University of Alberta

Political Districting: A Tabu Search Algorithm and Geographical Interfaces

by

Burçin Bozkaya

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

in

Management Science

Faculty of Business

Edmonton, Alberta
Spring 1999

0-612-39508-1

University of Alberta

Library Release Form

Name of Author: Burçin Bozkaya

Title of Thesis: Political Districting: A Tabu Search Algorithm and Geographical Interfaces

Degree: Doctor of Philosophy

Year this Degree Granted: 1999

Burçin Bozkaya

İpekçilik Cad.
Çatal Sok. 3/10
Bursa, Turkey 16360

Date: November 25, 1998

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled *Political Districting: A Tabu Search Algorithm and Geographical Interfaces* submitted by Burçin Bozkaya in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Management Science.

Erhan Erkut

Dr. Erhan Erkut (Supervisor)

T. Daniel

Dr. Terry Daniel

Ármann Ingólfsson

Dr. Ármann Ingólfsson

Dr. M. John Hodgson

J. Paul Johnston

Dr. J. Paul Johnston

Dr. Gilbert Laporte

Dr. Rajan Batta
SUNY at Buffalo

Date: 10 Nov. 98

To my dear Gökçe,
and my little son Ozanay

Abstract

The political districting problem deals with dividing a geographical area into a predetermined number of single-member electoral districts. As the results of a new census become available, the existing districts may need to be revised, so that they still conform to constitutional criteria. Almost all of the past studies consider a basic set of criteria only (i.e., population equality, compactness, and contiguity) and most of them are unable to handle large-scale districting problems. In this dissertation, a Tabu Search algorithm is developed for solving the political districting problem. Both basic criteria and political criteria are considered. The problem is recognized as a multi-criteria optimization problem and consequently the objective function is composed of multiple weighted terms. The proposed Tabu Search algorithm is also integrated with a meta-heuristic technique known as Probabilistic Diversification and Intensificiation. The algorithm is tested using enumeration area data for the City of Edmonton and the City of Montreal. The district plans that are produced with the Tabu Search algorithm dominate the ones that are currently in use, based on the criteria considered. The algorithm generates district plans that are more compact than the existing ones, and that also maintain integrity of communities, and similarity to the existing plan to a reasonable extent.

Keywords: political districting, electoral districting, redistricting, tabu search, GIS.

Acknowledgements

I would like to thank my supervisor Dr. Erhan Erkut and co-supervisor Dr. Gilbert Laporte who have provided all their support and guidance during the course of this dissertation work.

I also would like to thank Dr. Batta, Dr. Daniel, Dr. Hodgson, Dr. Ingólfsson, and Dr. Johnston for their valuable comments on how to improve this dissertation.

Special thanks are to my wife Gökçe, who has supported me to the maximum possible extent during my doctoral studies, and made this dissertation reach its conclusion.

Finally, I would like to thank my parents, who were not directly involved in this study, but the way they brought me up and invested in my education is definitely one of the main factors that allowed me to reach this milestone.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

In today's political world, most of the countries governed by parliamentary democracy and/or those that have an elected national assembly have adopted the notion of *electoral districts* in their representation systems and voting procedures. In such countries as Canada, Britain, and the United States of America, electoral districts are used as the means by which the residents/citizens of the country delegate their legislative power to a group of representatives. Specifically, an electoral district is a geographical area whose residents are represented by the same set of representatives in a legislative assembly. (The term electoral district is usually used interchangeably with the term *constituency*, which refers to the group of citizens on behalf of whom the elected representative acts.) The actual implementation of this concept appears in many different forms in all levels of governmental hierarchy, from municipal to federal.

The political districting problem studied in this dissertation is the problem of dividing the territory of a nation or other political unit into a set of *single-member* electoral districts. These districts are to be used in subsequent elections in a manner that best satisfies a set of criteria regarding the districts' preferred socio-political composition and territorial boundaries. The term single-member refers to the case where there is only one representative associated with and elected from each electoral district. The alternative is *multi-member* electoral districts, where the residents of a district are represented by multiple seats in the legislature. In a single-member district, the "winner-takes-all" rule is in effect. That is, the candidate who receives the majority of the votes cast in a district gets elected for the single seat in the parliament for that district. As many political scientists have argued, an advantage of this system over one using multi-member districts is the increased authority of the representative within the district. A disadvantage, on the other hand, is the potential lack of "proportionality" when all the districts are considered together. That is, with single-member electoral districts, the overall fraction of the seats won by a political party may not be proportional to the overall fraction of the votes the party gets. Clearly, this is an undesirable outcome, since any political group should have political power proportional to its share of the popular vote, according to the proportional representation principle. Countries such as Germany and New Zealand have therefore chosen to employ a mixture of single- and multi-member electoral districts to reduce the disadvantages of using single-member districts only.

In this dissertation, the political districting problem will only be studied in the single-member context. The problem is to divide a geographical area or a territory into a pre-determined number of single-member electoral districts, based on a set of objectives and restrictions. Note that this problem definition can also be used to address multi-member districting cases where each district is assigned the same number of representatives, therefore should have roughly equal population. (Having made this definition, the term "single-member" will be dropped in the rest of this dissertation and the terms "electoral district" and "political districting problem" should mainly be understood in the single-member context, unless otherwise noted.) Of course, the number of districts is the same as the number of seats in the parliament allocated to this territory. Note that the process of deciding on the number of seats in a territory, i.e. the process of "apportionment", is external to the political districting problem and is *not* studied in this dissertation. Its only role (as far as this dissertation is concerned) is to provide the number of districts a territory will be divided into.

The political districting problem is critical in its own right to the effectiveness of a country's election and representation system in that it provides the guidelines as to how the legislative power of citizens will be exercised by their representatives. If the districting is done in an unfair manner, i.e., in a way that is believed to provide an unfair advantage to a political, social, or ethnic group, basic principles of democracy will be violated to the disadvantage of some citizens. The term *gerrymandering*, which originated from the salamander-shaped district plan prepared by Massachusetts Governor Albright Gerry in 1812, has traditionally been used to refer to such practices. Many gerrymandering claims have enriched the history of districting since then, and many district plans have been challenged in court. A separate discussion on gerrymandering, together with how it is identified, measured and prevented, is also part of this dissertation. The bottom line is that identifying electoral districts that do not violate the constitution and laws of a country, and which are also perceived to be "fair" to all political groups involved, is crucial to the effective representation of the citizens in the legislature.

The electoral districts may need to be revised and the districting problem may need to be re-solved after every census, if the demographic data underlying the existing districts change. Perhaps the most important districting criterion is to have population-wise balanced districts, due to the "one person, one vote" principle common to the constitutions of most democratic countries. Consequently, when an existing district's population (according to the most recent census) goes out of the allowable range, its boundaries have to be altered so that the resulting population is within the acceptable limits. This could be done either by modifying the existing district boundaries (which is the common practice) or by creating new districts from scratch. Sometimes, a territory grows so much in population that a new district has to be added to the existing ones. In Canada and Britain, the usual practice is to put together a committee, which reports to the chief electoral officer for the territory being districted, in order to have a revised set of districts that conform to a set of districting criteria. In other countries, other administrative arrangements might be in effect.

In most cases, the process of reviewing and re-drawing the boundaries is mostly manual. Use of computer software that helps display electoral districts and analyze relevant demographic data has become common practice recently. However, generating alternative district plans is still generally done manually; we know of no commercial product that utilizes optimization tools for automated districting. This causes the process of re-drawing the districts to last for extended periods of time and some desirable alternatives may be overlooked. An optimization-based decision support system may solve these problems. Ideally, given a set of criteria, one should be able to design, using the decision support system, a district plan with boundaries that reflect those criteria and optimally meet the intent underlying their use for defining district boundaries. If the resulting plans have undesirable components, a revised set of criteria might be provided to the system and this process is repeated until satisfactory plans are produced.

This dissertation proposes an algorithmic approach to the districting problem in an effort to reduce the time needed to revise the existing districts and to provide the decision maker with many desirable alternatives in a reasonable amount of time. The algorithms developed and presented in Chapters 5 and 6 detail step by step the process by which the new district plans are to be produced. The main algorithm is based on the principles of the Tabu Search problem solving technique described in detail in Chapter 4. This is a heuristic technique that has been developed and enhanced by many researchers in the last decade and used to solve many decision problems. The application of this technique to the political districting problem is unique and it is based on the selection of a suitable set of criteria. The algorithms operate on both "objective" and "political" criteria, unlike most of the research previously done. These criteria are described in detail in Chapter 2. Some of the criteria might conflict with one another, and to address this issue, a weighted multi-criteria objective function approach is adopted. Each criterion is assigned a particular weight to reflect its relative importance. This approach lets the decision maker analyze several different scenarios by changing the weights of individual criteria or even the set of criteria considered.

Note that, as in any decision support system, the algorithmic approach proposed in this dissertation is meant to provide the decision maker with a set of alternative solutions, given the decision maker's preferences. Consequently, any district plan that emerges from the execution of the algorithm is based on the expressed preferences of the decision maker. Therefore, if the district plans produced are not acceptable in some aspect, it is up to the decision maker to revise the input parameters and carry out the process in an interactive manner. This approach also forces the decision maker to be part of a structured process. That is, the decision maker is forced to be precise as to which criteria are relevant for the decision, how these criteria should be measured, and what the relative importances of the criteria are. This is something that perhaps does not fit easily into how politics is commonly practiced. Clearly, there might be cases where politicians, given their stated set of districting criteria, may be provided with a collection of district plans, none of which appeal to them, perhaps because of a missing unstated criterion. Besides, arguably, some politicians may tend to reject a district plan no matter how objectively it was produced if there is one component that the politicians dislike, even if there is *no* conflict among the criteria. These issues may seem, at first, to work against an effective utilization of the algorithmic approach presented in this dissertation. To overcome this potential problem, the proposed approach should be used in an interactive manner to bring all the relevant criteria into the picture. That is, the decision maker should first be allowed to explain why (s)he dislikes certain components of a district plan, and re-submit a revised set of criteria that supports his/her case. The new criteria set would then be used to produce other district plans, and the process would be repeated until there is no unaddressed criterion left. The resulting set of district plans should then be acceptable to the decision maker.

As with any algorithmic approach, a computational study is needed for the algorithms developed in this dissertation in order to test whether the proposed algorithms are capable of producing good district plans. Chapter 7 provides the details of such a computational study. The experiments are carried out in two stages. The first stage involves parameter fine-tuning, in which different settings for various parameters of the algorithm are tested. The objective of this stage is to find out which settings perform well for a given set of data.

These settings are then used in the second stage to generate district plans under various scenarios. Specifically, the second stage is where different combinations of districting criteria are considered and good district plans are identified based on the criteria chosen. This stage is intended to address the multi-criteria nature of the problem and to analyze the impact of changing the districting criteria on the district plans produced.

The experiments reported in this dissertation are based on real data. Real data were used in order to enable comparisons between the best solutions found by the algorithm with the electoral districts that are in use. For this purpose, enumeration area data for the City of Edmonton, Alberta have been selected. These data are provided by Statistics Canada in the form of electronic enumeration area maps. Enumeration areas are used by the algorithm as building blocks of electoral districts since a district is created by combining a number of enumeration areas in most cases. The general methodology is to start with a complete district plan and then exchange enumeration areas between districts to identify new (and better) district plans. Enumeration areas, on the other hand, might be too fine to use if the territory being districted is as large as a province/state. In these cases, one can use building blocks with less detail, such as census tracts or counties. In any case, the concept of using building blocks in forming electoral districts makes the algorithms sensitive to geographic and demographic data. A Geographical Information System (GIS) is used to manage data and display the district plans produced.

The GIS used in this dissertation helps convert the electronic enumeration area maps provided by Statistics Canada into data usable by the Tabu Search algorithms. A second function of the GIS is to import the outputs of the algorithms and display the district plans produced. Finally, the GIS is used to query, summarize and analyze geographical and demographic data. In this process, the algorithm uses data from the GIS interface and sends the district plans produced back to the interface for display and analysis. These features of the algorithms and the interface developed in this dissertation constitute a semi-integrated system, which provides powerful tools for the authorities charged with the task of creating district plans to use in generating and evaluating plans more quickly and effectively. The GIS also makes it easier to communicate information about various demographic aspects of the district plans in a highly effective way.

The organization of this dissertation is as follows. In Chapter 2, the districting criteria are described along with the methods of measurement. Existing districting models and algorithms are reviewed in Chapter 3 and the basic principles of the Tabu Search technique are presented in Chapter 4. In Chapter 5, we describe the details of the proposed implementation of the Tabu Search technique to the political districting problem, and in Chapter 6, we describe specifics of a problem solving approach that complements Tabu Search, along with the details of its implementation. The results of the computational study using real data are reported in Chapter 7 and the concluding remarks are given in Chapter 8.

Chapter 2

Districting Criteria

Perhaps the most critical aspect of political districting is the choice of criteria used in the process of drawing electoral boundaries. Over the last 150 years, countries that have chosen to adopt district-based representation systems have developed an evolving collection of criteria in attempting to move towards more effective representation of their citizens. Especially in the last few decades, governments have strived to identify and legislate districting criteria that would prevent unfair representation of ethnic, cultural, social or political minorities. In this chapter, such districting criteria are reviewed. Note that no two countries share the same set of criteria. Therefore, the ones that are reviewed in this chapter constitute a selected set of criteria commonly used in a number of countries. The primary focus, though, is on criteria included in districting legislation in Canada and the United States.

The set of districting criteria used determines how the district boundaries will be drawn, which in turn can have an impact on the outcome of the next election. For this reason, districting criteria constitute one of the two important elements of political districting (the other being the actual method employed to draw the boundaries). The districting process can also be abused by individuals, political parties or other groups to acquire power in an unfair way. For example, a party may attempt to redraw the districts in a way that maximizes the number of seats it gets. This phenomenon is known as *gerrymandering* and it is discussed in detail in Section 2.3.

The districting criteria have been reviewed and classified in many different ways in the literature, as shown in Table 2.1 for three studies. In another study, Lijphart [35] provides an inventory of 16 criteria as well as a conflict matrix for these 16 criteria. The conflict matrix is of particular importance to the discussion in this dissertation, since it is directly

Grofman [24]	Morrill [42]	Williams [62]
formal	**constitutional**	**demographic**
· equal population	· equal population	· equal population
· contiguity	· equal probability of representation	· proportional minority representation
· compactness	**geographic**	**geographic**
· to follow political boundaries	· compactness	· contiguity
· preserve communities of interest	· contiguity	· compactness
· coterminality of house and senate plans	· integrity of communities of interest	· community integrity
racial intent	**political-geographic**	**political**
· no intent to dilute minority voting power	· representation of political units	· proportionality
political intent	· integrity of political boundaries	· safe vs. swing districts
· no intent to favor a political party	**political**	· similarity to the old plan
· no intent to favor an incumbent	· minimal changes to old plan	
· no use of political data in redrawing	· no political gerrymandering	
· effort to maintain population equality		
· intent to achieve political "fairness"		
· deference to legislative intent		
· "least-changed" plans		
racial outcome/anticipated outcome		
· no retrogression in representation		
· no dilution of racial minority voting strength		
· 'racial' proportionality		
political outcome/anticipated outcome		
· no (imposed) bias in favor of a party		
· no incumbent-centered partisan bias		
· responsiveness of electoral outcomes		
· preservation of political competitiveness		
· translation of vote majority into seat majority		
· partisan proportionality		

Table 2.1: Classification of districting criteria

related with the concept of multi-objective decision making. Lijphart's list of criteria and conflict matrix are reproduced below, as they originally appear in [35].

1. Representation must be equal for each citizen. This is the basic one-person, one-vote, one-value principle. For single-member district systems, it means that the districts must contain equal number of citizens, in line with the *Westberry* v. Sanders and *Reynolds* v. *Sims* decisions of the U.S. Supreme Court (Tribe, 1978:739-741). For non-single-member districts, the criterion calls for representation that is equal per capita. The number of representatives in multimember districts must be proportional to the population being represented, or the number of votes cast by representatives elected from districts of unequal sizes must be weighted according to the population being represented (Grofman and Scarrow, 1981a:2-3).
2. The boundaries dividing the electoral districts must coincide with local political boundaries as much as possible.
3. Electoral districts must be compact and contiguous in territory.
4. The boundaries of electoral districts should be drawn in such a way as to provide representation for political minorities (*sophisticated gerrymandering*).
5. The boundaries of electoral districts should be drawn in such a way as to provide representation for ethnic and racial minorities (*affirmative gerrymandering*).
6. The electoral system should not be biased in favour of any political party in awarding seats for a certain percentage of the total vote. This is the criterion of *neutrality*, the first of the four criteria of fair districting suggested by Richard G. Niemi and John Deegan, Jr. (1978:1304): "A districting plan which treats all parties alike in allocating seats per given vote totals is said to be neutral."
7. The electoral system should not be biased in favour of any racial or ethnic group in awarding seats for a certain percentage of the total vote. This is the Niemi-Deegan criterion of neutrality, extended to groups other than political parties.
8. The electoral system should have a wide range of responsiveness to changes in the electorate's party preferences. Niemi and Deegan (1978: 1304-1305) define the range of responsiveness as "the percentage range of the total popular vote (for the entire political unit) over which seats change from one party to the other. Specifically, the low end of the [range] is the minimum percentage of the total vote required for a party to win at least one seat ..., while the upper end is the minimum percentage of the total vote required to win all of the seats."

9. The electoral system should have a "constant swing ratio;" that is, the rate at which a party wins seats per unit gain in the percentage of its vote should be constant (Niemi and Deegan, 1978:1306). The principle that a party's share of the seats should be proportional to its vote share is a special case of this criterion.

10. There should be proportionality between the share of the seats won by any particular ethnic or racial group and its vote share. This criterion extends the above proportionality principle to groups other than political parties.

11. The system should be competitive in the sense that each party should have a chance of election in each district. Niemi and Deegan (1978:1309-1311) operationally define this criterion as requiring each party's vote to be in the 45- to 55-percent range in each district.

12. Each citizen should have equal power to affect the outcome of elections by casting the decisive vote (Grofman and Scarrow, 1981a:3-4). This voting power varies inversely with the square roots of the sizes of the district populations (Banzhaf, 1996).

13. Each citizen's vote should be "used" as much as possible toward the election of a candidate and the "wasted" vote should be minimized. The used vote is "the number of votes required by the winning party to attain a plurality and ... represents the votes that are actually used to elect its candidate to office." It can therefore also be defined as "the size of the vote received by the party polling the second highest vote plus one." All other votes—the votes received by the winner that exceed the plurality and the votes of all losing parties—are wasted (Cohan, McKinlay, and Mughan, 1975:365). This criterion is similar to criterion 11 (competitiveness) in the case of a two-party system.

14. Each legislator's power in the legislature should be proportionate to the number of citizens he represents. Banzhaf (1966) argues that not only the citizens's power to affect election outcomes (criterion 12) but also his influence on legislation varies inversely with the square roots of the district populations.

15. There should be equal numbers of representatives working on behalf of equal numbers of citizens. As Bernard Grofman and Howard A. Scarrow (1981a:18-19) point out, representatives have many functions in addition to voting on proposed legislation, such as performing various services for their constituents: "Any system of representation, such as adoption of the square root principle or weighted voting in any form, which results in some citizens having a proportionately greater access to these personal services because it does not apportion equal numbers of representatives to equal

numbers of citizens, would therefore be unfair."

16. A majority of citizens should be able to control legislative outcomes through their representatives (Grofman and Scarrow, 1981a:19-21). In other words, minorities of citizens should not be able to elect majorities of legislators. This the basic majoritarian principle.

Table 1 Conflicts among Sixteen Criteria of Fair Representation under the Plurality-Majority Rule and Geographical Districts *(Criteria, by number—see text)*

	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
1	-	-	X	-	X	-	-	-	-	-	-	-	-	X	X	-
2	-	X	-	X	-	X	X	X	X	X	X	X	X	X	-	-
3	-	X	-	X	-	X	X	X	X	X	X	X	X	-	-	-
4	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
5	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
6	-	-	-	X	-	X	-	-	-	-	-	-	-	-	-	-
7	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
8	-	-	-	X	-	X	-	-	-	-	-	-	-	-	-	-
9	-	-	-	X	-	X	-	-	-	-	-	-	-	-	-	-
10	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
11	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
12	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-
13	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
14	X	X	-	-	-	-	-	-	-	-	-	-	-	-	-	-
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Figure 2.1: Lijphart's conflict matrix for 16 criteria

The conflict matrix given above summarizes the conflicts between the 16 criteria listed. Lijphart argues that because of these conflicts, it is not possible to reach the most desirable levels on all of these criteria. Multi-objective decision support systems are used to quantify the trade-offs between conflicting criteria, and help decision-makers find an acceptable compromise solution. However, the criteria considered by a quantitative decision support system must be quantifiable, and preferably independent from one another. In our opinion, most of the criteria listed in [35] are hard to quantify. This is because some of these criteria are no more than re-statements of the relevant constitutional rights or political concerns, and it is not clear how they would be measured (e.g. criteria 4, 5, 6, 7 and 10). Furthermore, some of the criteria seem to be very similar in purpose (e.g. 9 and 16), which suggests that using all of these criteria in a multi-objective optimization model will amount to over-emphasizing a certain objective relative to others. Because of these reasons, we select a subset of the districting criteria, and include only those for which well-defined measures either exist or can be developed relatively easily. This subset contains criteria from Lijphart's list, but it also contains criteria that are not in this list (e.g. similarity to existing plan, maintaining integrity of communities). These criteria are combined into a single weighted aggregate objective function, where the user can manipulate the weights to increase or decrease the importance of a criterion and hence affect the resulting district plan. With this approach, decision makers can generate compromise solutions that reflect their preferences (or priorities) for the criteria.

Having briefly mentioned the relevant districting criteria and the conflicts among them, we now turn to the criteria that are considered in this study. In addition to Lijphart's inventory and the three classification schemes provided in Table 2.1, several studies (e.g. Harris [26], Hess et *al.* [27], Garfinkel and Nemhauser [15], Morrill [40], Horn [29], Mehrotra et *al.* [37]) recognize a distinction between two major groups of districting criteria which we adopt here. The two groups of criteria are:

- basic criteria, which include population equality, contiguity, and compactness
- additional (political) criteria, such as proportionality, safe vs. swing districts.

Almost all of the literature on political districting consider basic criteria only even though they discuss extensions to handle criteria from the second group as well. Some of the studies argue that using basic criteria only would produce district plans that would be free of gerrymandering claims. Note that the criteria required by the legislation of a country's electoral system may include a combination of criteria from both groups. In any case, the above classification will be used in this chapter, so that the following discussion is parallel to how these criteria have been treated by researchers in the past.

2.1 Basic Criteria

2.1.1 Population equality

Having districts of roughly equal population is by far the most important criterion in the entire districting process. It is associated with the 'one-person-one-vote' principle, which is common to the constitutions of all countries who exercise some form of districting (e.g. Canada, Britain, the United States of America, and Australia). The obvious intent behind this criterion is to maintain that each citizen is represented in the legislature with an (almost) equal weight. It is interesting to note, however, that this criterion has only gained considerable importance in the second half of this century. In 1960, a district plan in California had a population ratio 449:1 between the most populous and the least populous districts (Grofman [24]). Since then, many district plans have been rejected in both Canada and the United States of America on the grounds that this constitutional requirement is not met (to a sufficient degree).

The equal population criterion is embedded so deeply into the districting process that no re-districting attempt would be realistic without it. Typically, each district is required to have a population within a range around the mean population, which is usually referred to as the *district quotient*. The district quotient is computed as the ratio of total population residing in the territory being districted to the number of districts to be created for the territory. Federal or provincial courts or other districting authorities usually specify the permissible range around the district quotient in terms of a relative deviation from the district quotient. The smaller this percentage is, the closer to perfect population equality the district plan is. However, as this allowable deviation is decreased, it becomes more and more difficult to produce district plans that have all district populations within the allowable range. On the other hand, reasonably wide population ranges can also accommodate part of the future population movements.

As an example, consider the 1995/1996 re-districting of the Province of Alberta, Canada. The total population of the province in 1995 was 2 554 779, based on the 1991 census, and the number of districts to be re-drawn was 83. This gives a district quotient of 2 554 779/83 = 30 780. The Electoral Boundaries Commission Act requires that the population of each district be within 25% of the district quotient, and in exceptional situations that are described in the Act, it could be as low as 50% of the quotient. This is a wide range, but the 1995/1996 Alberta Electoral Boundaries Commission provides the following justification:

> "The Commission has interpreted the maximum twenty-five percent limit in population deviations as one of the provisions intended to accommodate regional representation reflecting the different communities of interest within the Province." (The 1995/1996 Alberta Electoral Boundaries Commission Report [1], section 1.6.2.)

Two other stated justifications are to avoid excessively large rural districts and to limit the distance from the Legislature Building in Edmonton to the nearest boundary of any proposed electoral district (same report, section 1.3). Note that the Province of Alberta possesses strong urban and rural characteristics, and political preferences between urban and rural regions are significantly different. This suggests that the 25% and 50% allowable deviations might as well serve the objective of balancing the distribution of seats between the parties that win most of the urban and rural districts. The allowable deviations for other provinces also seem to be consistent with this observation. The Province of Saskatchewan, for example, is a primarily rural district and the allowable deviation is only 5% (Courtney [7]). In the United States, the equal population requirement has been taken to such an extreme that a 1983 New Jersey district plan with an average deviation of 0.138% from district quotient was rejected by the Supreme Court of the United States of America (Grofman [24]). The allowable deviation seems to have been determined in this country on a case-by-case basis in the past.

The major problem with having a very narrow allowable range for district populations is the difficulty of generating district plans that satisfy this requirement. Mathematically, the districting problem can be defined as the partitioning of a set of *base units* (BU) that are used as the building blocks of districts. Consequently, narrowing down the allowable range would correspond to reducing the size of the feasible solution set, which would in turn make it difficult to identify feasible district plans. Even if such plans can be found, it would be at the expense of other criteria, such as compactness or integrity of communities that are described later in this chapter. This is, in fact, the (stated) reason behind the use of a ±25% allowable deviation in the Province of Alberta.

Researchers have developed several measures to measure population equality. Below is a list of the ones that are commonly used. Let m denote the number of districts, P_j population of district j, P_{min} population of the smallest district, P_{max} population of the largest district, $\bar{P} = \sum_j P_j/m$ the district quotient and β the allowable percentage deviation from $\bar{P}$.

- *Mean absolute deviation:* $\sum_j |P_j - \bar{P}|/m$.
- *Mean squared deviation:* $\sum_j |P_j - \bar{P}|^2/m$.
- *Maximum absolute deviation:* $\max_j |P_j - \bar{P}|$.
- *Extreme deviation:* $(P_{max} - P_{min})/\bar{P}$.
- *Extreme ratio:* P_{max}/P_{min}.

While some of the methods attempt to minimize one of the above expressions (e.g. Garfinkel and Nemhauser [15]), most of them seek to maintain the population of each district within the range $[(1-\beta)\bar{P}, (1+\beta)\bar{P}], 0 \leq \beta \leq 1$. In this dissertation, population equality is treated as a constraint, i.e., any plan that has district populations within the allowable range is feasible and acceptable. Of course, the user determines the value of β and it is possible to generate more balanced districts by using smaller values of β. The details are provided in Chapter 5.

2.1.2 Contiguity

In addition to population equality, the districts are usually drawn to be contiguous, even though a province/state may not legally require it. A district is said to be contiguous "if every part of the district is reachable from every other part without crossing the district boundary (i.e., the district is not divided into two or more discrete pieces)." (Grofman [24], p. 84). Since the districts are usually made up of smaller base units, this suggests that each BU in a district should be reachable from each other BU by visiting a sequence of adjacent BUs in that district. In some cases though, contiguity might be open to interpretation. Consider the two banks of a river, across which BUs are connected with a bridge. In essence, contiguity is associated with easy transportation from one part of the district to another. Accordingly, two parts of a district separated by a river but connected with a bridge should be deemed contiguous. Generally, a district would be considered contiguous if any two parts of the district are connected by means of a major transportation route *entirely within* the district. On the other hand, consider a district divided in two by a mountain range, which may make transportation from one side to the other impossible from within the district. In this case, the district might be deemed non-contiguous. To summarize, geographical adjacency is usually considered sufficient for contiguity purposes, but exceptional situations (such as the case with natural boundaries) could be relevant.

Contiguity is not required in many states in the United States of America, such as Arizona, Florida, and Kansas. However, it is still considered rational and therefore desirable to produce contiguous district plans. In Canada, contiguity is required by the Electoral Boundaries Commission Act. The main reasons behind imposing contiguity of districts as a requirement include the following:

- A single-member district is represented by a member of parliament, who is responsible for addressing the needs and concerns of the residents of the district. A non-contiguous district makes it more difficult for this person to reach all parts of his district effectively.
- A member of parliament elected to represent a group of people is likely to do his job

better if the people (s)he represents share similar ideas and have common backgrounds, needs and concerns. This is usually possible if the people live close to each other and perhaps come together on a regular basis at community houses or places alike. While it is still possible that people from different parts of a province or state have many things in common, a contiguous district is more likely to meet the above objectives.

- Contiguity is usually required as a precaution against gerrymandering. After all, why would anyone put together non-contiguous pieces from different parts of the province, if it is possible to create a contiguous district? Contiguous districts are more natural, and easier to defend compared to non-contiguous districts. An exception might be the case where a community might be living in two non-contiguous areas. But even in this case, including the two pieces in a district would be in conflict with the second point above.

Contiguity of a district is easy for humans to verify. However, computer programs that need to verify contiguity require a precise set of instructions. The one that is developed and used in this dissertation is provided in Section 5.4.2.

2.1.3 Compactness

Compactness is one of the two basic criteria that are used as safeguards against gerrymandering (the other being contiguity). Webster's Dictionary defines the noun 'compact' as "firmly put together, joined, or integrated; predominantly formed or filled". In the political districting context, a compact district is one that is close in shape to a circle or a square. That is, a compact district should not be "long and skinny", and it should not be dispersed haphazardly over a large area. The figures on the next page, taken from Young [63], display some of the shapes that would be considered non-compact. A compact district plan, on the other hand, is characterized as a collection of compact districts.

Compactness is perhaps the most fiercely debated criterion in political districting. It is included among the districting criteria primarily to avoid gerrymandering cases, but there is no single well-defined and widely-accepted compactness measure. Over the last 40 years, more than two dozen measures have been proposed in the literature, some of which have been used by districting authorities. Unfortunately, there is no consensus in the political world as well as among scholars as to which particular one should be used. Furthermore, many researchers mistakenly argue that the presence of odd non-compact districts indicates that gerrymandering has occurred. In reality, what gerrymandering is all about is whether or

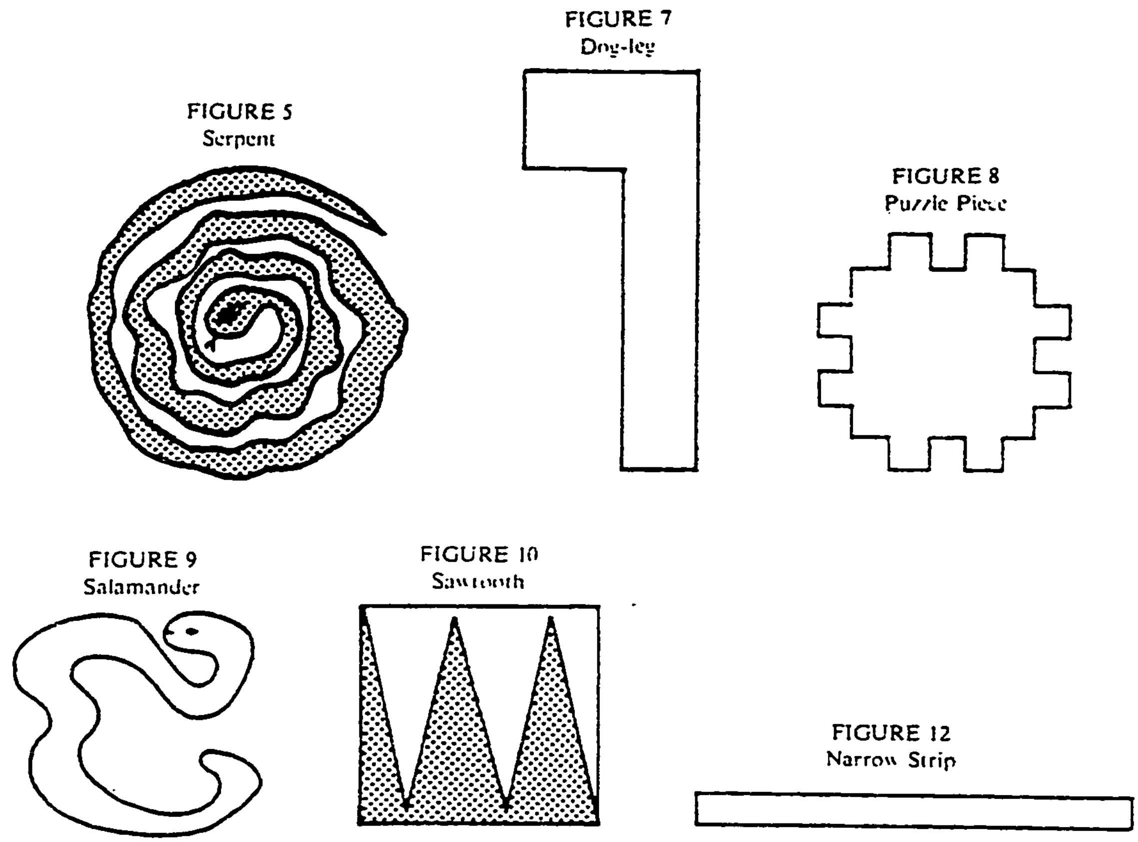

Figure 2.2: Example of non-compact shapes

not a district plan provides an unfair advantage to a particular group. From this viewpoint, a highly compact plan might have been gerrymandered, too. This suggests that as long as the constitutional rights of citizens are not violated, compactness should be less of a concern. Gerrymandering-related issues are discussed more in detail in Section 2.3.

Niemi et *al.* [46] empirically show that some of the compactness measures might be in conflict. These authors provide an extensive list of measures accompanied by computational testing. They first categorize the compactness measures in four major groups, as summarized in Table 2.2. Then, they compare some of the measures using real districts from Rhode Island, upstate New York, Indiana, Colorado and California. With the New York data, they find that two measures may rank a given set of districts in a drastically different order of compactness. The two specific measures that yielded this result are:

- Dis_{10}: ratio of district area to the area of the circle with diameter equal to district's longest axis

- *Per*$_2$: ratio of district area to the area of a circle with the same perimeter as the district

Dispersion measures
· Length versus width
· District area compared with area of compact figure
· Geographical moment-of-inertia
Perimeter measures
· Perimeter only
· Perimeter-area comparisons
Population measures
· District population compared with population of compact figure
· Population moment-of-inertia
Other measures

Table 2.2: Compactness measures

Furthermore, their results indicate that, among all the measures tested, no single measure appears to be the most effective one in all aspects. As a result, they recommend use of multiple measures whenever possible, since each type of measure addresses a different spatial property of a geographical shape. The ones that they prefer over the others are:

- *Dis*$_7$: ratio of district area to the area of the minimum circumscribing circle (Roeck [52])
- *Dis*$_{10}$: ratio of district area to the area of the circle with diameter equal to district's longest axis (Horton [30], Gibbs [19])
- *Per*$_1$: sum of district perimeters (Wells [61], Eig and Seitzinger [10], Adams [2])
- *Per*$_2$: ratio of district area to the area of a circle with the same perimeter (Cox [8])
- *Pop*$_1$: ratio of district population to the population of the minimum convex figure that completely contains the district (Hofeller and Grofman [28])
- *Pop*$_3$: population moment of inertia (Weaver and Hess [60])

In addition to this detailed study by Niemi et *al.* [46], Young [63] summarizes eight tests for compactness. He demonstrates example shapes, for each test, where he perceives the test to have falsely identified a district as compact or as non-compact. He also points out the lack of one consistent measure that could safely be used to evaluate compactness of districts and district plans. Young's tests are:

1. Visual: compare multiple districts visually.

2. Roeck Measure: compare area of district to the area of the smallest circumscribing circle (Roeck [52], measure Dis_7 above).

3. Schwartzberg Measure: construct the adjusted perimeter of the district by connecting by straight lines those points on the district boundary where three or more constituent units from any district meet. Divide the length of the adjusted perimeter by the perimeter of a circle with area equal to that of the district (Schwartzberg [55]).

4. Length-width Measure: Find a rectangle enclosing the district and touching it on all four sides, such that the ratio of length to width is a maximum (Harris [26], Papayanopoulus [49]).

5. Taylor's Measure: Construct the adjusted perimeter of the district as in the Schwartzberg Test. Subtract the number of 'reflexive angles' from the number of 'non-reflexive angles' and divide by the total number of angles (Taylor [56]).

6. Moment of Inertia Measure: Compute the square of the distance between the geographical center of a base unit and the center of gravity of the district, summed over all BUs (Weaver and Hess [60]).

7. Boyce-Clark Measure: Compute the distances between the center of gravity of a district and those points on the perimeter of the district formed by equally-spaced radial lines. Compute the average % deviation from the average of all such distances (Boyce and Clark [3]).

8. Perimeter Measure: Find the sum of the perimeters of all the districts.

To summarize, while there is growing interest in incorporating compactness into the districting processes, there is no single widely-accepted measure of compactness. The districting authorities usually prefer easy-to-understand measures that do not require massive data, while researchers propose more complex measures in an effort to capture different spatial characteristics of geometric shapes. In this dissertation, two compactness measures are incorporated into the proposed solution technique. The first is the sum of district perimeters (Per_1) scaled by the perimeter of the entire territory, and the other is the ratio of district perimeter to the perimeter of the circle that has the same area as the district (the perimeter version of Per_2), averaged over all districts. Unlike contiguity, which is treated as a hard constraint, compactness is included as an objective function term. This is because contiguity is well-defined and easy to check, and also it is a binary concept (i.e., a district is either contiguous or not). Compactness, on the other hand, is measured in many different ways, and the values come from a range as opposed to being binary.

2.2 Other Criteria

The three criteria described in the previous section are the basic criteria that are usually considered *and* implemented in any study on political districting. Some researchers prefer considering these three criteria only (i.e., ignoring all relevant political criteria), so that the resulting districts cannot be challenged by gerrymandering claims (e.g. Mehrotra et *al.* [37]). Some studies are *required* to do so, simply because of the instructions from the authority who commissions the districting task. Others choose to consider political criteria as well, so that the districting activity gets closer to reality and other issues of effective representation such as proportionality of percent votes to percent seats can be addressed. This section contains a brief description of a number of political criteria that are highly relevant to any districting process.

2.2.1 Following the boundaries of political sub-divisions

One important political criterion in redistricting is to preserve the boundaries of government units such as counties, townships, or even cities. Both in Canada and the United States, the legislation requires district authorities to respect the boundaries of such units to the greatest extent possible. Of course, there may be cases where a county or a city is too large in population to be completely included in a district. Also, strict population equality requirements may not leave the districting authority any other alternative but to divide some of the political units. In such cases, division of a political unit should be done with attention to community structures within the unit.

One reason behind imposing this criterion on the districting process is that "registration and elections are organized by such political units, and voters identify with them" (Morrill [42], p. 25). It is more natural and rational to respect these boundaries, when the voters can easily get confused and suspicious with arbitrarily drawn districts. Another reason is again gerrymandering-related. Having to follow the boundaries of sub-divisions clearly reduces the number of ways a district plan can be drawn. As a result, one has less flexibility in drawing gerrymandered districts, by putting together pieces that are significantly different in expected election outcome. One last reason is to achieve more effective representation, since the composition within counties, towns, or cities is relatively homogeneous. This results in better representation of the residents who may have common concerns and views.

For the Province of Alberta, the two major urban centers Edmonton and Calgary are essentially districted on their own. In other words, the municipal boundaries of these two cities are respected no matter what. Edmonton is divided into 19 districts and Calgary is divided into 21 districts. The Electoral Boundaries Commission Act requires that other municipal boundaries also be respected and the presence of understandable and clear boundaries is crucial.

2.2.2 Equal probability of representation

Another criterion based on the constitutional principles is equal probability of representation. This criterion maintains that each citizen have no less opportunity than others to participate in the political process and to elect representatives of his/her choice (United States Voting Rights Act, section 2(b)). It is mostly intended for ethnic minorities in the United States, who have historically been under-represented at various levels of American legislatures. To comply with this requirement, several states have created minority districts in proportion to the minority's population in the state (e.g. Maryland). Today, it is not allowed to exclusively and intentionally create minority districts. However, any district plan that appears to dilute the voting power of ethnic minorities may be subject to challenge in court.

Of course, the formation of minority districts may be in conflict with other criteria. For example, it may not be possible to create a compact district if the minority population is scattered along a river or a highway. Also, the minority population may be too large to form a single district, but may not be large enough for two districts. In this case, a minority community will have to be divided in two. These issues suggest that one may have to compromise other criteria in order to create minority districts. Another issue is that a 50% to 55% majority in the district may not be sufficient for an ethnic minority to elect its representative in a two-party system, because of historically low levels of voter registration and a high proportion of children among minorities. At the other extreme, a majority of 70% in a minority district would amount to wasting some of the minority votes that could be used elsewhere. The question is then what level of population to establish in a minority district so that the votes are not wasted and minorities can elect their representative. In this dissertation, no effort has been made to create minority districts or address this criterion in another way.

2.2.3 Similarity to the existing plan

In almost all re-districting cases, the courts or commissions attempt to identify a new set of districts by changing the existing plan slightly, just enough to accommodate the changes needed to satisfy constitutional requirements. This is in accordance with the objective of having the new plan as similar to the old plan as possible. The rationale behind this objective is two-fold. First, the existing plan might be a good plan in that it maintains population balance, compactness and integrity of communities fairly well. Consequently, the districting authority as well as the political parties would prefer keeping the existing plan as much as possible.

The second reason to leave districts in the existing plan intact is from an incumbent's or a political party's perspective. An existing district is the place where the incumbent is known well, (s)he got in good contact with the residents and had a chance to demonstrate his/her representation skills. In a completely new district, (s)he will have to familiarize himself/herself with the social/cultural composition of the district, what people's problems are and how they can be solved. This would in turn make it less likely for this incumbent to prepare for the election and get re-elected. Sometimes, it is simply the fact that an experienced incumbent has so much to offer to residents of the district where (s)he has been in service for a while. Furthermore, a whole new set of districts would necessitate a whole new set of interests and election strategies for each party, and a new power balance with the other parties. A less important factor is the requirement that incumbents should live within the boundaries of the district where they get elected. A drastic change in district boundaries may make it impossible for an incumbent to run in the same district again.

The above reasons are partially valid from the perspective of the residents of the old district as well. An incumbent whom the residents had known for a long time and with whom they had good communication would be preferred as opposed to a completely new candidate. In some cases though, the composition of the districts has to be changed simply because a new district is added to or removed from the territory (as happened in Edmonton and Calgary in 1996).

This criterion is one of the political criteria considered in this dissertation. Unfortunately, no mathematical measure of similarity between two plans has been proposed in the past. For this reason, a measure is developed in this study, which is detailed in Chapter 5.3.3.

2.2.4 Integrity of communities of interest

Morrill [42] argues that citizens vote, in part, according to their various interests such as religious values, occupation, rural and urban orientation, etc. This is parallel with the general principles of territorial representation in that a group of people who live close to each other usually share a common set of values, and would like to be represented by a person with a similar set of values. Today, communities are perhaps becoming less and less territorial as a result of new communication technologies and ease of transportation. However, geographic proximity is still the main determinant of community structure. As a result, the integrity of communities of interest appears as a valid criterion in the districting process.

There are many ways of defining communities, for example based on ethnic origin, native language, religion, occupation, education level, income level, family structure, urban vs. rural orientation, political orientation, and even being associated with a natural feature such as a lake, valley or oil resource. Of course, some of these are interrelated (e.g. the relation between occupation, education, and income level). Morrill [43] views communities to be composed of three components: partisan allegiance, racial or ethnic composition, and urban versus rural character. The challenge is then to identify a district plan that has as many of these communities integrated as possible. However, a districting authority may not (and would not) always consider all types of communities. For example, in the United States of America, only ethnic origin and language-based communities get considerable attention, while religious orientation and economic status-based ones do not (Williams [62]).

The more communities are to be considered, the more difficult it is to solve the redistricting problem. After all, some communities may be in conflict 'geographically', i.e., it may be impossible to include them in a contiguous district. Therefore, a decision maker should carefully choose the set of communities of interest. Another aspect is to group different communities in one district. The question is whether such communities should be very much alike or drastically different. Also, a community may be located in a non-compact geographical area, such as along the coast of a bay. In this case, it may not be possible to create a compact district that completely contains the community. As a result, this criterion would be in conflict with the compactness requirement. In this study, integrity of communities is one of the criteria considered, but the number of communities is kept small. This is done so that this criterion does not interfere much with the rest of the criteria, and also does not impose too many constraints on the problem that will reduce the number of alternative solutions significantly.

2.2.5 Proportionality

The term *proportionality* is used to describe the relation between the share of seats and the percentage of votes a party gets. Ideally, each party should get a number of seats proportional to its percentage of the popular vote, so that it has a legislative 'power' no more and no less than it should. However, since the number of seats is integer, this is usually impossible to achieve. In fact, the results of many elections in Canada and the United States of America indicate that the difference between the vote/seat shares can indeed be substantial. For example, in 1979 provincial election of Alberta, the Progressive Conservative Party won 94% of the 79 seats with 57% of the popular vote (Report of the Chief Electoral Officer [11]), and in 1992 in North Carolina, the Democrats won 67% of the congressional seats with 52% of the state-wide congressional votes (Lewyn [33]). Naturally, these election outcomes bring up the question whether gerrymandering has taken place.

Whether or not a district plan satisfies proportionality needs to be decided based on some (expected) election data. One could use the results of the last election or results based on pre-election polls. With the former, there is always the possibility that the outcome of the next election may be significantly different from the last one. On the other hand, using expected election results may not be very accurate or in some cases be biased towards a party. Therefore, using proportionality as a deciding criterion in districting might be inappropriate. As far as the measurement of proportionality is concerned, the absolute difference, for the entire region, between the percent seats and the percent votes received by a party is the one most commonly used.

Note that finding a least proportional district plan (i.e., the largest difference between % votes and % seats) would be the opposite of proportionality, and it would be considered gerrymandering, since it would favor at least one party. This is related to the concept of safe vs. swing districts, and is discussed in the next section.

2.2.6 Safe vs. swing districts

A district is considered *safe* for a political party, if the candidate of the party running in that district is almost guaranteed to be elected. On the other hand a *swing* district is one where two or more candidates have roughly equal support, and therefore either of them can possibly win in this district.

In general, politicians would like to see district plans with a certain number of safe districts, some of which are especially needed for the party leaders. This allows them to have a reasonable presence in the legislature, even if the election turns out to be a landslide for another party. The question is how many and which of the districts should be safe. Morrill [42] points out the parties' tendency to have one-fourth to one-third of all districts as safe, with the remaining one-third as swing districts (this, of course, assumes two dominant parties with roughly equal voting support). This is usually what the politicians have in mind in a bipartisan districting. When one of the parties is significantly dominant though, the number of safe districts tend to be higher than it should.

A limited number of studies have considered this criterion in their approach for solving the problem (e.g. Nagel [44], Owen and Grofman [48]). These are reviewed briefly in Section 3.2. The proposed methodology in this dissertation does not address this criterion directly, but provides a mechanism for maximizing the number of seats won through *minimizing* proportionality. The details are provided in Section 5.3.5.

2.2.7 Miscellaneous criteria

The districting criteria discussed in the previous sections are the ones that are most commonly addressed by politicians as well as researchers who attempt to develop solution techniques for the problem. This, however, is by no means an exhaustive list of criteria. An interested reader who would like to obtain more information on districting criteria is referred to the study by Grofman [24] (see Table 2.1 for the highlights). This study reviews a large set of criteria with supporting information from court cases.

2.3 Gerrymandering

Perhaps the most well-known concept regarding political districting is *gerrymandering.* Historically, this word has been used to refer to any districting activity that has taken place to provide unfair advantage in an election to a political/social/cultural/ethnic group. The term originates from a 1812 district plan prepared by the Massachusetts Governor Elbridge Gerry. This plan, which is suggested to have a salamander-like shape as it appeared in the Boston Gazette on March 26, 1812, is reproduced on the next page.

There are many forms of gerrymandering. *Partisan* gerrymandering is said to have occurred when the districting provides an unfair advantage to one political party against the others.

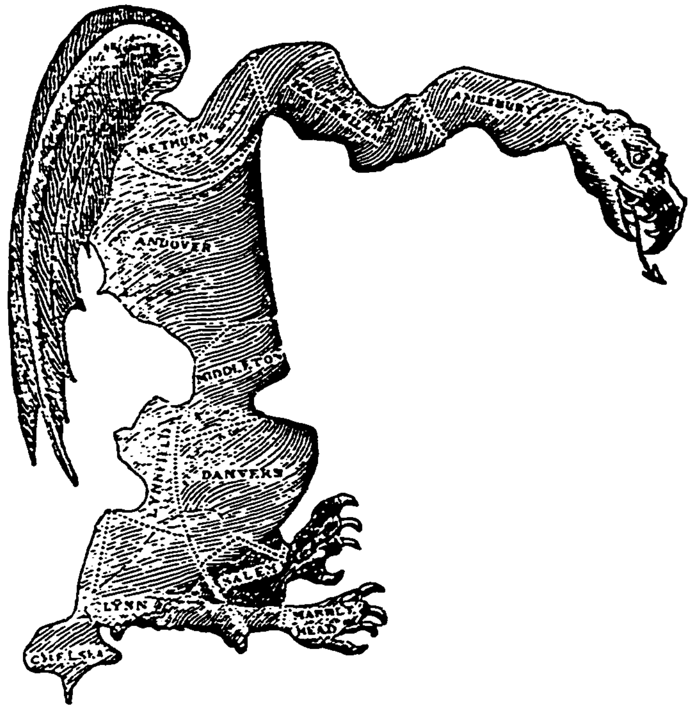

Figure 2.3: The original gerrymander

It is called *bi-partisan* gerrymandering when two parties are involved and benefit from the resulting district plan. *Racial* gerrymandering occurs when a racial minority is put at disadvantage by reducing its effective voting power. No matter in what form it appears, a common belief is that gerrymandering is unfair to the negatively affected groups, and also that it is unconstitutional. At the other extreme, some argue that "all districting is gerrymandering" (Dixon [9]) because "every districting method helps someone at least to the extent of hurting someone else" (Polsby and Popper [51]). Yet, another argument, as mentioned by Lewyn [33], is that "voters are just as well represented by a gerrymander as by any other districting plan, because

> [s]o long as candidates must compete for electoral pluralities or majorities in a district, simple political expediency dictates that they take into account the preferences and interests of all constituents whose votes they may need some day, including at least some of those who might ordinarily be expected to support their opponents (Schuck [54])."

In the presence of these drastically different points of view, it is not hard to imagine that the political science history is filled with a countless number of gerrymandering claims.

Discussion of these cases is clearly beyond the scope of this dissertation. Therefore the interested reader is referred to the article by Lewyn [33], which is a good resource for reviewing some of the cases and which also provides tests and standards for checking the presence of gerrymandering.

Packing and *cracking* are two well-known methods of gerrymandering. Packing refers to concentrating the expected votes of a party in a few districts, so that this party wins those districts with overwhelming majority, but loses the remaining ones. Cracking is the other extreme: the voting power of a party or an ethnic group is diluted across many districts in such a way that this party/group cannot win a proportional number of districts. Figure 2.4a and 2.4b demonstrate these two techniques on two hypothetical cases (source: Morrill [42] with minor modifications). In the first one, 160 liberals live in the city, 140 conservatives live in the country and the area is to be divided into 3 districts. By drawing the boundaries as shown in thick lines in Figure 2.4a, the urban population is 'packed' in an urban district, and the rest of the liberal votes are 'wasted' in the remaining two rural districts. In this case, the liberals have a clear overall majority, but they get only one of the three seats.

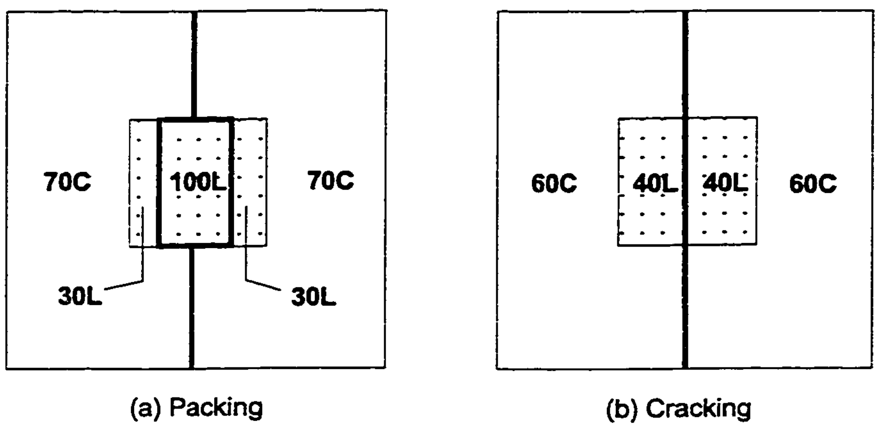

Figure 2.4: Packing and cracking as two gerrymandering techniques

In the second case, the vote distribution of the two parties suggest that each should get one seat, but conservatives get both of the two seats by diluting the voting power of the liberals across the two districts. Now suppose the numbers are 51-49 in both districts. While conservatives are still expected to win both districts, a slight swing in the voter preferences may result in a completely opposite outcome. By using some of their voting support to dilute liberals' votes, the conservatives actually weaken themselves in the rural area (i.e., weaken some of their safe seats).

Observe that, in both cases, a perfect population equality is achieved in all districts and the districts are highly compact. This brings us to the following question: how can one tell that a district plan is a gerrymander? The answer of this question is another debate among scholars. Many argue that the existence of non-compact or odd-shaped districts perhaps with violation of political boundaries and integrity of communities are strong indicators of gerrymandering. It is true that a gerrymandered plan may have to put together a very odd combination of base units in order to provide advantage to a political party or another group. However, it is still theoretically possible to create gerrymandered yet highly compact district plans, even though it is usually more difficult to do so. As a result, many others argue that contiguous and highly compact plans may also contain gerrymanders. Consequently, it is the resulting representation of the involved parties that should indicate whether gerrymandering has taken place, not the shapes of the districts. (Recall from the previous section that a district plan that respects integrity of communities may not necessarily be compact, depending on how the population of the communities is distributed.)

Proportionality, as a result, is commonly used as an indicator of gerrymandering. However, if an election results in a drastic difference between the percent votes and percent seats a party gets, there may be *other* reasons behind this outcome. For example, the political preferences might have changed in the meantime, which would invalidate the past election data used for measuring proportionality. Hence, a district plan which is supposed to ensure proportionality based on last election's results may not generate a proportional outcome in the next election. More importantly, the underlying population distribution may be such that it may be impossible to achieve an acceptable level of proportionality. For example, the supporters of a party might be scattered throughout the entire territory, and this would in turn make it difficult for this party to win a proportional number of seats. (Consider the case where there are 150 supporters for Party A and 100 supporters for Party B in all base units. No matter how the districting is done, Party A will win 100% of the seats with 60% of the votes.) The bottom line is that it may be very difficult, if not impossible, to determine whether a large proportionality gap is due to gerrymandering or due to one (or both) of the two reasons discussed above. Therefore, this criterion should be used with caution in determining whether gerrymandering has taken place or not.

Based on the discussion in this chapter and given the highly political nature of the problem, one could expect computer programs that implement algorithmic approaches to assume an increasing role in the process by providing an objective framework for creating district plans. In our opinion, this has not yet occurred to the level that today's fast computers can efficiently be made use of. To a great extent, politicians are reluctant to accept district plans suggested by a 'machine'. The objective of this dissertation is to overcome this barrier to a reasonable extent, by providing a multi-criteria approach to the problem that considers many of the political criteria involved. The algorithmic approaches developed in this study should be viewed as part of an interactive decision support system, which is intended to provide the decision makers as many alternatives as practically possible (even district plans that are intentionally gerrymandered). The decision makers should normally specify their criteria and how much importance they associate with each of them, then let the system identify district plans that meet these criteria. The next step is to review these plans and check whether they are desirable in all aspects. If not, the decision makers should supply a revised set of criteria that addresses the undesirable aspects of the plans. The process should be carried out on an interactive basis until satisfactory plans are produced.

The following chapters of this dissertation include an overview of the models and algorithms developed for solving the political districting problem, then the basics of a problem solving technique called 'Tabu Search'. Following these two chapters, the proposed algorithms are detailed and a computational study is carried out.

Chapter 3

Models and Algorithms

The political districting problem has been studied by many researchers in the past including operations researchers and geographers in addition to political scientists. This chapter contains an overview of the past research from mathematical and algorithmic perspectives. A recent survey by Williams [62] also provides a similar review.

Most of the literature review in this chapter recognizes the political districting problem as a discrete or combinatorial optimization problem. The districts are made up of building blocks referred to as *base units* (BU), such as counties, census tracts, and enumeration areas. These base units are small geographical areas that collectively cover the territory being districted. The problem is then to determine a partition of n BUs into m subsets, where m is the number of districts to be formed. The districts in general would need to conform to a given set of criteria. The problem has similarities to the set-partitioning or generalized-assignment problems, which are known to be NP-Hard (Garey and Johnson [14]). This suggests that, even with a single objective, the political districting problem is very difficult (if not impossible) to solve optimally, if the problem size is sufficiently large. However, there is no study that specifically proves the NP-hardness of the political districting problem for any given formulation. Since the problem is difficult due to its combinatorial nature, most of the proposed algorithmic approaches are heuristic approaches that seek near-optimal solutions. The next two sections provide an overview of these techniques as well as the models associated with some of them.

3.1 Models

Mathematical models developed for the political districting problem appear as early as in 1965 in an article by Hess et *al.* [27]. In this article, the authors use United States Census enumeration districts (ED) as the base units and seek the most compact district plan that also satisfies the population balance constraint. They provide an integer programming (IP) formulation for the problem, but point out that no exact algorithm known to date can solve the problem, even for a small state like Delaware with 650 EDs and 17 districts. The authors consider contiguity as one of their criteria, but neither their IP formulation nor their proposed heuristic approach *guarantees* contiguity.

The formulation in this article is based on the assignment of population units (i.e., EDs) to district "centers". The authors assume that only population units can serve as district centers. The binary variable x_{ij}, $i, j = 1, \ldots, n$, where n is the number of EDs, is used to denote these assignments:

$$x_{ij} = \begin{cases} 1 & \text{if the } j\text{th population unit is assigned to } i\text{th center} \\ 0 & \text{otherwise} \end{cases}$$

In addition, the authors use d_{ij} to denote the distance between base units i and j, P_j to denote the population of ED j, a and b to denote the allowable lower and upper percent deviation from the district quotient. The formulation in this study is given below:

$$\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{n} d_{ij}^2 P_j x_{ij}, \\
\text{s.t.} \quad & \\
& \sum_{i=1}^{n} x_{ij} = 1, && (j = 1, \ldots, n) \\
& \sum_{i=1}^{n} x_{ii} = m, \\
& \sum_{j=1}^{n} P_j x_{ij} \geq (a/100)(\sum_{j=1}^{n} P_j/m)x_{ii}, && (i = 1, \ldots, n) \\
& \sum_{j=1}^{n} P_j x_{ij} \leq (b/100)(\sum_{j=1}^{n} P_j/m)x_{ii}, && (i = 1, \ldots, n) \\
& x_{ij} \in \{0, 1\}
\end{aligned}$$

In this formulation, the objective function measures the compactness of a district plan as the moment of inertia of each district's population about its population centroid, summed over all districts. The four constraints of the formulation require that each ED be assigned to one and only one center (i.e., district), that there should be exactly m centers, and that each center have a population within $a\%$ and $b\%$ of average district population.

Note that there is nothing in the formulation regarding the contiguity of districts. However, minimizing moment-of-inertia will often result in contiguous districts. A much more important shortcoming of this model is that a feasible solution to the model could have $x_{ii} = 0$ for some i, but $x_{ij} = 1$ for some $j \neq i$. This situation would arise if the population equality constraints are not tight enough, and as a result many (i.e., more than m) small-size districts would form (even though the solution would still satisfy the second constraint). Adding the constraint $x_{ij} \leq x_{ii}$ $\forall i, j$ should eliminate this problem, but having tight population equality constraints could also help. The details of the algorithm used by the authors are given in Section 3.2. Nevertheless, this formulation deserves credit as one of the earliest ones developed for the problem.

A more recent study that provides a mathematical model is by Garfinkel and Nemhauser [15]. These authors also consider the three basic criteria: population balance, contiguity and compactness. They first generate, in Phase I, a set S of 'feasible' districts, where a feasible district has the following properties:

- its population is within a pre-specified range,
- it is contiguous,
- the distance between pairs of BUs is less than given individual thresholds, and
- it has an 'area measure' of compactness smaller than a given threshold value.

Once S is constructed, the second phase is to determine a subset of S of size m, that covers all the BUs and minimizes the maximum deviation of district population from district quotient. The binary variable x_j indicates whether or not district j is included in the selected subset. This formulation is also given below:

$$
\begin{array}{ll}
\min & \max_{j \in S} c_j x_j \\
\text{s.t.} & \\
& \sum_{j \in S} a_{ij} x_j = 1 \quad i = 1, \ldots, n \\
& \sum_{j \in S} x_j = m \\
& x_j = 0, 1 \qquad j \in S
\end{array}
$$

Here, c_j is a cost factor measured as the deviation of district j's population from the district quotient in either direction. The symbol a_{ij} indicates whether BU i is in district j or not. The two constraints of this model maintains that each BU is covered exactly once, and the number of districts selected is m.

While the above formulation is based on three criteria only, the authors claim that their methodology works with other criteria as well. The ones that they mention are to avoid crossing county boundaries, and to guarantee one or more political parties a minimum number of districts to win. However, the authors do not implement these other criteria nor do they mention the computational effort needed to implement them. The two phases of their methodology are further detailed in Section 3.2.

Note that the two formulations provided above are structurally different in that the former identifies a complete district plan by partitioning base units, while the latter puts together a district plan using a collection of previously constructed districts. A third model, which is somewhat similar to the first one, is given by Morrill [41]. Unlike the first model, where the district centers are chosen from population units, this model assumes that the locations of district centers are unknown. Furthermore, the decision variable x_{ij} now represents the *population* assigned from base unit i to district j. Below is the formulation of this model:

$$
\begin{aligned}
\min \quad & \sum_{i=1}^{n}\sum_{j=1}^{m} d_{ij}x_{ij} \\
\text{s.t.} \quad & \\
& \sum_{j=1}^{m} x_{ij} = p_i \qquad i = 1,\ldots,n \\
& \sum_{i=1}^{n} x_{ij} = P_j \qquad j = 1,\ldots,m \\
& x_{ij} \geq 0
\end{aligned}
$$

In this linear-programming model, d_{ij} is the unknown distance between BU i and center of district j, and p_i is the population of BU i. Observe that the model allows BUs to be split among multiple districts. It is the same as the model for the famous transportation problem, except for the fact that district centers, and therefore the d_{ij}, are unknown. As a result, the problem is solved as an iterative (location-allocation) optimization problem. That is, first an initial (arbitrary) set of locations is supplied and d_{ij} are computed. Next, the optimal solution is determined and this solution is used to revise the district centers and shift them to a new set of locations. These steps are repeated until there is no shift in the location of district centers. This procedure is in fact proposed earlier by Weaver and Hess [60] and its details are provided in Section 3.2.

A modified version of the above model appears in Morrill [42], where the following constraint, which forces districts to have equal populations, is added:

$$P_j = \sum_{i=1}^{n} p_i/m \qquad j = 1, \ldots, m \tag{3.1}$$

This is the same as treating the P_j in the original model as equal. One can very easily modify this constraint to allow deviations from the average as follows:

$$(a/100) \sum_{i=1}^{n} p_i/m \leq P_j \leq (b/100) \sum_{i=1}^{n} p_i/m \tag{3.2}$$

Two other studies use essentially the same criteria and similar models, but employ different measures of compactness. Plane [50] criticizes Hess et *al.* [27] for using moment-of-inertia in the objective function and proposes another expression. The one he suggests is a quadratic expression that measures the total spatial interaction (e.g. commuting) between pairs of BUs that make up a district. Alternatively, he uses another quadratic expression that measures compactness as the total "interpersonal separation". This latter concept is represented by the quantity $S_{ij} = \frac{p_i p_j D_{ij}}{2}$ for a pair of BUs i and j, where D_{ij} denotes the distance between population centroids of BU i and j. Consequently, the objective function minimizes the expression (over all possible solutions X)

$$s(X) = \sum_{i=1}^{n} \sum_{j=1}^{n} \sum_{k=1}^{m} x_{ik} x_{jk} S_{ij} \tag{3.3}$$

where x_{ik} is the binary variable that indicates the assignment of BU i to district k.

The second study is by Fleischmann and Paraschis [12] who consider the problem in the salesman districting context. They require that the districts be compact so that the total cost of travel within the district is low. Consequently, the expression

$$\sum_{j \in J} \sum_{i \in I} x_{ij} [(N_j - n_i)^2 + (E_j - e_i)^2] \tag{3.4}$$

is minimized, where x_{ij} denotes the amount of workload assigned from BU i to district j (could be less than the full workload), the variables N_j and E_j denote the north and east coordinates of the center of district j, and finally n_i and e_i denote the north and east coordinates of BU i. Since N_j and E_j are not known initially, an iterative location-allocation procedure is also applied to solve this problem.

Most of the other districting models in the literature share elements similar to those in the above basic models with the same set of districting criteria (see Marlin [36], George et *al.* [18], Horn [29], and Mehrotra et *al.* [37]). The remaining studies in this area propose solution techniques without providing a mathematical formulation in the traditional mathematical programming format (see Vickrey [59], Weaver and Hess [60], Harris [26], Nagel [44], Thoreson and Liittschwager [57], Gearhart and Liittschwager [16], Nagel [45], Morrill [40] and Browdy [5]). These solution techniques are covered in the next section.

All of the above models take into account a very limited set of criteria: population balance and compactness. Contiguity is also considered, but it is usually achieved through the algorithms instead of the formulations. Many of the studies mentioned above discuss the extensions of their methodologies to handle political criteria, but the only one that actually implements some of these extensions is by Nagel [44]. This particular study considers such political criteria as similarity to the existing plan, proportionality, and safe and swing districts. The details of this implementation are provided in Section 3.2.

One last study that is considerably different from the previous ones and therefore deserves a special mention is by Owen and Grofman [48]. These authors look at the districting problem from the perspective of a political party that has full control over the districting process. Consequently, they attempt to solve the problem in a way that maximizes this party's expected seat share. The other objective of interest in this study is to maximize the probability of legislative majority. In other words, these authors perform gerrymandering intentionally and optimally, in order to provide the maximum advantage to the political party in power. They assume that the outcome of the election is probabilistic and therefore the expressions in the objective functions are expected quantities. No compactness or any other criteria are involved.

3.2 Algorithms

In this section, several methodologies for solving the political districting problem are described. For easy reference, these studies are listed in Table 3.1. Note that all of these algorithms except the one by Garfinkel and Nemhauser [15] are heuristic approaches. This is because most of them were developed at a time when computational resources were not sufficient to solve a realistic large-scale instance of the problem. Also, depending on the criteria considered, the problem may have a lot of constraints, some of which may not be easy to represent mathematically (e.g. contiguity).

Vickrey, 1961
Weaver and Hess, 1963
Harris, 1964
Hess et *al.*, 1965
Nagel, 1965, 1972
Thoreson and Liittschwager, 1967
Gearhart and Liittschwager, 1969
Garfinkel and Nemhauser, 1970
Morrill, 1973, 1976, 1981
Tobler, 1973
Bourjolly et *al.*, 1978
Marlin, 1981
Plane, 1982
Fleischmann and Paraschis, 1988
Browdy, 1990
Horn, 1995
Mehrotra et *al.*, 1998

Table 3.1: Studies that contain algorithmic approaches

The earliest algorithmic approach that attempts to solve the political districting problem is by Vickrey [59], who considers population balance, contiguity and compactness as the districting criteria. He suggests using census tracts as the base units that make up the districts. A district plan is then constructed as follows: Initially, one of the census tracts is chosen at random. This tract is used as the reference tract to generate seeds for the construction of districts. A district seed is chosen from the unassigned tracts as the one that is currently the furthest from the reference tract. Once a seed is chosen, it becomes the first member of its district and neighboring unassigned tracts are added on a continuous basis until the district population reaches a target. The target district population is determined, before the construction of a district begins, as the total unassigned tract population divided by the remaining number of districts to be formed. The compactness of a district is achieved by appending the tracts that are closest to the seed tract. Once the construction of a district is complete, a new seed is determined and the above steps are repeated. The procedure stops when all unassigned census tracts are accounted for.

Vickrey's method may produce enclaves or residual districts, but the probability of this happening is reduced considerably by selecting seeds that are furthest from the initial reference tract. To avoid the occasional occurrence of enclaves, Vickrey suggests introducing a rule to the procedure that would not allow a census tract, whose removal from the set of unassigned tracts will make this set non-contiguous, to be added to the current district.

Vickrey further suggests that the above procedure can be modified to respect the boundaries of political sub-divisions. In this case, an entire political unit, such as a town or county, is included in the district, with attention to the district's resulting population.

Two other studies that provide a heuristic approach appear a few years later by Weaver and Hess [60] and Hess et *al.* [27], who consider the same three criteria. Both studies provide the location-allocation type heuristic that was briefly described in Section 3.1, while the latter provides an IP formulation as well (the formulation on page 31). However, the authors report that no realistic size problem can be solved at that time using the IP formulation, and as a result they resort to the location-allocation procedure detailed below. This procedure is also used later by Mills [38] for districting Bristol County in the United Kingdom and by Morrill [40, 41, 42] for redistricting Washington in the United States of America. Plane [50] provides an extension to the procedure by considering two quadratic functions of compactness (discussed in Section 3.1) instead of the moment of inertia one.

With the location-allocation approach, first an initial set of 'trial' district centers (LDs) is identified, and the distances d_{ij} between population units (EDs) and LDs are computed. Next, EDs are assigned to LDs by solving a transportation problem. This is done so as to minimize the sum of the moments of inertia of EDs about the trial district centers. The resulting solution may have an ED split among several LDs, in order for the algorithm to maintain equal district populations. Consequently, such split EDs are reunited into one in a way that an entire ED is assigned to the LD that has the largest share of that ED's population. Given these re-assignments, the next step is to calculate the new population center of gravity and the moment of inertia of each district. If the calculated centers of gravity are different from the trial ones, the above procedure is repeated with this new set of calculated district centers. The procedure stops when there is no change from one set of ED assignments to the next. The sequence of district plans that are so generated are further checked for contiguity and the ones that are non-contiguous as well as the ones that are dominated (i.e., the ones that are less compact and have higher population deviations) are discarded. The entire process is repeated until a desired number of alternative plans are generated.

Weaver and Hess [60] applied their algorithm to two counties of Delaware. Even though they were able to produce more compact plans with less violation of population equality than the plans proposed by the Senate, none of these plans were legislated. Hess et *al.* [27] report another application of the same heuristic to Delaware's counties in a few years later, after which they received a request from the Federal Courts for districting of Connecticut.

Another districting effort in the 60s came from Harris [26]. Harris considers the same three criteria, but measures compactness differently. The compactness measure he uses is the absolute difference between the length and the width of the bounding rectangle of a district, summed over all districts. Consequently, the districts are forced to look like squares. Harris carries out his procedure in two stages. In the first stage, he places a grid over the districting area and estimates the population in each cell of the grid. Next, using the grid cells, he forms districts that are as close to squares as possible. The population equality of the districts is maintained by having each district population within a specified range around the average. In the second stage, he modifies the districts so that the boundaries follow geographic, political or community boundaries. He demonstrates his method on 4-district Colorado data, for which he generates a district plan by hand.

Nagel [44, 45] is the only researcher who, in the same era, considered and implemented political criteria, such as districting on a partisan or bi-partisan basis. In [44] he even provides the statements of his program written for an IBM 7090 computer. His method is based on modifying an existing plan by means of transfers and swaps that do not violate contiguity and that also improve the objective function. Nagel uses an objective function which is initially the product of the following factors:

- number of districts whose population is outside the allowable range,
- average variation from population equality, and
- average variation from compactness.

The transfers and swaps are implemented until all district populations are within the allowable range, after which the algorithm starts considering political criteria. Specifically, it maximizes proportionality (based on the results of the last election) or depending on the user preference, it maximizes the number of districts won by a party. The algorithm stops and reports the best plan found when no more improvements can be made on the objective function.

Nagel [44] applied his method to the districting of a part of Illinois. He used 90 counties to form 18 districts. The maximum deviation from average population was consequently reduced from 39% to 17%. He further used his method in [45] for California, Mississippi, Virginia and Illinois again. His method was also used by Kaiser [32], who excluded political criteria and modified the original compactness measure. In an Illinois districting problem of 101 BUs to divide into 12 districts, Kaiser ran the algorithm with 200 random starting plans and reported successful convergence to local optimality.

The three studies by Thoreson and Liittschwager [57], Gearhart and Liittschwager [16] and Liittschwager [34] provide enhancements of the algorithms developed by Vickrey and Harris. The first one of these suggests the use of multiple random starts for Vickrey's algorithm. For Harris' method, the authors propose a different way of adding BUs to the districts. For both methods, they propose use of two other measures for population equality and two other measures for compactness. They test the revised algorithms for a redistricting problem in Iowa. The district plans they produce outperform the plan proposed by the Senate.

The second study by Gearhart and Liittschwager [16] further enhances Vickrey's algorithm by introducing new rules for assignment of BUs to districts. They further improve the algorithm so that the possibility of enclaves occurring is highly reduced. The last study is by Liittschwager [34], who adds new modules to Vickrey's algorithm to avoid enclaves completely. Two further refinements added by Liittschwager is to enable the user to modify part of the final district plan and to implement swaps to reach better plans.

Among the algorithms developed in 60s and 70s for solving the districting problem, the only exact algorithm is by Garfinkel and Nemhauser [15]. These authors provide an IP formulation (the one on page 32) which is used as the basis for determining the optimal district plan. They consider the three basic criteria, but also state that political criteria can successfully be incorporated into their methodology.

Garfinkel and Nemhauser's approach is a two-phase one. In Phase I, they construct a set S of eligible districts, which satisfy the four requirements of feasibility listed on page 32. The authors enumerate all possible feasible districts by a tree search method. Following the branches of an enumeration tree, BUs are added to districts one by one, with attention to the four conditions of feasibility. The resulting number of districts is usually too large, and the authors recommend modifying the feasibility conditions to reduce the size of S. As they construct districts in Phase I, they also create a BU-district 'incidence' matrix and perform reductions on this matrix (through row and column dominance) which consequently reduces the size of S.

The second phase is the place where m districts are chosen from set S, such that every BU appears in one and only one district, and all BUs are accounted for. The factor c_j in the objective function (see formulation on page 32) is the amount of absolute deviation of district j's population from the district quotient. Consequently, the objective is to minimize the maximum c_j value over all districts. The optimal combination of districts is found by means of a tree search algorithm. At each node of the tree, this algorithm adds another district from S to the district plan, by giving priority to the districts with lower population deviation. The algorithm is also capable of identifying multiple optimal solutions.

Garfinkel and Nemhauser applied their technique to hypothetical districting problems in the states of Delaware, Washington and West Virginia. With the first data set, they found that if S is small in size, no feasible district plan may exist. With the second set, they generated many more districts and then they were able to find solutions for three of the four runs they performed. For the third set, the problem was relatively large and the algorithm was not able identify a district plan in a reasonable amount of time. While the algorithm did not work effectively for some of these problems, a very recent study by Mehrotra et *al.* [37], which improves upon Garfinkel and Nemhauser's technique, reports a much more successful implementation. The details of this study are provided later in this section.

A considerably different approach for solving the problem is proposed by Tobler [58]. Tobler introduces a geographical transformation where the map of BUs is converted into a new map where each BU is enlarged or shrunk in relation to its population. The adjacency relationships between BUs are also maintained. Consequently, the problem of forming equal population districts becomes the problem of creating equal area districts. Tobler also considers contiguity and to some extent compactness. The only problem with his approach is that a compact district plan on the new map is not guaranteed to remain compact when its transformed back to the original map.

The next study in the chronological order is by Bourjolly et *al.* [4]. These authors also consider the three basic criteria, but they include integrity of communities in their criteria set as well. They use distribution of income throughout the City of Montreal as a proxy for defining communities. The method they propose starts with an initial plan, which could either be the existing plan or a plan generated by randomly choosing seeds and allocating each BU to the nearest seed by solving a p-median problem. It then moves iteratively from one plan to the next by transferring BUs one at a time between adjacent districts. The objective function employed by the algorithm combines all the criteria (except contiguity) as a single weighted function. Given a particular set of weights, the algorithm then identifies and implements in each iteration a BU transfer that improves the objective function the most.

Another study on districting is by Marlin [36], who considers the problem in a context of assigning salespeople to districts. With this setting, the objective is to form districts that are balanced in terms of total 'workload' assigned, and that are also compact so that the total travel cost is low. The resulting model is similar to the model on page 33, but the first constraint is relaxed as a greater-than-equal constraint, and the second constraint allows deviations from average district workload. Marlin [36] recognizes a slight modification of this model as a transportation linear program and consequently employs the algorithm by Ford and Fulkerson [13] to solve it as a minimum cost network flow problem. Marlin uses his approach to solve large-scale instances of the problem in the Milwaukee region.

A study that appeared in late 80s is by Fleischmann and Paraschis [12], who focus on a 168-district salesman districting problem for a German consumer goods manufacturer. As described in Section 3.1, these authors use a modified compactness measure which is a function of the unknown north and east corners of the district centers. As a result, the authors employ an iterative location-allocation heuristic for solving the problem. As in Weaver and Hess [60], they also take care of split BUs before moving on to the next iteration. Specifically, they use a transportation-based heuristic to reduce the number of splits. However, since they observe that many splits remain unresolved with this approach, they develop a new heuristic procedure for this step. The results of their implementation on the 168-district problem are reported to have perfectly satisfied the company management.

A study by Browdy [5] proposes a Simulated Annealing (SA) algorithm for solving the political districting problem. This is a relatively new technique that is usually considered in the category of modern heuristic techniques. It has analogies with the physical process of cooling down a hot metal slowly, so that the molecules can adjust to each level of energy possessed by the metal. The energy of the metal corresponds to the objective function of the problem, and the algorithm attempts to reach lower energy levels (i.e., improved values of the objective function) in an iterative manner. It occasionally allows the energy to increase from one iteration to the next, but with less and less frequency as the algorithm proceeds. Eventually, it reaches a stable energy level, where the algorithm is said to have reached a local optimal solution.

In Browdy's SA implementation, energy is computed as the sum of compactness, contiguity and population equality. Compactness is measured as the sum of the population moments of inertia of districts; contiguity is measured as the total number of non-contiguous districts; and population equality is measured as the total absolute deviation from average population. The last two terms are further multiplied by weights λ_1 and λ_2. The algorithm starts with an arbitrary solution (the author suggests using the existing district plan), and generates new solutions in an iterative manner by changing random pieces of the districts (e.g. moving BUs between districts). It always accepts improving solutions, but it occasionally accepts non-improving ones as well. Unfortunately, the author does not develop a computer program for his method, nor does he apply it to a real districting problem. Therefore, it is not possible to comment on the practicality of this algorithm on solving real problems.

Another recent districting study is by Horn [29], who also considers population equality, contiguity and compactness, as they are the relevant criteria in Australian legislation. Horn formulates the problem as one that maximizes compactness subject to contiguity and population equality. Compactness is measured as the sum of 'edge weights' between pairs of BUs that comprise a district, summed over all districts, where the edge weight for a particular pair of BUs is the length of the common boundary between the two BUs.

The algorithm that Horn employs is a 'transition-based' one. It starts with an initial plan and modifies this plan in an iterative manner by means of the following operations:

- transfer a single peripheral BU from its current district to an adjacent district,
- combine two adjacent districts into one,
- detach an arbitrary peripheral BU from a district and make it a district by itself.

Clearly, these operations can make a district plan non-contiguous easily, even if the initial plan was contiguous. What is unique in this study is that a district plan is never referred to as non-contiguous, but the number of districts is considered to have changed. For example, the second operation reduces the number of districts by one, and the third operation increases it by the number of resulting contiguous pieces of the *from* district minus one. Consequently, the number of districts is variable during the course of the algorithm, but it is always maintained to be greater than the required quantity. The difference is used to penalize the objective function. The algorithm also allows population equality to be violated, but the allowable amount of violation is gradually reduced to zero. Finally, the algorithm moves from one plan to the next by accepting the plan that improves the objective function the most. It stops when no more improvements can be achieved.

Horn applies his technique to the districting of seven municipalities in the central North Shore of Sydney. He experiments with various parameters of his algorithm, as well as with varying levels of the allowable % deviation from population equality. Unfortunately, he does not compare the district plans he produce with those that are currently in use or those that he could produce using the algorithms proposed by other researchers.

The most recent study available in this area is by Mehrotra et *al.* [37], which is based on the general framework in Garfinkel and Nemhauser [15]. Different from [15] is that the proposed technique is an optimization-based heuristic as opposed to an exact method. The authors consider the three basic criteria only, so as to make the districting process free of political criteria and therefore free of gerrymandering claims. They use the formulation on page 32, but the cost term c_j is modified to represent the penalty cost that measures the 'non-compactness' of a district. For the purpose of computing c_j, the authors recognize a district as a connected graph whose vertices are the BUs that comprise the district, and the edges represent geographic adjacency. Consequently, a 'center BU' is identified as the node that has the smallest sum of the number of BUs on the path to each other BU in the district. This sum is then used as the 'cost' of the district, c_j.

Similar to the Garfinkel and Nemhauser approach, the authors attempt to construct a set S of feasible districts as input for the IP formulation. Since enumeration of all possible districts is extremely costly, they "generate them on an 'as needed' basis" (Mehrotra et *al.* [37], Section 2.2) using a technique known as *column generation.* They start with an initial set of districts, which is constructed using the districts in the old plan and/or using a heuristic similar to the method of Vickrey [59]. Next, they solve a column generation sub-problem by means of a branch-and-price technique to expand the set S. If S is expanded, the sub-problem is solved again to see if other districts of higher quality can be identified. This process is repeated as long as new districts are added to S. The authors state that solving the sub-problem "is both theoretically and practically hard" (Mehrotra et *al.* [37], Section 3.5), therefore they use heuristic approaches while solving the sub-problem (e.g. while evaluating branches). Furthermore, the way they maintain contiguity during the construction of S may exclude some of the contiguous districts as a side effect. These characteristics in turn make their algorithm a heuristic technique.

Mehrotra et *al.* [37] apply their technique to a South Carolina case study. They produce district plans that have maximum deviations from the average district population as low as 2%. They report their results for the cases with and without a pre-processing that involves combining the counties in order to reduce the problem size and/or breaking them in order to generate more compact plans. They further allow a post-processing stage where districts are slightly modified to achieve even more balanced district populations. They conclude that their "optimization methodology provides an effective way of generating high quality districting plans" (Mehrotra et *al.* [37], Section 5). Overall, this study appears to propose the most effective heuristic to date for solving the political districting problem.

Chapter 4

Basics of Tabu Search

The algorithms presented in Chapter 5 for solving the political districting problem are based on the principles of a modern problem solving technique known as *Tabu Search* (TS). In this chapter, a brief history and motivation as well as the basic features and the mechanism of this technique will be provided. More extensive information on TS methodology for the interested reader is available from a book chapter by Glover and Laguna [22], an article by Glover [21], and a very recent book on Tabu Search by Glover and Laguna [23].

The origins of the Tabu Search technique go back to 1960s and 1970s when some of its elements appeared in independent studies that involved non-traditional ways of solving difficult problems. But it was not until late 1980s that the technique settled into a well-defined structure. Tabu Search is usually referred to as a 'modern heuristic technique' because of the fact that it emerged in the same recent era of non-traditional problem solving approaches such as Genetic Algorithms and Simulated Annealing. While the latter algorithms are exclusively based on the principles of biological and physical processes, "... The philosophy of tabu search (TS) is to derive and exploit a collection of principles of intelligent problem solving." (Glover and Laguna [23], p.1). The TS technique makes use of effective tools that combine artificial intelligence with optimization, in an effort to improve upon traditional problem solving methods. The main motivation behind the appearance of TS and other intelligent problem solving techniques is clearly the need to solve larger and increasingly more complex real-world problems.

The Tabu Search methodology is primarily based on the principles of neighborhood search. What makes it different from the traditional methods is its intelligent utilization of memory and its ability to cross the boundaries of feasibility. These features help a TS algorithm search the solution space of a particular problem more thoroughly, reaching solutions that the traditional algorithms are not usually able to reach. Through 'diversification', a TS algorithm visits radically different parts of the feasible region and through 'intensification', it focuses on 'elite' solutions and their 'neighbors'. These features coupled with the intelligent use of memory are generally known as 'adaptive memory' in TS terminology. The following sections further detail these features and other elements of the TS technique. But first, the principles of neighborhood search must be reviewed.

4.1 Neighborhood Search

The first step in understanding the mechanism of a TS algorithm is to understand the basics of neighborhood search (NS). To begin with, consider a sample combinatorial problem of choosing p out of n alternatives. A solution $X = \{x_1, \ldots, x_p\}$ to this problem is a subset of the set of alternatives $J = \{1, \ldots, n\}$ with $p < n$ and $|X| = p$. A *neighborhood* of X, denoted by $N(X)$, is defined to be the set of all solutions obtained from X by changing one or more components of X. To give an example for the sample problem, $N(X)$ may be defined as the set of all solutions obtained by replacing an index $x_k \in X$ with another index $x_l \in J \setminus X$. That is,

$$N(X) = \{X' : X \setminus X' = \{x_k\} \text{ and } X' \setminus X = \{x_l\}\}$$

We say that a solution $X' \in N(X)$ is *reachable* from X.

Given the above definition of neighborhood, a NS algorithm starts with an initial solution X_0, and moves from one solution to the next in an iterative manner. Typically, the solution to be visited next is selected from the neighborhood of the current solution. That is, the solution X_t that will be visited in iteration t (t$\geq$ 1) is selected from $N(X_{t-1})$. The algorithm visits solutions one after another in this way until a stopping criterion is met. A generic outline of this simple search procedure is given below:

<u>NEIGHBORHOOD_SEARCH</u>

1. Identify a starting solution X_0; set $t = 0$.
2. Repeat until a stopping criterion is satisfied:
   - 2.1. Set $t \leftarrow t + 1$.
   - 2.2. Identify a solution $X' \in N(X_{t-1})$.
   - 2.3. Set $X_t \leftarrow X'$.

The above algorithm is too general to implement in its current form. Several aspects need further clarification. Perhaps the most important one of these is the selection of X' from the neighborhood $N(X_{t-1})$. Typically, a NS algorithm selects an *improving* solution as the next solution to visit. For example, in a minimization problem, the first solution X' that satisfies $f(X') < f(X_{t-1})$ might be selected. If there is no such solution in the neighborhood, then the algorithm terminates in iteration t with X_{t-1} reported as the best solution found (an alternative termination rule is to run the algorithm for a fixed number of iterations). With these settings, the algorithm is an example of what is known as a *descent* algorithm. Alternatively, one may select the $X^* \in N(X_{t-1})$ that improves the current solution the most. That is, select X^* so that $f(X^*) = \min_{X \in N(X_{t-1})} f(X)$. This particular version of the algorithm is an example of a *steepest descent* algorithm.

Another component of a NS algorithm is the construction of the starting solution X_0. This is usually done randomly, i.e., without any particular attention to solution quality. This provides a quick start to the algorithm, with the hope that higher quality solutions can be identified in later iterations.

A NS algorithm can be trapped in a local optimum easily. This happens when there is no solution in the neighborhood to improve the current solution, but there are solutions elsewhere in the solution space better than the current solution. Using multiple random starts is usually a remedy to this problem. With each random start, the initial solution is used as the 'seed', hopefully from a different part of the solution space, to reach other solutions.

The last component of a NS algorithm is the definition of its neighborhood structure. A neighborhood could be as simple as the one in the above example, or may involve more complex operations. There are three major criteria to keep mind while defining a neighborhood:

- the neighborhood should contain sufficiently many solutions, i.e., the algorithm should never stop due to lack of solutions in the neighborhood.
- the neighborhood should not be too large that its exploration is time-intensive. Remember that the neighborhood must be constructed in every iteration and objective function values need to be computed for each solution in the neighborhood (or until an improving solution is found, in the case of a descent algorithm).
- the neighborhood definition should allow (at least theoretically) each solution to be reachable from each other solution by executing a finite number of iterations.

While the first two of the above criteria are easy to check (by measuring the size of the neighborhood via experimentation), the last one needs a theoretical proof. This may or may not be possible for a particular neighborhood definition, but usually, the larger the neighborhood is, the more likely it is that the last requirement will be satisfied.

A Tabu Search algorithm uses essentially all of the above basics of neighborhood search. What makes TS different and superior to plain neighborhood search is its use of additional features, mostly through use of adaptive memory for diversification and intensification. These features are described in the remaining part of this chapter.

4.2 Moves

In Tabu Search terminology, a *move* is the basic operation by which a solution in $N(X)$ will be reached from the current solution. For the n-choose-p example from the previous section, a move would be the exchange of two indices $x_k \in X$ and $x_l \in J \setminus X$. What makes moves critical in a TS algorithm, as far as the algorithm mechanics are concerned, is what is referred to as the *attributes* of moves. An attribute of a move is a piece of information that describes the move, either by itself or together with other attributes. For the above example, it is possible to define two attributes of a move: the index $x_k \in X$ being removed from X, and the index $x_l \in J \setminus X$ being included in X, as a result of the exchange operation. The two attributes collectively describe all the information as to how this particular move will be implemented. The most important feature of the move attributes is that they can be *tabu-active*. If a certain attribute (or a combination of attributes) of a move is tabu-active, the move is not allowed to be implemented. These concepts will be discussed in the next section in detail.

As for the size of the neighborhoods, consider again the n-choose-p example. Recall that a move is to exchange two indices $x_k \in X$ and $x_l \in J \setminus X$. With p choices for x_k and $n-p$ choices for x_l, there are $p(n-p)$ solutions in $N(X)$. For a hypothetical problem with $n = 300$ and $p = 10$, $N(X)$ would consequently consist of 2900 solutions, which may or may not be too much depending on how costly it is to compute the objective function value of a solution in $N(X)$. If the algorithm is reasonably fast in evaluating these 2900 solutions, then it can be modified to either run for more iterations or incorporate another type of a move. In the latter case, for example, one may consider the possibility of exchanging two pairs of indices instead of one. This would require construction and evaluation of $2 \cdot C(p, 2) \cdot C(n-p, 2)$ solutions to be included in $N(X)$ ($C(p, 2)$ denotes the number of p-choose-2 combinations).

For the $n = 300$, $p = 10$ problem, this results in $2 \cdot 45 \cdot 41\ 905 = 3\ 771\ 450$ solutions. This is probably too many in most cases. In this case, one might still be interested in including a small subset of these solutions in $N(X)$, for the sake of adding diversity to the search process. However, the availability of computational resources would in general determine what the appropriate size of $N(X)$ should be.

Parallel to the discussion in the previous section, moves should not be too restrictive that the resulting neighborhoods are dangerously small in size. This will restrict the choices used for selecting the next solution to visit. On the other hand, the resulting neighborhoods should not be so large that evaluation of the cost functions is computationally very expensive. This increases the time required for each iteration, therefore leaves the algorithm less time to complete a given number of iterations. What may be possible though, is to evaluate objective function values of the solutions incrementally, i.e., by only looking at the components that are changed and computing the change in objection function value accordingly. In other words, one should avoid, as much as possible, evaluating the objective function from scratch. If this is possible, the need for computation time will be less and the neighborhoods may be allowed to contain more solutions.

4.3 Tabu Definitions and Use of Memory

Perhaps the best-known aspect of a TS algorithm is the concept of *tabu*. Webster's dictionary defines "tabu" (or "taboo") as "set apart as charged with dangerous supernatural power and forbidden to profane use or contact..." or "banned on grounds of morality or taste or as constituting a risk...". In TS context, a tabu move is simply a forbidden move. Because the algorithm can accept, in each iteration, non-improving solutions as well, the tabu concept serves as a mechanism for avoiding cycles and also for diversifying the search process.

In Tabu Search, tabus are defined through use of move attributes. Typically, an attribute of a move constitutes the basis of how a move will be "declared tabu". To further explain this concept, consider the n-choose-p example again. A move defined by the exchange of two indices $x_k \in X$ and $x_l \in J \setminus X$ has two attributes: *index_out*, the index x_k being removed from X, and *index_in*, the index x_l being included in X. Therefore, the representation (*index_out*, *index_in*) is used to describe the move that involves exchange of x_k and x_l. Given this definition, one may choose to define a tabu in a way that forbids moves whose *index_out* (*index_in*) is equal to a particular value. In plain English, this means forbidding a particular index to leave (enter) solution X, no matter which other index it is exchanged with. Another way of defining tabu might simply be forbidding exchange of a particular pair of indices. This last tabu definition involves specifying values for both *index_out* and *index_in*. In all of these cases, the relevant attributes of the corresponding move are labeled *tabu-active* and solutions that result from moves with tabu-active attributes are excluded from the neighborhood (with certain exceptions).

Once a particular move is declared tabu, it keeps its tabu status for a certain number of iterations, which is referred to as *tabu tenure* and denoted by θ. Recall that the main functionality of the tabu feature is to avoid cycling. Therefore, the tabu tenure chosen for a particular move should be long enough that the algorithm will hopefully have moved to a significantly different solution by the end of θ iterations. In most cases though, it is difficult to prove that a particular tabu tenure selection prevents the algorithm from cycling for sure. For the first tabu definition example above, i.e., $index\_out = x_k$, having a tabu tenure of θ iterations means that the index x_k, once in X, will not be allowed to leave X for the next θ iterations. For the second example, i.e., $index\_in = x_l$, it means that x_l, once removed from X, will be allowed to enter X only after θ iterations have passed. For the last one, x_k and x_l, once they are exchanged, will not be allowed for another exchange for θ iterations. These tabu definitions are intended to avoid reversal of the exact order of exchanges. Selecting large tabu tenure values is likely to decrease the possibility of cycling, but it also reduces the number of available moves for choosing the next solution to visit. Therefore, we advise experimentation on the length of the tabu tenure.

The impact of imposing tabus on moves is to reduce the size of the neighborhood of a solution in a particular iteration, as mentioned previously. In other words, neighborhoods are dynamic and their composition depends on the moves implemented in previous iterations. Normally, the moves that are tabu are excluded from $N(X)$, which means that the next solution to be visited will be selected from $N'(X) = N(X, H)$, where H is the history of all previous solutions visited. The only exception is when the solution that results from a tabu move improves the best known objective function value, i.e., this solution is being visited for the first time. In this case, cycling cannot occur and therefore the tabu status of the move is overridden and the move is included in $N'(X)$. To keep track of which moves are tabu and which are not, a memory structure is employed. This is referred to as *recency-based memory*, since it involves storing information on most recent moves, and it is described in the next section with a numerical example.

4.3.1 Recency-based Memory

For the n-choose-p combinatorial problem, suppose tabus are defined based on the *index_in* attribute. That is, an index x_l that leaves X will not be allowed to return for θ iterations. To store the information needed to determine whether an index will be allowed to take part in an exchange in a future iteration, a single-dimensional array Λ of size n is sufficient. Typically, Λ_i will contain the iteration count at which x_i will be allowed to return to X. At the beginning, the values in this array are initialized to zero. Consider the following array for the example with $n = 10$, $p = 5$ and the current solution of $X_6 = \{1, 5, 6, 8, 9\}$ with $J \setminus X_6 = \{2, 3, 4, 7, 10\}$ (this means we are at the beginning of iteration 7). Suppose further that $\theta = 3$.

1	2	3	4	5	6	7	8	9	10
0	7	6	5	0	0	8	4	0	9

There are five non-zero values in the above array that we are interested in: the values that correspond to indices 2, 3, 4, 7 and 10. The value $\Lambda_2 = 7$ indicates that index 2 will be allowed back in the solution at the end of iteration 7, therefore any exchange that brings index 2 back into the solution is tabu until the end of iteration 7 ($\Lambda_2 = 7$ implies that index 2 has left the solution in iteration 4, since $\theta = 3$). Similarly, index 10 cannot return until the end of iteration 9. On the other hand, index 3 can return in the current iteration since the value $\Lambda_3 = 6$ indicates that the tabu status of all moves that involve index 3 has expired at the end of iteration 6.

Suppose index 3 is chosen to be exchanged with index 9. After this move is implemented, the corresponding tabu expiry information has to be recorded in Λ for index 9. That is, we must set $\Lambda_9 = 7 + 3 = 10$, where 7 is the current iteration count and 3 is the tabu tenure. This means that index 9 will be not be allowed to return until the end of iteration 10. Note that, this way of recording tabu status does not require any other updates of the array Λ. If, on the other hand, one chooses to record the number of iterations left until expiry (i.e., to record the value $\theta = 3$ in the iteration the move was implemented), then one would have to decrease this value for each index in each iteration. Clearly, the former way of recording tabu information is computationally more efficient, since it requires performing comparisons only, as opposed to performing comparisons *and* update with the latter.

In the above example, we assumed a fixed tabu tenure of 3 iterations for all indices. Observe that this will cause three indices in $J \setminus X_{t-1}$ to be tabu-active at any given time. This leaves the algorithm with 2 indices that can be exchanged with 5 others, which amounts to a total of 10 non-tabu moves. In other words, because of the way tabus are defined (including the tabu tenure), 15 of the 25 available moves are eliminated. These numbers (15 and 25) which are based on a small numerical example may not apply to full-scale real-world problem. However, with any tabu definition, one should keep in mind that tabu definitions may sometimes be too restrictive and reduce the size of $N(X_{t-1})$ significantly. This would leave the algorithm with too few alternatives to choose from, and therefore may negatively affect the diversity of the search process.

An immediate aspect of the tabu concept is the selection of the tabu tenure. In the above example, the fixed tabu tenure of 3 iterations was arbitrarily selected. This would prevent an index that is deleted from the solution from returning to the solution for 3 iterations, but it may be the case that any index that leaves will tend to come back as soon as the 3 iterations pass. This is likely to result in a cycle that repeats itself every 3 iterations, with the same pairs of indices being exchanged. To avoid this, it is recommended that the tabu tenure be different for each move implemented. One way of doing this would be to generate a random tabu tenure from a range $[\theta_{min}, \theta_{max}]$. This would help avoid the cycles that may result from assuming a fixed tabu tenure.

The recency-based memory structures described in this section is an example of what is known as *short-term memory*. There are numerous variations and implementations of short-term memory which are beyond the scope of this section. The examples provided in this section are intended only to illustrate the concepts. Interested readers are referred to the book by Glover and Laguna [23] for further details.

Another dimension of using memory in an intelligent search is to use *longer-term memory*. A key component of this concept, namely the *frequency-based memory*, is described in the next section.

4.3.2 Frequency-based Memory

Frequency-based memory is a major element of the longer-term memory concept that complements short-term memory. Even though TS implementations for some applications have proved successful without frequency-based memory and other elements of longer-term memory, these tools provide an additional dimension of diversification and intensification, and improve the performance of a TS algorithm. In this section, the frequency-based memory aspect of longer-term memory will be detailed.

Frequency-based memory is one of the TS components that coordinate diversification of the algorithm over the search space. Typically, information regarding certain attributes of moves are recorded over the entire length of a TS run, which is continuously used to diversify the search into unexplored parts of the solution space. Frequency-based memory uses two general types of measures: *transition measures*, which use the number of times an attribute changes from one iteration to the next, and *residence measures*, which use the number of times a particular value for an attribute was observed as part of the solutions visited. In both cases, the quantity is normalized typically by the total number of occurrences (e.g. total number of iterations).

Let us use the n-choose-p problem again to make these concepts more concrete. A transition measure for the frequency-based memory might be based on the number of times two indices were exchanged. In this case, one would need a two-dimensional array to represent pairs of indices (the lower part of this array is not used since the order of indices is not important). Each cell of this array would contain the number of times the corresponding pair of indices were exchanged. This quantity would then be normalized by the number of iterations that have been performed so far. The 10x10 array Γ below illustrates the implementation of this concept without the normalization. (Suppose we are at the beginning of iteration 11.)

	1	2	3	4	5	6	7	8	9	10
1		0	0	0	0	0	0	1	0	0
2			0	0	4	0	0	0	0	0
3				0	0	0	0	0	1	0
4					0	0	1	0	0	0
5						0	0	1	0	0
6							0	0	0	0
7								0	2	0
8									0	0
9										0
10										

Here the value $\Gamma_{25} = 4$ indicates the indices 2 and 5 were exchanged 4 times in the past 10 iterations (the total of all non-zero values in the array must equal 10). Consequently the frequency of this exchange is measured as 40%. The purpose of computing such a quantity is to avoid frequent exchanges in later iterations. This is done by penalizing the objective function value of a solution that results from exchanging indices 2 and 5.

Before explaining how the objective function values are penalized according to frequency-based memory, consider the following array for a residency measure example.

1	2	3	4	5	6	7	8	9	10
5	5	3	4	8	5	5	6	5	4

The residency measure used in this case is the number of times an index was part of the solution. A one-dimensional array is sufficient to store this information. The values in the above array indicate that index 5 was *in* the solution in 8 of the past 10 iterations (80% of the time). If a candidate exchange in iteration 11 still leaves index 5 as part of the resulting solution, the objective function value of this solution is penalized using the 80% frequency. This is the index that was part of the solution more than any other index, therefore the measure suggests that other indices should be given more chances to enter X.

The two simple examples above indicate how the frequency-based memory is utilized in order to diversify the search process, that is to visit solutions that have not been visited before, by discouraging visitations to solutions that are similar to those encountered previously. Their impact on the objective function values is to alter them in a way that reflects the history of iterations. For the first example above, let $X_{ij} \in N(X_{10})$ denote the solution that is reached from X_{10} by exchanging indices $i \in X_{10}$ and $j \in J \setminus X_{10}$. The penalized objective function value of X_{ij} might be computed as $f(X_{ij}) \cdot (1 + \frac{\Gamma_{ij}}{10})$. Since the ratio $\frac{\Gamma_{ij}}{10}$ is always between zero and one, $f(X_{ij})$ is magnified as a result. In other words, the solution X_{ij} is discouraged by a penalty factor proportional to the number of times the two indices i and j were exchanged.

For the second example, if a candidate exchange of indices i and j makes an index k part of the resulting solution X_{ij} (k not necessarily distinct from i or j), then $f(X_{ij})$ is modified as $f(X_{ij}) \cdot (1 + \frac{\Gamma_k}{10})$. That is, any move that results in index k to "reside" in X_{11} at the end of current iteration is penalized proportional to index k's residency frequency. If the penalty factor is too disruptive to the process, i.e., it is so high that the quality of solutions measured by f itself has no significance, then a scaling factor can be used to decrease the magnitude of the penalty term. It may also be worthwhile to include another scaling factor which is a function of the problem size, such as n in the above examples.

The above examples illustrate how frequency-based memory can be utilized to diversify the search for good solutions. Other elements of longer-term memory include such topics as exclusive intensification and diversification, strategic oscillation and path relinking. These topics are beyond the scope of this chapter, and the interested reader is again referred to the book by Glover and Laguna [23] for details.

4.4 Aspiration Criteria

In a TS algorithm, a tabu move is normally forbidden for the duration of the tabu tenure. However, there are certain exceptions to this rule. The term *aspiration criteria* is associated with these exceptions where the above rule can be violated.

The primary purpose of introducing tabu structures to a neighborhood search algorithm is to prevent it from cycling, i.e., visiting the same sequence of solutions over and over again. When an algorithm is considering to visit a solution that was never visited before, clearly there is no danger of cycling. In this case, this solution can be accepted even though the corresponding move is tabu. Of course, to check whether each solution in $N(X)$ was already visited can be time-costly and is probably unnecessary. The algorithm, on the other hand, keeps track of the best solution found at all times. At least this information can be used to check whether a solution is being visited for the first time or not. This is done by comparing the objective function value of the best known solution with that of the solution the algorithm is considering to visit. If the latter has a lower objective function value, it can be accepted even if it is tabu, since no cycling can occur. To summarize, encountering solutions that are better than the best known solution is an aspiration criterion that can be used to override tabu status. This particular criterion is known as *Aspiration by Objective - Global form.*

Another case where aspiration takes place is when all available moves are tabu. In this case, the algorithm chooses a move that is 'least tabu'. This means that the move to be selected is closest to the expiry of its tabu status. This aspiration criterion is known as *Aspiration by Default.*

A third aspiration criterion is referred to as *Aspiration by Search Direction* and as the name implies, it is based on the concept of *search direction.* The search direction of a TS algorithm in a particular iteration is said to be *improving* if the last move implemented has improved the objective function, and *non-improving* if the last move implemented did not improve the objective function. Consequently, a tabu move can be accepted under this aspiration criterion if the search direction does not change as a result of implementing this move. However, this criterion should be used with more caution than others, since cycling is of higher possibility. The search direction can be non-improving for t iterations after which it turns into improving for another t iterations, reversing the moves that were implemented.

The aspiration criteria described above can be used alone or in combination with others in a TS algorithm. If more than one criterion are used, they should be prioritized. For example, Aspiration by Objective is likely to be used first, after which Aspiration by Search Direction and Aspiration by Default are used in that order.

4.5 Stopping Rules and Other Features

The final element of a TS algorithm is the issue of stopping rules. Since a TS algorithm accepts non-improving solutions in addition to improving ones, the traditional rule of stopping when no more improvement is possible is no longer appropriate. Instead, a typical TS algorithm stops based on the following two rules (whichever occurs first):

- when the algorithm cannot improve the best known solution for a pre-determined number of successive iterations, T.
- when the algorithm reaches the maximum limit on the total number of iterations, t_{max}.

Both rules use one parameter: T and t_{max}. If T or t_{max} is too small, the algorithm will terminate before finding a sufficiently good solution. If they are too large, computational time will be wasted for iterations that do not improve the best known solution. But the first rule is superior to the second in that it lets the algorithm run as long as there is improvement. The second rule, on the other hand, may cause the algorithm to terminate in the middle of a very promising sequence of improving moves.

One feature of TS methodology that is not mentioned so far is its ability to cross the boundaries of feasibility in its search process. That is, the algorithm is allowed to temporarily visit infeasible solutions with the expectation that switching back to feasible solutions in later iterations may start a good thread of high quality solutions. While there is nothing that prevents a traditional NS algorithm from doing the same, it is with the TS technique that this feature has been introduced and has become popular in heuristic optimization literature. This feature becomes especially useful towards the end of a TS run, where a small change in the solution structure would cause an infeasibility, and a traditional algorithm would not accept this step that violates feasibility. Visiting infeasible solutions temporarily may cause the algorithm to cross a bridge towards a significantly different part of the feasible region. This clearly adds to the level of diversification in the search process.

Neighborhood Search heuristics (such as TS) have a distinct advantage over mathematical programming approaches: They are able to handle complex expressions (in the objective or in the constraints) which may be difficult (or impossible) to incorporate in a mathematical programming model. All that matters is computational time, i.e., time needed to evaluate complex objective function values or to make sure that all constraints are satisfied. For this reason, a TS algorithm is capable of handling very complex real-world constraints. Also, TS algorithms do not have the computational overhead that, for example, Genetic Algorithms have. A Genetic Algorithm has to manage chromosomes in the population, crossovers and the resulting update of parent chromosomes with child chromosomes, while a TS algorithm simply has to evaluate and implement moves. A TS algorithm can also be easily parallelized, in connection with the Probabilistic Diversification and Intensification algorithm described in Chapter 6.

4.6 A Generic Outline of a TS Algorithm

The following is the outline of a generic TS algorithm based on the principles described in Sections 4.1-4.5.

GENERIC_TS

1. Construct a starting solution X_0. Set $X^* = X_0$, the best known solution.
2. Initialize recency- and frequency-based memories. Set $t = 1$.
3. Repeat until a stopping rule is satisfied:
   - 3.1. (iteration t) Construct the neighborhood $N(X_{t-1})$. Let $f(X)$ denote the objective value of a solution $X \in N(X_{t-1})$.
   - 3.2. Penalize $f(X)$ for all $X \in N(X_{t-1})$, using the frequency-based memory information specific to the move that converts X_{t-1} into X. Let $f'(X)$ denote the resulting (penalized) objective function value.
   - 3.3. Sort the solutions $X \in N(X_{t-1})$ is ascending order of $f'(X)$.
   - 3.4. Traversing solutions from the top of the sorted list, pick the first solution that is non-tabu, or is tabu but satisfies one or more of the aspiration criteria. Denote this solution by X'.
   - 3.5. Set $X_t = X'$ and update recency- and frequency-based memories using the attributes of the move implemented.
   - 3.6. Set $X^* = X_t$, if X_t is a better solution than X^*, i.e., $f(X_t) < f(X^*)$.
   - 3.7. Set $t \leftarrow t + 1$.

4. Report X^* as the best solution found.

4.7 Applications

Since the appearance of the paper by Glover [20] that first defined and used the term "tabu search", the technique has been applied on a variety of problems. It became very popular over the last decade, as a result of many successful applications. In this section, a representative subset of these applications will be summarized.

One major area that TS has been applied extensively is the class of scheduling problems. Glover and Laguna [23] report such applications as scheduling in manufacturing systems, scheduling a flow-line manufacturing cell, audit scheduling as well as areas like forest harvest scheduling with a variety of constraints. Other application areas in scheduling include just-in-time scheduling, single-machine scheduling, job shop scheduling and resource scheduling.

Another typical application area for TS is the class of transportation and vehicle routing problems. Tabu Search has been used to solve many different types and versions of these problems including the fixed charge transportation problem, traveling salesman problem, vehicle routing problem, vehicle routing problem with time windows, and routing and distribution problems. The vehicle routing application by Rochat and Taillard [53] that uses the Probabilistic Diversification and Intensification technique also falls into this category.

Other applications of Tabu Search include: telecommunications applications such as the design of a telecommunications network; graph-theory applications such as p-median, graph partitioning, maximum clique; financial applications such as portfolio management and investment problems. For an extensive list of TS applications that have been developed over the last ten years, a recent survey by Osman and Laporte [47] is a good source.

Chapter 5

The Proposed TS Algorithm

In this chapter, the Tabu Search algorithm that is used to solve the political districting problem is described. Various components of the algorithm include construction of a starting solution, the evaluation of the objective function, the moves, tabu definitions and the memory structures used. These features will be described in detail in the following sections.

5.1 Notation

5.1.1 Base Units

n	number of base units in the geographical region being districted,
I	index set for base units, $I = \{1, \ldots, n\}$,
i, k	subscripts for base units ($i, k \in I$),
a_i, r_i, p_i	area, perimeter and population of base unit i,
t_{ik}	length of the common boundary between two adjacent base units i and k.

5.1.2 Districts

m	number of districts to be formed,
J	index set for districts, $J = \{1, \ldots, m\}$,
j, l	subscripts for district ($j, l \in J$),
D_j	index set of base units contained in district j ($D_j = \{i : i \in I, x_{ij} = 1\}$),
A_j, R_j, P_j	area, perimeter and population of district j,

$\bar{A}_j$	partial area of district j formed as a result of overlays.

5.1.3 Entire Territory

A, R	total area and perimeter of the territory being districted,
P	total population residing in the districting area ($P = \sum_{i \in I} p_i$),
$\bar{P}$	ideal average for district population ($\bar{P} = P/m$),
β	allowable % deviation from $\bar{P}$,
$[P_{min}, P_{max}]$	allowable range for district population ($P_{min} = (1-\beta)\bar{P}$, $P_{max} = (1+\beta)\bar{P}$).

5.1.4 Decision Variables, Objective Function

$$x_{ij} = \begin{cases} 1 & \text{if base unit } i \text{ is assigned to district } j \\ 0 & \text{otherwise} \end{cases}$$

$X = \{x_{ij}\}$	a solution to the problem,
$N(X)$	Neighborhood of X,
$f(X)$	multi-criteria objective function,
$h(X)$	penalty term for violation of population balance,
w'	weight of $h(X)$,
$f_i(X)$	objective function term for criterion i,
w_i	weight of criterion i,
$f'(X)$	objective function modified by frequency-based memory,
$C_i(X)$	compactness of solution X, computed using measure i (i=1,2),
$C_{2j}(X)$	compactness of district j of solution X, computed using measure 2.

5.1.5 Algorithm-Related

α	penalty coefficient for population balance violations,
$\mu, \bar{\mu}$	parameters for updating α,
k	index for political parties,
Q	total number of political parties,
S_k	percent seats won by party k,
V_k	percent vote received by party k,
$v_{kj}(X)$	percent vote received by party k in district j,

$J_k^W(X)$	index set for districts won by party k,
$J_k^L(X)$	index set for districts lost by party k,
$G_k(X)$	proportionality score of party k based on solution X,
t	TS iteration count,
T	maximum number of successive iterations without improvement,
F_1	objective value of best known feasible solution,
F_2	best objective value encountered (feasible or infeasible),
(i, j, l)	representation of a Type I move,
(i, k, j, l)	representation of a Type II move,
M_t^1, M_t^2	set of eligible Type I and Type II moves in iteration t,
X_{ijl}	solution that results from the implementation of Type I move (i, j, l),
X_{ikjl}	solution that results from the implementation of Type II move (i, k, j, l),
θ	tabu tenure,
$[\theta_{min}, \theta_{max}]$	allowable range for generating tabu tenures,
Γ	recency-based memory,
δ	largest increase in objective value from one iteration to the next,
ρ	scaling factor for frequency-based memory penalty term,
η_{it}	frequency of transferring or exchanging base unit i,
ν_{jt}	frequency of district j's being subject to a transfer or exchange,
n_i^b	number of times base unit i was moved or transferred,
n_j^d	number of times district j was subject to a transfer or exchange,
Υ_t	penalty term associated with the frequency-based memory in iteration t,
Λ^b, Λ^d	frequency-based memories for base units and districts,
$\mathcal{D}$	collection of high-quality districts (PDI),
s	number of solutions associated with $\mathcal{D}$,
g_i	"quality" of district $D_i \in \mathcal{D}$,
M	number of PDI iterations,
L	a collection of base units or districts.

5.2 Solution Representation

A solution to the single-member political districting problem is characterized as a collection of electoral districts that are mutually exclusive and that also collectively "cover" the territory being districted. In this study, such a collection will be referred to as a *district plan*. Typically, the territory is divided into a certain number of regions (districts), m. Districts are comprised of smaller geographical units, or *base units* (BU) for short, that are the building blocks of the districts. Depending on the level of districting (i.e., whether it is at the federal, provincial or municipal level), these building blocks may be chosen from counties, census tracts, enumeration areas, or even a mixture of some or all of these.

Given the concept of using BUs as building blocks, a solution to the districting problem is a partition of the set of all BUs into m subsets. Mathematically, let x_{ij} be a binary variable that indicates whether or not BU i is assigned to district j. Furthermore, let n be the number of all BUs in the territory. A solution X to the problem is then defined as

$$X = \{x_{ij} : i \in I, j \in J\} \tag{5.1}$$

where $I = \{1, \ldots, n\}$ is the index set for BUs and $J = \{1, \ldots, m\}$ is the index set for districts. In other words, a complete solution to the districting problem specifies which BUs are contained in which districts. Throughout the remainder of the text, the terms "solution" and "district plan" will be used interchangeably, and both will be represented by X. Note that, each BU can be assigned to one and only one district, i.e., $\sum_{j \in J} x_{ij} = 1$.

As far as the feasibility of X is concerned, various criteria might be relevant depending on the legislation in effect in the territory. One major criterion, as discussed in Chapter 2, is to have districts balanced with respect to population. Contiguity is another criterion in some jurisdictions. In this study, these two criteria will jointly define feasibility. In other words, a feasible district plan will have to satisfy the two conditions that are described below. If any one of these two conditions is violated, the district plan is defined to be infeasible.

- Population balance: The population of all districts in a district plan must be within a pre-determined range $[P_{min}, P_{max}]$. Here, $P_{min} = (1 - \beta)\bar{P}$ and $P_{max} = (1 + \beta)\bar{P}$ where $\bar{P} = P/m$ is the district quotient and β is the allowable $\pm$ % deviation from $\bar{P}$. The notation $P = \sum_{i \in I} p_i$ represents the total population residing in the territory, where p_i is the population of BU i.

- Contiguity: A district plan must be contiguous, i.e., each BU within a district must be reachable from every other BU in the district by visiting a sequence of adjacent BUs in that district.

From the modeling perspective, these two conditions are nothing but two constraints that a feasible solution has to satisfy. However, the TS algorithm will treat the first one as a soft constraint. That is, the algorithm will allow temporary violations of this constraint, as described in Section 5.3.1. On the other hand, the second one will be treated as a hard constraint, which means that the TS algorithm will always maintain the contiguity of district plans considered. This assumption is made because of the fact that once contiguity is relaxed, it is computationally expensive to keep track of the number of pieces each district is comprised of and how these pieces are affected as a result of TS moves.

Even if contiguity is not a legislated requirement, it may still be desirable for the reasons discussed in Chapter 2. To produce non-contiguous solutions with the algorithm, one could drop the contiguity requirement. However, the resulting districts will probably have many non-contiguous pieces, which would not be desirable by the districting authority.

5.3 Objective Function

The political districting problem is a multi-criteria problem. To address this nature of the problem, the weight method is adopted. That is, the objective function is made up of multiple terms, each of which is an appropriate measure of a relevant criterion, weighted with an appropriate non-negative weight. The general form of the objective function looks like the following:

$$f(X) \quad = \quad w'h(X) + w_1 f_1(X) + w_2 f_2(X) + ... \tag{5.2}$$

In Chapter 2, a detailed list of relevant districting criteria was provided. The subset of these criteria that has been incorporated in the TS algorithm is given in the following sections. Each criterion is further detailed as to how it is measured and included in the objective function. With all the terms of the objective function, smaller values are preferred to larger ones, i.e., the aggregate objective function $f(X)$ is to be minimized.

The first term in (5.2) is clearly different from the rest. This is the term associated with the first condition of feasibility, i.e., the requirement that the districts be balanced with respect to population. Consequently, this term measures the magnitude of violation of the population constraint. It is listed and will be discussed in the first order, because it is perhaps the most emphasized criterion in political districting. Description of the rest of the terms will follow the discussion of this criterion.

5.3.1 Population Balance

As mentioned above, population balance is one of the two conditions that define feasibility. It is treated as a soft constraint by the algorithm, which means that the algorithm is allowed to temporarily visit solutions that possess districts with population outside the range $[P_{min}, P_{max}]$. This is done with the expectation that feasible solutions of higher quality can be found in later iterations of the algorithm. Another objective is to make it easier for the algorithm to direct its search to unexplored parts of the solution space.

Since infeasibilities of this kind are undesirable, they should be discouraged. With this purpose, a penalty term to be added to the objective function is introduced. The penalty term is computed as the total amount of violation (if any) relative to the district quotient $\bar{P}$, multiplied by an adjustable parameter α. Mathematically, it is given by:

$$h(X) = \alpha \cdot \frac{\sum_{j \in J} \max\{P_j - P_{max}, P_{min} - P_j, 0\}}{\bar{P}} \tag{5.3}$$

where $P_j = \sum_{i \in D_j} p_i$ is the population of district j. This last quantity is computed as the total population of BUs in district j with $D_j = \{i \in I : x_{ij} = 1\}$ being the BU index set for district j.

The numerator of the ratio in (5.3) is the total (absolute) violation above and below the limits of the allowable population range. The *relative* amount of violation is computed simply by taking its ratio to $\bar{P}$. If $P_j \in [P_{min}, P_{max}], \forall j \in J$, there is no violation and therefore the penalty term $h(X)$ is equal to zero.

The coefficient α in (5.3) is used by the algorithm to adjust the frequency of switches between infeasible and feasible solutions. Initially, α is set equal to 1, and it is continuously reviewed for adjustment during the course of the algorithm. This parameter is either multiplied or divided by 2, depending on the history of solutions visited in the past iterations. Specifically, the following rules are used to determine whether α should be adjusted or not:

- If at least $\bar{\mu}$ of the last μ solutions visited were feasible, set $\alpha \leftarrow \alpha/2$,
- Else if at least $\bar{\mu}$ of the last μ solutions visited were infeasible, set $\alpha \leftarrow 2\alpha$,
- Else, leave α unchanged.

The above checks are performed every μ iterations. The other parameter $\bar{\mu}$ specifies exactly how many solutions will have to be feasible or infeasible for α to be adjusted. In essence, what α does is to increase the penalty of violations, if most or all of the solutions visited recently are infeasible (assuming $\bar{\mu}$ is large enough). Similarly, it decreases the penalty if most or all of the recent solutions are feasible. Note that $\bar{\mu}$ must be greater than $\mu/2$ so that the first two conditions are not satisfied simultaneously. This scheme of adjusting α allows the algorithm to switch between feasible and infeasible solutions on a continuous basis. If the algorithm visits too many infeasible solutions in succession, feasible solutions are encouraged by doubling the penalty term.

Table 5.1 shows a sample calculation of this penalty term with $m = 10$, $P = 800\,000$, $\bar{P} = 80\,000$, $P_{max} = 100\,000$ and $P_{min} = 60\,000$ (i.e., $\beta = 0.25$), and finally $\alpha = 1$.

District	P_j	Violation	Penalty
1	85 000	-	
2	73 000	-	
3	58 000	2 000	
4	70 000	-	
5	87 000	-	
6	105 000	5 000	
7	87 000	-	
8	104 000	4 000	
9	56 000	4 000	
10	75 000	-	
TOTAL	**800 000**	**15 000**	$= 1 \cdot 15\,000/80\,000 = 0.1875$

Table 5.1: Calculation of the penalty term for violation of population balance

To summarize, the penalty term $h(X)$ is part of the objective function $f(X)$ and it is non-zero whenever there is a violation of the allowable range $[P_{min}, P_{max}]$. The term $h(X)$ is further multiplied by the weight w', to adjust its balance with respect to the rest of the terms in f. (Note that an alternative way of doing this would be to change the starting value of α.) As a result, the objective function has the form in (5.2).

5.3.2 Compactness

The second criterion that is part of the objective function is compactness. It is associated with the term $f_1(X)$ and the weight w_1. The following two measures are used by the TS algorithm for the quantification of a solution's compactness value:

- the total length of 'interior' district perimeters.
- comparison of the district perimeter with the perimeter of a circle that has the same area as the district.

Recall from Chapter 2 that compactness measures can be in conflict, as shown by Niemi et *al.* [46], when it comes to measuring the compactness of an entire district plan or individual districts. For this reason, two measures have been selected for implementation instead of one. They will both be part of the computational study in Chapter 7, where a consistency check will also be performed. The remainder of this subsection describes in detail how the compactness of a district plan will be computed using these two measures.

Measure 1: Sum of interior district perimeters

This measure of compactness is a function of the total length of district perimeters. The objective is to minimize this function so that the algorithm would be forced to make district boundaries follow straight lines. Specifically, it is computed by taking the total length of *interior* district perimeters and scaling it by R, the perimeter of the entire territory:

$$f_1(X) = C_1(X) = \frac{\frac{1}{2}(\sum_j R_j - R)}{R} \tag{5.4}$$

Here, R_j denotes the perimeter of district j and is given by

$$R_j = \sum_{i \in D_j} r_i - \sum_{i,k \in D_j, i \neq k} \sum t_{ik} \tag{5.5}$$

where r_i is the perimeter of BU i and t_{ik} is the length of the common boundary between BUs i and k. Here, R_j is computed by first summing up the perimeters of all BUs in district j, then subtracting all BU perimeters that are double counted in $\sum_{i \in D_j} r_i$. The numerator of the ratio in (5.4) is the total length of interior perimeters, which is one half the difference between the sum of all district perimeters and R, ($\frac{1}{2}$ because of double counting). In computing $C_1(X)$, the compactness value of X, R is used as a constant scaling factor.

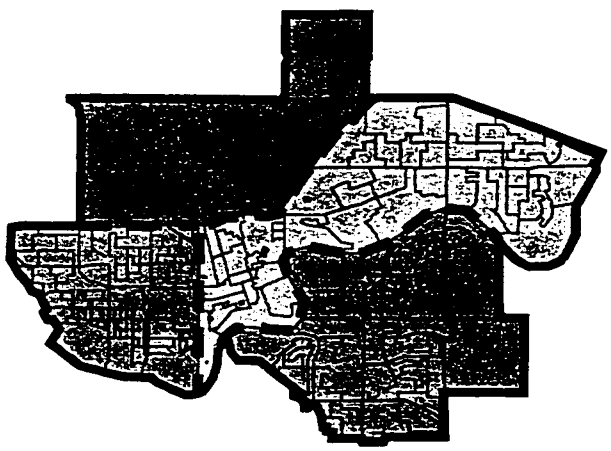

Figure 5.1: Compactness Measure 1: interior borders scaled by exterior border

As an illustrative example, consider the district plan in Figure 5.1. This district plan is composed of four districts that are separated from each other with the dashed lines shown. These dashed lines correspond to the district perimeters *interior* to the entire plan. The thick solid lines correspond to the perimeter of the territory and is denoted by R. Consequently, the compactness of this plan is computed as the ratio of the total length of dashed lines to R.

Measure 2: District perimeter vs. perimeter of circle with the same area

The second measure used to assess the compactness of a district plan is one that relates the shape of each district to a circle. Recall that the measure Per_4 reported in the article by Niemi et *al.* [46] (which was first proposed by Horton [30]) is based on this concept. In this dissertation, a modified version of Per_4 is used.

The original measure Per_4 computes the ratio of a district's perimeter to the perimeter of the circle that has the same area. Since a measure that takes values between 0 and 1 is more desirable for scaling purposes, the following modification of Per_4 is used to compute compactness of district j:

$$C_{2j}(X) = 1 - \frac{2\pi\sqrt{A_j/\pi}}{R_j} = 1 - \frac{2\sqrt{\pi A_j}}{R_j} \tag{5.6}$$

where $A_j = \sum_{i\in D_j} a_i$ is the area of district j, a_i being the area of basic unit i. Here, the numerator of the ratio is the perimeter of the circle that has area A_j, and therefore has radius $\sqrt{A_j/\pi}$.

Note that (5.6) gives the compactness of a single district. The question is how to combine individual district compactness values into an aggregate value. Two immediate approaches would be to look at the simple average of the individual district compactness values, or the maximum of them. With the simple average approach, one district may be highly non-compact, but this might be negated by a number of highly compact districts. With the minimax approach, one is guaranteed not to have a highly non-compact district, but at the same time one would lose from the overall average. To avoid highly non-compact districts, one could also choose, as a third alternative, to minimize the average of the *squares* of (1 + district compactness values), which would in turn penalize non-compact districts to a greater extent (compactness values are increased by 1 so that squaring them magnifies the relative distance from the maximum value). Among these three alternatives, the simple average method is chosen for its simplicity. That is, the compactness of a district plan is computed by

$$f_1(X) = C_2(X) = \frac{1}{m}\sum_{j\in J} C_{2j}(X) = \frac{1}{m}\sum_{j\in J}(1 - \frac{2\pi\sqrt{A_j/\pi}}{R_j}) \tag{5.7}$$

5.3.3 Changes with respect to Existing District Plan

One of the political criteria that have been included in the TS algorithm is to minimize changes that are made to the existing district plan. This is accomplished through the use of a measure developed in this section. This measure is intended to assess the *similarity* between any two district plans being compared. Furthermore, it is also in the form of an index that takes values between 0 and 1, just like the second compactness measure. The value 0 is measured when two plans are identical (i.e., perfectly similar), and higher values indicate increased 'distance' from perfect similarity. The objective function term associated with this measure is $w_2 f_2(X)$.

Consider Figures 5.2a and 5.2b for an illustrative description of this index. Figure 5.2a shows two district plans of a given territory: one with the solid border line, and the other with the dashed border line. Perfect similarity would have been achieved if the solid lines coincided with the dashed lines.

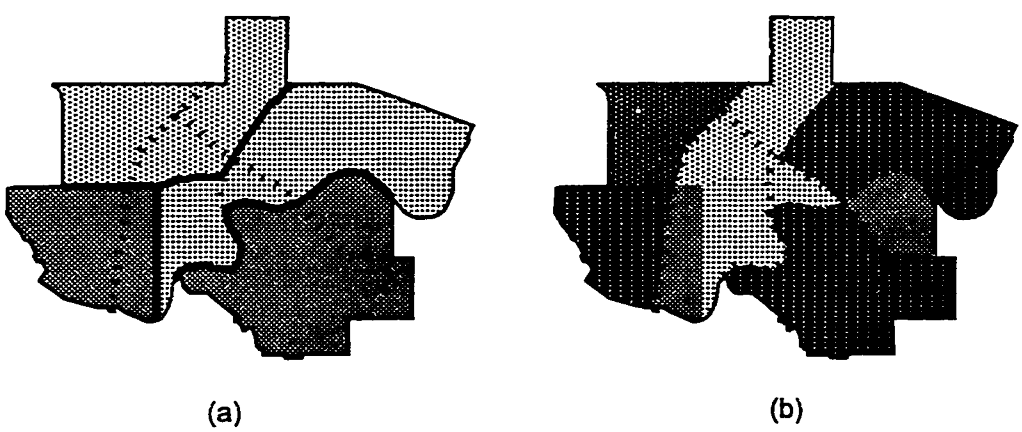

Figure 5.2: Measuring changes with respect to a base plan

Another way to recognize perfect similarity is to look at the content of each district. If all the BUs in each district of the first plan are also together in a district of the second plan, then the two plans would be identical.

Figure 5.2b demonstrates how the similarity between the two district plans is evaluated.

The plan with the solid lines is taken as the base plan. The plan that is compared with the base plan (i.e., the plan with dashed border lines) is then overlaid and intersected with the base plan. The next step is to compute, for each district in the base plan, the area of the largest piece within that district formed as a result of the overlay. These largest pieces are indicated as the shaded areas in Figure 5.2b. The similarity index is then computed as the ratio of the total shaded area to the entire territory area. To make the index minimizable, the ratio is also subtracted from 1. (In this form, the index is rather a *dis*-similarity index.) Mathematically,

$$f_2(X) = 1 - \frac{\bar{A}_1 + \ldots + \bar{A}_m}{A} \tag{5.8}$$

where $\bar{A}_j$ corresponds to the area of the largest piece in district j that is formed as a result of the overlay. For this particular objective function term, the base plan is the district plan currently in effect. One advantage of this measure is that it allows the number of districts in the two plans compared to be different.

5.3.4 Integrity of Communities

An important political criterion that is included in the objective function is the integrity of communities. This criterion is incorporated essentially the same way as the previous criterion (changes with respect to existing district plan) is. This time, the base plan is a "community plan", where the entire territory is divided into regions that represent communities. The measure again forces the BUs that are part of a community to be assigned to the same district to a maximum extent. Consequently, this reduces the number of communities that are split between two or more electoral districts.

The above approach does not require that the number of communities and the number of districts in a district plan be equal. In most cases, one would not be able to identify as many communities as the number of electoral districts anyway. In any case, the technique works equally well in both of the two cases where the number of electoral districts exceeds or is less than the number of communities.

The computation of the numerical value for this measure is the same as that described for Figure 5.2b. The only difference is that the number of districts in the base plan and that in the plan that is compared to the base plan might (and will most probably) be different. Therefore, in those cases, it will not be possible to achieve a perfect score. If the number of electoral districts exceeds the number of communities, it is inevitable that some communities will be split among districts. But the larger the pieces each district has from a community, the better the score of the district plan will be. In summary, if this criterion is minimized, one should expect that the districts in the resulting district plan should have big blocks of communities together and overall the communities get split among a few number of districts.

5.3.5 Proportionality

Proportionality suggests that the portion of the seats a party wins in an election should be (roughly) equal to the party's share of the popular vote. While it is difficult, in advance of an election, to predict the share of votes each party will get, historical data may have certain indications in certain regions. That is, some parts of the territory might have turned out to be conservative over the years, and some might have been liberal, etc. This historical data might be used to compute a *proportionality score* (PS) of a district plan. Furthermore, this proportionality score can be optimized on its own to produce a district plan that achieves the best possible level of proportionality. Clearly, there is no guarantee that the historical voting patterns will recur in the next election. Moreover, the surveys conducted for estimating the voting patterns will not be completely reliable, if not obviously in favor of one party. In addition to these, it may not be possible to achieve an acceptable level of proportionality due to the population distribution in the territory (as described in Section 2.3). For these reasons, it may not always be appropriate to incorporate this criterion into the algorithm for optimization purposes. However, assuming that none of these unfavorable conditions are present, one may be interested in computing (and optimizing) such a score. The objective is to see if large disproportionalities are present in a particular district plan, and avoid them as much as possible.

For the computation of the PS used in this study (for a system with Q parties), let S_k denote the fraction of seats won by party k, $k \in \{1, \ldots, Q\}$, and V_k denote the percent votes received by this party. Clearly, a party wins a seat in an associated district when $v_{kj}(X)$, percent votes received by party k in district j of plan X, is greater than that of the other parties. Let $J_k^W(X)$ ($J_k^L(X)$) be the index set for the districts won (lost) by party k, given a particular district plan X, computed as follows:

$$J_k^W(X) = \{j \in J : v_{kj}(X) > v_{lj}(X), \forall l \neq k\} \tag{5.9}$$

$$J_k^L(X) = J \setminus J_k^W(X) \tag{5.10}$$

In a district $j \in J_k^W(X)$, let v_{sj} denote the vote share of the party (party s) that has received the second largest share of votes. In a district $j \in J_k^L(X)$, let v_{tj} denote the vote share of the party (party t) that has received the largest share of votes. Given these definitions, the PS of a district plan is computed under the following three cases:

Case 1 ($S_k < V_k$): In this case, the party has won fewer seats than it should, according to the proportionality principle. Consequently, PS is designed to reflect the distance between S_k and V_k, so that when it is minimized, party k is forced towards achieving $S_k = V_k$, the perfect proportionality for party k. The exact formula is given below:

$$G_k(X) = \frac{1}{|J_k^L(X)|} \sum_{j \in J_k^L(X)} (v_{tj}(X) - v_{kj}(X)) \tag{5.11}$$

Observe that the proportionality score $G_k(X)$ is computed using the vote shares of party k only in districts that were lost.

Case 2 ($V_k < S_k$): In this case, the party has won more seats that it should. The PS is computed similarly, but this time based on the districts that this party has won:

$$G_k(X) = \frac{1}{|J_k^W(X)|} \sum_{j \in J_k^W(X)} (v_{kj}(X) - v_{sj}(X)) \tag{5.12}$$

Case 3 ($V_k = S_k$): This is the (rare) case where party k has won the exact number of districts it should, therefore proportionality is already achieved for party k. Consequently, $G_k(X) = 0$ in this case. Note that this is usually a very unlikely situation, since the number of seats a party can get is always an integer value, which in turn produces a very restrictive set of values for S_k. Theoretically, it may be impossible to achieve $V_k = S_k$ for given past election data.

So far, the PS for only one party, party k, has been computed. The next step is to combine all individual party scores into a single score that would reflect the PS of the entire district plan. To address this question, the simple average approach is adopted here as well:

$$f_4(X) = \frac{1}{Q} \sum_{k=1}^{Q} G_k(X) \tag{5.13}$$

5.4 Features of the Algorithm

Having described the composition of the objective function, the next step is to give an overview of the algorithm features. This section includes a detailed description of such features as

- construction of the starting solution
- tabu search moves

- recency- and frequency-based memory, and
- stopping rules

5.4.1 Construction of the Starting Solution

As mentioned earlier, the algorithm first constructs a starting solution, and performs iterations of TS moves in its search for better district plans. The module used for constructing the initial solution is similar to the heuristic method by Vickrey [59]. Below is the outline of this module.

<u>INITSOLN</u>

1. Set $j = 0$.
2. Repeat until each BU is assigned to a district
   - 2.1. Set $j \leftarrow (j + 1)$.
   - 2.2. Pick a BU randomly as a seed as follow:
     - 2.2.1. Construct the list of BUs that are not yet assigned to a district, and that are also along the periphery of the region being districted. If this list is empty, go to Step 2.2.3.
     - 2.2.2. Select a BU randomly from the list constructed in Step 2.2.1. Go to Step 2.3.
     - 2.2.3. Construct the list of all unassigned BUs.
     - 2.2.4. Select a BU randomly from the list constructed in Step 2.2.3.
   - 2.3. Assign seed found in Step 2.2 to district j.
   - 2.4. Create district j clustered around the seed as follows:
     - 2.4.1. Identify a "pivot" BU among the BUs that are already assigned to district j and are yet unused as pivots. Mark this pivot as "used". If no such pivot exists, construction of district j is complete; go to Step 2.1.
     - 2.4.2. Assign neighbors of the pivot BU to district j until the district population P_j reaches or exceeds $\bar{P}$ for the first time.
     - 2.4.3. If all neighbors of the pivot are assigned and $P_j < \bar{P}$, repeat steps 2.4.1 and 2.4.2. Else, construction of district j is complete; go to Step 2.1.
3. If $j = m$, STOP. Else if $j > m$, repeat steps 3.1-3.4 until j is reduced to m. If $j < m$ split the most-populated district into two districts of roughly equal population, and repeat the process until j is increased to m.

3.1. Find the district q with the min. population (break ties arbitrarily).

3.2. Find the district $l \neq q$, the least-populated neighbor of district q.

3.3. Combine districts q and l into one district.

3.4. Set $j \leftarrow (j-1)$.

As outlined above, this module first selects seeds from BUs and then clusters other BUs around these seeds to form one district per seed. The assignment of BUs to a district (seed) is stopped when the total population of the district reaches or exceeds $\bar{P}$ for the first time. The process is repeated until no unassigned base units remain. Enclaves (i.e., a set of BUs or a district completely enclosed by a single BU or a district) and residual areas might form along the way, and therefore it may not be possible to create exactly m districts in Step 2. If so, the resulting number of districts is usually larger than m. This case is treated in Step 3 where the number of districts is reduced to m by combining the minimum population district with its least-populated neighbor. Contiguity is maintained in all steps of the above algorithm.

The above module is one way of creating an initial district plan. The district plans so-produced are always contiguous, but they do not necessarily have low compactness values. Also, some districts are likely to have populations outside the range $[P_{min}, P_{max}]$, and no political criteria are considered. In fact, the plans produced by INITSOLN are observed to be poor in quality, but the objective of the module is to produce a starting solution to the main TS algorithm as quickly as possible. It is left to the TS algorithm to fix any deficiencies of the starting solution.

5.4.2 Tabu Search Moves and Neighborhoods

Once the starting solution is constructed, the next step is to perform an iterative search over the solution space, based on the tabu search principles described in Chapter 4. As far as the TS moves are concerned, the algorithm moves from one solution to the next by implementing the following two types of moves:

- Type I move: transfer of a single BU i from a district j to a neighboring district l,
- Type II move: exchange of two BUs i and k between districts j and l.

For practical reasons, these moves will be denoted by (*BU_index, from, to*)=(i, j, l) and (*BU_index_1, BU_index_2, from_1, to_1*)=(i, k, j, l) respectively. Furthermore, $M_t^1(X) =$

$\{(i,j,l)\}$ will be used to denote the set of all eligible Type I moves in iteration t and X_{ijl} to denote the solution that results from performing move (i,j,l) on the current solution X. $M_t^2(X) = \{(i,k,j,l)\}$ and X_{ikjl} are defined similarly. Type I and Type II moves are used to construct the neighborhood of the current solution in each iteration. Specifically, $N_1(X)$ is the one that contains solutions resulting from all possible Type I moves performed on X, and $N_2(X)$ is the one that results similarly from implementing Type II moves. That is,

$$N_1(X) = \{X_{ijl} : (i,j,l) \in M_t^1(X)\} \tag{5.14}$$

$$N_2(X) = \{X_{ikjl} : (i,k,j,l) \in M_t^2(X)\} \tag{5.15}$$

Figures 5.3 and 5.4 below further illustrate these two types of moves. Figure 5.3 displays the transfer of a single BU from district A to an adjacent district B, whereas Figure 5.4 displays the exchange of two BUs between two adjacent districts A and B. In both figures, the solid lines indicate district boundaries and BUs that are being transferred or exchanged are shown as shaded polygons.

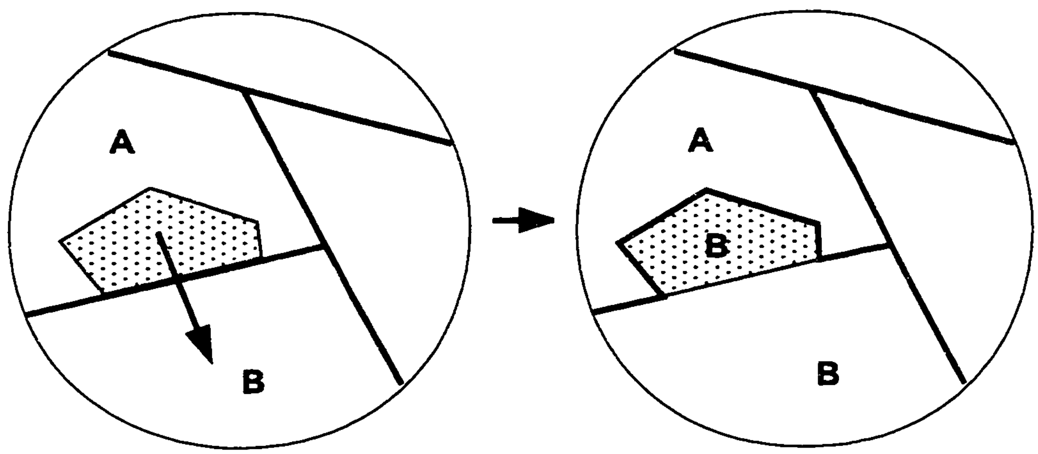

Figure 5.3: Type I move

Observe that if a BU i being transferred or exchanged <u>enclaves</u> other BUs, implementing the move causes discontiguity, as shown in Figure 5.5. The algorithm is able to handle this situation by transferring all the enclaved BUs together with BU i. (If this is not allowed, BU i is bound to remain assigned to the same district for the entire duration of a TS run, which is likely to affect the reachability of some solutions.) This does not constitute a problem as far as the composition of districts is concerned, since BU i and all the BUs it enclaves have to be in the same district in any case. The only exception is when the BUs that are enclaved have a large enough population to create a district different from the one BU i is assigned to. In this case, only BU i is transferred or exchanged.

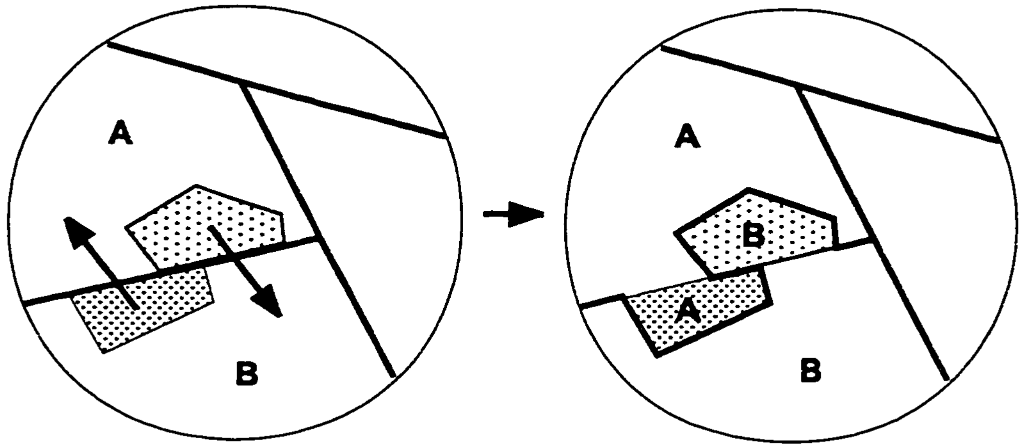

Figure 5.4: Type II move

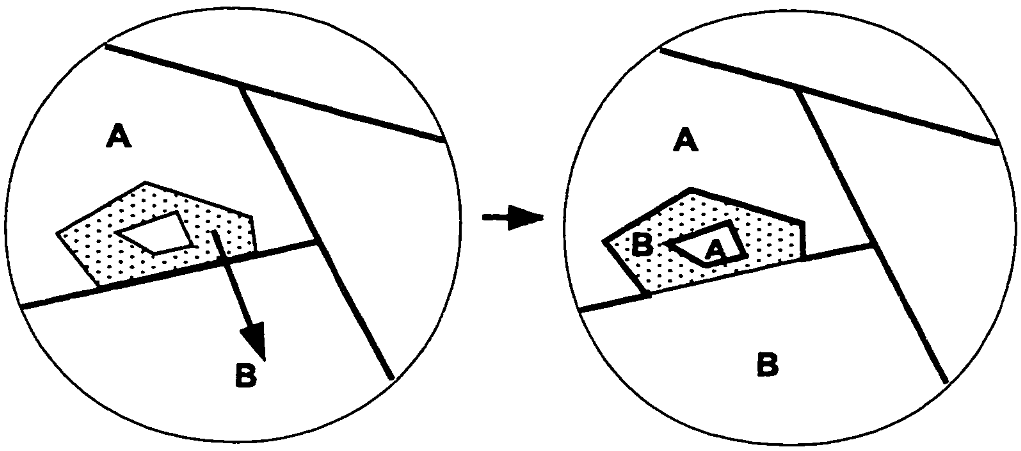

Figure 5.5: Discontiguity resulting from transfer of base units with enclaves

The algorithm first uses the neighborhood $N(X) = N_1(X)$ until it is not possible to improve the best known solution for T subsequent iterations. It then switches to $N'(X) = N_1(X) \cup N_2(X)$ for the rest of the run. We avoid using $N'(X)$ as much as possible because the extended set $N'(X)$ is computationally much more expensive to process (recall the numerical example in Section 4.2).

Maintaining contiguity is an important task while implementing the moves. Before explaining how this is actually done, let the term *border base unit* (BBU) be associated with a BU that is adjacent to at least one BU from a different district. The shaded polygons in Figure 5.6 are examples of such BUs. A necessary condition for contiguity is that a BU being transferred or exchanged must be a BBU. That is, BUs from the "interior" of a district cannot be transferred or exchanged without violating contiguity. This, however, is not a sufficient condition, as shown in Figures 5.7. Even if the enclaves are pre-processed and they are eliminated all together, cases like in Figure 5.7c may also arise. As a result, the necessary condition is used only to narrow down the list of BUs eligible for a transfer or exchange. Each BU in this list is still subject to a contiguity check. If a BU or a pair of BUs fail the contiguity check, the resulting solution is excluded from the corresponding neighborhood. This check is performed, for a Type I move (i, j, l), as detailed in the module ISCONTIGUOUS given on the next page.

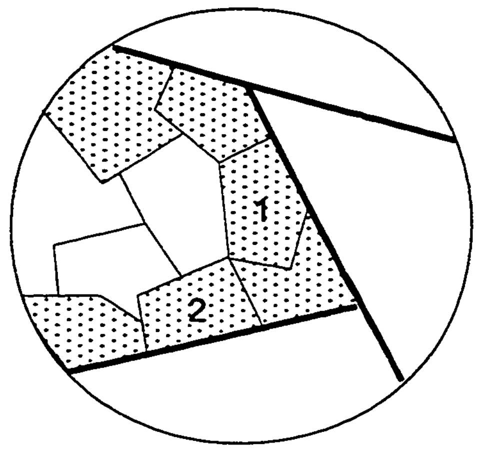

Figure 5.6: Border base units

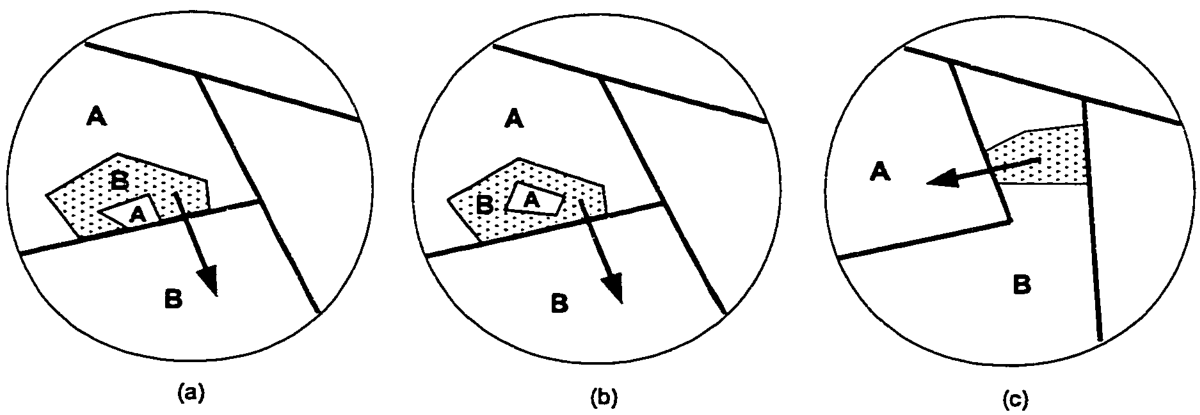

Figure 5.7: Non-contiguous solutions that may arise as a result of the two types of moves

One additional definition has to be made before the ISCONTIGUOUS module can be outlined. Two BUs i and k are defined to be *point-adjacent* if their intersection is a finite set of points (as opposed to a set of line segments). The BUs marked '1' and '2' in Figure 5.6 are point-adjacent in this regard. In this case, t_{ik}, the length of the common boundary between BUs i and k, is equal to zero, but the two BUs still "touch" each other. This definition is necessary as the ISCONTIGUOUS module operates mainly on a subset L of the source district's BUs: the set of BUs in the source district j that are adjacent *or* point-adjacent to BU i, the BU that is being transferred or exchanged. The idea is to check whether L is contiguous after BU i is removed from district j. Since BU i is also a BBU, the contiguity check needs to be done for the source district j only. This is because the target district l can never be made non-contiguous by gaining the BBU i from the source district.

ISCONTIGUOUS(i, j, l)

1. Let $L = \{k : k \in D_j, \text{BUs } k \text{ and } i \text{ are adjacent or point-adjacent}\}$.
2. Partition L into contiguous subsets $L_1, \ldots, L_q$.
3. If $q = 1$, return TRUE, i.e., BU i can be transferred from district j to district l without destroying contiguity of district j.
4. Else, check to see if L_j's can be combined into one contiguous set, using adjacencies between BUs in L and $D_j \setminus (L \cup \{i\})$. If they can, return TRUE.
5. Else, return FALSE, i.e., transferring BU i from district j to district l destroys contiguity of district j.

For a Type II move, it is sufficient to apply the module twice, once for the source district and once for the target district, since both districts gain and lose one BU, unless there are enclaves. The critical point is to execute ISCONTIGUOUS as if the exchange has already taken place. For a Type II move (i, k, j, l), this means that ISCONTIGUOUS is to be run for district j assuming that this district has already gained BU k, i.e., $D_j \leftarrow D_j \cup \{k\}$. Similarly for district l, ISCONTIGUOUS is to be run as if district l is composed of BUs in $D_l \cup \{i\}$. With this implementation, the move (i, k, j, l) (solution X_{ikjl}) will not be allowed in the set of eligible moves $M_t^2(X)$ (neighborhood $N_2(X)$), if one (or both) of the two contiguity checks fails. In the case of a transfer/exchange that involves a BU with enclaves, all the BUs that are enclaved by BU i are excluded from set L. Also in Step 4, $L \cup \{i\}$ is replaced with $L \cup \{i\} \cup E$, where E is the set of BUs enclaved by BU i.

Once the neighborhood of a solution is constructed, the next step is to pick a solution $X' \in N(X)$, which the algorithm will visit in the next iteration. As explained in Chapter 4, the TS algorithm selects the solution with the best objective function value, even though this solution might be a non-improving one. Of course, the objective values are modified based on frequency-based memory. Tabu status of moves and applicable aspiration criteria are also relevant in identifying X'. Generally speaking, the algorithm will choose the best non-tabu solution. A tabu solution will be chosen only if it has the lowest objective function value that is also lower than that of the best known solution.

Note that both of the neighborhoods $N_1(X)$ and $N_2(X)$ may contain infeasible solutions, i.e., solutions that violate the population balance constraint. As a result, one may choose an infeasible solution from the neighborhood, if that solution is of sufficiently high quality. In particular, the following set of moves are eligible for selection:

- all non-tabu moves (whether or not they result in feasible solutions).
- all tabu moves that result in infeasible solutions, but that also improve the best known objective function value (feasible or infeasible).
- all tabu moves that result in feasible solutions, but that also improve the best known feasible solution.

The memory implementations and the aspiration criteria that further complement these rules are described in the next two sections.

5.4.3 Basic Tabu Search Memory

Two types of basic memory have been implemented in the algorithm. The first one, the *recency-based* type memory, is for storing and handling tabu status of moves. The second one is the *frequency-based* memory, which provides a diversification over the solution space. These are detailed below.

Recency-based Memory

The main function of this memory is to store the tabu status of moves, i.e., whether or not a move is currently tabu and if tabu, when its tabu status will expire. In the TS algorithm, a move is declared tabu as soon as it is executed, and it remains tabu for the duration of its tabu tenure, θ. The question is how to declare a move tabu and what the value of θ should be.

Recall from Chapter 4 that a move is declared tabu based on one or more of its attributes. In the proposed algorithm, this is done based on the index of the BU being transferred and the index of the district from which this BU is transferred. For a Type I move, the corresponding attributes are *BU_index* and *from*. The move is immediately assigned a tabu tenure, θ. This means that once a BU leaves a district, it will not be able to return to the same district for θ iterations.

All the information necessary for handling tabu status and tabu tenures is stored in a two-dimensional array of size $n \times m$. Let this array be denoted by $\Gamma(i,j)$ where the row index i corresponds to BUs, and the column index j corresponds to districts. In each cell of Γ, the iteration at the end of which the tabu status will expire is stored. If $\Gamma(i,j)$ is less than the current iteration count, then the corresponding move is not tabu.

The update of a particular cell $\Gamma(i,j)$ is performed as follows: when BU i is transferred from district j to another district, $\Gamma(i,j)$ is assigned the value of (current iteration number + tabu tenure). Whenever BU i is considered (in later iterations) for returning to district j, the iteration count at that point is compared with the value stored in cell $\Gamma(i,j)$. If the iteration count is *less than or equal to* $\Gamma(i,j)$, the move is still tabu and therefore cannot be implemented (unless one or more aspiration criteria make it admissible).

For a Type II move (*BU_index_1*, *BU_index_2*, *from_1*, *to_1*) $= (i,k,j,l)$, *two* cells are updated when the move is executed. Cells $\Gamma(i,j)$ and $\Gamma(k,l)$ are updated similarly by recording the iteration the tabu status will expire. A Type II move remains tabu until *one* (but not both) of the transfers that make up the move becomes admissible.

	1	2	3	4
1			5	
2	7			
3				
4				6
5		9		
6	8		7	

Table 5.2: A sample array for storing tabu status

Table 5.2 is a small-scale example of this array, with $n = 6$ and $m = 4$. It is merely a snapshot of the tabu status at a particular iteration. Six moves have been assigned tabu status, but which of these six are tabu depends on the current iteration count. If we are in iteration 7, the Type I moves $(2,\cdot,1)$, $(5,\cdot,2)$, $(6,\cdot,1)$ and $(6,\cdot,3)$ are still tabu. That is, we cannot transfer BU 2 back to district 1, BU 5 back to district 2, and so on. The move $(1,\cdot,3)$ is *not* tabu, since its tabu status has expired at the end of iteration 5. This array needs *no* update other than storing values in the corresponding cells whenever BUs are transferred or exchanged.

For a Type II move $(1,5,2,3)$, which exchanges BUs 1 and 5 between districts 2 and 3, the tabu status would be determined by checking the cells $\Gamma(1,3)$ and $\Gamma(5,2)$. Since the tabu status of the first one has already expired, the Type II move $(1,5,2,3)$ is no longer tabu.

A final note on the selection of tabu tenure. In this study, a random way of assigning tabu tenures has been adopted, because of the reasons discussed in Chapter 4. That is, whenever a move is executed, its tabu tenure is generated randomly from the range $[\theta_{min}, \theta_{max}]$. This increases the number of parameters by one, compared to the deterministic case, but is expected to produce higher quality solutions.

Frequency-based Memory

This particular type of memory is incorporated into the TS algorithm to facilitate diversification over the solution space and also take the algorithm out of local optima that may occasionally arise. The method of using multiple random starts is not preferred since it increases the computational effort and also does not fit in the TS framework. While recency-based memory can also prevent, by coordinating the acceptance of non-improving solutions, the algorithm from getting stuck in local optima, frequency-based memory is used to further help achieve the above goals.

With frequency-based memory, more frequent moves are penalized by adding a penalty term to the objective function. The purpose of this is to encourage moves/solutions that are performed/visited less often. The penalty term is the product of the following four variables and parameters:

- past frequency of a move,
- problem size, as measured by $\sqrt{m}$,
- δ, the largest change in objective function from one iteration to the next,
- ρ, a constant scaling factor.

The first factor, frequency of implementing a move, is the driving force among the four. Typically, the more frequently a move has been implemented in previous iterations, the higher the value of this factor will be. It has two components that keep track of two different attributes of a move. The first component, denoted by η_{it}, stores the number of times BU i was transferred or exchanged in the previous $(t-1)$ iterations. Mathematically,

$$\eta_{it} = \frac{n_i^b}{t-1} \tag{5.16}$$

where n_i^b is the number of times BU i was part of a transfer or exchange in all previous $(t-1)$ iterations of the algorithm. For a Type I move, n_i^b is incremented by 1 at the end of the iteration the move took place. For a Type II move that involves BUs i and k, both n_i^b and n_k^b are incremented by 1.

The second component of the frequency factor, ν_{jt}, is associated with the *districts* that are subject to transfers or exchanges. In this case, ν_{jt} is the frequency of using district j as a *from* or *to* district in a Type I or Type II move in all of the previous $(t-1)$ iterations. It is given by

$$\nu_{jt} = \frac{n_j^d}{t-1} \tag{5.17}$$

where n_j^d is the number of times district j was involved in a Type I or Type II move in all previous $(t-1)$ iterations. For a Type I move (i, j, l), n_j^d and n_l^d are incremented by 1, whereas for a Type II move (i, k, j, l), n_j^d and n_l^d are incremented by 2.

Finally, the frequency factor is computed, using the two components η_{it} and ν_{it}, as follows. For a Type I move,

$$\Upsilon_t(i, j, l) = (1 + \eta_{it})(1 + \frac{\nu_{jt} + \nu_{lt}}{2}) - 1 \tag{5.18}$$

For a Type II move,

$$\Upsilon_t(i, k, j, l) = (1 + \frac{\eta_{it} + \eta_{kt}}{2})(1 + \frac{\nu_{jt} + \nu_{lt}}{2}) - 1 \tag{5.19}$$

The variables n_i^b and n_j^d are stored in single dimensional arrays of size n and m, respectively. These two arrays will be denoted by Λ^b and Λ^d.

The second factor in the penalty term formula, $\sqrt{m}$, is related to the problem size. The reason for incorporating the problem size into the penalty term is to magnify the diversification effect as a result of an increase in problem size, to account for the increased size of the solution space. Including n in addition to m is probably unnecessary, since an increase in n from one districting territory to another is usually associated with a proportional increase in m. (This assumes that the district quotient remains the same, which is usually a valid assumption for the cities in the same province/state.) For this reason, m is chosen arbitrarily to represent the problem size. The square root operator limits the magnifying effect as a result of an increase in m. Several studies report more successful results with $\sqrt{m}$ than with m (e.g. Cordeau et *al.* [6]).

The third factor, δ, is the variable that stores the maximum change in objective function from one iteration to the next. The reason for using this factor is to scale the magnitude of the penalty term, so that the entire penalty term is neither too large nor too small relative to the objective function value.

The last factor, ρ, is a constant scaling parameter. The role of this parameter is to provide an additional scaling effect on the penalty term. When the penalty term is too small that it has essentially no effect on perturbing the ranking of moves, ρ should be increased. If the penalty term is too large that it has a disruptive effect on the search process, ρ should be decreased. It provides additional convenience for scaling the penalty term, since its value is set by the user unlike the other three factors contributing to the magnitude of the penalty. To have an effective diversification without disrupting the search process, this parameter is pre-determined and fixed at a constant level for a given problem.

These four factors are multiplied together and added to $f(X)$ to yield the following modified objective function:

$$f'(X) = f(X) + \Upsilon_t \delta \rho \sqrt{m} \tag{5.20}$$

5.4.4 Stopping Rules

Two stopping rules have been incorporated into the Tabu Search algorithm. The first rule, which is the primary one, causes the algorithm to terminate when $T = \lceil 230\sqrt{m} \rceil$ iterations are executed consecutively without improvement to the best known feasible and infeasible solutions. This upper bound is designed so that as the problem gets bigger, more iterations are allowed before the algorithm stops. The constant 230 is selected to allow for a large yet practical number of attempts to get out of local optima.

The second rule is an upper limit on the total number of iterations. When this limit is reached, the algorithm stops no matter where it is in its search process. This limit should be large enough that the algorithm is not interrupted in the middle of a promising search thread. A limit of 30 000 is used in the algorithm.

5.5 The Proposed Algorithm

Based on the definitions of TS moves, memory structures, and the stopping rules, the outline of the algorithm will be provided in this section. This algorithm will be referred to as the *plain* tabu search algorithm, since it starts with a single initial solution and makes one pass of the iterative procedure. A meta-heuristic algorithm that executes multiple passes is described in Chapter 6.

Before the plain Tabu Search algorithm is outlined, recall that a district plan might be feasible or infeasible. Therefore, it is necessary to keep track of the best known feasible solution and the best known solution (feasible or infeasible) separately, as they are encountered during a TS run. The symbols F_1 and F_2 are used for this purpose. F_1 is used to denote the objective function value of the best known feasible solution, whereas F_2 is used to denote the best known objective function value (feasible or infeasible). F_1 and F_2 are initialized to a large number before the algorithm starts, and they are updated as necessary.

TABU_SEARCH

0. Initialization

    0.1. Use module INITSOLN to construct a starting solution X_0. Set $F_2 = f(X_0)$. If X_0 is feasible, set $F_1 = f(X_0)$; else, set $F_1 = \infty$.

    0.2. Set iteration count $t = 0$, pass $= 1$, $\alpha = 1$, $\delta = 0$.

    0.3. Initialize recency-based memory Γ, and frequency based memory Λ^b, Λ^d to 0.

1. Repeat until no improvement is made on F_1 and F_2 for any successive T iterations:

    1.1. Set $t \leftarrow t + 1$.

    1.2. Construct M_t^1 (the set of eligible Type I moves), and find the best move in M_t^1:

    1.2.1. Set $M_t^1 = \phi$.

    1.2.2. For each BBU i of each district j,

    1.2.2.1. Construct set L of all districts neighboring district j to which BU i can be transferred.

    1.2.2.2. For each $l \in L$, if ISCONTIGOUS(i, j, l) returns TRUE, and the move (i, j, l) satisfies one of the following conditions, set $M_t^1 \leftarrow M_t^1 \cup \{(i, j, l)\}$.

    - $t > \Gamma(i, l)$
    - $t \leq \Gamma(i, l)$, X_{ijl} is feasible, and $f(X_{ijl}) < F_1$
    - $t \leq \Gamma(i, l)$, X_{ijl} is infeasible, and $f(X_{ijl}) < F_2$

    1.2.3. For each move $(i, j, l) \in M_t^1$, compute $f'(X_{ijl})$ as follows:

    1.2.3.1. If $f(X_{ijl}) < f(X_{t-1})$, set $f'(X_{ijl}) = f(X_{ijl})$.

    1.2.3.2. Else, set $f'(X_{ijl}) = f(X_{ijl}) + \Upsilon_t \delta \rho \sqrt{m}$.

    1.2.4. Identify the move $(i^*, j^*, l^*) \in M_t^1$ as the best Type I move:

    1.2.4.1. Sort M_t^1 in ascending order of $f'(X_{ijl})$.

    1.2.4.2. Take the first non-tabu move or the first tabu move that improves F_1 or F_2 as (i^*, j^*, l^*).

1.3. If pass = 2, construct M_t^2 (the set of eligible Type II moves), and find the best move in M_t^2:

1.3.1. Set $M_t^2 = \phi$.

1.3.2. For each pair of BBUs $i \in D_j$ and $k \in D_l$, for each pair of neighboring districts j and l,

1.3.2.1. If ISCONTIGUOUS(i, j, l) and ISCONTIGUOUS(k, l, j) both return TRUE, and the move (i, k, j, l) satisfies one of the following conditions, set $M_t^2 \leftarrow M_t^2 \cup \{(i, k, j, l)\}$.

- $t > \Gamma(i, l)$ OR $t > \Gamma(k, j)$
- $t \leq \Gamma(i, l)$ AND $t \leq \Gamma(k, j)$, X_{ikjl} is feasible, and $f(X_{ikjl}) < F_1$
- $t \leq \Gamma(i, l)$ AND $t \leq \Gamma(k, j)$, X_{ikjl} is infeasible, and $f(X_{ikjl}) < F_2$

1.3.3. For each move $(i, k, j, l) \in M_t^2$, compute $f'(X_{ikjl})$ as follows:

1.3.3.1. If $f(X_{ikjl}) < f(X_{t-1})$, set $f'(X_{ikjl}) = f(X_{ikjl})$.

1.3.3.2. Else, set $f'(X_{ikjl}) = f(X_{ikjl}) + \Upsilon_t \delta \rho \sqrt{m}$.

1.3.4. Identify the move $(i^*, k^*, j^*, l^*) \in M_t^2$ as the best Type II move, similar to Step 1.2.4.

1.4. Implement the best move found above:

1.4.1. Set $X_t \leftarrow X_{i^* j^* l^*}$, if

- pass = 1 OR
- pass = 2 and $f(X_{i^* j^* l^*}) \leq f(X_{i^* k^* j^* l^*})$.

1.4.2. Else, set $X_t \leftarrow X_{i^* k^* j^* l^*}$.

1.4.3. Update P_j, P_l, A_j, A_l, R_j and R_l.

1.5. Update recency-based memory Γ:

1.5.1. Randomly generate a tabu tenure θ from $[\theta_{min}, \theta_{max}]$.

1.5.2. If a Type I move was implemented, set $\Gamma(i, j) = t + \theta$.

1.5.3. Else, set $\Gamma(i, j) = t + \theta$, and $\Gamma(k, l) = t + \theta$.

1.6. Update frequency based memory Λ^b and Λ^d:

1.6.1. If a Type I move was implemented, set

$\Lambda^b(i) \leftarrow \Lambda^b(i) + 1,$

$\Lambda^d(j) \leftarrow \Lambda^d(j) + 1,$

$\Lambda^d(l) \leftarrow \Lambda^d(l) + 1.$

1.6.2. Else, set

$\Lambda^b(i) \leftarrow \Lambda^b(i) + 1,$

$\Lambda^b(k) \leftarrow \Lambda^b(k) + 1,$

$\Lambda^d(j) \leftarrow \Lambda^d(j) + 2,$

$\Lambda^d(l) \leftarrow \Lambda^d(l) + 2.$

1.7. Update F_1, F_2, α and δ:

1.7.1. If X_t is feasible and $f(X_t) < F_1$, set $F_1 = f(X_t)$.

1.7.2. If $f(X_t) < F_2$, set $F_2 = f(X_t)$.

1.7.3. If $t \equiv 0 \pmod{\mu}$, update α:

1.7.3.1. If at least $\bar{\mu}$ of the last μ solutions were infeasible, set $\alpha \leftarrow 2\alpha$,

1.7.3.2. Else if at least $\bar{\mu}$ of the last μ solutions were feasible, set $\alpha \leftarrow \alpha/2$.

1.7.4. Set $\delta \leftarrow \max(\delta, |f(X_t) - f(X_{t-1})|)$.

2. If pass = 2, STOP. Else, set pass = 2 and go to Step 1.

Chapter 6

A Meta-Heuristic

The Tabu Search heuristic technique described in the previous chapters is known to be an effective method for solving combinatorial problems. What adds more power to this technique is its implementation within a meta-heuristic framework. A *meta-heuristic* is a master program that coordinates other heuristic(s), aiming to find better solutions than what the individual heuristics would be able to find. In this chapter, a meta-heuristic that employs the proposed Tabu Search algorithm is described. This particular one is known as *Probabilistic Diversification and Intensification* (PDI), and has been developed by Rochat and Taillard [53] originally for the vehicle routing problem. An overview of this technique is provided in Section 6.1. The details of its implementation with the political districting problem are given in Section 6.2.

6.1 The Method

The PDI technique is based on the idea that high quality (i.e., near-optimal) solutions possess high quality components and these components can be used to construct other high quality solutions. The main assumption is that some of these components are also present in the optimal solution, and therefore the objective is to find as many of such components as possible. The algorithm operates on a pool of good components, which it uses to construct a starting solution for the local search procedure it coordinates.

In the vehicle routing context, the 'components' of solutions are vehicle tours. The initial pool of tours is constructed by running the local search a pre-determined number of times and adding all generated tours to the pool. The algorithm then constructs a solution that uses as many tours from the pool as possible, and supplies it to the local search algorithm as a starting solution. Next, this solution is improved by the local search algorithm, and the tours of the best solution found is included in the pool.

The outline of the original PDI algorithm, as it appears in Rochat and Taillard [53], is reproduced below for reference:

PDI_VRP

1. Initialization

    1.1. Generate I different solutions with the local search.

    1.2. Label each tour with the value of the solution to which it belongs.

    1.3. Remove the tours having only one customer.

    1.4. Insert the remaining tours in a set T of tours.

    1.5. Sort T by increasing values of labels.

2. Diversification and intensification (to repeat until a stopping criterion is satisfied)

    2.1. Set $T' := T, S := \phi$.

    2.2. While $T' \neq \phi$, repeat

        2.2.1. Choose $t \in T'$, probabilistically, based on its relative evaluation.

        2.2.2. Set $S := S \cup \{t\}$.

        2.2.3. Remove from T' all the tours including one or several customers belonging to t.

    2.3. If some customers are not covered by the tours of S, construct a feasible solution S', including them, using the partial solution S.

    2.4. Improve with local search the solution S'.

    2.5. Label the tours of the improved solution, remove tours with only one customer, insert the remaining tours in T, sort T as in steps 1.2 to 1.5 of the initialization.

The key part of the above algorithm is to construct a feasible solution using the tours in set T. This is done by picking the tours probabilistically one after another, based on their positions in the ranking of all tours in T. Specifically, the i-the tour from the top is assigned a probability of selection equal to $\frac{|T|-i+1}{|T|\cdot(|T|+1)/2}$. As a result, tours that belong to better solutions are more likely to be selected. Once a tour is selected, all tours with whom it shares a customer are removed from consideration. The tours are selected in an iterative manner until there is no other tour left to pick. The selected tours then comprise the (partial) feasible solution S.

If S is indeed partial, i.e., it does not cover some customers, Step 2.3 is executed in order to convert it into a complete solution S'. S' is then improved by local search, and tours that belong to the improved solution are added to T. At this step, the authors discard the worst $|T| - L$ solutions in T, where L is the maximum allowable size for T. This is done in order to prevent the set T from continuously expanding. If I and L are large enough, this may not present a drawback, as the algorithm is likely to have representative solutions from different parts of the solution space. Consequently, diversification takes place as a result of the algorithm's ability to generate considerably different starting solutions for local search (especially when it uses low-rank tours). The algorithm exercises intensification by selecting tours from 'elite' solutions.

This technique can easily be modified to solve the political districting problem. The local search procedure that appears in Steps 1.1 and 2.4 is simply replaced with the proposed Tabu Search algorithm. The tours in the vehicle routing context correspond to districts in the political districting context. Customers that are part of a tour are equivalent to the base units that make up a district. The only part of the algorithm that needs a specific implementation is the construction of the complete solution S' from the partial solution S. All the details of the PDI implementation specific to the political districting problem are given in the next section.

6.2 Implementation

The PDI technique described in Section 6.1 is implemented in the political districting context as follows. First, a collection $\mathcal{D} = \{D_1, \ldots, D_{s \cdot m}\}$ of good districts is generated. This is done by running the plain TS algorithm s times, each run starting from a different (random) solution. The districts that appear in all of the s solutions reported by the plain TS algorithm are included in $\mathcal{D}$. Furthermore, let g_i be associated with each $D_i \in \mathcal{D}$, and denote the "quality" of D_i, measured as the objective value of the solution to which D_i belongs. The set $\mathcal{D}$ is sorted, at all times, in ascending order of g_i. However, if some D_i are associated with infeasible solutions, they appear in lower parts of the ranking. That is, $\mathcal{D}$ is sorted first with respect to feasibility (feasible solutions being preferred to infeasible ones), then with respect to the objective function value.

Once $\mathcal{D}$ is constructed, the next step is to run PDI for a pre-determined number of iterations, M. As higher quality solutions are identified, the districts that belong to these solutions are included in $\mathcal{D}$. The rest of the implementation follows the guidelines presented in the previous section. A few steps of the PDI implementation need further clarification, but first the outline of the PDI algorithm is provided:

PDI_POLDIS

1. Construct an initial adaptive memory of $m \cdot s$ districts:

   1.1. Run plain TS algorithm s times to obtain solutions $X_1, \ldots, X_s$.

   1.2. Set $\mathcal{D} = \{D_1, \ldots, D_{s \cdot m}\}$, the set of all districts in all X_j (repetitions allowed).

   1.3. Sort D_i in ascending order of g_i.

2. Repeat M times:

   2.1. Set $\mathcal{D}' = \mathcal{D}$.

   2.2. Construct a complete solution X using $\mathcal{D}'$:

   2.2.1. Set $L = \phi$.

   2.2.2. Repeat until $\mathcal{D}' = \phi$:

   2.2.2.1. Set $\Pr\{D_i\} = \frac{|\mathcal{D}'|-i+1}{|\mathcal{D}'|\cdot(|\mathcal{D}'|+1)/2}, \forall i = 1, \ldots, |\mathcal{D}'|$
   (probability of selecting district in the i-th position of the $\mathcal{D}'$ ranking).

   2.2.2.2. Randomly select a district $D_l \in \mathcal{D}'$, based on the probabilities $\Pr\{D_i\}$. Set $L \leftarrow L \cup D_l$.

   2.2.2.3. Remove from $\mathcal{D}'$ all districts $D_i, i \neq l$, such that $D_i \cap D_l \neq \phi$.

   2.2.3. Use collection L to construct solution X. If X is missing some BUs, go to Step 2.2.4. Otherwise, go to Step 2.3.

   2.2.4. Implement Steps 1 and 2 of the INITSOLN module to construct a set of districts L' that cover only the missing BUs from previous step.

   2.2.5. If $|L \cup L'| > m$, implement Step 3 of the INITSOLN module on $L \cup L'$ with $j = |L \cup L'|$. Let $\bar{L}$ be the resulting set of districts with $|\bar{L}| = m$.

   2.2.6. Use $\bar{L}$ to construct X.

   2.3. Improve X using the plain TS algorithm. Let X^* be the resulting solution.

   2.4. Update $\mathcal{D}$ with districts of X^*, if X^* is eligible:

   2.4.1. If X^* is feasible and $D_{s \cdot m}$ belongs to an infeasible solution $\bar{X}$ OR
   X^* is feasible, $D_{s \cdot m}$ belongs to a feasible solution $\bar{X}$, and $f(X^*) < f(\bar{X})$ OR

X^* is infeasible, D_{sm} belongs to an infeasible solution $\bar{X}$, and $f(X^*) < f(\bar{X})$, replace all districts of $\bar{X}$ in $\mathcal{D}$ with the districts of X^*. Label districts of X^* with $f(X^*)$.

2.4.2. Re-sort $\mathcal{D}$ in ascending order of g_i.

3. Report solution $\bar{X}$, the solution which $D_1 \in \mathcal{D}$ belongs to, as the best solution found.

One part of PDI_POLDIS that is implemented differently in the political districting context is the construction of $\bar{L}$, a collection of districts that cover all BUs in the territory, using the districts in L. The INITSOLN module is employed for this task and this module terminates with a collection of districts L' that cover only those BUs that are missing from the districts in L. If the number of districts in $L \cup L'$ is greater than m, Step 3 of the INITSOLN module is used to reduce it to m.

Another critical step of PDI_POLDIS is where the collection $\mathcal{D}$ is updated (Step 2.4). To keep the size of $\mathcal{D}$ fixed at $s \cdot m$, the districts of X^*, if eligible, replace the worst m districts in $\mathcal{D}$. Since both X^* and $\bar{X}$ might be infeasible, the conditions listed in Step 2.4.1 are evaluated. The collection $\mathcal{D}$ is updated, if one of these conditions is satisfied.

The last step that is described above is the mechanism for keeping the pool of high quality solutions up-to-date. It allows better solutions to enter $\mathcal{D}$ by replacing inferior ones, which consequently improves the overall quality of $\mathcal{D}$ over time. The improved collection is then used to generate even better district plans (if possible) and the collection is updated again. This process is repeated for M iterations and the best solution $\bar{X}$ encountered in the process is reported. It is also possible to report multiple solutions to the problem, which is especially important because of the multi-criteria nature of the districting problem. This would allow the analyst to compare alternative solutions based on several different criteria. The reported solutions may also be considerably different since a solution may contains districts from several different solutions.

Chapter 7

Computational Study

Like any other algorithm, the Tabu Search and PDI algorithms developed in Chapters 5 and 6 require computational testing so that the performance of the proposed algorithmic approach in solving real life political districting problems can be assessed. In this chapter, a computational study is performed with this objective. First, data requirements for the algorithms as well as a GIS interface are discussed in Section 7.1. Next, the elements of the experiment are described in Section 7.2. In Section 7.3, we describe the fine-tuning of several algorithm parameters. Finally, in Section 7.4, the results of the experiments are provided for various multi-criteria scenarios.

7.1 Data Requirements and Geographical Interfaces

7.1.1 Data Requirements

The algorithms presented in Chapters 5 and 6 require a considerable amount of data. The basic piece of data needed is the data regarding base units (BU). These data must be both geographic and demographic. The geographic component includes area and perimeter of each BU, the adjacency matrix of BUs, and the information on which BUs enclave which others. The demographic component includes data regarding the population, ethnic composition, income and education level associated with each BU.

Fortunately, all necessary data are available through Statistics Canada's databases. In this study, enumeration areas (EA) are selected as the base units, and Statistics Canada provides both demographic and geographic data for enumeration areas of any city/province in Canada. Demographically, the available data associated with each EA include population, age distribution, language spoken, ethnic origin, education, income level, religion, immigrant vs. citizen, family status, and unemployment rate. For the purposes of optimization though, only population data are used.

One also needs geographical information associated with each EA. This is necessary for optimization as well as for displaying the district plans produced by the algorithms. The algorithms need geographical data such as area and perimeters of EAs, their adjacency relationships, etc. For displaying purposes, one needs the electronic map data of the territory being districted. Most of the effort spent in this study on data preparation is related to geographical data. Therefore, these issues are discussed in a separate section below.

7.1.2 Geographical Data and Interface

As mentioned above, the algorithms developed in the previous chapters are highly dependent on geographical data. To make data preparation and output analysis easier, a Geographical Information Systems (GIS) is used in this study. Broadly speaking, a GIS is an information system that lets the user organize, analyze, query and display all sorts of geographical information. Typically, it displays electronic maps on a computer screen and lets the user perform spatial analysis using its various tools. There are many GIS software in today's market used for many different applications. Some of them have already established standard file types that are compatible with other applications. **ArcView GIS**, produced by Environmental Systems Research Institute, Inc., is currently one of the most powerful and popular desktop GIS software in the market. This particular GIS is used in this study primarily for two purposes:

- to extract geographical data from the electronic EA maps provided by Statistics Canada, and convert them into a format usable by the TS and PDI algorithms,
- to display the district plans produced and allow the user to compare them with the existing plan visually, once optimization is complete and an output is produced.

ArcView GIS was chosen for this study mainly because of its powerful tools and the script language Avenue that provides full control over the points, lines and polygons that make up an electronic map.

As far as the data extraction phase is concerned, the following data are needed by the TS algorithm:

- area and perimeter of each EA (a_i and r_i).
- area and perimeter of the entire districting area (A and R).
- adjacency information, i.e., which EA is adjacent to which other EA(s).
- length of the common boundary between two adjacent EAs (t_{ik}).
- enclave information, i.e., which EA enclaves which other EA(s).

Since ArcView does not readily provide any of these data, its script language Avenue is used in many occasions to generate the necessary data. In other words, several Avenue scripts have been developed to extract these data from the electronic EA maps provided by Statistics Canada. These Avenue scripts are outlined below.

The first item in the list is the easiest to extract. ArcView treats each EA as a polygon object and provides its area and perimeter as two properties of this object. For the second item, one needs to combine all EAs into a polygon that represents the entire territory. It is then easy to extract A and R. No Avenue script is needed for these two tasks, but one needs to know the object-oriented Avenue language to access the properties of an object.

The last three items require further work, since they are associated with *pairs* or *groups* of EAs. For the adjacency matrix, the algorithm below has been implemented with Avenue. This algorithm assumes that a view is created as part of an ArcView project and it contains two identical themes that correspond to the EA map of the territory. These themes are referred to as 'theme 1' and 'theme 2' in the algorithm.

<u>ADJMATRIX</u>

1. For each EA i in theme 1,
    1.1. Select EA i in theme 1.
    1.2. Select those EAs in theme 2 that intersect with EA i of theme 1.
    1.3. De-select EA i in theme 2.
    1.4. Create the index set L of EAs currently selected in theme 2.
    1.5. For each pair of EAs i, k where i is the EA selected in theme 1, and $k \in L$,
        1.5.1. Compute $R = r_i + r_k$.
        1.5.2. Combine the two EAs into one polygon, and denote its perimeter by r'.
        1.5.3. If $R > r'$, report EAs i and k as <u>adjacent</u>. Also report $R - r'$ as t_{ik}.

1.5.4. If $R = r'$, report EAs i and k as point-adjacent.

1.5.5. Undo the combine operation performed in Step 1.5.2.

1.6. De-select all EAs from both themes.

There are two by-products of the above algorithm. The first one is t_{ik}, the length of the common boundary between two adjacent EAs i and k. Recall that this information is needed for computation of district perimeters with equation (5.5). While EA perimeters are easily available from ArcView, t_{ik} are not, and therefore they are computed with ADJMATRIX. Note that $t_{ik} = 0$ for any non-adjacent or point-adjacent pair of EAs. The second by-product of ADJMATRIX algorithm is the point-adjacency data, which is employed by the ISCONTIGUOUS module.

The last piece of data that needs to be extracted is regarding the enclaves. Specifically, the information as to which EA enclaves which other EA(s) is needed. This is obtained using the following algorithm, which is also implemented with Avenue:

ENCLAVES

1. For each EA i in theme 1,

   1.1. Set $\mathcal{E}_i = \phi$.

   1.2. Select EA i in theme 1.

   1.3. Select those EAs in theme 2 that intersect with EA i of theme 1.

   1.4. De-select EA i in theme 2.

   1.5. Create the set L of indices of EAs currently selected in theme 2.

   1.6. Using the adjacency matrix constructed previously, partition L into $L_1, \ldots, L_q$ so that each L_j is a contiguous collection of EAs.

   1.7. For each $L_j \subset L$, check to see if any of the EAs in L_j are adjacent to an EA in $I \setminus (L \cup \{i\})$. If not, set $\mathcal{E}_i \leftarrow \mathcal{E}_i \cup L_j$.

   1.8. Report $\mathcal{E}_i$ as the set of EAs enclaved by EA i.

ArcView does not only help extract data for the algorithms, but it also acts as the interface between the algorithms and the user. In other words, it is used to communicate the results of the optimization to the analyst or decision maker. Since it is possible to summarize and analyze information easily and display it using color maps, pie charts, bar charts, graduated symbols, and other visual tools, ArcView is well-suited for reporting the results.

Typically, the optimization algorithm (PDI_POLDIS or TABU_SEARCH) solves the problem and produces an output of the best district plan found. This output has a tabular format that can be imported by ArcView. The table contains only two fields: the identification number of the EA, and the identification number of the district this EA is assigned to. The first row of the output table contains the column headings (or field names) separated by a comma. Following the first row come n rows, one row for each EA.

Once the table is imported by ArcView, it is possible to perform various types of queries with it, and it is also possible to combine it with another table based on a common field. ArcView provides the "table join" operation for the latter task. The table produced by optimization is "joined" with the attribute table of the ArcView theme, where the latter contains all the geographic and demographic information (such as identification number, area, perimeter and population) associated with each EA. The two tables are joined using the "id" field. As a result of this operation, a column (field) titled "District_Index" is appended to the attribute table and contains the assigned district index for each EA. The final step is to display the district plan by changing the legend of the theme. This is performed simply by choosing the "unique value" option for the "legend type" and the "District_index" option for the "values" field. Consequently, ArcView colors each district with a unique color. Figure 7.1 below shows a sample ArcView screen with such a district plan (using grey-scale colors).

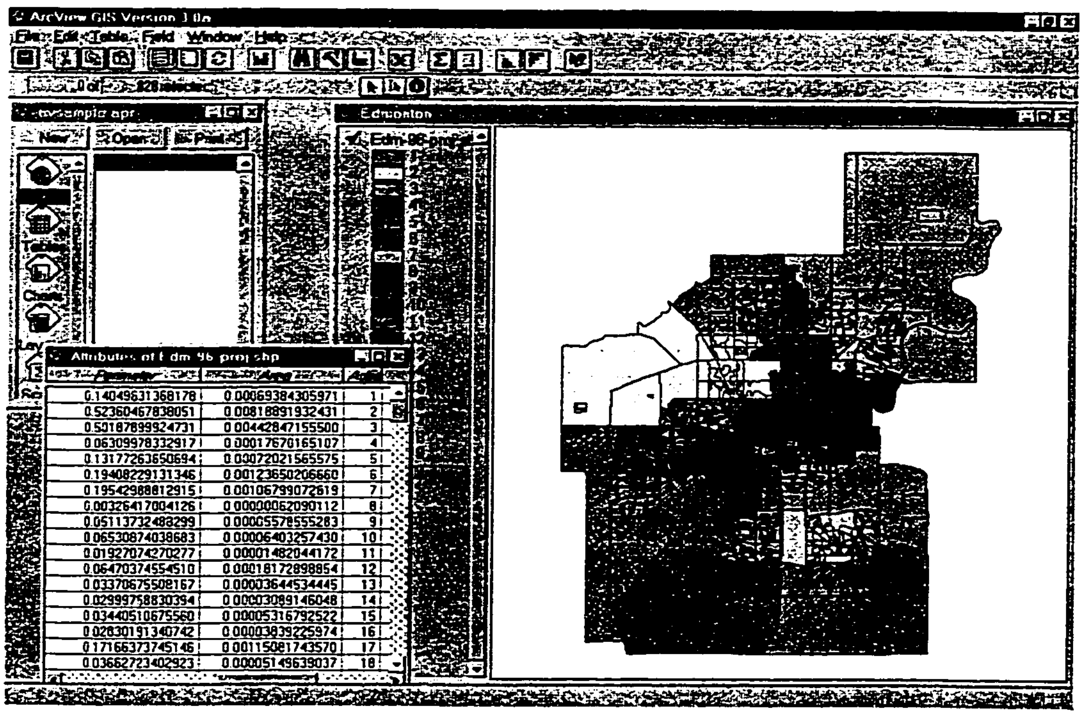

Figure 7.1: ArcView sample screen

Note that, it is also possible to create a customized application framework using Avenue scripts, where the optimization algorithms are fully integrated with ArcView and each iteration of the TS algorithm can simultaneously be displayed on the territory map. In this case, the user could also be allowed to supply the parameters of the algorithms as well as the optimization criteria. This would be desirable when creating commercial software. However, for the purposes of the computational experiments in this study, a semi-integrated system is sufficient. This involves, as described in this section, generating the input to the optimization algorithms using ArcView, and then importing the output of the algorithms back into ArcView for displaying purposes.

7.1.3 Test Data

For the computational experiments, the City of Edmonton, Alberta, is selected as the testbed. Edmonton was chosen primarily because this research is done at the University of Alberta in Edmonton, and therefore it is easier to have access to necessary data, relevant districting criteria and public opinion. The Edmonton data is provided in electronic enumeration area (EA) map format by Statistics Canada's 1996 census database as mentioned in Section 7.1.1. The data contain 828 EAs and the number of districts to be formed is 19. Associated with each EA is the population, ethnic composition, education level, and the income level information.

Another piece of data that is crucial in this study is the existing (provincial) district plan for Edmonton. This is needed for assessing the quality of the plans produced by the TS and PDI algorithms. Unfortunately, the plan is available only on paper, unlike the federal district plans that come with the electronic EA data. Furthermore, in a few cases, the district boundaries in the plan do not follow the boundaries of EAs. This constitutes a problem for comparing the plans produced by the algorithm with the one in use, since the plans will not be composed of the same building blocks. To address this issue, an approximation is made to the existing district plan, while converting it into electronic format. Wherever the boundaries of the districts do not follow EA boundaries, they are modified to do so, without resulting in a major population change in the districts. With this approximation, the existing district plan is made to be composed of the same building blocks as the plans produced by the TS and PDI algorithms. The original plan as well as the modified one are provided in Appendix A, in addition to Edmonton's EA map.

The incompatibility problem mentioned above also poses a special difficulty for implementing the proportionality criterion and for doing intentional gerrymandering. Not only the existing electoral districts do not coincide with EA boundaries, but the provincial election results are available in a format that is not compatible with the EAs. Specifically, each existing district is divided into a number of polling stations and the election results are provided for these stations individually. Unfortunately, the geographical boundaries of polling stations do not coincide with those of the EAs. Consequently, it is not possible to compute the election outcome for a hypothetical district. For this reason, the computational study presented in the following sections do not involve any testing with the proportionality criterion.

Note that this incompatibility problem could have been resolved by using *polygon overlays*, a well-known GIS tool, on the two sets of base units involved. This technique allows the user to intersect the two sets of polygons and to extract demographic information from either set for all resulting (intersection) polygons. Hence, this technique can be used to translate demographic data between two different partitionings of the same territory. In this study, we chose not to implement this method for two reasons. First, the set of polygons corresponding to the poll stations was not available electronically. Hence, one would have to digitize the polling station map which contains 1232 polling stations. Second, the polygon overlay method makes the implicit assumption that the demographic data are uniformly distributed. This assumption may not hold for certain parts of the city, and some demographic data may end up being allocated to the wrong polygon. We decided that the additional effort for digitizing the poll station map was not justified given the inaccuracy associated with the polygon overlay method, and instead we chose to use the approximation described above.

7.2 Experiment Design

The computational study presented in this chapter is composed of two stages:

> Stage 1: Experiments for finding good settings for the algorithm parameters, i.e., fine-tuning the algorithm parameters.
>
> Stage 2: Experiments for producing good district plans under various scenarios.

The TS algorithm described in the previous chapter has quite a few parameters. Before the algorithm can be employed to solve problems with the Edmonton data, these parameters should be assigned suitable values through experimentation. This constitutes the first stage of the computational study. The parameter settings identified in the first stage are then used in the second stage to solve the districting problem under different scenarios, using different sets of criteria.

7.2.1 Stage 1: Fine-tuning of Algorithm Parameters

In this first stage of the computational study, the objective is to find good settings for the algorithm parameters. Note that the objective is *not* to find a generic set of values for the parameters. The political districting problem is solved every 10 years or so, which makes it possible to pay exclusive attention to the territory being districted, in developing good district plans. For this reason, this stage is implemented exclusively for a given data set, to find parameter settings specific to that data set.

The following is the list of parameters critical to the performance of the TS and PDI algorithms:

$[\theta_{min}, \theta_{max}]$	allowable range for generating tabu tenures,
α	the starting penalty coefficient for the violation of population balance,
μ	the frequency of updating α (iteration-wise),
$\bar{\mu}$	minimum number of feasible/infeasible solutions needed to update α,
ρ	scaling factor for frequency-based memory,
T	number of successive plain TS iterations allowed without improvement,
s	size of adaptive memory (number of solutions constructed),
M	number of PDI iterations.

The first six parameters are associated with the plain TS algorithm, whereas the remaining two are associated with the PDI algorithm. The first parameter may further be considered as two parameters: $\bar{\theta} = \frac{\theta_{min}+\theta_{max}}{2}$, the mid-point of the tabu tenure range, and $\theta' = \theta_{max} - \theta_{min}$, the width of the tabu tenure range. The fine-tuning experiments are divided into two parts. In Part I, the objective is to find good settings for the listed TS parameters. In Part II, s and M are fine-tuned using the settings found in Part I.

Since it is not computationally practical, in Part I, to test all combinations of all candidate values for the seven TS parameters (even with two candidate values for each parameter), a fractional factorial design is implemented first. This step will be referred to as the screening stage. The objective here is to quickly identify the parameters that are the most influential on the quality of the best solution found by the algorithm. The parameters so-identified are further subject to detailed computational tests in Part I.

Test	Parameters $\bar{\theta}$	θ'	α	μ	$\bar{\mu}$	ρ	T
1	−	−	−	+	+	+	−
2	+	−	−	−	−	+	+
3	−	+	−	−	+	−	+
4	+	+	−	+	−	−	−
5	−	−	+	+	−	−	+
6	+	−	+	−	+	−	−
7	−	+	+	−	−	+	−
8	+	+	+	+	+	+	+

Table 7.1: 2^{7-4} Fractional factorial design

The list of the test runs for the fractional factorial design is provided in Table 7.1. Here, each parameter is allowed to take a 'low' and a 'high' value (indicated by the − and + signs, respectively), and only 8 out of the $2^7 = 128$ possible combinations are evaluated. Each test is performed for $\beta = 25\%$ and for 5 different random starting solutions, using the first compactness measure only, which gives a total of 40 runs. Once these runs are completed and the objective values reported by the algorithm are recorded, the next step is to find which parameters have made the most significant impact on solution quality in response to testing high and low values. For each parameter, the difference between the sum of the objective values for the tests with minus sign and the sum for the tests with plus sign is computed. This difference is further divided by 4, the number of pluses (or minuses) in each column. The resulting value indicates the relative influence of the parameter on the solution quality with respect to the other parameters. This analysis ignores the inter-dependence of the parameters (e.g. one parameter setting consistently performing well with a particular setting of another parameter), but it provides a quick way of identifying the most influential parameters. For further information on full factorial and fractional factorial designs, see Moen et *al.* [39].

The following table displays the low and high values tested for each of the seven parameters. Note that $\theta' = 0$ means that deterministic tabu tenures will be used for tests with low θ' value (tests 1, 2, 5, and 6). Also, the low and high values for $\bar{\mu}$ are expressed in terms of μ, and $\bar{\mu} > \mu/2$ is always maintained.

	Parameters $\bar{\theta}$	θ'	α	μ	$\bar{\mu}$	ρ	T
Low (−)	15	0	0.5	15	0.6μ	0.05	$\lceil 115\sqrt{m} \rceil$
High (+)	95	20	2	150	μ	0.2	$\lceil 230\sqrt{m} \rceil$

Table 7.2: Low-high values tested in fractional factorial experiments

Once the fractional factorial experiments are completed, a subset of the parameters are identified as the most critical ones and are subject to further detailed testing. In our case, only two parameters are chosen due to the availability of computational resources. These two parameters are then assigned several settings and the best objective values reported by the plain TS algorithm for these settings are recorded in a two-dimensional array, with rows corresponding to the settings of one parameter and the columns corresponding to those of the other parameter. This step is repeated for 20 random starting solutions. The parameter combinations that perform the best and therefore will be used in tests with PDI in Part II are then identified as follows:

First, the average of the 20 objective function values is computed for each combination. From the resulting two-dimensional 'averages' table, the following four groups of combinations are included in Φ, the set of combinations that will be tested with PDI:

1. The combination that yields the *minimum* value in the table of averages.
2. The combination that yields the minimum objective value across *all* 20 runs.
3. The combination that yields the best of the row minima and the best of column minima.
4. Compute, for each row (column), the number of times the parameter setting for that row (column) has yielded the column (row) minimum. Then, include the combination associated with the row and column having the highest such score.

If there are ties in identifying the combinations for each of these groups, all tied combinations are included in Φ. The resulting set of combinations Φ is then tested with PDI.

Before describing the details of the tests for the second part of the parameter fine-tuning (the part with PDI), note the objective function used in the above test runs. Given the weighted multi-criteria composition of the objective function, the question is which criteria should be included in the objective function at this step, and with what weight. For the fine-tuning tests, only two of the criteria described in Chapter 5 are considered. These are population balance and compactness, with weights 10 and 1, respectively. As a result, the fine-tuning runs should produce relatively compact plans that also satisfy the population balance constraint. The weights are somewhat arbitrary, but assigning 10 times more weight to population balance should be sufficient to emphasize the importance of this criterion over the other. Note that the parameters fine-tuned under these conditions may not necessarily produce good results for other criteria groups. Nevertheless, the best parameter combinations identified using the two criteria will be employed in tests with other criteria groups, since population balance and compactness will be common to all criteria groups considered and tested.

Each of the tests described above is repeated with the two compactness measures and also for β = 25%, 10%, and 5%. Recall that for the Province of Alberta, the allowable deviation from average population is 25%. While lower β values are not required by law, it is worthwhile to see if the algorithm would be able to produce a more balanced district plan that is also more compact than the existing plan.

The second part of the fine-tuning tests is implemented using the set Φ of the best combinations identified in the first part. This time, PDI parameters s and M are fine-tuned. For each combination in Φ, the initial memory is constructed using the corresponding solutions reported by the 20 plain TS algorithm runs. If $s < 20$ solutions are needed for the initial memory, the first s of the 20 solutions are considered.

The second PDI parameter that is subject to fine-tuning is M, the number of PDI iterations to be executed. Of course, the larger values for M should result in solutions of higher (or no worse) quality. The trade-off in this case is between the ultimate solution quality and the computation time. The following settings are tested for the two parameters s and M:

	Setting
s	10, 20
M	5, 10, 20

Table 7.3: Parameter settings tested for s and M.

At the end of the fine-tuning tests described above, one can decide which particular setting(s) for s, M and the two parameters from Part I should be used for the runs in Stage 2. The remaining parameters are assigned settings that are identified in fractional factorial implementation in Part I.

7.2.2 Stage 2: Experiments under Different Scenarios

In this stage, district plans are produced based on several different scenarios. Each scenario reflects a combination of both basic and other (political) criteria. As mentioned previously, the settings used in this stage for the algorithm parameters are determined in the fine-tuning stage. Table 7.4 is the list of the scenarios that are considered in this stage with the corresponding weights of the criteria involved. These scenarios are repeated for $\beta = 25\%$, 10%, and 5%.

	Weights			
	Basic		Other	
Scenario	**Pop. Bal.**	**Compact.**	**Sim. to old plan**	**Communities**
1	10	1	–	–
2	10	1	1	–
3	10	1	5	–
4	10	1	10	–
5	10	1	–	1
6	10	1	–	5
7	10	1	–	10

Table 7.4: List of scenarios considered in Stage 2

In each scenario, only three criteria at a time are considered, except the first one. This is because having too many criteria 'active' simultaneously would make it more difficult to trace the impact of each criterion on the district plan produced. Furthermore, presence of multiple conflicts among the criteria chosen would force the algorithm to move in a search direction that is hard to anticipate, since it would not be possible for the algorithm to simultaneously meet all the criteria to the desirable extent. The weight of compactness relative to population balance is set at 1:10, since this is the ratio that was used in parameter fine-tuning. The rest of the weights are simply associated with changing the influence of the remaining criteria on district plans produced. The results of the experiments under each of these scenarios are provided in Section 7.4.

7.3 Stage 1 Results: Parameter Fine-tuning

As explained in Section 7.2.1, the plain TS parameters are first subject to 'screening' tests and the two most critical parameters are identified. Next, these two parameters, followed by the PDI parameters s and M, are fine-tuned. The objective function is composed of the terms for the population balance and compactness criteria. The screening runs are performed using the first compactness measure and for $\beta = 25\%$, whereas the remaining runs are performed for both measures with $\beta = 25\%, 10\%$, and 5%. The results will be reported for each of these runs using the representation (Geography / Test-type / β / Compactness), which indicates the test data and experiment conditions to which a set of results belong. In this representation, "Geography" takes the value 'E' (for Edmonton); "Test-type" takes the values 'FT-1' (for fine-tuning part 1), 'FT-2' (for fine-tuning part 2), 'S-1', 'S-2', ... for the various scenarios considered in Stage 2; β takes values 25%, 10%, and 5%; and finally "Compactness" takes values 1 and 2, for measure 1 and measure 2, respectively. All of the runs in this stage are performed for 828 Edmonton enumeration areas with 19 districts form.

7.3.1 Screening Results

In the screening part, combinations listed in Table 7.1 of high and low parameter settings are tested, using the values in Table 7.2. The results of the screening runs are provided in Table 7.5.

Test	$\bar{\theta}$	θ'	α	μ	$\bar{\mu}$	ρ	T	S1	S2	S3	S4	S5	MIN	AVG
1	−	−	−	+	+	+	−	1.681	1.689	1.661	1.744	1.637	**1.637**	**1.682**
2	+	−	−	−	−	+	+	1.564	1.505	1.598	1.478	1.518	**1.478**	**1.532**
3	−	+	−	−	+	−	+	1.652	1.676	1.662	1.789	1.609	**1.609**	**1.678**
4	+	+	−	+	−	−	−	1.560	1.586	1.625	1.577	1.583	**1.560**	**1.586**
5	−	−	+	+	−	−	+	1.681	1.709	1.686	1.706	1.648	**1.648**	**1.686**
6	+	−	+	−	+	−	−	1.613	1.615	1.613	1.575	1.563	**1.563**	**1.596**
7	−	+	+	−	−	+	−	1.692	1.668	1.665	1.720	1.611	**1.611**	**1.671**
8	+	+	+	+	+	+	+	1.622	1.574	1.611	1.587	1.546	**1.546**	**1.588**
MIN	**-0.089**	**0.000**	**0.021**	**0.032**	**0.015**	**-0.027**	**-0.023**							
AVG	**-0.104**	**0.007**	**0.016**	**0.016**	**0.017**	**-0.018**	**-0.013**							

Table 7.5: Screening results with Edmonton data

The columns S1 through S5 display the objective values reported by the algorithm for the five random starting solutions. The columns labeled 'MIN' and 'AVG' are simply the row minimum and row average values for each of the eight tests. The last two rows of the table contain the statistics that measure the relative influence of a parameter's high and low settings on the solution quality. Recall that this statistic is computed as one-fourth the difference between the sum of the objective values corresponding to pluses and the sum of those corresponding to minuses. It is computed for each parameter based on the 'MIN' and the 'AVG' columns. For example, the statistic -0.089 for $\bar{\theta}$ is computed as $\frac{1}{4}(1.478 + 1.560 + 1.563 + 1.546 - 1.637 - 1.609 - 1.648 - 1.611)$, whereas the value -0.104 is computed using the corresponding values in the 'AVG' column. Positive (negative) values for the statistic indicate that the low (high) setting for the parameter has produced lower objective values on the average, and therefore is preferred to the high (low) setting. The magnitude of the absolute deviation from zero indicates the relative influence of this parameter on the solution quality.

The values in the 'MIN' and 'AVG' rows of Table 7.5 indicate that $\bar{\theta}$ is the most critical parameter, and therefore it is the first parameter chosen for additional testing. The second parameter, however, is not as clear. Four parameters, namely α, μ, $\bar{\mu}$ and ρ, have a roughly equal value of the statistic, based on the values in the 'AVG' row, where ρ is slightly more critical than the rest. Based on the 'MIN' row, however, μ is the most critical one. Combining results from the two rows, μ is selected as the second parameter. Notice that the statistic for θ', the width of the tabu tenure range, is almost zero, which suggests that there is no considerable difference between deterministic and random tabu tenures.

Additional tests that use a different set of high (+) settings for θ', α and ρ indicate that there is no considerable difference between $\theta' = 10$ and $\theta' = 20$, $\alpha = 1$ and $\alpha = 2$, and $\rho = 0.1$ and $\rho = 0.2$. Since the former settings are also the 'intermediate' values between all the settings tested for these parameters, they are selected as the ones that will be used in the rest of the runs. Also, when these settings are used, there is no significant difference between $\bar{\mu} = 0.6\mu$ and $\bar{\mu} = \mu$, and the latter gives slightly lower objective values than the former. As a result, $\bar{\mu} = \mu$ is used in the remaining fine-tuning experiments. To summarize, the settings $\theta' = 10$, $\alpha = 1$, $\bar{\mu} = \mu$, $\rho = 0.1$ and $T = \lceil 230\sqrt{m} \rceil$ are carried to Part I and Part II for the remaining fine-tuning tests, and also to Stage 2. The other two parameters $\bar{\theta}$ (mid-point of the tabu tenure range) and μ (frequency of updating α) are subject to further tests as detailed below.

7.3.2 Part I and Part II Results

In Part I of the fine-tuning experiments, the parameters $\bar{\theta}$ and μ are subject to detailed tests. Table 7.6 below is the list of the values that are tested for these two parameters. Note that $\bar{\theta}$ is shown as $[\theta_{min}, \theta_{max}]$ in this table, as combined with $\theta' = 10$.

Parameter	**Setting**
$[\theta_{min}, \theta_{max}]$	[10,20] [20,30] ⋮ [90,100]
μ	15, 30, 45, ..., 150

Table 7.6: Parameter settings tested for θ and μ.

Since $[\theta_{min}, \theta_{max}]$ takes on 9 settings and μ takes on 10 settings, the total number of combinations tested is 90. Each combination is tested with 20 different random starting solutions (generated with INITSOLN using 20 different random number generator seeds), for both compactness measures and for all three β values. This makes a total of 10800 plain TS runs. The combinations that perform the best are identified as described in Section 7.2.1 and they are used in Part II runs. All of the results are provided below for each β value separately:

$\underline{\beta = 25\%}$

Tables 7.7 and 7.8 on page 105 contain the average of the 20 runs for each combination of $([\theta_{min}, \theta_{max}], \mu)$ obtained for each compactness measure. With both tables, an immediate observation is that the average quality of solutions is improved (see row labeled 'AVG'), as $\bar{\theta}$ is increased. This means that the tabus *are* indeed successful in preventing cycles and reaching better solutions. With Table 7.7, the best tabu tenure ranges are observed as [80, 90] and [90, 100] (based on the row labeled 'AVG'). This result is also supported by the number of times a column value was minimum in its row (see row labeled '#'). A similar yet less apparent observation can also be made for μ. As the value of μ is increased, row minima and averages, and the number of times a row value was minimum in its column slightly deteriorate (with few exceptions; see columns 'MIN', 'AVG' and '#'). As a result, we can conclude that higher values of tabu tenure and lower values μ perform well together.

The values printed in bold indicate the best combinations to be carried to Part II of the fine-tuning tests. In Table 7.7, for example, the best column minimum (valued 1.539) and the best row minimum (valued 1.539) qualify the combination ([70, 80], 15) for inclusion in $\Phi_{25\%,1}$. The corresponding cell value 1.539 is therefore highlighted in bold. With the '#' row, there is a tie and two combinations arise as a result: ([80, 90], 15) and ([90, 100], 15). This is also the case for the 'AVG' row, and the same two combinations qualify for inclusion in $\Phi_{25\%,1}$. The last combination to enter $\Phi_{25\%,1}$ is ([90, 100], 105) and it is identified by the overall best of the 1800 individual objective function values associated with Table 7.7. This overall best solution has an objective value of 1.385 and it is obtained with random start #14. To summarize,

$$\Phi_{25\%,1} = \{([70, 80], 15), ([80, 90], 15), ([90, 100], 15), ([90, 100], 105)\}$$

is the final set of parameter combinations, associated with Table 7.7, that will be used for tests with PDI. For Table 7.8, the combinations are again indicated in bold, and the corresponding set $\Phi_{25\%,2}$ includes ([80, 90], 45), ([90, 100], 15) and ([90, 100], 60). The overall best solution in this case has objective value of 0.287.

		$[\theta_{min}, \theta_{max}]$											
		[10-20]	[20-30]	[30-40]	[40-50]	[50-60]	[60-70]	[70-80]	[80-90]	[90-100]	MIN	#	AVG
	15	1.675	1.644	1.622	1.598	1.579	1.567	**1.539**	**1.546**	**1.550**	**1.539**	**7**	**1.591**
	30	1.680	1.642	1.619	1.611	1.587	1.576	1.566	1.555	1.552	1.552	1	1.599
	45	1.689	1.651	1.627	1.614	1.588	1.573	1.572	1.565	1.561	1.561	0	1.605
	60	1.694	1.653	1.625	1.610	1.586	1.585	1.563	1.563	1.565	1.563	0	1.605
μ	**75**	1.695	1.657	1.628	1.611	1.597	1.579	1.571	1.560	1.563	1.560	0	1.607
	90	1.688	1.644	1.625	1.601	1.582	1.573	1.568	1.570	1.572	1.568	0	1.603
	105	1.690	1.645	1.626	1.598	1.589	1.579	1.572	1.562	**1.563**	1.562	1	1.603
	120	1.688	1.643	1.627	1.599	1.582	1.575	1.581	1.566	1.569	1.566	0	1.604
	135	1.690	1.635	1.623	1.602	1.583	1.577	1.583	1.564	1.560	1.560	1	1.602
	150	1.689	1.649	1.633	1.610	1.588	1.576	1.580	1.565	1.564	1.564	0	1.606
	MIN	1.675	1.635	1.619	1.598	1.579	1.567	**1.539**	1.546	1.550	1.539		
	#	0	0	0	0	0	0	3	**4**	**4**			
	AVG	1.688	1.646	1.625	1.605	1.586	1.576	1.569	**1.562**	**1.562**			

Table 7.7: Average objective function values for (E/FT-1/25%/1)

		$[\theta_{min}, \theta_{max}]$											
		[10-20]	[20-30]	[30-40]	[40-50]	[50-60]	[60-70]	[70-80]	[80-90]	[90-100]	MIN	#	AVG
	15	0.386	0.378	0.368	0.356	0.350	0.344	0.340	0.335	**0.337**	0.335	**6**	**0.355**
	30	0.388	0.380	0.368	0.359	0.352	0.345	0.340	0.336	0.335	0.335	0	0.356
	45	0.387	0.382	0.370	0.360	0.355	0.345	0.345	**0.342**	0.337	0.337	0	0.358
	60	0.388	0.381	0.369	0.360	0.352	0.347	0.345	0.339	**0.335**	**0.335**	1	0.357
μ	**75**	0.387	0.382	0.371	0.362	0.352	0.347	0.339	0.341	0.340	0.339	0	0.358
	90	0.388	0.380	0.370	0.364	0.353	0.351	0.335	0.338	0.337	0.335	1	0.357
	105	0.388	0.380	0.371	0.361	0.356	0.342	0.342	0.341	0.340	0.340	1	0.358
	120	0.388	0.381	0.372	0.362	0.356	0.349	0.341	0.340	0.337	0.337	0	0.358
	135	0.389	0.381	0.371	0.361	0.355	0.345	0.347	0.343	0.340	0.340	0	0.359
	150	0.389	0.381	0.372	0.364	0.354	0.350	0.346	0.339	0.341	0.339	0	0.360
	MIN	0.386	0.378	0.368	0.356	0.350	0.342	0.335	0.335	**0.335**	0.335		
	#	0	0	0	0	0	0	2	2	**6**			
	AVG	0.388	0.381	0.370	0.361	0.353	0.347	0.342	0.339	**0.338**			

Table 7.8: Average objective function values for (E/FT-1/25%/2)

s	M	([70,80],15)	([80,90],15)	([90,100],15)	([90,100],105)
20	5	1.392	1.437	1.430	1.385
	10	1.392	1.437	1.430	1.385
	20	1.392	1.401	1.427	**1.385**
Part-I best		1.454	1.437	1.430	1.385
	%	-4.28%	-2.51%	-0.20%	0.00%
10	5	1.440	1.449	1.413	1.458
	10	1.439	1.403	1.413	1.403
	20	1.403	**1.379**	1.384	1.396
Part-I best		1.454	1.449	1.430	1.458
	%	-3.54%	-4.85%	-3.17%	-4.23%

Table 7.9: Objective function values for (E/FT-2/25%/1)

s	M	([80,90],45)	([90,100],15)	([90,100],60)
20	5	0.287	0.301	0.297
	10	0.287	0.291	0.291
	20	**0.269**	0.278	0.291
Part-I best		0.287	0.301	0.299
%		-6.14%	-7.89%	-2.72%
10	5	0.307	0.286	0.299
	10	0.307	0.286	0.287
	20	0.295	**0.270**	0.277
Part-I best		0.307	0.301	0.299
%		-4.09%	-10.29%	-7.16%

Table 7.10: Objective function values for (E/FT-2/25%/2)

In Part II, the above combinations are used together with the test values for s and M. Tables 7.9 and 7.10 display the results for the two compactness measures. With the first table, we observe that the best solution identified by PDI (objective value 1.379) is slightly better than the best solution of plain TS. In general, even though $s = 10$ seems to improve the best solution of Part I much better than $s = 20$ does (comparing '%' rows), they perform roughly the same based on the ultimate solution quality (comparing $M = 20$ rows). Overall, $s = 10$ seems to have performed better, since many of the PDI iterations for $s = 20$ did not improve the best known solution. Also, all but one of the percent improvements indicate that PDI can considerably improve the best solution found by the plain TS algorithm.

In the second table, the best solution (objective value 0.269) comes with the $([\theta_{min}, \theta_{max}], \mu)$ combination of the best solution of Part I and for $s = 20$ and $M = 20$. An almost equally high quality solution is produced by $s = 10$ and $M = 20$ (objective value 0.270). Overall, no clear distinction between the performances of $s = 10$ and $s = 20$ can be made based on one-to-one comparisons of solutions visited at each PDI iteration (i.e., for each value of M).

The four best district plans associated with Part I and Part II of fine-tuning for each compactness measure are provided in Appendix B (Figures B.1 through B.4) together with the population of each district and the compactness value of the entire plan. Observe that with almost all districts in these plans, the boundaries coincide with the EA boundaries that are straight lines. In a few cases, the district boundaries follow roads through some neighborhoods. However, since most of these roads are represented by almost straight lines as well, from the algorithm's standpoint, they are no different from other straight lines.

Further note that some of the districts have almost perfect square shapes. Others appear to be collections of square or rectangular 'blocks', even though the entire district shape is neither square nor rectangular. A few districts traverse the periphery of the city, which suggests that these are residual districts that the algorithm may occasionally produce.

$\underline{\beta = 10\%}$

The next two tables contain results for $\beta = 10\%$. The patterns with $\beta = 25\%$ are also observed in Table 7.11, i.e., higher values for tabu tenure and lower values for μ perform well together. For Table 7.12, the values in the 'MIN' column seem rather out-of-pattern, though the '#' and 'AVG' columns recommend lower values for μ. Based on the 'MIN', '#', and 'AVG' statistics, the combinations to be used in Part II are consequently identified as

$$\Phi_{10\%,1} = \{([80,90],15],([90,100],15),([90,100],60)\} \quad (7.1)$$

$$\Phi_{10\%,2} = \{([70,80],135),([90,100],15),([90,100],90)\} \quad (7.2)$$

for measures 1 and 2, respectively. The overall best objective values are observed as 1.481 for measure 1 (with $([90,100],15)$ and random start #18) and 0.308 for measure 2 (with $([70,80],135)$ and random start #2).

The results for Part II using $\Phi_{10\%,1}$ and $\Phi_{10\%,2}$ are given in Tables 7.13 and 7.14. With the first measure, the best solution from part I is improved by 3.44% from 1.481 to 1.430, with $s = 10$, $M = 20$, $[\theta_{min}, \theta_{max}] = [80,90]$ and $\mu = 15$. In this set of runs, $s = 10$ appears to perform better than $s = 20$, based on the iteration by iteration comparison of objective values and percent improvements. With the second measure, the best solution from Part I is improved by 4.44% from 0.308 to 0.294, with the settings $s = 10$, $M = 20$, $[\theta_{min}, \theta_{max}] = [90,100]$ and $\mu = 90$. Other comparisons indicate that $s = 10$ is somewhat superior to $s = 20$, but the distinction is not as clear as in Table 7.13.

The district plans that correspond to the four best solutions mentioned above are provided in Appendix B (Figures B.5 through B.8) with the corresponding compactness scores. With these plans, an immediate observation is that the district boundaries do not follow major streets as much as they do in plans for $\beta = 25\%$. This is an expected result since the problem is more restricted with $\beta = 10\%$ than it is with $\beta = 25\%$. Consequently, the algorithm searches for new ways of clustering EAs that satisfy the $\beta = 10\%$ requirement. In this process, a district is likely to 'reach out' to EAs that will bring the district population within the $\pm 10\%$ limits. As a result, notice also that these four plans contain districts that are not as close to squares or circles as the districts in the previous case. Even though the objective function values deteriorate marginally from $\beta = 25\%$ to $\beta = 10\%$, the differences in district shapes is much more noticeable. Note that the objective values reported are equal to the compactness scores of the district plans. Since all solutions are feasible to the

		$[\theta_{min}, \theta_{max}]$										
	[10-20]	[20-30]	[30-40]	[40-50]	[50-60]	[60-70]	[70-80]	[80-90]	[90-100]	MIN	#	AVG
15	1.760	1.692	1.672	1.650	1.624	1.615	1.614	**1.590**	**1.594**	**1.590**	**5**	**1.646**
30	1.788	1.727	1.697	1.664	1.657	1.628	1.629	1.618	1.616	1.616	0	1.669
45	1.790	1.746	1.679	1.654	1.633	1.629	1.613	1.613	1.613	1.613	0	1.663
60	1.758	1.710	1.660	1.645	1.629	1.630	1.612	1.602	**1.602**	1.602	**4**	1.650
μ 75	1.770	1.735	1.687	1.656	1.650	1.648	1.631	1.618	1.615	1.615	0	1.668
90	1.782	1.711	1.685	1.658	1.649	1.628	1.620	1.620	1.616	1.616	0	1.663
105	1.780	1.726	1.697	1.671	1.637	1.632	1.637	1.620	1.619	1.619	0	1.669
120	1.774	1.735	1.693	1.667	1.650	1.649	1.639	1.634	1.633	1.633	0	1.675
135	1.782	1.725	1.680	1.665	1.653	1.651	1.626	1.639	1.623	1.623	0	1.672
150	1.777	1.720	1.690	1.675	1.653	1.643	1.628	1.640	1.626	1.626	0	1.672
MIN	1.758	1.692	1.660	1.645	1.624	1.615	1.612	**1.590**	1.594	1.590		
#	0	0	0	0	0	0	0	3	**7**			
AVG	1.776	1.723	1.684	1.661	1.644	1.635	1.625	1.619	**1.616**			

Table 7.11: Average objective function values for (E/FT-1/10%/1)

		$[\theta_{min}, \theta_{max}]$										
	[10-20]	[20-30]	[30-40]	[40-50]	[50-60]	[60-70]	[70-80]	[80-90]	[90-100]	MIN	#	AVG
15	0.404	0.389	0.377	0.363	0.360	0.354	0.351	0.349	**0.348**	0.348	**7**	**0.366**
30	0.411	0.393	0.376	0.373	0.364	0.359	0.352	0.353	0.357	0.352	1	0.371
45	0.413	0.395	0.383	0.372	0.360	0.359	0.359	0.353	0.359	0.353	0	0.372
60	0.413	0.397	0.382	0.367	0.365	0.354	0.354	0.354	0.355	0.354	0	0.371
μ 75	0.411	0.394	0.379	0.370	0.364	0.355	0.361	0.350	0.352	0.350	0	0.371
90	0.419	0.401	0.384	0.377	0.361	0.359	0.355	0.354	**0.345**	**0.345**	1	0.373
105	0.411	0.394	0.384	0.370	0.364	0.364	0.360	0.356	0.354	0.354	0	0.373
120	0.415	0.402	0.384	0.370	0.361	0.362	0.364	0.357	0.351	0.351	0	0.374
135	0.417	0.401	0.380	0.371	0.365	0.360	**0.355**	0.355	0.351	0.351	0	0.373
150	0.417	0.398	0.385	0.371	0.365	0.360	0.353	0.355	0.354	0.353	0	0.373
MIN	0.404	0.389	0.376	0.363	0.360	0.354	0.351	0.349	**0.345**	0.345		
#	0	0	0	0	0	0	3	2	**5**			
AVG	0.413	0.396	0.381	0.370	0.363	0.359	0.356	0.354	**0.353**			

Table 7.12: Average objective function values for (E/FT-1/10%/2)

s	M	([80,90],15)	([90,100],15)	([90,100],60)
20	5	1.502	1.481	1.492
	10	1.502	1.448	1.492
	20	1.502	**1.448**	1.461
Part-I best		1.542	1.481	1.492
%		**-2.62%**	**-2.25%**	**-2.08%**
10	5	1.447	1.456	1.507
	10	1.447	1.456	1.507
	20	**1.430**	1.448	1.464
Part-I best		1.542	1.528	1.531
%		**-7.27%**	**-5.23%**	**-4.39%**

Table 7.13: Objective function values for (E/FT-2/10%/1)

s	M	([70,80],135)	([90,100],15)	([90,100],90)
20	5	0.308	0.306	0.306
	10	0.308	0.300	0.306
	20	0.301	**0.295**	0.298
Part-I best		0.308	0.319	0.309
%		**-2.20%**	**-7.40%**	**-3.76%**
10	5	0.308	0.319	0.302
	10	0.308	0.310	0.302
	20	0.296	0.310	**0.294**
Part-I best		0.308	0.319	0.309
%		**-3.85%**	**-2.67%**	**-4.80%**

Table 7.14: Objective function values for (E/FT-2/10%/2)

population constraint, no population penalties are included in the objectives.

$\beta = 5\%$

When the problem is further restricted by allowing only a 5% deviation from average district population, the districts become much more distorted and the objective function values deteriorate. Tables 7.15 and 7.16 reflect this observation. As far as the parameter settings are concerned, the best combinations are

$$\begin{aligned} \Phi_{5\%,1} &= \{([70,80],150),([90,100],15),([90,100],120)\} \\ \Phi_{5\%,2} &= \{([70,80],75),([80,90],90),([90,100],60),([90,100],135)\} \end{aligned}$$

for measure 1 and measure 2, respectively. However, the numbers in the 'AVG' column of both tables appear to be different from each other by no more than 2% (most of them by no more than 1%). Furthermore, the numbers do not indicate an obvious pattern (especially with Table 7.16). This suggests that there is not much performance difference between the values tested for μ. On the average, $\mu = 120$ is the best for the first measure, and $\mu = 60$ is the best for the second measure. However, some higher values for μ also perform well. This is probably due to the fact that $\beta = 5\%$ is a much more restrictive constraint, and consequently the algorithm spends more time visiting infeasible solutions (especially at the beginning). As a result, the algorithm needs more iterations to identify and switch to a good feasible solution, before the violation penalty is doubled. When μ is small, the algorithm improves the solution quality by reducing the amount of violation, but before it can switch to a feasible solution, the amount of penalty is doubled and the process is disrupted. As a result, relatively large values of μ are also likely to have a high performance.

The overall best objective values associated with Tables 7.15 and 7.16 are 1.524 (obtained

		$[\theta_{min}, \theta_{max}]$										
	[10-20]	[20-30]	[30-40]	[40-50]	[50-60]	[60-70]	[70-80]	[80-90]	[90-100]	MIN	#	AVG
15	1.835	1.751	1.711	1.691	1.686	1.636	1.645	1.633	**1.610**	**1.610**	**3**	1.689
30	1.808	1.751	1.708	1.670	1.659	1.658	1.640	1.648	1.641	1.640	2	1.687
45	1.802	1.770	1.714	1.684	1.672	1.656	1.652	1.646	1.636	1.636	0	1.692
60	1.815	1.762	1.721	1.698	1.678	1.661	1.662	1.657	1.652	1.652	0	1.701
μ 75	1.823	1.750	1.724	1.690	1.683	1.650	1.640	1.653	1.635	1.635	0	1.694
90	1.814	1.725	1.713	1.676	1.663	1.650	1.649	1.645	1.642	1.642	1	1.686
105	1.821	1.736	1.706	1.681	1.672	1.657	1.641	1.643	1.637	1.637	0	1.688
120	1.781	1.747	1.704	1.678	1.659	1.654	1.643	1.654	**1.646**	1.643	0	**1.685**
135	1.796	1.731	1.698	1.693	1.657	1.661	1.657	1.647	1.640	1.640	1	1.687
150	1.781	1.765	1.694	1.686	1.661	1.665	**1.649**	1.646	1.643	1.643	2	1.688
MIN	1.781	1.725	1.694	1.670	1.657	1.636	1.640	1.633	**1.610**	1.610		
#	0	0	0	0	0	0	2	0	**8**			
AVG	1.808	1.749	1.709	1.685	1.669	1.655	1.648	1.647	**1.638**			

Table 7.15: Average objective function values for (E/FT-1/5%/1)

		$[\theta_{min}, \theta_{max}]$										
	[10-20]	[20-30]	[30-40]	[40-50]	[50-60]	[60-70]	[70-80]	[80-90]	[90-100]	MIN	#	AVG
15	0.426	0.401	0.393	0.389	0.372	0.375	0.356	0.363	0.355	0.355	1	0.381
30	0.431	0.406	0.395	0.387	0.371	0.374	0.372	0.355	0.361	0.355	0	0.384
45	0.421	0.407	0.388	0.378	0.374	0.369	0.361	0.372	0.363	0.361	0	0.381
60	0.424	0.403	0.389	0.380	0.368	0.362	0.356	0.359	**0.356**	0.356	1	**0.377**
μ 75	0.420	0.406	0.395	0.380	0.372	0.365	**0.372**	0.353	0.359	0.353	2	0.380
90	0.425	0.404	0.388	0.378	0.362	0.363	0.363	**0.363**	0.363	0.362	**3**	0.379
105	0.431	0.409	0.388	0.380	0.376	0.364	0.362	0.368	0.355	0.355	0	0.382
120	0.429	0.404	0.392	0.378	0.376	0.365	0.368	0.359	0.361	0.359	0	0.381
135	0.427	0.405	0.391	0.383	0.370	0.360	0.357	0.357	**0.352**	**0.352**	2	0.378
150	0.430	0.404	0.397	0.382	0.373	0.372	0.360	0.356	0.368	0.356	0	0.382
MIN	0.420	0.401	0.388	0.378	0.362	0.360	0.356	0.353	**0.352**	0.352		
#	0	0	0	0	1	0	2	**4**	3			
AVG	0.426	0.405	0.392	0.381	0.371	0.367	0.363	0.361	**0.359**			

Table 7.16: Average objective function values for (E/FT-1/5%/2)

s	M	([70,80],150)	([90,100],15)	([90,100],120)
20	5	1.524	1.492	1.514
	10	1.524	1.492	1.514
	20	1.495	**1.491**	1.514
Part-I best		1.524	1.536	1.540
%		**-1.89%**	**-2.99%**	**-1.70%**
10	5	1.542	1.506	1.562
	10	1.497	1.506	1.507
	20	1.497	**1.498**	1.503
Part-I best		1.572	1.536	1.572
%		**-4.77%**	**-2.49%**	**-4.36%**

Table 7.17: Objective function values for (E/FT-2/5%/1)

s	M	([70,80],75)	([80,90],90)	([90,100],60)	([90,100],135)
20	**5**	0.320	0.327	0.329	0.322
	10	0.320	0.324	0.325	0.322
	20	**0.316**	0.324	0.325	0.317
Part-I best		0.320	0.330	0.329	0.322
%		**-1.47%**	**-1.77%**	**-1.02%**	**-1.68%**
10	**5**	0.320	0.330	0.326	0.322
	10	0.320	0.330	0.326	0.322
	20	**0.311**	0.323	0.323	0.322
Part-I best		0.320	0.330	0.329	0.322
%		**-2.84%**	**-2.18%**	**-1.88%**	**0.00%**

Table 7.18: Objective function values for (E/FT-2/5%/2)

with $([70,80],150)$ and random start #18) and 0.320 (obtained with $([70,80],75)$ and random start #2), respectively. In Part II, these values are improved by 2.19% to 1.491 for measure 1 and by 2.88% to 0.311 for measure 2, as shown in Tables 7.17 and 7.18. Comparing the objective values iteration by iteration in each table, we observe that there is no clear distinction between the performances of $s = 10$ and $s = 20$. However, in terms of percent improvement on Part I's best solution, $s = 10$ is somewhat better than $s = 20$. One last observation is that with $\beta = 5\%$, Part II of fine-tuning is not as effective as it was for $\beta = 25\%$ and $\beta = 10\%$. This is probably due to the fact that with $\beta = 5\%$, the problem gets so restricted that even PDI cannot find many solutions of higher quality.

The district plans B.9 through B.12 in Appendix B are associated with the best solutions of Part I and II for the two compactness measures. We observe that many districts 'extend arms' to reach EAs that have populations suitable for satisfying the 5% deviation requirement. The most extreme district is in Figure B.12, which traverses the western border of the city from north to the south. Such a district would most probably be unacceptable since it combines communities from completely different parts of the city.

Another major observation with the plans is that most district boundaries no longer follow straight lines. As a result the districts are much more distorted and their shapes are long and slim. This is consistent with the algorithm's search for EA clusters that satisfy a narrow allowable population range. In spite of the deterioration in district compactness, many of the districts still look like circles or squares.

The following table summarizes the fine-tuning test results for Edmonton. All twelve best solutions previously mentioned are listed in this table with their objective values under both compactness measures. The numbers with the small fonts correspond to the measure that was NOT used for optimization. Also, the last row of this table provides the two objective function values for the existing district plan of Edmonton.

	Measure used for optimization							
	Measure 1				Measure 2			
	Part I		Part II		Part I		Part II	
	Comp1	Comp2	Comp1	Comp2	Comp2	Comp1	Comp2	Comp1
$\beta = 25\%$	1.385	0.301	1.379	0.298	0.287	1.445	0.269	1.390
$\beta = 10\%$	1.481	0.336	1.430	0.308	0.308	1.469	0.294	1.396
$\beta = 5\%$	1.524	0.335	1.491	0.311	0.320	1.533	0.311	1.546
Edm.-25%	1.638				0.369			

Table 7.19: Best solutions found for Edmonton fine-tuning

The most striking result in Table 7.19 is that even when population deviation is required to be as low as 5%, one can identify a 9% more compact plan than the current one using compactness measure 1 and a 16% more compact plan than the current one using measure 2. With $\beta = 25\%$, the best plans produced are 16% and 27% more compact than the existing plan using the two measures. Of course, it should be noted that the existing district plan has been drawn to consider other criteria as well (such as integrity of communities), and therefore the problems solved in this section are relaxations of the actual problem. Also, the objective values listed in Table 7.19 for Edmonton are computed based on the *approximation* of the original plan, and some compactness may have been lost during the approximation.

Another observation from Table 7.19 is the comparison of the compactness measures in terms of assessing the compactness of a district plan. Notice in Part I (with measure 1 optimized) that the 10% plan is more compact than the 5% plan according to compactness measure 1 (1.481 vs. 1.524). However, the compactness scores of the same two plans computed using measure 2 suggest that the 5% plan is more compact (0.335 vs. 0.336). This indicates that the two compactness measures can indeed be in conflict. This result further supports the finding by Niemi et *al.* [46], who showed the existence of conflict for some other measures as well.

Recall that the objective of Stage 1 was to find good parameter settings that will be used in Stage 2. Based on the extensive computational tests detailed so far, the parameter settings in Table 7.20 are chosen to be used for further tests with Edmonton data. With all β values, $s = 10$ and $M = 20$ has been chosen for Stage 2 simply because this selection requires construction of a smaller memory, which in turn saves computational time for more PDI iterations. Also, for the cases where $s = 20$ was superior to $s = 10$, the objective value difference was marginal. As to the issue of whether PDI is worth utilizing in spite of its long run time (see below for average times) and no more than 10% improvement over plain

TS, we note that the problem is solved every ten years, therefore a lot of computational time should be available to the analyst.

	Measure 1				Measure 2			
β	$[\theta_{min}, \theta_{max}]$	μ	s	M	$[\theta_{min}, \theta_{max}]$	μ	s	M
25%	[80,90]	15	10	20	[90,100]	15	10	20
10%	[80,90]	15	10	20	[90,100]	90	10	20
5%	[90,100]	15	10	20	[70,80]	75	10	20

Table 7.20: Best parameter settings found in Stage 1

One can further reduce the β values to see how the district plans will be affected. The smallest test value that yielded a feasible plan was 1%. This plan is provided in Figure B.13. As expected, the districts are highly non-compact, created in order to satisfy the $\beta = 1\%$ requirement. But at least it shows that almost perfect population equality is possible.

So far, no discussion has taken place on how long it takes the algorithm to run, and how long it takes to find the best solution it reports. Based on the 1800 runs for $\beta = 25\%$ with compactness measure 1, the average duration of a plain TS run is 217.1 seconds (about 3.5 minutes) and it takes 126.0 seconds on the average the find the best solution it reports. PDI is essentially a sequence of TS runs, and therefore it takes a lot longer to complete a PDI run than to complete a plain TS run. Based on the PDI runs for $\beta = 25\%$ with compactness measure 1 (i.e., for a total of 4 PDI runs), it takes on the average 57.8 minutes to complete 20 PDI iterations, excluding the construction of initial memory. The time it takes to construct the initial memory is clearly a function of the average plain TS run length, depending on the size of the initial memory. All run times mentioned above are recorded on a Pentium 233MMX PC with 64M RAM.

7.4 Stage 2 Results: Experiments under Different Scenarios

In this section, we consider the seven multi-criteria scenarios listed in Table 7.4 and produce alternative district plans for the City of Edmonton. The results for these scenarios are provided below.[1]

[1] Based on Stage 1 results, observe that a compactness score computed with measure 1 is one order of magnitude larger than that computed with measure 2. (Measure 2 is always bounded above by 1 whereas measure 1 can (and does) take on values greater than 1. It is not trivial to find an upper bound on measure 1.) This suggests that the *effective* relative weight of compactness relative to the other criteria is higher with measure 1 than with measure 2. Hence, one would expect more compact plans with measure 1 than with measure 2.

Scenario 1:

Recall that this scenario involves only population equality and compactness, and therefore was completed as part of the fine-tuning experiments. For easy reference, Table 7.21 summarizes the objective function values with each compactness measure for each β value.

β	Measure 1	Measure 2
25%	1.379	0.269
10%	1.430	0.294
5%	1.491	0.311

Table 7.21: Objective values for Scenario 1

The six district plans associated with the objective values in Table 7.21 are provided in Appendix B (Figures B.3, B.4, B.7, B.8, B.11, and B.12). Under this scenario, because of a relatively large weight attached to population equality, the algorithm first eliminates all infeasibilities (i.e., violations of population constraints) as soon as possible. Then it maximizes compactness subject to population constraints. Note that the solutions that are provided for this scenario will be used as a base case. By inclusion of additional criteria, one would then be able to see how compactness will be traded off against other criteria, i.e., how the plans will compromise compactness to address other criteria.

Scenario 2:

Under this scenario, we consider similarity to the existing plan as the third criterion. That is, the second term in the objective function is now associated with this criterion. The individual weights for this scenario are: $w' = 10$, $w_1 = 1$ (compactness), and $w_2 = 1$ (similarity to existing plan). Since population equality is treated as a constraint whose violations are penalized, and also its weight is relatively higher than that of the other two criteria, one should expect the algorithm to maintain population equality first, then find a solution that balances the other two criteria.

The objective function values for the six PDI runs performed for this scenario are given in Table 7.22. The breakdown of each objective value is also given in the table with respect to the compactness score and the similarity score, respectively. Recall from Table 7.19 that the existing Edmonton districts have a compactness score of 1.638 and the most compact plan found in Stage 1 has that of 1.379, both based on measure 1. While searching for plans that are similar to the existing plan, the algorithm will of course compromise compactness, and therefore will identify solutions that have compactness scores higher than 1.379. Notice in Table 7.22 for all six cases that the compactness scores are worse than the corresponding values in Table 7.19. In four of these cases (three cases for measure 1 and $\beta = 25\%$ case for measure 2), the plans are still more compact than the existing Edmonton plan.

β	Measure 1		Measure 2	
25%	**1.728**		**0.428**	
	$f_1 = 1.437$	$f_2 = 0.291$	$f_1 = 0.351$	$f_2 = 0.077$
10%	**1.763**		**0.450**	
	$f_1 = 1.501$	$f_2 = 0.262$	$f_1 = 0.378$	$f_2 = 0.072$
5%	**1.817**		**0.472**	
	$f_1 = 1.518$	$f_2 = 0.299$	$f_1 = 0.378$	$f_2 = 0.094$

Table 7.22: Objective values for Scenario 2

Another major observation in Table 7.22 is that the second measure consistently produces plans that are more similar to the existing plan than those produced by measure 1. This is in fact explained by the difference between the range of values each measure takes. The second measure consistently takes lower values compared to the first measure, and this reduces the importance of the compactness criterion relative to the other two criteria. Consequently, similarity to the old plan gains more weight, and the resulting plans have better score for this criterion.

The district plans produced under Scenario 2 are given in Appendix C. Comparing these plans to the existing district plan, we find that the proposed boundaries do not often follow the boundaries of existing districts. However, we observe that with almost all districts, a majority of EAs that are in an existing district also appear together in a proposed district. This is exactly what the similarity index measures, and the district plans visually verify that the index achieves its intended functionality.

<u>**Scenario 3:**</u>

This scenario is different from the previous one only in that w_2, i.e., the weight of the second criterion (similarity to the old plan), is increased to 5. This should cause the algorithm to switch to less compact plans that are more similar to the existing plan than the ones in Scenario 2. However, for $\beta = 25\%$, the compactness of such plans should be no worse than that of the existing plan, since otherwise they would be dominated by the existing plan. The figures in Table 7.23 suggest that this is indeed the case for $\beta = 25\%$, i.e., the two plans are still slightly more compact than the existing plan, and they are more similar to the existing plan than the corresponding plans in Scenario 2. With other β values, the compactness scores of the plans are worse than that of the existing plan simply because the algorithm has to satisfy a more strict population equality constraint. Also notice in

Table 7.23 that the second measure again produces plans with lower (i.e., better) similarity scores than the first one.

β	Measure 1		Measure 2	
25%	**2.060**		**0.522**	
	$f_1 = 1.622$	$f_2 = 0.088$	$f_1 = 0.360$	$f_2 = 0.034$
10%	**2.224**		**0.577**	
	$f_1 = 1.753$	$f_2 = 0.094$	$f_1 = 0.403$	$f_2 = 0.035$
5%	**2.473**		**0.812**	
	$f_1 = 1.764$	$f_2 = 0.142$	$f_1 = 0.432$	$f_2 = 0.076$

Table 7.23: Objective values for Scenario 3

The district plans associated with these six solutions are given in Appendix D. We can observe that the districts are less compact compared to Scenarios 1 and 2, and the proposed boundaries increasingly coincide with the boundaries of existing districts. A few districts are even exactly the same as they appear in the existing plan. In this case, a decision maker is simply in a position to choose a slightly more compact plan than the existing plan, or the existing plan itself.

<u>Scenario 4:</u>

Under this scenario, the weight of the similarity criterion is set equal to the weight of the population balance criterion (i.e., $w' = w_2 = 10$). Consequently, one could expect the algorithm to create district plans that are the most similar to the existing plan among all the plans constructed so far. Because of the increased value of w_2, the plans may even be expected to violate population equality in exchange for improved similarity score. This, however, should not be relevant for $\beta = 25\%$, since the existing plan itself is balanced in population. The objective values in Table 7.24 in fact present an interesting case. For $\beta = 25\%$ with the first measure, the algorithm suggests a plan that is slightly less compact than the existing plan. Why would the algorithm report a dominated solution as the best solution it finds? One reason might be the selection of the parameters used. In fact, when an alternative parameter combination is tried (one of the combinations tested in Stage 1), the resulting solution turned out to be slightly more compact. For the second measure (with $\beta = 25\%$), the algorithm reports the existing solution itself as the best solution it finds. In this case, $w_2 = 10$ turned out to be sufficiently high that the algorithm reached one extreme of the efficient frontier. With the remaining β values, the algorithm reports less compact districts again, attributable to the tightened population equality constraint.

The district plans associated with the above solutions are provided in Appendix E. Observe

β	Measure 1		Measure 2	
25%	**2.203**		**0.369**	
	$f_1 = 1.652$	$f_2 = 0.055$	$f_1 = 0.369$	$f_2 = 0$
10%	**2.564**		**0.749**	
	$f_1 = 1.804$	$f_2 = 0.076$	$f_1 = 0.424$	$f_2 = 0.033$
5%	**2.591**		**1.208**	
	$f_1 = 1.792$	$f_2 = 0.080$	$f_1 = 0.434$	$f_2 = 0.077$

Table 7.24: Objective values for Scenario 4

that many of the districts in the existing plan appear mostly intact in the proposed plans, even for $\beta = 5\%$. Increased compactness scores are of course the result of the effort to reach out to some EAs in order to satisfy the population constraint.

The last three scenarios implemented in this section consider integrity of communities instead of similarity to the existing plan. In order to use this criterion for optimization, one needs to have a pre-defined community structure, as described in Section 5.3.4. The one that is used for scenarios 5-7 is given in Appendix A (Figure A.4). In this 'community plan', we divide Edmonton into 9 regions that mostly correspond to collections of neighborhoods with similar characteristics. The division is made subjectively in that it is based mostly on the geographical locations and other characteristics that would distinguish each community from the others (example: the Old Strathcona, i.e., community #3, which is a mostly residential area close to the university containing a lively street with cafes and restaurants).

<u>Scenario 5:</u>

This scenario considers population balance, compactness and integrity of communities with the following respective weights: $w' = 10$, $w_1 = 1$, and $w_2 = 1$. In this case, the last criterion is associated with the term $f_2(X)$ in the objective function.

The objective function values of the best solutions found are given in Table 7.25 for each β value and compactness measure. The corresponding district plans are in Appendix F. Recall that the compactness scores (i.e., the f_1 value) of the current electoral districts of Edmonton are 1.638 for measure 1 and 0.369 for measure 2. Based on the community plan used, the f_2 score of Edmonton is computed as 0.569. A comparison of the existing plan with those produced under the current scenario shows that for $\beta = 25\%$, the two proposed solutions both dominate the existing district plan. As β is decreased, measure 1 still generates more compact solutions than the existing plan and measure 2 generates solutions that maintain integrity of communities better than the existing plan (again possibly due to the scaling issue described previously). Even for $\beta = 10\%$, the algorithm is able to identify a solution that is superior to the existing plan in both criteria (with measure 1).

β	Measure 1		Measure 2	
25%	**1.998**		**0.821**	
	$f_1 = 1.505$	$f_2 = 0.493$	$f_1 = 0.367$	$f_2 = 0.455$
10%	**2.088**		**0.867**	
	$f_1 = 1.555$	$f_2 = 0.533$	$f_1 = 0.373$	$f_2 = 0.494$
5%	**2.172**		**0.886**	
	$f_1 = 1.582$	$f_2 = 0.590$	$f_1 = 0.393$	$f_2 = 0.493$

Table 7.25: Objective values for Scenario 5

A visual check on the corresponding maps suggests that especially for $\beta = 25\%$, many of the districts are composed of one particular community only. The district plans for $\beta = 10\%$ and $\beta = 5\%$ appear less successful in maintaining integrity of communities, simply because some of the districts may have to extend arms to nearby EAs in order to satisfy the population constraint. Consequently, this may split the largest part of a community that still remains intact, or 'steal' EAs from it. Recall that the integrity of communities measure developed in Chapter 5 does not directly adjust the district boundaries to make them follow community boundaries. Instead, it checks how many pieces a community is divided into and tries to maximize the area of the largest piece. As a result, a district plan that is better than another in terms of the objective value may not look so visually. The key component here is not the community borders, but the largest area within a community that is maintained in a district.

<u>Scenario 6:</u>

In this case, the weight of the integrity of communities criterion is increased to 5. The numbers in Table 7.26 suggest that with this setting, the algorithm can no longer produce solutions that are more compact than the existing plan, due to the increased importance of the integrity of communities. But as a result, all solutions that it produces are better than the existing plan with respect to the community criterion. Also, all of the six solutions in the table are non-dominated solutions compared to the existing plan. This gives the decision maker the opportunity to decide between more compact solutions and the ones that maintain communities better, with the added flexibility of choosing a β level. The decision maker should of course be careful about choosing a low β value since this would in turn increase f_2 values. In fact, this is the (stated) reason behind the 25% deviation allowed by the Electoral Boundaries Commission Act: to be able to keep communities together to a larger extent.

β	Measure 1		Measure 2	
25%	**3.892**		**2.438**	
	$f_1 = 1.754$	$f_2 = 0.428$	$f_1 = 0.434$	$f_2 = 0.401$
10%	**4.036**		**2.709**	
	$f_1 = 1.692$	$f_2 = 0.469$	$f_1 = 0.430$	$f_2 = 0.456$
5%	**4.276**		**2.883**	
	$f_1 = 1.740$	$f_2 = 0.507$	$f_1 = 0.520$	$f_2 = 0.473$

Table 7.26: Objective values for Scenario 6

The maps associated with the above solutions are provided in Appendix G. Notice that since the relative importance of compactness is lower than what it was under Scenario 5, the resulting districts are highly non-compact. In many cases, residual districts have formed, as the PDI algorithm has attempted to collect districts with a good level of community integrity. Even though compactness is compromised, communities are preserved better under this scenario. A decision maker who weighs integrity of communities considerably more than compactness may well find these plans appealing.

<u>**Scenario 7:**</u>

The last scenario considered in this section assigns $w_2 = 10$ to integrity of communities, a weight equal to the weight of population equality and 10 times the weight of compactness. As a result of the further decreased relative importance of compactness, the resulting plans are highly non-compact compared to the previous ones and the existing plan (see Figures H.1 through H.6). However, all six solutions satisfy the population constraint and have improved integrity of communities in comparison to Scenarios 5 and 6, therefore they are non-dominated solutions. One observation under this scenario is that the algorithm becomes more sensitive to a change in β, compared to previous cases, that f_1 and f_2 change rather significantly as β is decreased. The solutions found may still be acceptable though, if the decision maker's underlying weights favor integrity of communities criteria a lot more than compactness.

β	Measure 1		Measure 2	
25%	**5.964**		**4.369**	
	$f_1 = 1.866$	$f_2 = 0.410$	$f_1 = 0.502$	$f_2 = 0.367$
10%	**6.453**		**4.899**	
	$f_1 = 1.844$	$f_2 = 0.461$	$f_1 = 0.546$	$f_2 = 0.435$
5%	**6.549**		**5.112**	
	$f_1 = 1.773$	$f_2 = 0.478$	$f_1 = 0.489$	$f_2 = 0.462$

Table 7.27: Objective values for Scenario 7

Overall, the experiments performed in Stage 2 show how the weighting method of multi-criteria decision making can help identify non-dominated solutions. As a summary, consider Table 7.28, the table of objective function values, for $\beta = 25$ only, for each objective function term of each solution found under Scenarios 1-7.

	Measure 1				Measure 2			
Scenario	**Pop.**	**Comp.**	**Simil.**	**Comm.**	**Pop.**	**Comp.**	**Simil.**	**Comm.**
1	0	1.379	–	–	0	0.269	–	–
2	0	1.437	0.291	–	0	0.351	0.077	–
3	0	1.622	0.088	–	0	0.360	0.034	–
4	0	1.652	0.055	–	0	0.369	0	–
5	0	1.505	–	0.493	0	0.367	–	0.455
6	0	1.754	–	0.428	0	0.434	–	0.401
7	0	1.866	–	0.410	0	0.502	–	0.367
Edm-25%	0	1.638	0	0.569	0	0.369	0	0.569

Table 7.28: Objective values with $\beta = 25\%$ for all scenarios

All but one of the solutions in Table 7.28 are non-dominated with respect to Edmonton's existing plan. In fact, the two solutions found under Scenario 5 are superior to the existing plan in all of the criteria considered. The only solution in Table 7.28 that was dominated by the existing plan was under Scenario 4 with compactness measure 1.

The solutions generated in this section by the multi-criteria approach constitute a very small subset of all the solutions that can be generated. Typically, one could attach weights to criteria in numerous different combinations, in order to obtain a larger set of non-dominated solutions. The multi-criteria framework is helpful to a decision maker especially in this regard. Since the task of generating a district plan 'fair' to everyone is quite challenging, the more the number of available alternatives is, the better the decision maker can weigh criteria against each other and come up with a compromise solution. In our view, the multi-criteria approach presented in this section is one of the most important contributions of this dissertation.

Chapter 8

Conclusion

In this dissertation, we have studied the political districting problem. This is a problem that is usually solved every ten years, to account for population movements in the region being districted and draw new districts accordingly. What makes it an interesting problem is its highly political multi-criteria nature. On the one side, there is a districting authority that is required by law to create a district plan that is fair to everyone. On the other side, there are political parties and candidates running for the election who wish the districts to be drawn to their advantage and to their opponents' disadvantage. As well, some of the popular criteria of redistricting such as compactness, proportionality and communities of interest can be in conflict. With all these competing objectives, it is not easy to identify a solution that satisfies everyone.

The algorithmic approach proposed in this dissertation is intended to provide a structured framework for solving the political districting problem. Unlike most of the past studies, we consider political criteria in addition to basic criteria, recognizing that such criteria are real and are used by districting authorities. Our approach is a multi-criteria approach, i.e., what we are trying to optimize is a combination of various interests. Consequently, a district plan produced would address each dimension to some extent, some more strongly than others. The criteria that we consider include population equality, contiguity, compactness, integrity of communities, similarity to the existing plan and proportionality. As a result, this study is unique in that it brings both basic and political criteria into the overall picture. Past studies which have excluded political criteria have either argued that the district plans would then be free of political influence and gerrymandering, or they preferred to work on a single objective function with a very limited number of constraints.

The solution methodologies presented in the previous chapters utilize the Tabu Search modern heuristic technique and a meta-heuristic known as Probabilistic Diversification and Intensification (PDI). These are problem solving methods that have proved very effective in solving a variety of problems, and therefore have become very popular in the last decade. A unique contribution of this dissertation to the literature is the application of these intelligent problem solving techniques to the political districting problem. While the general idea of starting with a district plan and changing elements of this plan to reach better solutions is not new, the use of intelligent memory in the process as well as combining districts from multiple plans *are*. The latter feature also fits well into the multi-criteria framework. The characteristic of the technique that good districts are brought together from multiple solutions may result in significantly different solutions, which may be highly desirable from a multi-criteria perspective. The decision maker would then compare each solution and weigh its score under each criterion to reach a decision.

The Tabu Search technique is also well-suited to problems with constraints that are difficult to deal with in a mathematical program. The contiguity requirement in the districting problem is an example of such a constraint. As well, mathematical programs for the districting problem can only be solved for relatively small instances. In contrast, TS is a heuristic approach that can handle much larger problem instances. It is possible to solve this problem with other heuristic approaches. However, TS searches the solution space in an efficient and effective way, and it has outperformed other heuristic techniques in computational tests on a wide variety of combinatorial problems (e.g. Gendreau et *al.* [17]).

The performance of the proposed algorithms is tested using the City of Edmonton enumeration area data, which contain not only the geographic units that make up the districts, but also various kinds of useful demographic information. To manage the large amount of data needed by the algorithms, we use the ArcView Geographical Information System. This GIS is used mainly for extracting geographic data from the enumeration area maps. A second use of the system is to read the output file of the Tabu Search algorithm and display the produced district plan as an electronic map. Next, the decision maker can use the powerful tools of this GIS to summarize and visualize information on the composition of the districts. This GIS interface between the optimization algorithms and the end-user is another important component of this dissertation. Currently, we know of no commercial product in the market that automates the re-districting process. While districting authorities have recently started to use geographical information systems for displaying maps and computing statistics, no such system that combines optimization and GIS currently exists. This dissertation provides a prototype for such an integrated system.

An important part of this dissertation was to assess the algorithm's performance by comparing the district plans it produces with the existing district plan of Edmonton. Before the algorithm can be run to produce acceptable district plans, one needs to calibrate its parameters. Since the proposed Tabu Search algorithm has many parameters, whose appropriate values are not known in advance, the first step is a parameter fine-tuning step. We implement this step for calibrating four parameters of the algorithm in detail. The best choices of parameters are then used to run the algorithm to solve various multi-criteria scenarios. The computational results at the fine-tuning step indicate that PDI (which executes a sequence of Tabu Search runs) produces plans that are 9% to 27% more compact than the existing plan.

In the multi-criteria stage, we consider combinations of basic and political criteria. The basic criteria include population equality and compactness, which are always in the criteria set. The combinations of criteria we consider include population equality, compactness and similarity to the old plan as one group, and population equality, compactness and integrity of communities as the second group. With both groups of criteria, we change the relative weights of the criteria to generate alternative district plans. As a result, several district plans that address different dimensions of the districting process are produced. This is perhaps the most important aspect of this study in that the more diversity there is among the alternatives, the more aware the decision maker will be of the impacts of the districting decision. The alternative plans produced in this study provide a good diversity in this regard, with respect to the criteria considered.

Note that the methodology developed in this study may also be useful for solving other types of districting problems. For example, in the salesman districting case, one is usually interested in creating districts of roughly equal workload or sales revenue. Compactness is a valid criterion in this context as well, since one would attempt to minimize traveling costs within a district. Another example is the school districting problem, where the districts may not have to be balanced, but the student profile of each school might become relevant. For example, a valid criterion might be assigning residential areas to a school in a way that the school has a certain diversity of ethnic origin. With both of these problems, the mechanics of the algorithm would remain the same (i.e., starting with a partition, which would be modified by swap or exchange of smaller units), but the districting criteria would change. As a result, one could still use the techniques presented in this dissertation by using a modified set of criteria.

Bibliography

[1] 1995/1996 ALBERTA ELECTORAL BOUNDARIES COMMISSION *Proposed Electoral Division Areas, Boundaries and Names for Alberta*, World Wide Web reference: www.assembly.ab.ca/electora/reports/toc.htm.

[2] ADAMS, B. (1977) A Model State Reapportionment Process: The Continuing Quest for 'Fair and Effective Representaion'. *Harvard Journal on Legislation*, 14, pp. 825-904.

[3] BOYCE, R.R. AND W.A.V. CLARK (1964) The Concept of Shape in Geography. *The Geographical Review*, pp. 561-572.

[4] BOURJOLLY, J-M., G. LAPORTE AND J-M. ROUSSEAU(1981) Decoupage Electoral Automatise: Application a L'Ile de Montreal. *Infor*, 19(2), pp. 113-124.

[5] BROWDY, M.H. (1990) Simulated Annealing: An Improved Computer Model for Political Redistricting. *Yale Law and Policy Review*, 8(146), pp. 163-179.

[6] CORDEAU, J.F., M. GENDREAU AND G. LAPORTE (1995) A Tabu Search Heuristic for Periodic and Multi-Depot Vehicle Routing Problems. Research Report No. CRT-95-76, Centre for Research on Transportation, University of Montreal.

[7] COURTNEY, J.C. (1996) Drawing Electoral Boundaries. In *Canadian Parties in Transition*, 2nd ed., A.B. Tanguay and A-G. Gagnon (Eds.), Toronto: Nelson Canada.

[8] COX, E.P. (1927) A Method of Assigning Numerical and Percentage Values to the Degree of Roundness of Sand Grains. *Journal of Paleontology*, 1, pp. 179-183.

[9] DIXON, R.G.Jr. (1968) *Democratic Representation: Reapportionment in Law and Politics*, New York: Oxford University Press.

[10] EIG, L.M. AND M.V. SEITZINGER (1981) State Constitutional and Statutory Provisions Concerning Congressional and State Legislative Redistricting. Congressional Research Service, Report No. 81-143A.

[11] ELECTIONS ALBERTA (1997) *Report of the Chief Electoral Officer.* Edmonton: Province of Alberta.

[12] FLEISCHMANN, B. AND J.N. PARASCHIS (1988) Solving a Large Scale Districting Problem: A Case Report. *Computers and Operations Research*, 15(6), pp. 521-533.

[13] FORD, L.E. AND D.R. FULKERSON (1962) *Flows in Networks*, pp. 162-169, Princeton University Press.

[14] GAREY, M.R. AND D.S. JOHNSON (1979) *Computers and Intractability: A Guide to the Theory of NP-Completeness.* San Francisco: W.H. Freeman and Company.

[15] GARFINKEL, R.S. AND G.L. NEMHAUSER (1970) Optimal Political Districting by Implicit Enumeration Techniques. *Management Science*, 16(8), pp. B495-B508.

[16] GEARHART, B.C. AND J.M. LIITTSCHWAGER (1969) Legislative Districting by Computer. *Behavioral Science*, 14, pp. 404-417.

[17] GENDREAU, M., A. HERTZ AND G. LAPORTE (1994) A Tabu Search Heuristic for the Vehicle Routing Problem. *Management Science*, 40(10), pp. 1276-1290.

[18] GEORGE, J.A., B.W. LAMAR AND C.A. WALLACE (1993) Determining New Zealand Electoral Districts Using a Network-based Model. *Operational Research Society of New Zealand, Proceedings*, 29, pp. 276-283.

[19] GIBBS, J.P. (1961) A Method for Comparing the Spatial Shapes of Urban Units. In *Urban Research Methods*, J.P. Gibbs (Ed.), New York: Van Nostrand.

[20] GLOVER, F. (1986) Future Paths for Integer Programming and Links to Artificial Intelligence. *Computers and Operations Research*, 13, pp. 533-549.

[21] GLOVER, F. (1990) Tabu Search: A Tutorial. *Interfaces*, 20(4), pp. 74-94.

[22] GLOVER, F. AND M. LAGUNA (1993) Tabu Search. Chapter 3 in *Modern Heuristic Techniques for Combinatorial Problems*, C.R. Reeves (Ed.), pp. 70-150, New York: Wiley.

[23] GLOVER, F. AND M. LAGUNA (1998) *Tabu Search*, 2nd ed., Boston: Kluwer.

[24] GROFMAN, B. (1985) Criteria for Districting: A Social Science Perspective. *UCLA Law Review*, 33, pp. 77-183.

[25] GROFMAN, B., A. LIJPHART, R.B. McKAY AND H.A. SCARROW (1982) *Representation and Redistricting Issues*, Lexington: Lexington Books.

[26] HARRIS, C.C.Jr. (1964) A Scientific Method of Districting. *Behavioral Science*, 9, pp. 219-225.

[27] HESS, S.W., J.B. WEAVER, H.J. SIEGFELDT, J.N. WHELAN AND P.A. ZITLAU (1965) Nonpartisan Political Districting By Computer. *Operations Research*, 13, pp. 998-1006.

[28] HOFELLER, T. AND B. GROFMAN (1990) Comparing the Compactness of California Congressional Districts under Three Different Plans: 1980, 1982, and 1984. In *Toward Fair and Effective Representaion*, B. Grofman (Ed.), New York: Agathon.

[29] HORN, M.E.T. (1995) Solution Techniques for Large Regional Partitioning Problems. *Geographical Analysis*, 27(3), pp. 230-248.

[30] HORTON, R.E. (1932) Drainage Basin Characteristics. *Transactions of the American Geophysical Union*, 13, pp. 350-361.

[31] JOHNSTON, J.P. AND H.E. PASIS (1990) *Representation and Electoral Systems*, Scarborough: Prentice-Hall.

[32] KAISER, H.F. (1966) An Objective Method for Establishing Legislative Districts. *Midwest Journal of Political Science*, 10, pp. 200-213.

[33] LEWYN, M.E. (1993) How to Limit Gerrymandering. *Florida Law Review*, 45, pp. 403-486.

[34] LIITTSCHWAGER, J.M. (1973) The Iowa Redistricting System. *Annals of the New York Academy of Sciences*, 219, pp. 221-235.

[35] LIJPHART, A. (1990) Criteria of Fair Representation. In *Representation and Electoral Systems: Canadian Perspectives*, J.P. Johnston and H.E. Pasis (Eds.), Scarborough: Prentice-Hall.

[36] MARLIN, P.G. (1981) Application of the Transportation Model to a Large-scale "Districting" Problem. *Computers and Operations Research*, 8, pp. 83-96.

[37] MEHROTRA, A., E.L. JOHNSON AND G.L. NEMHAUSER (1998) An Optimization Based Heuristic for Political Districting. *Management Science*, 44(8), pp. 1100-1114.

[38] MILLS, G. (1967) The Determination of Local Government Electoral Boundaries. *Operations Research Quarterly*, 18, pp. 243-255.

[39] MOEN, R.D., T.W. NOLAN AND L.P. PROVOST (1991) *Improving Quality Through Planned Experimentation*. New York: McGraw-Hill.

[40] MORRILL, R.L. (1973) Ideal and Reality in Reapportionment. *Annals of the Association of American Geographers*, 63(4), pp. 463-477.

[41] MORRILL, R.L. (1976) Redistricting Revisited. *Annals of the Association of American Geographers*, 66, pp.548-556.

[42] MORRILL, R.L. (1981) *Political Redistricting and Geographic Theory*, Washington D.C.: Association of American Geographers.

[43] MORRILL, R.L. (1991) Making Redistricting Models More Flexible and Realistic. *The Operational Geographer*, 9, pp. 2-9.

[44] NAGEL, S.S. (1965) Simplified Bipartisan Computer Redistricting. *Stanford Law Review*, 17, pp. 863-899.

[45] NAGEL, S.S. (1972) Computers and the Law and Politics of Redistricting. *Polity*, 5, pp. 77-93.

[46] NIEMI, R.G., B. GROFMAN, C. CARLUCCI AND T. HOFELLER (1990) Measuring Compactness and the Role of a Compactness Standard in a Test for Partisan and Racial Gerrymandering. *Journal of Politics*, 52(4), pp. 1155-1181.

[47] OSMAN, I.H. AND G. LAPORTE (1996) Metaheuristics: A Bibliography. *Annals of Operations Research*, 63, pp. 513-623.

[48] OWEN, G. AND B. GROFMAN (1988) Optimal Partisan Gerrymandering. *Political Geography Quarterly*, 7(1), pp. 5-22.

[49] PAPAYANOPOULOS, L. (1973) Quantitative Principles Underlying Apportionment Methods. *Annals of the New York Academy of Sciences*, 219, pp. 181-191.

[50] PLANE, D.A. (1982) Redistricting Refomulated: A Maximum Interaction/Minimum Separation Objective. *Socio-Economic Planning Sciences*, 16(6), pp. 241-244.

[51] POLSBY, D.D. AND POPPER R.D. (1991) The Third Criterion: Compactness as a Procedural Safeguard Against Partisan Gerrymandering. *Yale Law and Policy Review*, 9, pp. 301-353.

[52] REOCK, E.C.Jr. (1961) Measuring Compactness as a Requirement of Legislative Apportionment. *Midwest Journal of Political Science*, 5, pp. 70-74.

[53] ROCHAT, Y. AND E. TAILLARD (1995) Probabilistic Diversification and Intensification in Local Search for Vehicle Routing. *Journal of Heuristics*, 1(1), pp. 147-167.

[54] SCHUCK, P.H. (1990) Partisan Gerrymandering: A Political Problem without Judicial Solution. In *Political Gerrymandering and the Courts*, B. Grofman (Ed.), New York: Agathon Press.

[55] SCHWATRZBERG, J.E. (1966) Reapportionment, Gerrymanders, and the Notion of Compactness. Minnesota Law Review, 50, pp. 443-452.

[56] TAYLOR, P.J. (1973) A New Shape Measure for Evaluating Electoral District Patterns. *The American Political Science Review*, 67, pp. 947-950.

[57] THORESON, J.D. AND J.M. LIITTSCHWAGER (1967) Legislative Districting by Computer Simulation. *Behavioral Science*, 12, pp. 237-247.

[58] TOBLER, W.R. (1973) A Continuous Transformation Useful for Districting. *Annals of the New York Academy of Sciences*, 219, pp. 215-220.

[59] VICKREY, W. (1961) On the Prevention of Gerrymandering. *Political Science Quarterly*, 76, pp. 105-110.

[60] WEAVER, J.B. AND S.W. HESS (1963) A Procedure for Nonpartisan Districting: Development of Computer Techniques. *The Yale Law Journal*, 72, pp. 288-308.

[61] WELLS, D.I. (1982) How to Inhibit Gerrymandering: The Purpose of Districting Criteria. *National Civic Review*, 71, pp. 183-187.

[62] WILLIAMS, J.C.Jr. (1995) Political Redistricting: A Review. *Papers in Regional Science*, 74(1), pp. 13-40.

[63] YOUNG, H.P. (1988) Measuring the Compactness of Legislative Districts. *Legislative Studies Quarterly*, 13(1), pp. 105-115.

Appendix A

Enumeration Area Maps and Existing Electoral Districts

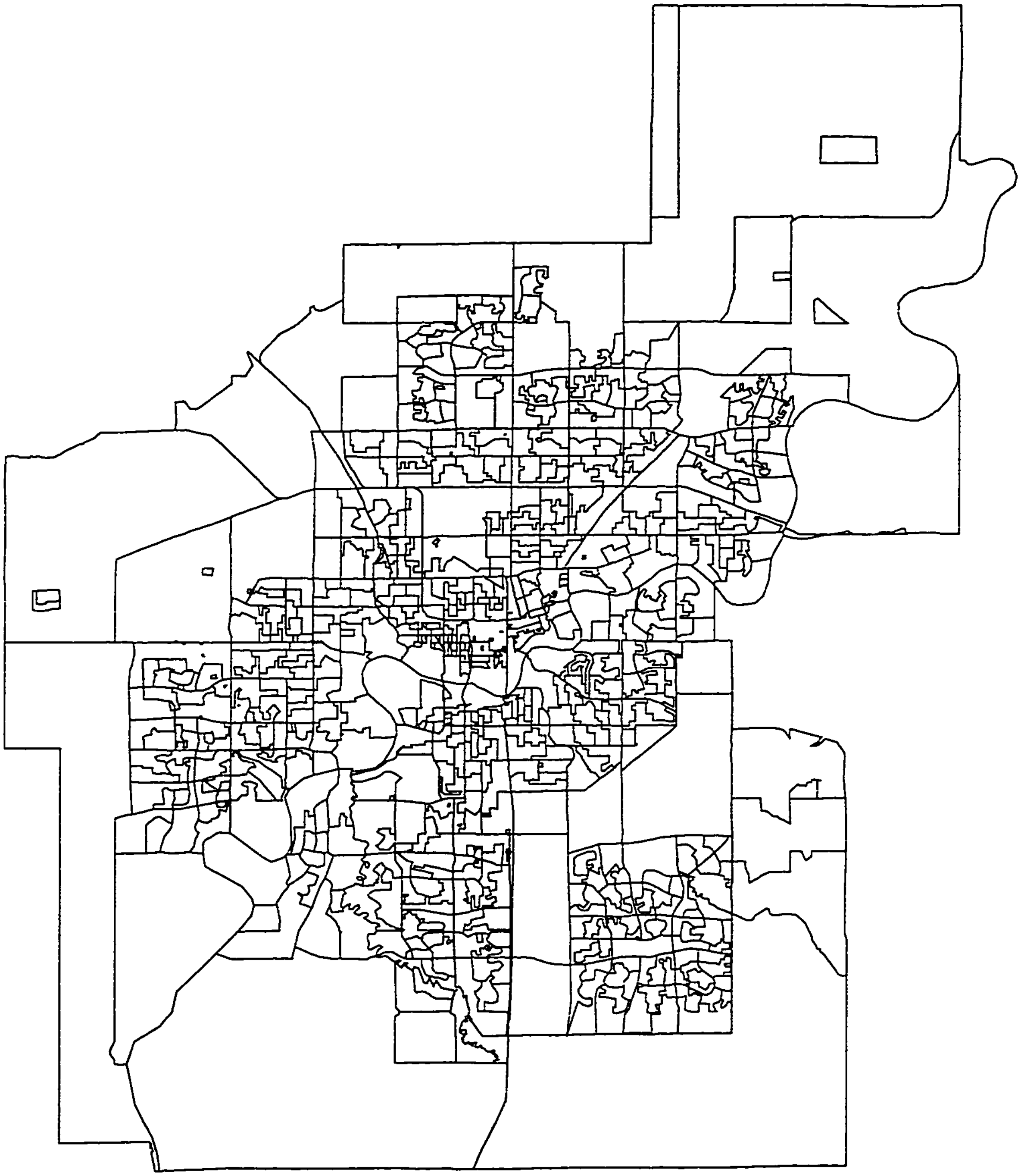

Figure A.1: Enumeration areas for the City of Edmonton

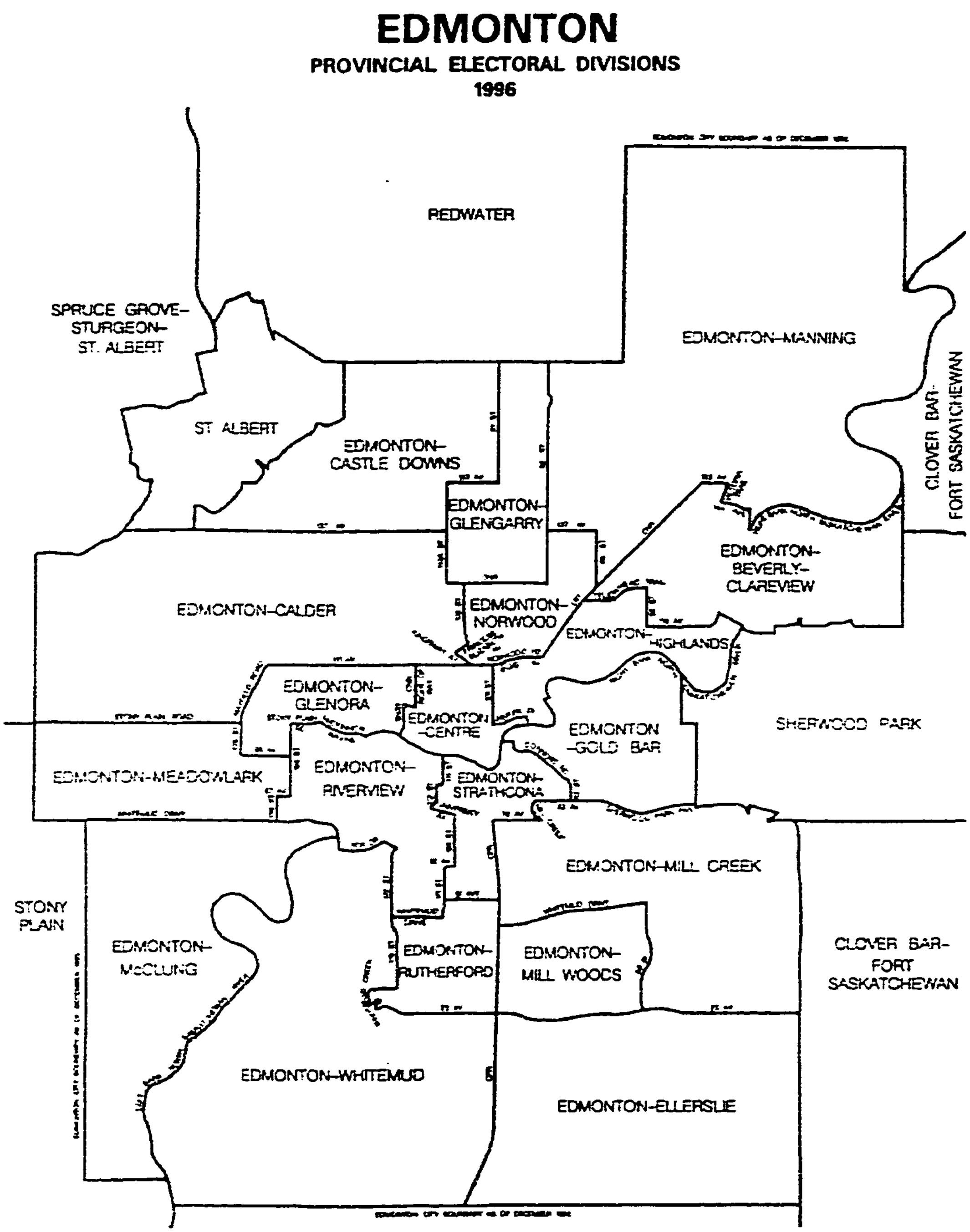

Figure A.2: Official Electoral Districts for the City of Edmonton

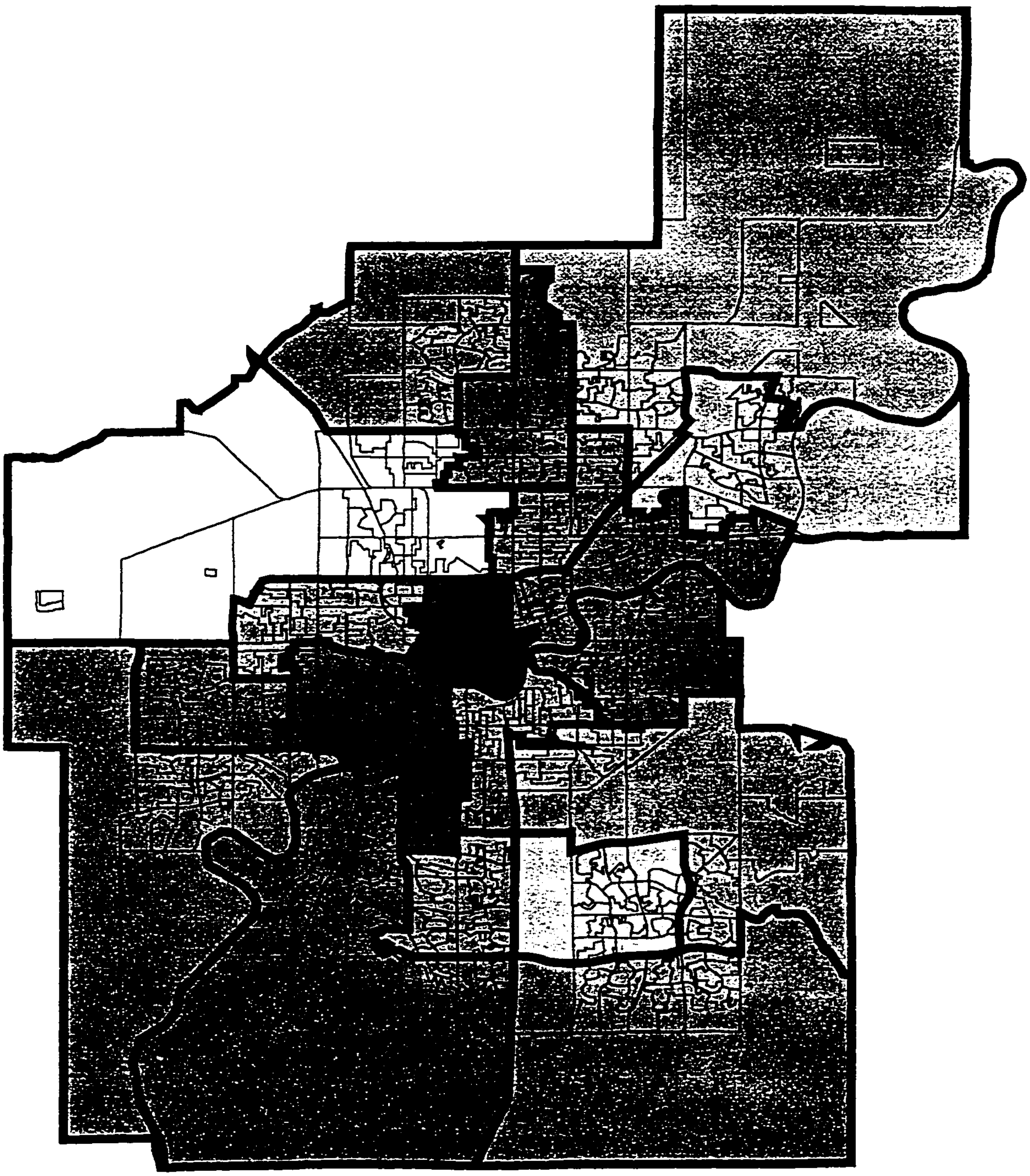

Figure A.3: Approximated Electoral Districts for the City of Edmonton

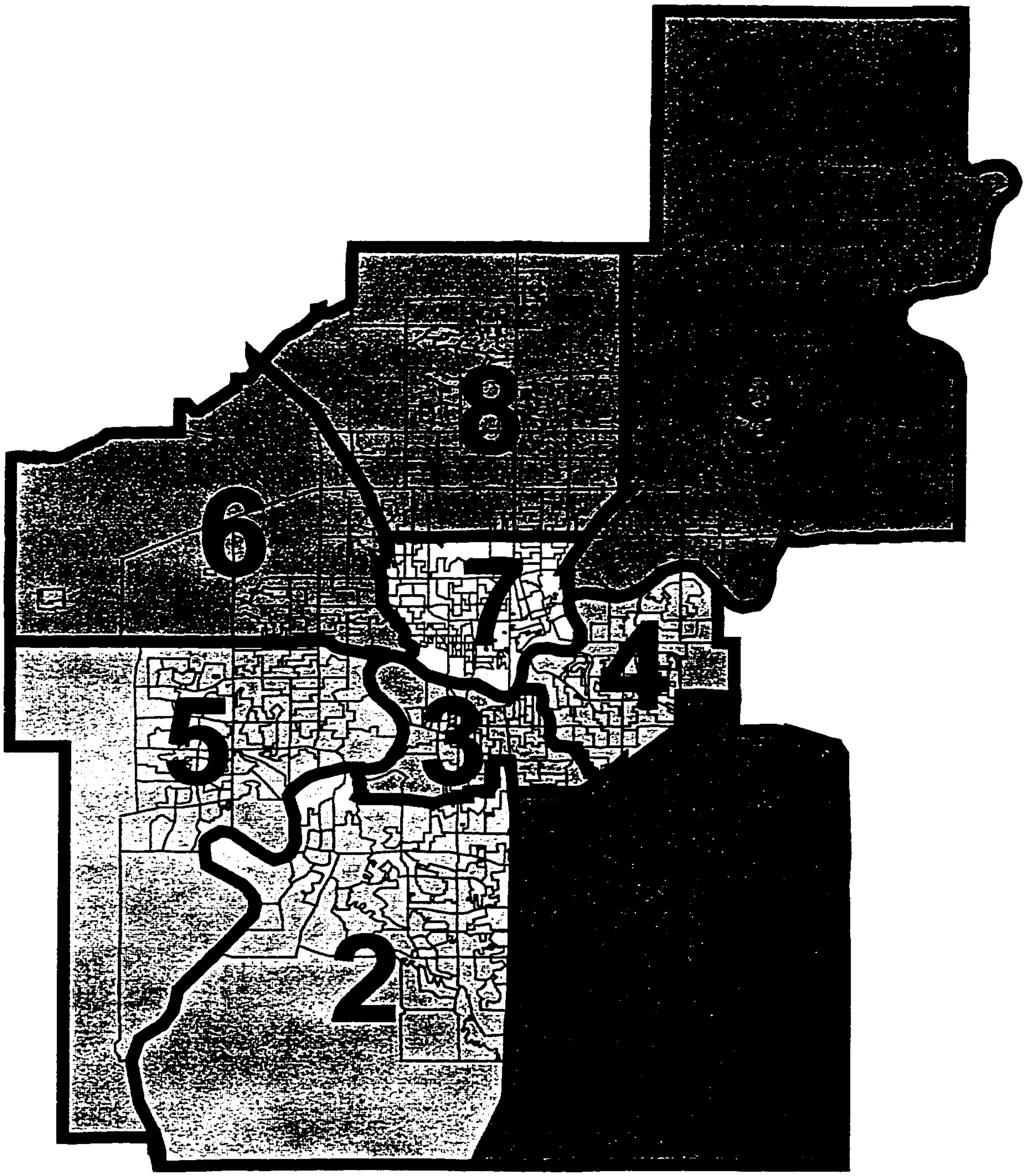

Figure A.4: A General Community Plan for the City of Edmonton

Ideal average:		32437			
Allowable range:		[24327,40546]			

District	**Population**	**% Dev.**	**District**	**Population**	**% Dev.**
Ellerslie	32090	-1.1%	Stratchona	30226	-6.8%
Mill Woods	30630	-5.6%	Riverview	30384	-6.3%
Rutherford	32927	1.5%	Glenora	30885	-4.8%
Whitemud	37435	15.4%	Calder	31238	-3.7%
McClung	34976	7.8%	Glengarry	30398	-6.3%
Meadowlark	31443	-3.1%	Castle Downs	33991	4.8%
Mill Creek	37143	14.5%	Norwood	28750	-11.4%
Gold Bar	31611	-2.5%	Manning	38022	17.2%
Highlands	31770	-2.1%	Beverly-Clareview	31850	-1.8%
Centre	30537	-5.9%	**TOTAL**	**616306**	
Compactness 1:	1.385				
Compactness 2:	0.301				

Table A.1: 1996 District Populations for Edmonton's Existing Plan

Appendix B

Best Plans with Fine-Tuning

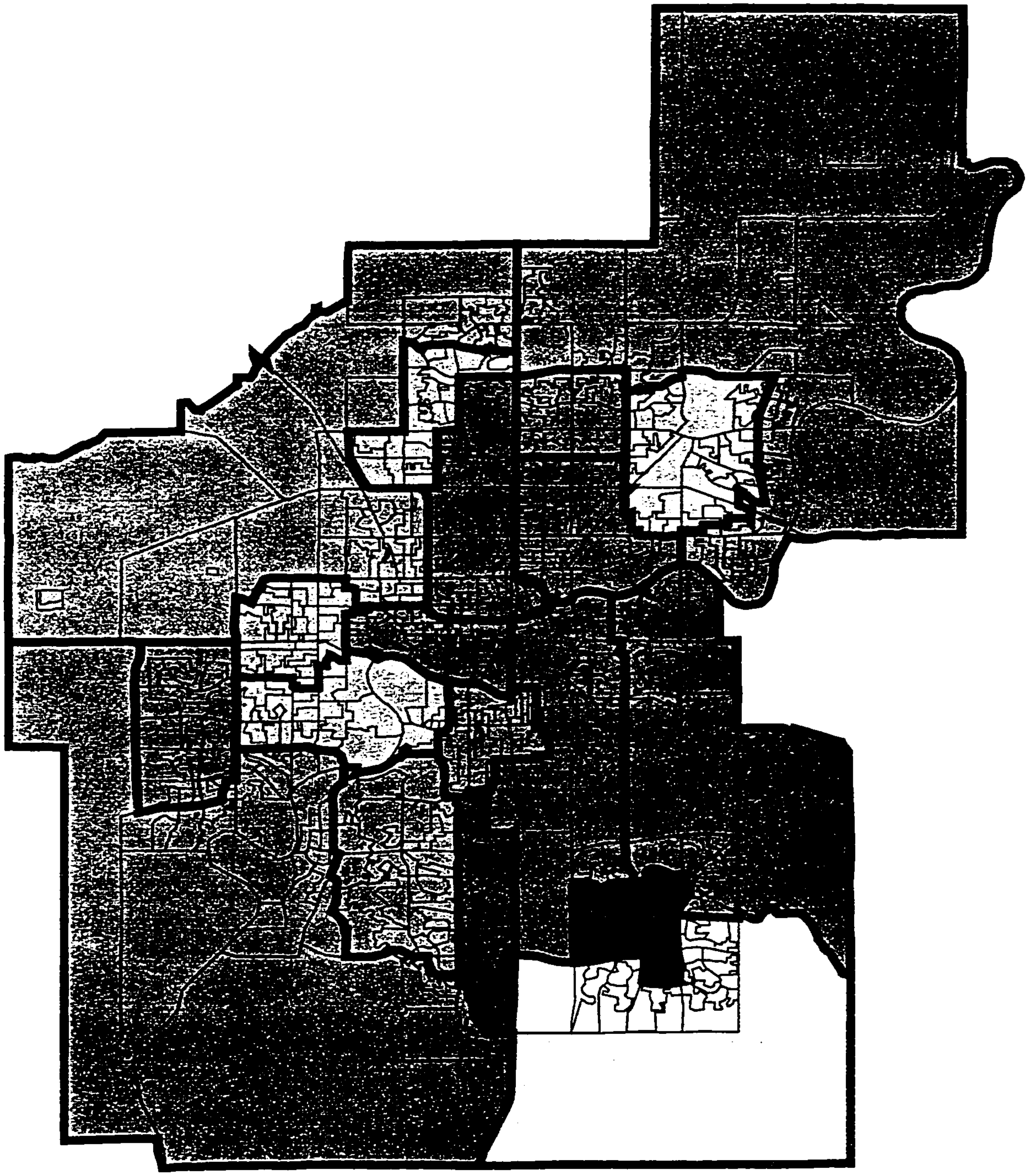

Figure B.1: District plan for (E/FT-1/25%/1)

Ideal average:		32437			
Allowable range:		[24327,40546]			

District	**Population**	**% Dev.**	**District**	**Population**	**% Dev.**
1	40455	24.7%	11	24969	-23.0%
2	38797	19.6%	12	25501	-21.4%
3	39833	22.8%	13	25346	-21.9%
4	40402	24.6%	14	36545	12.7%
5	37633	16.0%	15	39307	21.2%
6	30088	-7.2%	16	34097	5.1%
7	24882	-23.3%	17	36626	12.9%
8	39607	22.1%	18	25563	-21.2%
9	24743	-23.7%	19	24982	-23.0%
10	26930	-17.0%			
Compactness 1:		1.385			
Compactness 2:		0.301			

Table B.1: District populations for (E/FT-1/25%/1)

Ideal average:		32437			
Allowable range:		[24327,40546]			

District	**Population**	**% Dev.**	**District**	**Population**	**% Dev.**
1	32933	1.5%	11	24969	-23.0%
2	36693	13.1%	12	25549	-21.2%
3	27409	-15.5%	13	35187	8.5%
4	27483	-15.3%	14	26242	-19.1%
5	39599	22.1%	15	40420	24.6%
6	24460	-24.6%	16	31223	-3.7%
7	35829	10.5%	17	39915	23.1%
8	35931	10.8%	18	25839	-20.3%
9	40070	23.5%	19	27187	-16.2%
10	39368	21.4%			
Compactness 1:		1.445			
Compactness 2:		0.287			

Table B.2: District populations for (E/FT-1/25%/2)

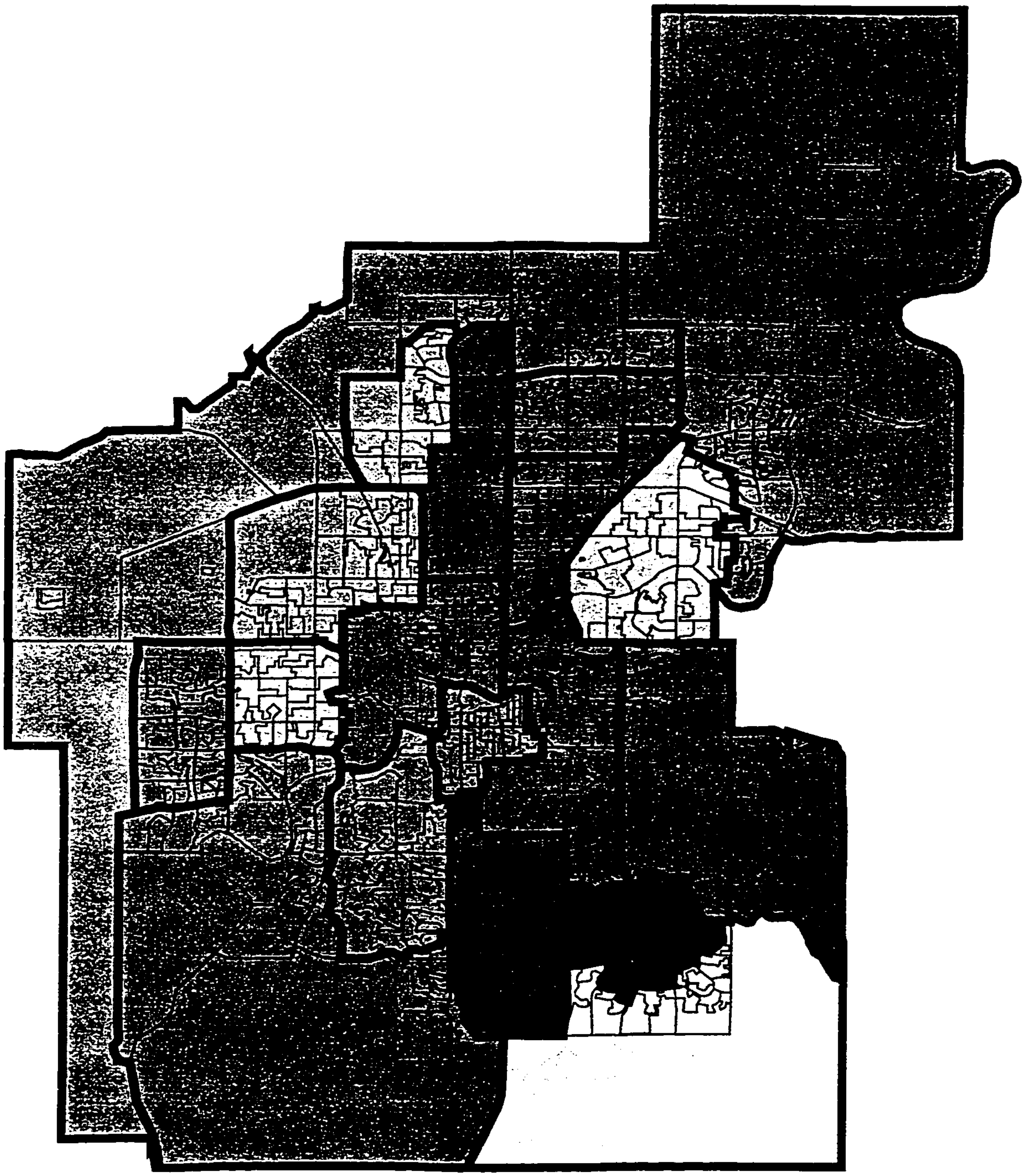

Figure B.2: District plan for (E/FT-1/25%/2)

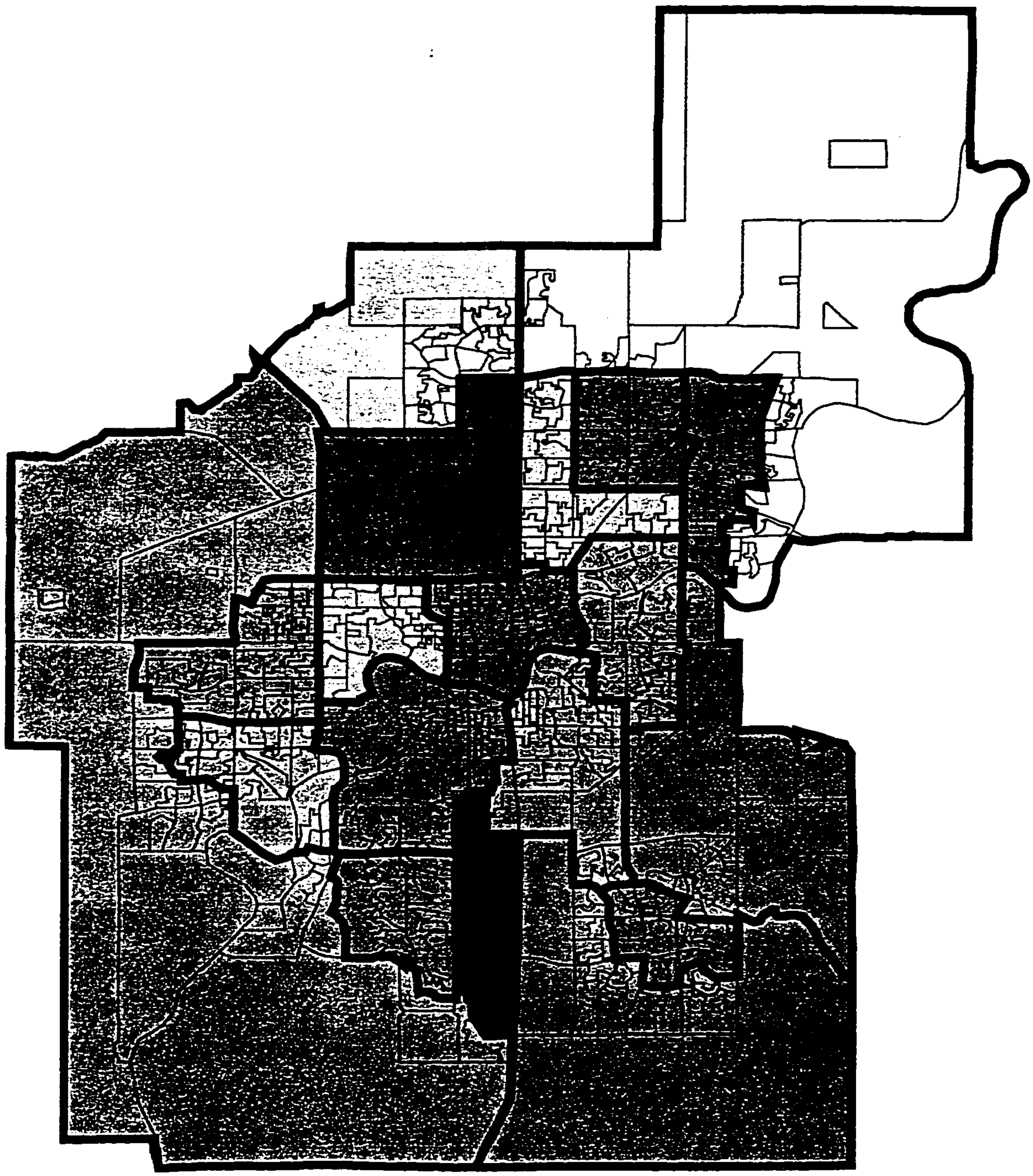

Figure B.3: District plan for (E/FT-2/25%/1)

Ideal average:		32437			
Allowable range:		[24327,40546]			

District	Population	% Dev.	District	Population	% Dev.
1	40048	23.5%	11	39488	21.7%
2	33991	4.8%	12	40386	24.5%
3	34942	7.7%	13	25348	-21.9%
4	36370	12.1%	14	35238	8.6%
5	34467	6.3%	15	24642	-24.0%
6	26981	-16.8%	16	27973	-13.8%
7	32242	-0.6%	17	25191	-22.3%
8	31247	-3.7%	18	25626	-21.0%
9	36954	13.9%	19	39640	22.2%
10	25532	-21.3%			

Compactness 1:		1.379			
Compactness 2:		0.298			

Table B.3: District populations for (E/FT-2/25%/1)

Ideal average:		32437			
Allowable range:		[24327,40546]			

District	Population	% Dev.	District	Population	% Dev.
1	26865	-17.2%	11	34454	6.2%
2	25069	-22.7%	12	33186	2.3%
3	30633	-5.6%	13	39458	21.6%
4	30468	-6.1%	14	31159	-3.9%
5	39391	21.4%	15	34715	7.0%
6	27483	-15.3%	16	40387	24.5%
7	35919	10.7%	17	39994	23.3%
8	35829	10.5%	18	27672	-14.7%
9	28365	-12.6%	19	28439	-12.3%
10	26820	-17.3%			

Compactness 1:		1.390			
Compactness 2:		0.269			

Table B.4: District populations for (E/FT-2/25%/2)

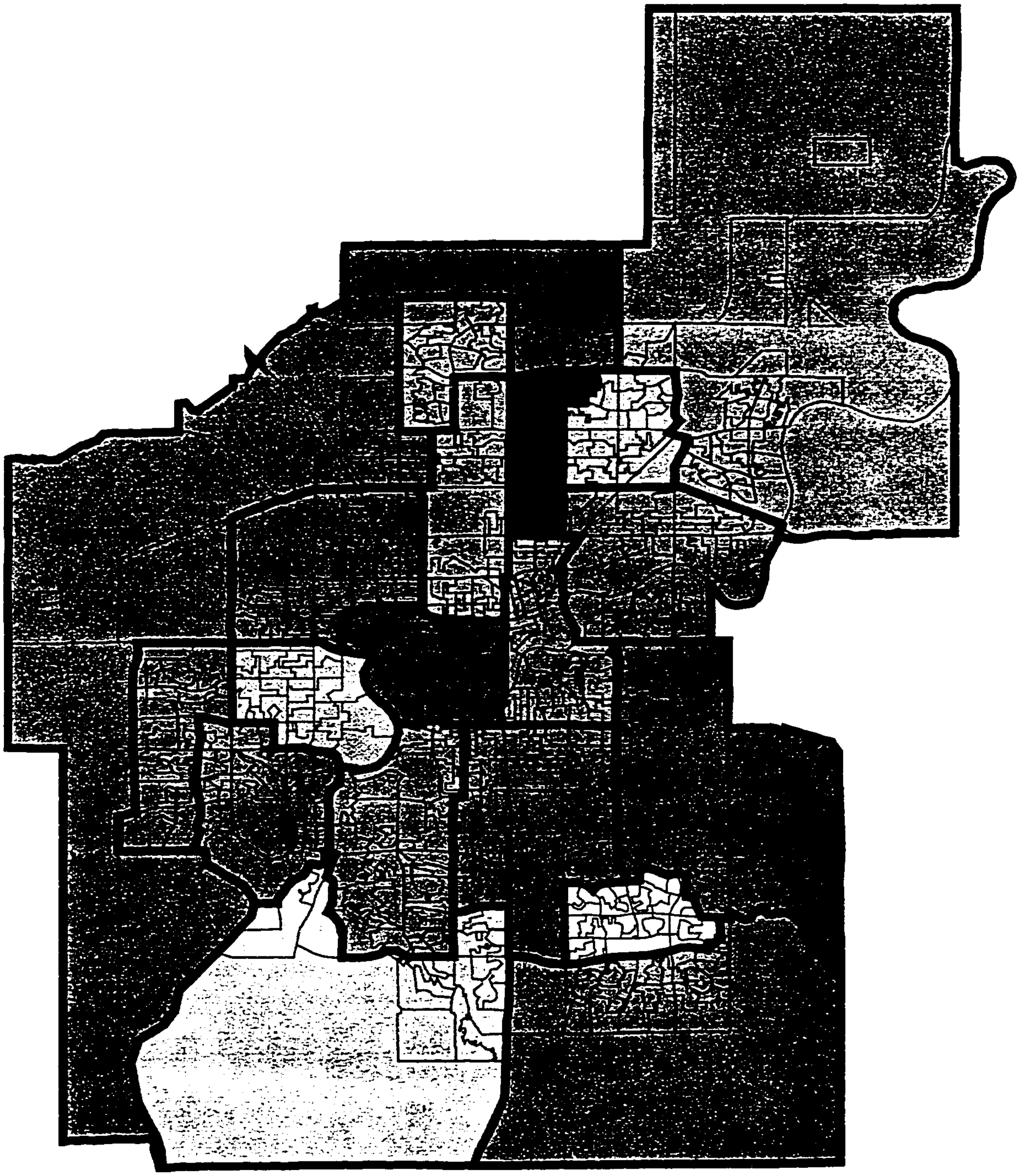

Figure B.4: District plan for (E/FT-2/25%/2)

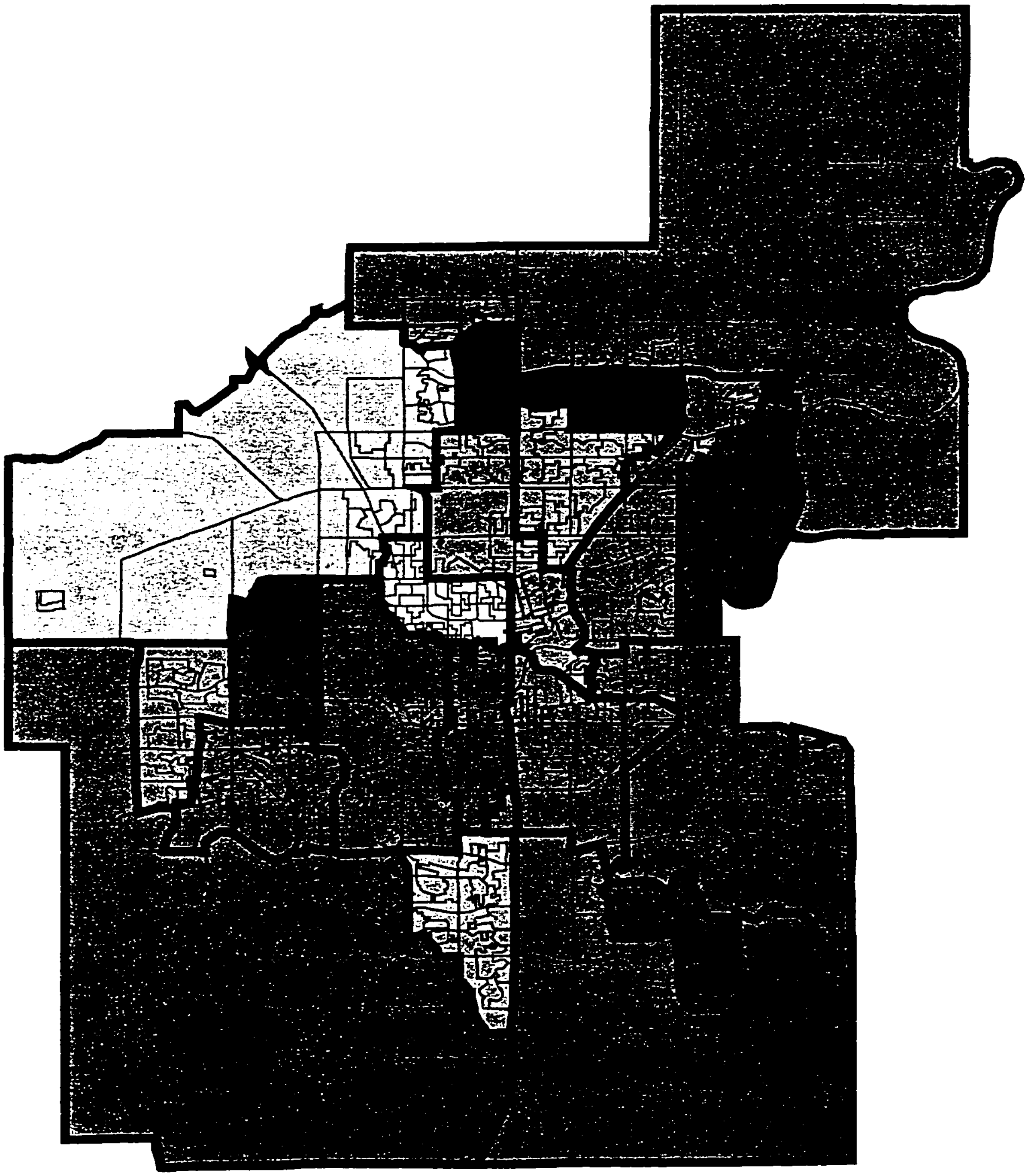

Figure B.5: District plan for (E/FT-1/10%/1)

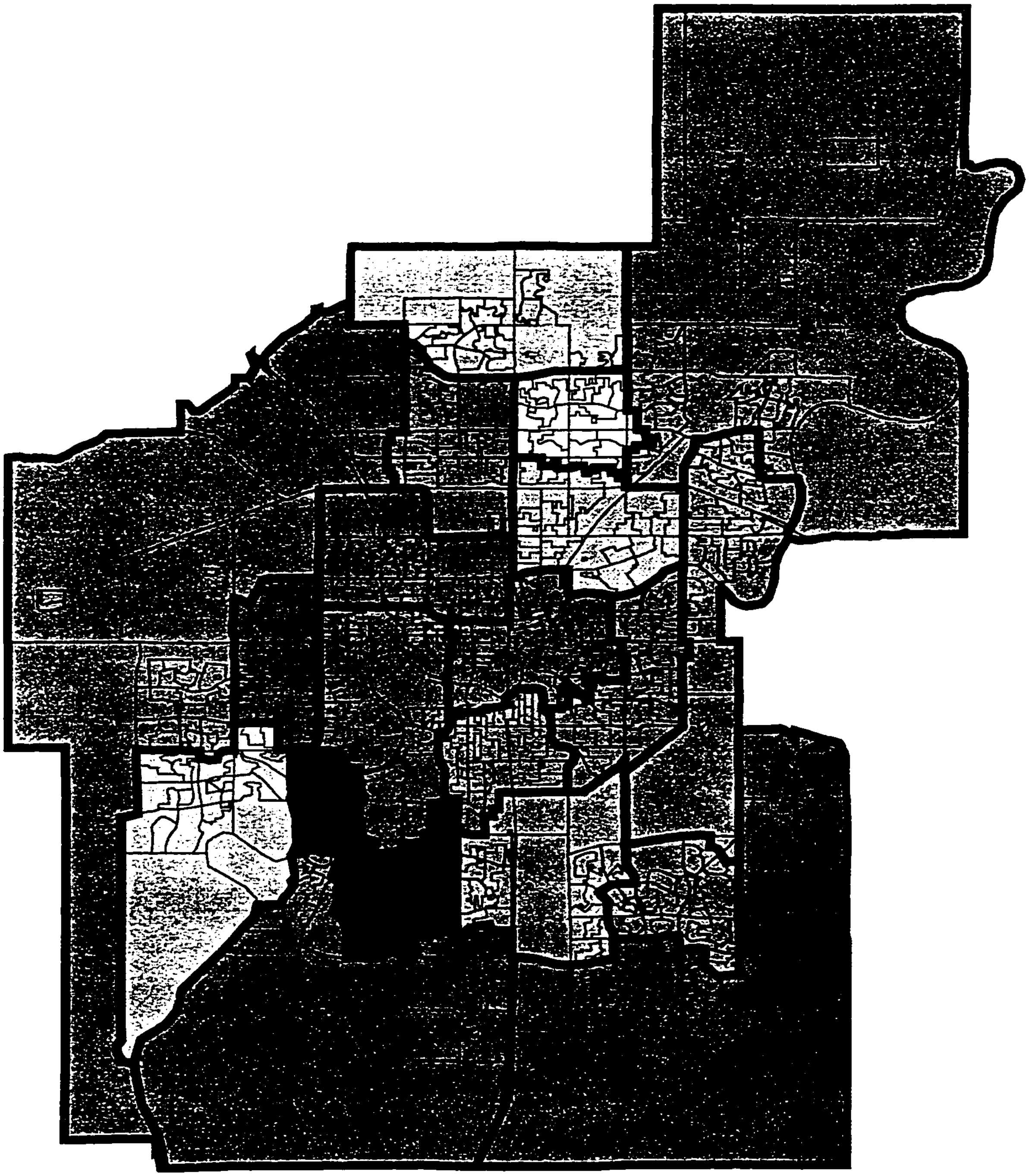

Figure B.6: District plan for (E/FT-1/10%/2)

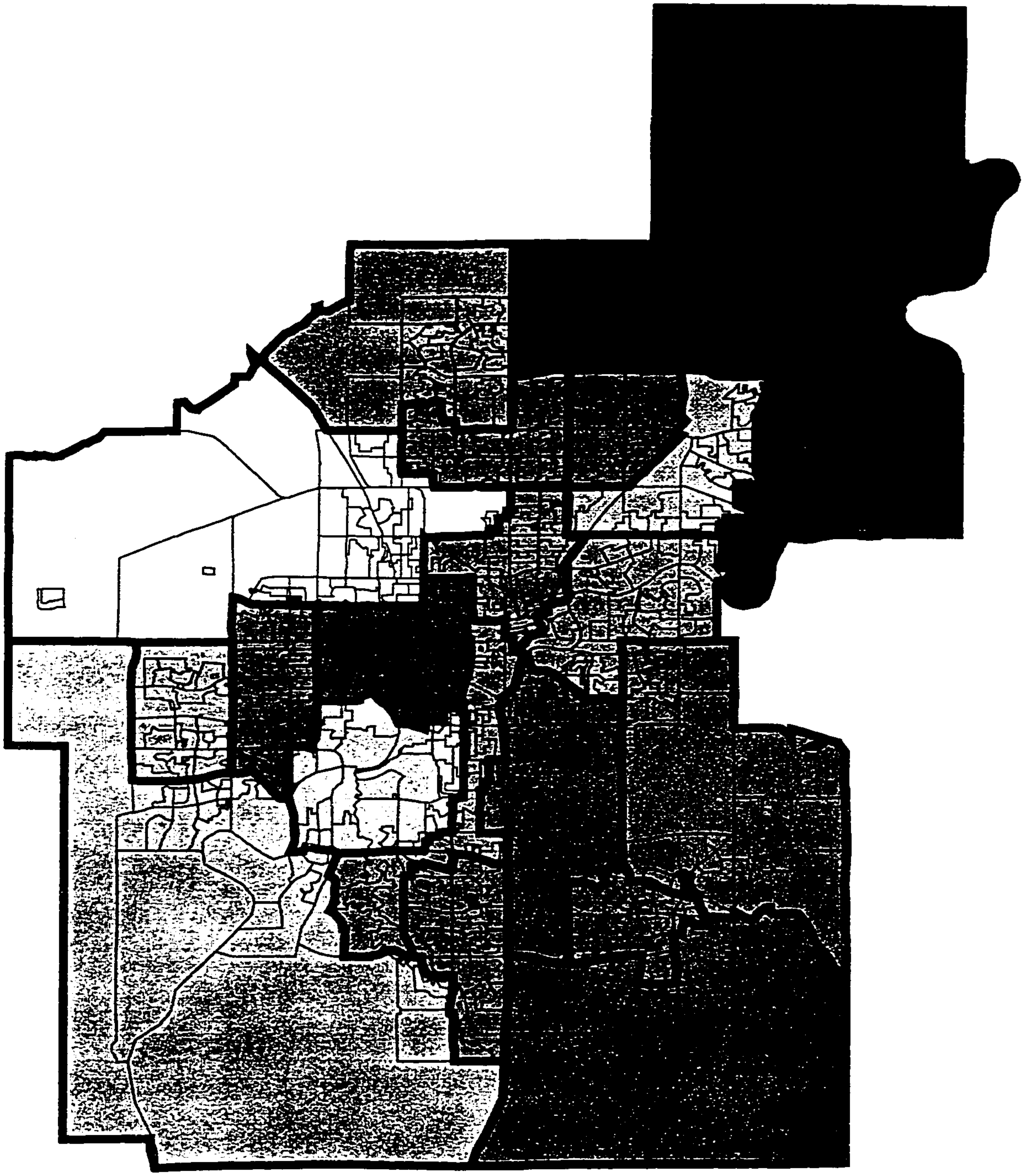

Figure B.7: District plan for (E/FT-2/10%/1)

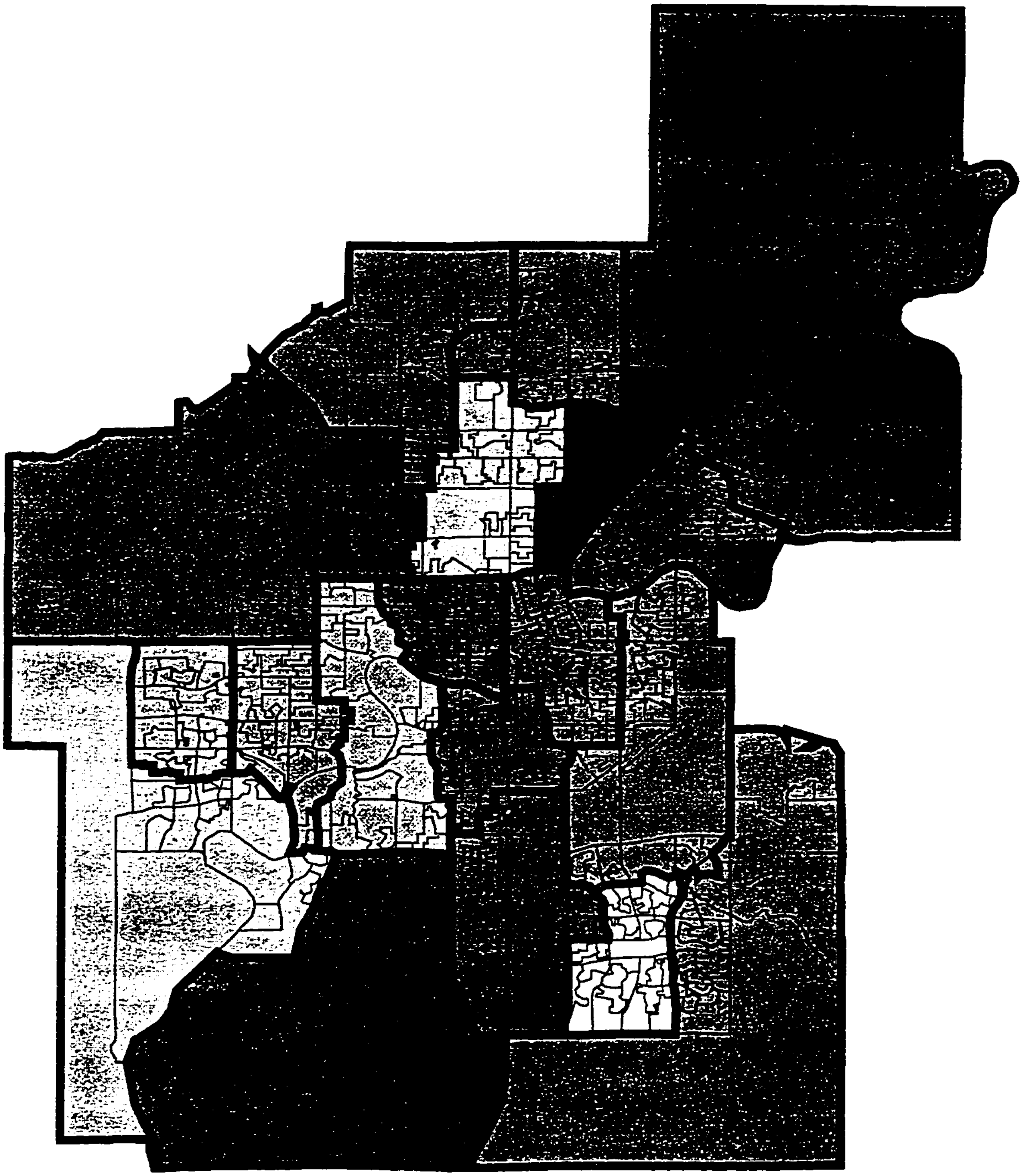

Figure B.8: District plan for (E/FT-2/10%/2)

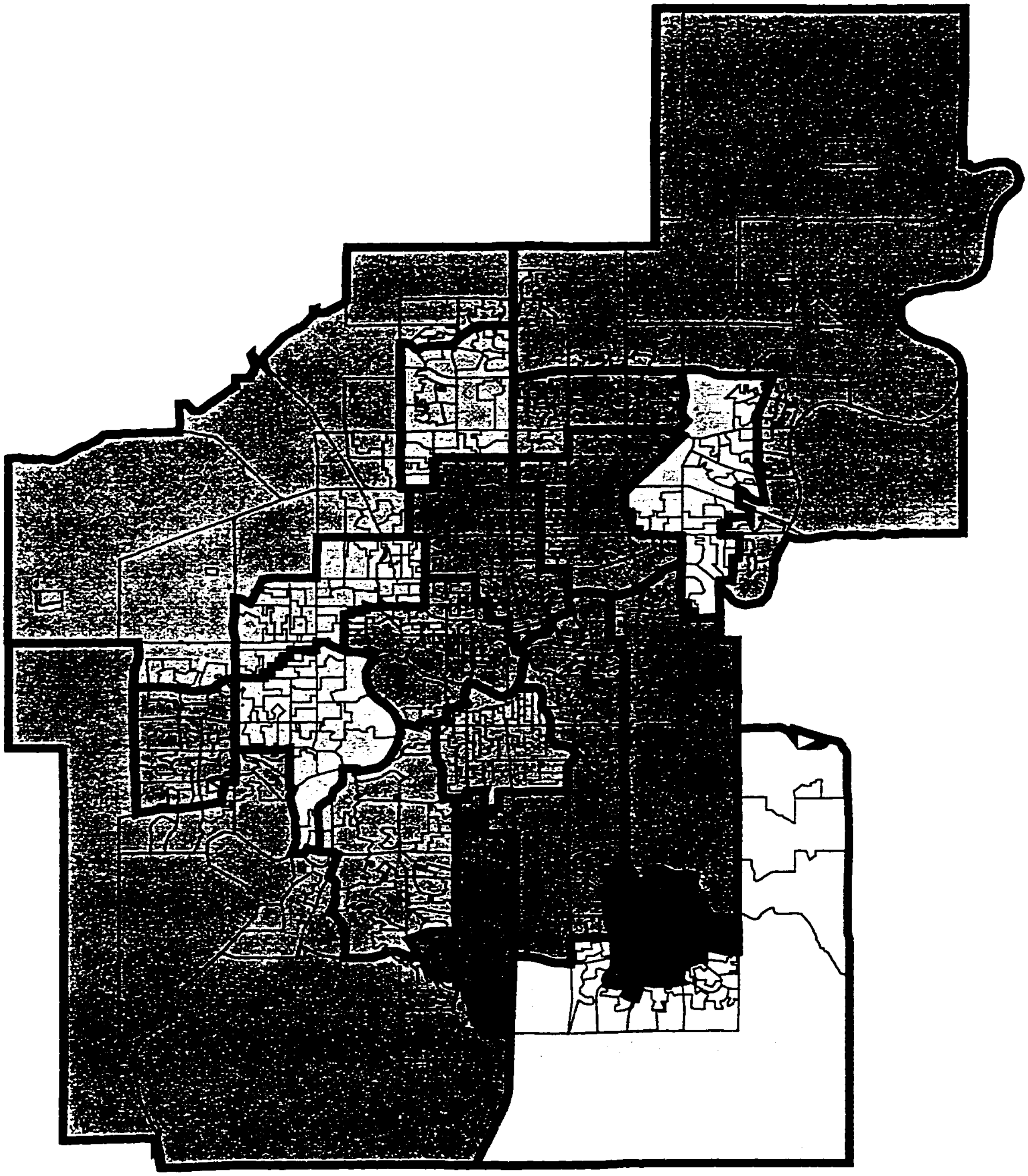

Figure B.9: District plan for (E/FT-1/5%/1)

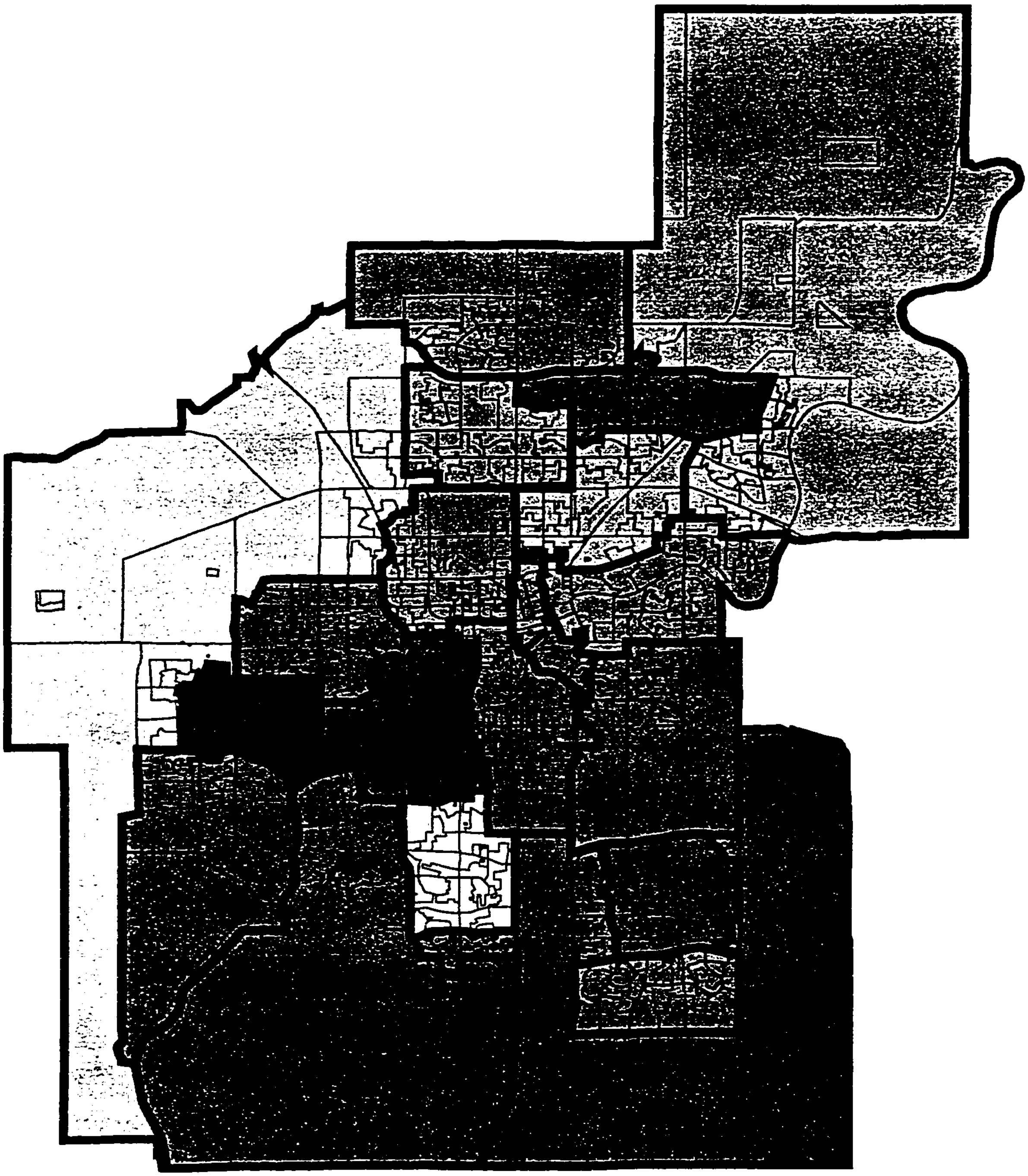

Figure B.10: District plan for (E/FT-1/5%/2)

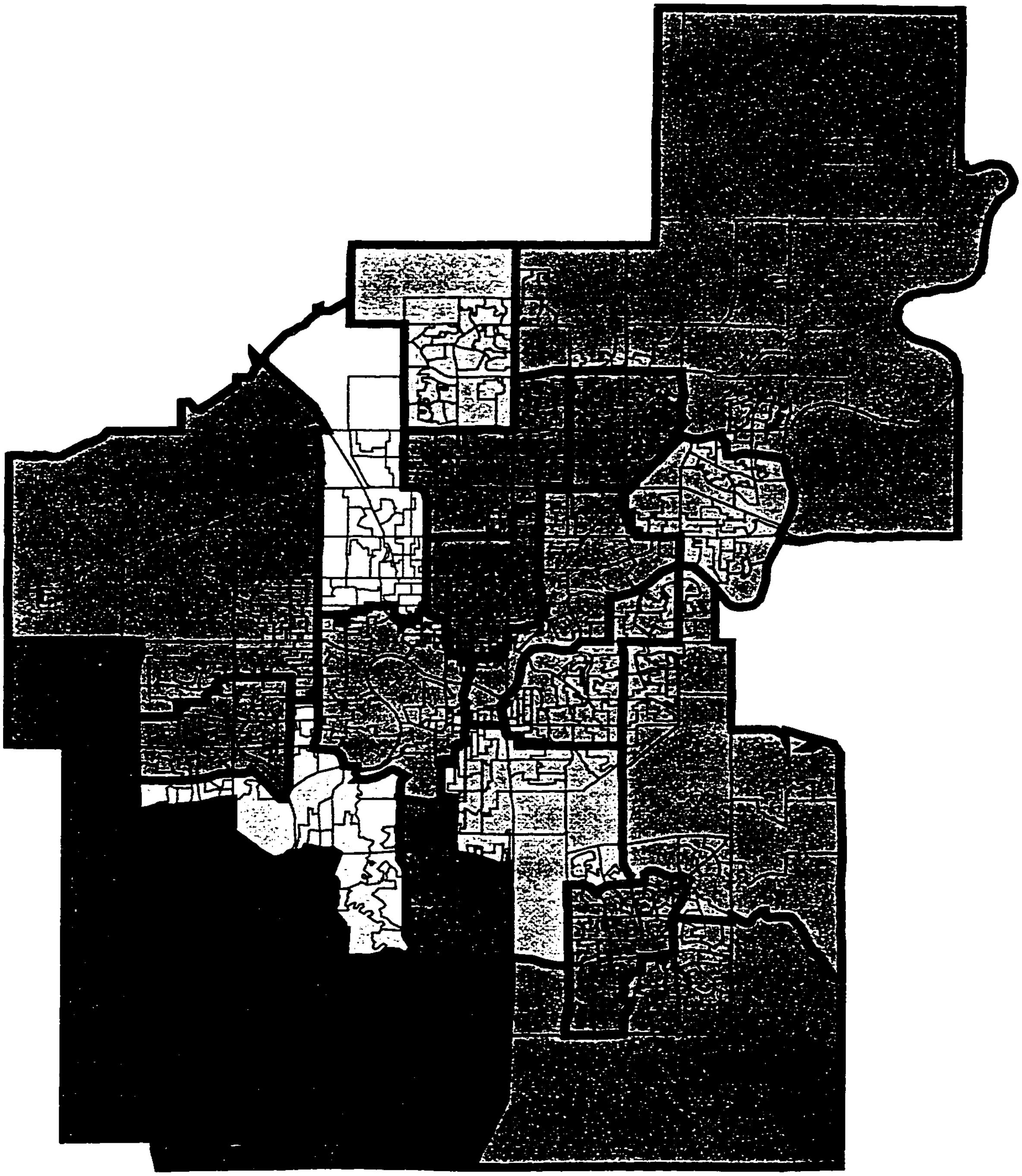

Figure B.11: District plan for (E/FT-2/5%/1)

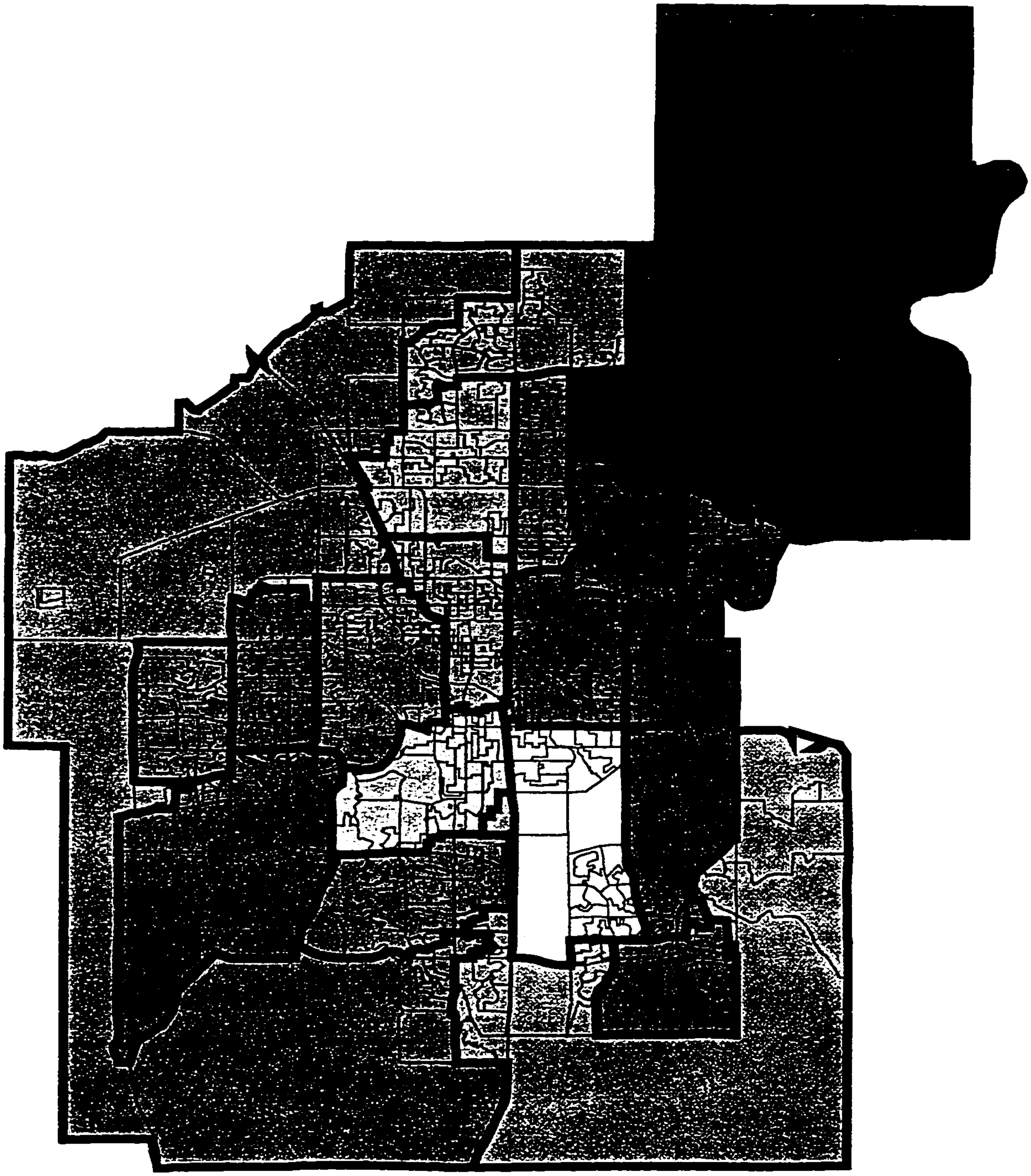

Figure B.12: District plan for (E/FT-2/5%/2)

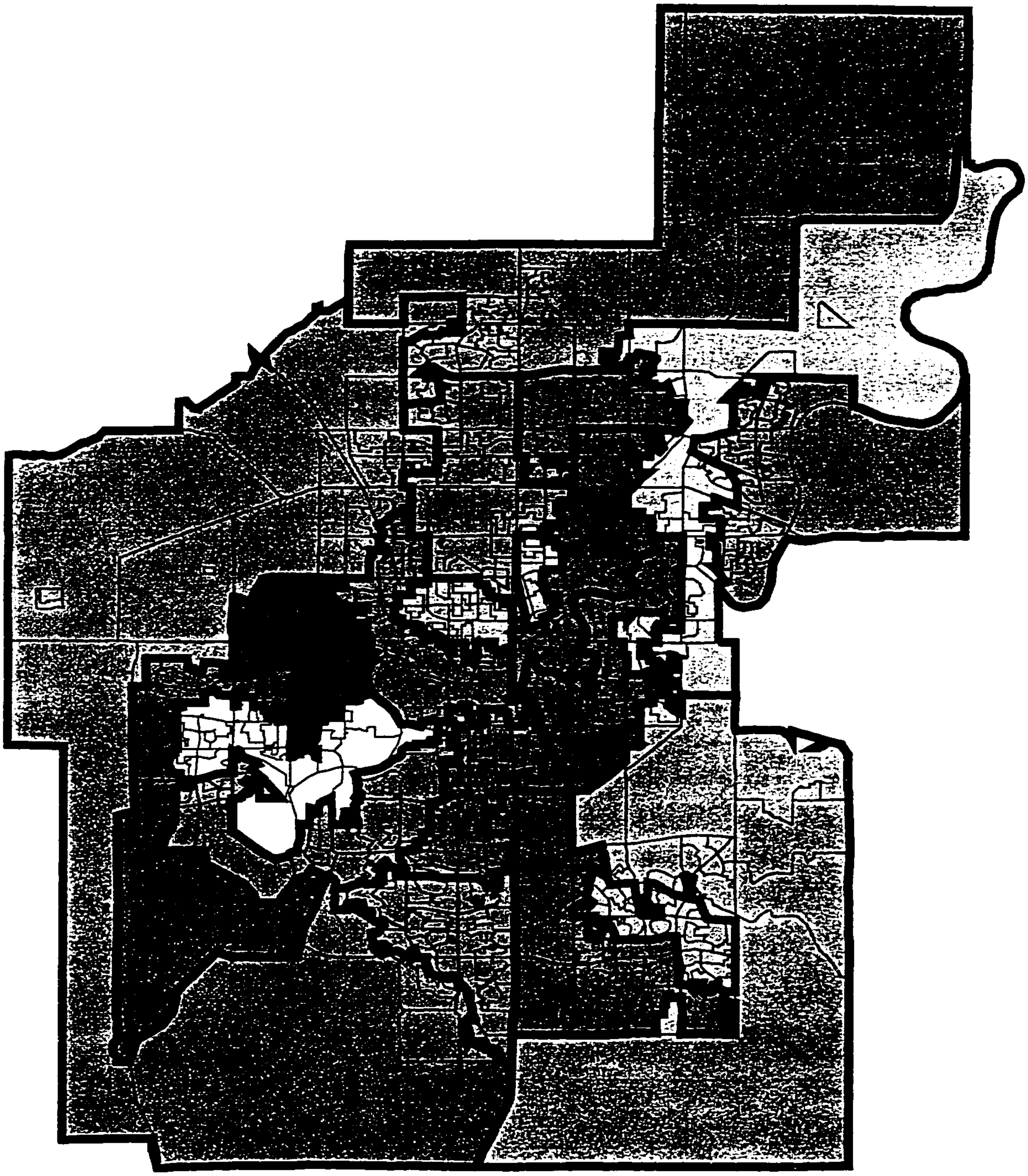

Figure B.13: A 1% district plan for Edmonton

Ideal average:		32437			
Allowable range:		[32113,32761]			
District	**Population**	**% Dev.**	**District**	**Population**	**% Dev.**
1	32537	0.3%	11	32433	0.0%
2	32695	0.8%	12	32195	-0.7%
3	32267	-0.5%	13	32124	-1.0%
4	32598	0.5%	14	32722	0.9%
5	32725	0.9%	15	32136	-0.9%
6	32366	-0.2%	16	32150	-0.9%
7	32433	0.0%	17	32409	-0.1%
8	32482	0.1%	18	32341	-0.3%
9	32631	0.6%	19	32544	0.3%
10	32518	0.2%			
Compactness 1:		2.583			
Compactness 2:		0.570			

Table B.5: District populations for the 1% plan

Appendix C

District Plans under Scenario 2

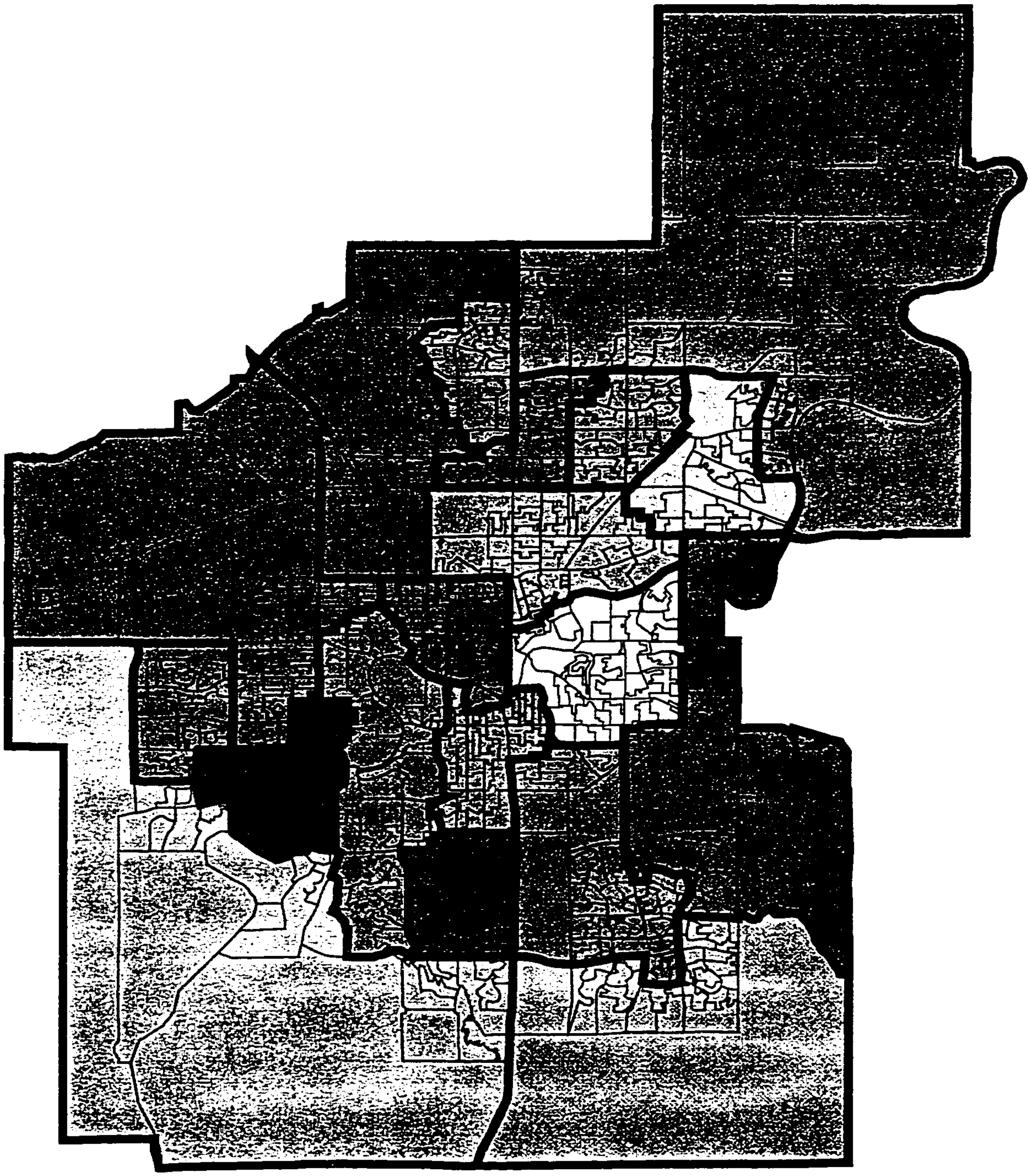

Figure C.1: District plan for (E/S-2/25%/1)

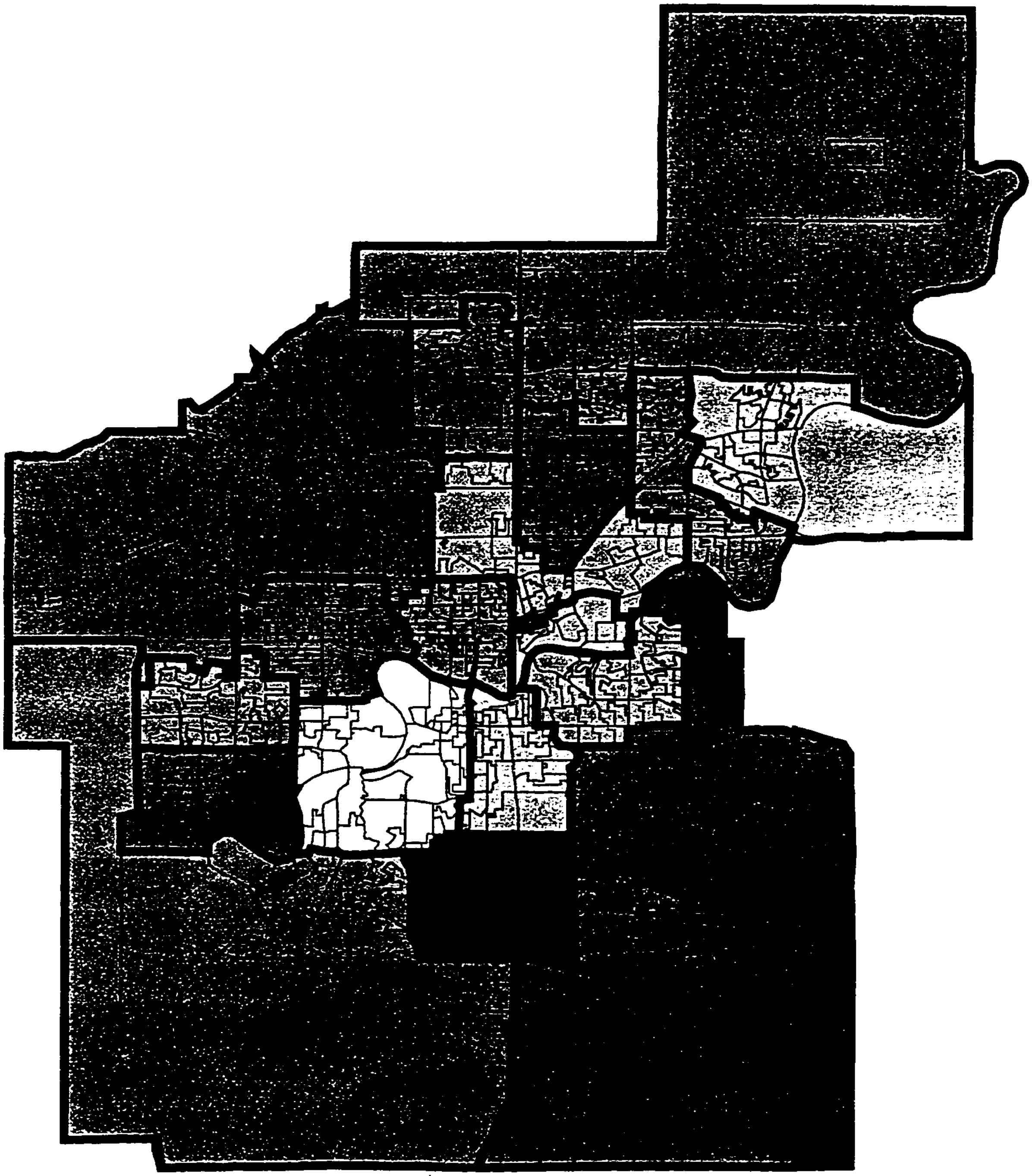

Figure C.2: District plan for (E/S-2/10%/1)

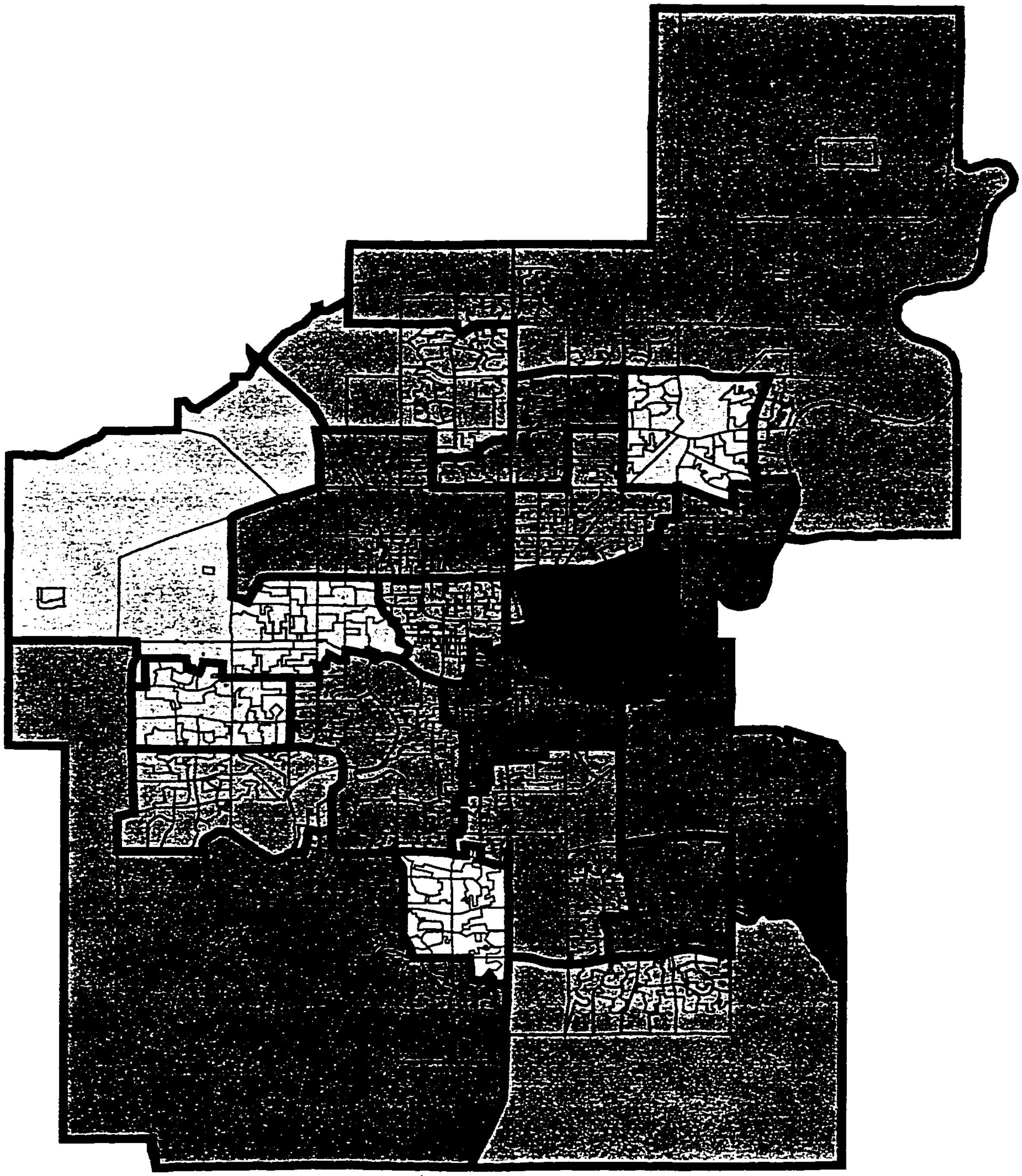

Figure C.3: District plan for (E/S-2/5%/1)

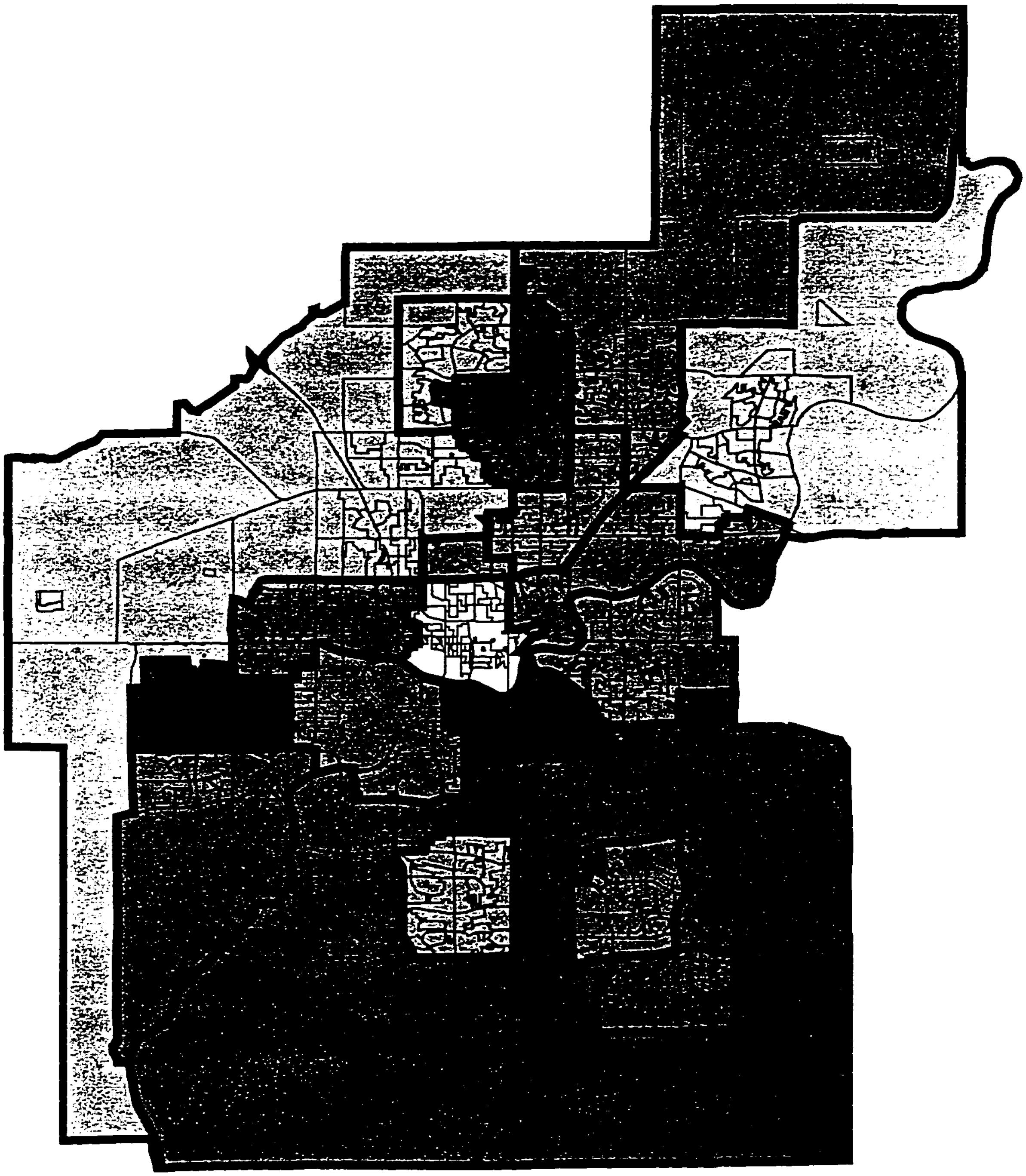

Figure C.4: District plan for (E/S-2/25%/2)

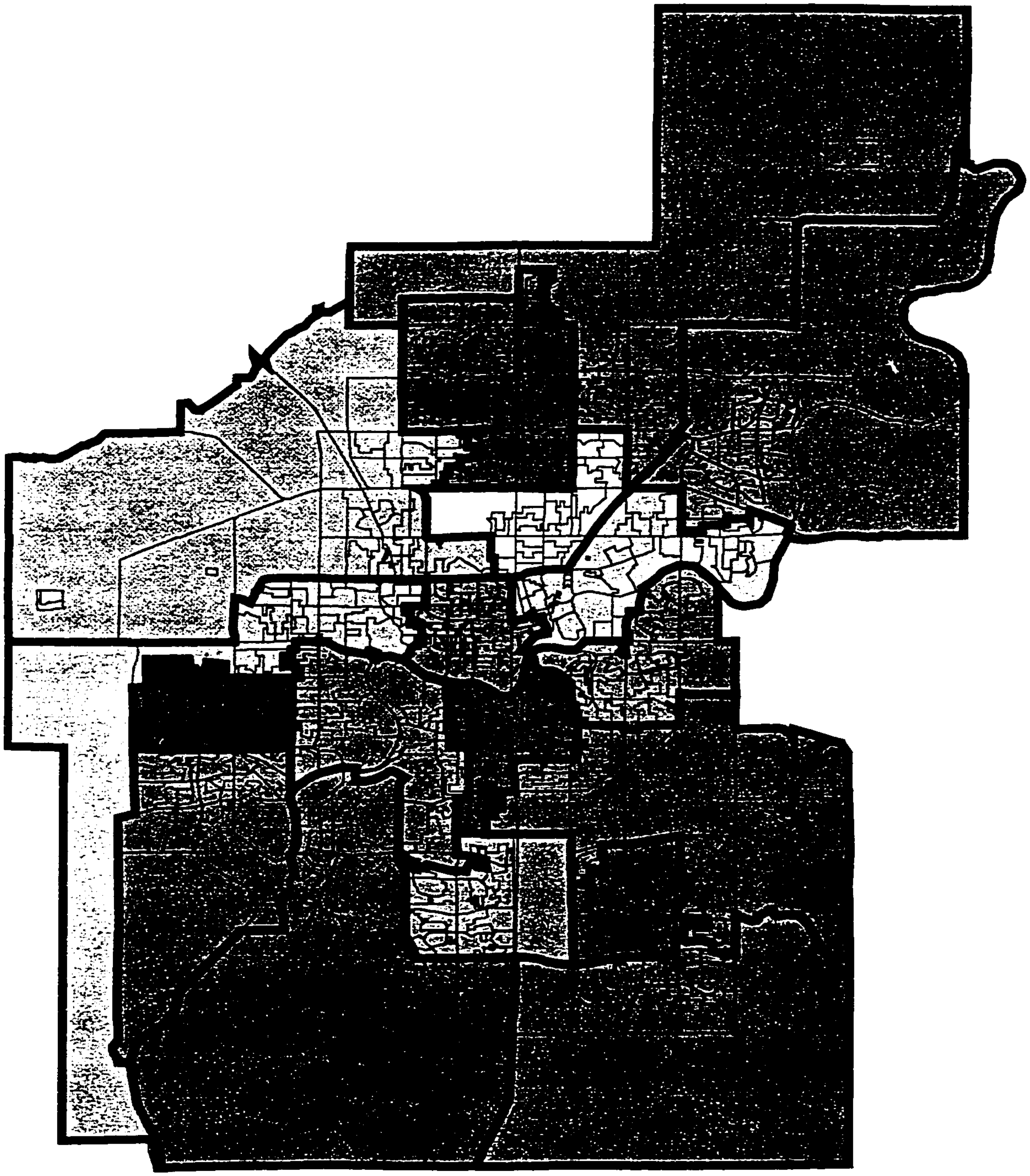

Figure C.5: District plan for (E/S-2/10%/2)

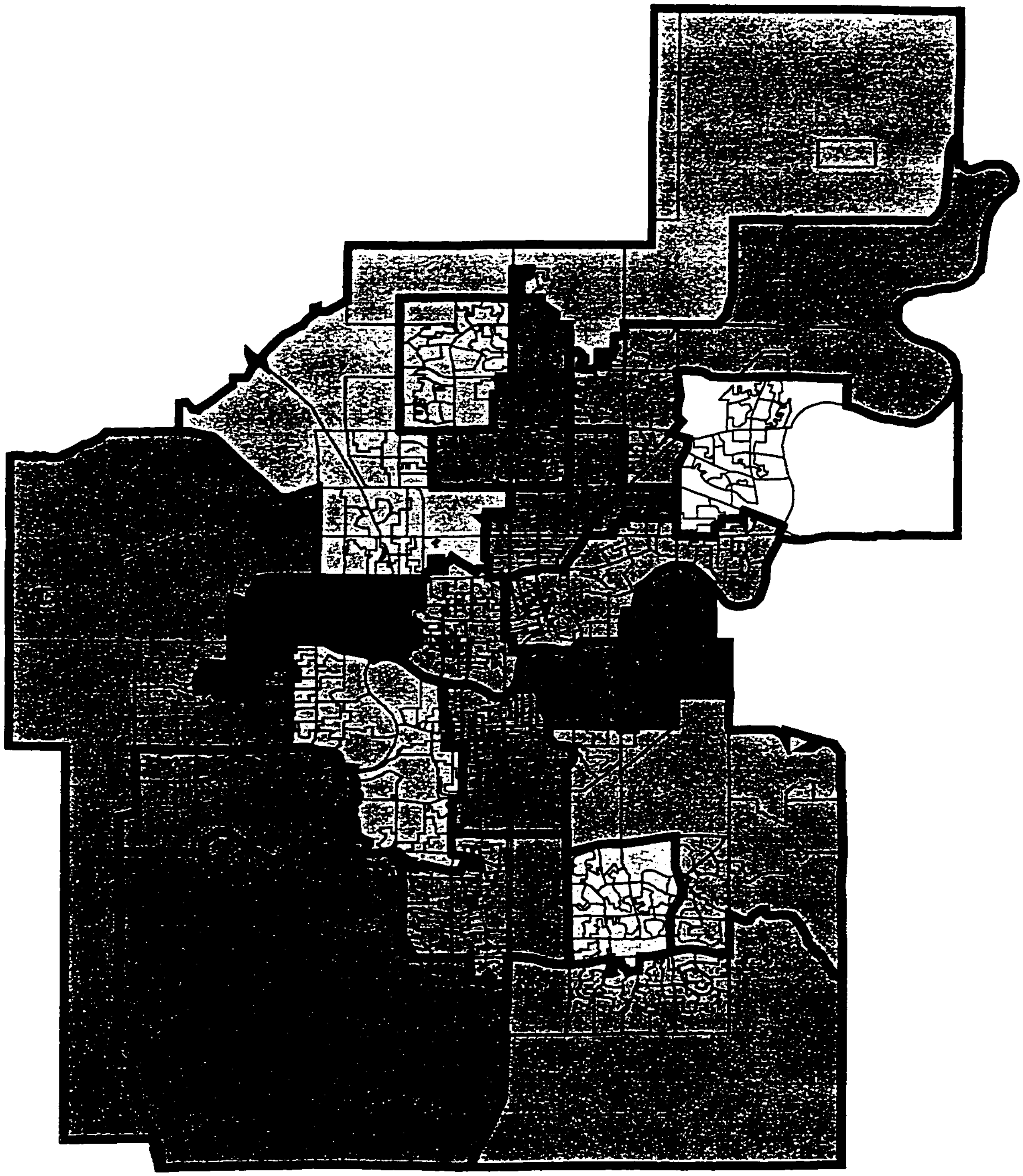

Figure C.6: District plan for (E/S-2/5%/2)

Appendix D

District Plans under Scenario 3

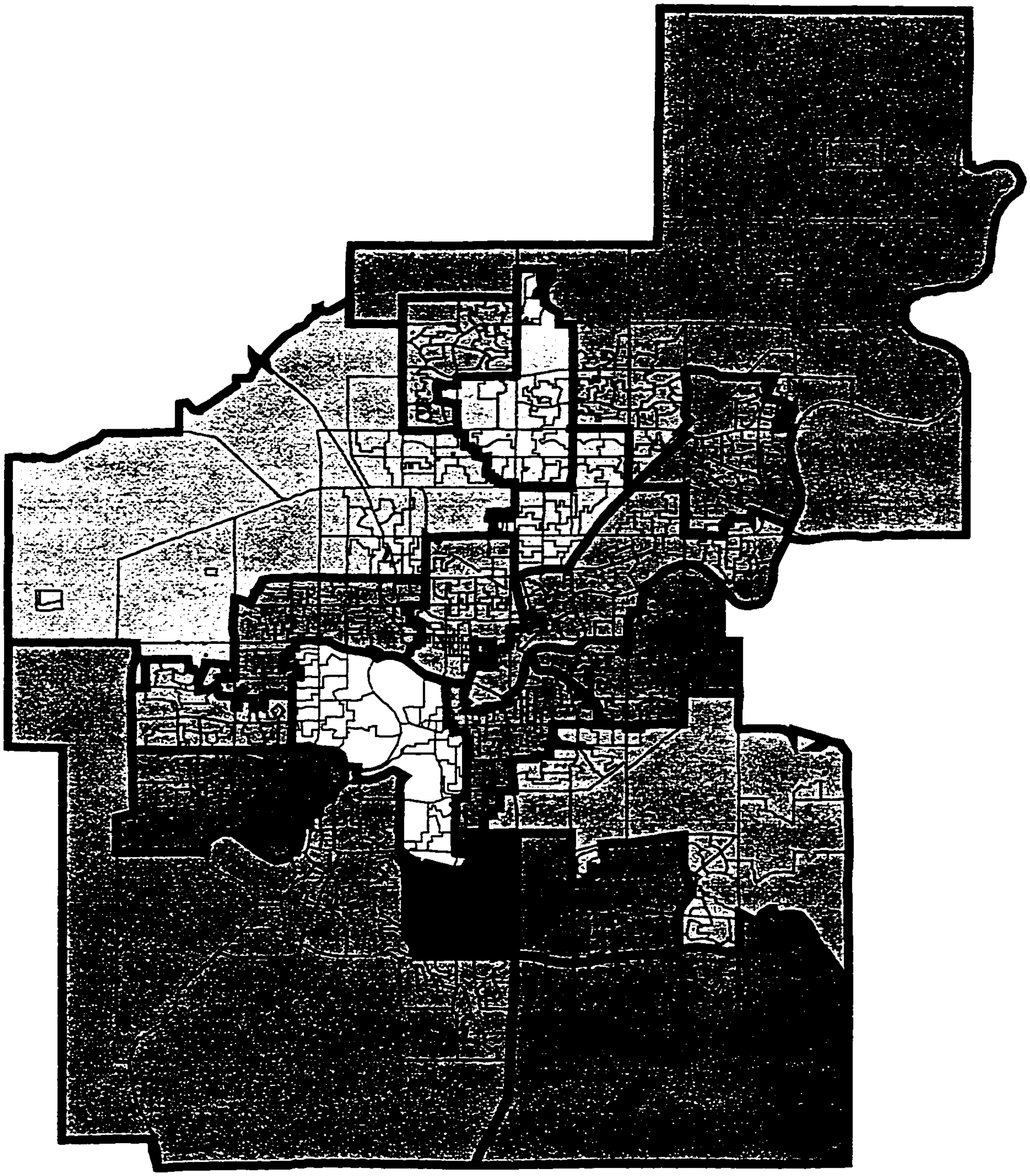

Figure D.1: District plan for (E/S-3/25%/1)

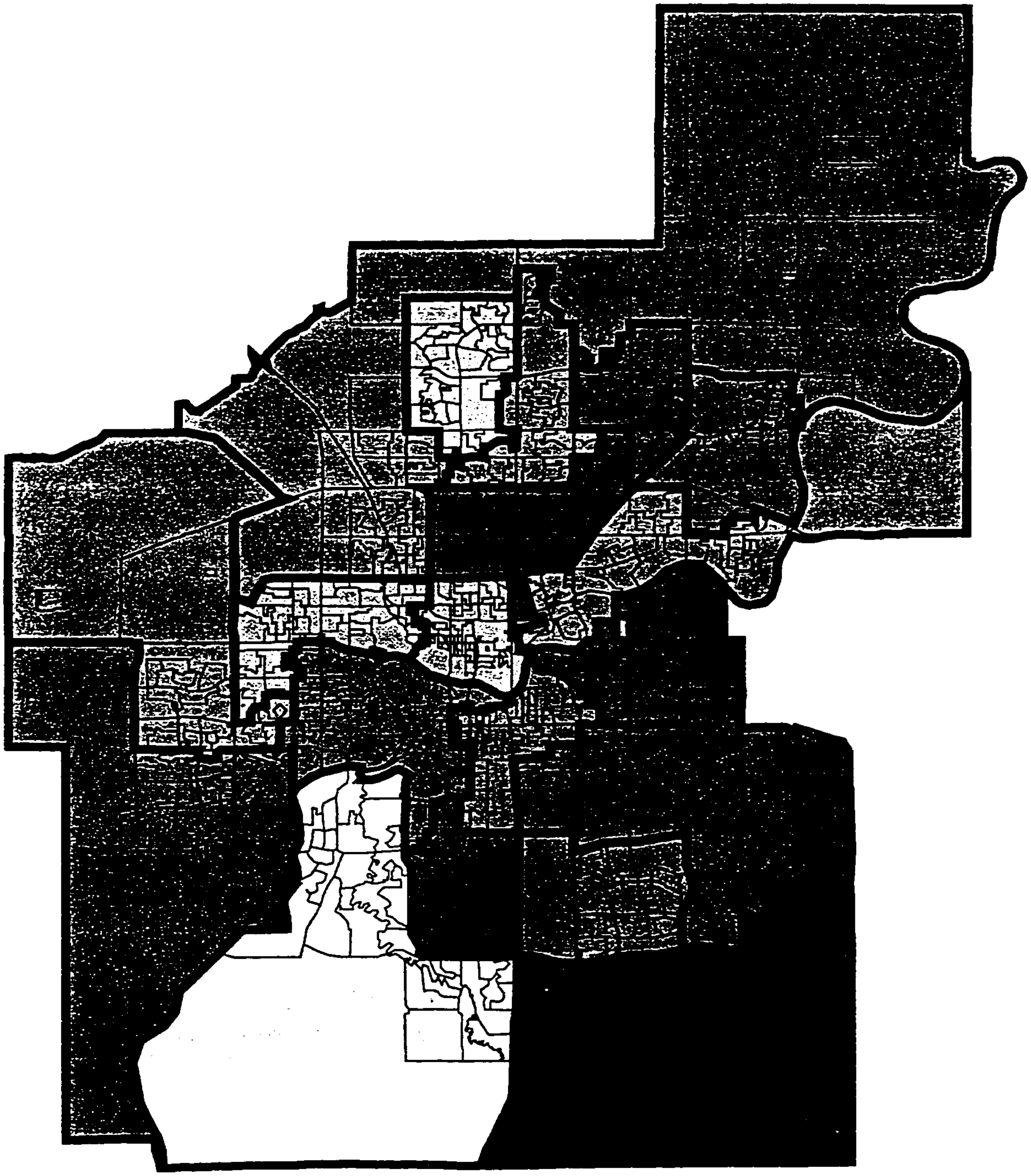

Figure D.2: District plan for (E/S-3/10%/1)

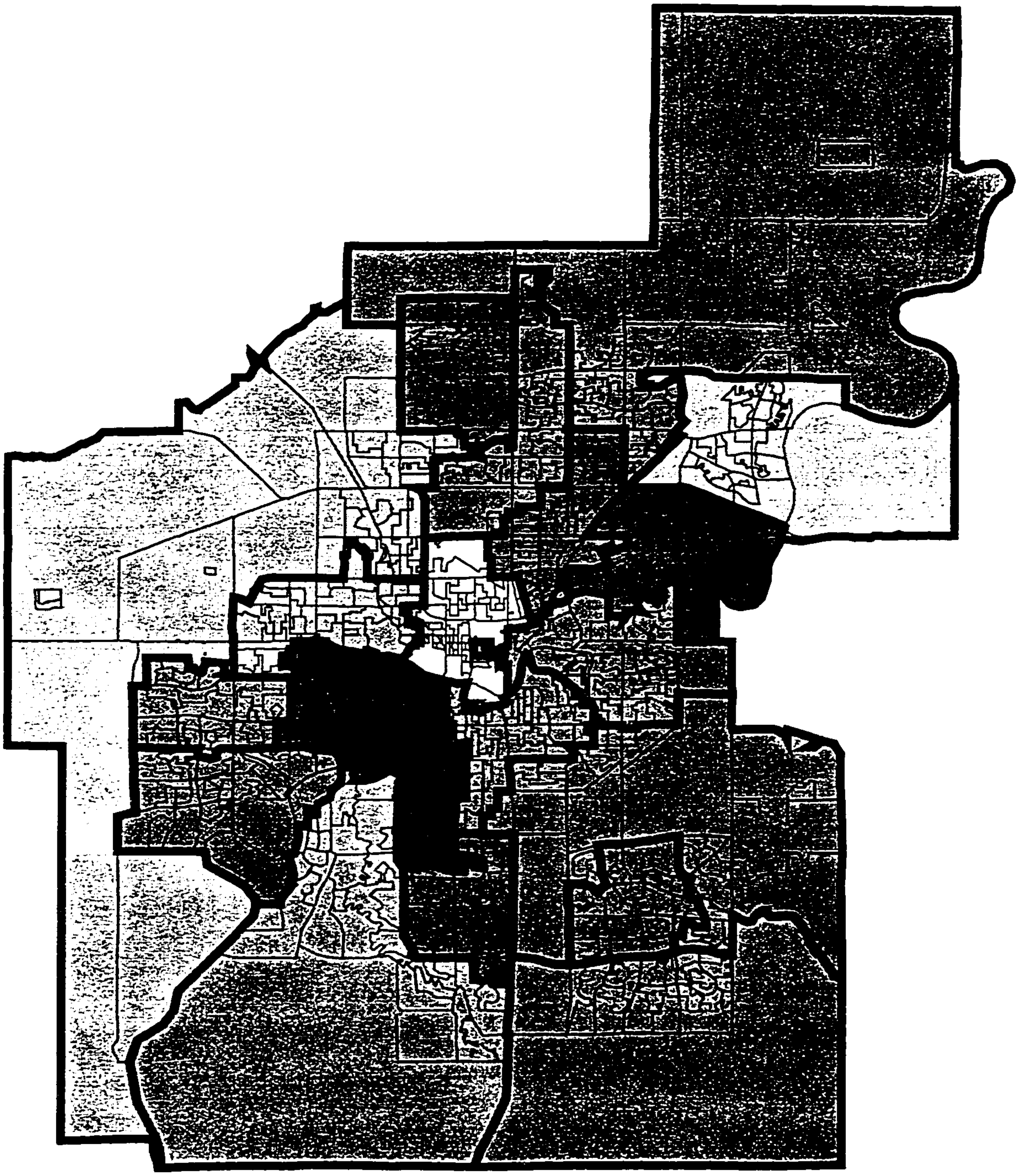

Figure D.3: District plan for (E/S-3/5%/1)

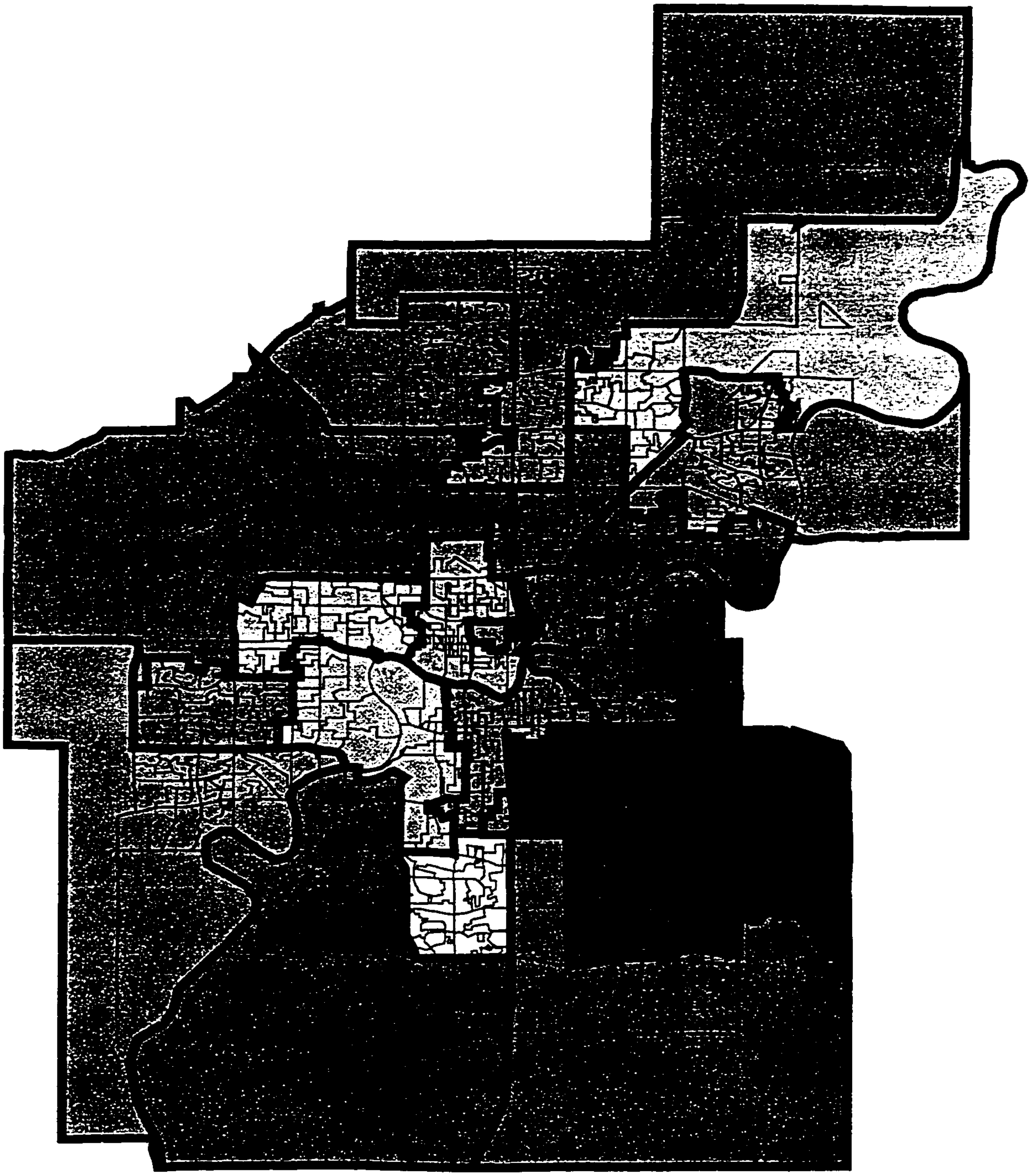

Figure D.4: District plan for (E/S-3/25%/2)

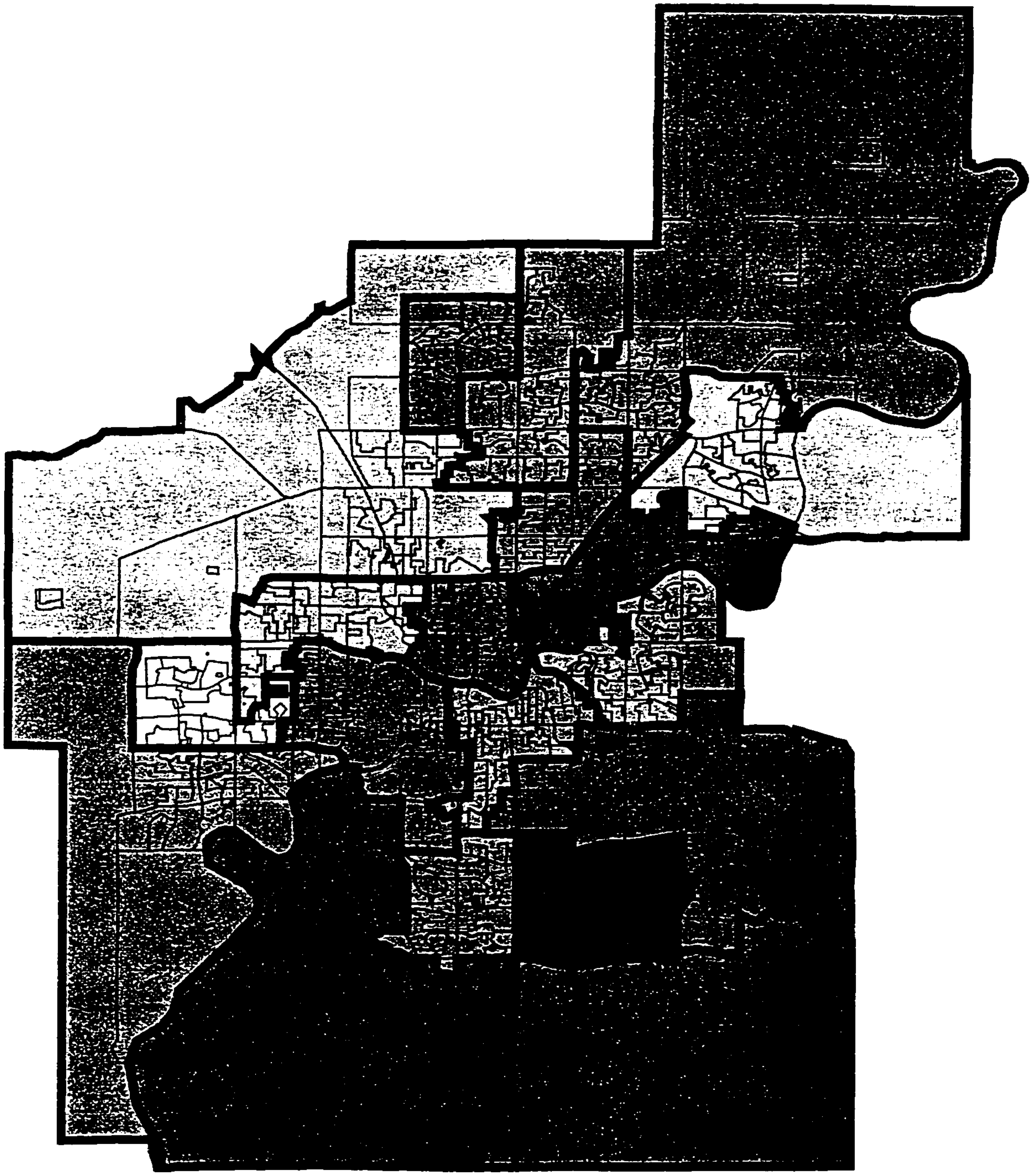

Figure D.5: District plan for (E/S-3/10%/2)

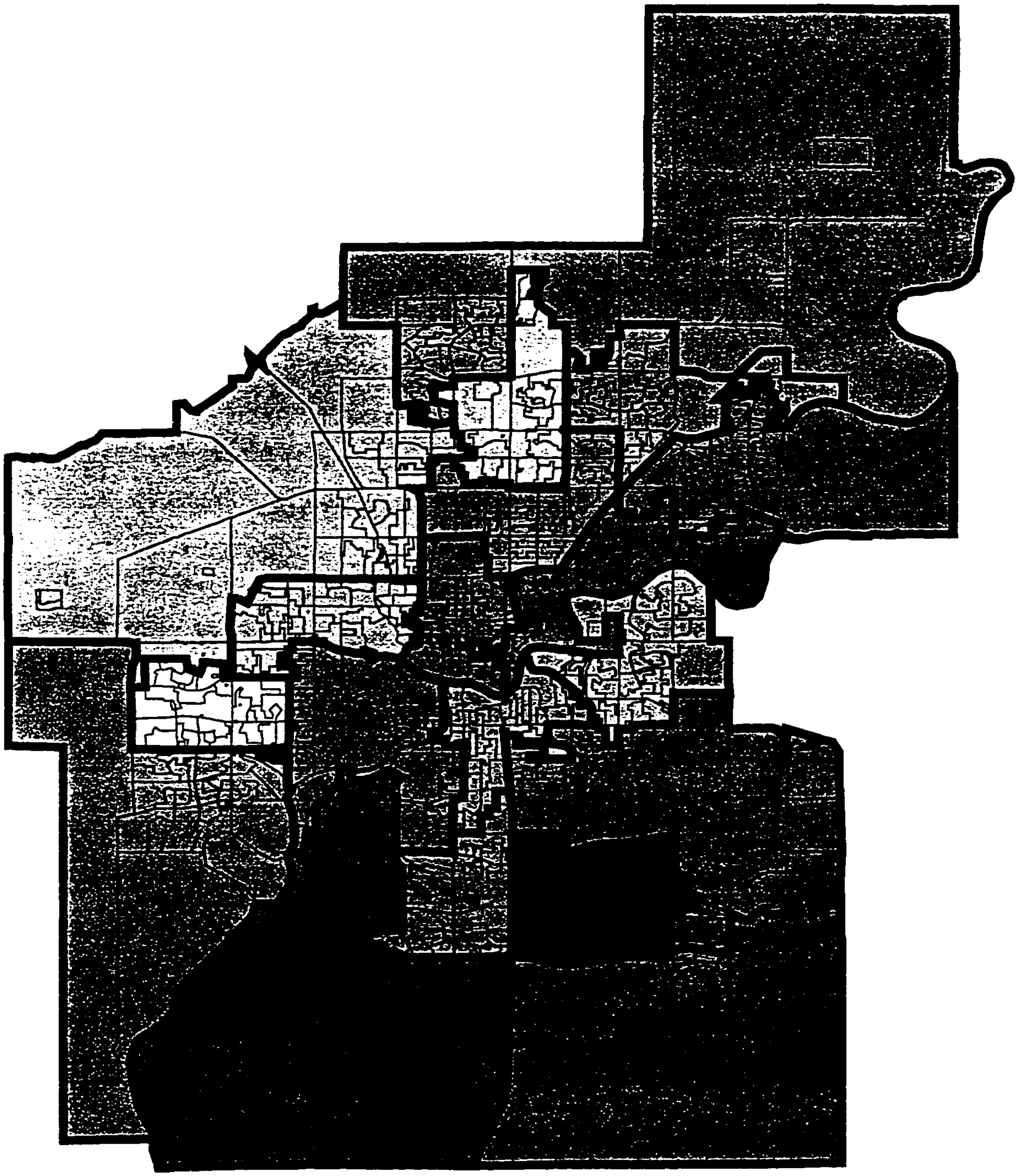

Figure D.6: District plan for (E/S-3/5%/2)

Appendix E

District Plans under Scenario 4

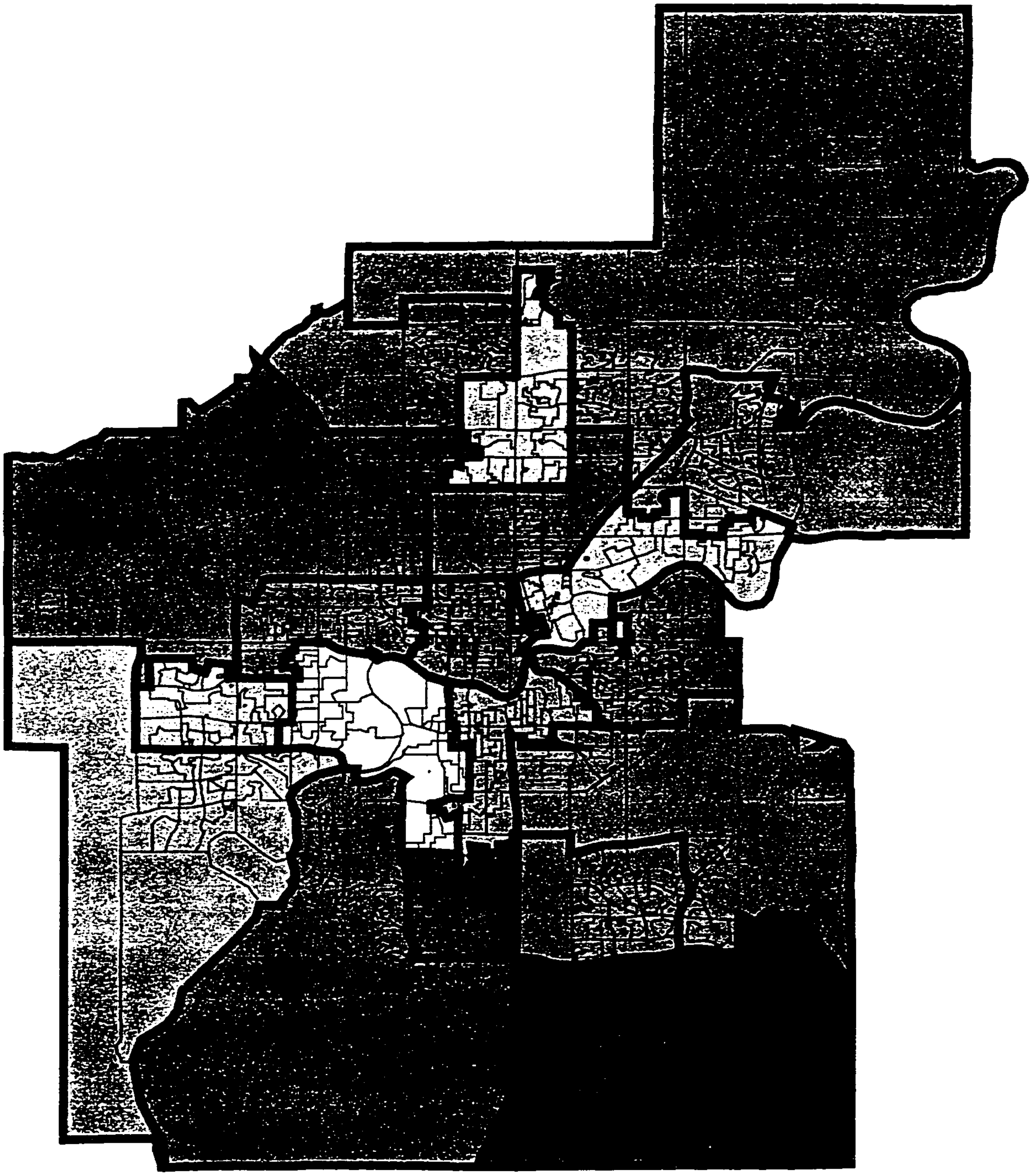

Figure E.1: District plan for (E/S-4/25%/1)

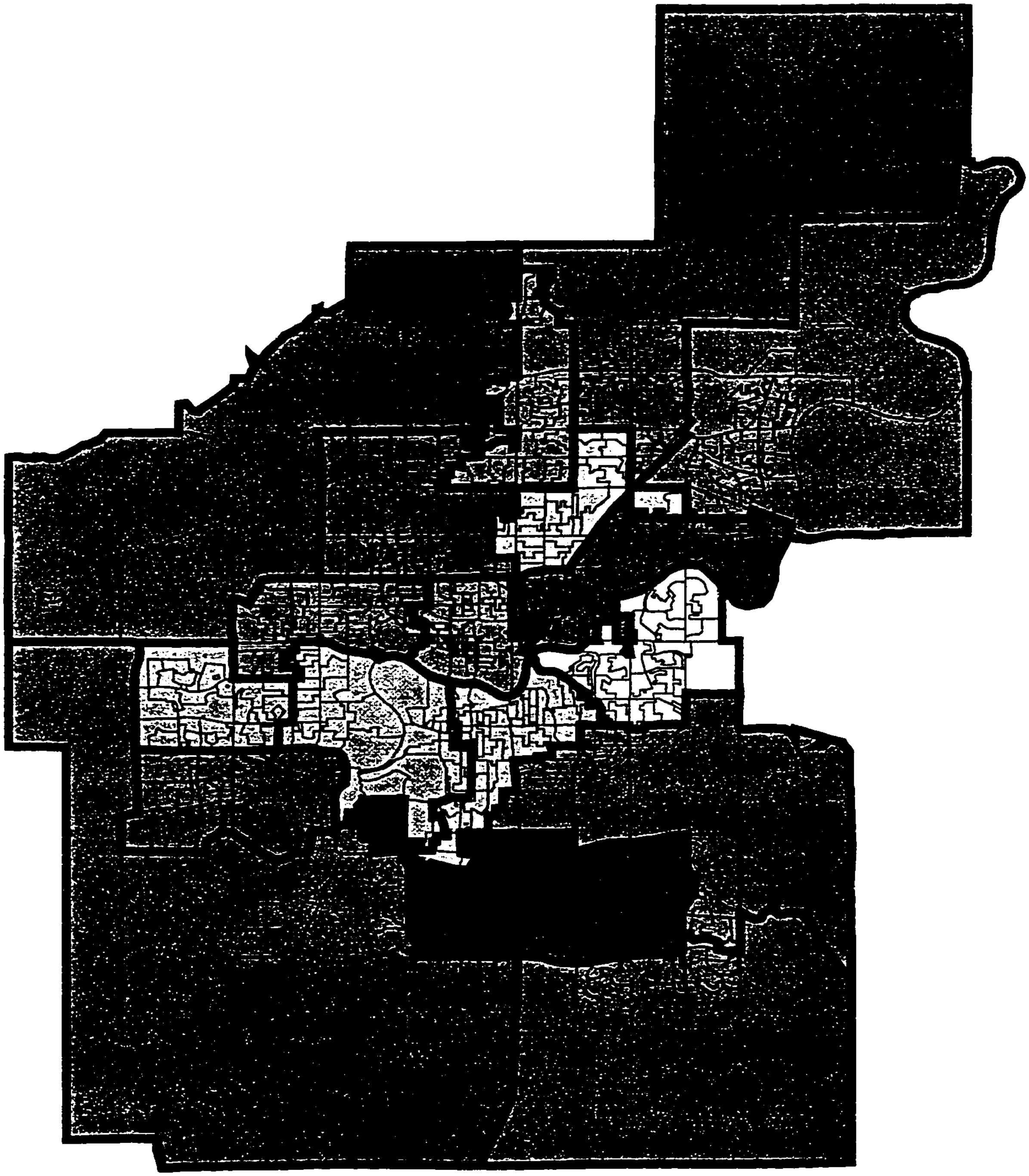

Figure E.2: District plan for (E/S-4/10%/1)

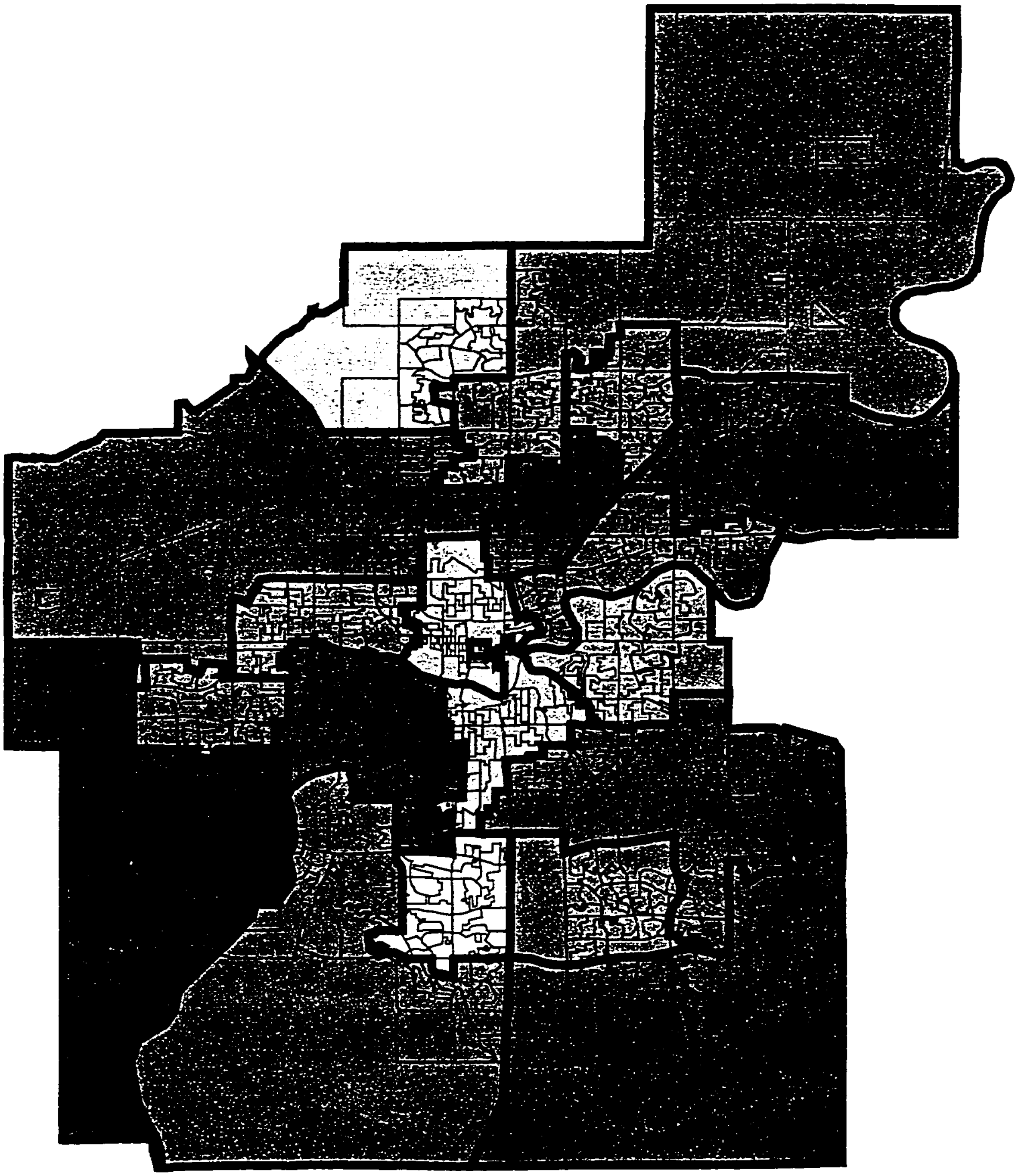

Figure E.3: District plan for (E/S-4/5%/1)

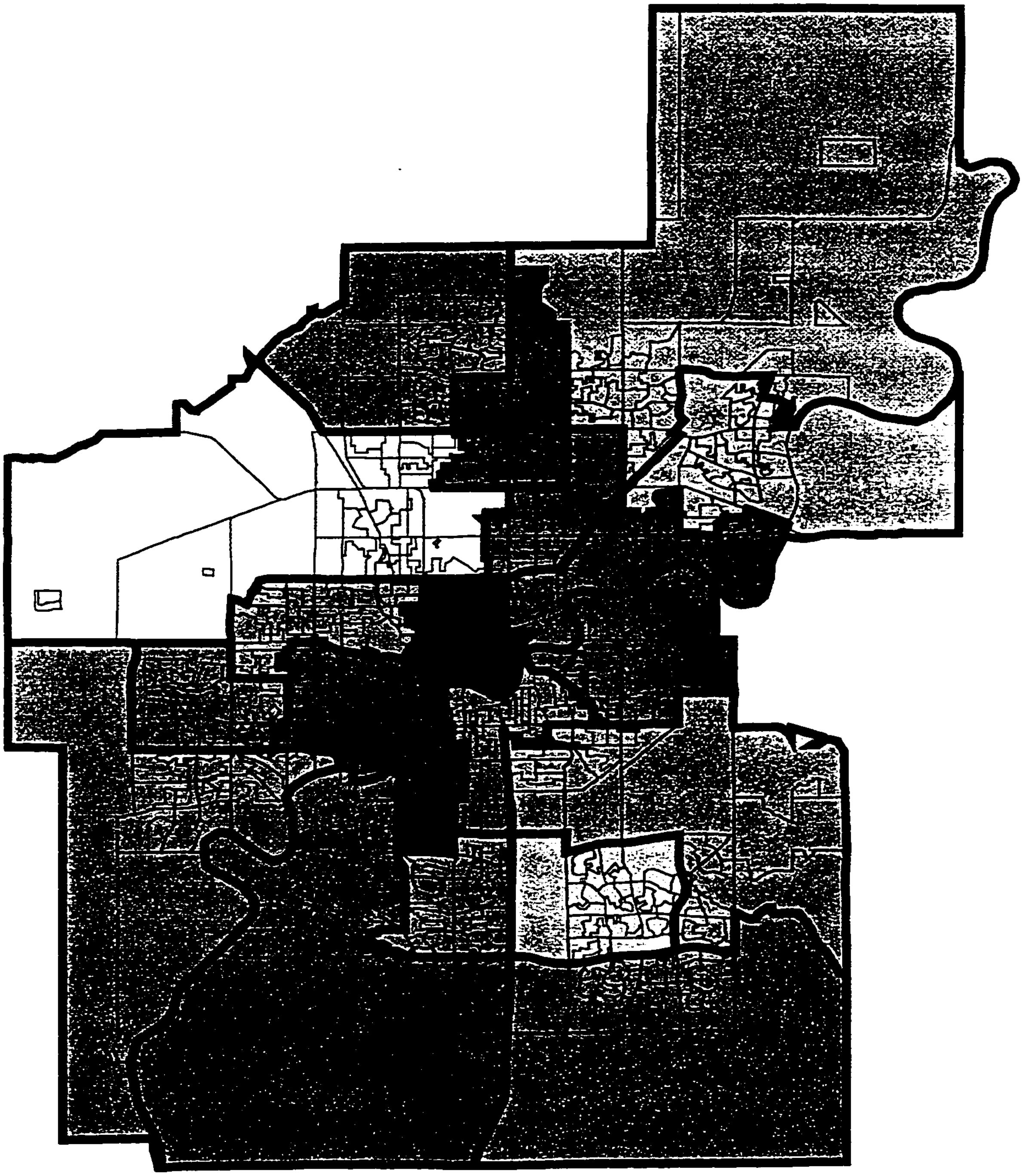

Figure E.4: District plan for (E/S-4/25%/2)

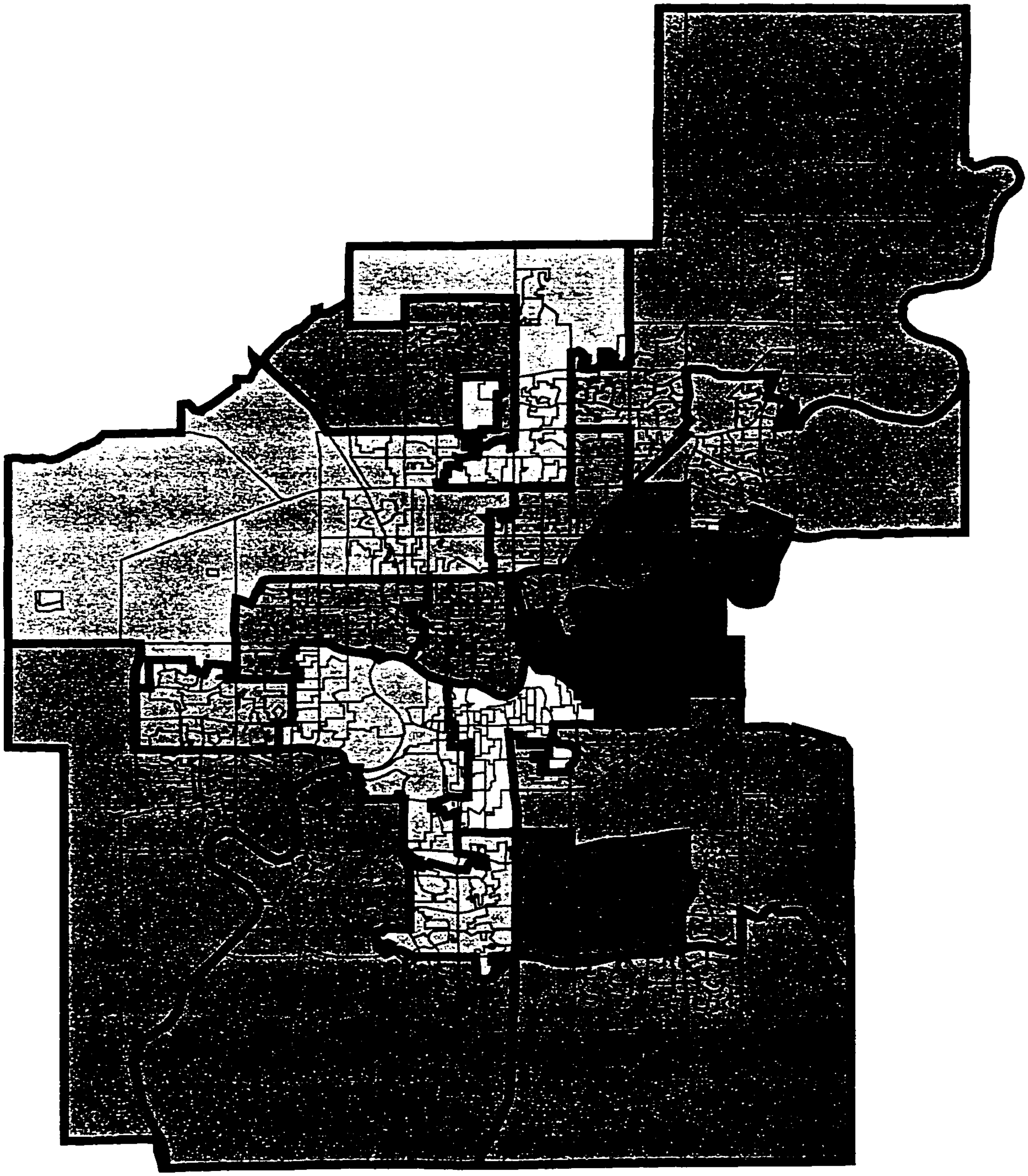

Figure E.5: District plan for (E/S-4/10%/2)

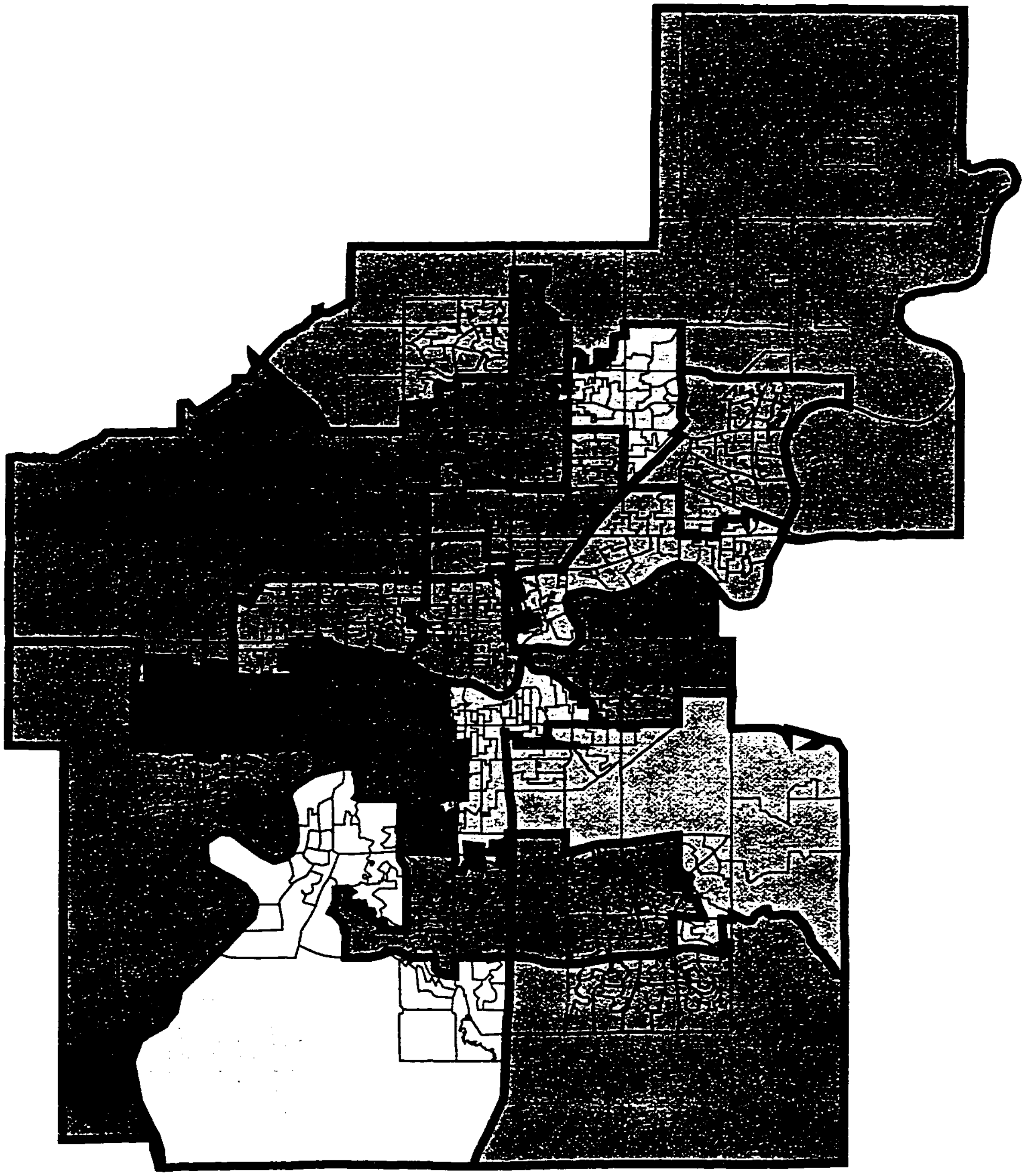

Figure E.6: District plan for (E/S-4/5%/2)

Appendix F

District Plans under Scenario 5

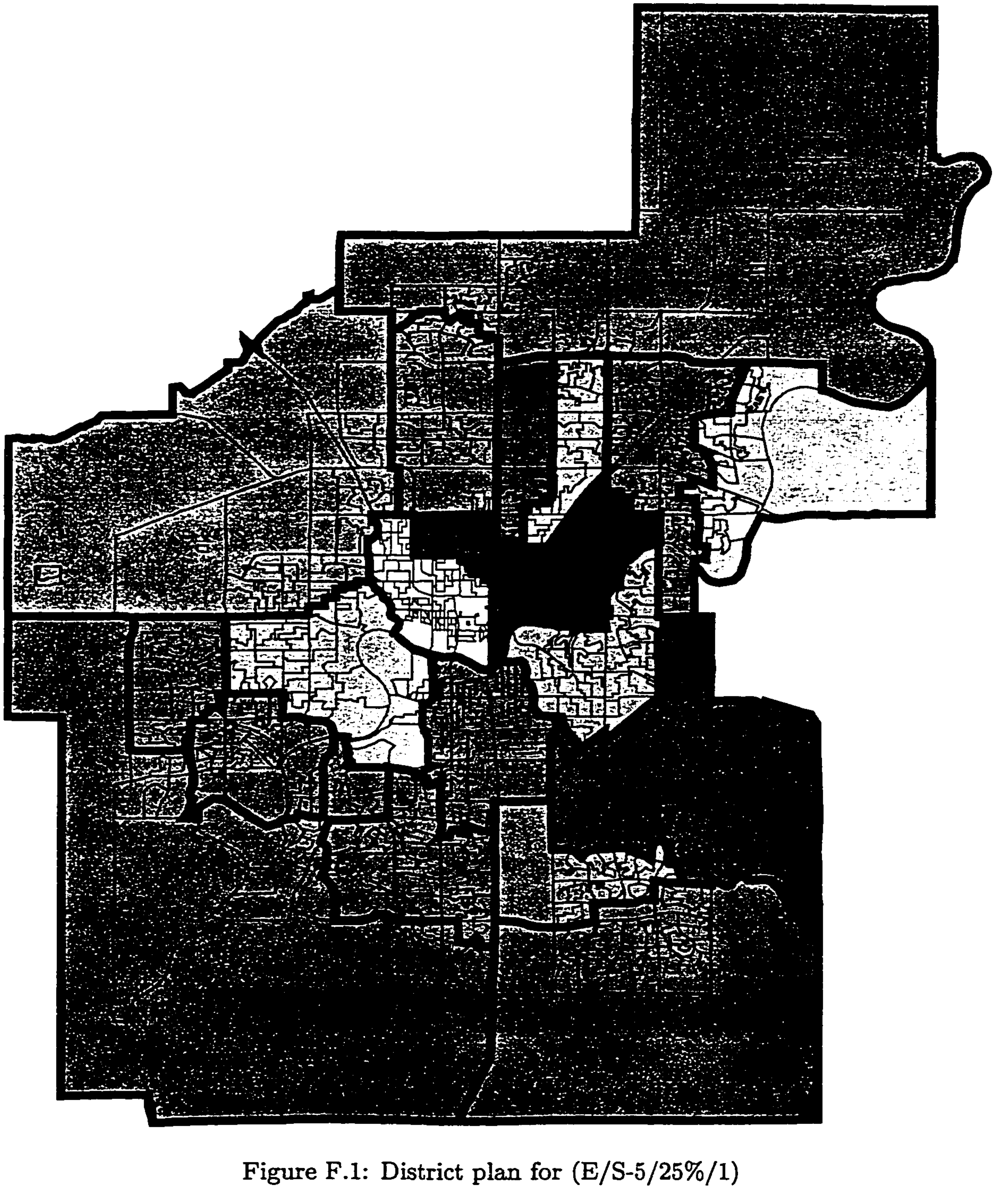

Figure F.1: District plan for (E/S-5/25%/1)

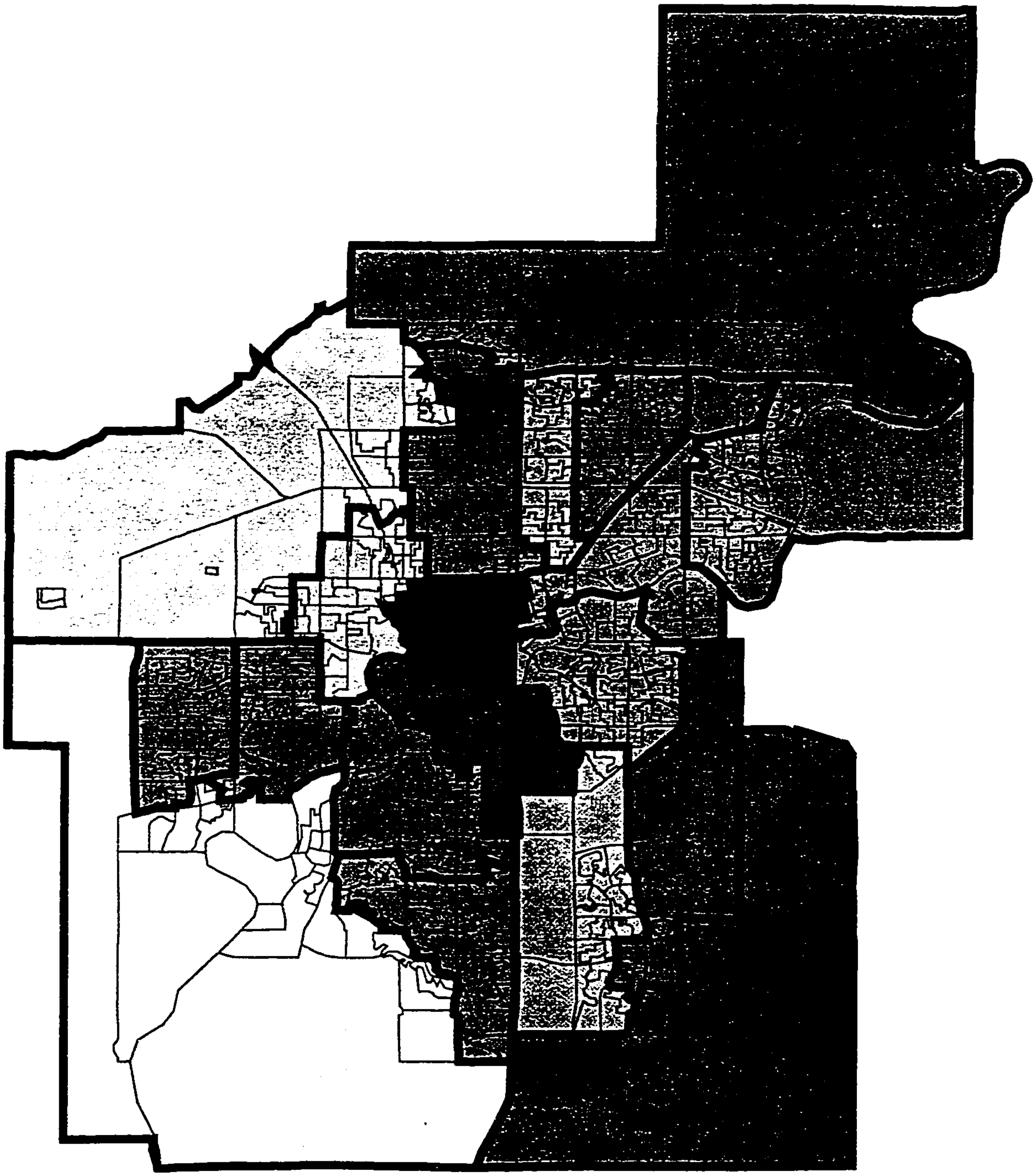

Figure F.2: District plan for (E/S-5/10%/1)

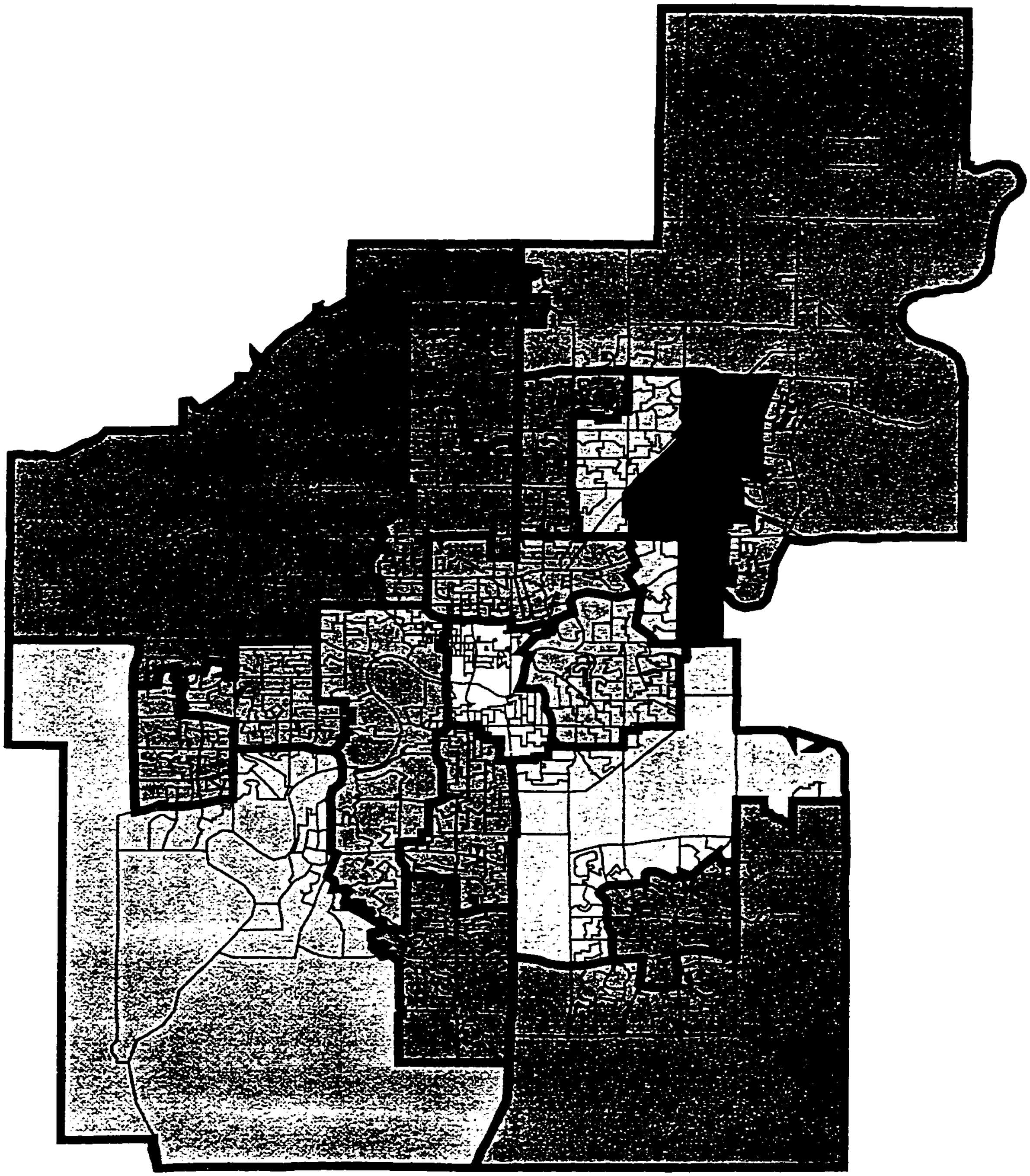

Figure F.3: District plan for (E/S-5/5%/1)

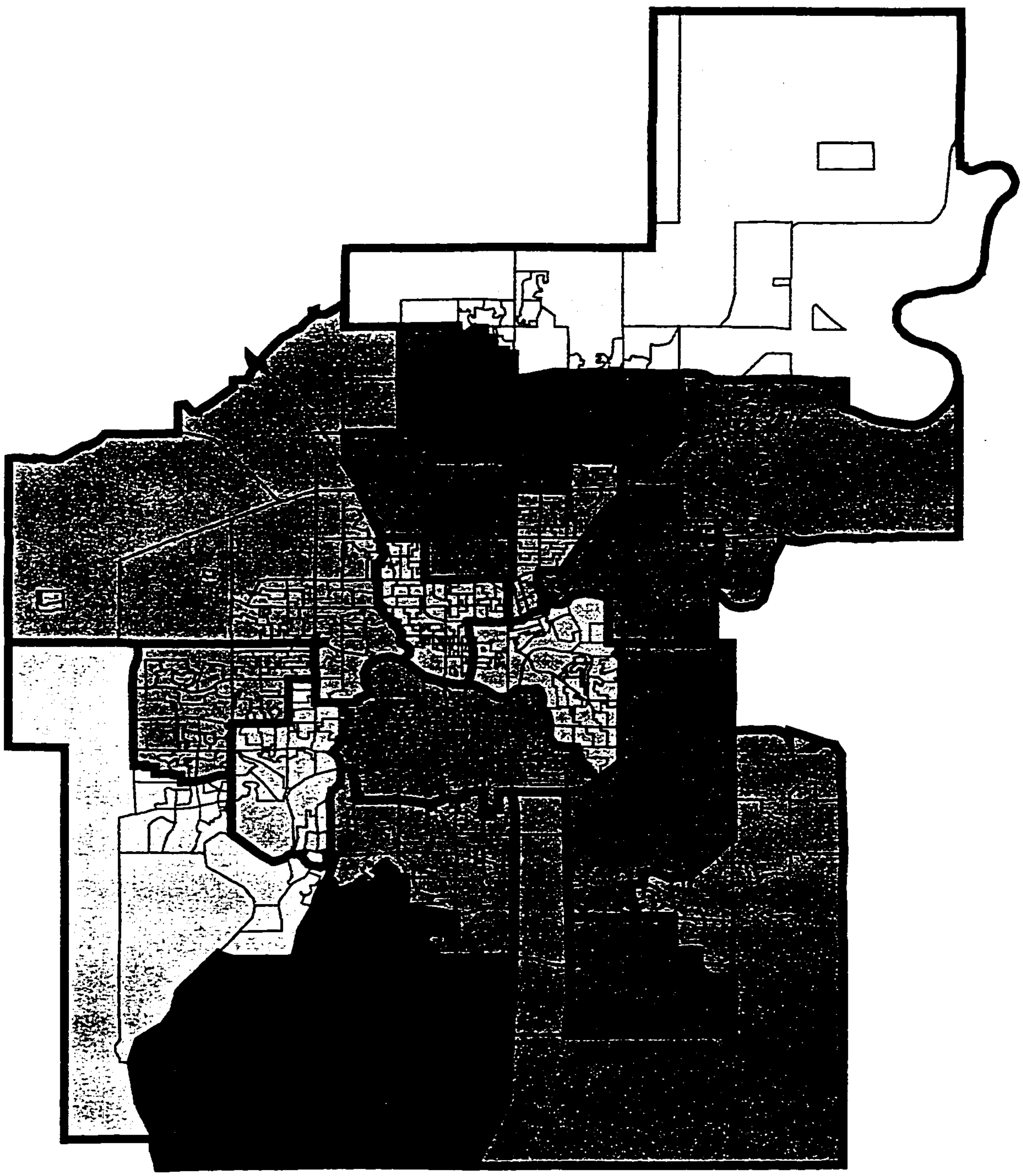

Figure F.4: District plan for (E/S-5/25%/2)

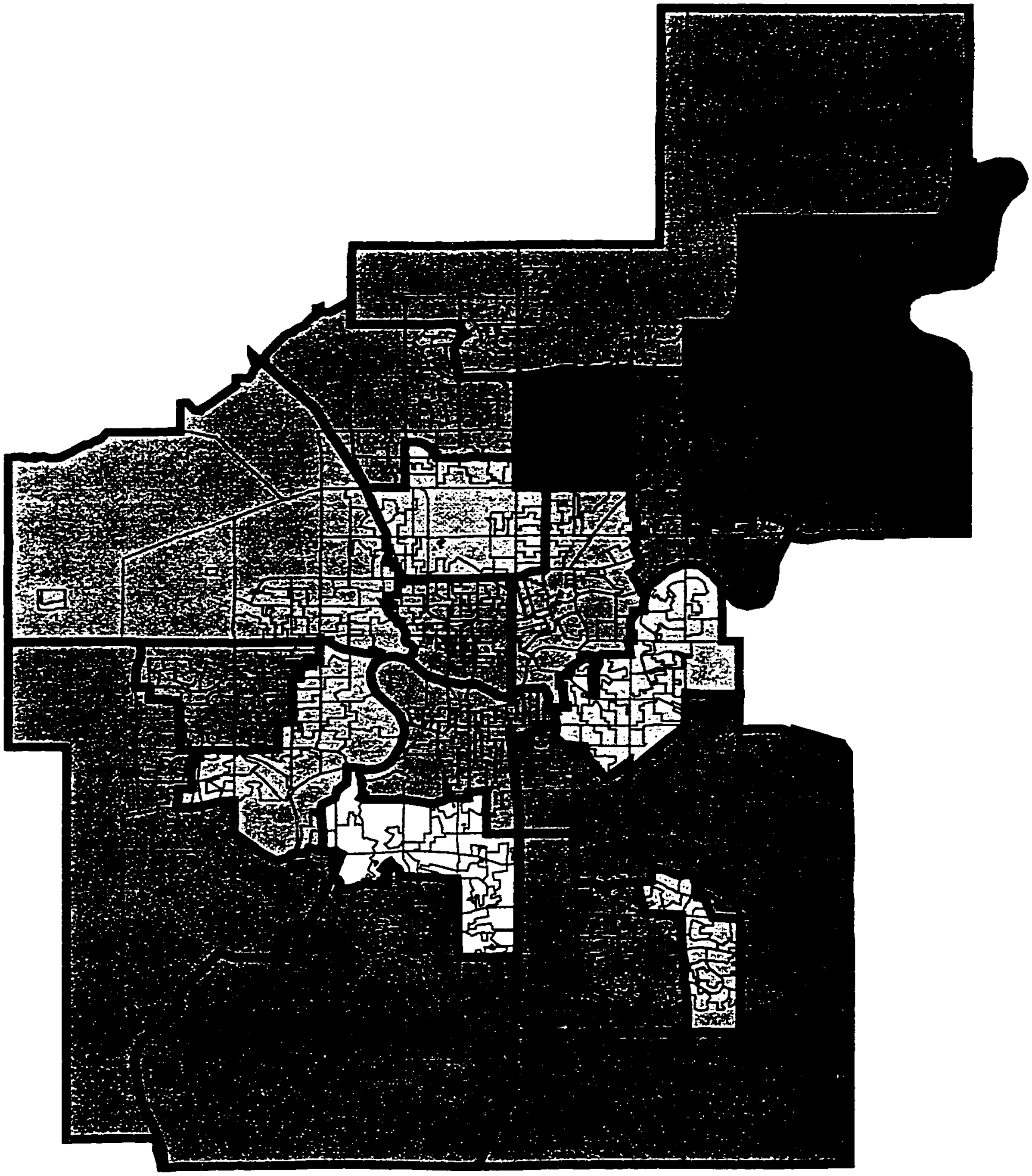

Figure F.5: District plan for (E/S-5/10%/2)

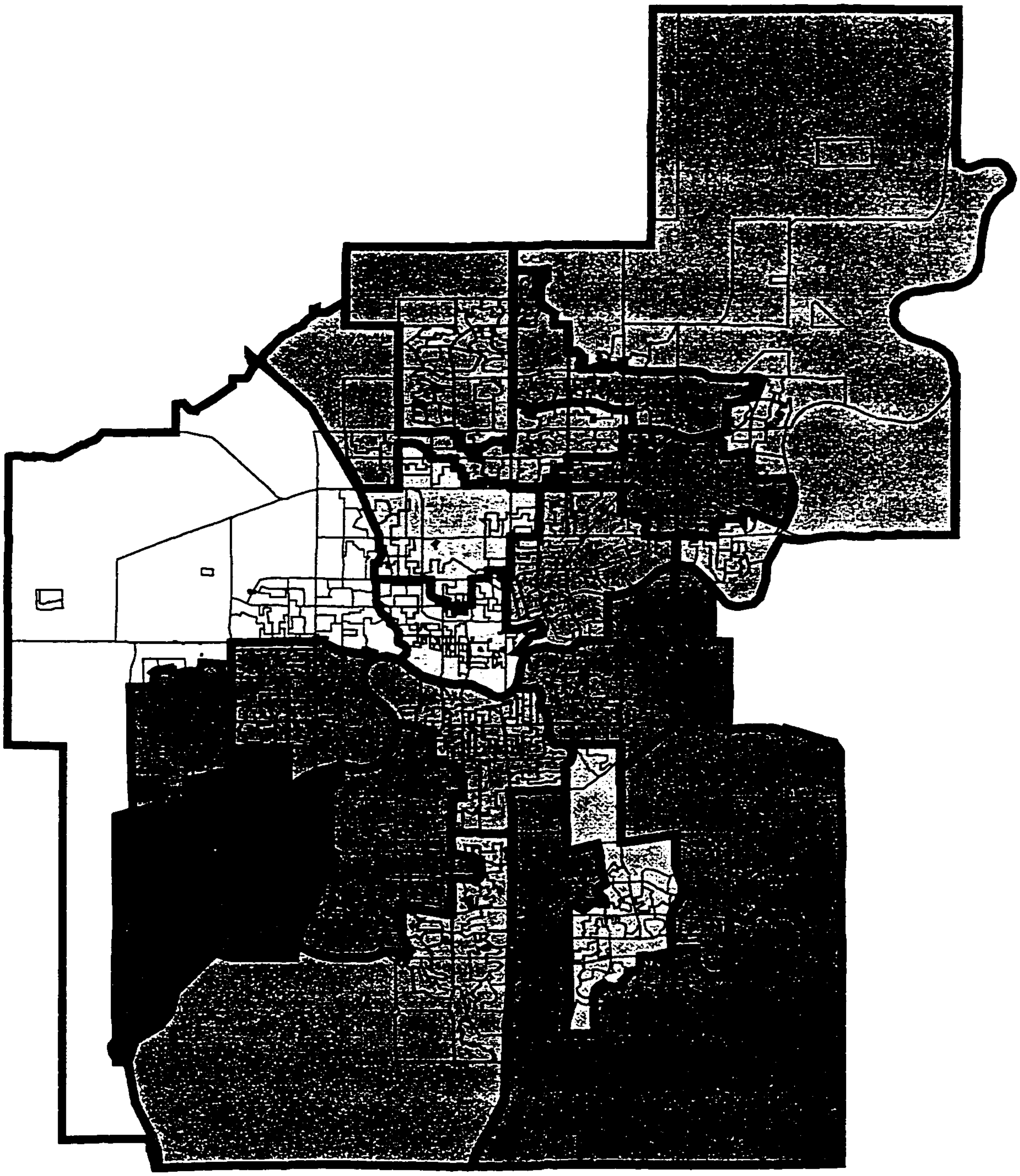

Figure F.6: District plan for (E/S-5/5%/2)

Appendix G

District Plans under Scenario 6

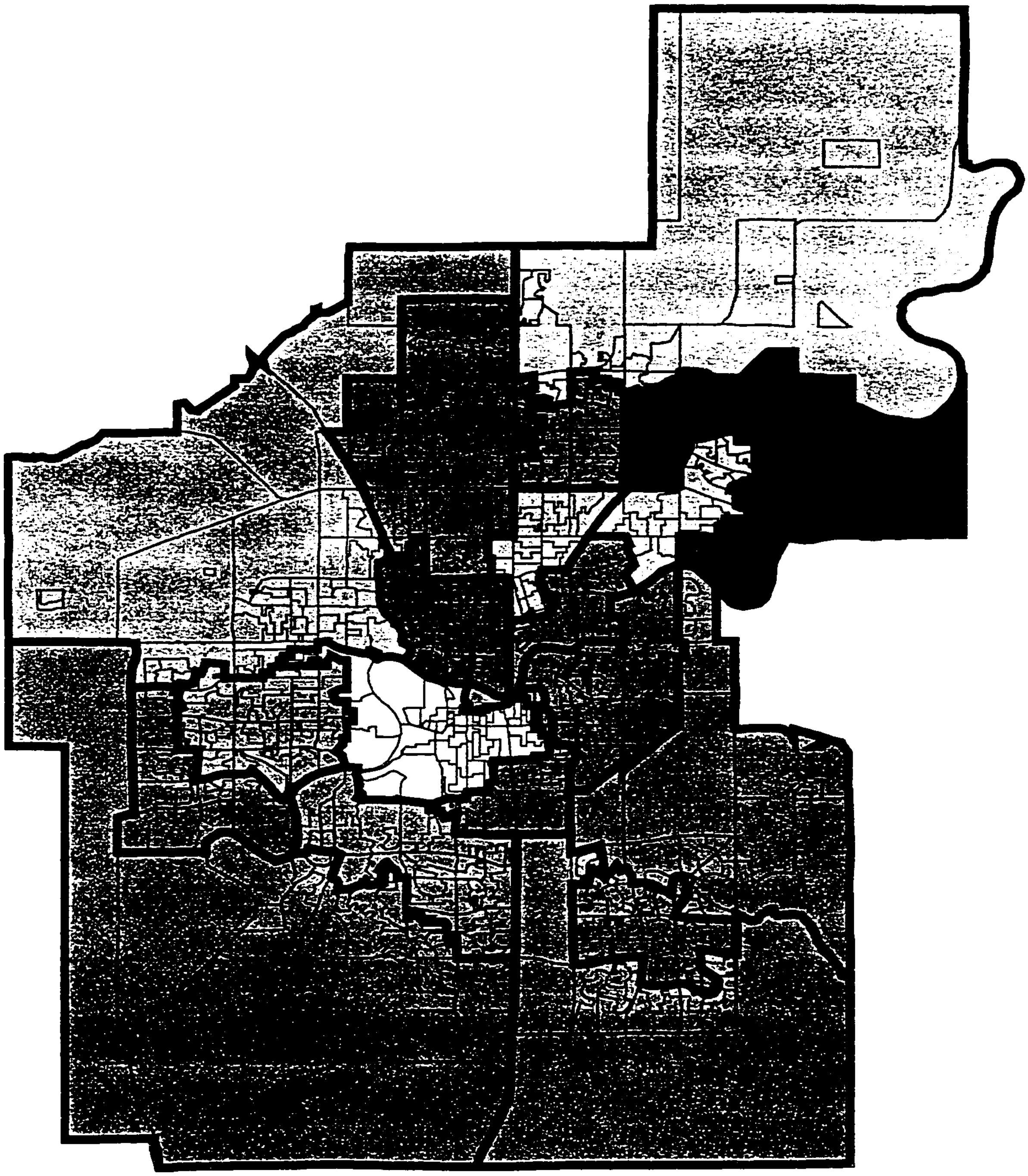

Figure G.1: District plan for (E/S-6/25%/1)

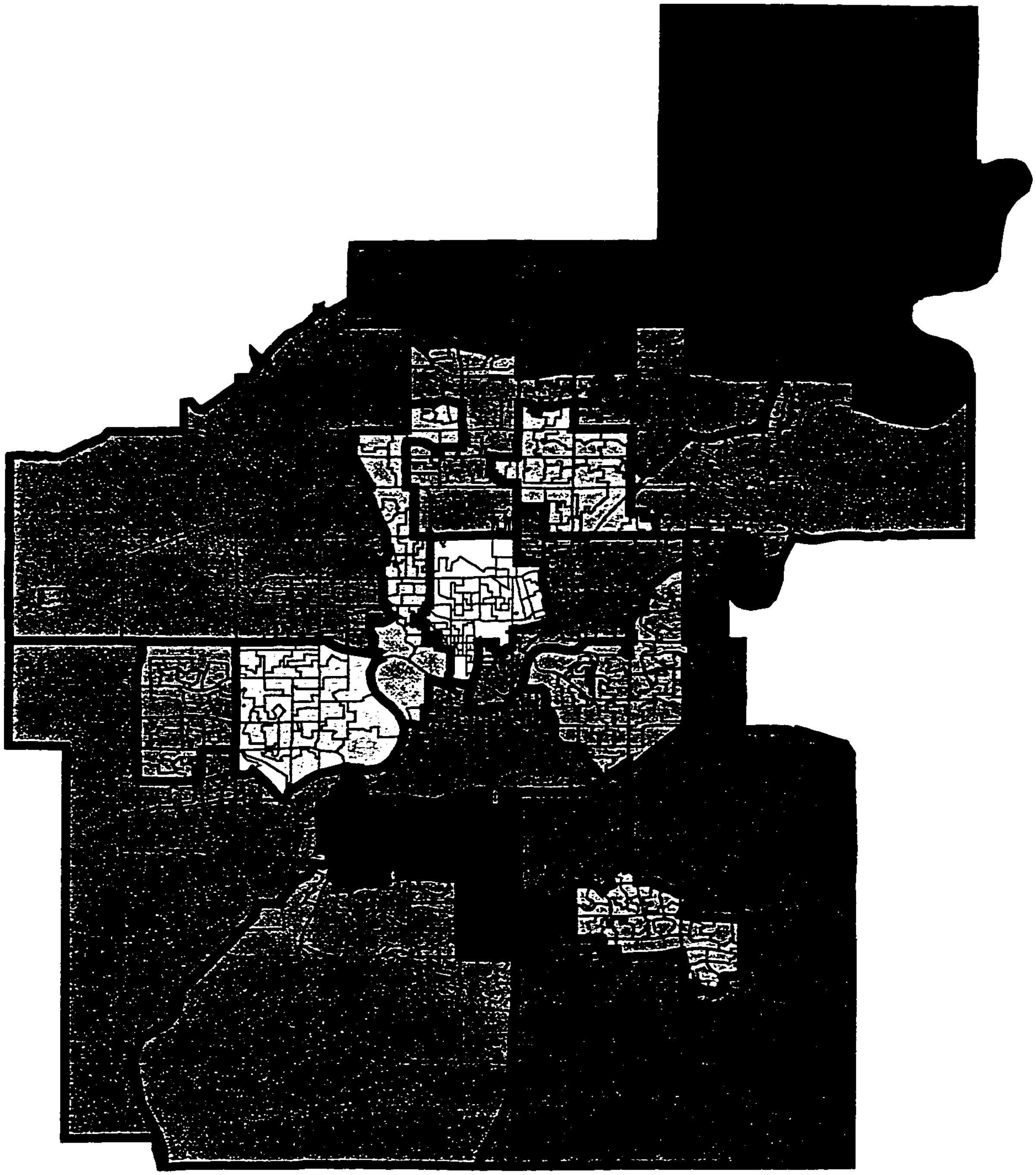

Figure G.2: District plan for (E/S-6/10%/1)

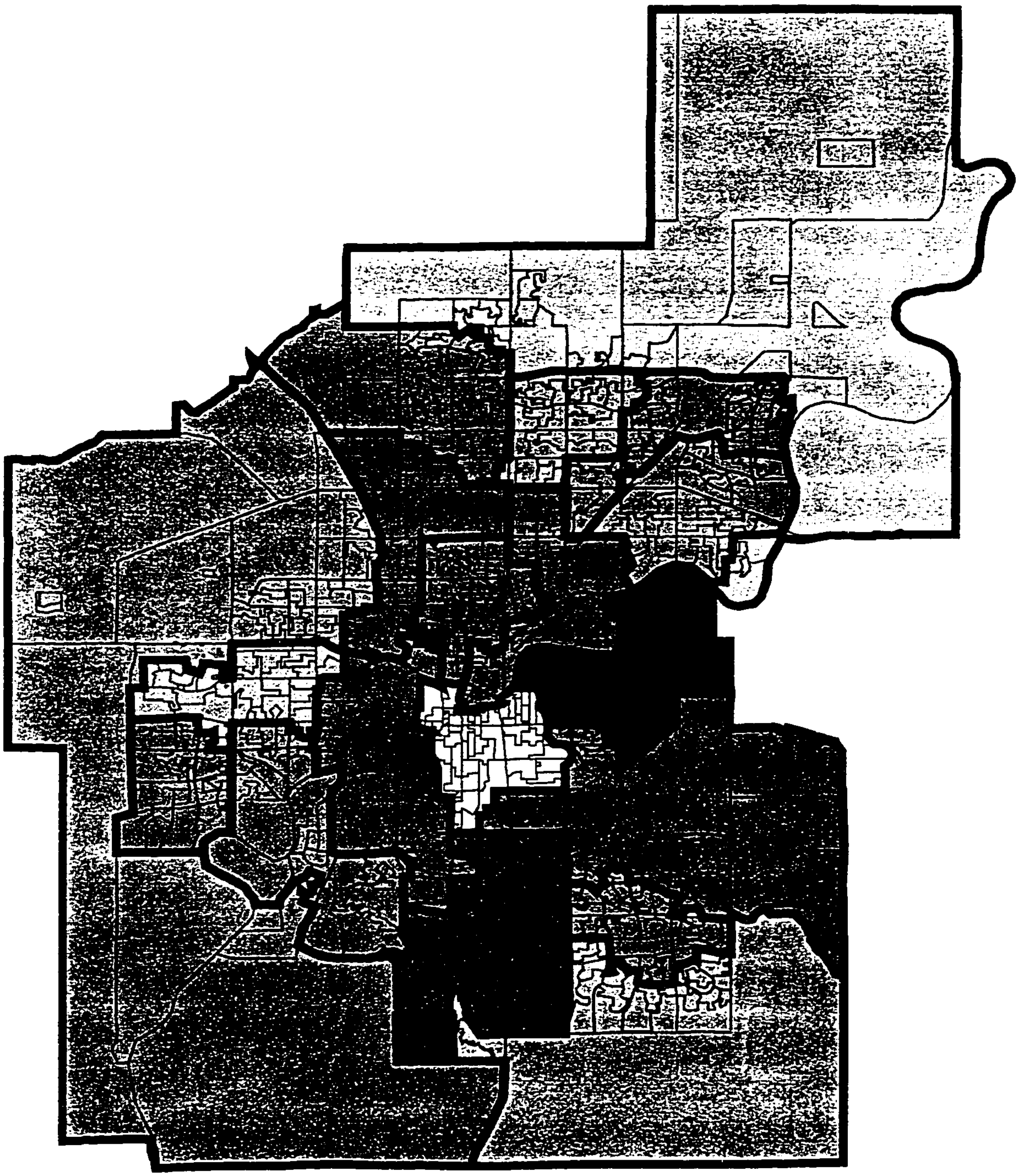

Figure G.3: District plan for (E/S-6/5%/1)

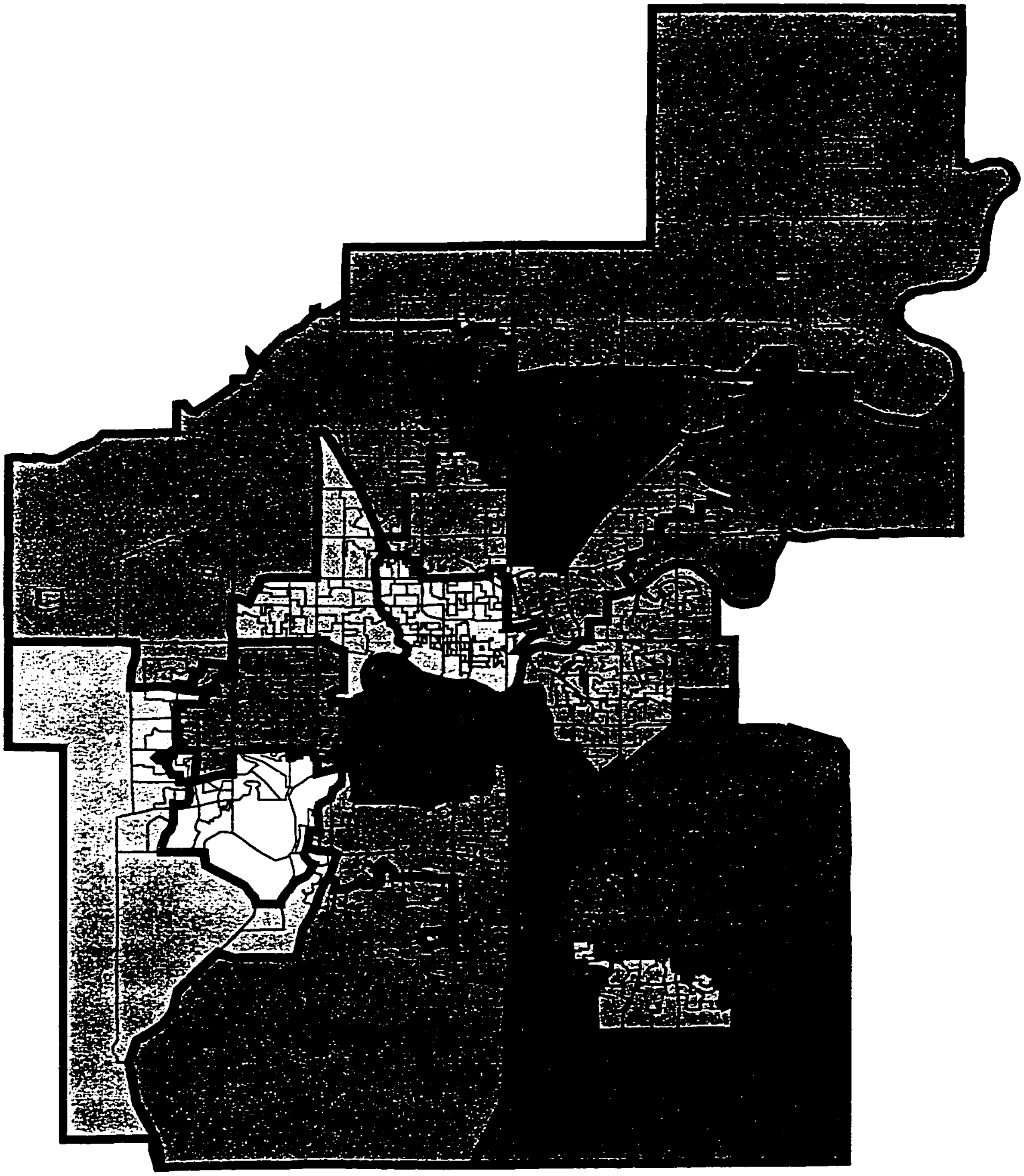

Figure G.4: District plan for (E/S-6/25%/2)

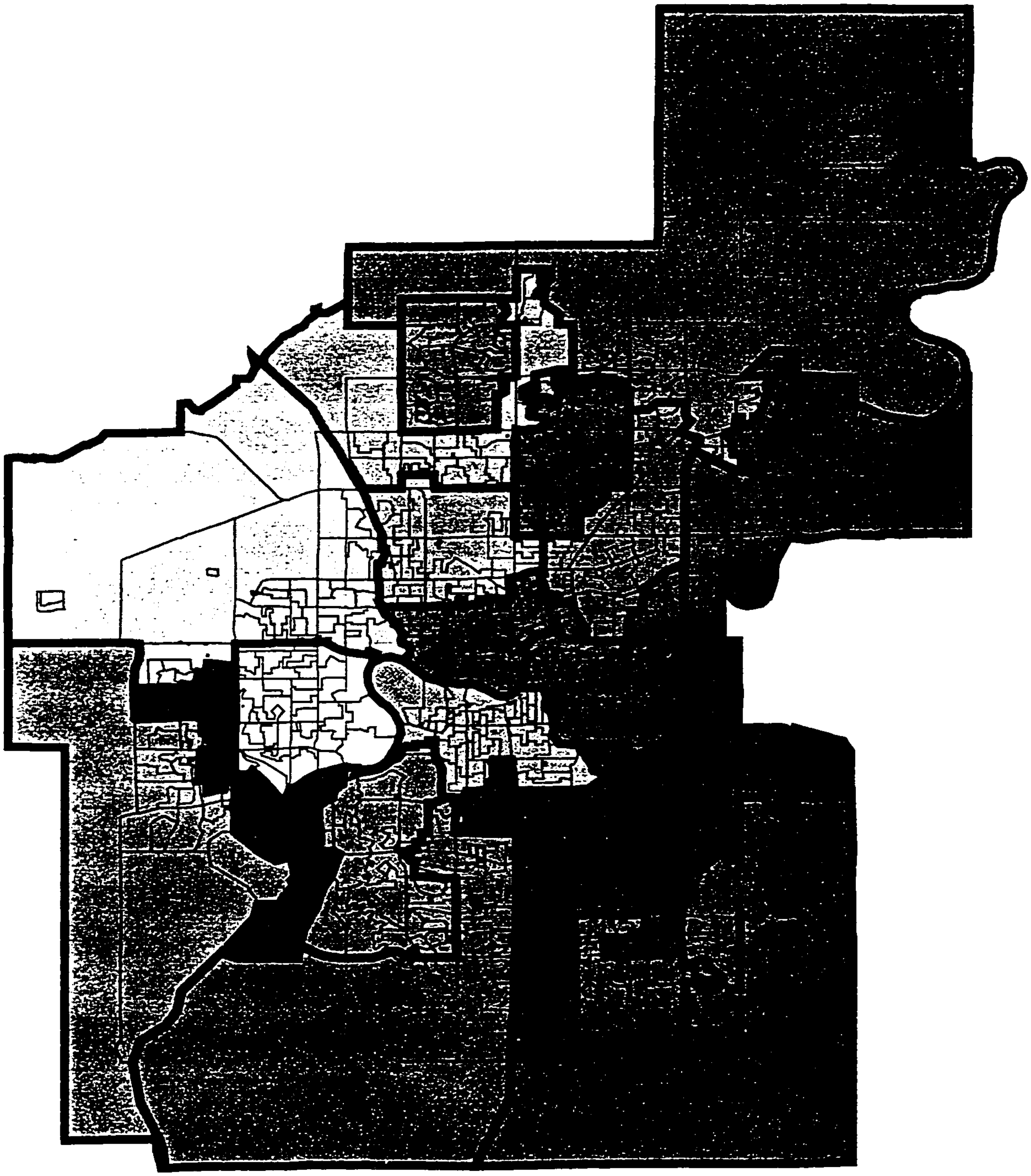

Figure G.5: District plan for (E/S-6/10%/2)

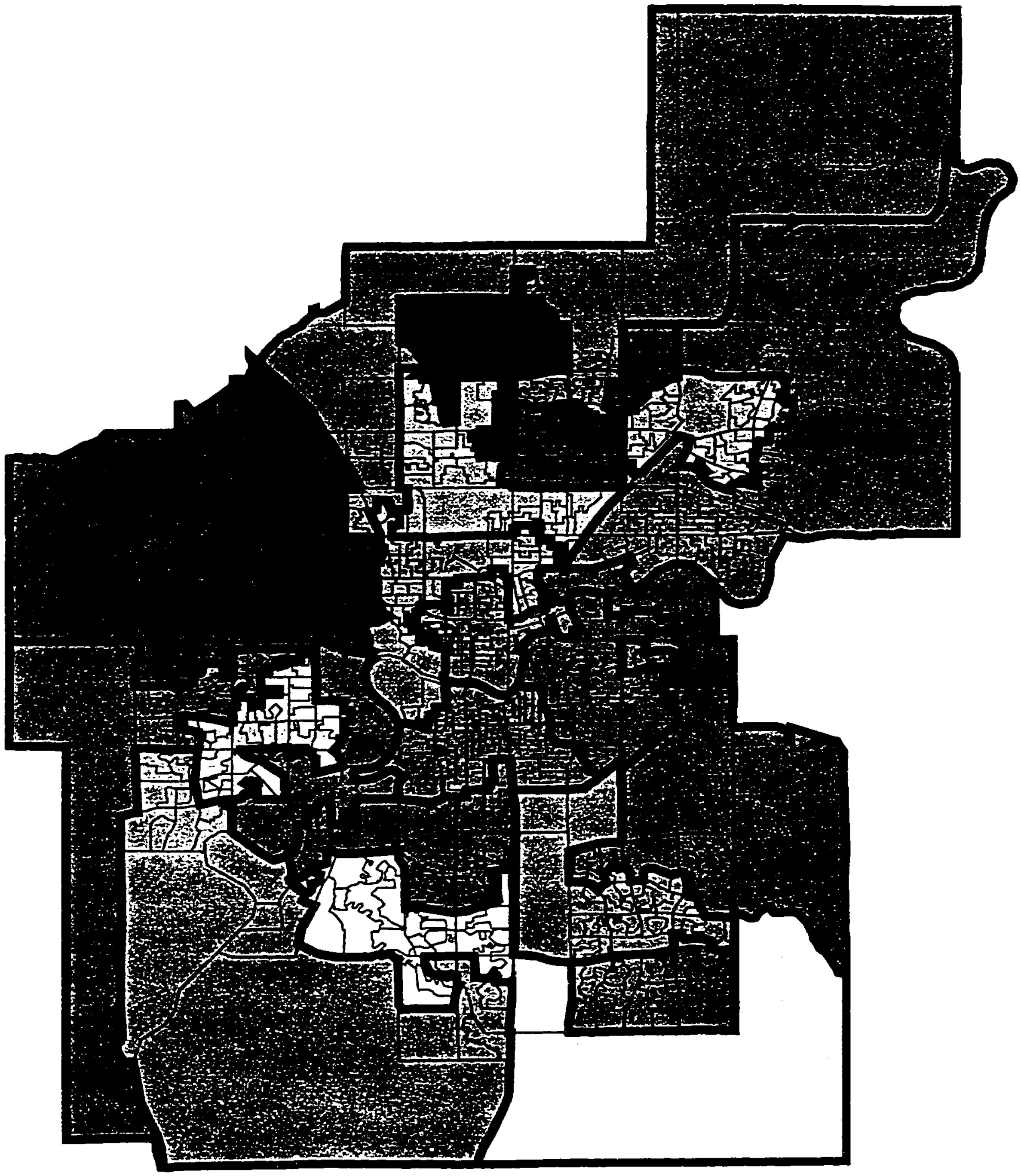

Figure G.6: District plan for (E/S-6/5%/2)

Appendix H

District Plans under Scenario 7

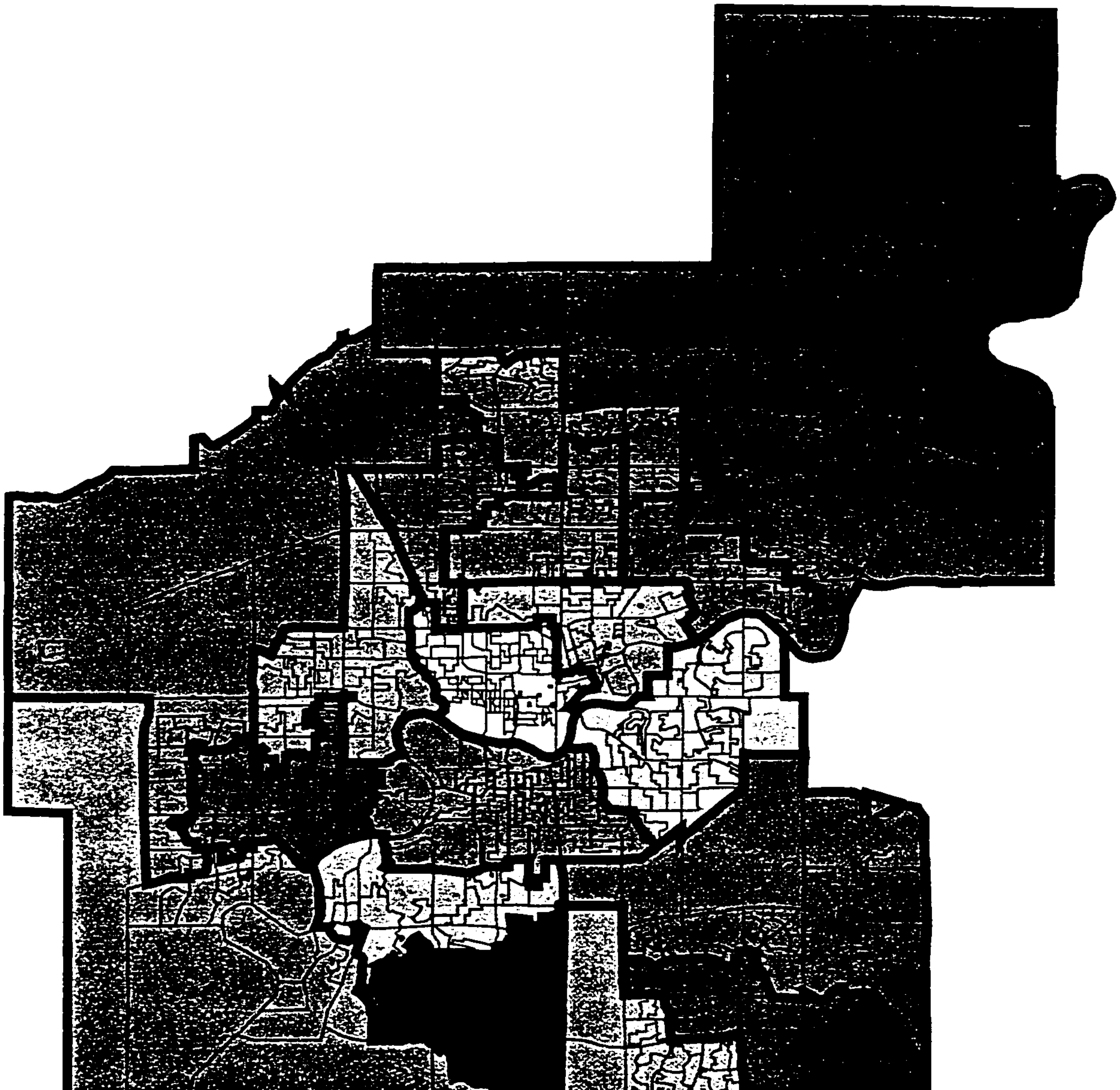

Figure H.1: District plan for (E/S-7/25%/1)

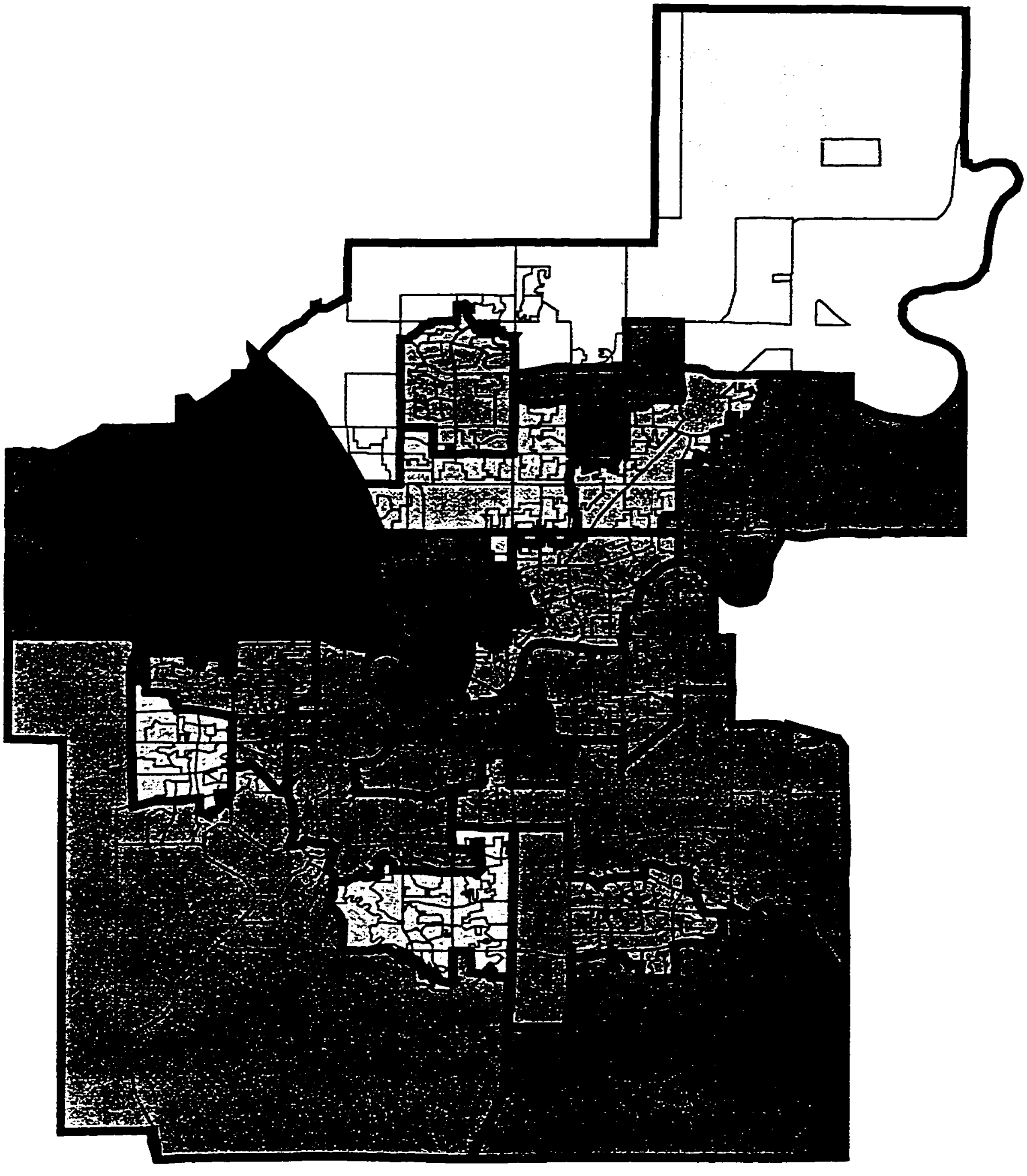

Figure H.2: District plan for (E/S-7/10%/1)

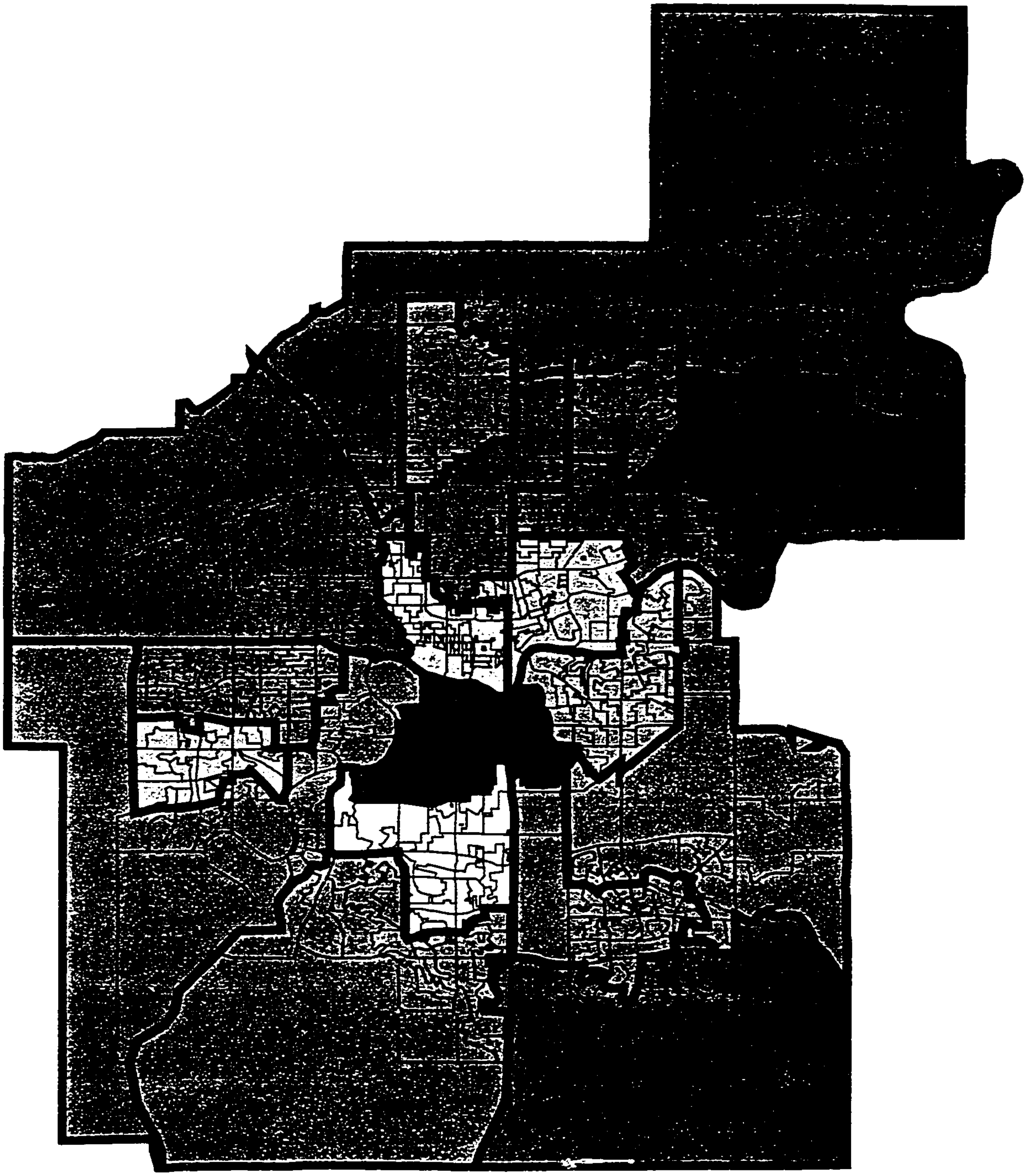

Figure H.3: District plan for (E/S-7/5%/1)

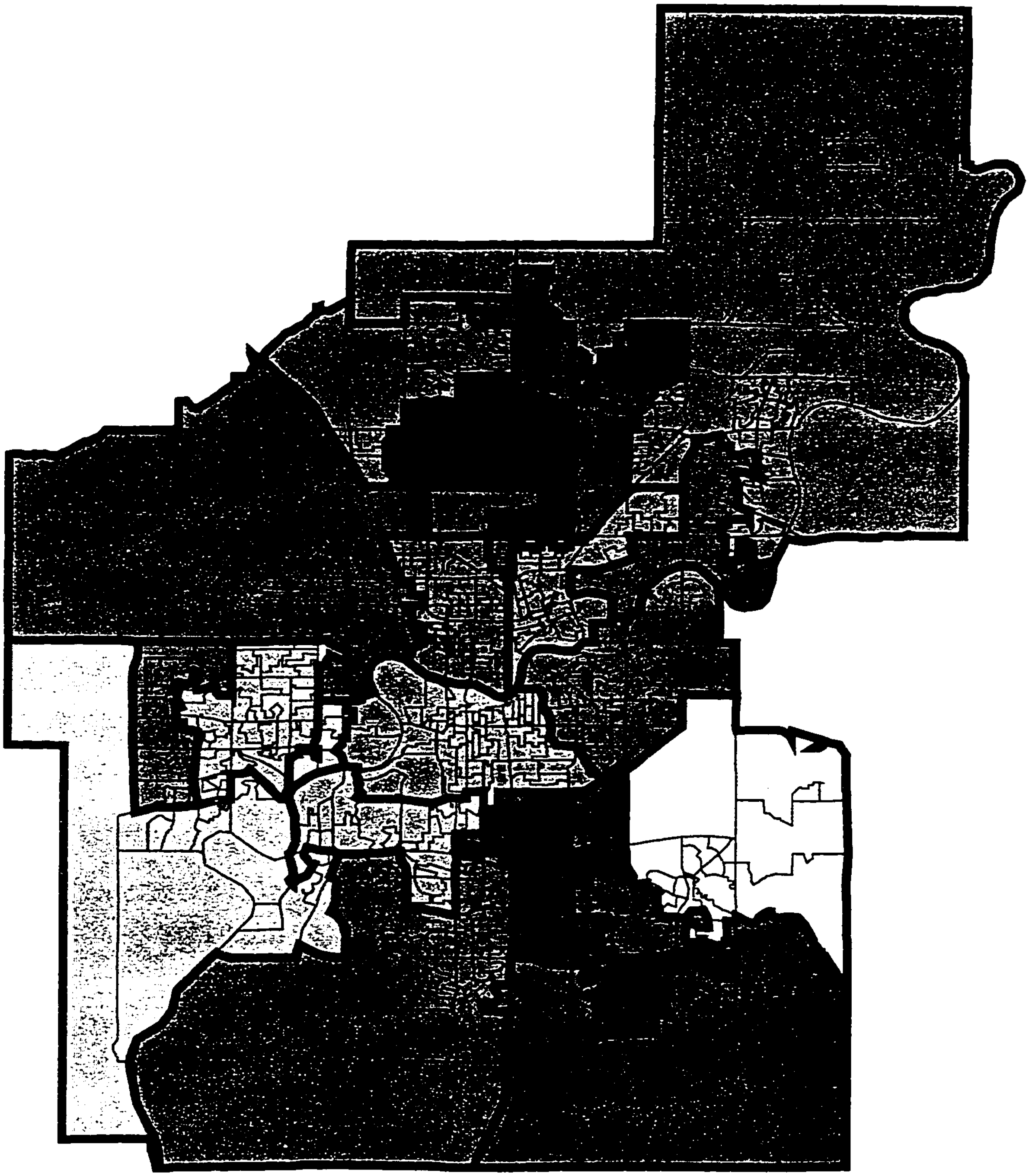

Figure H.4: District plan for (E/S-7/25%/2)

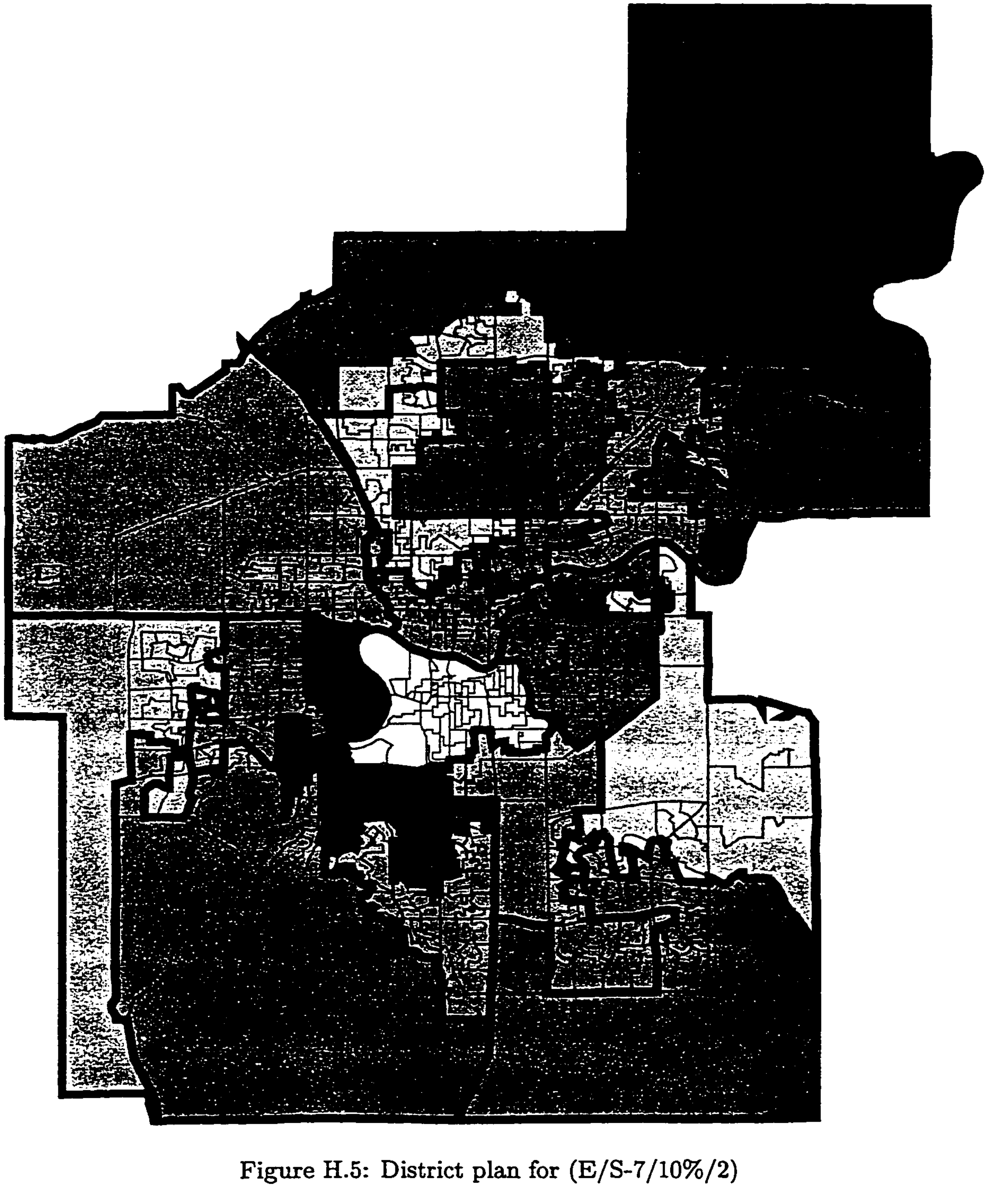

Figure H.5: District plan for (E/S-7/10%/2)

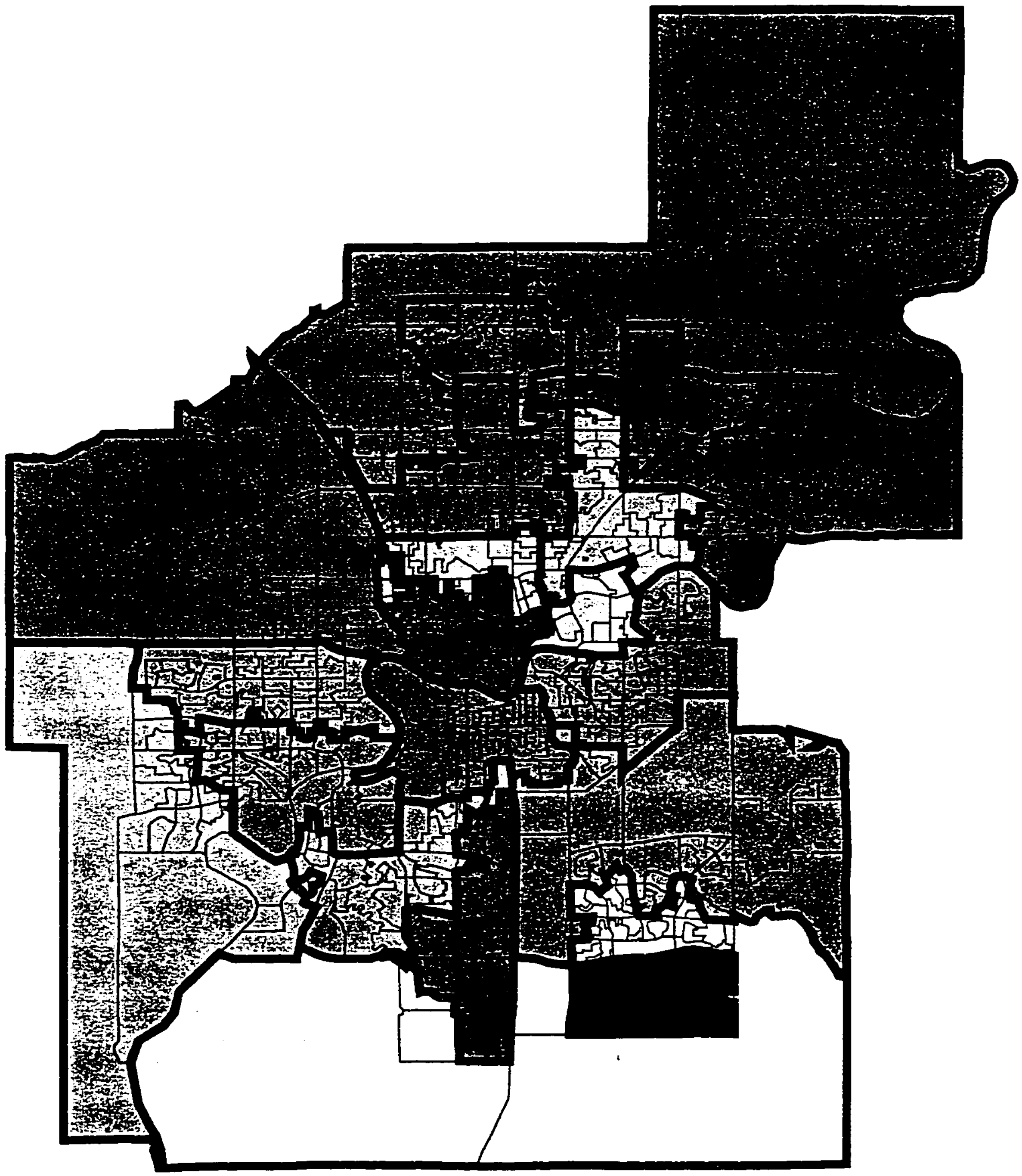

Figure H.6: District plan for (E/S-7/5%/2)