significantly reduce the complexity of the algorithm which enables us to solve relatively large problems.

4.2 The Bus Stop Routing Algorithm

The solution generated by the branch and bound algorithm used to solve the clustering formulation results in each resident assigned to one bus stop which is in turn assigned to one cluster. Consequently, each resident is assigned to only one cluster. Hence, at the end of the clustering stage, we have residents grouped into clusters each of which is serviced by one bus. Then the problem reduces to a number of independent routing problems. Each routing problem is defined on a different cluster and all potential bus stop sites in that cluster. We also add the company as bus stop number 1 where the route starts and ends, and to which no resident assignments are made. In each cluster, the routing problem is to locate a subset of bus stops among the potential bus stop sites, to assign residents to bus stops, and to form bus routes on these bus stops. To solve the routing problems we develop a bus stop routing algorithm that generates one route at a time.

The routing procedure is based on repeating assigning residents to bus stops and building routes on these bus stops until no more improvement in the objective function is obtained. The two optimization criteria considered in the routing phase are the total route length and the total walking distance. Both criteria are measured in the same unit of distance, however they are not proportional. Consequently, we use weighting [26] to form a single objective function. The weights associated with each of the objective terms highlight the importance accorded to one of the terms over the other. The algorithm can be described by the following three steps:

1. assign residences to bus stops according to the minimum walking distance assignment criterion. Calculate the total assignment cost which is equivalent to the total resident walking distance.

2. solve a traveling salesman problem over the set of bus stops with non-empty loads. Find the cost of the TSP tour which is equivalent to the total route length,

3. Improvement Scheme: remove a subset of bus stops , reassign residences, where each is assigned to the nearest available bus stop. Check if all residents are within walking distance from their assigned stops, then record the weighted cost and continue the improvement scheme. Else, the last removal is infeasible , restore the bus stop and its assignments and continue the improvement scheme.

Step one is similar to the first part of the clustering algorithm where each residence is assigned to a bus stop according to the minimum walking distance criterion we described earlier. Consequently, by recording the assignments of residences to bus stops done in the first step of the clustering algorithm, Step 1 of the routing algorithm can be omitted.

To construct a route on a set of bus stops, we use Eastman's algorithm for the traveling salesman problem which is a branch and bound algorithm that finds the optimal TSP tour. The Eastman-algorithm solves the easier assignment problem that allows sub-tours and then systematically forbids sub-tours until finally a tour is obtained that is optimal. The algorithm finds the exact solution in a reasonable amount of time for small to moderate size traveling salesman problems [16]. The last step of the algorithm is described below in more details.

4.2.1 Routing Improvement Scheme

The improvement scheme attempts to decrease the total weighted objective of the routing phase. The objective is the weighted sum of the total walking distance and the total route length. The scheme iterates over the potential bus stop sites in a cluster: remove a subset of bus stops at a time, repeatedly perform assignments and routing on the bus stops left, and find the weighted objective value. Select the configuration that gives the minimum weighted objective value. The improvement scheme can be performed for subsets of bus stops of different cardinalities until no more improvement is obtained. In the current implementation of the algorithm, we perform the improvement scheme for subsets of cardinality one only. We now define the notation used through the algorithm and describe its steps.

Notation

For a cluster k defined by,

$S_k = \{j_1, j_2, \ldots, j_{n-1}\}$, Set of bus stops assigned to cluster k, found by the branch and bound algorithm.

$S = S_k \cup F = \{j_1, j_2, \ldots, j_n\}$, Updated set of bus stops with the firm as bus stop number 1 and $j_l = j_{l+1}$ for $l \in \{1, \ldots, n-1\}$.

$I_k = \{i_1, i_2, \ldots, i_m\} = I_{j_1} \cup I_{j2} \cup, \ldots, \cup I_{jn-1}$. Set of residences assigned to bus stops in S_k, it is the union of sets I_{j_h} which represent set of residences assigned to bus stop j_h in S_k. Each residence i has a weight w_i.

D: $(n-1)\times(n-1)$ matrix giving distances between every pair of bus stops in S.

ω: the maximum walking distance.

$w1, w2$: weights associated with assignment and routing total distances respectively.

Improvement Algorithm

Initialize Step 1: Initialize

1. $\hat{Z} = \infty$, best weighted objective found so far.
2. $l = 0$, step index.
3. $Z_{1j_h} = \sum_{i \in I_{j_h}} w_i d_{ij_h}$, resident walking distance for every bus stop $j_h \in S_k$.
4. $Z = \sum_{j_h \in S_k} Z_{1j_h}$, total resident walking distance for all bus stops $j_h \in S_k$.

Iterate Step 2: $l = l + 1$.

Test: If $l = 1$, then go to 3. Else, form distance matrix for TSP by removing l^{th} row and l^{th} column from distance matrix D. Go to step 3.

Step 3: Using Eastman's Algorithm, solve the TSP over the distance matrix defined in Step 2. Find the length of the TSP tour, call it Z_{2l}. Go to step 5.

Step 4: Assign each residence in I_{j_l} to the nearest bus stop in the set $S_k - \{j_l\}$.

Test: If every resident is assigned to a bus stop within the maximum walking distance ω, then calculate Z'_{j_l} which is the weighted sum of the distances from the newly assigned residences to their bus stops. The weight here corresponds to the number of residents to be serviced from the same residence. Go to Step 2. Else, the last iteration is not feasible with respect to resident assignments, set $Z'_{j_l} = \infty$. Go to step 5.

Test for New Assignments Step 5: Calculate the weighted objective for iteration l as $Z_l = w1 \cdot (Z - Z_{1j_l} + Z'_{j_l}) + w2 \cdot Z_{2l}$.

Test: If $Z_l < \hat{Z}$, then update $\hat{Z} = Z_l$, store the TSP tour found at step 3 and the new assignments made at step 4. Go to step 6. Else, restore bus stop l, and the its assignments. Go to step 6.

Step 6: if $l < n - 1$, then go to step 2. Else, the best weighted objective found and stored in $\widehat{Z}$. Stop the algorithm.

In the next chapter, we present our computational experience with the algorithms described in this chapter.

Chapter 5

Numerical Testing

The proposed solution method is implemented in a series of three programs written in standard Fortran 77. The three modules that were implemented are:

- Initialization module: this constructs the matrices defining the GAP to be solved in clustering. The input to this code is the number of residences and their respective coordinates and weights, and the number of potential bus stops and their respective coordinates.
- GAP solver module: this solves the GAP defined by the initialization module. It outputs the number of clusters generated and the list of assignments of bus stops to each cluster.
- Bus routing module: here, bus routes are constructed on each cluster generated by the GAP solver module.

The clustering and routing algorithms are then tested on a sample test problem and for different characteristics of the PBRP. The aim is, first, to test the validity of the approach and the behavior of the algorithms. Second, we want to investigate the effect of the problem characteristics on the behavior of both algorithms and on the quality of solutions generated. Third, we wish to test the impact of different algorithmic strategies on the solutions obtained. We discuss in what follows the results of the tests of the clustering algorithm. Then, the results of the routing algorithm are presented.

5.1 Tests of the Clustering Algorithm

In this section we describe the test problem and the criteria used to test the clustering algorithm. We then investigate the effect of varying these criteria on the behavior of the algorithm in terms of running time and on the quality of clusters generated in terms of shape and size.

5.1.1 Experimental Setup

Test Problem

Empirical tests are conducted using the algorithm on sample data obtained from an electronics company situated in the suburbs of Ankara. Data provided consists of residence addresses for employees living in a selected district of Ankara. The municipal bus stops in the district are chosen to serve as potential bus stop sites. For this test problem, the number of residences is 155 with total number of employees to be serviced 270. The number of bus stops is 68. The number of bus types is set to 3 with capacities: 50, 40, and 30. Suppose there 10 vehicles, then the $0-1$ integer formulation we propose in Section 3.3.2 results in over 153000 variables and over 469400 constraints. After the first Step of the clustering algorithm, 58 bus stops turn out to have non zero loads. These bus stops are used in the tests of the branch and bound algorithm. With each cluster corresponding to one bus route, the number of employees assigned to each cluster shouldn't exceed cluster capacity. Consequently, at least six clusters need to be generated for this problem in order to satisfy total transportation requirements: e.g., five clusters of type 1 with capacity 50 and one cluster of type 3 with capacity 30.

Criteria Tested

The clustering algorithm is tested for four different criteria:

- **Distance Measure.** Abbreviated DM. The distance measure defines the cost added when a bus stop is assigned to a cluster. In order to control the cluster shapes generated we define in Section 4.1.3 four different distance measures.

- distance measure 1: Euclidean distance between the bus stop and the reference point of the cluster. The reference point is, as defined previously, the site of the cluster or the bus stop that corresponds to the cluster.
- distance measure 2: Euclidean distance between the bus stop and the last assigned bus stop to the cluster.
- distance measure 3: minimum Euclidean distance between the bus stop and the two endpoints of the cluster. The endpoints are the bus stops which do not precede any of the bus stops in the cluster.
- distance measure 4: minimum Euclidean distance between the bus stop, and the endpoints of the cluster or its reference point.

- **Cost Function.** Abbreviated CF. The cost function or the objective function of the clustering problem represents the optimization criteria considered when clusters are generated. The optimization criterion is to minimize the number of clusters and the cost of assignments of bus stops to clusters. The first term is measured by taking the sum of all assignment costs relative to the specified distance measure. The second term is measured by incurring a fixed cost whenever a new cluster is generated. To test the behavior of the algorithm with different combinations of both terms, we run the GAP solver module for two objective functions defined by: assignment term only, weighted sum of both terms. When weighting is used both weights sum up to one. Let WT1 be the weight associated with the assignment cost term. Then, WT2 = 1−WT1.

  Moreover, we test the algorithm by setting an upper bound on the unused capacity for generated clusters. We then compare the effect of imposing an upper bound on the unused capacity versus including a fixed cost term in the objective function.

- **Seed Selection Rule.** Abbreviated SSR. This is the criterion used to select the unassigned bus stop with which the next iteration in the branch and bound algorithm is initiated. We propose three different seed selection criteria.

  - seed selection rule 1: it chooses the unassigned bus stop with smallest index.

  - seed selection rule 2: the unassigned bus stop with maximum weight.
  - seed selection rule 3: the unassigned bus stop with minimum number of assignment sets in the corresponding list of potential assignments.

- **Fleet Characteristics.** Two characteristics of the fleet of buses are expected to affect the behavior of the algorithm. The number of bus types K increases the size of the problem and complicates clustering. The capacity (Cap i, where $i \in \{1, \ldots, K\}$) of buses is expected to affect the running time of the algorithm. Both characteristics are tested below. Since each cluster is traversed by one bus, than we use bus types and cluster types to refer to the same characteristic. In the test problem, there are three bus or cluster types: cluster type 1 (CT1) with capacity 50, cluster type 2 (CT2) with capacity 40, cluster type 3 (CT3) with capacity 30.

Measures of Performance

In what follows we use the following measures of performance to test the behavior of the clustering algorithm.

- **Running Time.** Abbreviated RT. This is the time the GAP solver module takes in cpu seconds. This quantity is used as a measure of the efficiency of the algorithm for all the tests except when the effect of the distance measure on the shape of clusters generated is tested. Using different distance measures aims mainly to generate different cluster shapes.

- **Objective Function Value.** Abbreviated OFV. This is the total cost of the clusters generated. This depends on the distance measure and the cost function used. For different distance measures, the objective function values are not comparable.

- **Number of Clusters.** Abbreviated NbCl. This gives the number of clusters generated by the clustering algorithm. This measure is of interest mainly when different cost functions are tested, since one of the main optimization criteria in the clustering stage is the number of clusters.

- **Number of Clusters of each Type.** Abbreviated CT i, where $i \in \{1, 2, 3\}$. This gives the number of clusters generated of each type. This measure is used to compare the effect of using different forms of the objective function.

- **Total Capacity Incurred.** Abbreviated TC. This is the total sum of capacities of clusters generated. The lowest feasible total capacity is equal to the total weight of residences which is 270.
- **Percent Excess Capacity.** Abbreviated %EC. This is the percent unused capacity of the clusters generated. The unused capacity is the total bus capacity less the total weight of residences.

5.1.2 Experiment 1: Effect of Distance Measure

The distance measure defines the cost associated with assigning a bus stop to a cluster. The cost is used in solving knapsack problems generated at Step 2 of the branch and bound algorithm. The four different distance measures described previously are used to generate different cluster shapes. The effect of each of the four distance measures on cluster shapes is illustrated by carrying out test runs of the clustering algorithm using the available data set. At this stage, we are interested in the generated cluster shapes which depend on the distance measure and not on the seed selection criteria or on the cost function used . Thus, we fix the cost function to be the sum of assignment costs and the seed to be the bus stop with the least number of possible assignments. Figures 5.1 to 5.4 show the clusters generated for each of the four distance measures respectively.

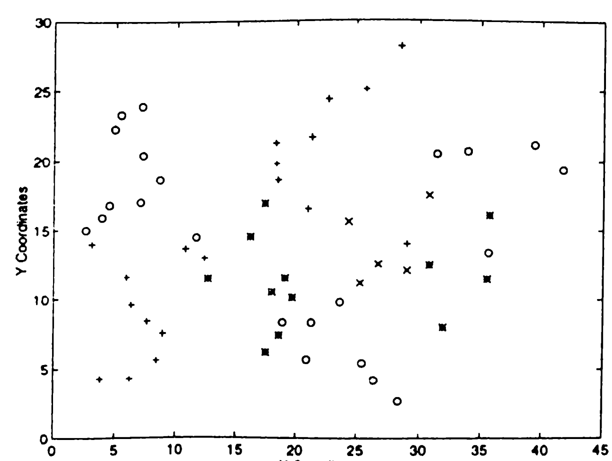

Figure 5.1: Clusters Generated via Distance Measure 1

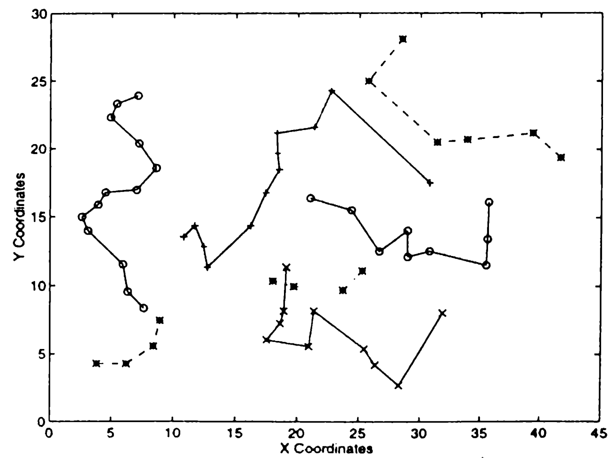

Figure 5.2: Clusters Generated via Distance Measure 2

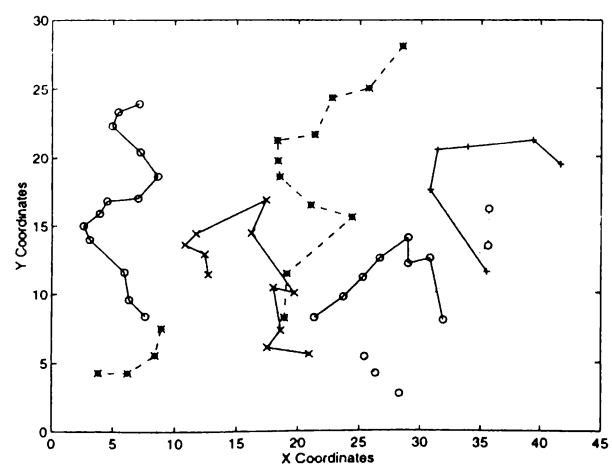

Figure 5.3: Clusters Generated via Distance Measure 3

As can be seen in Figure 5.1, clusters generated using distance measure 1 tend to take circular forms. This follows from the fact that the cost associated with each assignment using this distance measure is the Euclidean distance from the bus stop to the reference point of the cluster. Distance measures 2 and 3 result in linear clusters since the bus stop assigned next is the nearest to one of the endpoints and not to the cluster site itself. This is illustrated in Figures 5.2 and 5.3. Distance measure 4 gives a compromise between including in a cluster the bus stop nearest to the reference point of the cluster or to the endpoints which are bus stops already assigned to the cluster. Refer to Figure 5.4.

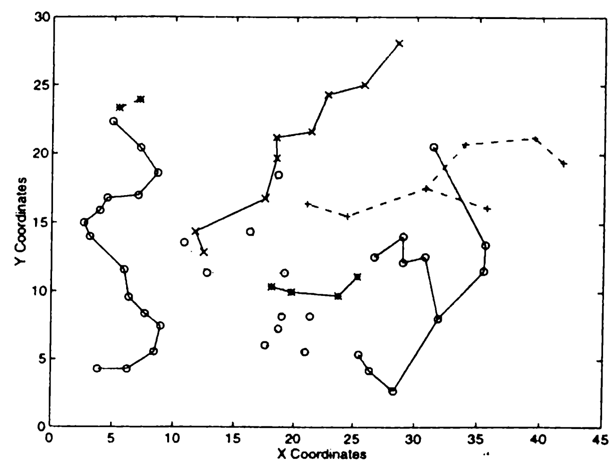

Figure 5.4: Clusters Generated via Distance Measure 3

5.1.3 Experiment 2: Effect of Cost Function

The objective function of the clustering problem consists of two terms: a fixed cost term associated with establishing a cluster, and an incremental cost term associated with the cost of assignment of a bus stop to a cluster. The fixed cost stands for the cost incurred when a new bus is used, this includes bus procurement costs and driver-hiring costs. The incremental costs are measured in distance and, using different distance measures, approximate the detour in the bus-route induced by assigning the next available bus stop to the cluster. The cost terms are not in the same unit of measure, so we combine both cost terms into a single overall objective by assigning weights to each of them. The weights assigned show the relative importance of each of the objective terms. Tests are conducted to illustrate the effect of various weights on the clustering process.

Additional tests are conducted to determine the effect of using an upper bound on the unused capacity for the clusters generated. The smaller the upper bound the lower the number of clusters generated. In these tests only the incremental costs are included in the objective function. The clustering code was executed with the forth distance measure and with seed selection rule one. The runs are carried out for three values of weights. We mention that the weights of both terms sum up to one. The output of main interest here is the number of clusters generated, the total capacity allocated and the cluster types used.

WT1	RT	NbCl	% EC	TC	CT1	CT2	CT3
1.0	2.25	9	6.90	290	0	4	7
.75	3.19	7	3.57	280	1	3	3
.25	2.50	6	3.57	280	4	2	0

Table 5.1: Effect of Weights in Cost Function

Referring to Table 5.1, as the weight assigned to the incremental cost term decreases with respect to the weight assigned to the fixed cost, the total assignment costs decrease while the total fixed costs increase. Hence, the number of clusters generated decreases and the proportion of unused capacity from the sum of cluster capacities obtained also decreases. In the case where a weight of 1 is assigned to incremental costs, the algorithm tends to assign bus stops to nearer clusters since no cost is incurred when a new cluster is opened. Consequently, a higher number of clusters with smaller capacities are generated. The optimal solution to such an objective function is to assign each bus stop to one of the clusters located in the same site, this will result in an objective function value of zero. However, the algorithm doesn't lead to such a solution because it tries to assign the maximum feasible number of bus stops to a cluster, once the cluster is opened. Assigning a weight of 0 to the incremental costs implies that all costs incurred are due to the fixed cost term. Consequently, the minimum number of clusters possible is generated. To achieve this, the clusters tend to be of higher capacities to allow for more assignments. As the incremental cost weight decreases, the number of clusters of types 2 and 3 decreases, while the number of clusters of type 1 increases. This results from the fact that as the weight of the fixed costs increase, the latter dominate the incremental costs. So, the number of clusters generated decreases. In addition, since only the number of clusters causes an increase in the objective function value, the total capacity may not be minimal even if the minimum possible number of clusters is generated.

Table 5.2 shows the results obtained for four different upper bound values on the unused capacity per cluster. First, the upper bound is set to infinity, that is, there is no restriction on the unused capacity per cluster. The resulting objective function value is lowest and the total capacity is highest. As the upper bound on the unused capacity decreases, the capacity usage increases and the objective function value increases also. Even though the number of clusters is the same, the total capacity decreases and the types used also change. When the capacity upper bound decreases type 3 is used more because smaller capacitated clusters imply lower assignment costs. However, this does not necessarily apply when the upper bound approaches zero.

UBd	RT	NbCl	OFV	% EC	TC	CT1	CT2	CT3
∞	2.40	8	117.29	6.90	290	0	5	3
7	2.68	8	131.98	3.57	280	1	2	5
4	2.58	8	129.91	0	270	1	1	6
2	2.14	6	179.39	0	270	3	3	0

Table 5.2: Effect of Unused Capacity Upper Bound

Comparing the number and type of clusters generated by each of the two forms for the cost function, we note that in the first case a lower number of clusters is generated, also clusters generated tend to have higher capacities. In the second case, the unused capacity is lowered while the number of clusters is not minimized. With the fixed cost term in the objective function, the clusters tend to be of higher capacities in order to generate less clusters. With the unused capacity bound, the number of clusters generated is not taken into account, and the clusters generated are the ones with lower incremental costs and unused capacity below the upper bound. The results of the test runs are consistent with expectations. Refer to Tables 5.1 and 5.2.

To sum up, we have the following results.

- Including a fixed cost term in the objective function decreases the number of cluster generated, while it does not necessarily minimize the total capacity.
- Imposing an upper bound on the unused capacity per cluster improves the capacity usage by allocating lower total capacity. However, it does not necessarily result in the minimum number of clusters required.
- Comparing the types of clusters generated by including a fixed term in the objective function to the ones generated by imposing an upper bound on unused capacity per cluster, we find that types of higher capacities are

generated in the first case, while types of lower capacities are generated in the second case, except when the upper bound approaches zero.

5.1.4 Experiment 3: Effect of Seed Selection Rule

At Step 2 of the branch and bound algorithm, we specify a seed selection rule that is used to form the list of potential assignments to be inspected in the next steps of the algorithm. The seed chosen at any stage of the algorithm affects the assignments made as well as the running time of the algorithm. The three seed selection criteria proposed are: unassigned bus stop with smallest index, unassigned bus stop with maximum weight, and unassigned bus stop with minimum number of assignment sets.

Table 5.3 shows a summary of the relevant performance measures for each seed

SSR	DM	RT	NbCl	OFV	% EC	TC
1	1	1.56	8	200.92	10.0	300
2	1	2.40	8	117.29	6.90	290
3	1	2.22	8	241.87	0	270
1	2	2.06	9	101.72	12.9	310
2	2	1.72	9	63.68	6.90	290
3	2	2.41	7	144.83	10	300
1	4	2.25	9	107.74	6.90	290
2	4	3.07	7	93.16	3.57	280
3	4	1.68	7	127.48	6.90	290

Table 5.3: Effect of Seed Selection Rule with Cost Function 1

selection rule. The results in this table are found for objective function 1, and distance measures 1, 2, and 4. Table 5.4 shows the same performance measures for objective function 2, and distance measure 1. The second selection criterion dominates the other two in terms of objective function value, which is to be expected since bus stops with big weights tend to determine the clusters generated. In terms of running time, the third criterion dominates the first two criteria in most of the runs. This is to be expected since the third criterion is designed to fit with the branch and bound algorithm. We note that the first criterion which is random performs fairly well both in terms of objective function values and running time. In summary:

SSR	WT1	RT	NbCl	OFV	% EC	TC
1	.75	2.04	8	223.5	3.57	280
2	.75	3.13	8	152.92	6.90	290
3	.75	1.24	7	269.83	10	300

Table 5.4: Effect of Seed Selection Rule With Cost Function 2

- Seed selection rule 2 dominates the other two rules in terms of objective function value.
- Seed selection rule 3 dominates the first two in terms of running time.

5.1.5 Experiment 4: Effect of Bus Types

SSR	DM	K	RT	NbCl	OFV	% EC	TC
2	4	3	3.07	7	93.16	3.57	280
2	4	2	1.89	7	88.22	12.9	310
3	4	3	1.68	7	127.48	6.90	290
3	4	2	9.71	7	145.61	12.9	310
3	2	3	2.44	7	144.83	10.0	300
3	2	2	1.44	7	148.64	12.9	310

Table 5.5: Effect of Bus Types

Two characteristics of the buses affect the behavior of the algorithm: the number of bus types, and the corresponding bus capacities. First, the number of potential clusters increases proportionally with the number of bus types and the location decision becomes more complicated. For example, for a clustering problem with $n = 20$ bus stops to be grouped into clusters, the number of potential clusters when two bus types ($K = 2$) are considered is $m = 40$, and when three bus types($K = 3$) are considered is $m = 60$. A bigger number of potential clusters results in longer assignment lists in Step 2 of the branch and bound algorithm which leads to more branches to be inspected in the search tree. Thus we expect the running time to increase when increasing the number of bus types. Table 5.5 shows sample runs of the algorithm for $K = 2$ and $K = 3$, and for two different distance measures. The cost function counts only the assignment costs. Confirming our expectations, the running time in the three cases is reduced by about 38.4%, 40.2% and 42.2% respectively.

SSR	DM	Cap 1	Cap 2	Cap 3	RT	NbCl	OFV	% EC	TC
3	4	50	40	30	1.68	7	127.48	6.90	290
3	4	60	50	40	1.54	6	134.29	10.0	300
3	4	70	60	50	1.04	5	144.48	3.57	280

Table 5.6: Effect of Bus Capacities

Second, increasing bus capacities allows for more assignments to be made whenever a cluster is established. Then, both the number of bus stops and the number of potential clusters remaining for the next steps are lower. Both factors cause the number of knapsack problems in Step 2 of the algorithm to be smaller and the assignment lists to be shorter in length. Thus the program is expected to terminate faster. In fact, the ratio of total bus stop weights to individual bus capacities is a main factor in determining the running time of the branch and bound algorithm. A higher ratio implies that a higher number of clusters need to be generated, a bigger search tree and longer running time. Table 5.6 shows sample tests of the algorithm over three different capacity values while bus stop weights and number of bus types are kept constant. The third seed selection rule, the fourth distance measure, and the first form of cost function are used in these tests. According to the results shown in the table, the running time decreases by 8.3% when each of the three capacity values is increased by 10, and by 38.1% when increased by 20. As a summary, the following results are highlighted.

- Fleet heterogeneity complicates the clustering problem. Moreover, increasing the number of types results in significant increases in running time: around 40%.
- Using buses of higher capacities decreases the running time of the algorithm significantly.

5.2 Tests of the Routing Algorithm

Routing is the second phase of the heuristic solution method we propose to solve the PBRP. At this stage, we have a set of independent clusters generated by the clustering algorithm. We aim to construct bus routes to service the bus stops in each cluster. In addition to routing, we also locate the bus stops and assign residents to them. The optimization criteria in routing are:

- total walking distance of residents to their bus stops, and
- total route length.

Since the routes are constructed for each cluster, then the routing problem reduces to a set of independent smaller problems, each defined on one cluster. The routing algorithm we described earlier uses the minimum walking distance criterion to make initial assignments of residents to bus stops and Eastman's TSP algorithm to construct tours on the located bus stops. Since routes are constructed on individual clusters, the TSP algorithm finds the optimal solution in a reasonable amount of time. The main step of the routing algorithm is the improvement scheme which is used to improve the quality of solutions found through a tradeoff between shorter routes and better assignments of residents. This is achieved by using weighting. The first part of the tests we perform in this section tests the quality of solutions generated by the routing algorithm and the behavior of the algorithm for different weights. The second part tests both the clustering and the routing algorithms: for a selected weighting, we perform the routing algorithm on clusters resulting from different distance measures and test how well the distance measures approximate the actual routes.

5.2.1 Experiment 5: Effect of Weights on Routes Generated

Following the optimization criteria of the routing phase, the objective function terms are: the total resident walking distance, and the total route length. We associate weights with each of the terms to signal the importance of each of the terms over the other. If WT1 is the weight associated with the total resident walking distance, than WT2 = 1−WT1, this is the weight associated with the total route length. In the tables below, the notation follows that used in Section 4.2.2:

Z_{1l}: initial total resident walking for bus stop j_l. $l = 1$ correspond to the firm, so $Z_{11} = 0$.

Z'_{1l}: total resident walking distance for residents previously assigned to the removed bus stop j_l. Since, no assignments are made to the firm then $Z'_{11} = 0$.

Z_{2l}: length of tour resulting at feasible improvement step l, here bus stop j_l is removed. For $l = 1$, the tour visits the firm and all potential bus stops in the cluster.

In Tables 5.7 and 5.8, the entries in the three rightmost columns give the weighted objective function values for the weights specified in row 2. All quantities are measured in the same unit of distance.

Total Resident Walking Distance, Z=40			Number of Bus Stops, n=6		
Z_{1l}	Z'_{1l}	Z_{2l}	WT1		
			0.25	0.5	0.75
0	0	208	166	124	71.5
6	18	207	168.25	129.5	90.75
2	6	205	164.75	114.5	84.25
Minimum Objective Value in Step			3	3	1

Table 5.7: Effect of Varying Weights on Routes Generated: cluster of 5 potential bus stop sites, and 2 feasible improvement steps.

Referring to Tables 5.7 and 5.8, when the weight associated with the route length objective is low, the use of all potential bus stops in the cluster gives the minimum total weighted objective function. Using all potential bus stops results in a minimal value for total resident walking distance since residents are then assigned according to the minimum walking distance criterion. As the weight associated with the total route length objective increases, routes with less number of bus stops give smaller weighted objective function values. The results depicted in the tables confirm these expectations.

Total Resident Walking Distance, Z=42			Number of Bus Stops, n=9		
Z_{1l}	Z'_{jl}	Z_{2l}	WT1		
			0.25	0.5	0.75
0	0	170	138	106	74
5	10	169	126.75	121.5	77.5
1	4	169	138	95.75	76
4	10	168	138	108	78
4	15	169	140	111	82
Minimum Objective Value in Step			2	3	1

Table 5.8: Effect of Varying Weights on Routes Generated: cluster of 8 potential bus stop sites, and 4 feasible improvement steps.

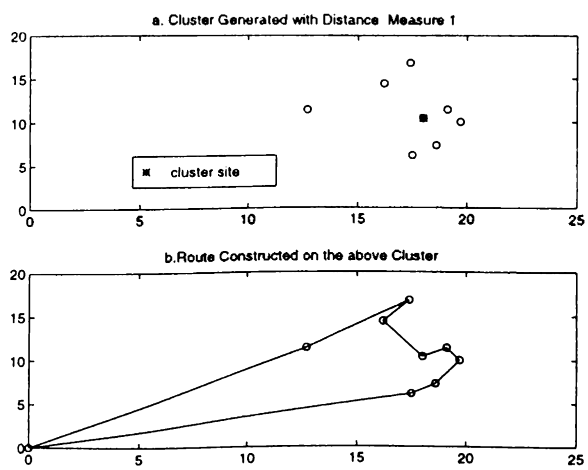

Figure 5.5: Route on Cluster Generated via Distance Measure 1

5.2.2 Experiment 6: Quality of Routes for Different Distance Measures

In the branch and bound algorithm of the clustering phase, we propose four different distance measures. The first distance measure is the simple Euclidean distance, while the other three aim to approximate the detour in the route caused by the inclusion of a candidate bus stop. In order to analyze the performance of the distance measures, we construct the routes on a set of sample clusters for the four distance measures. Figures 5.5 through 5.7 show the routes generated by the clustering algorithm and the routes corresponding to clusters.

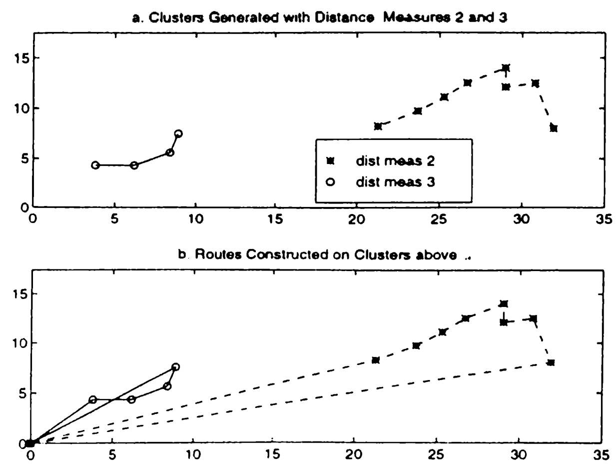

Figure 5.6: Routes on Clusters Generated via Distance Measures 2 and 3

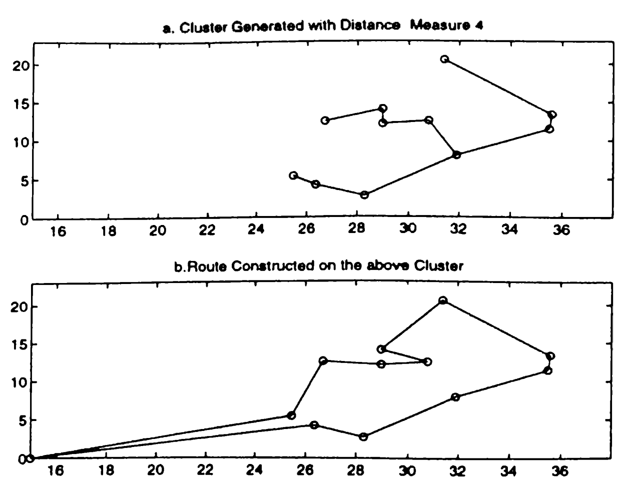

Figure 5.7: Route on Cluster Generated via Distance Measure 4

The routes constructed on clusters generated with the use of distance measure 2 and 3 imitate the line of assignments of bus stops to clusters. This results from the fact that the distance measures are developed to generate clusters in the form of routes. Distance measures 2 and 3 would be more efficient when the bus stops are well spaced in the cluster. If the bus stop are densely distributed, then distance measure 4 would result in better clusters and consequently routes. This is illustrated in Figures 5.6 and 5.7 respectively.

Chapter 6

Conclusion

In this thesis, we discussed the characteristiocs of a personnel bus routing problem termed PBRP. We gave a multi-objective mathematical formulation of the problem, and developed a heuristic method to solve it. We defined the PBRP as the problem of designing a set of pickup and delivery routes, locating a set of bus stops, and assigning employees to bus stops in order to daily transport the employees back and forth from the company to their residences.

The PBRP is a location routing problem compounded by several complications. First, the problem has a multi-objective nature. The main objectives of the problem are: minimize personnel satisfaction with the transportation system, minimize system operating costs, and minimize fleet size. Second, the PBRP is a large scale problem in terms of the number of personnel to be serviced at a time and the underlying geographical area. Third, the fleet of vehicles is assumed to be heterogeneus. Vehicles have different capacities. Which type of vehicle to put in operation is a decision of the problem. Finally, location routing problems, and consequently the PBRP, are difficult problems and are in most of the instances NP-hard problems.

Based on the PBRP characteristics and objectives, we gave a mathematical model that considers the multi-objective nature of the problem and the heterogeneity of the fleet. Then, we developed a two-phase solution methodology to generate solutions to the model. Phase one of the proposed solution method is a clustering algorithm. The clustering algorithm is devised to group employees' residences into clusters each of which is serviced by one vehicle. This is achieved by solving a structured generalized assignment problem via a branch and bound heuristic algorithm. The branch and bound heuristic algorithm is based on the set partitioning tree search methods and it uses iterative column generation. The branch and bound algorithm minimizes fleet size, decides on vehicle types to be put in operation, and implicitely minimizes total route length through using different distance measures. The distnace measures are designed to appoximate the route length resulting once a cluster is set up. Moreover, we judged the objective of employees satisfaction to be of high priority. Hence, we kept this objective at its maximal level at the clustering stage by using a minimum walking distance criterion to assign residents to bus stops. In phase two, we solved a set of independent routing problems each defined on a cluster. The routing problem consists of three related subproblems: locate a set of bus stops, assign employees to these bus stops, and construct a route on the bus stops. The objectives dealt with here are total employees walking distance and total route length. To solve the routing problem, we used a routing algorithm that solves a traveling salesman problem and performs an improvement scheme iteratively. At each iteration, a subset of bus stops is removed in a trial to improve the total objective function value. The numerical testing of both the clustering and routing algorithms indicate that the results are consistent with expectations.

- Different distance measures affect the shape of the clusters generated and appoximate the routes.
- Using different seeds to initialize the steps of the branch and bound algorithm affect both the costs incured and the running time.
- Using fixed cost terms to penalize the use of clusters result in a minimum number of clusters generated.
- Imposing upper bounds on the unused capacity in generated clusters improves capacity usage.
- Vehicle characteristics affect the complexity of the branch and bound algorithm, the running time, and the size of problems it can handle.
- When distance measures that appoximate the route are used in the clustering phase, the routes constructed have approximately the same shape as the

clusters.

- Associating higher weight with the walking distance objective leads to more bus stops located and higher routing costs, and vice-versa.

By explictly examining all these considerations, decision makers can generate a set of routes that are cost efficient and that achieve a desirable level of personnel satisfaction.

For future research, we would suggest to include the load balancing criterion that we propose in the mathematical model as an objective in the heuristic method. Other measures for loadbalancing could also be tried such as minimizing the maximum excess capacity. In this case, the load balancing objective would be included in the clustering algorithm since at that stage the assignment of employees to buses is performed, and consequently the loads determined. In addition to that, other measures of personnel satisfaction can be used. Such a measure may be minimizing the maximum walking distance instead of the total walking distance of employees. Furthermore, the routing algorithm can be changed to make use of the cluster shapes generated by the clustering algorithm, since they are found to approximate route shapes.

Bibliography

[1] Assad, A. A. and Golden B. L. "Arc Routing Methods and Applications" in Ball, M. O., Magnanti, T. L., Monma, C. L., and Nemhauser, G. L. (editors), *Handbooks in OR & MS: Network Models* Elsevier, Amsterdam (1995) 375-481.

[2] Bodin, L. and Berman, L. "Routing and Scheduling of School Buses by Computer" *Transportation Science* **13**, (1979) 113-129.

[3] Bowerman, R., Hall, B. and Calamai, P. "A Multi-objective Optimization Approach to Urban School Bus Routing: Formulation and Solution Method" *http://dial.waterloo.ca/ phcalama/home.html*, (1995).

[4] Chapleau, L., Ferland, J. and Rousseau J. "Clustering for Routing in Densely Populated Areas " *European Journal of Operational Research* **20**, (1985) 48-57.

[5] Christofides, N. "Graph Theory: an Algorithmic Approach" *Academic Press, Inc.* (1975) 39-46.

[6] Christofides, N. , Mingozzi, A. and Toth P. "State-space Relaxation Procedures for the Computation of Bounds to Routing Problems" *Networks* **11**, (1981) 145-164.

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

[8] Christofides, N. "Vehicle Routing", in Lawler, E. L., Lenstra, J. K., Rinnooykan, A. H. G. and Shmoys D. B. (editors), *The Travelling Salesman Problem: a Guided Tour of Combinatorial Optimization*, Wiley, Chichester (1985) 431-448.

[9] Clarke, G. and Wright, J. "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", *Operations Research* **12(4)**, (1964) 568-581.

[10] Dulac, G. , Ferland, J. and Fogues, P. A. "School Bus Routes Generator in Urban Surroundings" *Computers and Operations Research* **7**, (1980) 199-213.

[11] Fisher, M. L., Jaikumar, R. and Van Wassenhove, L. N. "A Multiplier Adjustment Method for the Generalized Assignment Problem" *Management Science* **32**, (1986) 1095-1103.

[12] Fisher, M. L. and Jaikumar, R. "A Generalized Assignment Heuristic for Vehicle Routing" *Networks* **11**, (1981) 109-124.

[13] Francis, R. L., McGinnis, L. F. and White, J. A. "Locational Analysis" *European Journal of Operational Research* **12**, (1983) 220-253.

[14] Garfinkel, R. S. and Nemhauser, G. L. "The Set Partitioning Problem: Set Covering with Equality Constraints" *Operations Research* **17**, (1969) 848.

[15] Gillett, B. E. and Miller, L. R. "A Heuristic Algorithm for the Vehicle Dispatch Problem" *Operations Research* **22**, (1974) 340-349.

[16] Gillett, B. E. "Introduction to Operations Research: A Computer-Oriented Algorithmic Approach" *McGraw-Hill, Inc.* (1976)110-125 / 210-210.

[17] Kulkarni, R. V. and Bhave, P. R. "Integer Programming Formulations of Vehicle Routing Problems" *European Journal of Operational Research* **20**, (1985) 58-67.

[18] Laporte, G. "The vehicle Routing Problem: an Overview of Exact and Approximate Algorithms", *European Journal of Operational Research* **59**, (1992) 345-358.

[19] Laporte, G. "Location-Routing Problems" in Golden, B. and Assad, A. (editors), *Vehicle Routing: Methods and Studies* North Holland (1988) 163-197.

[20] Magnanti, T. L. "Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospects" *Networks* **11**, (1981) 179-214.

[21] Ross, G. T. and Soland, R. M. "Modelling Facility Location Problems as Generalized Assignment Problems" *Management Science* **24**, (1977) 345-357.

[22] Ross, G. T. and Soland, R. M. "A Branch and Bound Algorithm for the Generalized Assignment Problem" *Mathematical Programming* **8**, (1975) 92-103.

[23] Savas, E. "On Equity in Providing Public Services" *Management Science* **24**, (1978) 800-808.

[24] Savelsbergh, M. W. P. and Sol, M. "The General Pick-up and Delivery Problem" *Transportation Science* **29**, (1995) 17-29.

[25] Swersey, A. J. and Ballard W. "Scheduling School Buses" *Management Science* **30**, (1984) 844-853.

[26] Szidorovsky, F. , Gerson, M. and Duckstein, L. "Techniques for Multi-objective Decision Making in Systems Management" *Elsevier*, Amsterdam (1986).

VITA

Fatma Gzara was born on July 23, 1972 in Djemmal, Tunisia. She received her high school education at the Pioneer Technical School in Ariana, Tunisia. She has graduated from the Department of Industrial Engineering, Bilkent University, in 1995. In October 1995, she joined to the Department of Industrial Engineering at Bilkent University as a research assistant. Since then, she worked with Dr. Mustafa Ç. Pınar for her graduate study at the same department.