INTEGRATING OPTIMIZATION HEURISTIC TECHNIQUES AND GIS FOR SOLVING THE DISTRICTS DETERMINATION PROBLEM IN URBAN AREAS

By

HISHAM ABDEL MASSIH

A Thesis

Submitted in Partial Fulfillment of the Requirements for
the Degree of Master of Science in Computer Science
Department of Computer Science

Faculty of Natural and Applied Sciences
Notre Dame University -Louaize
Zoukl Mousbeh, Lebanon
Spring 2000

INTEGRATING OPTIMIZATION HEURISTIC TECHMIQUES AND GIS FOR SOLVING THE DISTRICTS DETERNINATION PROBLEM IN URBAN AREAS

By

HISHAM ABDEL MASSIH

Approved:

Fouad Chedid: Associate Professor of Computer Science and Chairperson.

Advisor.

Khaldoun El-Khaldi: Assistant Professor of Computer Science.

Member of Committee.

Hoda Maalouf: Assistant Professor of Computer Science.

Member of Committee.

Badih Ghusayni: Associate Professor of Mathematics.

Member of Committee.

Date of thesis defense: Thursday, 22 June 2000

ACKNOWLEDGMENTS

"Attainment of the optimum is much less important for complex systems. It would be nice to be perfect: meanwhile, we can only strive to improve."

I wanted to start the acknowledgements with the above quotation for its relevance to the subject discussed in this thesis and as an affirmation of the solitary perfection of GOD.

I would like to express my sincere thanks to Dr. Fouad Chedid for his help and guidance during the process of preparing and writing this thesis. Without his patient analysis and valuable advice I wouldn't have had the chance to finish this thesis.

I am grateful to my parents for their support, sacrifice, and encouragement during the difficult and most delicate times.

ABSTRACT

Recent research in Geographic Information Systems (GISs) has created the kind of systems capable of modeling a number of interesting real world phenomena. Some applications where GISs have made significant contribution include routing, scheduling, dispatching, transportation logistics, vehicle tracking, market research, construction planning, facility management and resource distribution. However, there seems to exist little connection between GIS solutions and much related research being carried out in other disciplines. This thesis studies the advantages of integrating recent research in heuristic combinatorial algorithms with GIS to solve real world problems. In particular, we investigate the use of four different heuristic algorithms to solve the problem of mail distribution in urban areas where districts determination is a priority goal for automation. A variation to the local search phase of the GRASP (Greedy Randomized Adaptive Search Procedures) algorithm is proposed and implemented which improved the workload balance between mailmen.

TABLE OF CONTENTS

LIST OF FIGURES vii

I THE PROBLEM 1

II BACKGROUND 3

2.1 What Makes an Optimization Problem ? 3

2.2 Heuristic Combinatorial Algorithms 4

2.2.1 Genetic Algorithms 5

2.2.2 Simulated Annealing 6

2.2.3 Tabu Search 7

2.2.4 Greedy randomized Adaptive Search Procedure (GRASP) 8

2.3 Random Algorithms 9

2.4 Geographic Information Systems 10

2.4.1 GIS Data Sources 12

2.4.2 Spatial Functions 13

2.5 Integrating Heuristic Combinatorial Algorithms With GIS 15

III CASE STUDY 18

3.1 Overview of Mail Distribution in Urban Areas 18

3.2 Problem Parameters 19

3.3 Graph Representation 20

3.4 Heuristic Solutions and GRASP Algorithms 22

3.4.2 Algorithms H1 & H2 22

3.4.2 Algorithms H3 & H4 23

IV IMPLEMENTATION & IMPROVEMENT 24

4.1 GIS Model Versus Graph Partitioning Model 24

4.2 Implementation of Heuristic Algorithms and Empirical Observation 26

4.3 Revised Local Search Phase 36

V CONCLUSION AND FUTURE WORK 39

VI REFERENCES 41

VII APPENDIX 43

LIST OF FIGURES

Figure

2.1.	GRASP Algorithm	8
2.2.	Layer Approach in GIS	11
2.3.	GeoProcessing	14
2.4.	Dynamic Segmentation	15
3.1.	Graph Representation Model	21
4.1.	Problem Representation in GIS Model	25
4.1.	Snapshot of the Prototype Interface	26
4.2.	Greedy Random	29
4.3.	Choose-Root for H2	29
4.4.	Districts Generated by H1	30
4.5.	Districts Generated by H2	31
4.6.	GRASP Algorithm For The Districts Determination Problem	32
4.7.	Choosing a SDU in H3 and H4	33
4.8.	Local Search For The Districts Determination Problem	34
4.9.	Districts Generated by H3	35
4.10.	Districts Generated by H4	35
4.11.	District Fusion Process	36
4.12.	Districts Generated by H5	37
4.13.	Districts Generated by H5	38

CHAPTER I

THE PROBLEM

Heuristic approaches have always had a very strong appeal in solving optimization problems. This is due mainly to the fact that most interesting optimization problems are NP-complete, and unless P = NP, their complexity is exponential in the size of the input. Heuristic optimization techniques trade optimality for time complexity. Recent researches in most fields where a large number of parameters are involved, are relying more and more on adopting different heuristic techniques. We mention, for example, scheduling (bus-driver and flight scheduling), routing (vehicle routing with time window) and several other operation research applications. On the other hand, Geographic information systems (GISs) are becoming routine analysis and display tools for spatial data and are extensively used in applications such as land-use mapping (for urban planning purposes), transportation mapping and analysis (for determining efficient transportation routes for deliveries and emergency response), geo-demographic analysis (for facilities location), utilities infrastructure mapping (for precise gas, water, and electric line mapping), and multiple applications in natural resource assessment (including water quality assessment and wildlife habitat studies). GISs allow efficient and flexible storage, display, and exchange of spatial data, as well as interface opportunities for models of all kinds. Users of GIS include governmental and non-governmental organizations such as conversation groups, universities, and research institutes.

Of the many areas where GIS is being applied, the transportation industry in particular, is vitally tied to geography. GIS links geography to data so goods and people can be moved more efficiently and economically than ever before. In this thesis we investigate the integration of heuristic optimization techniques with GIS for solving the districts determination problem for mail distribution in urban areas. This problem asks to find an optimal way to deliver mails in large populated cities. We are given a network of streets, and a number of mailmen along with some criteria. Each city is divided into several distribution zones. Our purpose is to determine the districts in each distribution zone, where each district represents a connected street network that will be traversed by one mailman, so that two main goals are met: **(i)** to minimize the number of mailmen working in a distribution zone; and **(ii)** to evenly balance, among these mailmen, the daily workload of mail delivery, within this same distribution zone. [3]

The rest of the thesis is organized as follows: chapter II gives a review of the major algorithms used in solving the districts determination problem, and illustrates the needs and impacts of integrating heuristic optimization algorithms with GISs. Chapter III defines the problem of mail distribution in urban areas while emphasizing the major constraints and parameters involved. Chapter IV proposes improvements at the model and algorithmic levels to solve the mail distribution problem, and describes some implementation and empirical observations. Finally chapter V concludes the thesis with a summary and directions for future research.

CHAPTER II

BACKGROUND

This chapter provides a review of major heuristic algorithms used for solving optimization problems in general. We begin by reviewing the characteristics of optimization problems, then we focus on four different heuristic combinatorial techniques, namely genetic algorithms, simulated annealing, tabu search and Greedy Randomized Adaptive Search Procedures (GRASP). The last two sections introduce the GIS technology with a focus on the benefits of integrating GIS with heuristic optimization techniques to solve real world problems.

2.1 What Makes an Optimization Problem ?

In general, optimization problems consist of three basic components:

- An *objective function* which will be minimized or maximized.
- A set of *unknowns* or *variables* which affect the value of the objective function.
- A set of *constraints* that allow the unknowns to take on certain values but exclude others.

An optimization algorithm is then defined as the sequence of steps that will be followed in finding values for the *variables* that minimize or maximize the *objective function* while satisfying the *constraints*.

Optimization algorithms could be classified into two categories : exact algorithms where optimal solutions are always found (e.g. the Simplex method for linear programming, Dijkstra's algorithm for finding shortest paths), and approximate algorithms where sub-optimal solutions are sought.

2.2 Heuristic Combinatorial Algorithms

Most combinatorial optimization problems suffer from the combinatorial explosion effect where the number of possible solutions that the algorithm has to search through grows exponentially with the size of the input. The term heuristic is used to describe those algorithms which may or may not lead to optimal solutions and therefore they may be considered as approximation algorithms. Central to many heuristic algorithms is the idea of *iterative improvement*, which consists of taking current solutions and apply a perturbation to move to another possible solution in an attempt to find the globally optimal solution. The most popular method used in heuristic algorithms is the greedy method. This method searches a set of candidates for the best remaining candidate and tries to add it to a set of chosen candidates that was initially empty. If the enlarged set of candidates is no longer feasible, the just added candidate is removed and never considered again. However, if the enlarged set is still feasible, then the just added candidate stays in the set of chosen candidates [13].

Heuristic combinatorial optimization algorithms aim to find optimal solutions to problems where exact algorithms are not practical. One of the most famous such problem is the Travelling Salesman Problem (TSP) where finding the shortest tour between n cities results in about n! possible number of tours. Testing all possible tours even for reasonably sized problems requires impractical amount of time. The aim of combinatorial optimization algorithms is to search only a fraction of all possible solutions to find a near-optimal solution using heuristic methods [11].

Strategies for optimization range from simple approaches such as hill climbing, to more complex algorithms such as simulated annealing [9], tabu search [6], genetic algorithms [7], and GRASP (Greedy Randomized Adaptive Search Procedures) [4][14].

2.2.1 Genetic Algorithms

Genetic algorithms (GA) were formally introduced in the 1970s by John Holland at University of Michigan. The continuing performance improvements of computational systems has made them attractive for some types of optimization. In particular, genetic algorithms work very well on mixed (continuous and discrete) combinatorial problems.

A GA represents a solution to the problem as a genome or chromosome. The genetic algorithm then creates a population of solutions and applies genetic operators such as mutation and crossover to evolve the solutions in order to find the best one (s).

The three most important aspects of using genetic algorithms are: **(i)** definition of the objective function, **(ii)** definition and implementation of the genetic representation, and **(iii)** definition and implementation of the genetic operators. Once these three aspects have been defined, the genetic algorithm allows many different variations to improve its performance in finding optimal solutions .

Genetic algorithms are emerging from the labs and proving mature enough to help solve some real world NP-complete problems. Researchers are using GAs to design communication networks and nuclear power plant fuel configuration, manage financial portfolios, optimize manufacturing schedules, build adaptive control systems, interpret noisy data, etc.

In particular, GAs would be very beneficial for solving GIS problems. One recent application shows major improvements in solving the map labeling problem which has been around for twenty years and has been proven to be NP-complete. In brief, the map labeling problem could be defined as follows: given a set of n points in the plane, with labels associated with them. Each label can be placed in a fixed number of predefined positions and orientations. We need to give a positioning for each label such that the number of labels which do not intersect with other labels is maximized The GA based solution provides the GIS field with high quality map labeling [20]

2.2.2 **Simulated Annealing**

The topic of simulated annealing was first introduced in 1983 [9]. In that paper titled "Optimization by Simulated Annealing" the authors described an iterative improvement algorithm for finding solutions to arbitrary optimization problems. The main idea of this method was to recognize that nature performs an optimization of the energy of a crystalline solid when it is annealed to remove defects in the atomic arrangement. Hence, SA is based on an analogy of the annealing of metals. It works by searching through the solution space looking for continual improvement of the solution, but with a large enough random element included so that it is not likely to become stuck prematurely at a local minima. The random element starts off quite large and is then reduced to zero in the same way that temperature starts out high and is reduced in annealing [9].

Deciding whether the optimization problem has been enough annealed is a delicate matter that should be treated carefully. One possible situation is that the generated solution might have become frozen into a local minima and not the global or lowest possible minimum of the objective function. Hence, the procedure of getting out of this trap is to make more random moves at each temperature or make smaller temperature decreases in each annealing cycle.

A recent GIS project demonstrates the advantages of using the simulated annealing optimization technique on large-scale field survey data to inform the design of protected area networks. This study identifies all bird species that did not occur in the major parks of the province in southern Ontario, along with the minimum number and location of UTM grid squares that would represent 5 percent of the breeding occurrences of different bird species. A number of possible combinations of grid squares that meet the criteria was tested and the results were visualized using GIS to provide some insights on potential gaps in bird representation in the southern Ontario park network.

2.2.3 Tabu Search

Tabu search is a strategy for guiding known heuristics to overcome local optimality. Although still in its infancy, this metaheuristic has been reported in the literature during the last few years as providing successful solution approaches for a great variety of problem areas. We mention that more advanced issues include specialized methods that incorporate different means of intelligence in providing history based search trajectories and allowing the inclusion of learning capabilities.

Using this many solution approaches are characterized by identifying a neighborhood of a given solution This neighborhood may contain other transformed solutions that can be reached in a single iteration. A transition from a feasible solution to a transformed feasible solution is referred to as a move. A starting point for tabu search is to note that such a move may be described by a set of one or more attributes (or elements), and these attributes (properly chosen) can become the foundation for creating an attribute based memory. A move may either result in a best possible improvement or a least possible deterioration of the objective function value. Without additional control, such a process can cause a locally optimal solution to be re-visited immediately after moving to a neighbor, or in a future stage of the search process. To prevent the search from endlessly cycling, the tabu method restricts the search to a subset of admissible moves (consisting of admissible attributes or combinations of attributes). The goal is to permit good moves in each iteration without re-visiting solutions already encountered [6].

Recent applications in transportation logistics are incorporating GIS as Spatial Decision Support System (SDSS). GIS provides realistic risk analysis methods which improve transportation decisions that rely on risk predictions. The inspection station location model (ISLM) is one undergoing project that can be used to locate facilities inspecting trucks travelling on the street network. ISLM locates inspection stations to intercept traffic flow as early in their trip as possible so that the risk resulting from dangerous trucks is minimized. In solving this optimization problem tabu search algorithm performed much better than the greedy algorithm. It overcome a major drawback of the greedy algorithm where facilities cannot be relocated even if the protection they provide is taken over by other facilities. Tabu search was able to find the optimal solutions for all the test problems in the full protection case.

2.2.4 Greedy Randomized Adaptive Search Procedure (GRASP)

Greedy Randomized Adaptive Search Procedure or GRASP is a modern heuristic search technique based on an iterative process consisting of two phases, a construction phase and a local search phase. A generic pseudo-code is given in Figure 2.1 below.

```
Procedure grasp ()
1.    InputInstance ();
2.    for Grasp stopping criterion not satisfied
          2.1    ConstructGreedyRandomizedSolution(Solution);
          2.2    LocalSearch (Solution);
          2.3    UpdateSolution (Solution, BestSolutionFound);
3     end for;
4     return (BestSolutionFound);
end grasp;
```

Figure 2.1. GRASP Algorithm

In the construction phase, a solution is iteratively constructed from scratch, building one district at a time until all nodes have been assigned. At each iteration, the choice of the next element (SDU) is determined by ordering all elements in a candidate list with respect to a greedy function. This function measures the benefit of selecting each element. The heuristic is adaptive because the benefits associated with every element, are updated at each iteration of the construction phase to reflect the changes brought on by the selection of the previous element. The probabilistic component of GRASP is characterized by randomly choosing one of the best candidates but not necessarily the top candidate. The solution generated by GRASP construction phase are not guaranteed to be locally optimal. Hence, it is beneficial to apply local search to attempt to improve each constructed solutions [14].

A local search algorithm works in an iterative way by successively replacing the current solution by a better solution in the neighborhood of the current solution. It terminates when no better solution is found in the neighborhood [4].

In this thesis, GRASP will be studied in details, implemented using GIS, and improved in the context of a case study .

2.3 Random Algorithms

Randomness has proven itself to be a useful resource for developing provably efficient algorithms and protocols. As a result, the study of randomized algorithms has become a major research topic in recent years.

Complex problems often require tremendously slow processes to solve. Randomized algorithms are introduced as an alternative method for solving these problems. Randomized algorithms save time by choosing a starting point arbitrarily, rather than deciding where the best starting point is. As a result, the algorithm will have a high probability of producing the correct solution. Almost always, several iterations may be required to ensure a correct solution, but this is often much faster than executing a complicated, deterministic algorithm.

Randomization has played a crucial role in the design of both sequential and parallel algorithms. The last decade has witnessed enormous growth in the area of randomized algorithms. During this period, randomized algorithms went from being a tool in computational number theory to finding widespread applications in many problem domains. In particular, GIS optimization problems where complex spatial constraints are involved, often require tremendously slow processes to solve. Randomized algorithms as an alternative method to solving these spatial problems save time and have a high probability of producing the correct solution.

Major topics covered include randomization techniques for linear and integer programming problems, randomization in the design of approximate algorithms for combinatorial problems, randomization in parallel and distributed algorithms, Practical implementation of randomized algorithms, de-randomization issues, and pseudo-random generators [18].

2.4 Geographic Information Systems

Maps model geographic features that can occur naturally (rivers, vegetation), can be constructions (roads, pipelines, buildings) or subdivision of land (counties, lots, political divisions). In modeling the real world, maps uses geometrical objects like points, lines and areas. Symbols and labels (text) are used to describe these objects. Spatial relationships are implicit on map sheets; however, they depend upon a map reader to interpret them.

GIS provides the same capabilities as maps in modeling the real world. However, the power of GIS come from not only the ability to store geographic data, but also from the ability to analyze it more efficiently and conveniently than is possible with paper maps.

The GIS literature is full of definitions of what does a GIS exactly mean. One definition that we find very intriguing is the following: "A Geographic Information System (GIS) is a computer based tool for mapping and analyzing things and events that happen on earth." Given that definition, a GIS is then expected to store and represent information about the world as is. To do so, a GIS collects data into thematic layers [17] that can be linked geographically. For example the data themes for urban planning may contain streets, utility lines, transportation features, tax and zoning conditions, landowners and improvements to properties (See Figure 2.2.a). Layers can also represent different elevations, as with the floor space type in different stories of buildings (See Figure 2.2.b) or they may represent different points or intervals of time for a given theme, for example, in the case of data collected from different censuses (See Figure 2.2.c).

This layered data set approach is simple but extremely powerful and versatile in solving many real world problems ranging from tracking delivery vehicles to recording details of planning transportation logistics, to modeling global atmospheric circulation. In fact different data layers can be analyzed together or individually to deepen insights into problems. Hence, when new data layers are added into analysis, it expands the perspective of the decision making process.

In addition, GIS technology integrates common database operations such as query and statistical analysis with the unique visualization and geographic analysis benefits offered by maps. These abilities distinguish GIS from other information systems and make it valuable to a wide range of public and private enterprises for explaining events, predicting outcomes, and planning strategies.

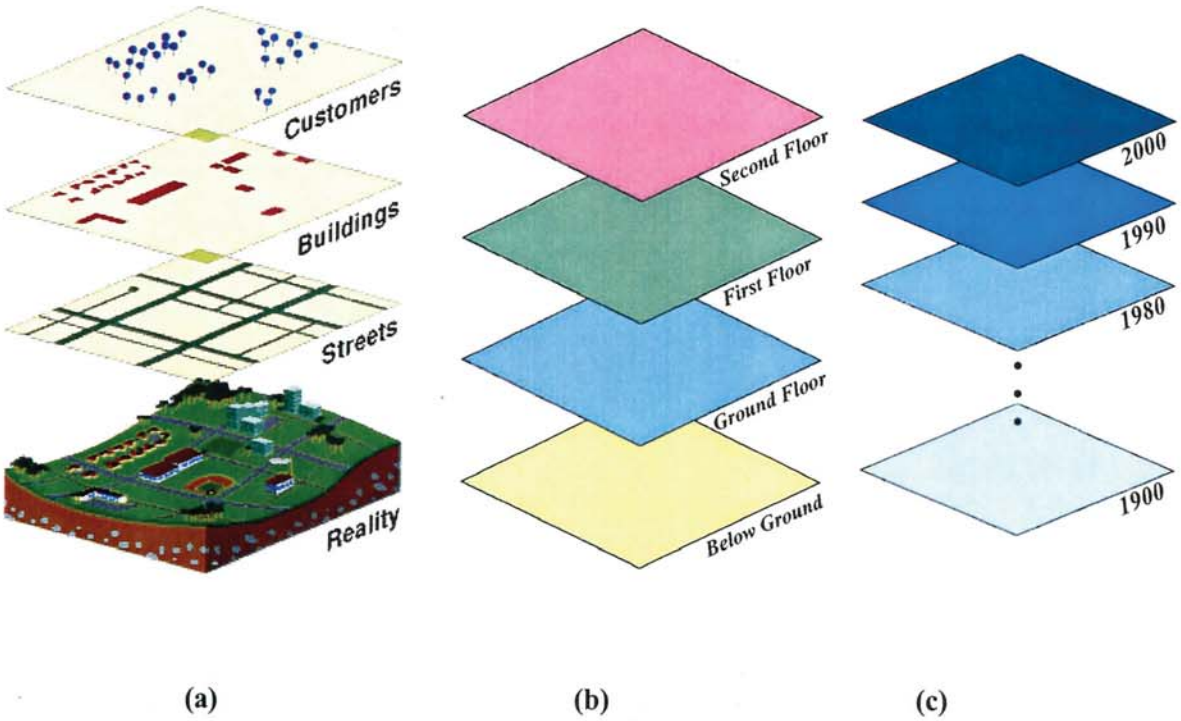


Figure 2.2. Layer Approach in GIS

Other than statistical and tabular data GIS deals with three spatial data model for representing geographic features. These spatial data models are:

A. The vector data model which closely resembles to the map model discussed earlier. In this model, points are recorded as single x, y coordinate, lines as a series of ordered x, y coordinates, and areas as a series of x, y coordinates defining line segments that enclose an area. Hence, points, lines, and polygons are represented as a list of coordinates instead of a picture or graph.

B. The raster data model which focuses on location rather than the representation of geographic features. The raster model is more like a photograph than a map. Similar to the way of storing pictures in computers, the raster data model is a regular grid of dots (cells or pixels)filled with values. Like the vector model, the raster model can represent discrete point, line and area features. A point feature is represented as a value in a single cell, a linear feature as a series of connected cells that portray length, an area feature as a group of connected cells portraying shapes. One frequent use of the raster model is in the georeferencing process where aerial photos and satellite images are in raster format.

C. The triangulated irregular network data model, called TIN, which is an alternative to the raster data model for representing continuous surfaces. It allows the generation and display of efficiently analyzed surfaces and terrains. The TIN model represents a surface as a series of linked triangles which can occur at irregular locations. The TIN model is fundamentally built from nodes that are connected to their nearest neighbor by edges according to a set of rules. Left and right topology is associated with the edges to identify adjacent triangles. These triangles are constructed based on the input of mass points and breaklines, which provide information and constraints about the surface. TIN modeling is mainly used for surface draping and 3D visualization.

2.4.1 GIS Data Sources

Data is brought into GIS through several tools and techniques:

A. *Keyboard data entry,* i.e.; data sets can be created with GIS software from tabular data that contains locational information. For example, point data layers could be generated from lists of customers addresses or a list containing x, y coordinates of electric poles locations. This process is named "geocoding" and used to create pin maps to show locations of various events by addresses; it is just the same as pushing stick pins on a wall map to mark the locations of all branch offices of a bank.

B. *Scanning,* i.e.; vector data can be created by scanning maps images, aligning it to a map coordinate system, correcting errors, removing unwanted features or noise, then extracting raster lines to vector data sets. This raster to vector conversion technique is based on the latest advances in the field of digital image processing

C. *Global Positioning system (GPS).* GPS technology is a key component in the GIS and remote sensing data acquisition fields. GPS-collected data can provide high accuracy for engineering surveys and GIS data collection. Differential GPSs allow for centimeter precision when the hardware and software are carefully chosen.

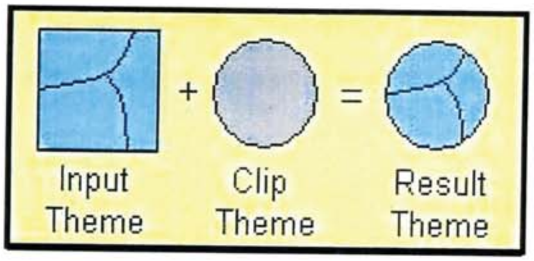
D. *Digtizing.* This technology has been around for a long time. GIS make use of two digitizing methods: **(i)** overhead digitizing where the drawing process is done on top of a registered scanned image displayed on a computer screen and **(ii)** digitizer tablet on top of which a paper map is fixed and drawn using the digitizer's puck. Although digitizing is time consuming and tedious, it is a popular choice because of its affordability and high accuracy.

E. *Conversion of .other vector formats.* One major GIS source of data is its ability to convert drawings produced by Computer Aided design (CAD) applications. Recent GIS software provides conversion utilities for MicroStation design files (.dgn) and two kinds of AutoCAD drawing files (dwg and .dxf).

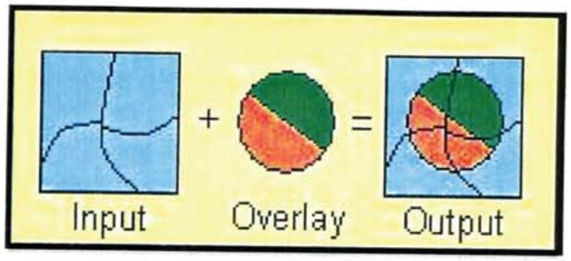
2.4.2 Spatial Functions

GIS relies on a set of advanced spatial functions and techniques to extract, manipulate, convert, analyze and visualize its spatial and tabular databases. In particular, GeoProcessing and dynamic segmentation represent two of these crucial techniques:

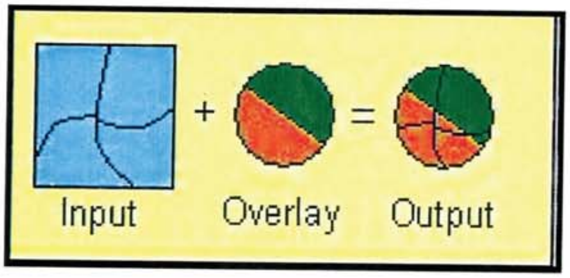
A. ***GeoProcessing.*** This technique consists of creating create new data based on the existing layers and their geographic features. In most cases the geometric properties of these features is altered while controlling some aspects of how their attributes data are handled. Hence, layers can be intersected, clipped, combined, dissolved, merged, and assigned data (spatial join) by location. Figure 2.3 shows different GeoProcessing operations:



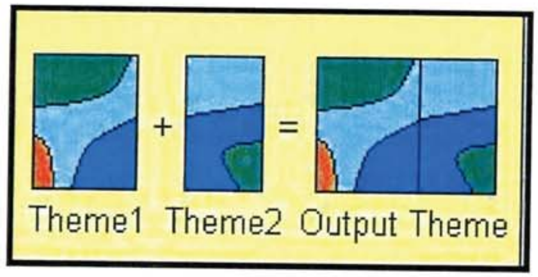
(e) Clip



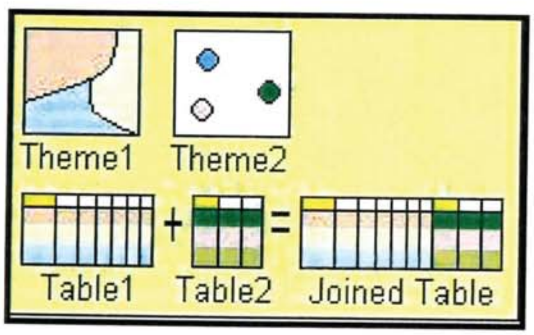
(f) Combine (Union)



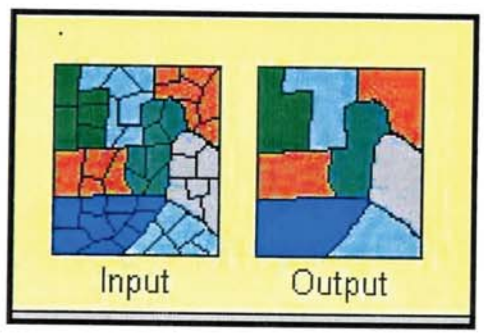
(d) Intersect



(e) Merge



(e) Spatial Join



(f) Dissolve

Figure 2.3. GeoProcessing

B. *Dynamic Segmentation.* This technique associates multiple sets of attributes to any portion of a linear feature. Dynamic segmentation models linear features using routes and events. A route contains measures and represents a linear feature, such as a city street, highway or river. The measures are used to locate data that describe parts of the route. Data along routes is modeled using events. .Several applications involving linear features can benefit from using the functionality provided by dynamic segmentation. For Example, the pavement quality in a highway application are known as events; pavement quality might be labeled according to route measures, poor from 0 to 150, good from 150 to 316 and fair from 316 (See Figure 2.4 below). With such model a planner would decide which portions to resurface first. Note that this techniques is used later in this thesis, for converting the street network into Street Delivery Units (SDUs)

Figure 2.4. Dynamic Segmentation

2.5 Integrating Heuristic Combinatorial Algorithms With GIS

GIS are becoming a crucial means for decision support in several domains. GIS treats very large databases and brings considerable impact to organize tasks including routing, scheduling, dispatching, transportation logistics and others. However, existing GIS solutions do not seem to take advantage of recent research results in other disciplines like combinatorial optimization (C & O), and operations research. The potential benefits of introducing heuristic optimization techniques in a GIS framework involves:

- Enhanced analysis for a wide range of geographical systems [1]. The data collected through surveys or other modern GIS related techniques like remote sensing, can be voluminous and distributed through time. Heuristic optimization techniques might perform detailed analysis to extract long-term trends and to quantify answers to the driving questions that motivated the construction of several geographical systems. Analysis can produce information in tabular, graphic, map, animation, and textual summary forms. GISs have greatly enhanced this phase by displaying data in forms that can be easily understood by diverse audiences.

- Improved geographical visualization where statistical summaries and plots are viewed at the same time with geographical data in the form of a map. This capability of simultaneously displaying and dynamically cross-referencing different views of the data, some geographical, others purely numerical, gives the developer as well as the user much more flexibility to interact with data. A simple example would be to select a record in a table and immediately get its related geographic feature (a street or a building) highlighted in the map view. In addition, the dynamic color graphics for the model output and the symbolic representation of major problem components, allow easy and immediate understanding of basic patterns and relationships. Rather than emphasizing the numerical results, symbolic representations and the visualization of complex patterns, support an intuitive understanding of complex systems behavior [15].

- Direct linkage of basic optimization techniques to the spatial query and selection functions of the GIS [1], and at the same time to perform proximity analysis and derive spatial properties or relationships, such as area, distance, adjacency, or direction along a network. One could, for example, compute the average area of the parcels affected by a planned street or list all customers living at 500 meters from a drug store.

In light of the above mentioned benefits, the nature of the mail distribution problem in urban areas, is strictly tied up to both GIS as well as optimization techniques. In fact, two things are essential for the delivery service business to persist, a street network to reach customers (GIS) and the minimization of delivery costs (Optimization). Thus, the districts determination problem for mail distribution is one of the most engaging real world problems that would benefit from the flexible and effective environment of building application in current GIS systems once optimization techniques are integrated with spatial databases.

CHAPTER III

CASE STUDY

This chapter provides a comprehensive description of the mail distribution problem in urban areas. It focuses on the large number, non-uniform aspect and complexity of the involved parameters. We review the work of Feo and Resende [4] on greedy Randomized Adaptive Search procedures (GRASP). These techniques will form the basis of the improvements proposed in chapter IV.

3.1 Overview of Mail Distribution in Urban Areas

The data used in mail distribution comprises both spatial (GIS vector format) and non-spatial (statistical) data. Spatial data is a set of city maps, where a map is represented as a collection of street segments. Buildings are placed with respect to this street network. The side definition is important since distinct mailmen may work on different sides of the same street. Street segments are grouped into districts, each of which representing a mail delivery area being under the responsibility of one mailman. Districts determination may partition a given street into several segments, each of which allocated to one mailman. Districts are grouped into distribution zones that are aggregated into main distribution units usually being cities.

Textual data available from post offices, mainly consist of temporal statistics on mail volume and mailmen average time. One complicating factor in the work of De Souza, Medeiros and Pereira [3], is that statistics are usually defined in terms of *Street Delivery Unit* (SDU) that is internal to post offices, and which is normally not compatible with the spatial (street segments) database. Ideally a SDU measures 600 meters. One statistical record, for example, provides information like: " In SDU S, the delivery time is 25min from November to January". This means that the system must perform a nontrivial spatial data conversion in order to link statistics and street maps. The manual definition of the streets for the districts in a single distribution zone in a city like Sao Paolo (population 12 million), may take one week of work for a team of 3 people working full time (120 man hours) [3]. Thus, *districts determination* is a priority goal for automation.

As stated earlier, districts determination corresponds to specifying a (connected) street network which will be traversed by one mailman. Algorithms for this problem, should satisfy two main tasks: **(i)** to minimize the number of mailmen working in a distribution zone; and **(ii)** to evenly balance, among these mailmen the daily workload of mail delivery. The workload is defined as having an upper limit of 480 minutes per person/day (8 hours). The average delivery time for all SDUs should be estimated. However, in order to obtain this statistic, several factors such as topography of a region, different season and type of mail must be computed for each SDU over a period of time.

3.2 Problem Parameters

A wide variation of parameters is mainly associated with mail distribution in a given urban area. Some of these parameters are:

- Non-Uniform demand for mail services, i.e.; there is no direct correlation between population density and mail volume. For example, commercial zones captivate much higher concentration of mail recipients despite that they occupy a small spatial area compared to large and densely populated areas.

- Spatio-temporal fluctuation of mail volume. Residential areas often show a marked decrease in correspondence volume during summer vacations, whereas summer resorts present an increase in volume during the same period, causing workload imbalance.

- Spatio-temporal evolution of street topology and population density. The marked migration from villages to big cities causes a crucial impact in postal services. New streets might be created on a monthly basis, which requires constant updates and replanning of mail distribution tasks.

- Social Issues. Planned schedule might fail because of mailmen social habits, for example, a given mailman might stop several times for a coffee in his favorite coffee shop.

- Topography of a region. For instance, as reported in [3], in residential areas with little incidence of slopes, the average speed to deliver mail on foot is 5.74 km/h, whereas commercial areas in hilly zones average 4.41 km/h.

- Type of a mail. The difference in weight between regular letters and express packages, is another factor that causes disproportional workload between SDUs.

The above raised issues increase the difficulty in maintaining updated versions of databases at both spatial and relational levels. Moreover, each of these factors deeply affects the process of computing the average delivery time for each SDU. However, the focus in this thesis is not to discuss and implement various methods for gathering information and statistical data to help solving the districts determination problem, but to show the benefits of integrating heuristic combinatotrial algorithms with GIS in solving the districts determination problem once all influencing parameters are well defined. Hence, for implementation purposes, the workload factor for each SDU is computed in a way that approximates the non-uniform aspects of different SDUs.

3.3 Graph Representation

To tackle the problem of districts determination, this problem is modeled as graph partitioning problem where both spatial and statistical data are used. The two fundamental constraints involved in defining a district are: mailmen daily load (non-spatial) and streets connectivity (spatial).

Given an instance of the districts determination problem, an undirected graph **G** = (**V**, **E**) is constructed as follows:

1- Transform a street map into a SDU map.

2- Associate each SDU to a separate node in V.

3- Assign to each node $\mathbf{u} \in \mathbf{V}$ a weight $\mathbf{t_u}$ equal to the average delivery time of the corresponding SDU.

4- Add an edge from a vertex **u** to a vertex **v,** iff there exists a path from the SDU corresponding to node **u** to that corresponding to node **v** without passing through any intermediate street of the city.

The resultant graph would look something like this:

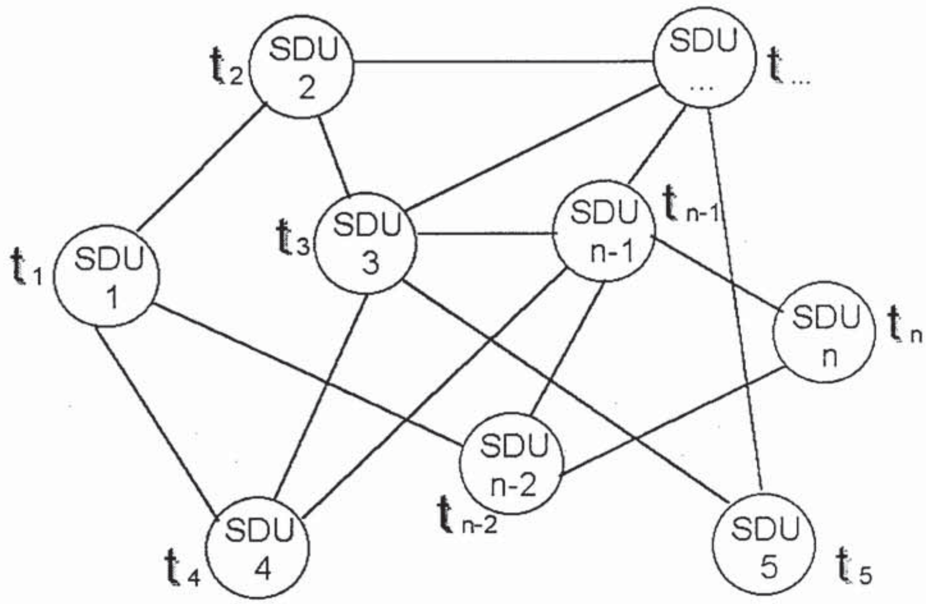


Figure 3.1. Graph Representation Model

Next, let **W** be the maximum workload allowed for a mailman in a day (i.e., 480 minutes).

Consider a partition $(\mathbf{V_1}, \mathbf{V_2}, ..., \mathbf{V_k})$ of the nodes in **V** such that:

(ii) The sub-graph induced by $\mathbf{V_j}$ is connected for all $\mathbf{j} \in \{\mathbf{1},...,\mathbf{k}\}$.

(i) $\sum_{u \in V_j} \mathbf{t_u} \leq \mathbf{W}$ for all $\mathbf{j} \in \{\mathbf{1},...,\mathbf{k}\}$

Denote a partition satisfying **(i)** and **(ii)** by *W-Connected* partition of **G.** Clearly, every W-Connected partition of G corresponds to a valid district partition of the considered distribution zone [3].

It is theoretically possible to enumerate all combinations of solutions (W-connect partitions) and evaluate each with respect to the stated objectives. However, from a practical perspective, it is unfeasible to follow such a strategy of complete enumeration because the number of combinations grows exponentially with the size of the problem namely SDUs number.

Unfortunately, finding a minimum size W-connected partition of **G** is computationally intractable which preclude the use of exact algorithms. The NP-completeness of the problem can be shown from a simple reduction of the bin packing problem [5]. Therefore, heuristic algorithms are often used to tackle this problem.

3.4 Heuristic Solutions And GRASP Algorithms

this section considers two sorts of heuristics: *construction heuristics*, where the solution is built from scratch, building one district at a time until all nodes have been assigned; and *local improvement (search) heuristics* [16], which assume the existence of an initial partition of **V** , and performs iterative visits on several altered solutions. Both of these heuristics are used in [3] in solving the districts determination problem, mainly two *simple greedy algorithms* named H1 and H2; and two **GRASP,** *Greedy Randomized Adaptive Search Procedure* [4], named H3 and H4.

3.4.1 Algorithms H1 & H2

Given a graph corresponding to a districts determination problem, both H1 and H2 start from a root ($\mathbf{R}_i$) belonging to district ($\mathbf{V}_i$) and search the network for a node **v** (SDU) with the smallest number of adjacent nodes which are not assigned; such that the sum of node weights in $\mathbf{V}_i$ plus $\mathbf{t}_v$ does not exceed **W**. Then **v** is added to $\mathbf{V}_{i;}$; otherwise**,** a new district is initiated. This process is repeated until all nodes **v** in **V** are assigned. H1 differs from H2 in choosing the root for a new district. In H1 the choice is made randomly; however, in H2 the root is chosen by picking, among the not assigned nodes, the node with the smallest number of not assigned adjacent nodes [3].

3.4.2 Algorithms H3 & H4

H3 and H4 are based on GRASP described in section 2.2.4. H3 and H4 in their first step, use a randomized version of H1 and H2 respectively in the sense that they randomize their choice of node **v** among a set of **K** best elements. The second step is a local improvement search procedure which starts from the solution generated in the first step. These two steps are then repeated **L** times and the heuristic returns the best solution found during this process [3][8].

CHAPTER IV

IMPLEMENTATION AND IMPROVEMENT

This chapter describes a GIS model to represent the problem of districts determination. This GIS model constitutes the base for all implementations and improvements done in this thesis. The procedures of implementing and testing the four heuristic algorithms are then described along with the data preparation phase. The last section describes more improvements and shows empirical studies done on the local search phase of GRASP algorithms.

4-1 GIS Model Versus Graph Partitioning Model

A GIS model is constructed to provide a more practical representation of the mail distribution problem. Similar to the graph partitioning usually adopted in the literature, the GIS model is built using the following four steps:

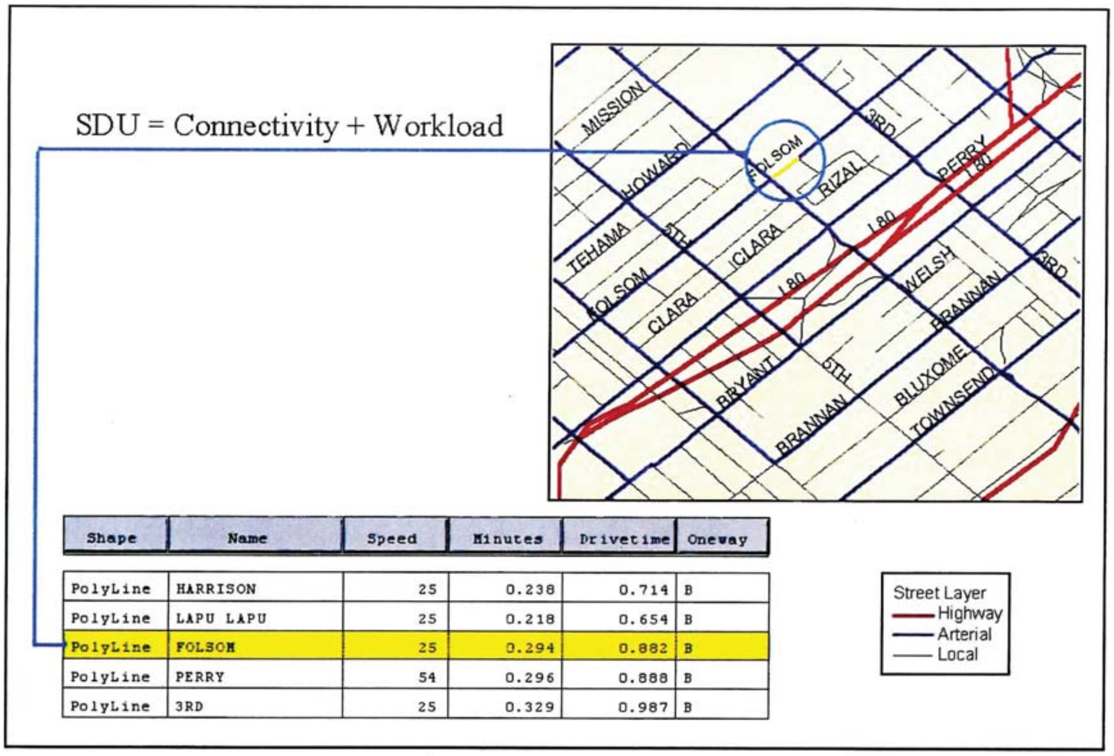
1- Transform the street map into a SDU map.

2- Associate every SDU to a street segment in the transformed street network layer.

3- Assign to each street segment a weight $\mathbf{t_u}$ equal to the average delivery time of the corresponding SDU.

4- By the nature of a GIS layer, the connectivity between street segments is embedded within the layer internal structure. Hence, there is no need to worry about representing the edges connecting two SDUs since they are no longer represented as nodes, but as edges.

Figure 4.1 shows a portion of the street network of St. Francisco. The circled segment shows an SDU that represents a part of the Folsom Street. Notice that this network layer of SDUs embeds a database inside it, where the connectivity of the SDUs can be inferred from the shape field shown in the table, and the SDU workload can be computed from some fields involved in the database like speed, drive time, etc.



Shape	Name	Speed	Minutes	Drivetime	Oneway
PolyLine	HARRISON	25	0.238	0.714	B
PolyLine	LAPU LAPU	25	0.218	0.654	B
PolyLine	FOLSOM	25	0.294	0.882	B
PolyLine	PERRY	54	0.296	0.888	B
PolyLine	3RD	25	0.329	0.987	B

Figure 4.1. Problem Representation in GIS Model

Clearly, working with the GIS model without having to build a graph partitioning model, is more practical and simpler for implementing heuristic algorithms. In fact, a crucial power of the GIS technology springs from its simplified yet very rich representation of the real world features. The ability to use the GIS visualization model as is, in the process of implementing any solution paradigm, omits the burden of constructing graph models, and then reconstructing GIS maps for better illustration and visualization of the solution. Moreover, the above proposed GIS model combines the graph concept of nodes and edges into one GIS entity (street segment), which facilitate the extraction and manipulation of various SDU attributes and constraints, e.g. SDU length, degree of connectivity, etc. Thus, the GIS model adds more flexibility to the development and implementation of heuristic algorithms especially through a set of built-in functions that control its spatial database [15][11].

4.2 Implementation of Heuristic Algorithms and Empirical Observation

In order to test and compare the quality of solutions generated by the four heuristic optimization techniques H1, H2, H3 and H4 described in section 3.4, a GIS based simulation prototype is built using ESRI ArcView software (See Figure 4.1). For demonstration purposes, and in order to transform and construct the GIS model, we have used the US sample data and shape files that come with ArcView. All heuristic optimization algorithms and data transformation procedures where coded in Avenue, the object oriented programming language of ArcView. In a given distribution zone, the solution of the districts determination problem is displayed as a map where each district is assigned a different color.

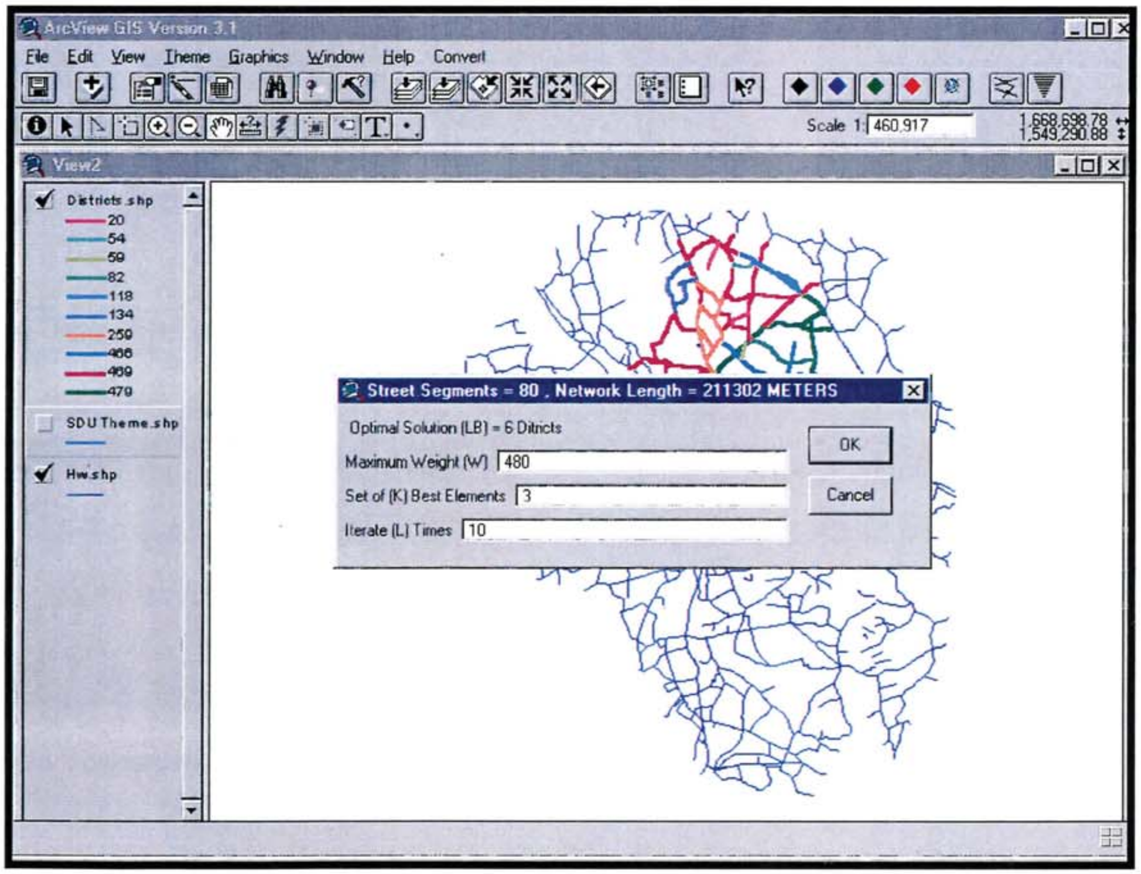


Figure 4.1. Snapshot of the Prototype Interface

Before describing the implementation of the four heuristic algorithms described in the section 3.4., and in order to have a better insight of the quality of the solutions they would generate, we should ideally compare those solutions with the solution generated by an optimal algorithm. But, since the problem is NP-complete, an optimal solution is out of reach for large instances of the problem. The other alternative is to make the comparison against some known lower bounds on the number of mailmen needed in a single distribution zone. A trivial lower bound here is given by rounding up the quotient between the sum of all SDUs workload and W- the normal maximum load allowed for one mailman per day (480 minutes). Thus, if LB denotes this lower bound, the following formula holds:

$$\mathbf{LB} = \frac{\sum \text{all SDU workload}}{\mathbf{W}}$$

In other words, this expression states that at least LB mailmen (districts) are necessary to guarantee the distribution in the considered region. Hence, the closer the number of districts to LB, the better the generated solution. In addition, knowing the lower bound for this problem, allows the user to easily indicate when a solution found hits the optimal number of districts which might come true in some distribution zones.

The other important aspect that must be taken into consideration is that this prototype considers a worst case scenario in solving the districts determination problem. In other words, we assume that every SDU in a distribution zone contains at least one customer that the post office should deliver mail to. Therefore, every SDU segment belonging to the street network of a given city will be allocated to one district.

Besides generating solutions to the districts determination problem, the prototype allows user interaction in the following ways:

First, it transforms a GIS street layer into a GIS SDU layer. This is implemented through a separate Avenue module to resolve the complicated issue resulting from the mismatch between the SDU spatial unit and the street network in the spatial database. In this module, the user defines a common length for SDUs; a dynamic segmentation technique is then used to explode or combine streets in order to match corresponding SDUs. Note that the SDU *Polyline* layer should be transformed into a.*PolylineM* layer. before using dynamic segmentation [19]

Second, it assigns workload to every SDU in the SDU Layer. Since several parameters involved in defining the SDU workload are subject to frequent changes, we choose in this implementation to randomly assign workloads to SDUs. Hence, a SDU workload is defined in terms of the time in minutes needed for delivering mail to one or more customers located on that SDU. In this module, the user may define the lower and upper bounds for workloads. Naturally, SDU workloads should fluctuate between a minimum of 1 minute and a maximum of 480 minutes [3].

Third, it chooses one of heuristic methods H1, H2, H3, or H4 to solve the problem. Both H1 and H2 are based on Greedy Random whose pseudocode is shown in Figure (4.2). H1 and H2 differ in step 3 of Greedy Random. The Choose-Root function for H1 is a random choice of any non-assigned SDU, while Choose-Root for H2 is a routine that searches the set of non-assigned SDUs, and picks the SDU with the least connectivity degree. In other words, it picks the SDU with the smallest number of not assigned adjacent SDUs. A pseudocode of Choose-Root for H2 is shown in Figure (4.3).

1. Initialize all SDU segments in the street layer as not assigned, and set $i \leftarrow 0$.
2. Create a district layer $\mathbf{D_i}$ with no segments (empty layer)
3. $r \leftarrow$ Choose_Root($\mathbf{D_i}$), mark the SDU corresponding to r as assigned in the street layer and add it to $\mathbf{D_i}$.
4. Select all SDUs that are spatially connected to the root r.

   Clear all assigned SDUs from the previous selection.

   Clear all SDUs whose weight $\mathbf{t_u}$ exceeds $\mathbf{W}$ when added to the sum of SDUs Weights in $\mathbf{D_i}$.

   If (Selection is empty) go to step 6,.Else select from remaining not assigned SDUs the SDU that has the smallest number of adjacent not assigned SDUs.
5. $D_i \leftarrow D_i \cup \{$Chosen SDU$\}$, mark the chosen SDU as assigned in the street layer and repeat the previous step.
6. If there are more not assigned SDU segments in the street layer, set $i \leftarrow i+1$ and go to step 2

Figure 4.2. Greedy Random

1. LeastCD $\leftarrow \infty$ /* Least Connectivity Degree */
2. Repeat for all not assigned SDUs

   2.1 Select all segments adjacent to the current SDU

   2.2 Clear all assigned SDUs from previous selection

   2.3 ConnectivityDegree $\leftarrow$ Number of SDUs remaining in the selection

   2.4 if ConnectivityDegree < LeastCD, LeastCD.$\leftarrow$ ConnectivityDegree, BestRootCandidate $\leftarrow$ Current SDU
3. Return BestRootCandidate

Figure 4.3. Choose-Root for H2

Figures 4.4 and 4.5 below, show two solutions generated by H1 and H2 respectively. The distribution zone chosen for demonstration contains a street network of about 210 kilometers in length. This street network was transformed into a layer of 100 SDU segments where the sum of their assigned workloads is equal to 2775 minutes. If we apply the lower bound formula described at the beginning of this section we get: LB = 6. This means that the best solution for this instance of the problem can't generate fewer than 6 districts. Note that this chosen distribution zone is used for the testing of all other heuristics.

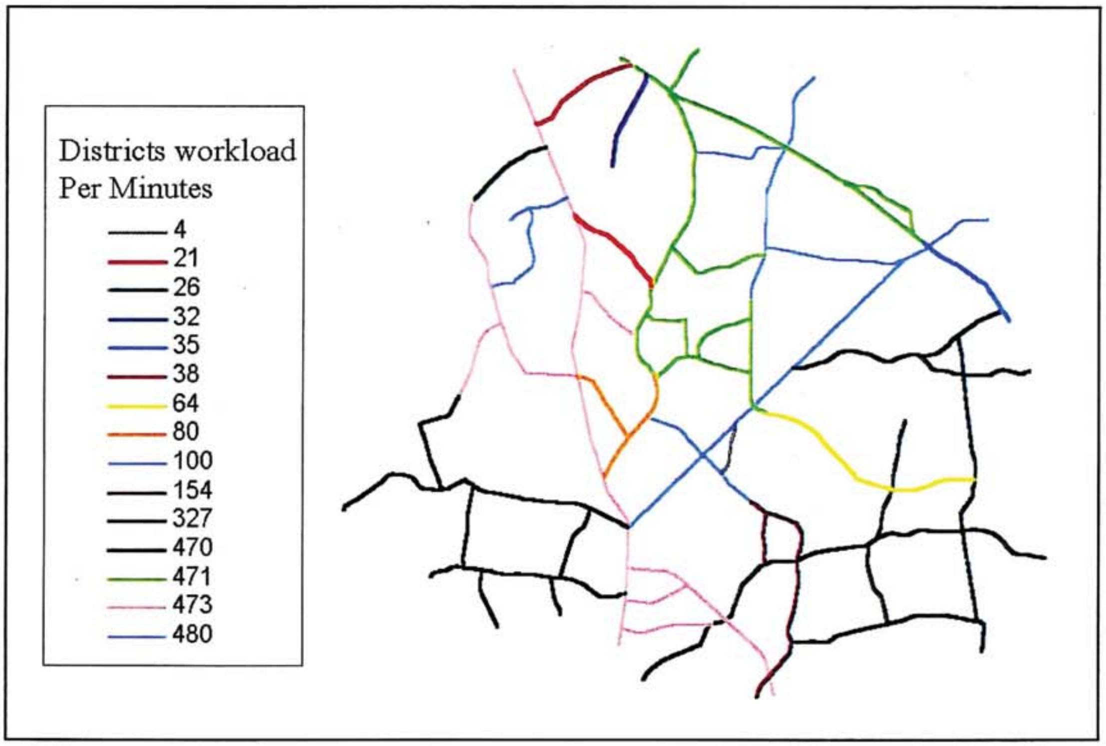


Figure 4.4. Districts Generated by H1 (15 districts)

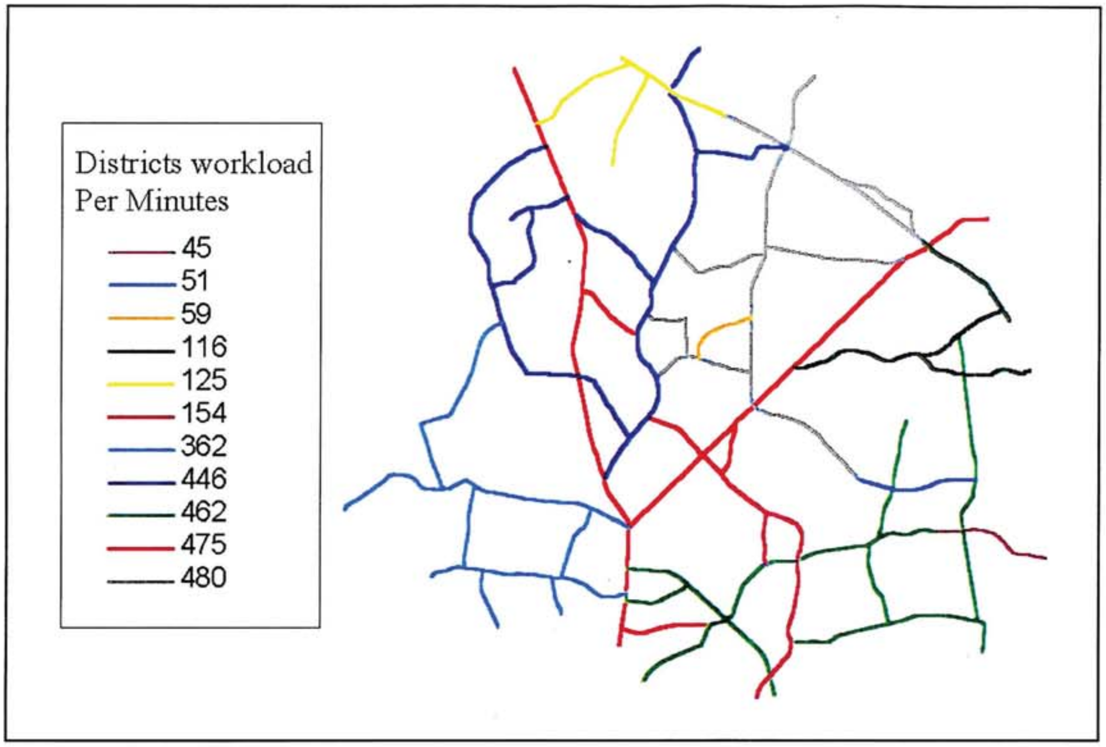


Figure 4.5. Districts Generated by H2 (11 districts)

We notice that heuristic H2 generates a better solution than H1 in terms of minimizing the number of districts. However, both solutions suffer from the fact that the generated districts are not balanced (For e.g. in the case of H2, minimum workload = 45 minutes, and maximum workload = 480 minutes) .

As described in [3] heuristics H3 and H4, are improvements of heuristics H1 and H2 respectively. This improvement is in terms of minimizing the number of districts generated as to reconstruct the discrepancy in workload balance generated by H1 and H2. To do so, and as mentioned in section 3.4, H3 and H4 make use of a GRASP algorithm. A modified GRASP pseudocode for this minimization problem is shown in Figure 4.6 where our own measure of the error factor involved **c(S)** is introduced.

$$\mathbf{c(S) = \sum (W\text{-}W_i)^2}$$

where **W** is the maximum workload allowed for a mailmen per day (480 minutes), and $\mathbf{W_i}$ is the workload of district $\mathbf{D_i}$ generated by the solution **S.**

1. cBest $\leftarrow \infty$
2. Repeat **L** Times:
   - 2.1 **S** $\leftarrow$ Greedy_Random (GIS_Model)
   - 2.2 **S** $\leftarrow$ Local _Search (GIS-Model)
   - 2.3 if **c(S)** < cBest, cBest $\leftarrow$ **c(S)** and Sbest $\leftarrow$ **S**
3. Return Sbest

Figure 4.6. GRASP Algorithm For The Districts Determination Problem

Greedy_Random used in H3 and H4 are similar to the ones used in.H1 and H2 respectively with a variation in their fourth step where the choice of the next SDU to be added to district $\mathbf{D_i}$ is done by a random pick from a set of K SDUs that have the smallest number of adjacent not assigned SDUs. This operation is illustrated by the pseudocode shown in Figure 4.7.

1. ConnectivityList ← EmptyList( )
2. Repeat for all not assigned SDUs
    2.1 CD ← connectivity degree of current SDU in the set of not assigned SDUs
    2.2 ConnectivityList ← ConnnectivityList ∪ { (CD,SDU) }
3 End
4 Sort the ConnectivityList in ascending order according to the CD. Note that each element in the ConnectivityList is a couple (CD, SDU) containing the SDU polyline and its connectivity degree.
5. j ← Random number between 1 and K
6. ChosenSDU ← The SDU residing in the jth element of the ConnectivityList
7 Return ChosenSDU

Figure 4.7. Choosing a SDU in H3 and H4

The workload of some districts in solution **S** generated by Greedy_Random are equal or approximate **W** while others are far below **W.** The local search phase of grasp aims to balance the workload between those districts. In our implementation, the Local_Search module classifies the districts into two categories: Fat Districts and Thin Districts denoted respectively by FD and TD. FD represents a district with high workload while TD represents a district with low workload. It is a simple common sense to get FDs loose weight (workload) for the sake of TDs. Hence, SDU segments are migrated from FDs to TDs. This. exchange of SDUs between FDs and TDs is done with a great caution so as not to fall back into the initial situation where some FDs become TDs and vice versa. To avoid such a trap, the average workload of the districts in the neighborhood of a thin district $TD_{i,}$ denoted by $\mathbf{W_i'}$ is calculated and SDUs are moved into TD_i so as not to exceed $\mathbf{W_i'}$ or break the connectivity constraint of an adjacent FD. A pseudocode of the local search phase is shown in Figure 4.8.

1. Sort the district layer according to workload field.
2. Classify districts as **FD** or **TD**.
3. Repeat for all **TD** districts.
   - 3.1 **i** ← district ID, $\mathbf{W_i}$ ← district workload
   - 3.2 Select all **FDs** in the district layer that are adjacent to $\mathbf{TD_i}$.
   - 3.3 Let **n** and **SW** be the number of selected districts and the summation of their workload, respectively.
   - 3.4 $\mathbf{W_i}'$ ← **SW/n** (average workload between $\mathbf{TD_i}$ adjacent districts)
   - 3.5 Select all SDU segments in the street layer that are adjacent to $\mathbf{TD_i}$ and belong only to **FD**s.
   - 3.6 Repeat for all selected SDUs in the street layer
     - 3.6.1 $\mathbf{W_i}$ ← $\mathbf{W_i}$ + current SDU workload
     - 3.6.2 **dc** ← degree of connectivity of current SDU
     - 3.6.3 if ($\mathbf{W_i}$ <= **W'**) and (**dc** = 2) then move current SDU from its current **FD** to $\mathbf{TD_i}$
4. **S** ← update district layer

Figure 4.8. Local Search For The Districts Determination Problem

In implementing step 2 in the above figure we have tested two different statistics: mode and mean. However, for most of the tested instances the results in both cases were very close . Hence, in demonstrating H3 and H4, the mode statistic is used.

Figures 4.9 and 4.10 show two samples of the solution generated respectively by H3 and H4. Note that due to the iterative and randomization aspects of the GRASP algorithms, heuristics H3 and H4 were able to improve the distribution of the workload between districts and to decrease the number of districts generated by H1 and H2 respectively, from 15 districts in H1 to 9 districts in H3, and from 11 districts in H2 to 8 districts in H4. The number of iterations for both H3 and H4 in this run was 10, and the local search phase was recursively called for 5 times in each iteration.

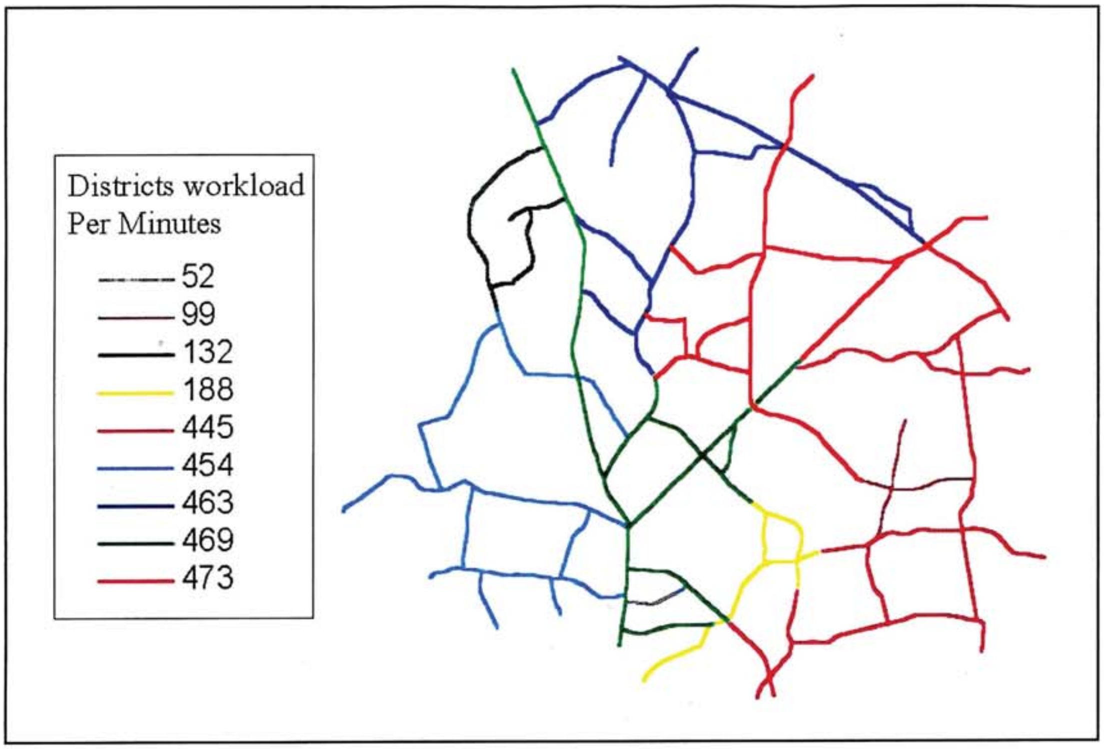


Figure 4.9. Districts Generated by H3 (9 districts)

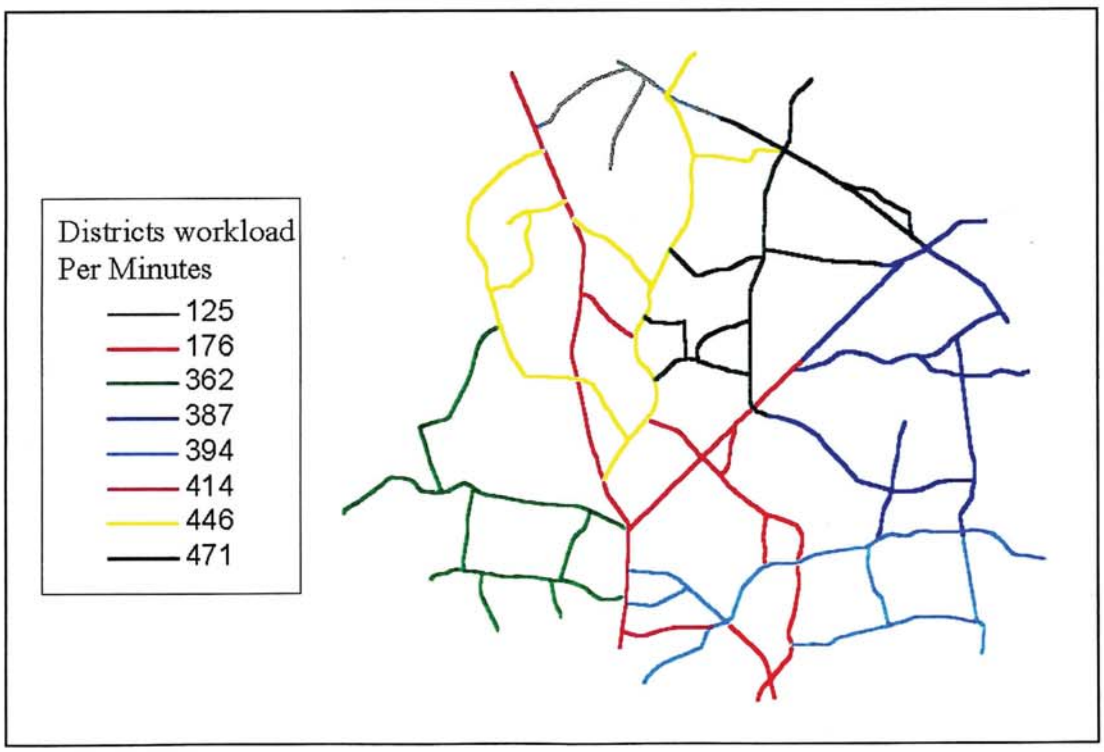


Figure 4.10. Districts Generated by H4 (8 districts)

4.3 Revision of Local Search Phase

Although most of the solutions generated by H3 and H4 were better than those generated by H1 and H2, we were seeking better improvement in the sense of approximating the optimal number of district with enhanced workload balanced while keeping a practical number of iterations. To achieve such a result, heuristic H5 is introduced as an improvement on the local search of H3 and H4. In fact, the role of the local search phase in H3 and H4 was restricted to balancing the workload between districts, without influencing the resulting number of districts. H5 introduces a simple process of district fusion in the local search phase. This process is initiated after step 4 and before the next iteration of the local search. The role of the. district fusion process is to combine any two adjacent districts $\mathbf{D_i}$ and $\mathbf{D_j}$ so that the sum of $\mathbf{W_i}$ and $\mathbf{W_j}$ does not exceed $\mathbf{W}$. The pseudo code for the district fusion process is shown in Figure 4.11 below.

1. Initialize all districts in district layer as not visited, set $\mathbf{i} \leftarrow 0$
2. Select all adjacent districts to $\mathbf{D_i}$, mark $\mathbf{D_i}$ as visited, set $\mathbf{j} \leftarrow 0$
3. If (no more $\mathbf{D_j}$ in selection) then Go to step 5
4. If $\mathbf{W_i} + \mathbf{W_j} <= \mathbf{W}$ then combine $\mathbf{D_i}$ and $\mathbf{D_j}$, update district layer, go to step 1

   Else clear $\mathbf{D_j}$ from Selection, set $\mathbf{j} \leftarrow \mathbf{j} + 1$, go to step 3.
5. If (non-visited $\mathbf{D_i}$ continue to exist) then $\mathbf{i} \leftarrow \mathbf{i} + 1$, go to step 2
6. S $\leftarrow$ Updated district layer

Figure 4.11. District Fusion Process

Figures 4.12 and 4.13 shows two solutions generated by H5 for the same distribution zone and number of iterations (10) used in previous heuristics. These two solutions have generated 7 districts compared to 9 districts in H3 and 8 districts in H4. In addition, the workload between the resulting districts was more balanced. Note that the workload balance in Figure 4.13 where the mean is used for classifying the districts as FDs and TDs, is slightly improved compared to Figure 4.12 where the mode statistic is applied.

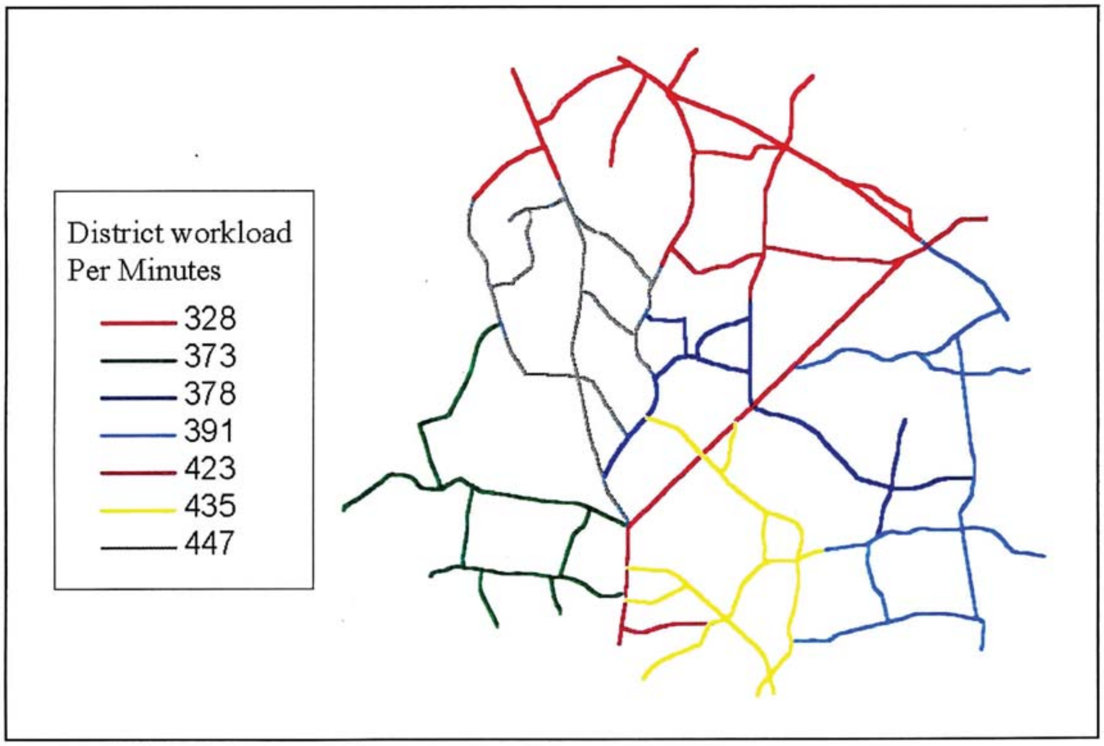


Figure 4.12. Districts Generated by H5 (7 Districts)

Figure 4.13. Districts Generated by H5 (7 Districts)

CHAPTER VI

CONCLUSION AND FUTURE WORK

This thesis described the benefits resulting from bridging the gap between GISs and heuristic combinatorial optimization techniques. This integration shows several advantages in solving large and complex spatial problems. Most of these real world problems are NP-complete; therefore, exact algorithms would fail to generate solutions in a reasonable amount of time. Hence, heuristic combinatorial techniques are used as an alternative in addressing the special needs of such GIS related problems.

In particular, this thesis addressed the problem of districts determination for mail distribution in urban areas where the street network of a distribution zone in a given city must be partitioned into n districts under the responsibility of n mailmen. Two main goals should be accomplished in order to provide the post office with feasible , effective and economical solution: **(i)** to minimize the number of mailmen, and **(ii)** to evenly balance the workload between districts. To tackle this problem where a large number of parameters and constraints are involved, GIS is integrated with five heuristic combinatorial optimization algorithms based on a modern technique named GRASP.

The prototype built in this thesis was based on a GIS model where the graph concept of nodes and edges is combined into one GIS entity as the SDU segment. This new representation helped in testing, comparing and improving the results generated by the GRASP based heuristics.

Besides the time saving compared to the manual work that is still applied in some post offices, several advantages could be highlighted:

The prototype developed in this thesis could be used as a base for future development and research, not only in tackling the mail distribution problem, but in solving other spatial NP-complete problems. This is mainly due to the open system paradigm of existing GIS software where modularization is easily maintained. This modularization aspect allow, with a little programming efforts, the reuse and extension of the implemented heuristic algorithms, and data conversion procedures in other systems.

Another important advantage resides in enhancing the visualization and simulation aspects of solving real world problems. The GIS interface of the developed prototype allows the user to interactively communicate with the heuristic solution techniques, modify the parameter setting and study the effect on the problem in a visual manner.

Nevertheless, some of the solution generated by the prototype suffer from the fact that they may have inadequate geometrical district formats. In other words, one district might consist of a straight line of SDUs traversing the whole distribution zone while another district might be occupying a circular portion of the street network in a corner of the distribution zone. One possible solution that should be investigated in the future is the use of a bounding rectangle as an additional restriction in the districts determination, thereby increasing the spatial constraints imposed on heuristic algorithms

Finally, in order to widen the range of applications and bring considerable improvement on combinatorial intractable problems, further research should target the integration of other heuristic optimization methodologies with GISs. In particular, we would like to investigate simulated annealing, tabu search and genetic algorithms which share with GRASP the fundamental heuristic concepts that can be used to classify their operations and impacts.

REFERENCES

[1] T. C. Bailey, "A review of statistical spatial analysis in geographical information systems", 1994.

[2] B. De Backer, V. Furnon, P. Prosser, P. Kilby, and P. Shaw, "Local search in constraint programming: Applicaton to the vehicle routing problem", Presented at the CP-97 Workshop on Industrial Constraint-based Scheduling, August 1997.

[3] C. C. De Souza, C. B. Medeinos, and R. S. Pereira, "Integrating heuristics and spatial databases: a case study", December 1996.

[4] T. A. Feo, and M. G. C. Resende, "Greedy Randomized Adaptive Search Procedures", Journal of Global Optimization, 1:1-27, 1995.

[5] M. R. Garey and D. S. Jonshon. "Computer and Intractability: A guide to the NP-Completeness. Freeman, San Francisco, 1979.

[6] F. Glover, and M. Laguna, "Tabu Search", Kluwer Academic Publishers, 1997.

[7] D. E. Goldberg, "Genetic algorithms in search, optimization and machine learning". Addison-Wesley, 1989.

[8] P. Kilby, P. Prosser, and P. Shaw, "Guided local search for the vehicle routing problem", Proceedings of the 2nd International Conference on Metaheuristics, July 1997.

[9] S. Kirkpatrick, "Optimization by Simulated Annealing: Quantitative Studies", Journal of Statistical Physics, 34:975-986, 1984.

[10] G. Kontoravdis, and J. F. Bard, "A GRASP for the vehicle routing problem with time windows", ORSA J. on Computing, 7:10-23, 1995.

[11] C. H. Papadimitriou, and K. Stieglitz, "Combinatorial Optimization: Algorithms and Complexity", Prentice-Hall, 1982.

[12] T. Pierce, and W. Macmillan, "Optimization modeling in a GIS framework: the problem of political redistricting", 1994.

[13] P. Prosser and P. Shaw, "Study of greedy search with multiple improvement heuristics for vehicle routing problems", Research report 96/201, Department of Computer Science, University of Strathclyde, December 1996.

[14] M. G. C. Resende, "Greedy Randomized Adaptive Search Procedures (GRASP)", December 22, 1998.

[15] E. Rolland, and R. Gupta, "New linkages between GIS and Combinatorial Optimization", 1996.

[16] P. Shaw, "Using constraint programming and local search methods to solve vehicle routing problems", Submitted to CP '98, 1998.

[17] D. Thompson, and R. Laurini, "Fundamentals of spatial information systems", Academic Press Limited, 1994.

[18] C. Wallace, "A Survey of Randomized Algorithms", CS 391, June 17, 1998.

[19] ESRI, ArcView Online Documentation

[20] S. Zar Dijk, D. Thierens, M. De Berg, "Robust Genetic Algorithms for High Quality Map Labeling", December 9, 1998.