3 5 - 2 . 47 - 0

National Library of Canada — Bibliothèque nationale du Canada

Canadian Theses Division — Division des thèses canadiennes

Ottawa, Canada
K1A 0N4

PERMISSION TO MICROFILM — AUTORISATION DE MICROFILMER

- Please print or type — Écrire en lettres moulées ou dactylographier

Full Name of Author — Nom complet de l'auteur

[illegible]

Date of Birth — Date de naissance

[illegible]

Country of Birth — Lieu de naissance

MONTREAL

Permanent Address — Résidence fixe

955 Charles Rd
Boucherville Quebec

Title of Thesis — Titre de la thèse

Interactive Graphics for Mapping Location-Allocation Solutions

University — Université

University of Alberta

Degree for which thesis was presented — Grade pour lequel cette thèse fut présentée

Master of Arts

Year this degree conferred — Année d'obtention de ce grade

1985

Name of Supervisor — Nom du directeur de thèse

Dr Jean-Claude Muller

Permission is hereby granted to the NATIONAL LIBRARY OF CANADA to microfilm this thesis and to lend or sell copies of the film.

The author reserves other publication rights, and neither the thesis nor extensive extracts from it may be printed or otherwise reproduced without the author's written permission.

L'autorisation est, par la présente, accordée à la BIBLIOTHÈQUE NATIONALE DU CANADA de microfilmer cette thèse et de prêter ou de vendre des exemplaires du film.

L'auteur se réserve les autres droits de publication; ni la thèse ni de longs extraits de celle-ci ne doivent être imprimés ou autrement reproduits sans l'autorisation écrite de l'auteur.

Date

Signature

[illegible]

THE UNIVERSITY OF ALBERTA

INTERACTIVE GRAPHICS FOR MAPPING LOCATION-ALLOCATION SOLUTIONS

by

Lise Allard

A THESIS

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES AND RESEARCH

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE

OF MASTER OF ARTS

DEPARTMENT OF GEOGRAPHY

EDMONTON, ALBERTA

SPRING 1985

THE UNIVERSITY OF ALBERTA

RELEASE FORM

NAME OF AUTHOR Lise Allard

TITLE OF THESIS INTERACTIVE GRAPHICS FOR MAPPING LOCATION-ALLOCATION SOLUTIONS

DEGREE FOR WHICH THESIS WAS PRESENTED MASTER OF ARTS

YEAR THIS DEGREE GRANTED SPRING 1985

Permission is hereby granted to THE UNIVERSITY OF ALBERTA LIBRARY to reproduce single copies of this thesis and to lend or sell such copies for private scholarly or scientific research purposes only.

The author reserves other publication rights, and neither the thesis nor extensive extra[illegible] from it may be printed or otherwise reproduced without the author's written permission.

(SIGNED) ....[illegible]..........

PERMANENT ADDRESS

..........[illegible]..........

..........[illegible]..........

..........Québec..........

DATED ..December 17th........ 1984

THE 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 INTERACTIVE GRAPHICS FOR MAPPING LOCATION-ALLOCATION SOLUTIONS submitted by Lise Allard in partial fulfilment of the requirements for the degree of MASTER OF ARTS.

....
Supervisor

....

....

Date 5/12/84

ABSTRACT

The ever accelerating evolution of knowledge and technology affected cartography in its instrumentation, methodologies and concept of map design. Less than 25 years ago computers introduced automation to the field of cartography. More recently, computer technology has promoted new interactive mapping processes which provide the cartographers with a new ability to display, analyse and integrate geographic information.

Interactive computer cartography has broadened the scope of applications and encourages research on topics which have not traditionally been assisted by map analysis. Location-allocation modelling is a good example of a field in which maps can be used to support the analysis of spatial distributions. An overview of the developments in location-allocation system mapping shows that efforts could be made to overcome a deficiency of cartographic representation in probabilistic location-allocation studies.

Based on the assumption that computer and display technology offer more possibilities than traditional cartography to map location-allocation solutions, we intend to demonstrate how interactive graphics can be used successfully in mapping solutions to probabilistic location-allocation problems. From a definition of the requirements in location-allocation system mapping and a discussion of the theoretical concepts of cartography involved, an interactive location-allocation mapping program (ILAMAP) was designed and developed to meet high cartographic standards and allow efficient data analysis. A PDP 11/70 was used in conjunction with an Intergraph System. The objectives in the program development included a study of the Intergraph's adaptability to meeting specific user-defined programming requirements, the assessment of the effectiveness of the Design File Processor Interface Subroutines (DFPI), and finally the impacts of the PDP 11/70 addressing limits of 32K.

The implementation of ILAMAP permitted to illustrate the potential of interactive graphics in the production of maps and simultaneously allowed an evaluation of the Intergraph System interface for cartographic programming applications. A series of maps illustrating the range of applicability of the program are presented and provide an example

of how cartographic analysis can become an integral part of research.

In conclusion, the results of this study illustrate new possibilities for information display and new working methods leading to solution to traditional mapping problems. Interactive graphics has been successfully used in a context of parallel interactions and encourages the pursuit of research applications which could illustrate the benefits of a complete integration of interactions for dynamic information processing and display.

PREFACE

Interactive computer graphics have been recently introduced to cartography, offering the cartographer a new ability to display, analyse and integrate geographic information. The recent acquisition of an Interactive Graphics Design System, namely the Intergraph by the department of Geography of the University of Alberta is an unprecedented occurrence in Canadian universities. The Intergraph system is an integrated configuration of hardware and software which is offered as a general-purpose turnkey graphics computer system. Although engineers are the general users in Computer-Assisted Design and Manufacturing (CAD-CAM), a variety of applications software can be integrated with the Intergraph system providing a sophisticated tool for cartographers.

The main purpose of this study is the implementation of a cartographic program in order to provide an example of the potential of interactive graphics in the production of maps. Related to this study are the evaluation of the Intergraph system for cartographic purposes and the creation of a particular type of map, flow map, for illustrating visually location-allocation solutions. An applied example is presented in which maps generated by the program illustrate how graphic displays can be used for research assistance.

The first chapter presents an historical perspective of computer-assisted cartography and the present state of location-allocation system mapping. The second chapter illustrates the methodology, analyses the major problems faced when mapping location-allocation solutions and discusses the cartographic processes involved. The third chapter describes the graphics system used, then focuses on the development of a particular cartographic program to interface with the Intergraph system, and finally stresses the problems encountered in the implementation process. The fourth chapter shows some maps as examples of the range of applicability of the program, analyses the results from a cartographic point of view and discusses the use of the Interactive Graphics Design System. The fifth chapter evaluates the user-interface for cartographic program developments on the Intergraph system and concludes with recommendations for future research applications.

ACKNOWLEDGEMENTS / REMERCIEMENTS

I would like to thank my committee members. Dr. Stan Cabay from the Department of Computing Sciences, for reviewing this thesis and Dr. John Hodgson from the Department of Geography for his precious help and invaluable assistance on the text.

My thanks go to Wade Johnson for providing me computer advice in the early stage of my thesis and grateful thanks also to Martin Scott who gave me precious recommendations and teached me a lot about computers during this project.

Je tiens à exprimer de sincères remerciements au Docteur Jean-Claude Muller directeur de cette thèse pour son encadrement et particulièrement pour la confiance dont il m'a fait preuve tout au long de cette étude.

Enfin, merci à Ronald Collette qui a travaillé aussi fort que moi sur sa thèse et m'a aidée avec une ultime complicité durant ces deux années.

Table of Contents

List of Figures

Figure | Page

List of Plates

I. COMPUTER CARTOGRAPHY AS AN AID IN MAPPING LOCATION-ALLOCATION SOLUTIONS

A. INTRODUCTION TO COMPUTER CARTOGRAPHY

An historical perspective

Cartography, as an ancient art as well as a science, has experienced considerable changes throughout the ages. The ever accelerating evolution of knowledge and technology affected its instrumentation, methodologies and the very basic concept of map design. Mapmaking can be traced back to ancient Mesopotamia and the Nile Valley. The oldest authentic map found, made out of a clay tablet, is nearly 5000 years old. In the second century A.D., Claudius Ptolemy's "Geography" describing how the spherical Earth could be projected onto flat maps, became one of the first major innovations. At the end of the fifteenth and the beginning of the sixteenth centuries, the European exploration of the Americas and the Orient generated an unprecendented need for accurate maps. The invention in Europe, shortly after 1450, of printing and engraving opened the doors for mass production. During the late 19th century, further advances in image reproduction, notably the development of offset lithography and photography, made possible easy and inexpensive duplication of drawings. The early 20th century witnessed the development of photogrammetry and photographic surveying techniques. All these innovations made maps more widely available and more accurate.

Less than 25 years ago, computers introduced automation to the field of cartography. Computer-assisted cartography as a general term refers to any aspect of cartography where the computer is used as an aid. A fundamental difference arises between an automated mapping process and computer mapping in general, however.

> "Automated mapping is defined as the automation of map making processes .... Computer mapping, on the other hand, is defined as the production of maps utilizing primarily the analytical power of the computer."[1]

According to Taylor[1], computer mapping is more desirable for the cartographer and is a more appropriate use of the computer than is mere automation.

---

[1]TAYLOR D.R.F. (1980), *The Computer in Contemporary Cartography*, vol.1, John Wiley & Sons, p.2.

The first successful attempts to produce graphics from computers were reported in the early 1950's (Foley and Van Dam 1982). Since that time, cartography has been increasingly influenced by major advances and sophistication in computer technology. Morrison (1980) defined three stages in the development of computer mapping. The first stage consisted of the rapid development of cartographic algorithms in the early 1960's. The SYMAP (Synagraphic Mapping System) Program² created by Howard T. Fisher in 1963 and developed by the Laboratory for Computer Graphics and Spatial Analysis at Harvard University, was commonly used in more than 100 institutions by the end of the 1960's and in more than 300 by the end of the 1970's. It is a mapping program using a standard line printer with a grey level display capability achieved by character overprinting. In these early days the generally reluctant cartographic discipline adopted a 'wait-and-see' attitude toward these new types of maps characterized by crude plotting on hard copy devices such as teletypes and line printers. By 1970 a second stage had arrived in which cartographers began to accept the computer's assistance for replication of hand made products. The replication quality had improved considerably since the early years of SYMAP. Important sophistication of display technologies introduced more appropriate graphic output devices. The main factor differentiating graphics output devices is the resolution, which is the number of distinguishable elements per unit of distance. The low resolution of the line printer giving unwanted visual effects such as the jagged 'staircase' appearance of lines, has been superseded by a variety of systems using plotters, matrix printers and graphics terminals. These graphics output devices, either raster or vector displays, allowed the production of more attractive final maps with higher resolution. The pen plotters provide high contrast images where different colours and line thickness can be reproduced.

This progress resulted in an increasing number of cartographic users producing new products, such as virtual maps defined as coordinates existing in machine storage or temporary maps. Riffe (1970) defined temporary maps as a cartographic CRT image portrayed for but a short period of time. Up to then, the mapping programs, such as Symap and Symvue, were developed in batch-oriented environments. By definition, batch

----

[2]FISHER H.T. et al. (1968), *Reference Manual for Synagraphic Computer Mapping (SYMAP) Version V*, Laboratory for Computer Graphics and Spatial Analysis, Graduate School of Design, Harvard University, Cambridge.

processing is a method whereby items (programs) are coded and collected in groups in a queue for later processing (Robinson, 1978, Monmonnier, 1982). This method was a response to the only hardware facilities available then and it is still used today as a low cost and high volume alternative output for standard mapping. The post-1980 stage shows a more complete implementation of computer technology and cartographers universally acknowledge the potential of the computer for the discipline.

This brief historical perspective of cartography reflects a tremendous evolution where efforts have been mainly concentrated on the improvement of map accuracy, production speed and costs. In this context, the introduction of the computer has been a real revolution although the quality of the final products has not always met high cartographic standards. More recently, attention has been directed to the map as part of a communication system. General theories of cartographic communication emphasized that one of the most important and difficult task for the cartographer remains to communicate information (Robinson & Petchenik, 1975). The latest advances in computer technology have promoted new interactive mapping processes which provide the cartographer with a new ability to display, analyse and integrate geographic information.

New approach: interactive computer cartography

The field of interactive mapping emerged from the development of cathode ray tube (CRT) terminals capable of displaying a graphic image, thus allowing maps to be edited at a terminal before being sent to a plotter for final output (Dudycha, 1981). The interactive mode is a method of operation that allows on-line communication between a person and a machine. Interactive computer cartography refers to this new approach where maps and figures are generated with computers and in which the cartographer can take advantage of an immediate interaction at both input and output levels. Input devices are used either to load graphics information into a system, or to initiate manipulations of graphic elements directly on the screen. The input devices commonly used are the typewriter keyboard, the manual and semiautomatic digitizer, the tape drive and disk storage. The common display device used is a console terminal containing a CRT. The major graphic output facilities are the printers, the plotters and microfilm. A camera used to photograph an image on screen can also be used to get hard copies.

The computer mapping literature defines an interactive graphics system as a computerized assembly with a relatively rapid response time to user s instructions re entering editing manipulating or displaying graphic data (Monmonnier 1982). It is the combination of a computer and graphic input-output devices through which visual information can be manipulated. With such systems, the cartographer takes advantages of the inherent speed with which computers can perform sequential processing and repetitive tasks, while he can exercise greater control over operations which are not easily performed in a sequential manner. Some tasks in cartography, such as placement of lettering or selective map correction require many contingent considerations. Those types of operation are difficult to automate in batch mode where they are time consuming and do not necessarily lead to the best solution. But in the interactive mode they can be easily performed using a graphic cursor on the screen display. Therefore, interactive mapping appears more efficient when use is made of the full ability of the computer for the processing of repetitive sequential operations while complete control is allowed over specifications of map layout and design. Interactive computer cartography has definite advantages over conventional non-interactive, or batch systems. It offers the flexibility of combining manual and automated methods. For applications in which the output is essentially of a graphical nature, such as those encountered in cartography, there is a strategic advantage. Information can be manipulated instantaneously on a visual basis, any cartographic stage can be isolated and reviewed independently, and final results can be immediately examined through experimentation with different ways of representing geographic information.

Since the advent of automation in cartographic information process, significant strides have been made, both in graphics hardware and mapping software. The development of interactive graphics systems is the latest major innovation influencing cartography. According to Stutz (1975) the barriers to interactive cartographic systems have always been primarily high cost and software unavailability. Fortunately, interactive graphics is now more feasible because of the rapidly decreasing cost of hardware. However, software availability remains a critical problem because there are still very few truly cartographic-oriented computer systems in operation. Display and treatment techniques must usually be supplied by the user in spite of duplicated effort, because there is a general lack of professional communication between institutions working on the same subjects or because different systems are used (Brassel, 1977). This results in the problem of software incompatibility. When software is nonportable, moving to new display devices necessitates very expensive and time-consuming rewriting of working programs (Foley & Van Dam, 1982). Moreover, a lack of standardization (as to program language, computer installation, user access etc.) has caused needless delay in the full realization of the potential of automated thematic cartography (Muerchke, 1972).

Computer mapping techniques have been used in a wide range of applications in many different areas of industry, business, government and education. Interactive computer cartography has broadened the scope of applications, introducing new possibilities for information display and new working methods, perhaps leading to solutions to traditional mapping problems. It encourages research on topics which have not traditionally been assisted by map analysis. Cartographers could beneficially concentrate their work in applied field research where there is an obvious need to provide an effective means of displaying information for decision-making. This could enhance computer mapping's invaluable attributes as an aid in research applications such as those related to the analysis of spatial distributions. Location-allocation modelling is a good example of a field in which maps can be used to support the analysis. In a situation of alternative solutions to a specific location-allocation problem, maps could clarify the interrelationships between the distribution of facilities and the allocation pattern.

B. DEVELOPMENTS IN LOCATION-ALLOCATION SYSTEM MAPPING

Location theory made its early developments in the 19th and early 20th centuries with Von Thünen, Weber and other economists. It is of significant interest and importance in economic geography, spatial economics and regional science. Since 1957, research has concentrated essentially on operational normative models seeking the optimal location of discrete facilities. The computer revolution has allowed tremendous progress in the development of optimal location techniques, including location-allocation methods. Locational models have become important tools in planning and management and are actually considered as effective aids to decision-making problems.

As defined by Lea, location-allocation problems in their general form, have the following features "a number of discrete facilities providing one or more goods or services is to be located in geographic space to optimize an objective function subject to constraints. In addition to finding a system of locations it is required to optimally allocate a set of spatially distributed users, customers, or suppliers to the facilities"(Lea, 1978). The problem becomes simultaneously the location of central facilities and the assignment of flows, such that the total costs of operation are minimized.

A very large literature of development in location-allocation systems has grown over the last 25 years. However it deals almost exclusively with computational techniques and formal theory. I shall not review the extensive literature on the subject because my interest is the cartographic representation of these spatial distribution systems. Solutions to location-allocation problems seem appropriate to cartographic representation, but in fact the location-allocation field has relied very little on map analysis. In personal communication, Lea mentioned that very little work has been done on the cartographic representation of location-allocation problems or solutions. Rushton stated that there is a lot of scope for truly interactive (man-machine) modelling in this area where graphics would play a major role. He added that their research group[3] is about to publish two monographs on location-allocation topics but neither involved graphics considerations.

The first attempt to automate location-allocation maps is the work of Kern and Rushton, in 1969, with a computer program for production of flow maps, MAPIT.[4] Although the cartographic representation of location-allocation systems has generally been given very limited attention in the literature, some developments are to be expected in a near future making use of the computer's ability to generate graphic displays rapidly. Computer mapping can be [illegible] to display actual flow data and provide means of visualizing the outcome of lo[illegible]-allocation simulation. Changes in flow pattern in response to relocation of facili[illegible]ould thus be made obvious, thereby allowing more efficient assessment of alternative [illegible]tions.

----

[3]McNULTY Michael L. and RUSHTON Gerard, Institute of Urban and Regional Planning, University of Iowa.

[4]KERN R., and G. RUSHTON (1969), "MAPIT: A Computer Program for Production of Flow Maps, Dot Maps and Graduated Symbol Maps", *The Cartographic Journal*, vol.6, pp.131-137.

Two natural stages of development in the conceptual and theoretical framework of location-allocation models must be differentiated in order to illustrate the actual achievements of this discipline in relation to the cartographic representation of solutions. The classical least-cost allocation approach characterizes the first developments in the field, and some recent works introduce a more realistic approach based on spatial interaction models.

Classical location-allocation approach

The general framework of a location-allocation problem consists of two subproblems. The first is the problem of facility location. The second is the problem of allocation of demand between the patron locations and the service facilities. The classical location-allocation approach, the P-Median, is based on two assumptions:

1. Patrons are best served if the distance they must travel is minimized.
2. Patrons will invariably employ the facility which is closest to them.

The P-Median model implies mathematically the least-cost travel assumptions. It solves the location of "P" facilities, minimizing the aggregate distance between patrons and the nearest facilities to them. This classical model ensures that each demand area is allocated to the closest facility and is assigned to one and only one central facility, the so-called "all or nothing" allocation. The fundamental location-allocation problem treated is therefore an optimal partitioning-type problem. Two major cases can be distinguished: the case of partitioning a point set in continuous space, and the case of partitioning a point set in discrete space. In continuous space, the problem is solved for all locations whereas in discrete space predefined locations are selected.

The main elements of solution to classical location-allocation problems have been traditionally portrayed on maps by point symbols for the facilities' locations and by desire lines for the allocations. This graphic representation of allocations of demand points to a single facility gives typically a "starbust" outlook and is called "spider diagrams". The number of patrons is usually represented by a separate choropleth map.

A limited number of attempts have been made to design and write computer programs producing maps of such solutions to classical location-allocation applications. Among those, we find MAPIT, a computer program for production of flow maps, dot maps and graduated symbols maps, developed in 1969 by Kern and Rushton.[5] It is a batch-oriented program, written in FORTRAN, which can produce outline maps, flow maps, graduated symbols maps and dot maps on a CALCOMP plotter. Figure I.1 reproduced from Kern and Rushton (1969), shows a flow map. As they described in this example, the asterisks indicate the locations of households interviewed in a consumer spatial choice survey; the business centres are located with circles proportional in area to the number of activities present, and the flow lines show places actually patronised by the households for a particular purpose. MAPIT is intended to produce flow maps but as illustrated in this example, it actually draws lines which do not refer to any flow data. By definition a flow map is a representation of quantitative data showing movement in space. Thus showing flows by uniform sized lines, such as MAPIT does, is inadequate as these are not flows in the real sense. Although in this particular case map legibility needs to be improved, it remains that this work was influential to the subsequent role of computer mapping in location-allocation applications.

Ten years later, ILACS (Interactive Location-allocation in Continuous Space) was developed by Goodchild and Rizzo.[6] The program is designed to use the capabilities of interactive storage tube graphics to solve location-allocation problems in a wide range of applications of interest to the public and private sectors. It is an interactive program. The user has access to a number of commands which permit displays at the terminal and manipulation by means of a cursor. On request, the user can obtain hard copy reports including the solutions' descriptions summary tables, final reports, and plots of maps. Examples of the maps are shown in Figures I.2 and I.3.

SPYDER, a FORTRAN program for creating maps of solutions to location-allocation problems, developed in 1983 by Charest-Berglove and McKeagney is intended for the cartographic display of point and flow-type data within a specified study area.[7] It is a batch-oriented program. Two types of map, vector maps and symbol maps, can be generated. The vector maps represent the flow-type data by plotting vectors from an origin to one or more endpoints (example Figure I.4). The resulting cartographic display is

----

[5]KERN R., and G. RUSHTON (1969), *Op. Cit.*
[6]GOODCHILD M.F., and Brian RIZZO (1979), *ILACS Documentation*, Department of Geography, The University of Western Ontario, 16 pages.
[7]CHAREST-BERGLOVE D., and Don McKEAGNEY (1983), *SPYDER-A FORTRAN Program for Creating Maps of Solutions to location-allocation Problems*, Institute of Urban and Regional Planning, University of Iowa, 22 pages.

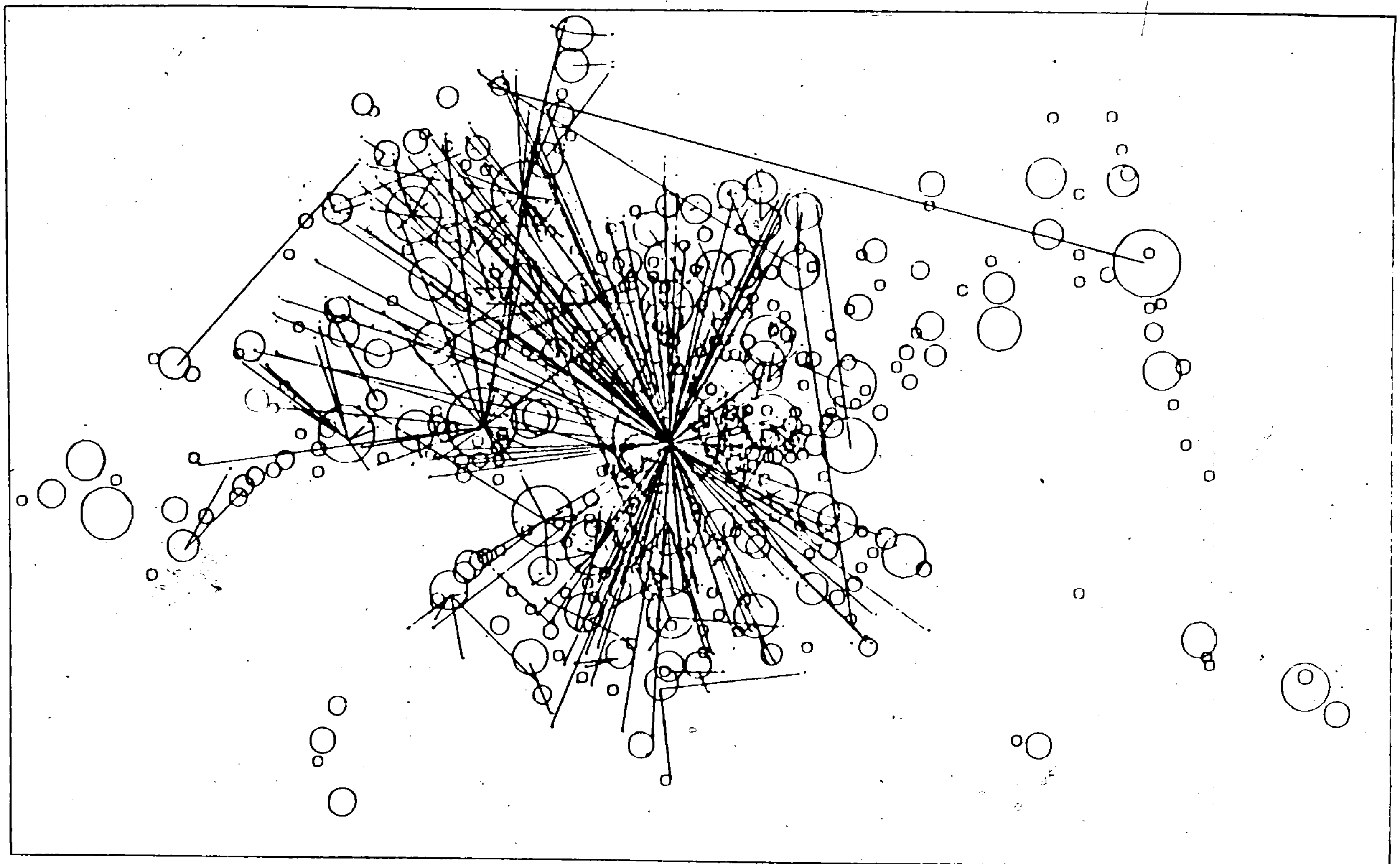


FIGURE I.1 A representation of flow map with graduated symbols produced with MAPIT. Figure reproduced by permission of the Editor from Kern R. and G. Rushton (1969), "MAPIT: A computer program for production of flow maps, dot maps and graduated symbol maps", *The Cartographic Journal*, vol.6, page 133.

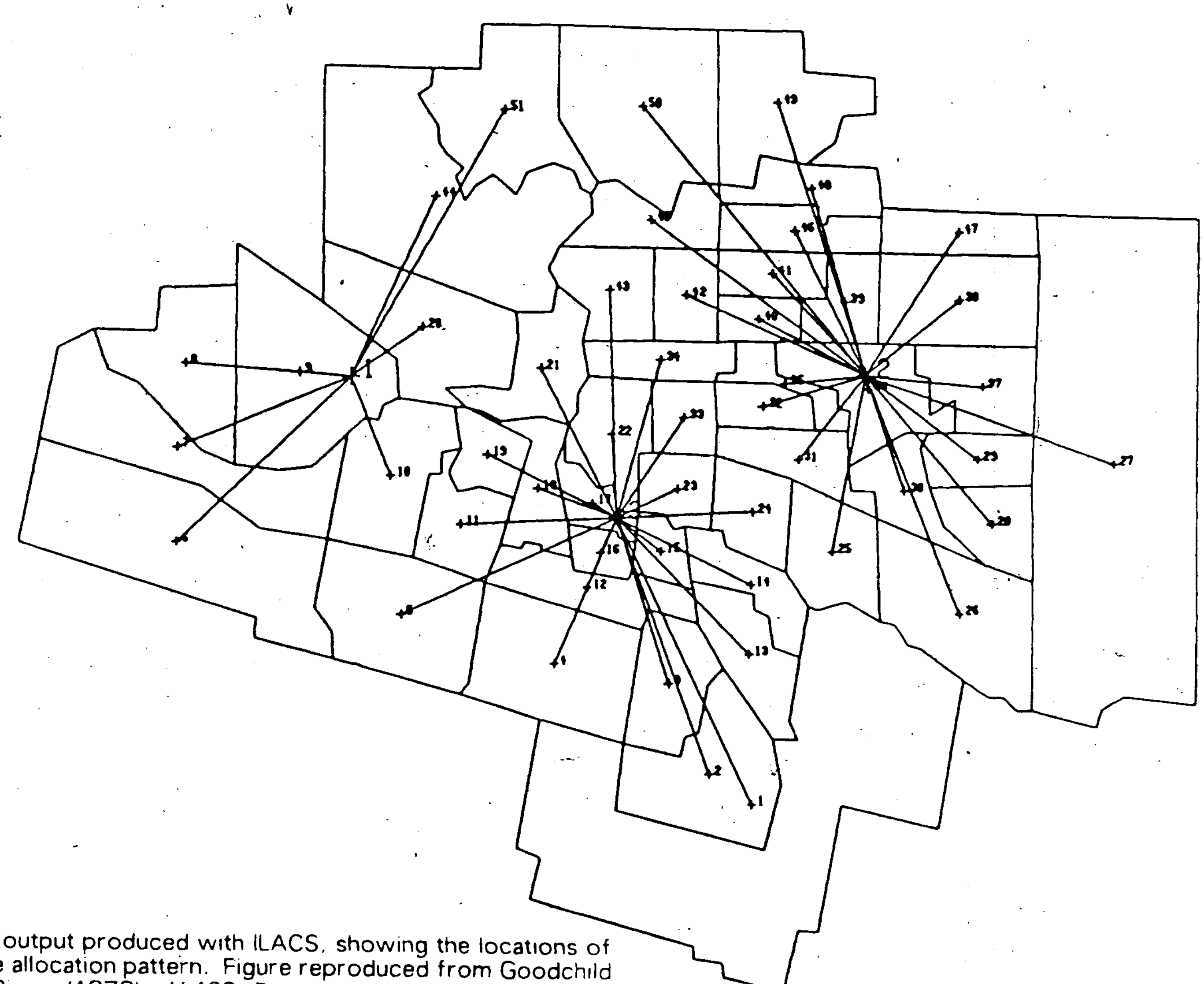


FIGURE I.2 A standard plot output produced with ILACS, showing the locations of facilities and the allocation pattern. Figure reproduced from Goodchild M.F. and B. Rizzo (1979), *ILACS Documentation*, Department of Geography, The University of Western Ontario.

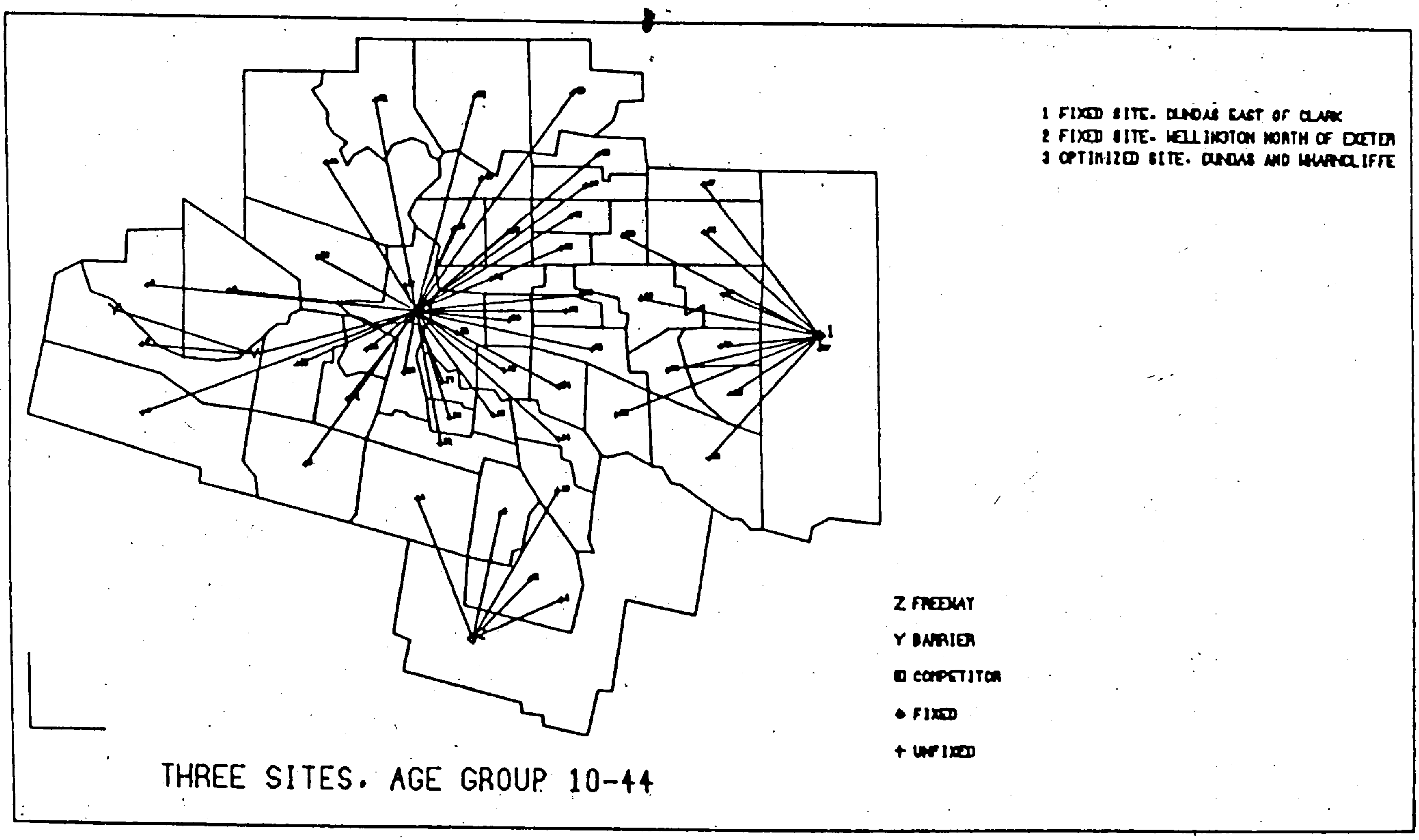


FIGURE I.3 An example application illustrating a final output map produced with ILACS. Figure reproduced from Goodchild M.F. and B. Rizzo (1979), *ILACS Documentation*, Department of Geography, The University of Western Ontario.

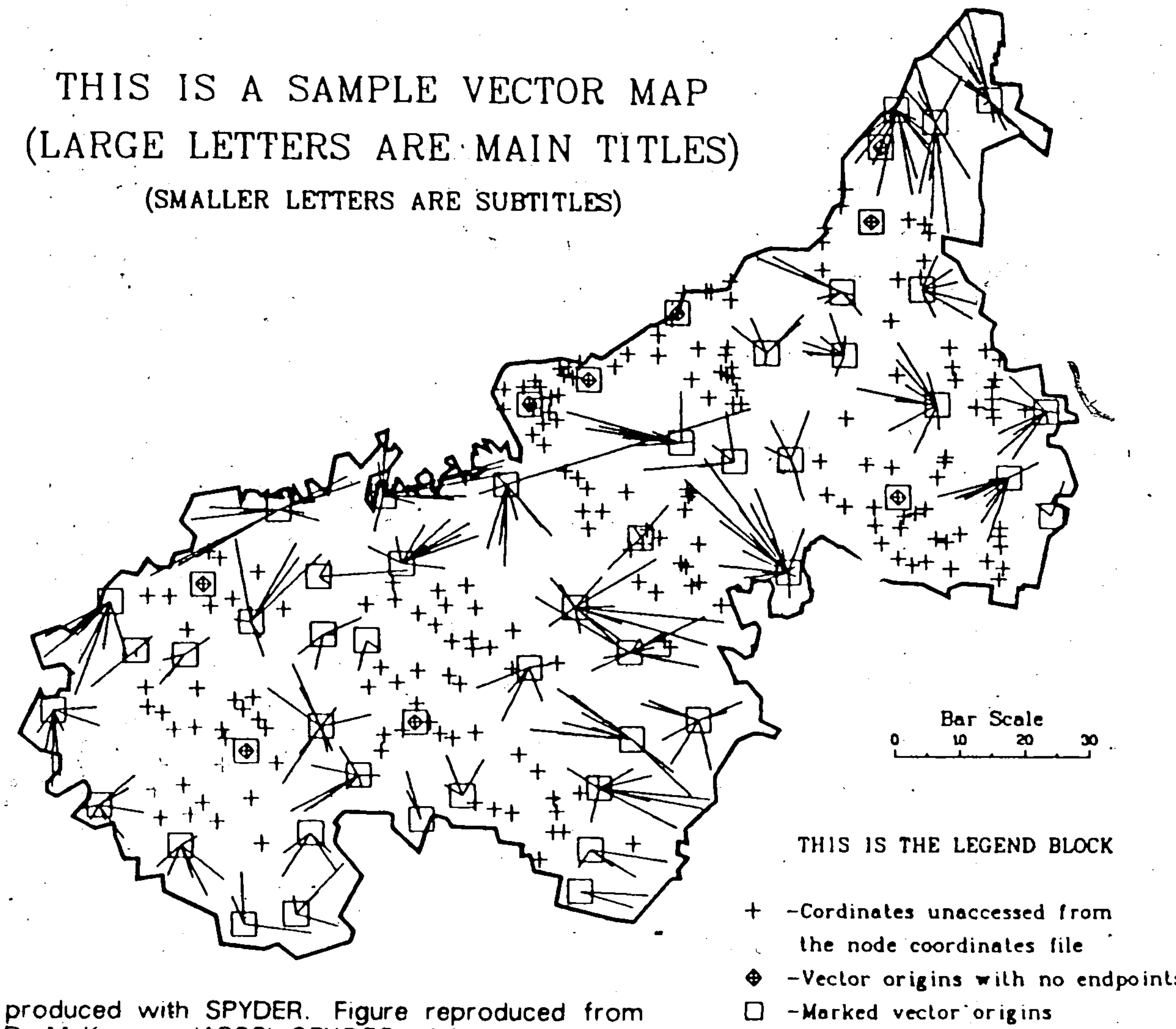


FIGURE I.4 A sample vector map produced with SPYDER. Figure reproduced from Charest-Berglove D. and D. McKeagney (1983). *SPYDER - A fortran program for creating maps of solutions to location-allocation problems*, University of Iowa.

nominal in that no degree of relationship is implied between points. Figure I.5 illustrates an example of the so-called "spider diagrams" on a map. The symbol maps are used to display location attributes. SPYDER offers three possibilities: the point-symbol map, the graduated-symbol map and a combination of the two. In the point-symbol map plots (as in Figure I.5), the symbols are selected from 15 predefined symbols which indicate values of corresponding discrete location attributes based on nominal or ordinal data. The graduated-symbol map displays interval data by plotting graduated symbols according to the value of a locational attribute. The symbol sizes are scaled by the square root of the value of a second attribute (example Figure I.6). A third symbol map may be obtained from a combination of the two previous maps.

Spatial interaction approach

Although the classical least-cost allocation paradigm is commonly used, more recent works have introduced probabilistic allocation. "In the probabilistic allocation model demand can be shared between equally distant units while, in deterministic 'all' or nothing' allocation models, ties are usually broken arbitrarily" (O'Kelly and Storbeck, 1984). These models are based on the so-called gravity, or spatial interaction assumption. The main feature of a spatial-interaction model is to replace the very sharp assumption of travel-cost minimizing with the smoother assumption of user's behaviour (Leonardi, 1981). The costs minimizing approach is measured in terms of efficiency, equity and fair distribution of service to users. Spatial interaction theory suggests that a variety of factors, such as differential facility attractiveness, and not only the least travel cost, will potentially influence a person's travel behaviour (Hodgson, 1978). Conceptually, the allocation rule in the interaction-based modelling allocates patrons to facilities in accordance with spatial interaction patterns and maximizes a welfare function based upon their revealed travel preferences (Hodgson, 1981). Current models built to accomodate different types of application, such as person movement or goods movement of all kinds, are used to make predictions and simulations of behaviour upon selected constraints. A family of spatial interaction models each identified by the nature of the constraints treated is presented by Wilson (1971).

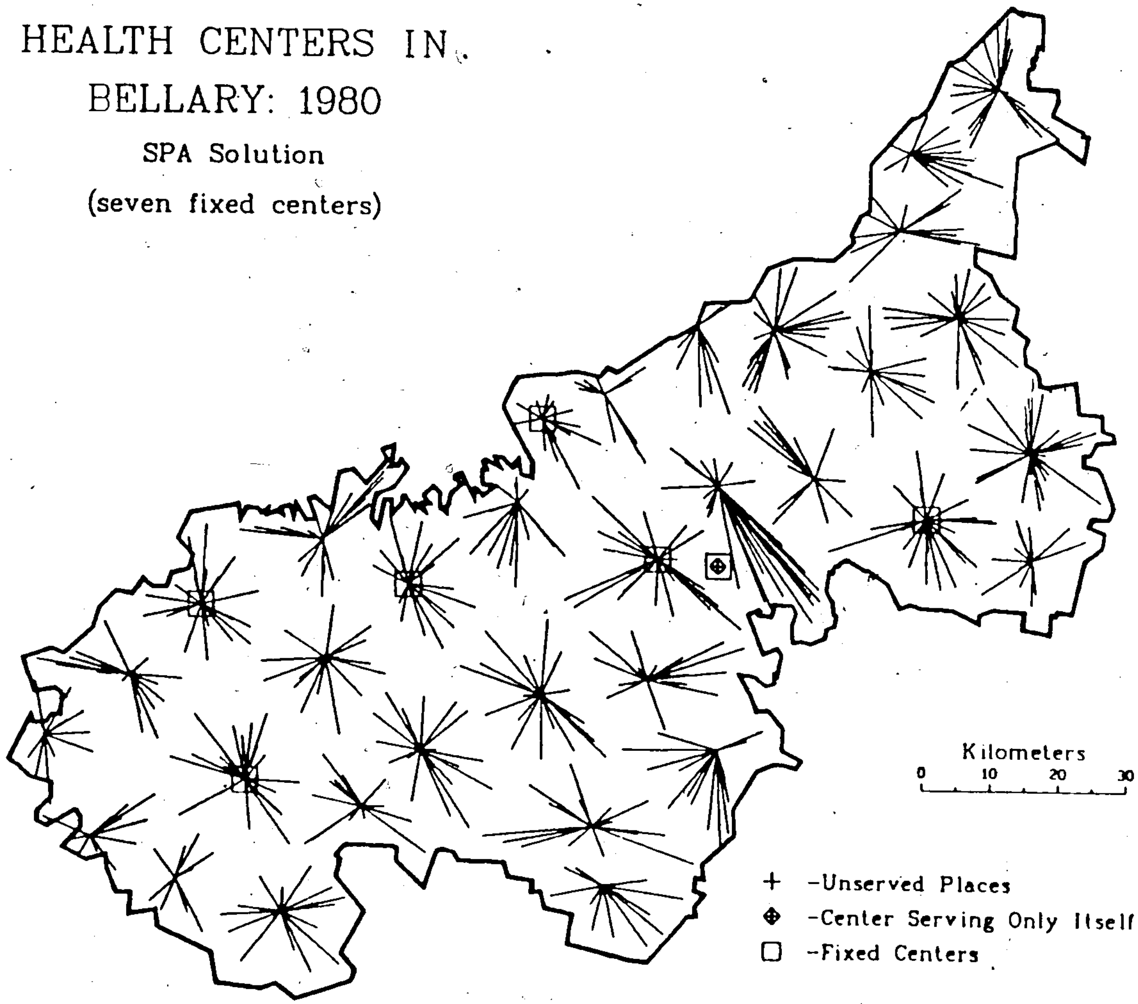


FIGURE I.5 An example application produced with SPYDER. Figure reproduced from Charest-Berglove D. and D. McKeagney (1983). *SPYDER - A fortran program for creating maps of solutions to location-allocation problems*, University of Iowa.

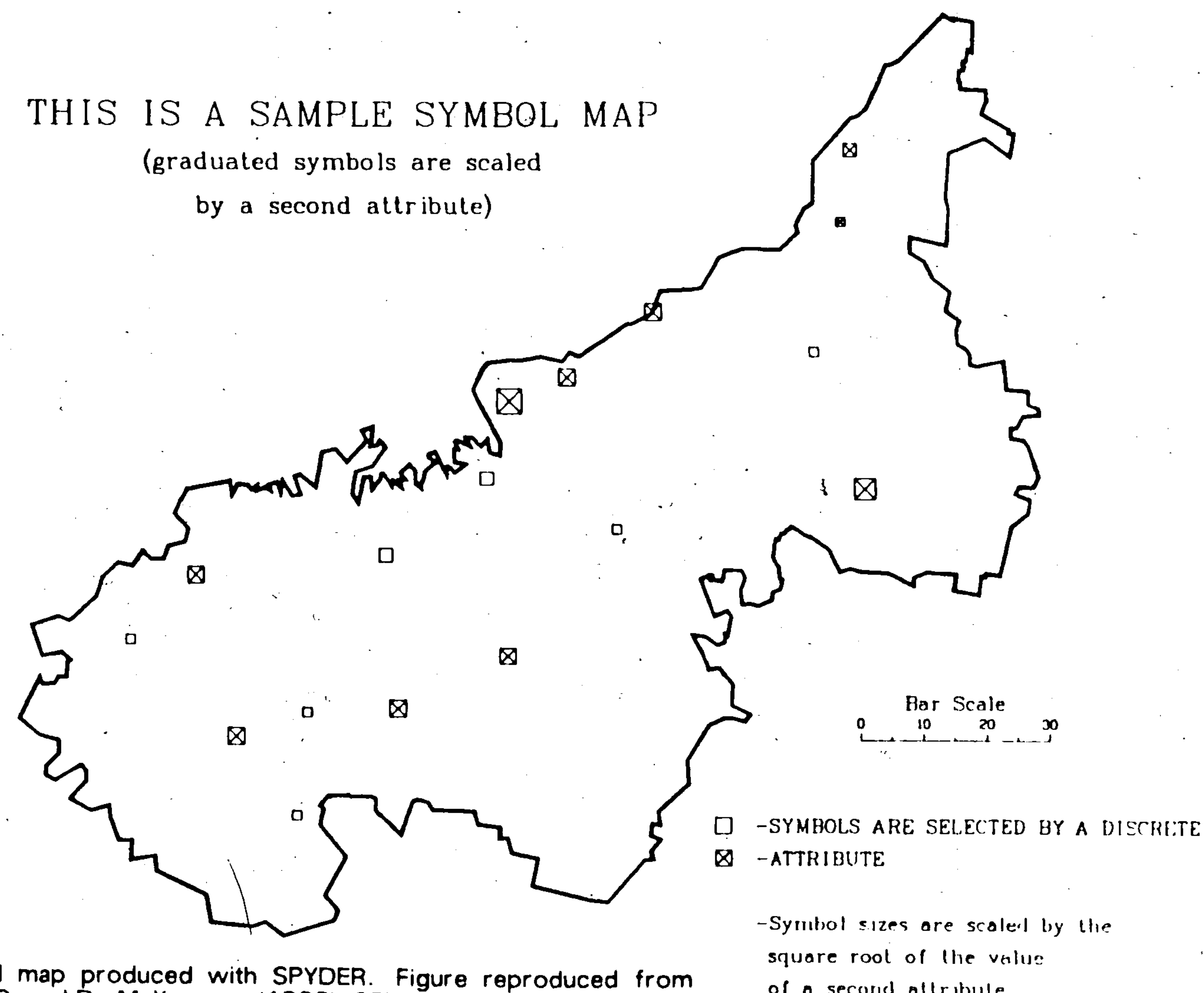


FIGURE I.6 A sample symbol map produced with SPYDER. Figure reproduced from Charest-Berglove D. and D. McKeagney (1983). *SPYDER - A fortran program for creating maps of solutions to location-allocation problems*, University of Iowa.

Apart from technical details, the spatial interaction approach is new. Its theory is based on a more realistic allocation where various proportions of the demand at demand points are allocated to different facilities. For such probabilistic allocations, a cartographic representation of solutions would have to involve point symbols for the facilities' locations and proportional flow-line symbols to portray the allocations.

Applications making use of spatial interaction models have appeared in regional science and urban geography literature, however cartographic representation of solutions in most studies are nonexistent. A typical example is the study conducted by Hodgson (1981) in which solutions of a spatial-interaction location-allocation model maximizing consumers' welfare are compared with the P-Median solutions. In this study the P-Median solution is cartographically portrayed with a standard spider diagram image but the consumers' welfare solution is illustrated by a map showing the facilities' locations and a table of numbers describing the proportional allocations. When treating a complex spatial relationship, the production of a legible map becomes a difficult task. Moreover the cartographic problem of maximizing information flow without losing accuracy must be solved because no one can visualize solutions to location-allocation problems in the form of large tables of numbers. As Goodchild mentioned in a personal communication, the locations can presumably be shown by point symbols, however the allocations may create problems, particularly if the allocation rules are complex, as when attendance behaviour is described by a spatial interaction model.

No attempt has been made to propose a general approach to mapping these applications. The work of Tobler deserves to be mentioned here, even though it is not related to precisely the same type of application. He designed in 1979 a "Geographic Flow Mapping Program" intended for the representation of proportional flows.[1] In this work, a considerable effort has been devoted to producing maps of publishable quality since this is the main purpose of academic computing (Tobler, 1979). An architectural display program (ARCH2D) has been used for overlay deletion, providing better graphics results. This program was initially implemented in a batch-oriented environment, although in a subsequent phase, use was to be made of interactive graphics software and display techniques.

----

[1]TOBLER W. (1979). *A Geographic Flow Mapping Program*. University of California.

This overview of the developments in location-allocation system mapping shows that efforts could be made to overcome this deficiency of cartographic representation in probabilistic location-allocation studies. Based on the assumption that computer and display technology offer more possibilities than traditional cartography to map location-allocation solutions, we intend to demonstrate how interactive graphics can be used successfully in mapping solutions to probabilistic location-allocation problems. The ability to graphically display data in map form for practical applications of location-allocation models thus complementing numerical data would be a major step toward a better assessment of solutions. Attention was given to meet high cartographic standards and to allow efficient spatial relationship interpretation. As Kern and Rushton mentioned in 1969: "The quality of the product is in the hands of the programmers and, given sufficient attention to detail, it is possible to produce maps that meet rigorous cartographic standards."[9]

---

[9]KERN R. and G. RUSHTON (1969), *Op.Cit.*, p.135.

II. METHODOLOGY AND PROBLEM DEFINITION

A. METHODOLOGY

The Intergraph system, as described by its designers, is an automated drafting and mapping system which can be used easily by noncomputer oriented personnel, as well as by computer programmers who may address the development of application programs to satisfy future needs (Intergraph Co., 1981). Its unique configuration, described in the next chapter, although claimed by its designers to be easy to use, must be evaluated from a cartographic point of view. Giving full credit to the Intergraph system and its designers, it remains important to the field of computer-assisted cartography to study how successfully the Intergraph s man-machine interface can be used in the development of programs for mapping applications.

Considering that any man-machine interface evaluation is intimately related to the use made of the system, the resulting evaluation has to be considered as qualitative and highly related to the actual needs of the people involved with the system. In other words, a system is to be evaluated in its actual study s environment. The Intergraph system together with its operation rules and constraints were interacting with the cartographic requirements involved in location-allocation system mapping and with the programmers' and users' needs.

The methodology followed is illustrated in Figure II.1. A survey of the literature has indicated the lack of cartographic representation in location-allocation studies and illustrated the need to study the use of interactive computer cartography in relation to this practical application. From the definition of requirements in location-allocation system mapping and a discussion of the theoretical concepts of cartography involved, an interactive computer program for mapping location-allocation solutions was designed and developed to meet high cartographic standards and allow efficient data analysis. The implementation of this cartographic program demonstrated the potentials of the Intergraph system for cartographic programming applications. Finally several examples of application are presented to illustrate the range of applicability of the program, to allow a cartographic evaluation of the results and to lead to a general assessment of interactive graphics communication combined with alphanumeric man-machine communication.

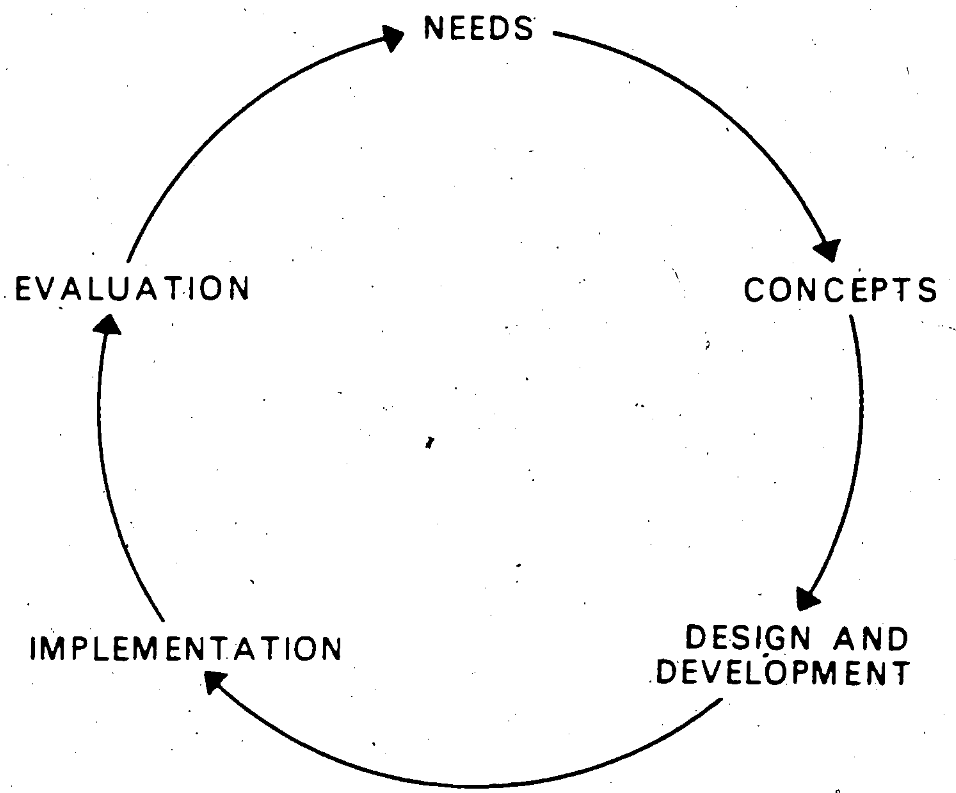


FIGURE II.1 Methodology.

The conceptual framework

The conceptual framework, on which the application study is based, is shown schematically in Figure II.2. The programmer's model of interactive graphics (Foley & Van Dam, 1982), used as a conceptual framework, is appropriate for applications involving data manipulation in order to create a graphic image. The model is useful in understanding all the steps required to generate a picture from numerical coded information. The development of user-written application software involves in this case the handling of three major components. The first component consists of the application data or data base. The data are retrieved by the second major component, the application program which generates a picture and sends graphics commands to the third component, the graphics system itself.

Consequently, the actual application study on the Intergraph system refers to the following components. The application data represent, in a numeric format, the solutions of location-allocation problems as stored after the execution of specific location-allocation programs. The application program consists of a collection of Design File Processor Interface Subroutines (DFPIS) provided by Intergraph and compatible with map data processing operations written in Fortran. The program retrieves the user-supplied input data, and as part of an interaction sequence, makes use of the graphics system to create the map. The Interactive Graphics Design System (IGDS) is initiated at the display unit to review or edit the map interactively on the screen.

In order to proceed to the accomplishment of the application study it is necessary to define the basic requirements and discuss the cartographic processes involved in mapping location-allocation solutions.

B. PROBLEM DEFINITION

Location-allocation system mapping

As illustrated in the Chapter 1, the classical solutions of location-allocation problems are represented cartographically in most studies whereas no one has attempted to portray the results of spatial-interaction models. In most cases, these models attempt to describe patterns of movement in space. In order to represent cartographically the

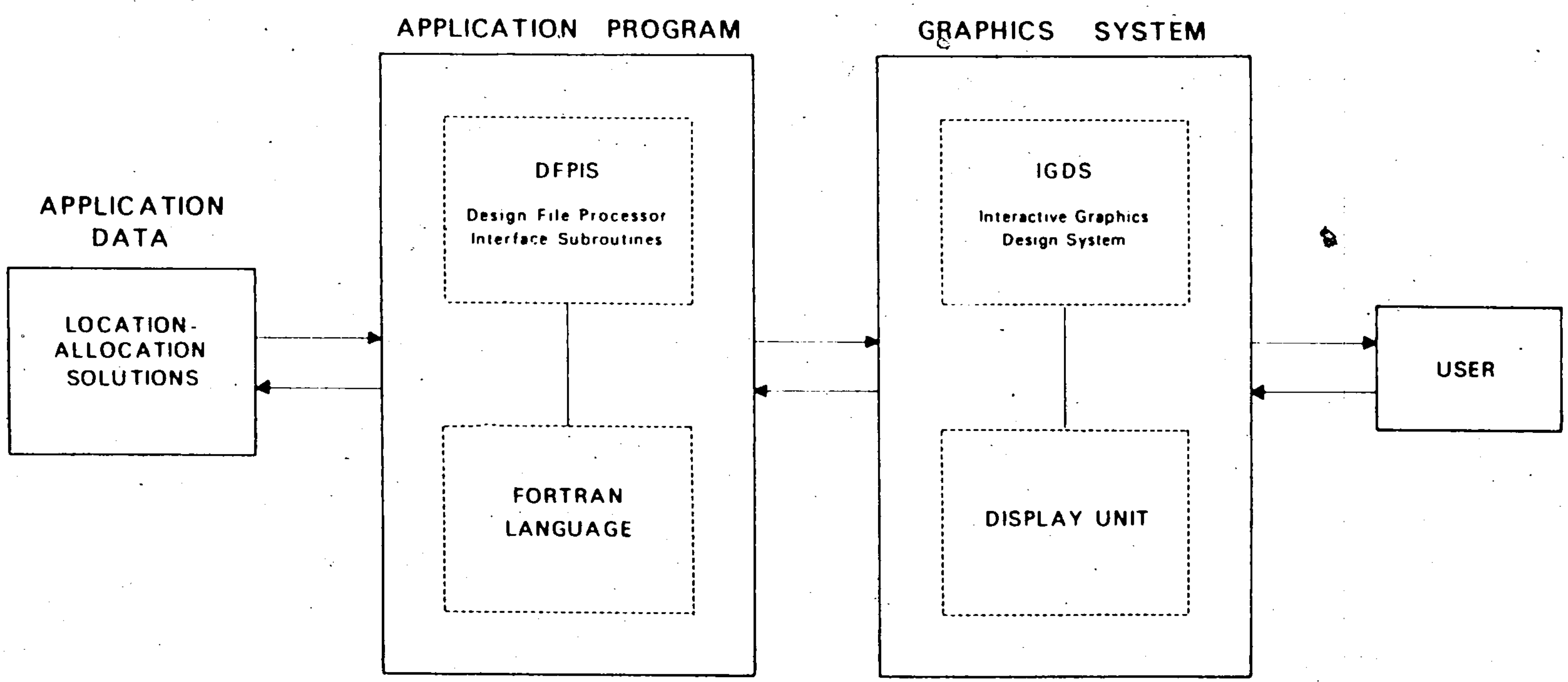


FIGURE II.2 Conceptual framework.
Adapted from Foley and Van Dam (1982), page 24.

solutions of location-allocation problems, it is important to portray the initial data, intermediate solutions and the final 'optimal' result.

We defined the main elements to be mapped in probabilistic location-allocation as follows :

1. The users, or patrons, in each demand area.
2. The facilities' location and attributes.
3. The allocation pattern.

Graphically, the patrons are best represented by proportional circles. The allocation pattern is here exclusively expressed in percentage of patrons attending each facility and is portrayed by proportional flow-line symbols. The absolute values are not used because of their unmanageable range of distribution. The facilities' locations and attributes are shown by point symbols.

Depending on the problem's size and parameters, the allocation patterns can reach a considerable level of complexity. This is basically where major difficulties arise and where traditional cartography has always failed. Because the realisation of a legible map for this spatial phenomenon is very time-consuming and involves a series of repetitive tasks based on a trial and error approach, a manual approach becomes virtually impossible. For example, a study area composed of 30 zones where 5 facilities are located, generates a potential allocation pattern of 150 movements in space. These movements or attendance patterns are illustrated by flow lines stretching from demand zones to the facilities' locations with different line width associated to the relative percentage of patrons attending each facility. Even without including the absolute number of patrons in each demand zone, in the form of proportional circles, to actually portray the importance of each zone in the attendance pattern, the graphic becomes so complex that the map cannot possibly answer the question of "who is going where?".

Different techniques can be applied to the map legibility problem, such as generalisation (classification) or the production of a collection of maps to cover the large number of interactions. Although these techniques can be easily computerized, they both present limitations in the map interpretation. Whereas the classification process may involve a loss of information or some data distortion, the collection of maps offers exhaustive data perhaps more difficult to integrate mentally. As defined by Muerchke

(1972): "The cartographic process is viewed as a series of transformations involving the selection of data from real world, the transformation of the data into a graphic map, and the retrieval of information through an interpretative map reading process". In general, in order to represent the spatial configuration of a location-allocation solution, the consideration of some form of data simplification as well as symbolization is mandatory.

Discussion of the cartographic processes involved

Maps have visual information processing as their ultimate objective. The visual impression given by a map is a function of data quality, data representation, map scale and graphic design. Theoretically, in most cases the cartographer summarizes or simplifies the data. This is usually achieved by dividing the data set into meaningful categories or classes. Different methods[10] can be employed and each results in a different representation. Robinson (1978) gives us a definition of the classification process: ,

> "Classification is a standard intellectual process of generalisation that seeks to sort phenomena into classes in order to bring relative order and simplicity out of the complexity of incomprehensible differences, inconsequential differences, or the unmanageable magnitudes of information."[11]

Robinson's definition would be more complete with the acknowledgement that classification is also a process of generating meaningful information from a complex data set.

The discussion of this process reflects a lack of agreement among cartographers. Although the process of data simplification is often limited to choropleth maps (i.e. maps of areas), the problems and procedures involved apply equally well to point or flow-line symbol maps. The problem of selecting class intervals involves the subjective stage of determining the optimal number of classes and the range of the class intervals best suited to representing the data distribution in an unbiased manner (Evans, 1976). The choice of class limits is a critical decision because it alters substantially the visual impression, hence the map's interpretation. Up to now, no universal "correct" method has been accepted among cartographers (Monmonier 1976, in Brassel and Utano 1979).

----

[10]Robinson (1978) suggested the following: equal steps, the use of mean and standard deviation, quantiles, equal area steps, arithmetic, geometric, reciprocal and graphic techniques.
[11]ROBINSON Arthur (1978), *Elements of Cartography*, 4th edition, p.152.

Traditional mapping approaches for selecting class limits are based on either natural breaks or arbitrary divisions. The number of classes would rarely exceed ten or fall below four. Jenks and Coulson (1963) demonstrated the consequences of manipulating mappable data in various ways. They maintained that classification is necessary since its outcome controls map interpretation. Holding similar views on the desirability of limiting the number of classes, Robinson and Sale (1969), Jenks and Caspall (1971) and Dobson (1973) argued that classification helps regionalization and thus contributes to the effectiveness of traditional choropleth maps. With the introduction of automation, iterative techniques have come into widespread use by cartographers. Based on a solution proposed by Fisher[12], Jenks (1977) developed an automated procedure for optimizing class limits for univariate distributions with a finite number of classes, usually not more than seven.

The introduction of non-classed maps by Tobler (1973) has stirred considerable controversy. Tobler suggested the use of the non-classed choropleth maps, arguing that they avoid information loss through classification and eliminate distortions from inappropriate choices of class limits. He pointed out that it is relatively easy to generate a density of shading or a range of sizes according to some rule of proportionality, with a computer program.

Most perceptual studies regarding the relevance of non-classed maps have dealt with areal symbolization such as choropleth mapping. Limited research has been directed toward circles and flow-line symbols maps. The traditional approach in choropleth mapping has been to group the information for a regionalization of the mapped variable, whereas the approach for graduated circles maps has been to scale the circles uniquely in order to symbolize the actual amounts at each location. The results of some experiments conducted by Meihoefer (1973) suggest that the traditional and popular method of comparing and presenting quantitative data by using continuously graduated circles should be modified. Instead, the use of range-graded circles which indicate only the group or category that applies to each location is more useful and effective, and should be employed at all times. Therefore, the analogy in visual perception between the classification in choropleth maps and the question of scaling symbols either literally or

----

[12]FISHER Walter D. (1958), "On Grouping for Maximum Homogeneity", *Journal of the American Statistical Association*, Dec.1958, pp.789-798.

according to grouped values is very strong (Cuff and Mattson, 1982). The discussion of the arguments presented for and against class generalisation refers mainly to theoretical concepts of cartographic communication.

Among the arguments in favor of classification is that the human eye s inability to distinguish gradations weakens the relevance of a continuous shaded map (Jenks 1977, in Muller 1979). Also, classification is intended to regionalize the mapped variable and consequently to provide a better transfer of information to the map reader (Monmonier 1977, in Muller 1979). The strongest argument against unquantized maps remains their visual complexity.

Cartographers opposed to classification maintain that the map without class intervals is graphically accurate, in other words is a "true image" of the distribution. Considering that a map is made to be seen, not read (Bertin, 1977), the purpose of the thematic map is to create information about relative intensities more so than about absolute values; in the latter case, a table of numbers would be more appropriate than a map. It is to be expected that the unquantized map would not leave the cartographer in full control of the message being conveyed. In fact, classification might be necessary in order to stress a particular threshold value of a distribution.

Although quantized and unquantized maps are different in nature, they both function as communication devices in their own right (Muller, 1979). The conventional classified map still appears to be the first choice of the majority of cartographers, being considered generally more legible. The conclusion of that debate seems to be that each type of map has its particular advantages. The final decision remains in the cartographer's hands and should be based on the characteristics of the data set as well as the intended audience.

After any classification has been applied to the data or after a non-classed map is created, the cartographer must still decide on how the phenomenon is to be mapped. In other words, the process of symbolization is needed to reflect positional, linear, areal or volumetric information.

"Symbolization is the graphic coding of the summarizations resulting from classification and the coding of the essential characteristics, comparative significance, and relative positions that result from simplification."[13]

The symbolization of quantitative data necessitates the use of a system, usually arbitrary, that symbolizes the variations in the quantities. It is important to distinguish that absolute quantities gathered for areas, such as numbers of persons or numbers of housing units, do not apply to the whole area in the same sense that density measures or other ratios do. In such cases, it is recommended that area tones not be used but rather that spot symbols be (Cuff and Mattson, 1982; Muller, 1983). Variation in the quantities is, therefore, symbolized by variation in the size of the symbol applied to the areas. The graduated circle is most commonly employed for its ease of construction. When quantitative data occur on lines, such as flows or interactions between locations, the mode of symbolization typically used is to vary the width of line according to the size of flow (Cuff and Mattson, 1982; Campbell, 1984).

Among the various methods of scaling graduated symbols [14], such as circles, the psychological scaling method is the most suitable for cartography. Studies have demonstrated that the perceptual response to differences between symbol areas is not a linear function; rather the ordinary observer will underestimate the sizes of the larger symbols in relation to the smaller ones (Flannery, 1971; Robinson, 1978). Flannery (1971) has devised a procedure for graduated circles that compensates for such underestimation. The procedure is best expressed as a modification of the square root calculation. The use of a power of 0.5716 instead of the square root (equivalent to a power of 0.5) adjusts progressively the size of the larger circles so that they are more than proportional to the data values they represent. On the other hand, the definition of proportional line widths does not seem to need any correction for psychological bias (Muerchke, 1972). "In general, the construction of flow lines is a relatively simple task that is made even easier because human visual response to the varying widths of flow lines seems to be quite accurate and thus does not require scaling adjustments" (Robinson *et al*, 1978). Until research shows that there is a need for correction, it seems appropriate to adopt the traditional linear graduated scale. The width of the line is made in linear proportion to the flow so that the reader may properly interpret the relative values.

----

[13]ROBINSON Arthur (1978), *Op. Cit.*, p.153.

[14]Robinson (1978) suggested the following: proportional area scaling or the square root method, the range-graded scaling method and the psychological scaling method.

It should perhaps be pointed out that maps must be designed and not just arbitrarily thrown together. The function of a map as communicative device is to provide the reader with a graphic display of information in a format such that it promotes a conceptual relationship with spatial arrangements. The message of a map is communicated to the reader by a process of visual integration in which data classification and symbolization contribute to a manipulation of the map message. This discussion leads to the next chapter in which the mapping program structure and content as well as its implementation on the Intergraph system are presented.

III. DEVELOPMENT AND IMPLEMENTATION OF A CARTOGRAPHIC PROGRAM ON THE INTERGRAPH SYSTEM

This chapter starts with a brief description of the Interactive Graphics Design system used, introducing the context of this study and familiarizing the reader with a technical vocabulary. It then focuses on the design and implementation of the computer mapping program.

A. THE INTERACTIVE GRAPHICS DESIGN SYSTEM

System description

The Interactive Graphics Design System was developed by M & S Computing Incorporated. It is an integrated configuration of hardware and software for user-controlled interactive graphics. It was designed to create graphics for various disciplines such as engineering, electronics and cartography. The Intergraph System is composed of a central processor, together with magnetic tape, disk storage and graphics subsystems. In our case,the central processor is a PDP 11/70, and uses the standard Digital Equipment Corporation (D.E.C.) Operating System RSX-11M-PLUS, a disk-based, priority structured, event-driven operating system. The magnetic tape subsystem is used to enter and off-load data to other computer systems and provide back-up facilities to archive data. The disk storage system is used to store all designs currently active in the system. The graphics subsystem consists of a graphics workstation equipped with a microprocessor needed to perform some local graphics functions thus relieving the central processor to perform other tasks. The graphics system also includes an electrostatic plotter as output device for producing graphics of images on the screen.

The Intergraph graphics workstation has dual screens, one in colour and the other monochromatic. The screen sizes are 19-inch diagonal with a resolution of 1280 by 1024 pixel addressability. Each screen supports a full-screen or four-view quadrant display mode for a total of eight independent views.

The workstation is conceived as a design digitizing system. The input system consists of a floating menu tablet, a digitizing table, a keyboard and a cursor. The floating menu is simply a 15 by 18-inch sheet of heavy plastic with two flat positional coils. The digitizer is mounted on an adjustable drafting base. The 12-button free-floating cursor is an absolute coordinate device which can be used interchangeably on the menu or on the digitizing table.

Intergraph has developed many specialized applications software packages tailored to meet specific needs. The basic general purpose graphics system, IGDS, is a set of programs that can be addressed through the menu tablet, the keyboard or under program control. A brief description of the operational capabilities of IGDS is presented in the following section.

IGDS Operation Overview

The Interactive Graphics Design Software (IGDS) is a multi-purpose graphics system which supports 2-D and 3-D capabilities for interactive graphics operations. Among the six major subsystems of IGDS (see Figure III.1), only four of them are relevant to graphic applications, namely the *Graphics*, *Utilities*, *Help* and *Bye* systems. The *Graphics* subsystem supports the creation and the editing of drawings. The *Utilities* allows the creation of new design files, as well as file management operations such as size adjustment, file copying and deleting. *Help* provides information about each subsystem. *Bye* is used to log off the workstation.

Two types of files, resident on disk, are associated within a graphic session: the design file and the cell file (or symbol file). The drawings are stored in what are called design files. The cell file is merely a library of standard design elements generated by the user, stored and retrieved by name. Both files are open-ended, meaning that their size is limited only by the availability of disk space.

Once the log-in, or sign-on, procedure has been completed, the *Drawing Management Utilities* is first initiated to create a new design file with given name, dimension and size specifications. Following, the *Graphics* is accessed to create, display or edit the drawing file. At this stage the command menu becomes active, and is one of the key elements in making the IGDS a truly interactive system.

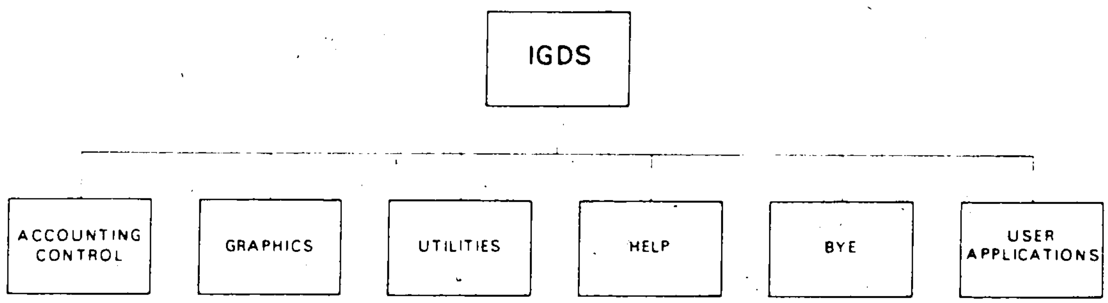


FIGURE III.1 The six major subsystems of the Interactive Graphic Design Software (IGDS).

The establishment of the design working parameters is the first step to achieving any drawing creation. By selecting the *Design Options* tutorial display from the command menu, a direct access to define the design working units is provided. The definition of the working units allows a means of relating the system internal measurement of units of resolution (UORs) to any desired measurable units that a specific drawing would require. The standard format, for all readout data, is given according to the defined units in master unit, subunit and positional unit *(e.g. MU:SU:PU = 1 km:1000m:100)*. The positional unit indicates the degree of accuracy for the drawing and is the determining factor for the optimum design file space. The maximum design area available is a cube of 4,294,967,296 units of resolution. The design file is composed of 63 identically sized overlaid planes, commonly called levels. The capability of a multi-level display, in any combination with the eight-view quadrant display mode provides a tremendous flexibility.

When the working units are defined, graphics elements manipulation such as placement, deletion, modification of lines, arcs, circles, complex shapes, text, etc., may be addressed through the menu interactively on the screen. [15] Direct access in editing interactively a design file is an operational capability of the Intergraph system that can be integrated in computer-assisted cartography. Beyond the fact that the basic IGDS is a sophisticated drafting tool, the Intergraph system provides a method of interfacing specialized user-written programs. This user interface provided with the IGDS to facilitate user development of software represents an interesting potential for cartographic applications. Up to now, no research has been directed toward an assessment of this user interface in terms of cartographic program development. The following section introduces a functional description of programming on the Intergraph system.

An introduction to programming on the INTERGRAPH system

According to the system's description, the Intergraph's adaptability to specific user-defined programming requirements is attributable to the possibility, for the user, to develop his own interactive program or software in order to create, edit and access design files (Intergraph Co., 1980).

----

[15] INTERGRAPH CO, *The IGDS Operating manual*.

Because we are using the RSX-11M-PLUS operating system, four steps are required in any program development process (see Figure III.2)

1. Creation of a source program.
2. Compilation of the source file to produce an object module.
3. Linking the object module to create a task image file.
4. Execution of the task.

Each of the steps is briefly described in this section.

1. Create a source program.

The creation and editing of a source program in a file on disk necessitate the use of the DEC standard editor, EDT. Among the several computer languages available with DEC, such as ASSEMBLER, FORTRAN and COBOL, the FORTRAN 77 language is used here for its compatibility with the IGDS interface software. As suggested in the *IGDS Application Software Interface Document*, the development of a program for mapping applications can be achieved through the use of the Design File Processor Interface (DFPI).

> "DFPI is a collection of software modules which can be called by a FORTRAN and/or a MACRO Assembly language application program to insert data into an IGDS Design File resident on disk."[16]

The Intergraph's system provides the Standard DFPI Interface Subroutines that must be called by any program. These subroutines are INDFPI and DEDFPI. INDFPI initializes and activates DFPI. DEDFPI terminates or detaches DFPI. Moreover the system provides the Place Elements Subroutines allowing every graphic element available on the menu to be placed in a design file under program control.

Supported by the Intergraph system, the DFPI subroutines appear to be of significant importance when developing an IGDS interface program. These subroutines can be manipulated as any FORTRAN subroutines. The interface between the application program written in FORTRAN and the actual design file in IGDS format is therefore performed automatically. DFPI is appealing, but its effectiveness remains to be evaluated.

---

[16] INTERGRAPH CO., *IGDS Application Software Interface Document*, p.1-4

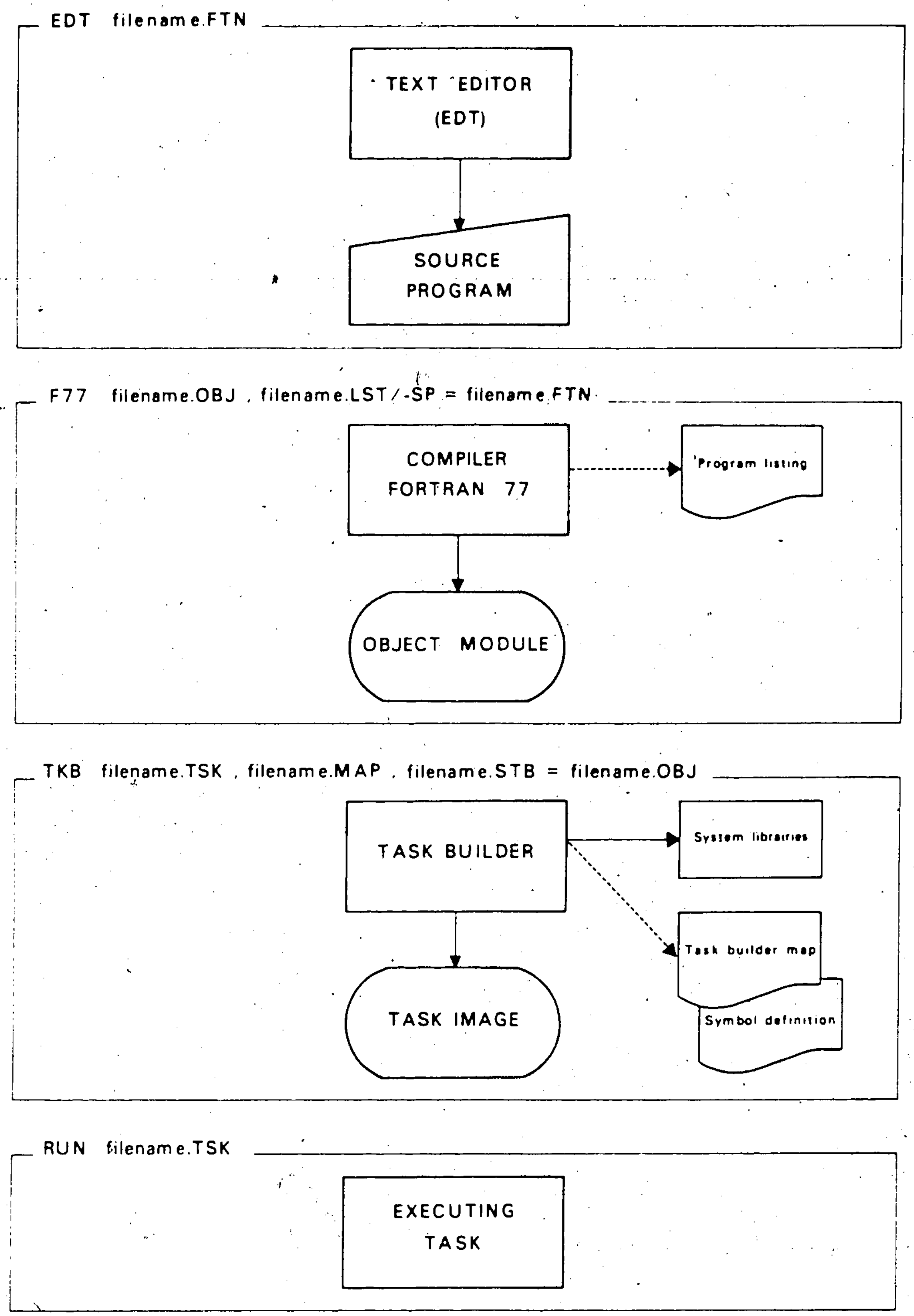


FIGURE III.2 The program development process.
Adapted from D.E.C. (1979), *RSX-11M Beginner's Guide*, page 2-8, and *RSX-11MPLUS Guide to program development*, page 1-11.

2. <u>Compilation of the source file to produce an object module.</u>

The source program must be translated into machine-readable (binary) form. The FORTRAN 77 compiler achieves the translation of the source program into an object module (compiled program) and simultaneously makes sure that the source program follows the language syntax rules. The default file types produced by the compiler are the object file and the listing file (see Figure III.3). The listing file provides information on compilation errors and symbolic names used in the program. The correction s step involves reediting and recompilation of the program module.

3. <u>Linking the object module to create a task image file.</u>

The task building process converts the object module or multiple object modules into a single task image file. It corresponds on other computer systems to the linking or loading process. The default file types generated by the task builder are the task image file, the map file and the symbol definition file (see Figure III.4). The task image file is basically a file of executable code, i.e. is in a format suitable to be loaded into memory and executed. The map file contains the memory allocation information. The symbol definition file is to be used when constructing the task.

Because the PDP-11 processor can address only 32K words (the address limit of 16 bits) at any one time, a task cannot reference more than 32K words at a time (D.E.C., *Guide to Program Development*). This may present serious limitations, although certain advanced programming techniques exist that can help to overcome the addressing limits, such as:

- Overlaying segments of a task with either disk-resident or memory-resident code.
- Mapping to different regions of memory outside the physical limits of the current task space.

4. <u>Execution of the task.</u>

Executing the task is equivalent to running the program. The task image file is first located on the system disk, then loaded into memory and executed. If any errors are

FILES		DEFAULT FILE TYPES
INPUT	OUTPUT	
Source Program		.FTN
	Listing File	.LST
	Object Module	.OBJ

FIGURE III.3 Default Fortran 77 compiler file types.
Adapted from D.E.C. (1979). *RSX-11M Beginner's Guide.*

FILES		DEFAULT FILE TYPES
INPUT	OUTPUT	
Object Module		.OBJ
	Task Image File	.TSK
	Task Builder Map	.MAP
	Symbol Definition	.STB

FIGURE III.4 Default task builder file types.
Adapted from D.E.C. (1979). *RSX-11M Beginner's Guide.*

encountered, the program development process has to be restarted from the beginning. The source file must be edited and recompiled, then a new task image file must be built prior to the run.

The above description of the programming processes is the general set of operation rules which constitutes the starting point for our application study. The next two sections discuss the development and implementation of this specific cartographic application within its working environment.

B. DESIGN OF A CARTOGRAPHIC PROGRAM

Interactive programming requirements

The increasing importance of interactions between the user and the computer has introduced the need for considering new factors in designing interactive graphics systems. This attitude toward the design of interactive systems focuses on the "human factors", also called "ergonomics". Similarly, interactive program developments necessitate attention to the so-called "human factor" aspects. The main concern of conventional noninteractive programs was essentially to ensure the functional capability of the program. For interactive purpose programs, the value of a program rests on its ease of use as much as on its functional capabilities. As more interactive programs become available, the "human factor" aspects are determining factors of the quality of the user-program interaction.

The present cartographic program is interactive oriented and is directed to the non-specialist in computers. Special attention was given to the guidelines, proposed by Foley and Van Dam (1982), in writing interactive programs. These include:

1. Provide simple, consistent interaction sequences.
2. Do not overload the user with too many different options and styles for communicating with the program.
3. Prompt the novice user at each stage of the interaction.
4. Give appropriate feedback to the user.

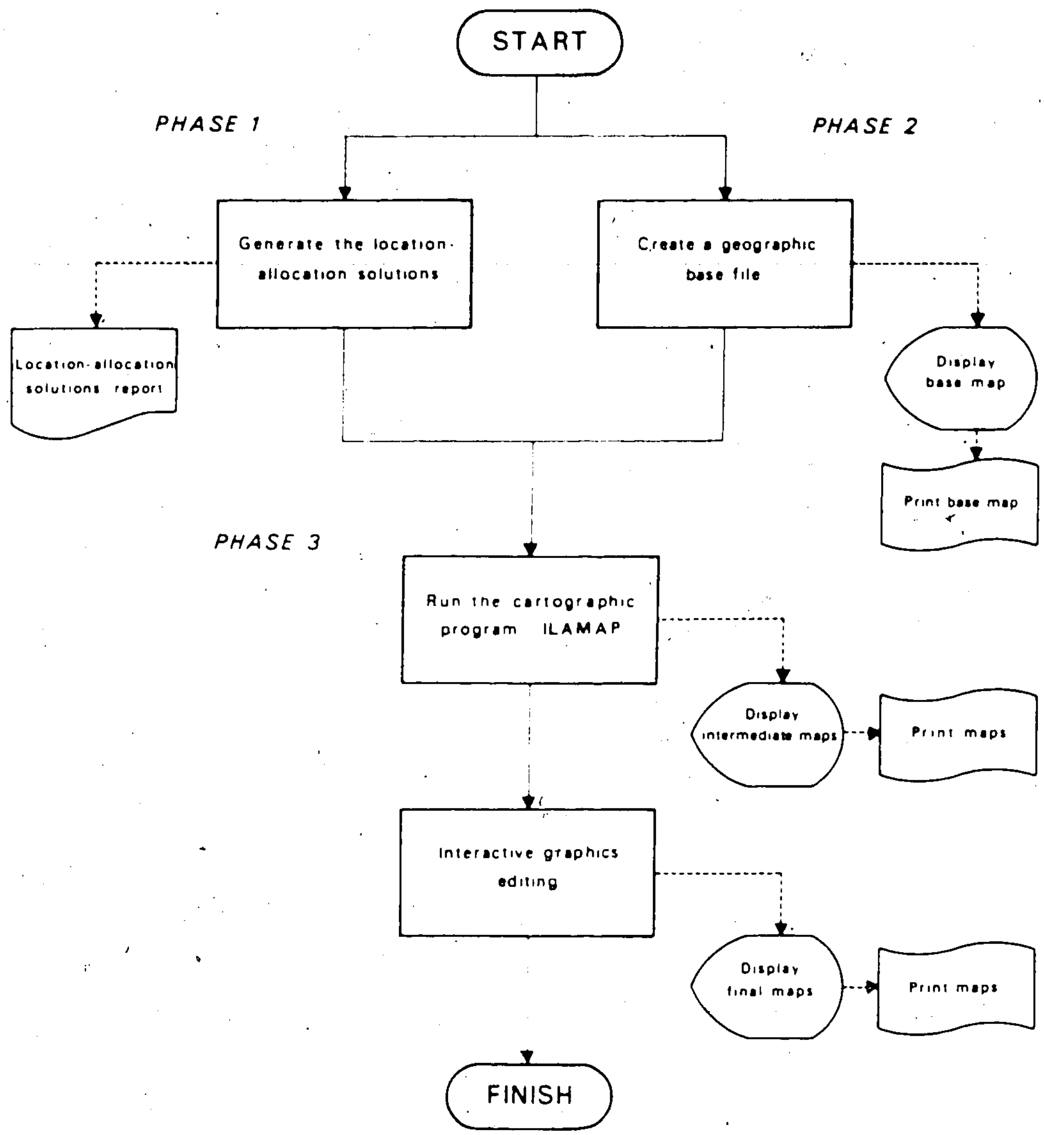
5. Allow the user graceful recovery from mistakes.

The interactive approach to programming differs from batch mode in that it tends to be event-driven as a communication is established between the user and the program. The general dialogue consists basically of a simple loop where the program provides a choice or poses a question, followed by the user's response after which the program reacts by branching to the appropriate procedure. The interactive dialogue introduces an important factor, the response time of a program, which affects considerably the program s usability. Unfortunately, in time-sharing systems, such as the one actually used, control over response time is not possible.

The program design strategy

The program developed for this study, called ILAMAP (Interactive Location-Allocation Mapping Program) was designed as part of a threefold process generating cartographic representations of location-allocation solutions. Phases One and Two of this process are associated with data generation and Phase Three is essentially the program execution and the map's finalisation. As illustrated in Figure III.5, the first two phases are independent, but both are required for the program execution.

Phase One consists of running the appropriate location-allocation program in order to solve a particular problem. The computed solution is presented as numerical maps. At this stage, a report on the location-allocation solution can be printed out. The elements of the solution required by the cartographic program include the absolute number of patrons in each zone, the facilities' locations, and the percentage of a zone's patrons attending each facility. These data are stored in a separate file and become part of the data subsequently used to create a graphical representation of the solution: Phase Two involves the creation of a geographic base file. The base map is digitized in a design file so as to be used as background display. The cartesian coordinates indicating the centroids of each zone of the study area must be recorded from the base map and stored in digital form in order to complete the data file. These two phases of data generation lead to Phase Three, the cartographic program execution. This last phase involves firstly, running ILAMAP at the terminal and accessing the *Graphics* to display the resulting map, and secondly performing interactively on the screen the graphics editing necessary to give the



PHASE 1 - *Generate the location-allocation solutions*

PHASE 2 - *Create a geographic base file*

PHASE 3 - *Run the cartographic program*

FIGURE III.5 A threefold process generating cartographic representations of location-allocation solutions.

map its final form. The program execution sequence is detailed in the next section of this chapter.

ILAMAP was designed to perform a number of operations which could adequately map location-allocation solutions. As seen in Chapter Two, because there is a lack of agreement among cartographers, the most suitable approach to ease the user's task in the process of data generalisation, is to allow for flexibility. In this respect, the program was designed in order to provide the choice to marry data processing and graphics using Jenks' program[17] for an optimal classification in interaction with graphics, or to define the class intervals independently and then access graphics, or to generate a non-classed map. Moreover the program flexibility is also reflected by the relative ease of plotting the elements on the map independently or in any combination.

A simple control language was used through the program to make it relatively easy to use, however there are no default options. In other words, a map cannot be created using only predefined options. In fact, the definition of default options is impractical since the Intergraph's design file works with real world coordinates or distances specific to each case study. There are no standards, such as final map scale or pen size as there would be on final hard copy map, on which symbol dimensions can be determined in order to produce a reasonable 'default map'. Instead, the map author must exercise control over his work. Abuses of computer programs to generate useless maps are well known in cases where creating a map does not require any thinking.

The role of a computer program is essentially to take over the objective processing of the information being mapped, while the subjective contribution of the map design is left to the cartographer. Only the combination of the cartographer's subjective skills and the computer's objective abilities can make a map a product of art and science (Henning and Hargreaves, 1983).

Program Overview

ILAMAP is an interactive FORTRAN package. The basic sequence of operations is illustrated in Figure III.6. As shown in the figure, following the DFPI initialization, access is given to read out the data file. The data required consist of :

---

[17]JENKS George F. (1977), *Optimal Data Classification for Choropleth Maps*, Department of Geography, The University of Kansas.

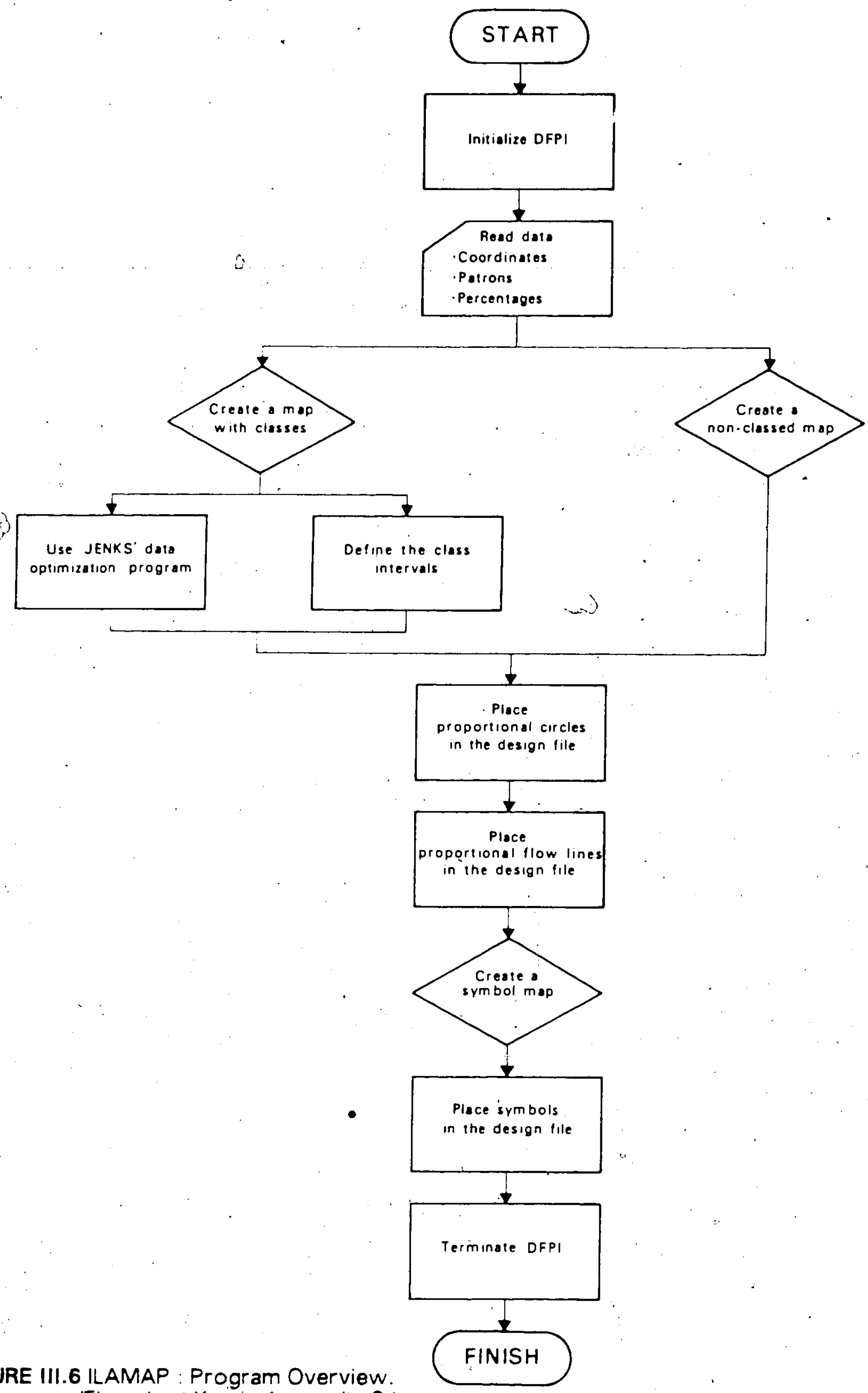


FIGURE III.6 ILAMAP : Program Overview.
(Flowchart Key in Appendix 3.)

1. The centroids of each zone in the study area as recorded in the geographic base file.

2. The number of patrons in each zone according to the initial data as required by the location-allocation program.

3. The attendance pattern solution expressed in percentage of flows of patrons from each demand zone to each facility.

The program provides two independent ways of representing the location-allocation solution, the first being the creation of a map with discrete classes and the second being the creation of a non-classed map. A data generalisation (classification) process can thus be performed if necessary. The data classification is, therefore, based either on the results of Jenks' optimal data classification program or on the user's own selection of class intervals. The graphics generation involves proportional circles representing the number of patrons in each zone and proportional lines portraying the flows of patrons attending each facility. As a last option, a symbol map can be created to display the facilities' locations and attributes. The program is then terminated.

ILAMAP handles the symbolization process accordingly to the theory explained in Chapter Two. The psychological scaling method is used to calculate the proportional circles in order to compensate for the underestimation occurring between circle symbols areas. When the data are classified (range-graded), all values within one class are shown by a standardized circle scaled to the size of the midpoint of that class. For a continuous scaling, relative circle sizes are assigned to each individual value. The establishment of proportional line width follows the conservative approach of a linear graduated scale. As with the circles, when the data are classified, all percentages within one class are shown by a standardized line width established on the midpoint of that class. For continuous scaling, relative line widths are assigned to each individual percentage. Because the range of line widths available on the Intergraph system falls between 0 and 31, we are forced to normalise the proportionality within that range.

Technically, ILAMAP makes use of the Design File Processor Interface Subroutines to achieve the graphics functions. DFPI initialization must be successfully completed before any other DFPI subroutines can be used. It performs the following functions:

1. Creates, initializes and attaches a user-defined region to DFPI and activates DFPI. The communication with DFPI is established through a user-defined region.

2. Retrieves the design file as specified in the argument.

3. Retrieves the cell library as specified in the argument.

4. Swaps the input data to a format compatible with the design file.

5. Sets up the region type. A circular region is recommended.

6. Sets up the terminal identification.

In order to draw the circles in the design file the subroutine ELDFPI which places ellipses in 2-D only is used. The circle is defined by the coordinates of its origin and the two equal axes as given length specified in units of resolution. The specifications, such as colour or line weight, are all set to zero so that the user is not overloaded with too many questions in communicating with the program. These specifications can be easily altered interactively in the design file when reviewing the results. The level on which the circles are to be placed, remains the only parameter to be set during execution of the program. The lines are placed in the design file by defining the endpoints. The subroutine LNDFPI is used with the specifications all set to zero apart from the level and, of course, the line width. The symbols are retrieved from a cell library and the subroutine CLDFPI places each cell by using an angle and scale factor. The symbol specifications' definition are left to the user so that he can exercise more control over the facilities' attributes symbolization. The zones' numbers or names are attached to text nodes, using TNDFPI. The title is placed with the subroutine TXDFPI. The subroutine DEDFPI is used to detach DFPI.

C. DEVELOPMENT AND IMPLEMENTATION

Program development objectives

The major objective in program development was to study the Intergraph's adaptability to meeting specific user-defined programming requirements. As mentioned earlier in this chapter, the Intergraph system provides a method of interfacing specialized user-[illegible]tten programs to create, access and edit design files. Since no studies have tested the Intergraph's adaptability for programming applications, special attention was given, throughout the program development to the estimation of the reliability of the application program interface.

In order to carry out our major objective, a number of smaller discrete objectives have been distinguished. These include the assessment of the effectiveness of the Design File Processor Interface Subroutines (DFPI), of the relative ease of generating one's own application software, and finally of the impacts of the PDP addressing limits of 32K.

Implementation process

The implementation process refers to the necessary steps in order to install an application program on a system so that it functions properly and yields the expected output. This section presents the implementation of the ILAMAP program using a PDP 11/70 in conjunction with an Intergraph system, and stresses the problems encountered throughout.

The basic function of the ILAMAP program consists of generating and editing disk-resident design files. The internal communication between the application program and the design file is established through the Design File Processor Interface (DFPI) and the File Builder (see Figure III.7). The DFPI link is made in a user-defined dynamic region which requires 4K of memory. The region type is set during the initialization of DFPI. It is specified in the argument 6 of the INDFPI calling statement as an option defining the region as circular or non-circular. The Intergraph documentation manual [1] recommends the selection of a circular region. However several tests produced consistent CRAW (Circular Request Area) errors when using a circular region. This CRAW error was related to a physical m[illegible]ry allocation problem. Since storage limitations were more critical problems than ex[illegible] time a non-circular region was attached to the ILAMAP program allowing the use [illegible] words of extra storage. For more sophisticated programming, if the program[illegible] more con[illegible] in conserving storage rather than time he should select a noncircular region (Inte[illegible] 31[illegible]

---

[1] INTERGRAPH CO. 1981. *IGDS Application Software Interface Document.*

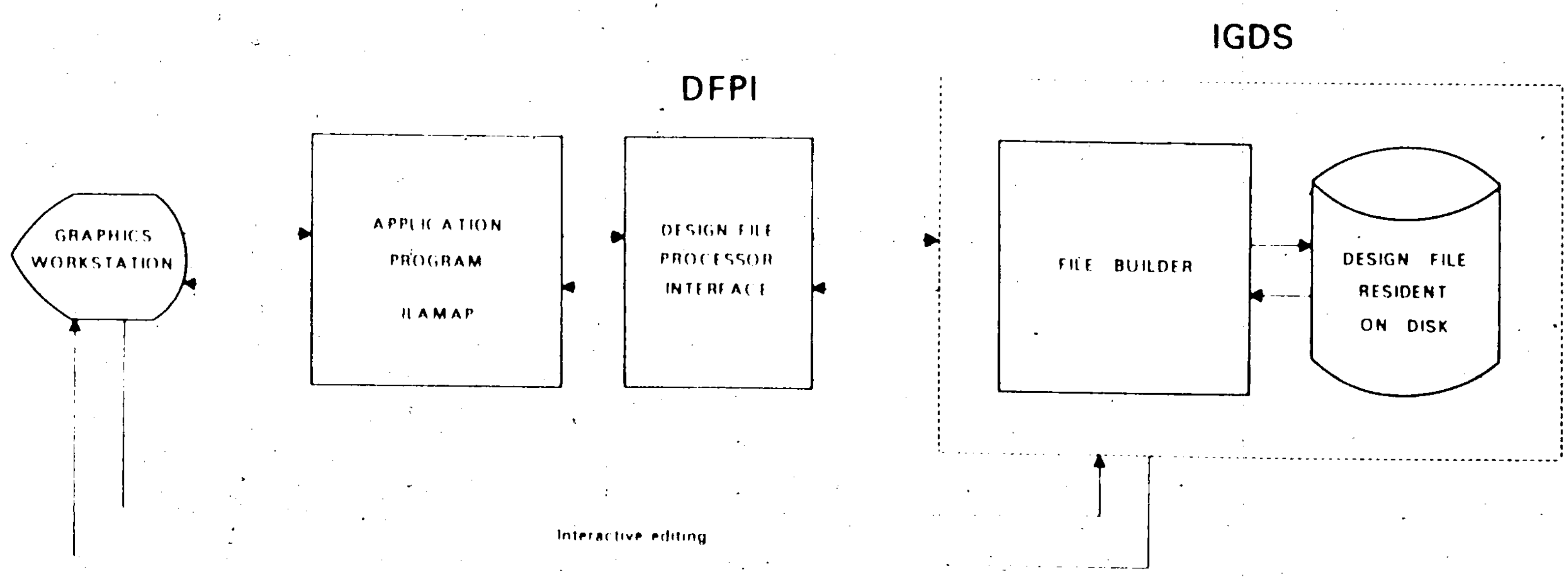


FIGURE III.7 Generation and editing of design files.
Adapted from Intergraph (1981). *IGDS Application Software Interface Document*, page 4-2.

The Design File Processor Interface requires that all design files used by the program be preallocated, contiguous and of sufficient size to contain all the elements to be placed. Every element is placed at the current logical end-of-file sequentially. DFPI locates the end-of-file, either for an empty design file or for a design file containing previous drawings, through the initialization call. In order to ensure that the end-of-file is set up properly, it is preferable to create the design file via the *Utilities* subsystem.

One of the most important aspects of implementing an application program consists of debugging the computer code. The debugging facilities provided for any DFPI error conditions involves the use of two subroutines, ERROR and ERROR9. The ERROR9 subroutine applies to the element last sent to the DFPI interface subroutines, while ERROR is basically used for DFPI errors in general. Any DFPI error messages are transmitted to the program in the form of a code and therefore require the consultation of an explicit error message description. Both ERROR and ERROR9 subroutines are used automatically to process the error conditions. They use three logical output units (5,6 and 7). Logical unit 5 supports the reply from the terminal and is used by the DFPI subroutines as well as by the program. Logical unit 6 is required by several other subroutines to list all error messages. Logical unit 7 allows access to the File Builder message file on the PDP. The programmer can write his own error subroutine and eliminate the use of logical unit 7 for storage economy. In general, the debugging facilities are helpful for the novice programmer, however the same DFPI error message may apply to different problems and thus can cause a lot of confusion.

The documentation provided by Intergraph does not give many guidelines on the use of the DFPI Place Element Subroutines beside mentioning that no arguments can be omitted in any call and that they should be used in the same way as any FORTRAN subroutines. Their use seemed simple and straightforward until scaling and graphic element positioning problems became frequent. Many trials were needed to arrive at the series of operations necessary to relate the system internal measurement of the design file to any desired measurable unit that a specific map would require. The total design area, as mentioned previously, is made up of 4,294,967,296 (2^{32}) units of resolution which units range from -2,147,483,648 to +2,147,483,647 along the X and Y axes. In order to establish a scale relation between the map generated and the design file's units of resolution, two steps are required :

1. Any coordinate or element measure must be multiplied by a positional unit factor every time a DFPI subroutine requires the processing of a coordinate or measure. This positional unit factor is equal to the positional unit value times the subunit and is based on the working units definition (*e.g. MU:SU:PU = 1 km:1000m:100* then the positional unit factor *PUFA = 100 x 1000*). It implies that all values keyed in during the program's execution are expressed in master units.

2. Following this multiplication, the new coordinates or values must be converted to the internal system coordinates to allow accurate element positioning. This conversion can be easily performed using the subroutine CONVER, which converts a floating point double-precision number with a range of 0. to +4,294,967,296. to an Integer*4 FORTRAN format with a range of -2,147,483,648 to +2,147,483,647.

Among the DFPI Place Element Subroutines, LNDFPI is the one normally used to place lines. Severe limitations on its usage have forced us to consider alternative approaches to placing lines in a design file. In fact, the subroutine LNDFPI when used systematically to place a rather small number of lines (less than 50) performed well and within a reasonable execution time. However, independently of the system load, when the subroutine was used to place a larger number of lines the system crashed while performing the line-processing. The critical limit has been found to be as low as 75 lines. Repeated communications with the Intergraph Software Specialists Services have indicated that the problem appeared to be insoluble due to physical limitations of the File Builder itself (refer to Figure III.7).

The alternative approach proposed by the Intergraph Software Specialists Services was to place lines without using DFPI. They provided us with a copy of the computing code they use for that purpose as well as some guidelines to integrating these subroutines into our application program. The experiments conducted in order to verify the performance of these subroutines brought out different problems. This time the File Builder was not contributing to the problems. The major limitations were that only empty design files could then be used. In a case where DFPI was used beforehand to place circles, the lines overwrote some of the circles. Consequently the overall results of the bypass method were not useful.

Finally the subroutine LSDFPI, which places a line string instead of a line, was tried as the last possible solution even though a File Builder problem, similar to the one raised by LNDFPI was expected. For some reason, still unexplained, the subroutine LSDFPI gives consistent results and is not affected by the File Builder's limitations. There is no apparent difference between LNDFPI and LSDFPI, but when placing a large number of lines, the performance of LSDFPI is clearly superior. The reason behind this problem is still unclear.

Early in the implementation process, the size of directly addressable memory space became a major preoccupation. The DFPI task requires a 4K dynamic region of memory beginning at virtual address 160000. For this reason and because we were working on a PDP, all tasks using DFPI and the interface subroutines could not exceed 28K words in size.

The first attempt to integrate Jenks' program within our task demonstrated clearly the need to overcome a major storage problem. For that purpose the operating system provides an overlay capability to reduce the memory requirements of the task image file. The overlay structure consists of dividing a task into a series of segments composed of :[1]

1. A single root segment, which is always in memory.
2. Any number of overlay segments which can reside on disk and share virtual address space and physical memory with one another (disk-resident overlays).

The segments consist of one or more object modules which correspond to program sections. Segments that overlay each other must be logically independent. When using the overlay capability, the amount of physical memory required for the task is determined by the length of the root segment and the length of the longest overlay segment. The total memory used by the root segment plus the longest overlay segment must not, therefore, exceed the 28K words available. This is true even if the longest of

----

[1] DIGITAL EQUIPMENT CORPORATION (1979), *RSX-11M/M-PLUS Task Builder Manual*, p.4-1.

the overlay segments does not use DFPI.

In fact because the ILAMAP task could move sequentially through a set of modules it was well suited to the use of an overlay structure. The arrangements of the overlay segments within the physical memory of a task can be represented schematically as a tree-like structure (see Figure III.8). Each branch of the tree represents a segment. The ILAMAP tree has a single root segment and six main branches which support the following functions

Root segment controls the program sequence by calling the overlay segments.

. Overlay segment 1 sets the options as selected by the user and return them to the root.

. Overlay segment 2 executes Jenks data optimization program.

. Overlay segment 3 creates a map using Jenks results.

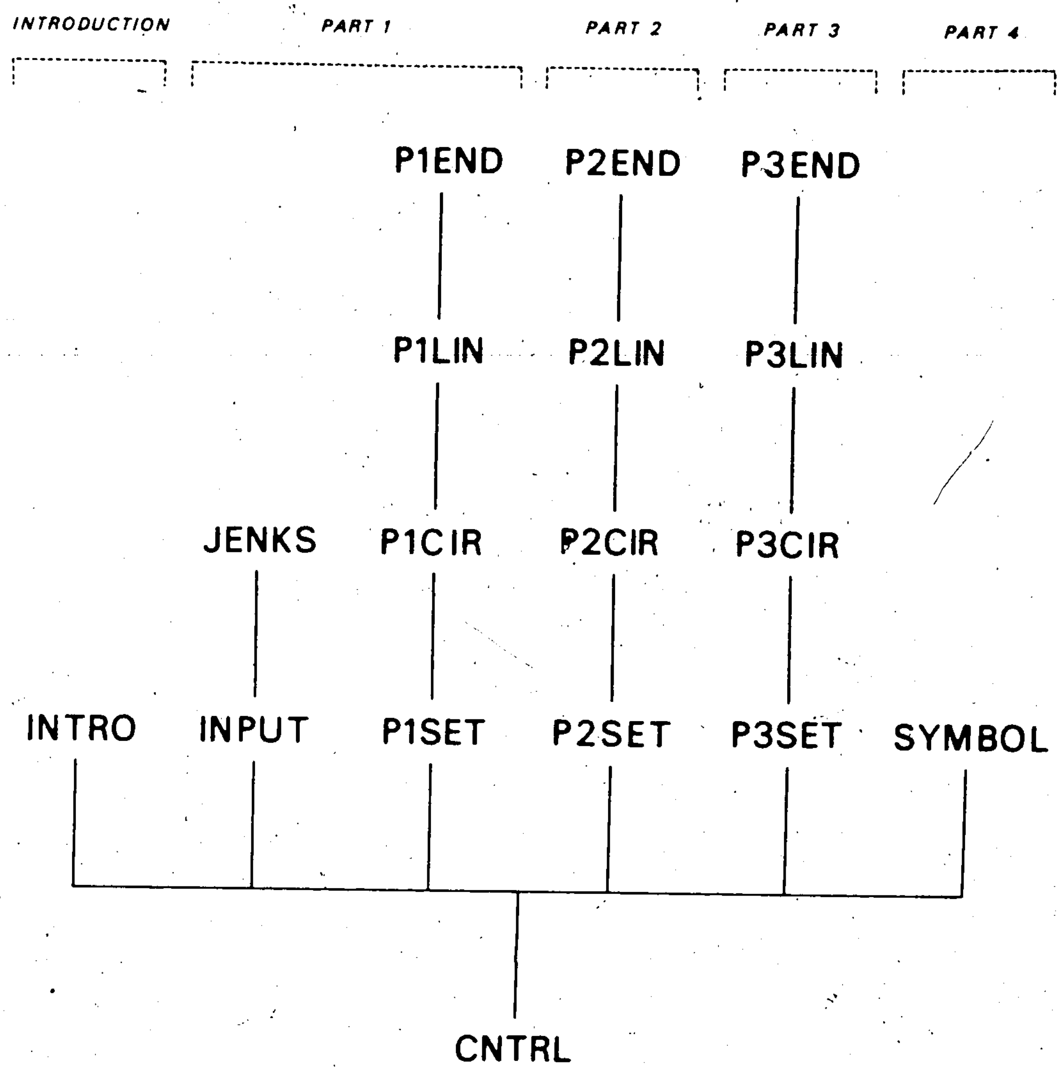
. Overlay segment 4 creates a map with one's own class definition.

. Overlay segment 5 creates a non-classed map.

. Overlay segment 6 creates a symbol map.

The tree-like overlay structure has to be created with the Overlay Description Language (ODL). The ODL file (see Figure III.9) contains a series of overlay description directives in order to control the allocation of the physical memory in the task. So when it comes to the task building process the ODL file containing a description of the overlay structure is used instead of a single object file (see Figure III.10).

The memory allocation map (see Figure III.11), generated by the the task building process provides information on the effect of using a disk-resident overlay on the allocation of the physical memory for the ILAMAP task. As illustrated in Figure III.12, each overlay segment, being logically independent, can thus take advantage of the 28K words available. However, as the following section explains, even when maximizing the allocation of the physical memory the addressing limit of 28K for each segment can still present important limitations.



INTRODUCTION	-	*Set the program's options*
PART 1	-	*Create a map using Jenks' data classification*
PART 2	-	*Create a map using your own class intervals*
PART 3	-	*Create a non-classed map*
PART 4	-	*Create a symbol map*

FIGURE III.8 ILAMAP tree-like overlay structure.

```
        .PSECT  APREGN, RW, REL, GBL, OVR
        .ROOT   ILACNTRL-F77-DF-F4-UT-OVR
OVR:    .FCTR   *(OV1, OV2, OV3, OV4, OV5, OV6)
OV1:    .FCTR   ILAINTRO-F77-F4-UT
OV2:    .FCTR   ILAINPUT-ILAJENK-F77-F4-UT
OV3:    .FCTR   ILAP1SET-ILAP1CIR-ILAP1LIN-ILAP1END-F77-DF-F4-UT
OV4:    .FCTR   ILAP2SET-ILAP2CIR-ILAP2LIN-ILAP2END-F77-DF-F4-UT
OV5:    .FCTR   ILAP3SET-ILAP3CIR-ILAP3LIN-ILAP3END-F77-DF-F4-UT
OV6:    .FCTR   ILASYMBOL-F77-DF-F4-UT
F77:    .FCTR   QS0:[14,11]F77.OLB/LB
DF:     .FCTR   QS0:[15,1]DFPIIS.OLB/LB
F4:     .FCTR   QS0:[1,1]F4POTS.OLB/LB
UT:     .FCTR   QS0:[15,1]UTILITIES.OLB/LB
        .END
```

FIGURE III.9 Overlay Descriptor List using DFPI Interface Subroutines for the ILAMAP task.

```
ILAMAP.TSK/FP,ILAMAP.MAP/-SP=ILAMAP.ODL/MP
UNITS=7
VSECT=APREGN:160000
VSECT=IREQ:174340
VSECT=DFPI:177400
WNDWS=1
ASG=TI:5
ASG=TI:6
GBLDEF=RSX11$:1
GBLDEF=EFNX1:1
GBLDEF=EFNX2:2
GBLDEF=MSGLUN:7
GBLDEF=MSGFLG:7
GBLDEF=EFNX3:3
GBLDEF=LUNX1:1
GBLDEF=LUNX2:2
GBLDEF=LUNX3:3
//
```

FIGURE III.10 Command file with overlays using DFPI Interface Subroutines for the ILAMAP task.

```
ILAMAP.TSK;4   MEMORY ALLOCATION MAP  TKB                PAGE 1
                      1-AUG-84   15 14

PARTITION NAME : GEN
IDENTIFICATION : 01AUG
TASK  UIC      : [266.10]
STACK    LIMITS: 000310 001307 001000 00512.
PRG XFR ADDRESS: 002232
TOTAL ADDRESS WINDOWS: 2.
TASK  IMAGE  SIZE  : 28064. WORDS
TASK ADDRESS LIMITS: 000000 155477
R-W DISK BLK LIMITS: 000002 000654 000653 00427.

ILAMAP.TSK;4 OVERLAY DESCRIPTION:

BASE     TOP       LENGTH
----     ---       ------
000000  026757  026760  11760.        ILACNT
026760  050143  021164  08820.           ILAINT
026760  155477  126520  44368.           ILAINP
026760  140413  111434  37660.           ILAP1S
026760  142643  113664  38836.           ILAP2S
026760  136773  110014  36876.           ILAP3S
026760  142447  113470  38712.           ILASYM

:

*** TASK BUILDER STATISTICS:

    TOTAL WORK FILE REFERENCES: 535836.
    WORK  FILE  READS: 416.
    WORK  FILE WRITES: 387.
    SIZE OF CORE POOL: 25088. WORDS (98. PAGES)
    SIZE OF WORK FILE: 46080. WORDS (180. PAGES)

    ELAPSED TIME:00:04:12
```

FIGURE III.11 Memory allocation map for the ILAMAP task.
The arrows indicate memory (in octal) allocated to each segment.

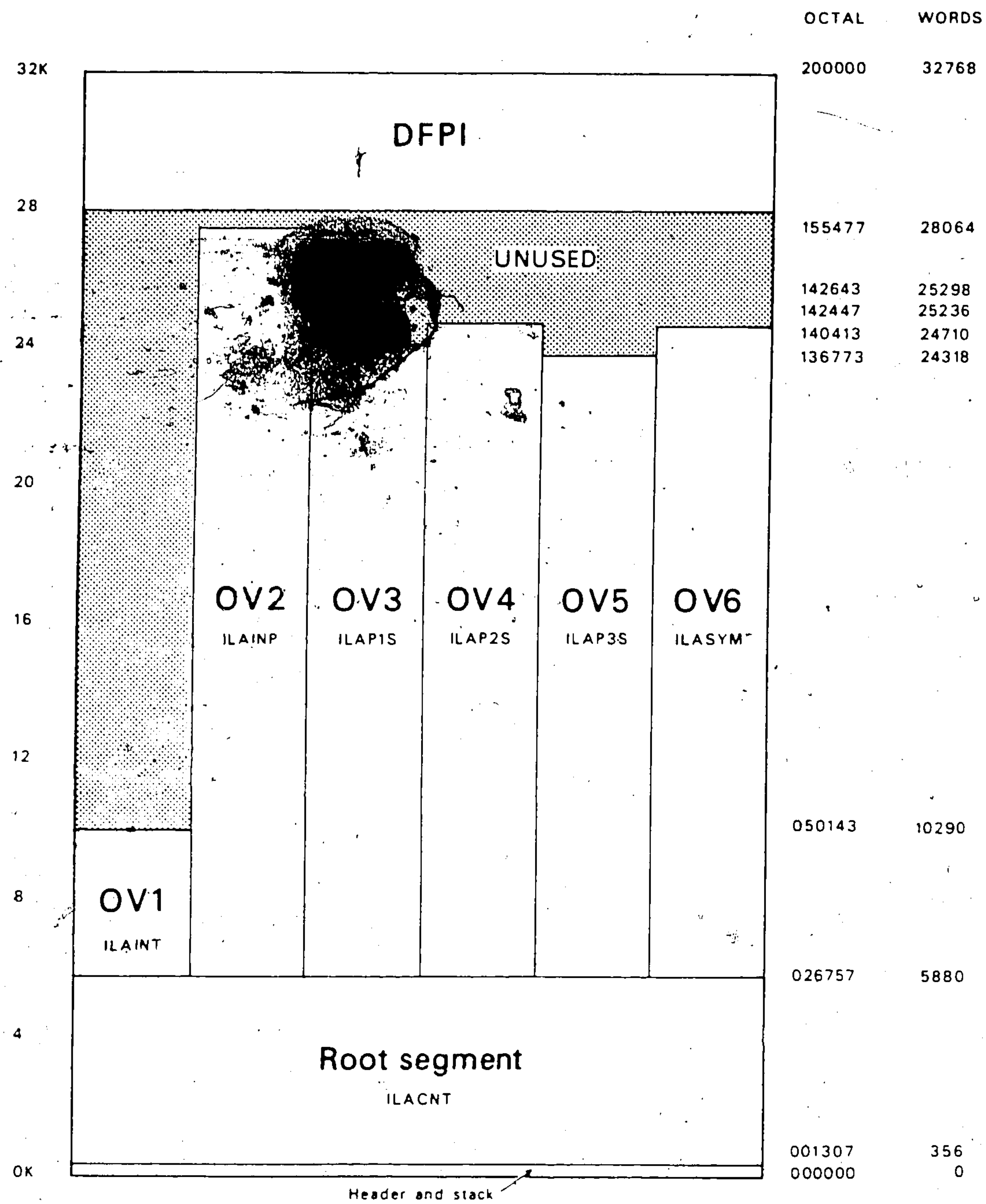


FIGURE III.12 Allocation of physical memory for the ILAMAP task.

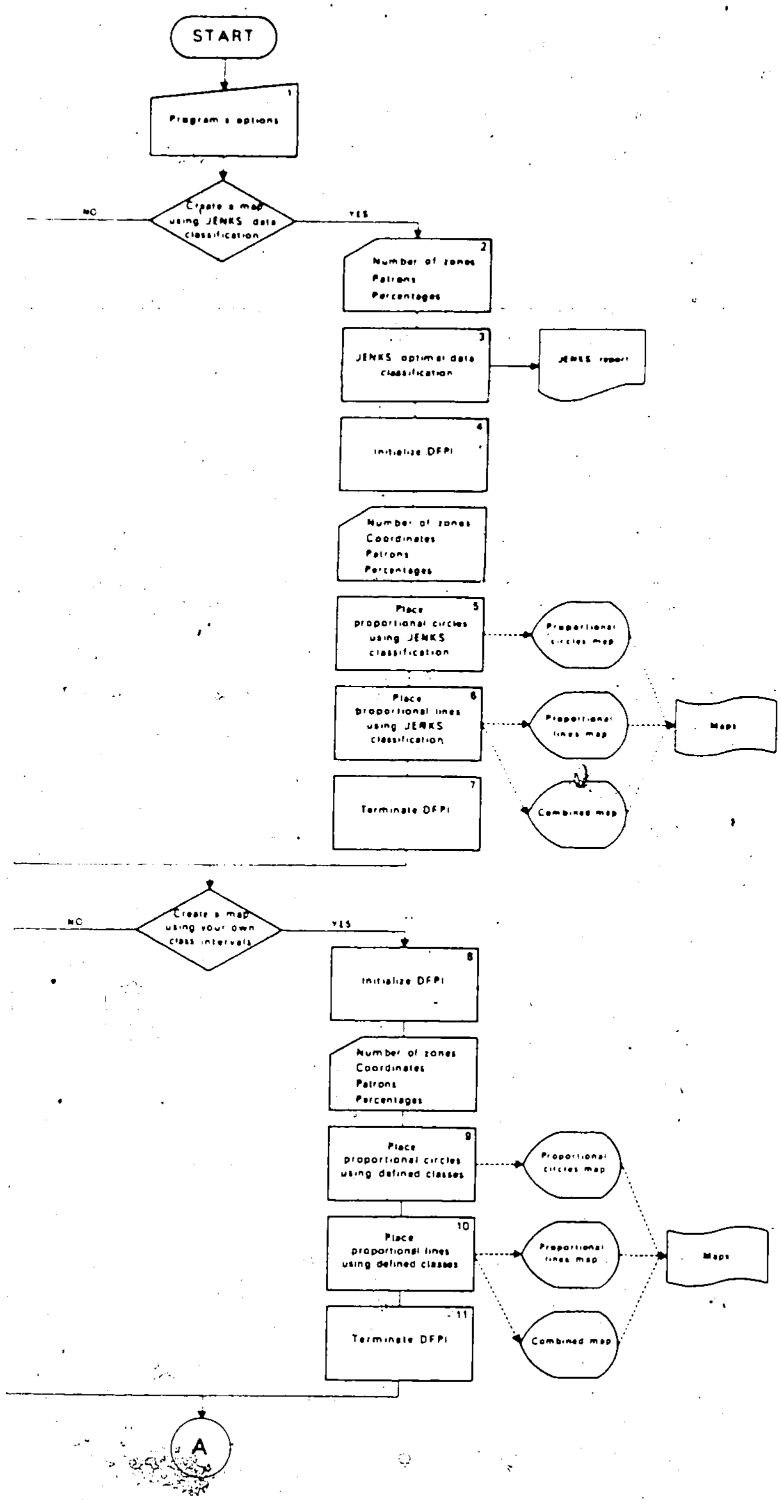
The final program structure

The ILAMAP program has been restructured using the overlay capability of the PDP 11 70, as detailed in the previous section. The computer program is, therefore, composed of a root segment and six main branches presented to the user in four major modules or parts. Figure III.13 summarizes the final program sequence in a flowchart and lists the subroutines names associated to each program segment. A complete listing of the programming code is presented in Appendix 1.

Modularity and interaction are the key features of the final program structure. As Figure III.13 shows, each module is entirely independent and supports the creation of a different type of map. This structure presents some advantages. The program s options allow the creation of up to four maps in any combination in the same run. The four maps can be stored in a single design file using the multi-level capability for display or they can be stored in four individual design files, using the reference files system for simultaneous display. The Intergraph software supports the capability to associate the active design file with as many as three reference files as background data.

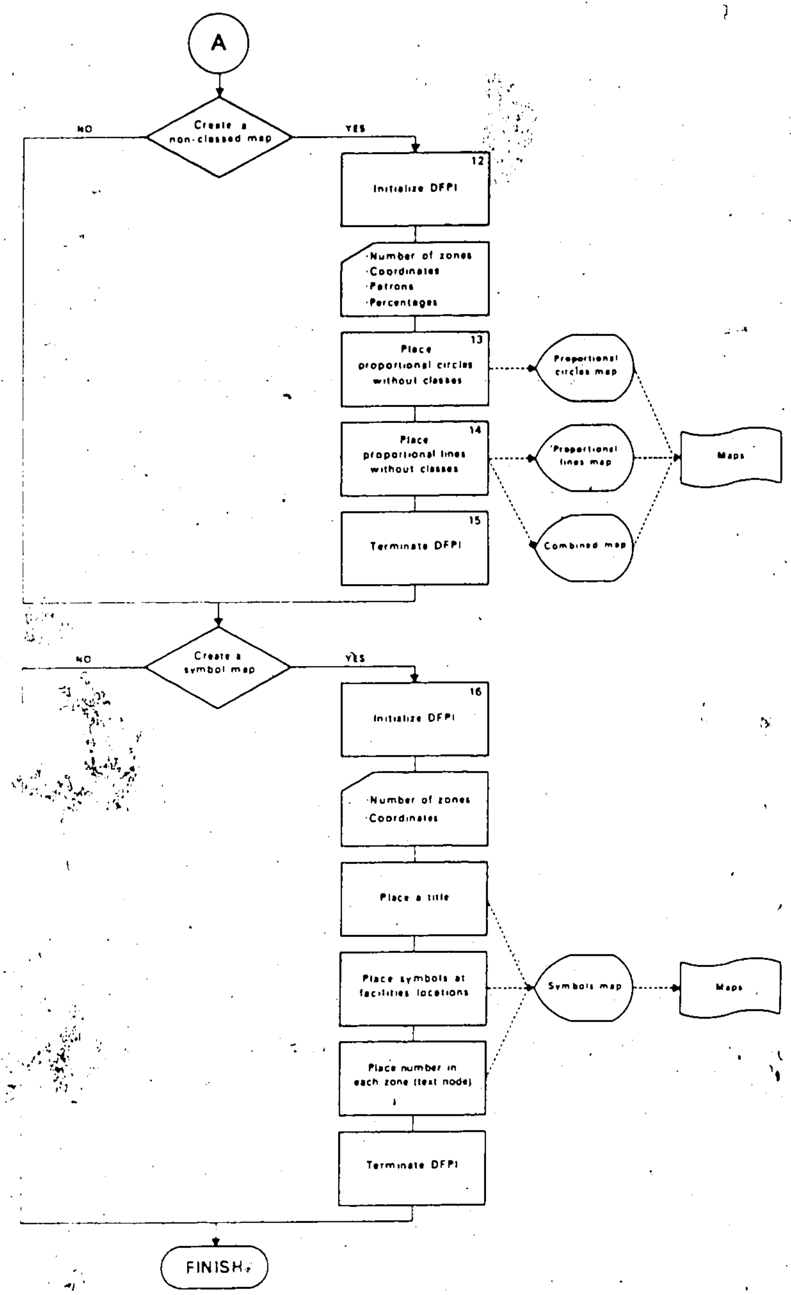
Basically, the first three modules of ILAMAP performs a similar graphics generation sequence. All three use one subroutine to initialize DFPI and read the data (see Figure III.14), one subroutine to place the circles (see Figure III.15), another to place the lines (see Figure III.16) and a last one to detach DFPI. It is recommended that INDFPI and DEDFPI be included in the same overlay. As part of an interactive sequence, the graphics elements of the map can be generated independently or in combination. This independent design allows flexibility often needed when dealing with a computer for cartographic applications. The user does not have to formulate all queries and problems in advance, but instead controls the step-by-step map generation. Each module provides a different way of communicating information and can be used as a reference to evaluate the most suitable representation for specific purposes and objectives. The last module generates a symbol map which can be used to add supplementary information to the final map.

There are certain limits in the program. One major limitation with respect to the impact of the physical memory addressing limit of 28K for each segment lies in the limited size of cartographic problems that can be treated. In fact, to use Jenks' optimization program, the number of zones in the study area as well as the number of percentage


START
1
Program's options
Create a map using JENKS data classification
NO
YES
2
Number of zones
Patrons
Percentages
3
JENKS optimal data classification
JENKS report
4
Initialize DFPI
Number of zones
Coordinates
Patrons
Percentages
5
Place proportional circles using JENKS classification
Proportional circles map
6
Place proportional lines using JENKS classification
Proportional lines map
Maps
7
Terminate DFPI
Combined map
Create a map using your own class intervals
NO
YES
8
Initialize DFPI
Number of zones
Coordinates
Patrons
Percentages
9
Place proportional circles using defined classes
Proportional circles map
10
Place proportional lines using defined classes
Proportional lines map
Maps
11
Terminate DFPI
Combined map
A

Continued ...

... Continued



SUBROUTINES ASSOCIATED TO EACH PROGRAM SEGMENT

1 INTRO	5 P1CIR	9 P2CIR	13 P3CIR
2 INPUT	6 P1LIN CTLINE	10 P2LIN CTLINE	14 P3LIN CTLINE
3 JENKS FAVAR	7 P1END	11 P2END	15 P3END
4 P1SET	8 P2SET	12 P3SET	16 SYMBOL

FIGURE III.13 ILAMAP : Program Flowchart.
(FlowChart Key in Appendix 3.)

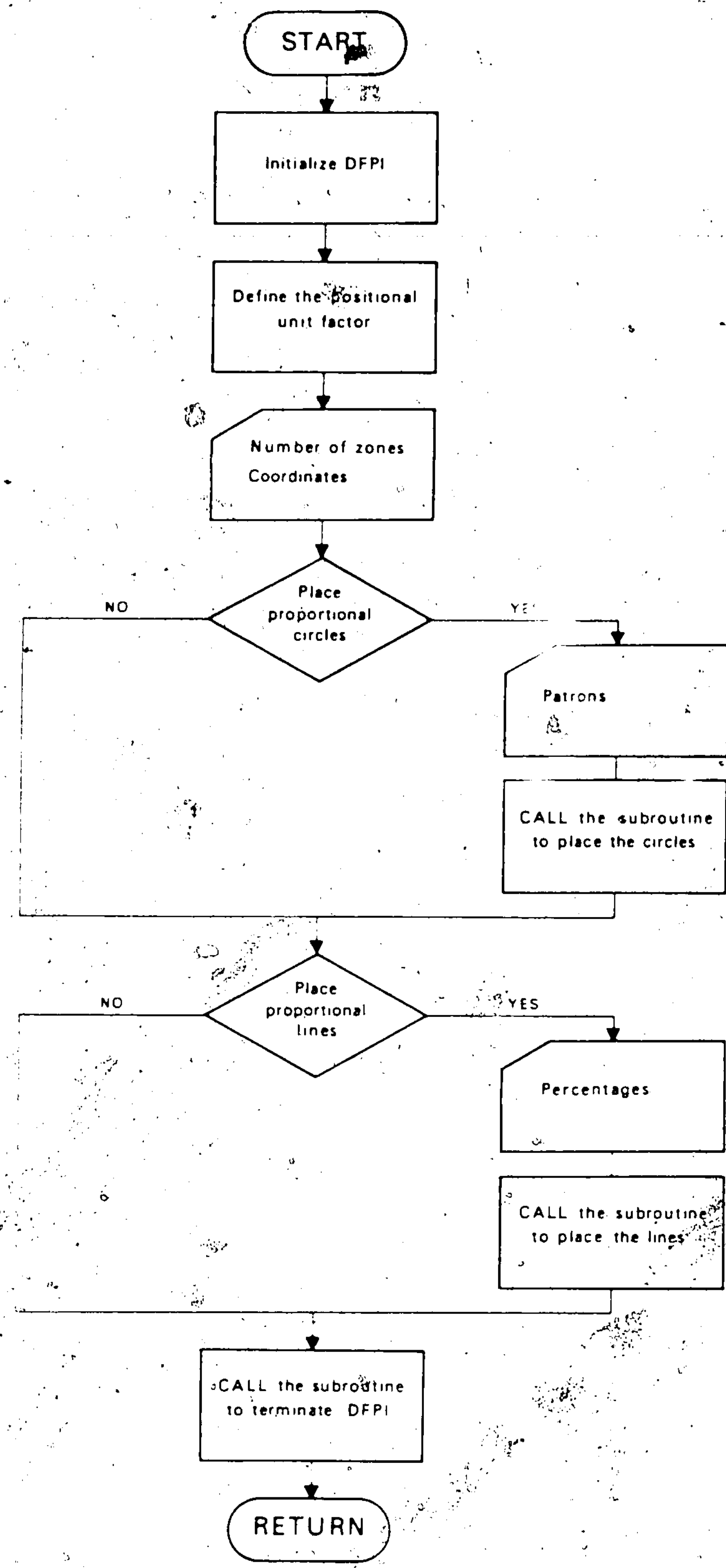


FIGURE III.14 General flowchart of subroutines P1SET, P2SET and P3SET to initialize DFPI and read the data.
(Flowchart Key in Appendix 3.)

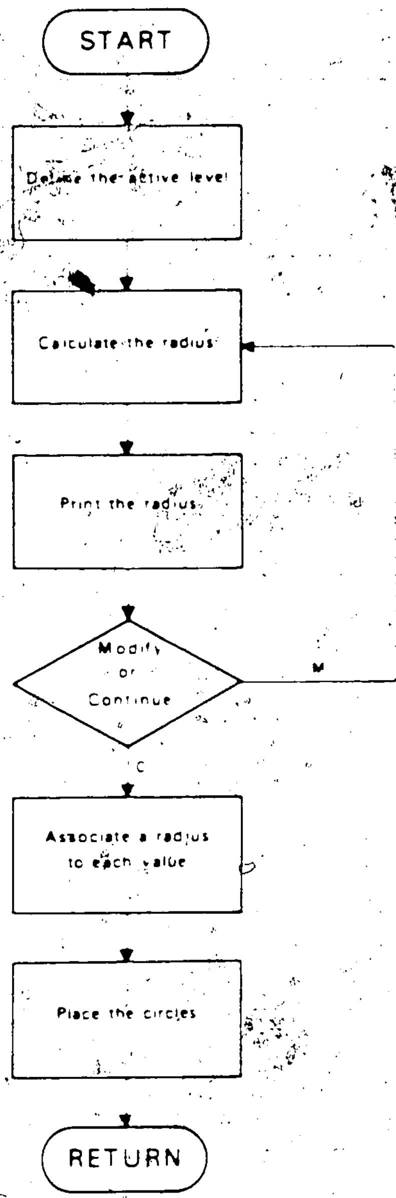


FIGURE III.15 General flowchart of subroutines P1CIR, P2CIR and P3CIR to place the circles.
(Flowchart Key in Appendix 3.)

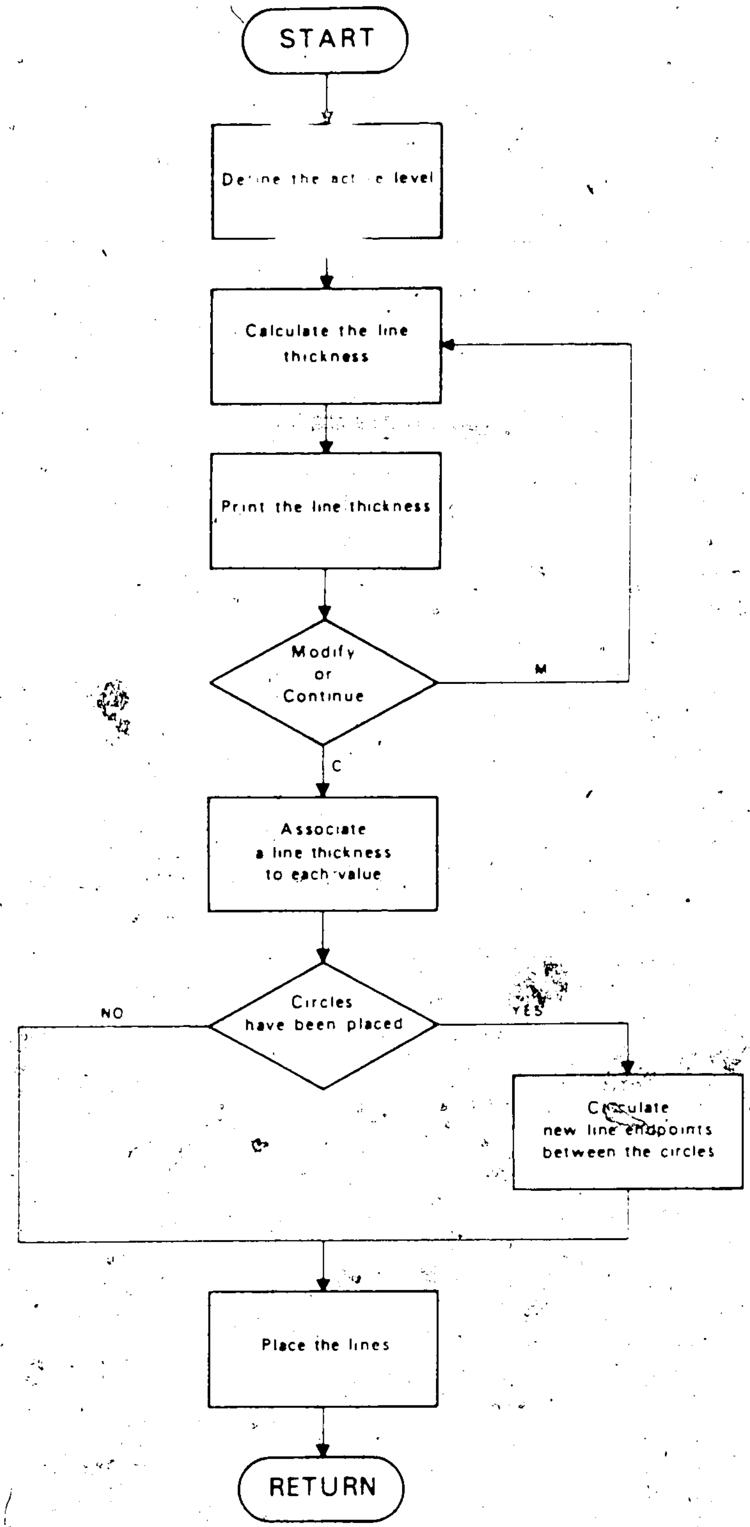


FIGURE III.16 General flowchart of subroutines P1LIN, P2LIN and P3LIN to place the lines. (Flowchart Key in Appendix 3.)

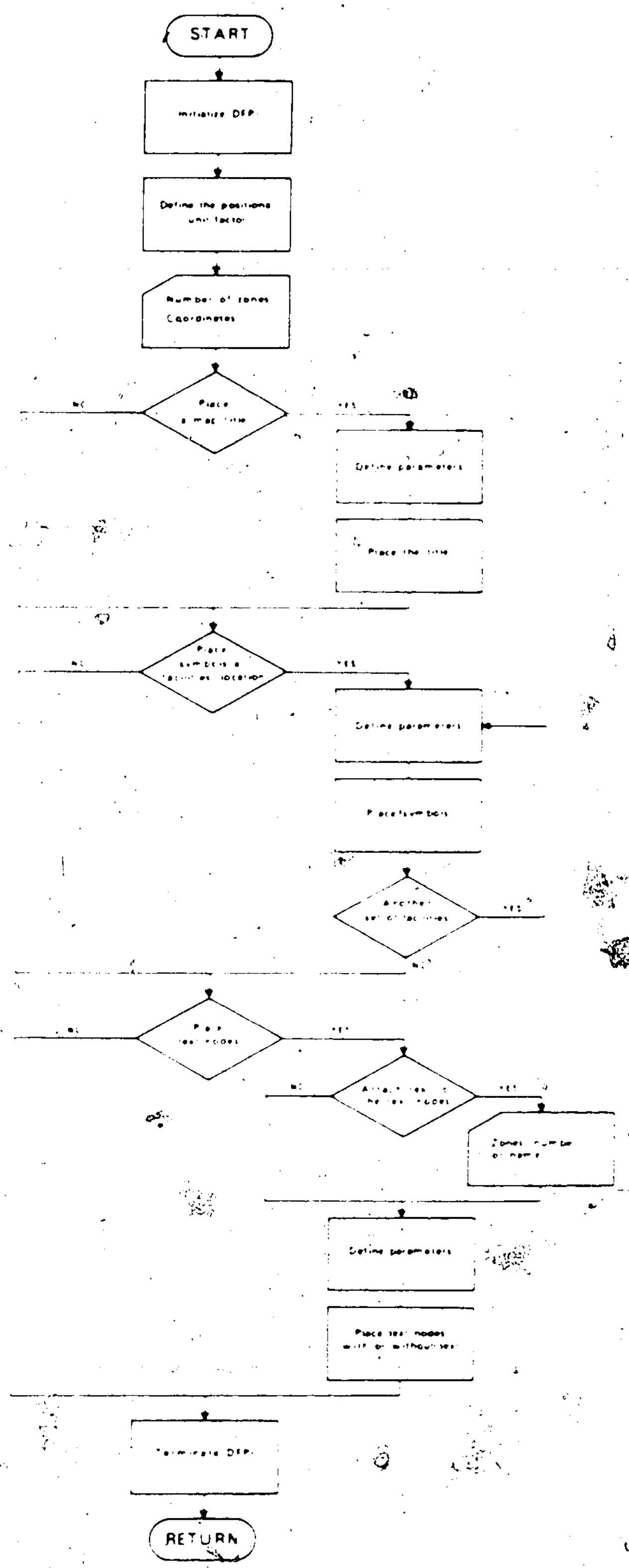


FIGURE III.17 General flowchart of the subroutine SYMBOL to generate a symbols map.
(Flowchart Key in Appendix 3.)

flows of the attendance pattern must be less than or equal to 50. This severe constraint reduces the interaction between data classification and graphics to a useless level. A more reasonable problem size can be handled when a map using the user's class definition or a non-classed map is generated. The maximum number of zones or demand points in that case can reach 100 and the maximum number of percentage flows 150. The number of facilities is not critical. In fact, up to 10 facilities can be processed at once but, as the Figure III.17 illustrates, the symbols are placed within a loop which allows the possibility of placing different sets of symbols by looping as many time as desired.

An ILAMAP application example is presented in the next chapter with a series of maps illustrating the range of applicability of the program. The results are analysed from a cartographic view point and the use of the Interactive Graphics Design System is discussed.

IV. ILAMAP : EXAMPLE APPLICATION AND RESULTS ANALYSIS

This chapter presents a series of example maps produced with ILAMAP and their analysis from a cartographic view point. The example application must be taken as a general case intended to illustrate the range of applicability of ILAMAP and to demonstrate how cartographic analysis can become an integral part of research. For this purpose it is felt that no interpretation of the results from a location-allocation point of view is needed.

A. EXAMPLE APPLICATION

The interaction and modularity of ILAMAP allow flexibility and facilitate the generation of a series of maps from a unique data set. The program is especially oriented toward the creation of flow maps to represent solutions of location-allocation problems but it can also be used in a broad range of applications in order to generate proportional circle maps and symbol maps. An application of ILAMAP usually consists of a number of stages as follows:

- Problem formulation
- Initial maps
- Intermediate maps
- Final maps

A complete guide to the procedure involved as well as examples of data files and runs are presented in the User's Manual in Appendix 2.

Problem formulation

Every application of ILAMAP must begin with a clearly defined objective in order to select the appropriate options offered by the program. More important is the decision as to whether or not the map should be considered strictly as an end product since maps can play a major role in information processing as well as in the production of final outputs. In a typical cartographic application for mapping location-allocation solutions, the problem formulation involves a definition of the study area, geographic scale and variables used to characterize the behaviour of each zone.

Our example represents the attendance patterns of automobile registrants in the City of Edmonton. This case study has been conducted by Hodgson (1981). The map of the City of Edmonton has been digitized and stored in a design file (see Plate IV.1) so as to serve as background information. Thirty demand zones, based upon the Post Office's postal code were employed. The motor vehicle owners constituted the patrons to be associated with each zone. The public facilities to be located are licensing offices where the motor vehicle owners are required to renew their automobile registration annually. The attendance data are produced by a consumer's welfare model integrating interaction-based allocation. The location-allocation model employs a production-constrained gravity method in order to allocate patrons to facilities, and maximizes a consumer's welfare function by incorporating patron's reactions to facility size and travel time, which are respectively described by parameters *A* and *B*[20]. The effect of travel time is modelled by a negative exponential. Changes in *B* affects the allocation pattern and results in alternative solutions. The lower is *B*, the more numerous are the low percentage flows expressed in the solution.

In Hodgson's study, the number of patrons in each zone is illustrated by a choropleth map, the facilities' locations positioned on a separate map and the proportional allocations are presented in table form. The use of a choropleth map to portray absolute values is cartographically inappropriate and the table of numbers does not help to visualize the allocation pattern.

ILAMAP allows a complete integration of these elements in a single map which provides a visualization of the spatial phenomenon. Proportional circles give an accurate representation of patrons and proportional flow lines portray effectively the attendance pattern. The example presented here on maps, focus on the five-facility solution resulting from the consideration of uniform sized facilities ($A = 0.1$) and the automobile travel time ($B = 1.0$).

----

[20]The model's specification is described by the equations (8) to (10) in Hodgson (1981), page 499.

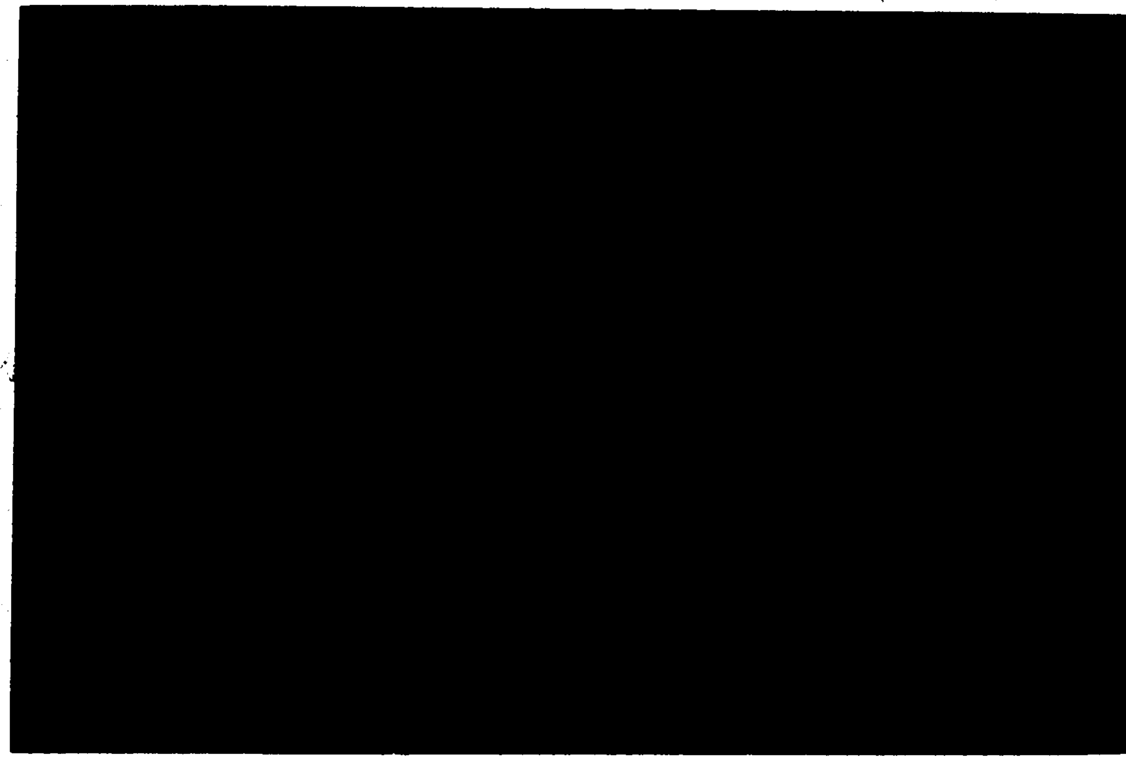


PLATE IV.1 Digitized base map of the City of Edmonton.

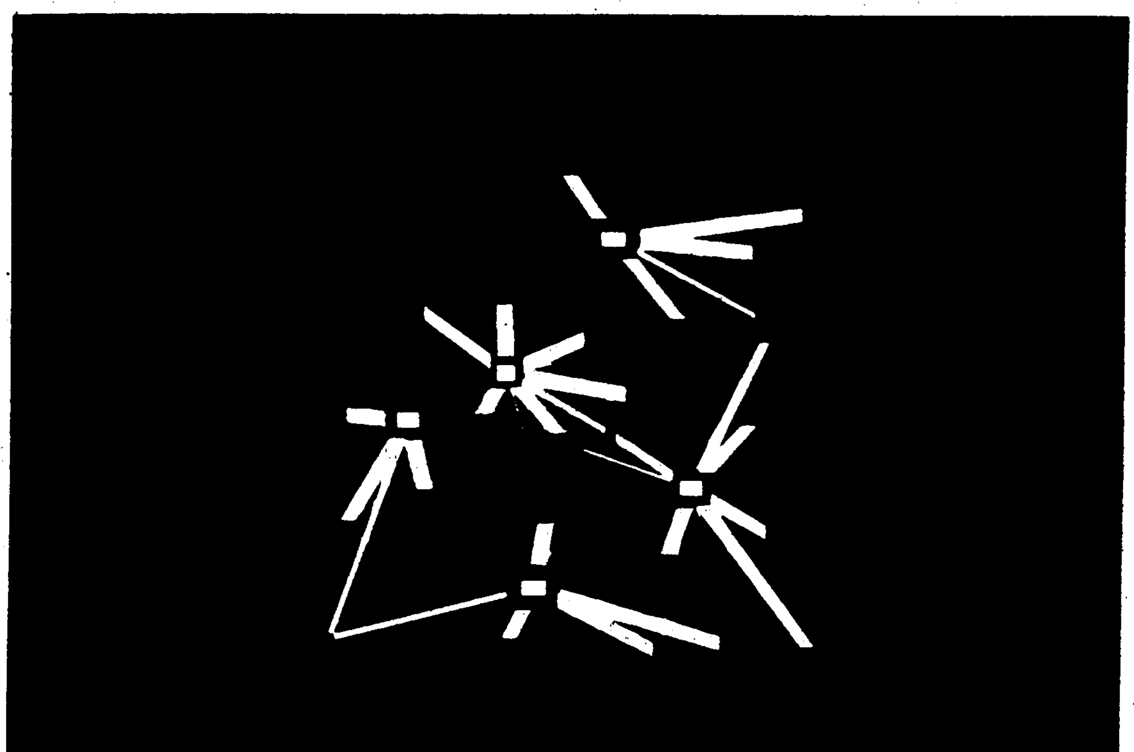


PLATE IV.2 Initial map generated using PART 2 of ILAMAP.

Initial maps

The initial maps are those directly generated by the program and not yet edited. They are considered as experimental. Initial maps are temporary maps, produced on a CRT screen providing the researcher with an image of the distribution. Since they are usually used only by the researcher there is no substantial need for title, legend etc.. For example, several such maps can be produced making use of different data groupings and varied circle sizes or line widths in order to select the most appropriate combination. This stage is also essential when a non-classed map is created because it may necessitate several runs to achieve a good representation. The generation of temporary maps represents a tremendous increase in flexibility for the researcher as well as a drastic methodological change in map production.

Plate IV.2 shows an example of initial map as generated with ILAMAP (PART 2). It illustrates a typical example of how initial results are overlayed on the base map. No interactive graphics editing has been performed but the spatial distribution the phenomena is well expressed in terms of contrasts, general tendencies and relative order. Despite the fact that no legend has been yet included, the map should stand as a complete and consistent graphic expression since a map which absolutely requires a legend, can be considered as a graphic failure (Muller, 1983).

Intermediate maps

The level design concept of the Intergraph system lends itself well to cartographic applications. The map information can be reviewed on a level by level basis, allowing a visualization of the results through temporary maps or "intermediate maps".

The intermediate maps serve not only to compose final maps but they also play a very important role in the analysis of location-allocation results. As mentioned before, changes in the model's parameters produce alternative solutions. For a single location-allocation problem numerous sets of alternative solutions may be sought which would lead to significant differences in the map representation. The ability to quickly produce graphic displays which summarize the different solutions and the capabilities for comparisons between them provides an invaluable support for management decision-making. Systematic analysis of the results can be conducted by reviewing the detailed information contained in the intermediate maps. The information is presented in various form of map allowing visual enhancement for greater comprehension.

Plates IV.3 to IV.9 illustrate the detailed allocation features of each individual class interval and Plate IV.10 constitutes the combination of the seven intermediate maps. In this example, the combined map (Plate IV.10) remains legible, however this is not always the case. In cases where the interactions between the demand zones and the facilities are too complex, the capability to display the mapped information in any combination desired can help selecting the information which may have to be excluded in order to present a legible final map. Plate IV.11 shows an example of alternative solution. In this particular case, the interaction model has been applied with B equal to 0.4. The attendance pattern is, therefore, characterized by an increase in lower percentage flows. Plate IV.12 shows the lower classes representing the flows of less than 5%. They were excluded in the final map (Plate IV.11) to preserve maximum legibility.

Final maps

The final maps constitute the end product intended to satisfy the traditional illustrative role of graphics. Depending on their intended reader and the objective followed by the cartographer, different maps can be generated using the ILAMAP program, the communicative purpose of the map being determining for the final form.

Plate IV.13 shows the map produced with PART 1 of ILAMAP making use of Jenks' data classification. As noted, the percentage of patrons attending each facility of less than 1% are not included in the map ormation. The main reason to eliminate these percentages was related to the program's own size limitation problem. Although there is a loss of information in the final map, the class interval selection, as generated by Jenks optimization program, allows an accurate representation of the distribution. The advantage of creating a map using Jenks' data classification is in the tremendous time economy that can be made in processing the data.

Plate IV.10 illustrates the case where the class intervals have been previously defined by the user in order to use PART 2 of ILAMAP which allows larger problems to be treated. The map gives more details on the lower values and has the definite advantage of presenting the entire range of data, while remaining legible.

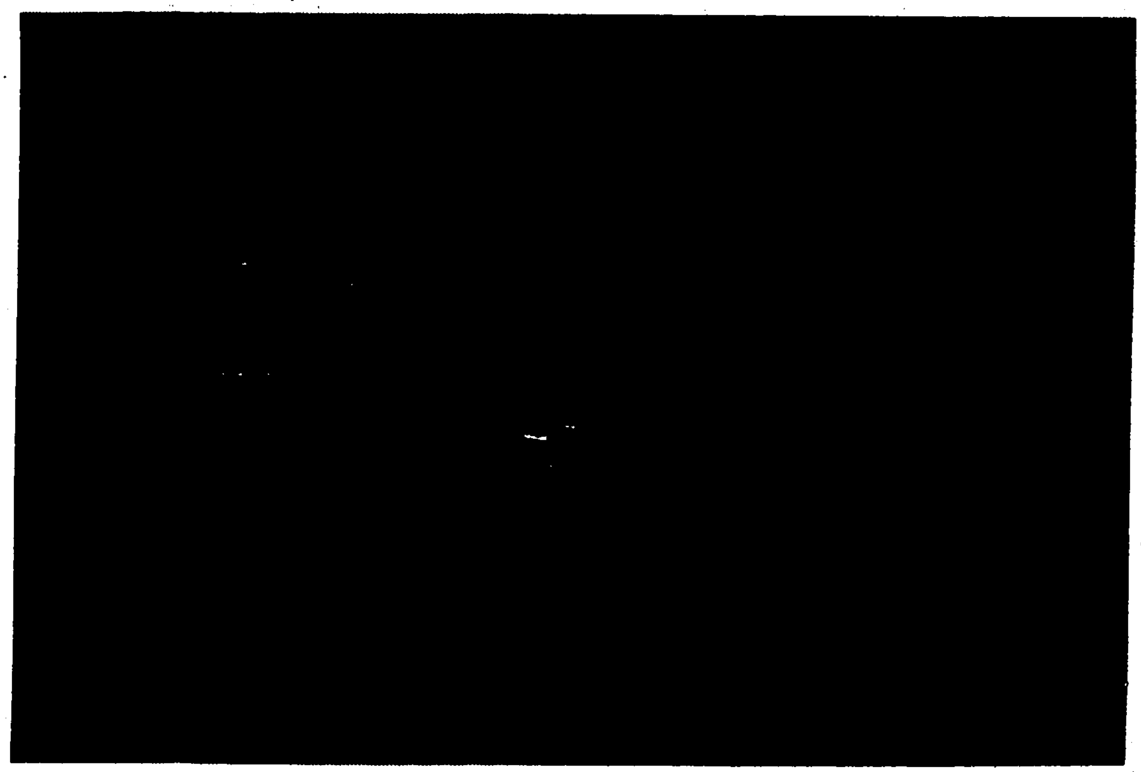


PLATE IV.3 Intermediate map showing the percentage of patrons between 0.01 to 0.99% attending each facility.

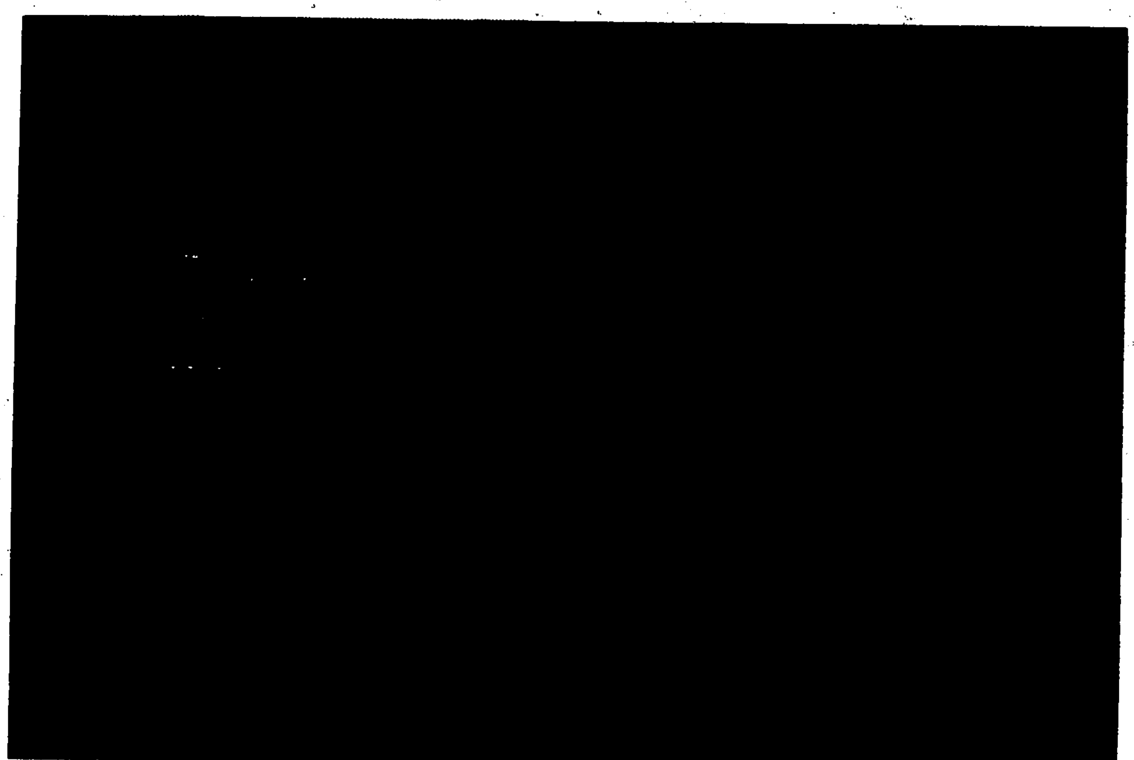


PLATE IV.4 Intermediate map showing the percentage of patrons between 1.00 to 4.99% attending each facility.

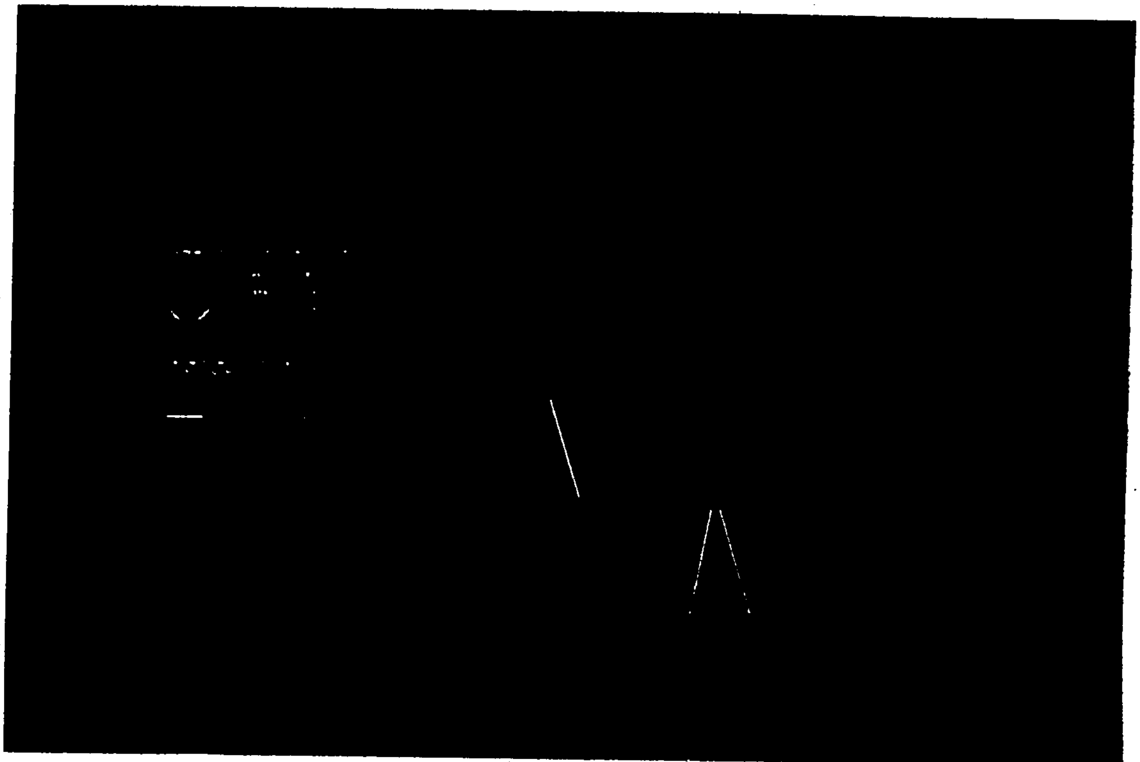


PLATE IV.5 Intermediate map showing the percentage of patrons between 5.00 to 9.99% attending each facility.

PLATE IV.6 Intermediate map showing the percentage of patrons between 10.00 to 24.99% attending each facility.

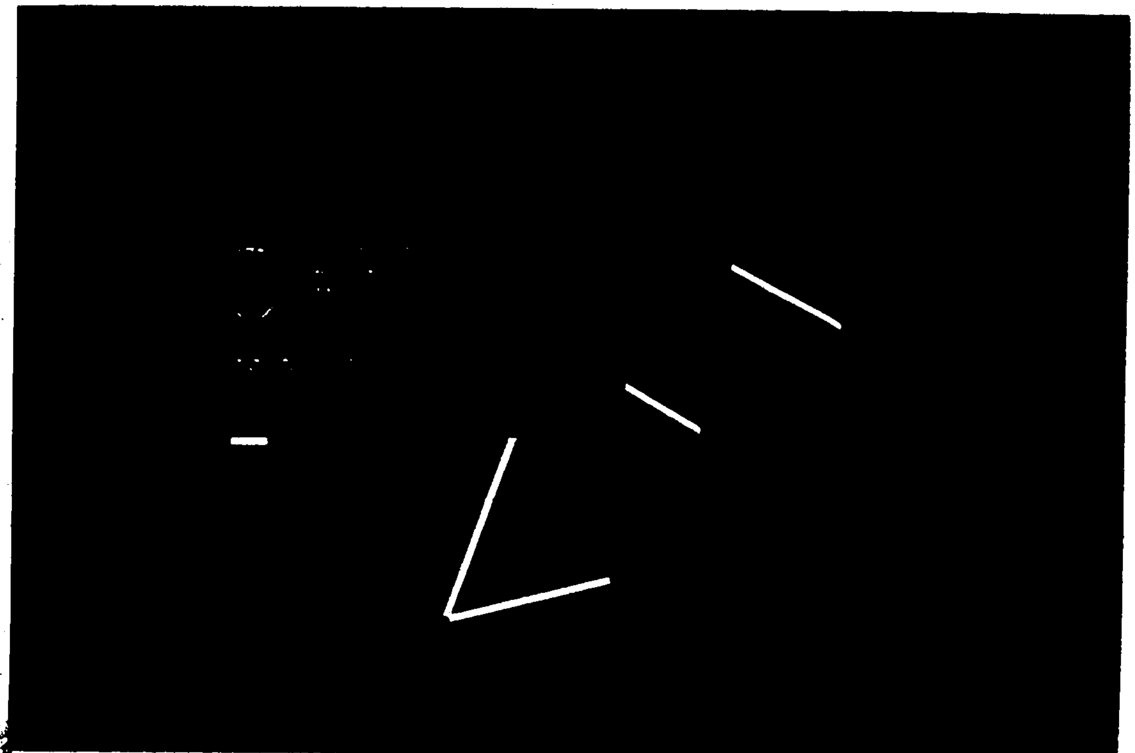


PLATE IV.7 Intermediate map showing the percentage of patrons between 25.00 to 49.99% attending each facility.

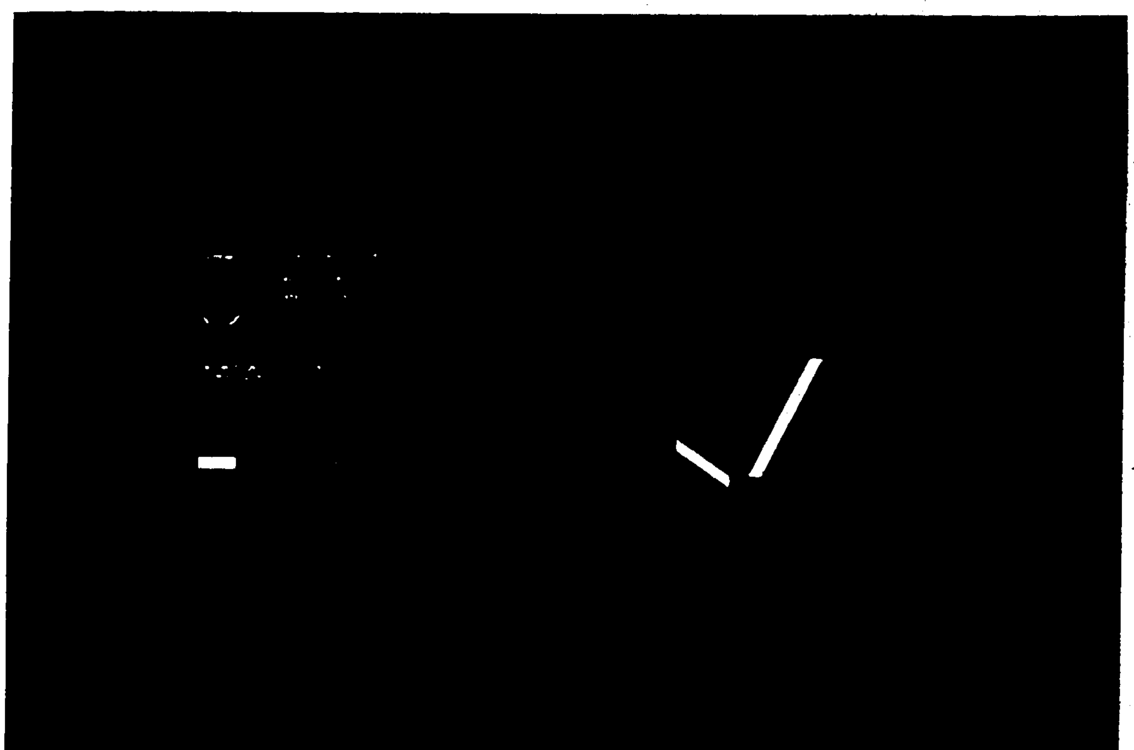


PLATE IV.8 Intermediate map showing the percentage of patrons between 50.00 to 74.99% attending each facility.

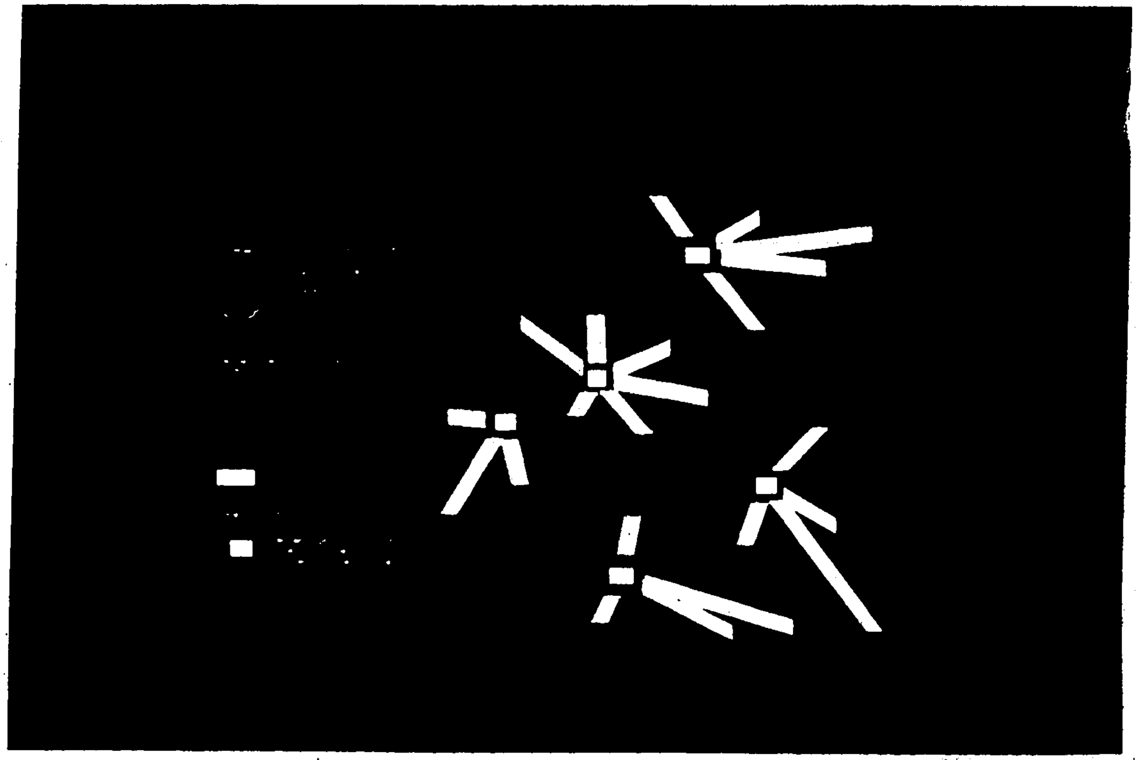


PLATE IV.9 Intermediate map showing the percentage of patrons between 75.00 to 100.00% attending each facility.

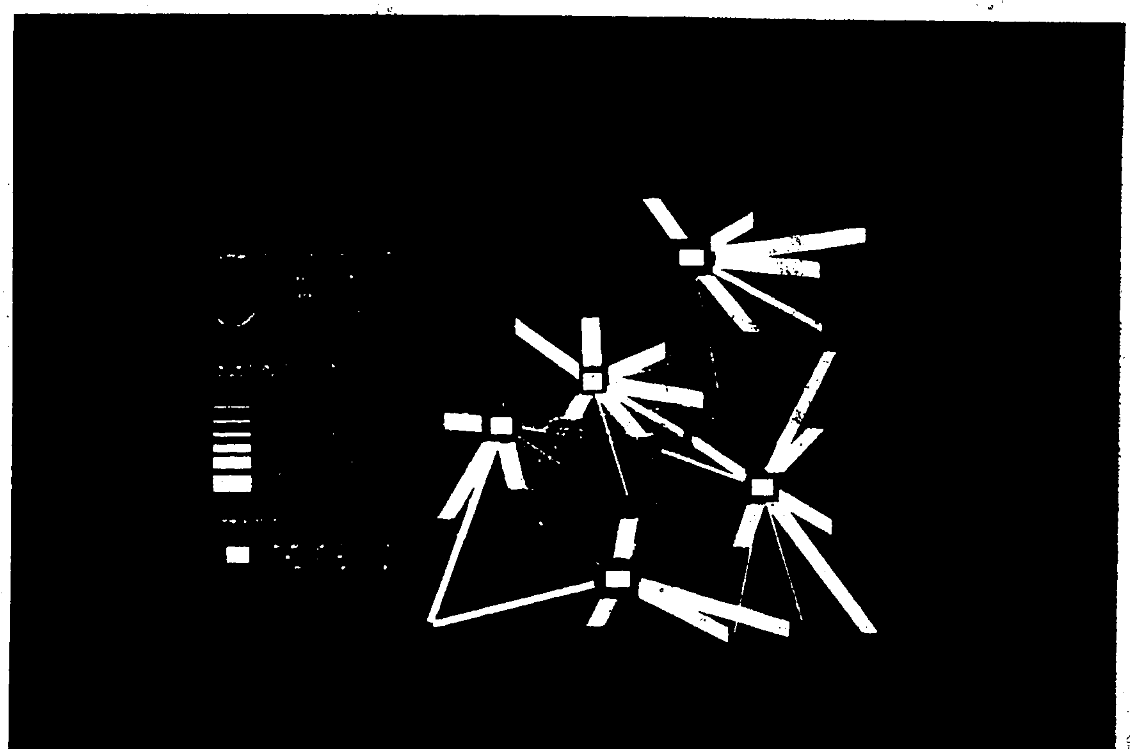


PLATE IV.10 Final map made out of the combination of the seven intermediate maps.

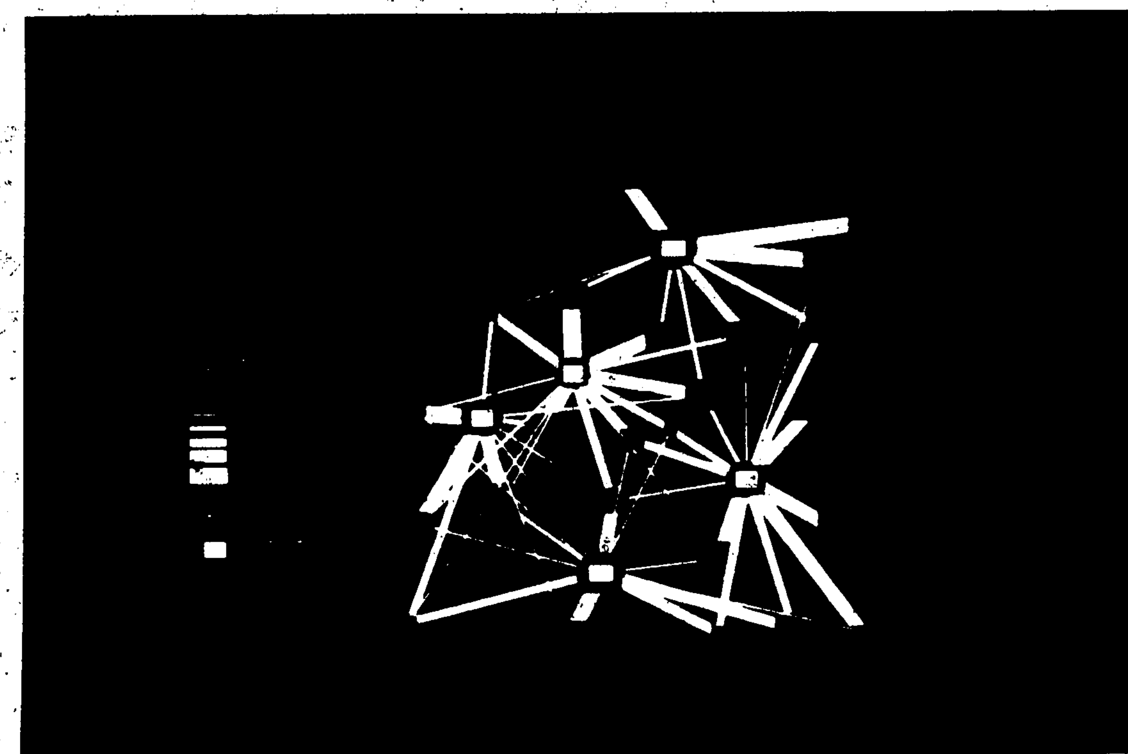


PLATE IV.11 Final map showing an alternative solution from which the two lower classes including the flows of less than 5% have been excluded.

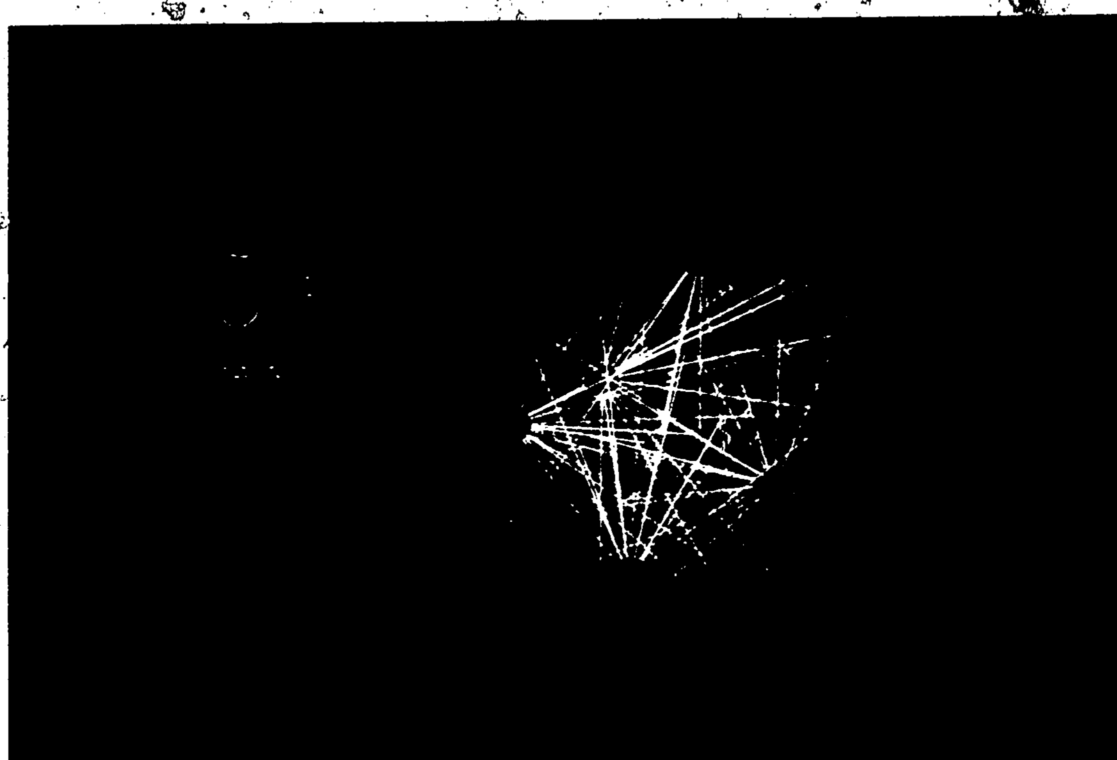


PLATE IV.12 Intermediate map illustrating the pattern complexity being displayed by the two lower classes which were excluded in Plate IV.11.

Plate IV.14 represents the non-classed map. As discussed in Chapter Two, the non-classed map provides a "true" image of the distribution. In this example, the map could adequately function as a communication device. The continuous size range of the circles gives a much more realistic image than the five discrete classes map. Only limited experimentation with the application of unquantized maps to flow-lines data was conducted but as this example demonstrates, the results can be graphically satisfying.

The three final maps (Plates IV.10, IV.13, IV.14) presented for this example yield substantially similar results with the only difference residing in the form in which the information is presented. In other cases, the results could have been significantly different. In general, it might be necessary to choose one of the three approaches, the decision being based on the relative importance attributed to the information transferred by each map while maximizing the visual quality.

Beside these three types of maps especially dedicated for mapping location-allocation solutions, ILAMAP can also be used to generate proportional circle maps and symbol maps. Taken from the same example, Plate IV.15 illustrates the proportional circles map with five class intervals and Plate IV.16 the unquantized proportional circles map. The symbol map, as shown in Plate IV.17, can be of interest when mapping a hierarchical location-allocation solution or representing the facilities locations with different symbols and various scales. The name or number associated with each zone can also be included. In a more general context, the symbol map offers the possibility of placing any desired point symbols and names on maps.

Discussion of the use of IGDS.

Making a map on an interactive graphics system presents some advantages. The map can be displayed on the screen to help monitor complex information and ease data sorting. The speed with which the graphics can be generated and reviewed is much greater than any manual method of drawing. The scale of display is independent of the final output allowing enlargement of map sections for greater detail.

Any displayed feature can be referred to directly and manipulated by a cursor on the screen. Once a point symbol or a line is identified, the required action, such as deletion or displacement, is entered through the appropriate menu command. If one point symbol

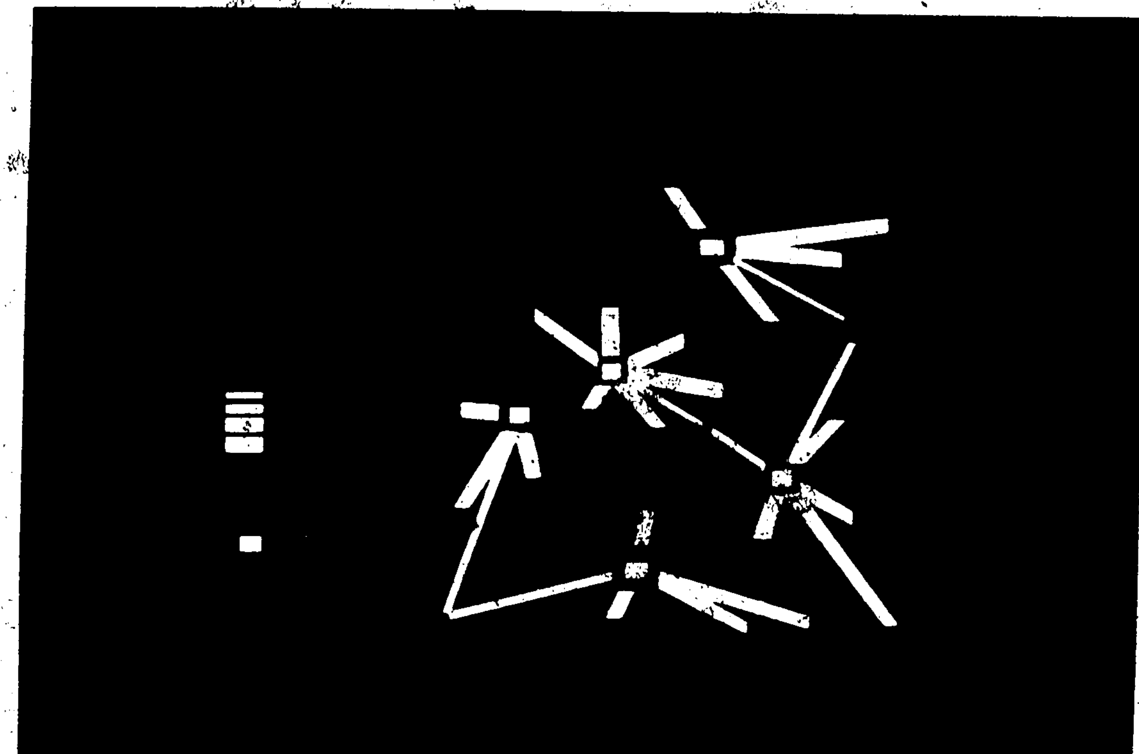


PLATE IV.13 Final map generated with PART 1 of ILAMAP. The non-contiguous class intervals reflect Jenks optimal data classification.

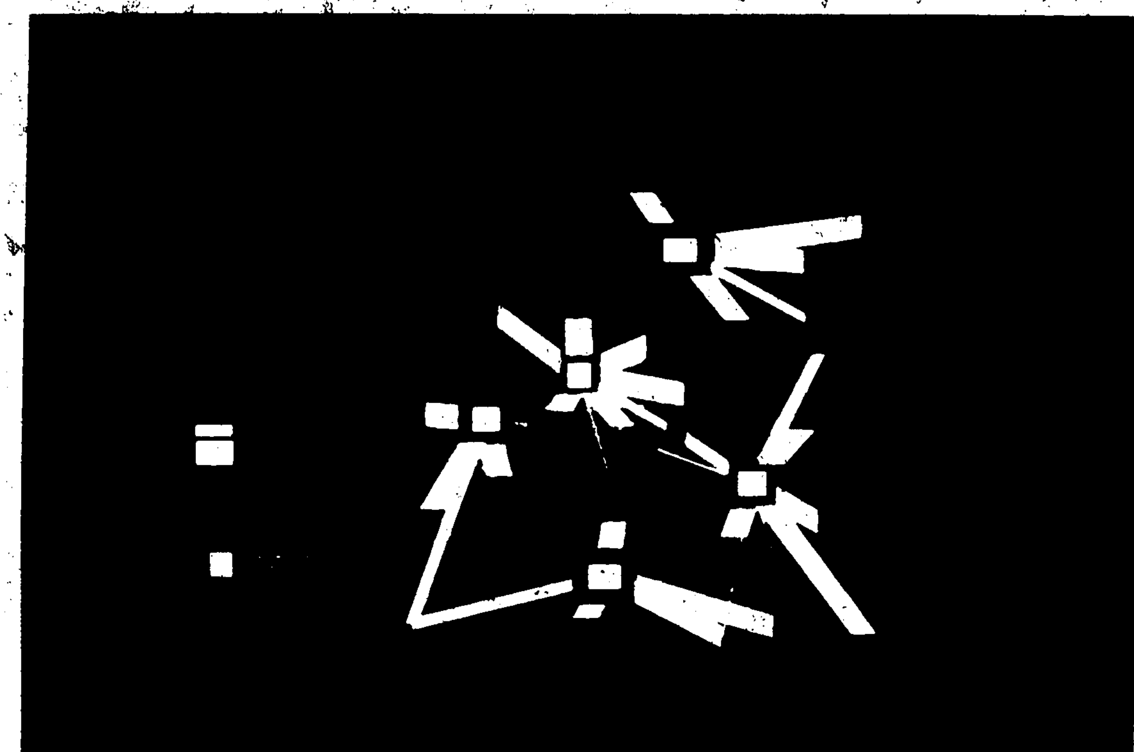


PLATE IV.14 Non-classed map produced with PART 3 of ILAMAP. Relative circle sizes and line widths are assigned to each individual value.

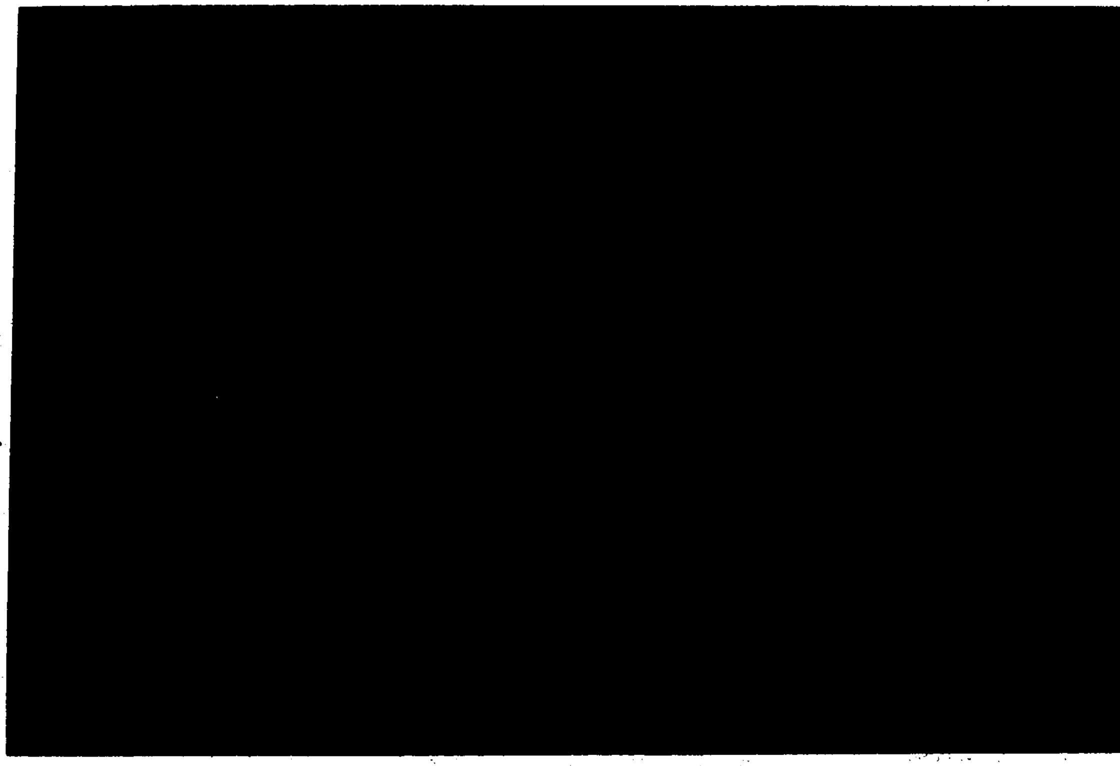


PLATE IV.15 Proportional circles map with five class intervals.

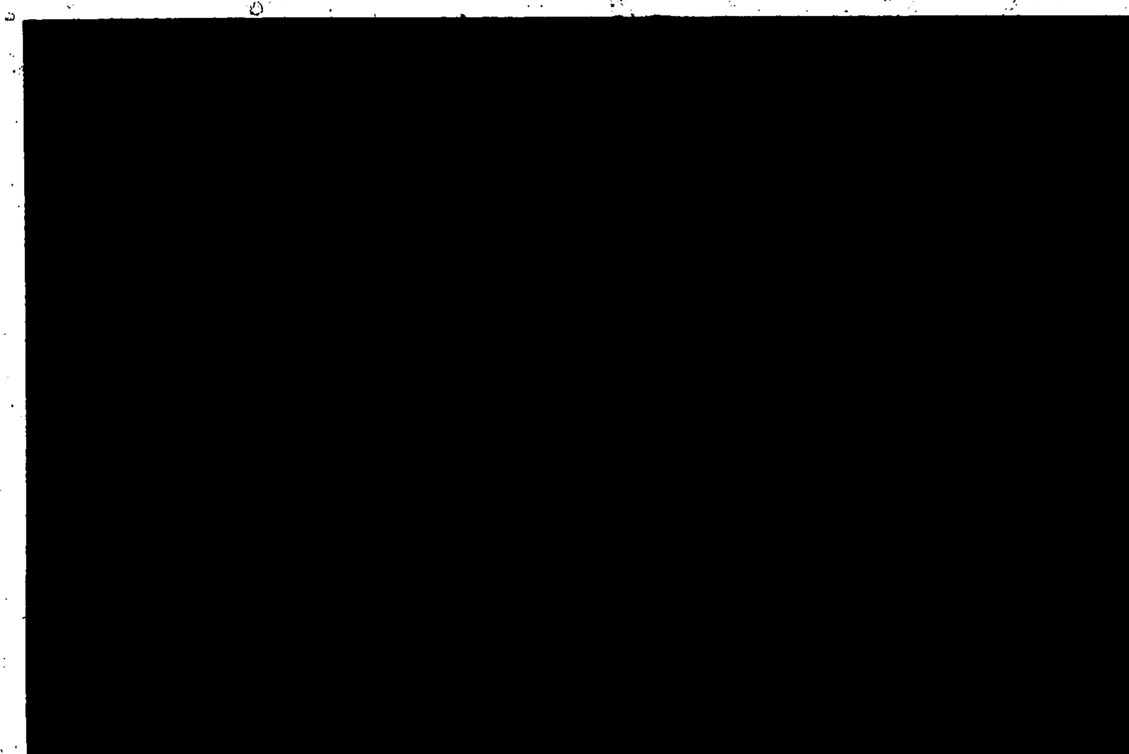


PLATE IV.16 Unquantized proportional circles map.

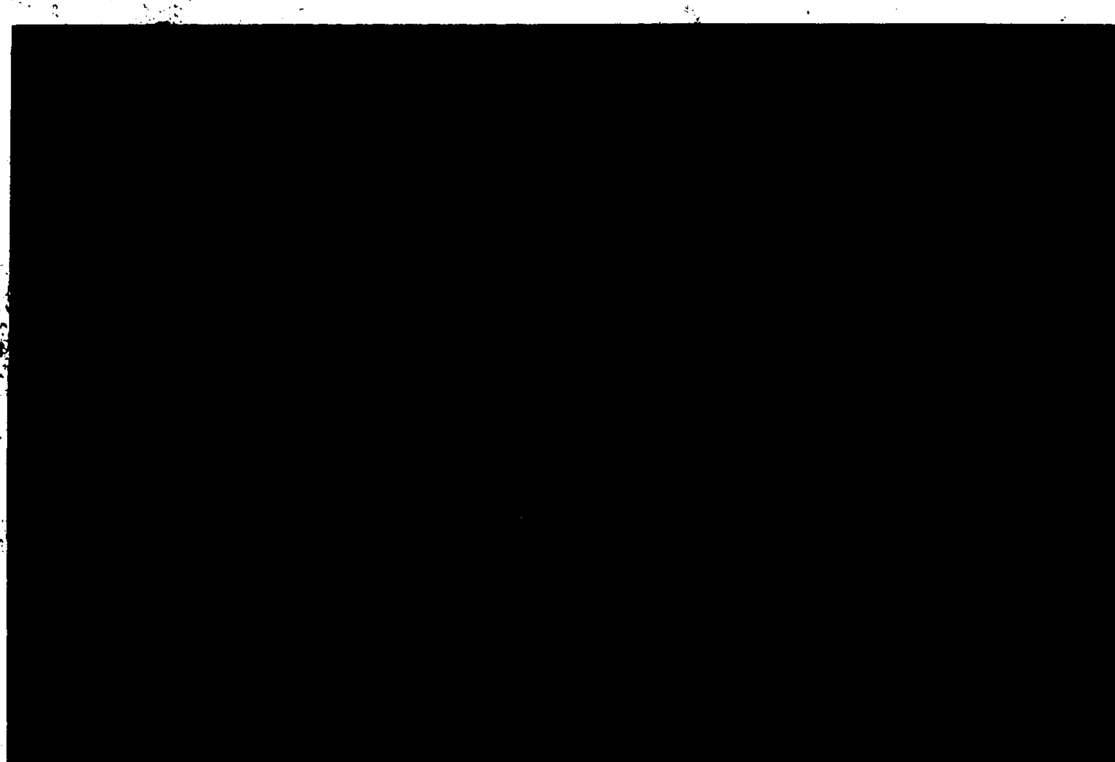


PLATE IV.17 Symbol map representing the facilities locations and attributes. A unique symbol at the same scale results from the consideration of uniform sized facilities in this example.

overlaps another, their positions can be adjusted, and similarly, a section of a line which overlaps another can be re-drawn. Plates IV.18 to IV.21 illustrate the action of displacing a circle and redrawing the flow line in order to improve clarity in the map. All these modifications of the image do not affect the basic data but only the particular map concerned. There is no more need to run the program over again, as would be the case in batch mode, in order to make minor corrections.

The graphic appearance of the map remains to some extent the responsibility of the cartographer. The creation of a legend is a difficult cartographic task to automate on the Intergraph system. The working units definition, establishing a scale relation between the map generated and the design file's units of resolution, is specific to each particular case study and makes the setup of a generalised approach very difficult. But in interactive graphics mode the legend can be easily performed using the graphic cursor on the screen display. The total elimination of the cartographer's subjective input in conventional noninteractive computer mapping applications created boredom and aesthetically unpleasing maps. With interactive graphics, control over map layout and design is left to the cartographer. This allows higher graphic quality map production.

With the actual setup, interaction is possible between the user and ILAMAP while in execution mode, and between the user and graphics after the map is generated. No interactive link remains between the user and the program once graphics mode is initiated. In order to reprocess the data it is necessary to leave the graphics mode. This configuration provides nonetheless an integration of two distinct phases in a cartographic production. Use is made of the full ability of the computer for the processing of repetitive sequential operations to generate a graphic expression of the location-allocation solution while complete control is allowed over the specifications of map layout and design.

B. RESULTS ANALYSIS

Many maps produced with computers are real disasters because emphasis is directed more toward the computer technique than toward the map as a fundamental graphic expression (Muller, 1983). Among the most important graphic elements in map design: clarity, legibility and visual contrast are fundamental to effective cartographic communication.

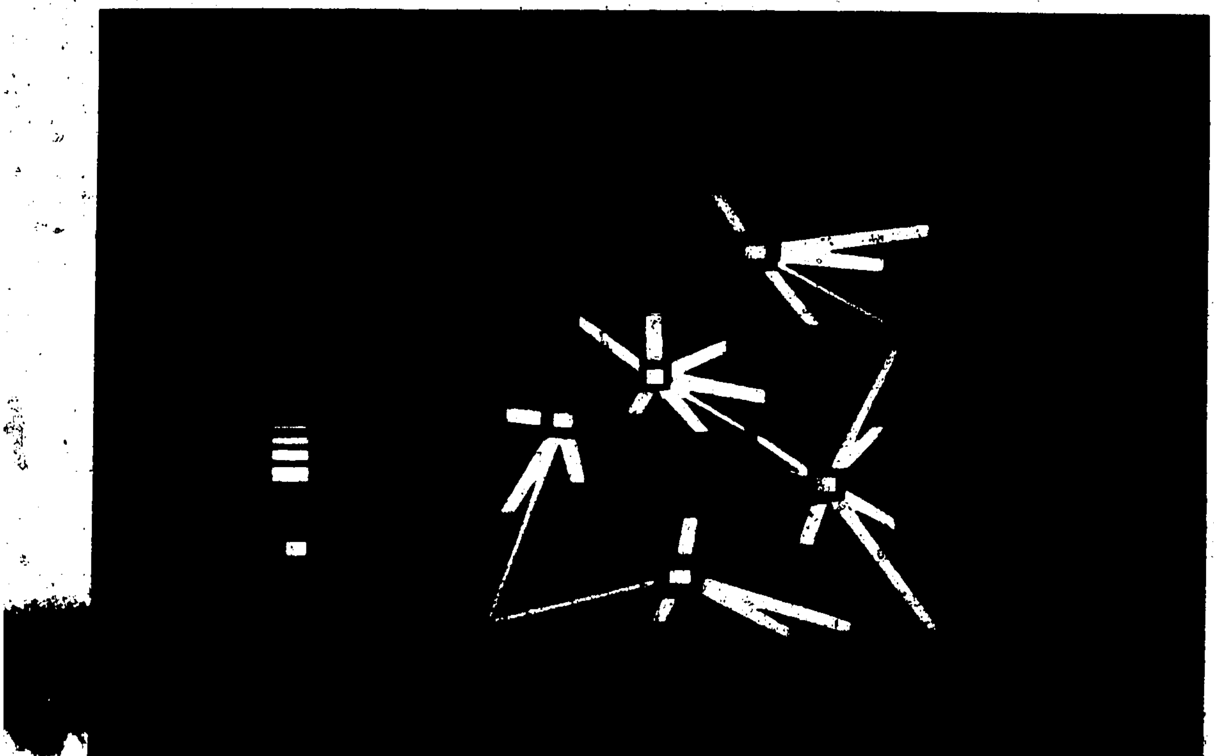


PLATE IV.18 The upper right area of the map presents some confusion: the lines overlap the circle and require some adjustment.

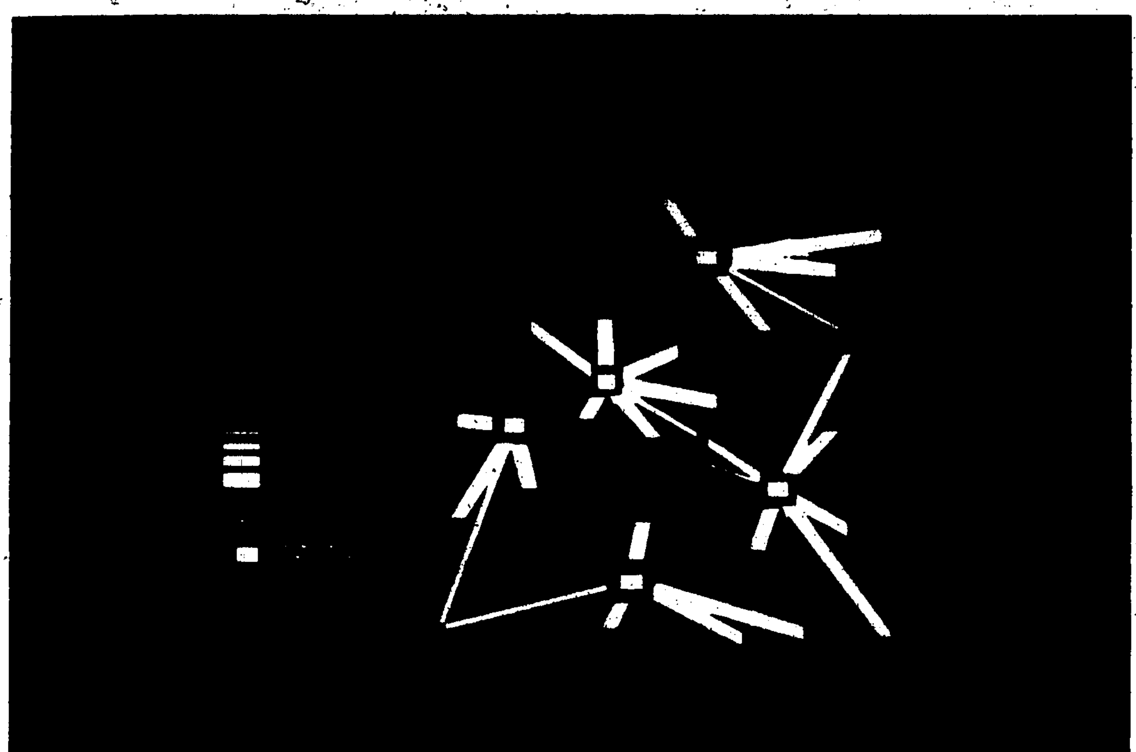


PLATE IV.19 Interactive editing using a cursor on the screen : the circle, shown in white, is identified as a single element.

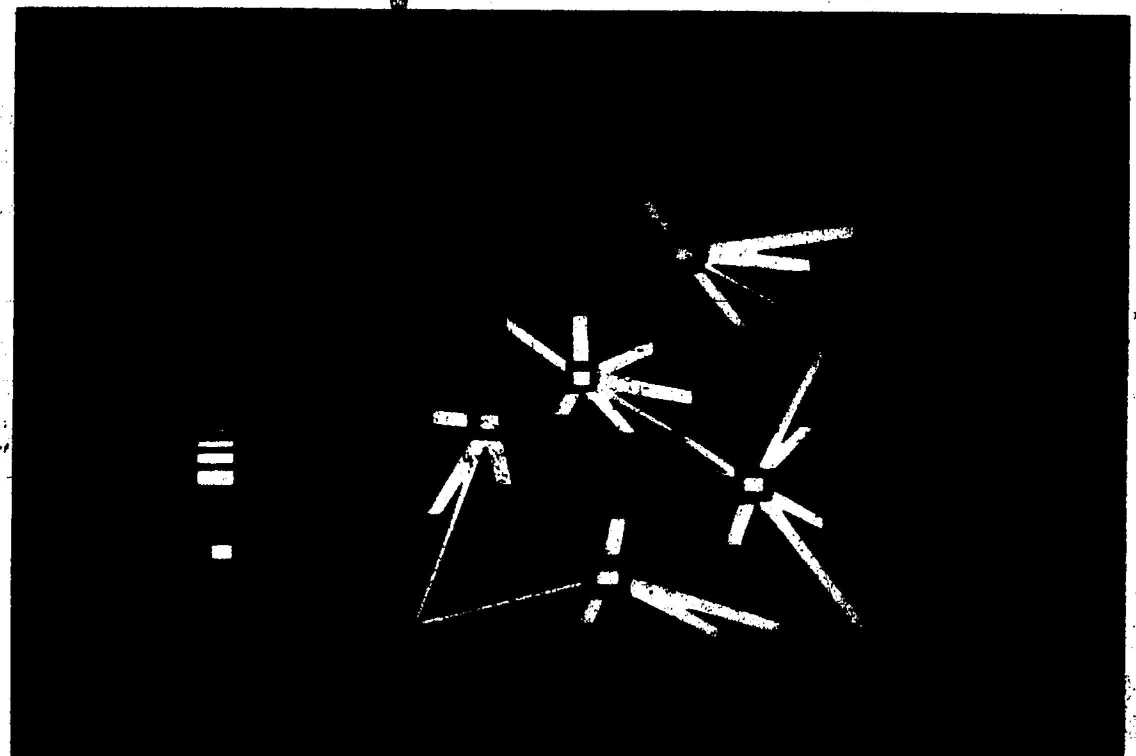


PLATE IV.20 Selecting the MOVE command on the menu, the circle shown in white is moved to a less crowded area of the map.

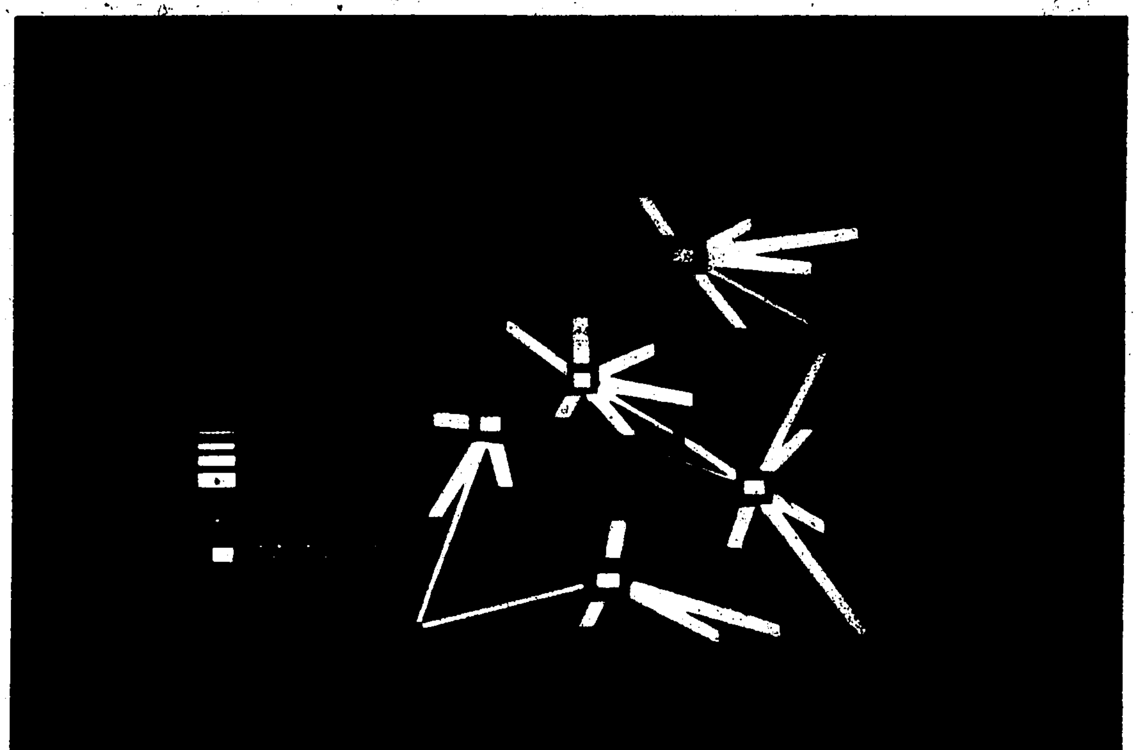


PLATE IV.21 After re-positioning the circle, the line has been re-drawn improving clarity in the map.

Visual quality

Theoretically, the task of achieving clarity and legibility in a map, is "accomplished when the intellectual aspects of the map are not open to doubt or misinterpretation (Robinson, 1978). In the present case, the choice of circles, lines and points symbols seems precise and appropriate for the data being mapped and helps minimize confusion.

The maps have been designed for black-and-white production, despite the fact that a colour computer workstation was used. On a black-and-white map, category differentiation depends entirely on differences in form, and therefore the symbols must exhibit sufficient contrast. The visual contrast refers here especially to the contrast in line width. An occasionnal complication can occur when the lines are combined, or follow similar routes (see Plate IV.18, upper right area of the map). But, as mentioned in the previous section, selective corrections or manipulations can be performed interactively on the screen to overcome that kind of problem.

In some cases for which there is a large variation between the smallest and the largest value to be represented, the limitations imposed by perceptible differences may necessitate restricting both ends of the scale for a better visual contrast. For example, different line symbols (dotted, dashed) can be used for the lower classes.

One important element of visual quality in a computer environment concerns the resolution of the monitors for the temporary maps and the plotter's quality for hard copies. There is a general belief that computer maps are aesthetically unpleasing and are not of "publication quality". The maps presented here demonstrate clearly an improvement which is satisfactory to a certain extent. Figure IV.1 presents an example of hard copy obtained using the electrostatic plotter as output device. As shown in the figure, the resolution still produces quantization or "staircase" effects when lines are drawn at angles other than 0 or 90 degrees and gives, especially, an unpleasant image where the lines wider than 10 or 12 (based on the Intergraph line width range, 0 to 31) are stopped at the circle's edges. Also the resolution may prohibit display of fine details, for example small circles will appear as polygons.

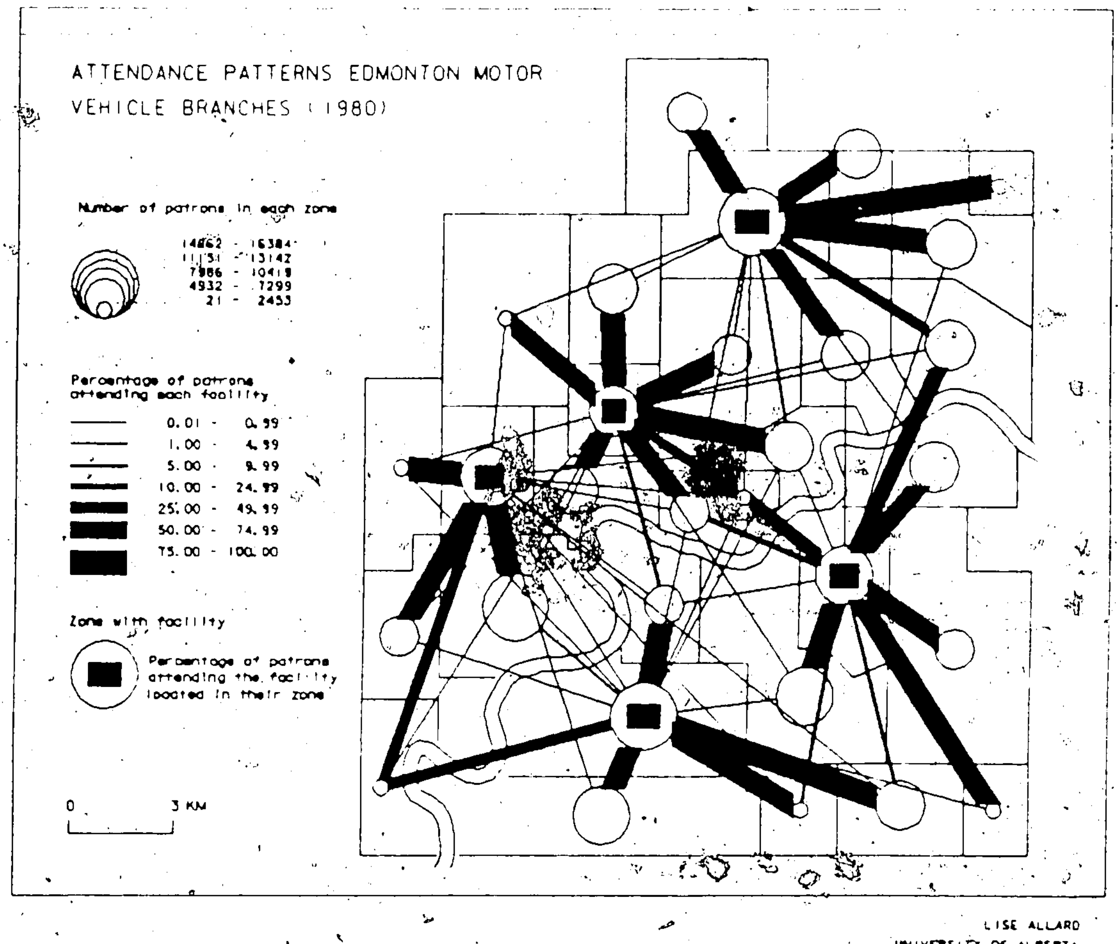


FIGURE IV.1 Hard copy from electrostatic plotter.

Effective cartographic communication

Maps used for particular purposes deal with the problem of communication. The graphic presentation of the information contained in the map is referred to as map design. The function of design is to communicate effectively this information to the user. Graphic design is a vital part of the cartographic process because effective communication requires that the various marks (lines, tones, colours, patterns, lettering, etc.) be carefully modulated and fitted together (Keates, 1973). A map can be ineffective because it is visually confusing to the reader.

The maps presented in this Chapter illustrate how the fundamental objective of communicating some kinds of spatial relationships can be met. The combination of proportional circles and flow lines organize the map so that it makes sense visually. Perceptually the importance of each flow line is reinforced by the circle size. The width of the line gives immediately the importance of the flow coming from a demand zone in which a circle illustrates the total number of patrons involved. Cases in which the assignments of patrons are split between facilities are made obvious. The map evokes in the mind of a reader the desired perceptual image of the spatial phenomena. Graphically the overall character of the distribution is perceptible. The maps thus constitute precious aids in visualizing numerical location-allocation solutions. It encourages location-allocation research with map analysis assistance.

V. CONCLUSIONS AND RECOMMENDATIONS

A. FUTURE ILAMAP IMPROVEMENTS

The ILAMAP program was designed to perform a number of operations which could adequately map location-allocation solutions as well as generate proportional circles maps and symbols maps for various applications. Extension of this program in the future could focus on improving the graphics in order to preserve map clarity and on tying graphic information together with a data base for better data management.

Graphic improvements

When producing flow-maps and symbols maps with a computer program, the major difficulty in preserving map clarity consists of dealing adequately with maps which contain dense clusters of symbols. Traditional solutions to this problem involve minimizing overlaps by repositioning symbols or reducing their size and by the removal of hidden lines where overlaps cannot be eliminated. ILAMAP does not provide any of these solutions. The option left to the user is to accomplish the tasks of repositioning symbols interactively, allowing selective adjustment of symbol locations with the use of a graphic cursor as well as reprocessing ILAMAP with changes in the symbol scaling factor.

Nevertheless, both approaches, the repositioning of symbols and hidden line removal, can be automated and included within ILAMAP in future developments. They could improve map production by reducing the number of individual corrections. The automation of these approaches involves necessarily extensive simultaneous processing. The symbol priorities must be defined according to an appropriate visibility criterion, such as symbol size. Usually smaller symbols have priority over larger ones. The symbols may then be processed and tests made on symbol overlaps based on the defined priorities. Repositioning and scaling of symbols must be performed and each symbol tested once more for overlaps in order to solve for intersections and finally trace the visible portions of symbols outlines. For the flow-lines, if overlaps occur, the intersections of lines must be found by solving equations in order to permit the suppression of hidden portions of the line with a lower priority. This particular problem does not necessitate special attention when solid lines are used, such is the case with ILAMAP, although with empty lines the removal of hidden portions improves clarity in the map. The Intergraph's hidden line removal routine can be applied only on 3-dimensional designs. A new routine would have to be developed for 2-dimensional applications.

Considering the present hardware configuration (PDP 11/70 as CPU) the implementation of subprograms to carry out the above computations would surely face critical storage limitations. Sophisticated programming would be required to overcome storage limitations and still allow reasonnable size problem to be treated. In a long term the acquisition of a new hardware configuration would be the only solution to avoiding compromising major implications.

Data management

The Intergraph supports a Data Management and Retrieval System (DMRS) which offers possibilities to generate various reports from graphics via a predefined data base. ILAMAP could be extended to perform the required processing in order to tie the graphic information together with the data.

Although there is no attribute linkage in the actual program, most variables involved for such processing are already present. DFPI allows control of the attribute data linkage in each Place Element Subroutine. The data base has to be previously set up in DMRS mode before it is retrieved by DFPI. The subroutine DEDFDB must be executed prior to placing the first element with attribute linkage. Each DFPI Place Element Subroutine contains the argument ATLK in the calling statement whose purpose is to provide automatic linkage of attribute data to graphic element. This involves attaching preassigned attribute data with possibility of multiple linkages to the element to be placed and writing the corresponding data record in the data base. The results would be greater data management and retrieval requirements in reviewing the maps. For example: the selection of a specific flow-line on the map on the right screen would display a report on the left screen with all the information attached to that specific line, such as the absolute number of patrons, or percentages assigned to the line, the origin and destination attributes and so on.

The implementation of DMRS within ILAMAP would require deciding on the most appropriate data base for general cases and to allow for maximum flexibility. The researcher could take advantage of relevant information display associated with more efficient data management.

B. EVALUATION OF THE INTERGRAPH SYSTEM

General assessment

Specific graphics devices vary greatly in their capabilities and applications requirements. Our application study leads to the general conclusion that the Intergraph System is more than a sophisticated drafting tool. The possibilities of combining data processing and graphics display make it an invaluable aid in a research and teaching environment.

Technically some familiarity with the system is necessary to perform any work and the number of steps required to generate a map, such as indicated in the ILAMAP User's Manual, can be discouraging to many users. But the interactive menu concept of the Intergraph System, facilitates graphics manipulations and the level concept gives a tremendous display flexibility. The advantages of this system configuration have been illustrated in the example application in Chapter 4, which confirmed our assumption that interactive computer cartography offered more possibilities than traditional cartography to map location-allocation solutions.

The output device available in conjunction with the Intergraph System is an electrostatic plotter. It produces acceptable black and white printout but the resolution still gives staircase effects such as illustrated in Chapter 4 (Figure IV.1). A higher resolution printer could significantly improve the output quality. Regarding the production of colour maps, no colour printer is available with the actual configuration. The use of a camera for hard copies, as presented in this study, is not however always appropriate.

One important feature to be considered in the design of the Intergraph System is that considerable emphasis has been directed toward engineering design requirements since its general use is centered in CAD-CAM (Computer-Assisted Design and Manufacturing). There are only a limited number of software packages dedicated for truly cartographic applications. Among them are those specifically designed for geographic information processing (DTM and GPPU)[21], and for geographic information display (WMS,

---

[21] DTM - Digital Terrain Modeling, GPPU - Graphic Polygon Processing Utilities.

EBSALS and Edge Matching[22]. Software for thematic mapping applications is to a large degree nonexistent and must be user-supplied.

Man-machine interface for cartographic program development

The implementation of ILAMAP has demonstrated the potential of interactive graphics in the production of maps and has simultaneously allowed an evaluation of the Intergraph System interface for cartographic program development. Among the two methods supported by the Intergraph System for interfacing specialized user-written programs to edit design files, the first one which accesses design files resident on disk has been experimented through the implementation of ILAMAP. The second method which performs desired functions in the user's active design file has not been explored in this study.

The Design File Processor Interface (DFPI) is reliable for meeting specific user-defined programming requirements. Its use necessitates, however, a fundamental understanding of the system of internal measurement of the design file which is, unfortunately, badly supported in documentation. In general the functional capabilities of DFPI are efficient. The problem experienced with the File Builder, through the implementation of ILAMAP, showed important limitations but constituted an isolated case. The reason behind this software problem remains unclear and more experimentation would be needed in order to make any generalisations. The compatibility of the DFPI graphics subroutines with the FORTRAN programming language facilitates the development of user-application software. This important factor eases the development of new cartographic programs as well as future implementation of already existing cartographic programs. Hopefully, our contribution in defining the basic procedure required for cartographic programming (Chapter 3) should facilitate future implementation and program development.

Although considerable rewriting would be necessary to implement existing cartographic FORTRAN programs, this effort would bring to hand the full potential of interactive graphics. Two levels of problem have to be considered, however. First, a standardization problem so as to computer installation will have to be dealt with. Old batch

---

[22]WMS - World Mapping System, EBSALS - Elastic Body Small Angle Least Squares.

vector plotting programs work with a specific final map scale and specific pen sizes on a CALCOMP plotter whereas the Intergraph System works in units of resolution totally independent of final scale or size. Secondly, a methodological problem is to be expected. Future attempts should be made with the awareness of program structure differences between interactive and batch programs. As noted by Foley and van Dam (1982): "In general it is difficult to graft an interactive graphics interface onto an existing batch program: the style and structure of an interactive graphics application program are usually quite different from those of a batch program, and good program design therefore requires a before-the-fact, top-down analysis of all the required interaction sequences to lead to a properly structured design and implementation".

Despite these difficulties, there is a need for the implementation of already existing programs in order to broaden the range of cartographic applications and use the Intergraph System as a research tool for geographic information processing and display. The Intergraph System does not support any thematic mapping packages and without cartographic programs to manipulate information and experiment displays, the Intergraph System is reduced to its basic function of a drafting tool. There are more possibilities such those as demonstrated with ILAMAP, where interactive computer cartography can contribute to provide an effective means of displaying information for research applications.

C. POTENTIAL OF INTERACTIVE GRAPHICS IN CARTOGRAPHY

Interactive graphics has the potential of providing important benefits in the production of maps. There is little doubt that computers have influenced all aspects of cartography and will continue to do so in an increasingly important way. Resulting from computer-assistance, trends toward the design and implementation of map types better suited to contemporary and future needs are more and more evident in cartography today.

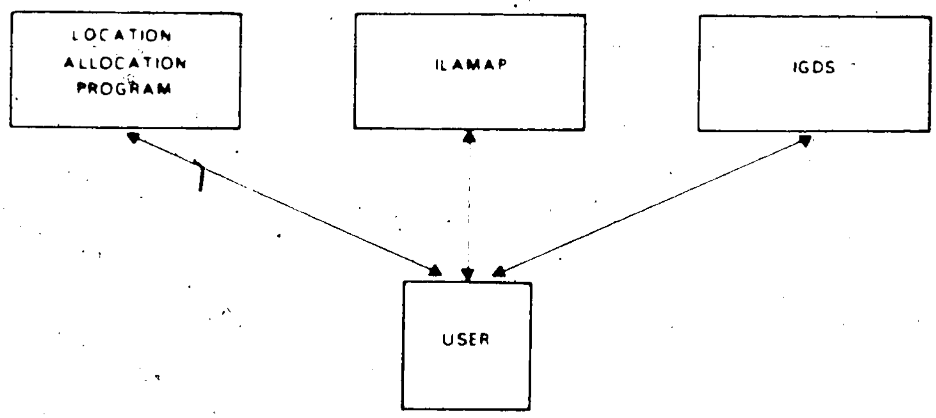
Production of maps

The graphic capabilities of automated and interactive devices studied through the implementation of ILAMAP clearly demonstrated significant potentials of interactive graphics in cartography. The results have been a phenomenal increase of map production speed along with greater flexibility in data manipulation and display. Also the advantages of a combination of manual and computer production methods in map design permitted the production of higher-quality maps. The examples illustrated in Chapter 4 represent but a few of the potential applications of computer cartography.

In the production aspects of cartography, the development of ILAMAP provided the basic cartographic features required for the creation of a specific type of map and provided the cartographer greater flexibility and repeatability. It gave the cartographer the option of reviewing the results of several alternative map representations before selecting a final one. This kind of option is economically unfeasible with manual production. Additionally, last-minute map design changes are possible and easy with interactive devices, whereas with manual methods the lock-in point for the map's design is much earlier. With a batch computer-assistance, considerable loss of time and effort is involved in last-minute changes. This is no longer true with interactive graphics. The time factor crucial to any computer-assisted program, is more efficiently dealt with. Moreover, updated maps can be produced whenever changes in the data occur thus avoiding the traditional problem of producing maps which are outdated by the time they are published.

Interactive graphics allows the cartographer higher man-machine communication in a computer mapping environment. Different levels of interaction can be performed making this communication more or less efficient. Figure V.1 summarizes the levels of interaction in a computer graphics environment. The actual configuration (Figure V.1-A), resulting from the implementation of ILAMAP, shows the integration of data analysis, cartographic processing and information display into a context of parallel interactions. In fact, several location-allocation programs conceived to treat specific problems are available, the cartographic representation of their solutions can be generated with ILAMAP and then IGDS must be initiated to review the results. This configuration illustrates a form of man-machine interaction which combines the features of the interactiveness of alphanumerical communication via online keyboard terminals, with a parallel graphical communication of two-dimensional maps via IGDS. This has great advantages in terms of spatial data analysis and information display. Large amounts of information can be treated and presented within a short space of time thus making a significant improvement in the cartographer's ability to understand data, perceive trends, and visualize real phenomena.

A Actual configuration showing parallel interactions



B Proposed configuration showing a complete integration of interactions

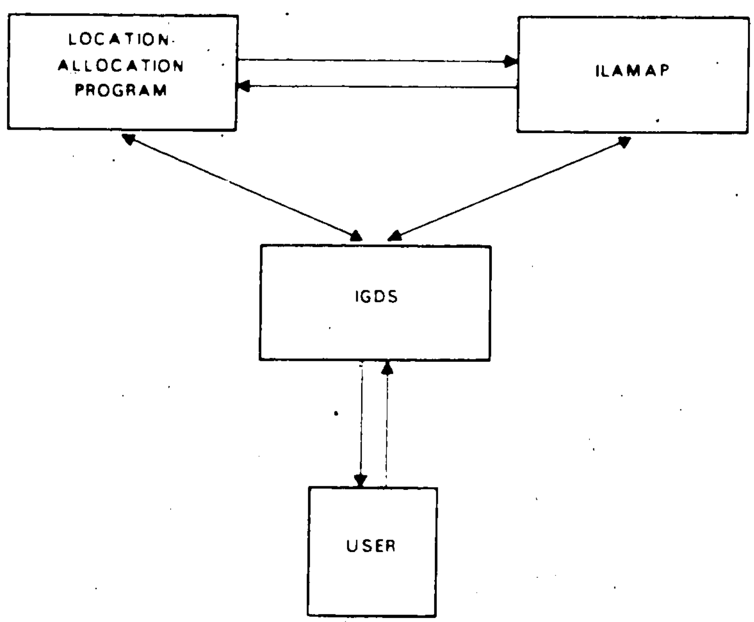


FIGURE V.1 Levels of interaction in a computer graphics environment.

This configuration allows successful use of interactive graphics in mapping location-allocation solutions to support analysis of spatial distributions. It illustrates the potential of interactive graphics in a context of parallel interactions and encourages the pursuit of research applications which could illustrate the benefits of a complete integration of interactions.

Dynamic information processing and display

By using interactive graphics to support analysis of spatial distributions, our study gathers all the components necessary to proceed to a dynamic information processing and display. As Figure V.1-B illustrates, it is now potentially feasible to integrate data analysis, cartographic processing and information display into a functional interactive context. This integration would provide real-time feedback to a user. For example, it would be possible to illustrate on the screen, instantaneously, how changing certain assumptions or parameters of a location-allocation model can drastically distort, or improve the results. Graphical communication would permit visual positioning of a starting solution which would initiate the computation of the allocations, and finally ILAMAP would display the results as they are computed. Interventions would be permitted at any stage of computations in order to exercise a complete control over data processing and adequate cartographic representation. The establishment of links between each component would ensure continuous interaction between data and graphics.

Such a dynamic approach can give graphically useful information. This represents tremendous potential of interactive graphics for handling large volumes of information commonly associated with efficient management of location-allocation decision-making. It is in this context that the potential of interactive cartography would be most fully realized. The capabilities of the Intergraph System to support such dynamic information processing and display could be tested because all the components presently exist. The method of establishing the links is presented as a recommendation for future research applications.

D. RECOMMENDATIONS FOR FUTURE RESEARCH APPLICATIONS

Since interactive graphics appear to provide at least a solution to crucial problems in mapping probabilistic location-allocation solutions, we should anticipate greater emphasis in this area. The possibility of integrating each component within a truly interactive cartographic process (refer Figure V.1) suggests an experimental approach toward the accomplishment of a dynamic information processing and display. Many questions concerning the realisation of a dynamic integration of interactions remain unanswered and necessitate applied research.

The results of our study are expected to encourage research devoted to ascertaining the relative effectiveness of the Intergraph System in the development of 'integrated' interactions. The method we propose for establishing links, which would provide efficient recursive interaction, develops a loop ensuring real-time feedback. For this purpose, the User Command concept of Intergraph would be the technique to use in order to combine many system operations, from data processing to display, under a single control. By definition a user command is a sequence of commands combined together to perform desired functions of numerical or graphical nature in the user's active design file. A specific command language is used for this purpose. DFPI is compatible with it, thus allowing ILAMAP to be used so as to achieve graphical processing. The user command may operate with predefined parameters or allow parameters to be supplied by the user at execution time. Although the Command Interpretor buffer limits the size of a user command, extension seems possible.

This method needs, however, some experimentations and development in order to illustrate concretely the potential benefits as well as the limitations involved in dynamic information processing and display. Future research applications, aimed toward the display of computed data with immediate interaction, are desirable for progress in computer-assisted cartographic methods. These methods are more and more associated with systematic analysis of information in the decision-making process.

E. CONCLUSION

The results of this study provide an example of new possibilities for information display and new working methods leading to solutions to traditional mapping problems. This study has demonstrated the potential of interactive graphics in the production of maps, has permitted the evaluation of the Intergraph System from a cartographic point of view and has introduced a new type of map useful in supporting analysis for location-allocation modelling. It has also underlined the need for more research based on a definition of new needs relevant to the advancement of interactive computer cartography conceived as an essential component of a complete spatial analysis system. The potential of computer mapping is only beginning to be realized.

BIBLIOGRAPHY

BEAUMONT J.R. (1980). "Spatial Interaction Models and the Location-Allocation Problem", *Journal of Regional Sciences* 20, 37-50.

BERTIN Jacques (1977). *La Graphique et le Traitement Graphique de l'Information*, Flammarion, Paris, 277 pages.

BERTIN Jacques (1968). "La Cartographie dans la Civilisation de l'Informatique", *Association Cartographique Internationale, Symposium Technique S40*, New-Delhi, 12 pages.

BICKMORE D.P. (1980). "Future Research and Development in Computer-Assisted Cartography", *The Computer in Contemporary Cartography*, 235-249.

BOYLE R.A. (1981). "Concerns about the Present Applications of Computer-Assisted Cartography", *Cartographica*, vol.18, no.1, Spring 1981, 31-33.

BOYLE R.A. (1976). "The Requirements of an Interactive Display and Edit Facility for Cartography", *The Canadian Cartographer*, vol.13, no.1, June 1976, 35-59.

BRASSEL Kurt (1977). "A Survey of Cartographic Display Software", *International Yearbook of Cartography*, 60-76.

BRASSEL Kurt E., UTANO Jack J. (1979). "Design Strategies for Continuous-Tone Area Mapping", *The American Cartographer*, vol.6, no.1, 39-50.

BURCHI R.S. (1982). "Interactive Graphics Today", *Interactive Computer Graphics Systems*, House William (editor), PBI, 153-179.

CAMPBELL John (1984), *Introductory Cartography*, Prentice-Hall, N.J., 406 pages.

CHANG K.T. (1974). "An Instructional Computer Program on Statistical Class Intervals", *Canadian Cartographer*, vol.11 no.1, 69-77.

CHAREST-BERGLOVE D., McKEAGNEY Don (1983). *Spyder - A FORTRAN Program for Creating Maps of Solutions to Location -Allocation Problems*, Institute of Urban and Regional Planning, University of Iowa, 22 pages.

COOPER L. (1963). "Location-Allocation Problems", *Operations Research 11*, 331-343.

CUFF David J., MATTSON Mark T. (1982), *Thematic Maps, Their Design and Production*, Methuen Inc., 169 pages.

DIGITAL EQUIPMENT CORPORATION (1979), *RSX-11M Beginner's Guide*, Mass.

DIGITAL EQUIPMENT CORPORATION (1979), *RSX-11M/MPLUS Guide to Program Development*, Mass.

DIGITAL EQUIPMENT CORPORATION (1979), *RSX-11M/MPLUS Task Builder Manual*, Mass.

DOBSON Michael W. (1974), "Refining Legend Values for Proportional Circle Maps", *The Canadian Cartographer*, vol.11, no.1, 45-53.

DOBSON Michael W. (1973), "Choropleth Maps without Class Intervals? A Comment", *Geographical Analysis*, vol.3, 358-360.

DUDYCHA D.J. (1978), "Computer Mapping of Socio-Economic Patterns in Metro Toronto", *The Canadian Cartographer*, vol.15, no.1, 23-34.

DUDYCHA D.J. (1981), "The Impact of Computer Cartography", *Cartographica*, vol.18, no.2, Summer 1981, 117-150.

EVANS I. (1977), "The Selection of Class Intervals", *Transactions*, Institute of British Geographers, New series vol.2, no.1, 98-124.

FISHER Howard T. (1982), *Mapping Information, The Graphic Display of Quantitative Information*, Abt Books, Mass., 384 pages.

FISHER Howard T. et al. (1968), *Reference Manual for Synagraphic Computer Mapping (Symap), Version V*, Laboratory for Computer Graphics and Spatial Analysis, Graduate School of Design, Harvard University, Cambridge.

FISHER Walter D. (1958), "On Grouping for Maximum Homogeneity", *Journal of the American Statistical Association*, 789-798.

FLANNERY J.J. (1971), "The Relative Effectiveness of Some Common Graduated Point Symbols in the Representation of Quantitative Data", *The Canadian Cartographer*, vol.8, 96-109.

FOLEY James D., Andries VAN DAM (1982), *Fundamentals of Interactive Computer Graphics*, Addison-Wesley Publishing Co., 664 pages.

GOODCHILD Michael F. (1980), *ILACS: Example Application and Interpretation*, Department of Geography, The University of Western Ontario, 26 pages.

GOODCHILD Michael F., Brian RIZZO (1979), *ILACS Documentation*, Department of Geography, The University of Western Ontario, 16 pages.

GROOP Richard E., and Daniel COLE (1978), "Overlapping Graduated Circles / Magnitude Estimation and Method of Portrayal", *The Canadian Cartographer*, vol.15, no.2, 114-122.

HENNING Mervin D., David HARGREAVES (1983), "Techniques of Computer-Assisted Generalization / Accomodating Subjective Cognition and Objective Logic", *Cartographica*, vol.20, no.4, 55-64.

HODGSON John M. (1981), "A Location-Allocation Model Maximizing Consumer's Welfare", *Regional Studies*, vol.15, no.6, 493-506.

HODGSON John M. (1978), "Toward More Realistic Allocation in Location-Allocation Models: An Interaction Approach, *Environment and Planning A*, vol.10, 1273-1285.

HOUSE William C. (Editor) (1982), *Interactive Computer Graphics Systems*, P.B.I., New-York, 185 pages.

HOLMES J., WILLIAMS F.B., BROWN L.A. (1972), "Facility Location Under Maximum Travel Restriction: An Example Using Day Care Facilities", *Geographical Analysis 4*, 258-266.

INTERGRAPH Co. (1981), *IGDS Application Software Interface Document*, no.79-076C.

INTERGRAPH Co. (1981), *Interactive Graphics Design System (IGDS) Operating Manual*, no.79-067C.

INTERGRAPH Co. (1981). *Proposal for an Intergraph Interactive Graphic System*

JENKS George F. (1977). *Optimal Classification for Choropleth Maps*. Department of Geography, University of Kansas.

JENKS George F. (1976). "Contemporary Statistical Maps - Evidence of Spatial and Graphic Ignorance". *The American Cartographer*, vol.3, no.1, 11-19.

JENKS George F., Fred C. CASPALL (1971). "Error on Choroplethic Maps: Definition, Measurement, Reduction". *Annals of the Association of American Geographers*, vol.61, no.2, June 1971, 217-244.

JENKS George F., Michael COULSON (1963). "Class Intervals for Statistical Maps". *International Yearbook of Cartography 3*, 119-134.

KEATES J.S. (1982). *Understanding Maps*. Longman Group Ltd. 139 pages.

KEATES J.S. (1973). *Cartographic Design and Production*. Longman Group Ltd. 240 pages.

KERN R., G. RUSHTON (1969). "MAPIT: A Computer Program for Production of Flow Maps, Dot Maps and Graduated Symbols Maps". *The Cartographic Journal*, vol.6, 131-137.

LAWRENCE G.R.P. (1971). *Cartographic Methods*, Methuen & Co. Ltd. 162 pages.

LEA Anthony C. (1978). *A Model Taxanomy and a View of Research Frontiers in Normative Locational Modelling*. Presented to the International Symposium on Locational Decisions, Banff, Alberta. 166 pages.

LEA Anthony C. (1973). *Location-Allocation Systems: An Annotated Bibliography*. Department of Geography, University of Toronto, Discussion Paper no.13, 274 pages.

LEONARDI G. (1981). "A Unifying Framework for Public Facility Location Problems - PART 1: A Critical Overview and some Unsolved Problems". *Environment and Planning A*, vol.13, 1001-1028.

MacEACHREN Alan M. (1979). "The Evolution of Thematic Cartography: A Research Methodology and Historical Review". *The Canadian Cartographer*, vol.16, no.1, 17-33.

MAGNENAT-THALMANN N., D. THALMANN (1982). "La Conception des Cartes Géographiques assistées par ordinateur". *Cartographica*, vol.19, no.1, 41-50.

MEIHOFER Hans-Joachim (1973). "The Visual Perception of the Circle in Thematic Maps: Experimental Results". *The Canadian Cartographer*, 10, 63-84.

MOELLERING Harold (1980). "Strategies of Real-Time Cartography". *The Cartographic Journal*, vol.17, no.1, 12-15.

MONMONIER Mark S. (1982). *Computer-Assisted Cartography: Principles and Prospects*. Prentice-Hall Inc., 214 pages.

MONMONIER Mark S. (1977). *Maps, Distorsion and Meaning*, Resource Paper, no.75-4, Association of American Geographers, 51 pages.

MORRISON Joel L. (1980). "Computer Technology and Cartographic Change". *The Computer in Contemporary Cartography*, Edited by D.R.F. Taylor, John Wiley & Sons Ltd, 5-23.

MORRISON Joel L. (1974). "Changing Philosophical-Technical Aspects of Thematic Cartography". *The American Cartographer*. Vol. 1 no. 1. 5-14.

MUEHRCKE Phillip (1972). *Thematic Cartography*. Resource Paper, no. 19. Association of American Geographers. 66 pages.

MULLER Jean-Claude (1983). "Ignorance Graphique ou Cartographie de l'Ignorance". *Cartographica*. vol. 20. no. 3. 17-30.

MULLER Jean-Claude (1981). "Bertin's Theory of Graphics / A Challenge to North American Thematic Cartography". *Cartographica*. vol. 18. no. 3. 1-8.

MULLER Jean-Claude (1979). "Perception of Continuously Shaded Maps". *Annals of the Association of American Geographers*. 240-249.

NEWMAN William M. and Robert F. SPROUL (1979). *Principles of Interactive Computer Graphics*. 2nd Edition. McGraw Hill. 541 pages.

O'KELLY M.E. and J.E. STORBECK (1984). "Hierarchical Location Models with Probabilistic Allocation". *Regional Studies*. vol. 18.2. 121-129.

PEUCKER Thomas K. (1972). *Computer Cartography*. Resource Paper no. 17. Association of American Geographers. 75 pages.

REVELLE C.S., R.W. SWAIN (1970). "Central Facilities Location". *Geographical Analysis*. 2. 30-42.

RIFFE P.D. (1970). "Conventional Map, Temporary Map, or Non-map?". *International Yearbook of Cartography*. 10. 95-103.

ROBINSON Arthur et al. (1978). *Elements of Cartography*. 4th Edition. John Wiley & Sons. 448 pages.

ROBINSON A.H. and SALE R.D. (1969). *Elements of Cartography*. 3rd Edition. New-York.

ROBINSON A., B.B. PETCHENIK (1975). "The Map as a Communication System". *The Cartographic Journal*. vol. 12. no. 1. 7-15.

SCOTT Allen J. (1971). *An Introduction to Spatial Allocation Analysis*. Resource Paper no. 9. Association of American Geographers. 36 pages.

SCOTT Allen J. (1970). "Location-allocation Systems: a Review". *Geographical Analysis*. 2. 95-119.

STUTZ F.P. (1975). "Interactive Computer Cartography". *The Cartographic Journal*. vol. 12. no. 1. 17-21.

TAYLOR Fraser D.R. (Editor) (1980). *The Computer in Contemporary Cartography*. vol. 1. John Wiley & Sons. 252 pages.

THALMAN et al. (1982). "ELECT: An Interactive Graphical System for the Automatic Generation of Electoral Maps". *Cartographica*. vol. 19. no. 1. 28-40.

TOBLER W.R. (1979). *A Geographic Flow Mapping Program*. Dept. of Geography. University of California.

TOBLER W.R. (1973). "Choropleth Maps without Class Intervals". *Geographical Analysis*. 3. 262-265.

WAUGH T.C., D.R.F. TAYLOR (1976). "GIMMS / An Example of an Operational System for Computer Cartography". *The Canadian Cartographer*. 158-166.

WILLIAMS Robert L. (1976). "The Misuse of Area in Mapping Census-type Numbers". *Historical Methods Newsletter*, vol. 19, no.4, 213-216.

WILSON A.G. (1971). "A Family of Spatial Interaction Models and Associated Developments". *Environment and Planning*, 3, 1-32.

Personal Communications

M.F. GOODCHILD, Professor
Department of Geography
The University of Western Ontario
London, Ontario

A.C. LEA, Ph.D. Research Director
Compusearch Market and Social Research Limited
Toronto, Ontario

G. RUSHTON, Professor
Department of Geography
The University of Iowa
Iowa City, U.S.A.

N. WATERS, Professor
Department of Geography
The University of Calgary
Calgary, Alberta

E. WILTS, Software Specialist
Intergraph Software Specialists Services
Intergraph Corporation
Calgary, Alberta

APPENDIX 1. II. AMAP : Program listing

```
C     ***********************************************
C
C
C     R O O T   S E G M E N T
C
C
C     ***********************************************
C
C     VARIABLES        DESCRIPTION
C     ---------        -----------
C     IOP              ARRAY OF PROGRAM'S OPTIONS
C     K1               NUMBER OF CLASSES FOR CIRCLES
C     K2               NUMBER OF CLASSES FOR LINES
C     PERL             ARRAY OF LARGEST VALUES
C     PERS             ARRAY OF SMALLEST VALUES
C     ZL               ARRAY OF LARGEST VALUES
C     ZS               ARRAY OF SMALLEST VALUES
C
C     ...............................................
C
C     PROGRAM NAME  : CNTRL.FTN
C
C     LANGUAGE      : FORTRAN
C     SYSTEM        : PDP11/70 RSX-11M-PLUS
C     AUTHOR        : LISE ALLARD
C     UIC           : [266,10]
C     LAST UPDATED  : JULY 1984
C
C     COMPILE       : >F77 ILACNTRL.OBJ,ILACNTRL.LST/-SP=ILACNTRL.FTN/TR:NONE
C     TASK BUILDER OVERLAY DESCRIPTOR : ILAMAP.ODL
C
C     TASK BUILD    : >TKB @ILAMAP.CMD
C
C     PURPOSE       : CNTRL.FTN IS THE MAIN ROOT SEGMENT
C                     AND CONTROLS THE PROGRAM'S OPTIONS.
C
C     DFPI SUBROUTINES USED :
C     NONE
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     INTRO
C     INPUT
C     P1SET
C     P2SET
C     P3SET
C     SYMBOL
C
C     ...............................................
C
C     VARIABLE TYPE DECLARATION
```

```
C       ---------- ---------------
C
        COMMON /BLK1/ ZL(10), ZS(10), PERL(10), PERS(10), K1, K2
        COMMON /BLK4/ IOP(4)
        COMMON /IREQ/ IREQ
        COMMON /INFLAG/ INFLAG
        COMMON /WAITFL/ WAITFL
        COMMON /APREGN/ APREGN
        COMMON /DFPI/ DFPI
C
        REAL ZL, ZS, PERL, PERS
        INTEGER K1, K2, IOP
C
        CALL INTRO
C
        IF (IOP(1) .EQ. 1) THEN
          CALL INPUT
          CALL P1SET
        END IF
C
        IF (IOP(2) .EQ. 1) THEN
          CALL P2SET
        END IF
C
        IF (IOP(3) .EQ. 1) THEN
          CALL P3SET
        END IF
C
        IF (IOP(4) .EQ. 1) THEN
          CALL SYMBOL
        END IF
C
     10 CONTINUE
        STOP ' * PROGRAM COMPLETED * '
        END
C
C       ***********************************************
C
C
C       O V E R L A Y   S E G M E N T   O N E
C
C
C       ***********************************************
C
C       VARIABLES       DESCRIPTION
C       ---------       -----------
C       IOP             ARRAY OF PROGRAM'S OPTIONS
C
C       ...............................................
C
C       SUBROUTINE INTRO
```

```
C
C     LANGUAGE       : FORTRAN
C     SYSTEM         : PDP11/70 RSX-11M-PLUS
C     AUTHOR         : LISE ALLARD
C     UIC            : [266,10]
C     LAST UPDATED   : JULY 1984
C
C     COMPILE        : >F77 ILAINTRO.OBJ,ILAINTRO.LST/-SP=ILAINTRO.FTN/TR:NONE
C
C     PURPOSE          INTRODUCTION TO THE PROGRAM AND ITS
C                      OPTIONS.
C
C     DFPI SUBROUTINES USED :
C     NONE
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     NONE
C
C     ...............................................................
C
      SUBROUTINE INTRO
C
      COMMON /BLK4/ IOP(4)
C
      INTEGER*2 IOP
C
C     PROGRAM IDENTIFICATION
C     ----------------------
      WRITE (5,10)
   10 FORMAT (///, ' *** ILAMAP : INTERACTIVE LOCATION-',
     1        'ALLOCATION MAPPING PROGRAM ***', ///, 10X,
     2        'BY LISE ALLARD', /, 10X, 'UNIVERSITY OF ALBERTA', ///,
     3        10X, 'THIS PROGRAM IS SET UP IN 4 PARTS :', //, 10X,
     4        'PART 1 - CREATE A MAP USING JENKS''DATA CLASSIFICATION', /
     5        /, 10X, 'PART 2 - CREATE A MAP WITH YOUR OWN CLASS INTERVALS
     6 ', //, 10X, 'PART 3 - CREATE A NON-CLASSED MAP', //, 10X,
     7        'PART 4 - CREATE A SYMBOL MAP')
C
C     SET PROGRAM'S OPTIONS
C     ---------------------
      WRITE (5,20)
   20 FORMAT (///, ' ENTER OPTIONS :', /, 5X, ' 0 = NO', /, 5X,
     1        ' 1 = YES ')
      WRITE (5,30)
   30 FORMAT (//, '$USE OF JENKS'' CLASSIFICATION = ')
      READ (5,40) IOP(1)
   40 FORMAT (I1)
      WRITE (5,50)
   50 FORMAT (/, '$SET YOUR OWN CLASS INTERVALS = ')
      READ (5,40) IOP(2)
      WRITE (5,60)
```

```
   60 FORMAT (/, '$CREATE A NON-CLASSED MAP = ')
      READ (5,40) IOP(3)
      WRITE (5,70)
   70 FORMAT (/, '$CREATE A SYMBOL MAP = ')
      READ (5,40) IOP(4)
C
      RETURN
      END
C
C     *********************************************
C
C
C     O V E R L A Y   S E G M E N T   T W O
C
C
C     *********************************************
C
C     VARIABLES      DESCRIPTION
C     ---------      -----------
C     INFILE         DATA FILE NAME
C     K1             NUMBER OF CLASSES FOR CIRCLES
C     K2             NUMBER OF CLASSES FOR LINES
C     N              NUMBER OF ZONES
C     NC             NUMBER OF CHARACTERS
C     NPER           NUMBER OF PERCENTAGES
C     PER            ARRAY OF PERCENTAGES
C     PERL           ARRAY OF LARGEST VALUES
C     PERS           ARRAY OF SMALLEST VALUES
C     ZL             ARRAY OF LARGEST VALUES
C     ZONE           ARRAY OF ZONE NUMBERS
C     ZS             ARRAY OF SMALLEST VALUES
C     ZVAL           ARRAY OF WEIGHTS
C
C     .............................................
C
C     SUBROUTINE INPUT
C
C     LANGUAGE     : FORTRAN
C     SYSTEM       : PDP11/70 RSX-11M-PLUS
C     AUTHOR       : LISE ALLARD
C     UIC          : [266,10]
C     LAST UPDATED : JULY 1984
C
C     COMPILE      : >F77 ILAINPUT.OBJ,ILAINPUT.LST/-SP=ILAINPUT FTN/TR:NONE
C
C     PURPOSE      : READ DATA AND CALL JENK FOR AN
C                    OPTIMAL DATA CLASSIFICATION.
C
C     DFPI SUBROUTINES USED :
C     NONE
C
```

```
C      SUBROUTINES AND FUNCTIONS REQUIRED :
C      JENK
C
C      .............................................
C
       SUBROUTINE INPUT
C
       COMMON /BLK1/ ZL(10), ZS(10), PERL(10), PERS(10), K1, K2
C
       INTEGER*2 ZONE(50), INFILE(31), N, NPER, K1, K2
       REAL ZVAL(50), PER(50), ZL, ZS, PERL, PERS
       CHARACTER*20 FMT
C
C      PROGRAM IDENTIFICATION
C      ---------------------
       WRITE (5,10)
    10 FORMAT (/////, 10X, ' * PART 1-A : JENKS'' DATA ',
     1        'CLASSIFICATION *')
C
C      KEY-IN DATA FILE NAME
C      ---------------------
       WRITE (5,20)
    20 FORMAT (//, '$DATA FILE NAME= ')
       READ (5,30) NC, INFILE
    30 FORMAT (Q, 31A1)
       IF (NC .EQ. 0) STOP
       IF (NC .GT. 30) NC = 30
       INFILE(NC + 1) = 0
       OPEN (UNIT=2,NAME=INFILE,TYPE='OLD',READONLY,ERR=82)
       GO TO 60
C
    40 WRITE (5,50)
    50 FORMAT (/, ' PROBLEM OPENING DATA FILE')
       STOP
C
C      READ THE NUMBER OF ZONES
C      ------------------------
    60 CONTINUE
       READ (2,70) N
    70 FORMAT (I3)
C
C
C      READ THE ZONES' NUMBERS
C      -----------------------
       DO 90 I = 1, N
         READ (2,80) ZONE(I)
    80   FORMAT (1X, I3)
    90 CONTINUE
C
   100 CONTINUE
       WRITE (5,110)
```

```
  110 FORMAT (///, '$CALCULATE CLASS INTERVALS FOR CIRCLES?(Y/N) ')
      READ (5,120) ANSER1
  120 FORMAT (A1)
      IF (ANSER1 .EQ. 'N') GO TO 150
C
C
C     READ DATA VALUES
C     ----------------
      DO 140 I   1, N
        READ (2, 30) ZVAL( )
  130   FORMAT (F10 2)
  140 CONTINUE
C
          [illegible]        ZVAL, ZONE, ZL, ZS)
C
  1
        TE  5,160)
       ORMAT (///, '$CALCULATE CLASS INTERVALS FOR FLOW LINES?(Y/N) ')
      READ (5,120) ANSER2
      IF (ANSER2 .EQ. 'N') GO TO 190
C
C     [illegible]D THE NUMBER OF PERCENTAGES
C     ------------------------------
      READ (2,70) NPER
C
C     READ PERCENTAGES OF FLOW
C     ------------------------
      DO 180 I = 1, NPER
        READ (2,170) PER(I)
  170   FORMAT (F6.2)
        ZONE(I) = I
  180 CONTINUE
C
      CALL JENK(NPER, K2, PER, ZONE, PERL, PERS)
C
  190 CONTINUE
      CLOSE (UNIT=2,DISPOSE='SAVE')
      RETURN
      END
C
C     ...................................................
C
C     SUBROUTINE JENK
C
C     LANGUAGE      : FORTRAN
C     SYSTEM        : PDP11/70 RSX-11M-PLUS
C     AUTHOR        : GEORGE F. JENKS (1977), UNIVERSITY OF KANSAS
C     UIC           : [266,10]
C     LAST UPDATED  : JULY 1984
C
C     COMPILE       : >F77 ILAJENK.OBJ,ILAJENK.LST/-SP=ILAJENK.FTN/TR:NONE
```

```
C
C     PURPOSE       : OPTIMAL DATA CLASSIFICATION
C                     - BASED ON A SOLUTION PROPOSED BY
C                     WALTER D. FISHER 'ON GROUPING FOR
C                     MAXIMUM HOMOGENEITY', JOURNAL OF THE
C                     AMERICAN STATISTICAL ASSOCIATION,
C                     DECEMBER 1958, PP.789-798.
C                     - SUBROUTINE FAVAR ADAPTED FROM
C                     SUBROUTINE FISH CLUSTERING ALGORITHMS
C                     BY JOHN A. HARTIGAN, JOHN WILEY AND
C                     SONS, 1975, P.141.
C
C     REMARKS       : THE ORIGINAL VERSION HAS BEEN MODIFIED
C                     TO MEET OUR REQUIREMENTS.
C                     1) THE SUBROUTINE FAABS HAS BEEN
C                        ELIMINATED.
C                     2) THE PROGRAM HAS BEEN CONVERTED INTO
C                        A SUBROUTINE.
C                     3) LINES HAVE BEEN ADDED TO MAKE AN
C                        INTERECTIVE VERSION AND ADAPT IT
C                        TO THE PDP11/70.
C
C     DFPI SUBROUTINES USED :
C     NONE
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     FAVAR
C
C     .................................................
C
      SUBROUTINE JENK(N, K, Z, ZONE, ZL, ZS)
C
      DIMENSION Z(50), W(50), LC(50,10), OP(50,10)
      DIMENSION ERR(50,50), EMIN(10,50), ZMED(50,50)
      DIMENSION ZL(10), ZS(10)
      INTEGER*2 ZONE(50)
      INTEGER*4 REFILE(31)
C
      DATA W /50*1.0/
C
C     KEY-IN THE NUMBER OF CLASSES
C     ---------------------------
   10 CONTINUE
      WRITE (5,20)
   20 FORMAT (/, '$NUMBER OF CLASSES (1-10) = ')
      READ (5,30) K
   30 FORMAT (I3)
C
C WRITE DATA AS ENTERED OR AS ARRAYED
C
C     KEY-IN REPORT FILE NAME
```

```
C      ----------------------
       WRITE (5,40)
    40 FORMAT (/, '$JENKS'' REPORT FILE NAME= ')
       READ (5,50) NC, REFILE
    50 FORMAT (Q, 31A1)
       IF (NC .EQ. 0) GO TO 290
       IF (NC .GT. 30) NC = 30
       REFILE(NC + 1) = 0
       OPEN (UNIT=3,NAME=REFILE,TYPE='NEW',ERR=60)
       GO TO 80
C
    60 WRITE (5,70)
    70 FORMAT (/, ' PROBLEM OPENING REPORT FILE')
       GO TO 290
C
    80 WRITE (3,90)
    90 FORMAT (1H1, 'CONTROL PARAMETERS', //)
C
       WRITE (3,100) N, K, FMTA, IA, IB, IW
   100 FORMAT (1X, I5, 1X, '= NUMBER OF ENUMERATION UNITS', //, 1X, I5,
      1        1X, '= MAXIMUM NUMBER OF CLASSES', //, 1X, 80A1, //, 7X,
      2        '= FORMAT CARD', //, 1X, I5, 1X, '= IA OPTION', //, 1X, I5,
      3        1X, '= IB OPTION', //, 1X, I5, 1X, '= IW OPTION')
C
C
       WRITE (3,110) N, K
   110 FORMAT (1H1, 'THERE ARE', I5, 3X, 'OBSERVATIONS', //, 3X,
      1        'TO BE PARTITIONED INTO  1  THROUGH', I3, 2X,
      2        'CLASS GROUPINGS', ///)
       WRITE (3,120)
   120 FORMAT (1H0, 15X, 'DATA AS INPUT', //, 9X, 'N', 4X, 'NAME', 8X,
      1        'Z VALUE', 4X, 'WEIGHT', //)
   130 DO 150 I = 1, N
         WRITE (3,140) I, ZONE(I), Z(I), W(I)
   140   FORMAT (5X, I5, 3X, 'N', I3, F13.3, F10.3)
   150 CONTINUE
       IF (IA .EQ. 1) GO TO 190
C
C AREALLY STANDARDIZE DATA
C
       DO 160 I = 1, N
         Z(I) = Z(I) / W(I)
         W(I) = 1.0
   160 CONTINUE
C
C ARRAY DATA
C
       NNN = N - 1
       DO 180 I = 1, NNN
         IX = I + 1
         DO 170 J = IX, N
```

```
      IF (Z(I) .LE. Z(J)) GO TO 170
      TZ = Z(I)
      NT = ZONE(I)
      WT = W(I)
      Z(I) = Z(J)
      ZONE(I) = ZONE(J)
      W(I) = W(J)
      Z(J) = TZ
      ZONE(J) = NT
      W(J) = WT
170   CONTINUE
180 CONTINUE
190 CONTINUE
C
C WRITE ARRAYED AND STANDARDIZED DATA
C
    WRITE (3,200)
200 FORMAT (1H1, 11X, 'ARRAYED STANDARDIZED DATA', //, 9X, 'N', 4X,
   1       'NAME', 8X, 'Z VALUE', 4X, 'WEIGHT', //)
C
    DO 210 I = 1, N
      WRITE (3,140) I, ZONE(I), Z(I), W(I)
210 CONTINUE
C
    CALL FAVAR(N, K, Z, W, EC, OP, IW, ZL, ZS)
C
    WRITE (5,220) K
220 FORMAT (//, ' A', I3, ' CLASS MAP')
    DO 240 I = 1, K
      WRITE (5,230) ZS(I), ZL(I)
230   FORMAT (5X, 2F10.3)
240 CONTINUE
C
    WRITE (5,250)
250 FORMAT (//, '$MODIFY THE NUMBER OF CLASSES', ' OR CONTINUE?(M/C) '
   1       )
    READ (5,260) OPTION
260 FORMAT (A1)
C
    IF (OPTION .EQ. 'M') THEN
      CLOSE (UNIT=3,DISPOSE='DELETE',ERR=270)
      GO TO 10
    ELSE
      CLOSE (UNIT=3,DISPOSE='SAVE',ERR=280)
    END IF
C
    RETURN
270 STOP 'PROBLEM DELETING REPORT FILE'
280 STOP 'PROBLEM CLOSING REPORT FILE'
290 STOP 'PROBLEM OPENING REPORT FILE, PROGRAM TERMINATED'
    END
```

```
C
C     .......................................................
C
C     SUBROUTINE FAVAR
C
C     LANGUAGE       : FORTRAN
C     SYSTEM         : PDP11/70 RSX-11M-PLUS
C     AUTHOR         : GEORGE F. JENKS (1977)
C     UIC            : [266,10]
C     LAST UPDATED   : ORIGINAL VERSION (1977)
C
C     COMPILE        : >F77 ILAJENK.OBJ,ILAJENK.LST/-SP=ILAJENK.FTN/TR:NONE
C
C     PURPOSE        : CREATES OPTIMAL DATA CLASSES AS
C                      MEASURED BY VARIANCE.
C
C     DFPI SUBROUTINES USED :
C     NONE
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     NONE
C
C     .......................................................
C
      SUBROUTINE FAVAR(N, K, Z, W, LC, OP, IW, ZL, ZS)
C
C     LOCAL VARIABLES
C     ---------------
C     LC     AN N BY K MATRIX OF OPTIMAL LOWER CLASS LIMITS
C     OP     AN N BY K MATRIX OF OPTIMAL VARIANCE
C            COMBINATIONS FOR ALL CLASSES
C
      DIMENSION OP(50,10), LC(50,10), Z(50), W(50)
      REAL ZL(10), ZS(10)
C
      DO 20 I = 1, K
        LC(1,I) = 1
        OP(1,I) = 0.0
        DO 10 J = 2, N
          OP(J,I) = 99999999.
   10   CONTINUE
   20 CONTINUE
C
      IF (IW .LT. 2) GO TO 50
      WRITE (11,30)
   30 FORMAT (1H1, 'CALCULATE ALL POSSIBLE N BY K VARIANCES')
      WRITE (11,40)
   40 FORMAT (1HO, 3X, 'I', 2X, 'II', 2X, 'III', 7X, 'ZSQ', 10X, 'SZ',
     1        13X, 'WT', 8X, 'V', 6X, 'IV', 4X, 'J', 2X, 'LC(I,J)', 7X,
     2        'OP(I,J)', 7X, 'OP(IV,J-1)')
   50 CONTINUE
```

```
C
      DO 150 I = 2, N
        SZ = 0.0
        ZSQ = 0.0
        WT = 0.0
        IF (IW .EQ. 0) GO TO 70
        IF (IW .EQ. 1) GO TO 70
        WRITE (11,60) I
   60   FORMAT (I5)
   70   CONTINUE
C
        DO 130 II = 1, I
          III = I - II + 1
          ZSQ = ZSQ + (Z(III)**2) * W(III)
          SZ = SZ + (Z(III)*W(III))
          WT = WT + W(III)
          CN = II
          V = ZSQ - ((SZ**2)/WT)
          IV = III - 1
          IF (IV .EQ. 0) GO TO 130
          IF (IW .LT. 2) GO TO 90
          WRITE (11,80) II, III, ZSQ, SZ, WT, V, IV
   80     FORMAT (5X, 2I4, F15.3, F12.3, F12.3, F12.3, I4)
   90     CONTINUE
          DO 120 J = 1, K
            IF (J .EQ. 1) GO TO 100
            IF (OP(I,J) .LT. V + OP(IV,J - 1)) GO TO 100
            LC(I,J) = III
            OP(I,J) = V + OP(IV,J - 1)
  100       IF (J - 1 .EQ. 0) OP(IV,J - 1) = 0.0
            IF (IW .LT. 2) GO TO 120
            WRITE (11,110) J, LC(I,J), OP(I,J), OP(IV,J - 1)
  110       FORMAT (68X, 2I5, F17.3, F15.3)
  120     CONTINUE
  130   CONTINUE
        LC(I,1) = 1
        OP(I,1) = V
        IF (IW .LT. 2) GO TO 150
        WRITE (11,140) OP(I,1)
  140   FORMAT (90X, 'OP(I,1)=', F15,3)
  150 CONTINUE
      IF (IW .EQ. 0) GO TO 220
      WRITE (11,160)
  160 FORMAT (1H1, 8X, 'MATRIX OF OPTIMAL CLASS VARIANCES'//, 9X,
     1       'N ROWS     COLUMNS', //)
      DO 180 I = 1, N
        WRITE (11,170) (OP(I,J),J=1,K)
  170   FORMAT (5X, 7F13.3)
  180 CONTINUE
      WRITE (11,190)
  190 FORMAT (1H1, 9X, 'MATRIX OF LOWER LIMITS FOR OPTIMAL CLASSES', //,
```

```
     1         10X, 'N ROWS BY K COLUMNS', //)
      DO 210 I = 1, N
        WRITE (11,200) (LC(I,J),J=1,K)
  200   FORMAT (5X, 7I11)
  210 CONTINUE
  220 CONTINUE
C
C WRITE HEADERS FOR RESULTS
C
      WRITE (3,230)
  230 FORMAT (1H1, 'OPTIMAL CLASSES AS MEASURED BY VARIANCE')
      DO 290 J = 1, K
        JJ = K - J + 1
        WRITE (3,240) JJ, OP(N,JJ)
  240   FORMAT (///, 7X, 'A', I3, 2X, 'CLASS MAP WITH A VARIANCE OF',
     1          F15.3, //, 14X, 'CLASS      N      LARGEST   SMALLEST      M
     2EAN          VARIANCE')
        IL = N + 1
C
C CALCULATE CLASS MEANS AND CLASS VARIANCE
C
        DO 280 L = 1, JJ
          LL = JJ - L + 1
          SZ = 0.0
          WT = 0.0
          ZSQ = 0.0
          IU = IL - 1
          IL = LC(IU,LL)
          DO 250 II = IL, IU
            SZ = SZ + Z(II) * W(II)
            WT = WT + W(II)
            ZSQ = ZSQ + (Z(II)**2) * W(II)
  250     CONTINUE
          CN = IU - IL + 1
          NC = IFIX(CN)
          ZBAR = SZ / WT
          VV = ZSQ - ((ZBAR**2)*WT)
C
C WRITE CLASS CHARACTERISTICS
C
          IF (JJ .NE. K) GO TO 260
          ZL(LL) = Z(IU)
          ZS(LL) = Z(IL)
  260     WRITE (3,270) LL, NC, Z(IU), Z(IL), ZBAR, VV
  270     FORMAT (14X, I3, 3X, I5, 5X, F7.2, 3X, F7.2, 3X, F7.2, 4X,
     1            F13.2)
  280   CONTINUE
  290 CONTINUE
      RETURN
      END
C
```

```
C     **************************************************
C
C
C     O V E R L A Y   S E G M E N T   T ' R E E
C
C
C     **************************************************
C
C     VARIABLES        DESCRIPTION
C     ---------        -----------
C     AA               ACTIVE ANGLE
C     ANTIL            ANTILOGARITHM
C     AS               ACTIVE SCALE
C     ATLK             ATTRIBUTE LINKAGE
C     AXES             ARRAY OF CIRCLE'S AXIS
C     COOR             COORDINATES(R*8)
C     DBRAD            RADIUS DOUBLE-PRECISION
C     ELSPE            ELLIPSE SPECIFICATIONS
C     FILE             DESIGN FILE NAME
C     GG               GRAPHIC GROUP
C     IAD              ARRAY CONTAINING ADDRESSES FOR LINES
C     INFILE           DATA FILE NAME
C     IRC              ERROR CODE
C     K1               NUMBER OF CLASSES FOR CIRCLES
C     K2               NUMBER OF CLASSES FOR LINES
C     LILVS            LEVELS USED FOR THE LINES
C     LINSTR           COORDINATES(I*4)
C     LNSPE            LINE SPECIFICATIONS
C     LOGA             LOGARITHM
C     LV               LEVEL
C     N                NUMBER OF ZONES
C     NC               NUMBER OF CHARACTERS
C     NORAD            ARRAY OF RADII
C     NPER             NUMBER OF PERCENTAGES
C     ORIG             ARRAY OF COORDINATES(I*4)
C     PEMID            MIDPOINT OF ONE CLASS INTERVAL
C     PMID             ARRAY OF MIDPOINTS
C     PER              ARRAY OF PERCENTAGES
C     PERL             ARRAY OF LARGEST VALUES
C     PERS             ARRAY OF SMALLEST VALUES
C     PTS              ARRAY OF COORDINATES(R*8)
C     PUFA             POSITIONAL UNIT FACTOR
C     PUSU             NUMBER OF PU'S BY SU'S
C     RAD              RADIUS
C     REGNA            REGION NAME
C     SUMU             NUMBER OR SU'S BY MU'S
C     UNVA             UNIT VALUE
C     WIT              LINE WIDTH
C     ZL               ARRAY OF LARGEST VALUES
C     ZMID             MIDPOINT OF ONE CLASS INTERVAL
C     ZONE             ARRAY OF ZONE NUMBERS
```

```
C      ZS              ARRAY OF SMALLEST VALUES
C      ZVAL            ARRAY OF WEIGHTS
C
C      ...................................................
C
C      SUBROUTINE P1SET
C
C      LANGUAGE      : FORTRAN
C      SYSTEM        : PDP11/70 RSX-11M-PLUS
C      AUTHOR        : LISE ALLARD
C      UIC           : [266,10]
C      LAST UPDATED  : JULY 1984
C
C      COMPILE       : >77 ILAP1SET.OBJ,ILAP1SET.LST/-SP=ILAP1SET.FTN/TR:NONE
C
C      PURPOSE       : INITIALIZE DFPI, READ THE DATA AND
C                      CALL THE SUBROUTINES P1CIR,P1LIN
C                      AND P1END.
C
C      REMARK        : IT PLAYS THE ROLE OF A MAIN PROGRAM
C                      IN THE THIRD SEGMENT OF THE OVERLAY
C                      STRUCTURE.
C
C      DFPI SUBROUTINES USED :
C      ASC2RD          CONVERTS A STRING OF ASCII CHARACTERS
C                      INTO AN EQUIVALENT STRING OF PACKED
C                      RAD50 CHARACTERS.
C      CONVER          CONVERTS FROM FLOATING POINT DOUBLE-
C                      PRECISION NUMBER WITH RANGE OF O TO
C                      +4294967296 TO I*4 FORTRAN FORMAT
C                      WITH RANGE OF -2147483648 TO +2147483647.
C      ERROR9          RETURNS ERROR MESSAGES TO LUN6 FOR DFPI
C                      INTERFACE ERROR.
C      INDFPI          INITIALIZE DFPI AND SET UP DESIGN FILE.
C      LKTCSI          FILE NAME CONVERSION
C
C      SUBROUTINES AND FUNCTIONS REQUIRED :
C      P1CIR
C      P1LIN
C      P1END
C
C      ...................................................
C
       SUBROUTINE P1SET
C
       COMMON /BLK1/ ZL(10), ZS(10), PERL(10), PERS(10), K1, K2
       COMMON /BLK2/ IAD(200), PTS(200), ORIG(200), PER(100), ZVAL(100),
      1        NORAD(100), PUFA, N, NPER
       COMMON /BLK3/ ANSER1
       COMMON /BLK4/ IOP(4)
       COMMON /IREQ/ IREQ
```

```
      COMMON /INFLAG/ INFLAG
      COMMON /WAITFL/ WAITFL
      COMMON /APREGN/ APREGN
      COMMON /DFPI/ DFPI
C
      INTEGER*2 INFILE(31), FILE(15), DGN50(7)
      INTEGER*2 REGNA(3), N, NPER, K1, K2, IOP
      INTEGER*4 ORIG
C
      REAL PER, ZVAL, NORAD, ZL, ZS, PERL, PERS
      REAL*8 PIS, PUFA, PUSU, SUMU
      CHARACTER*1 ANSER1, ANSER2
      CHARACTER*20 FMT
C
      DATA REGNA /'DF', 'PI', 'RG'/
C
C     PROGRAM IDENTIFICATION
C     ----------------------
      WRITE (5,10)
   10 FORMAT (/////, 10X, ' * PART 1-B : CREATE A MAP WITH ',
     1        'OPTIMAL CLASS INTERVALS *')
C
C     KEY-IN DESIGN FILE NAME
C     -----------------------
      WRITE (5,20)
   20 FORMAT (///, '$DESIGN FILE NAME=  ')
      READ (5,30) NC, FILE
   30 FORMAT (Q, 15A2)
      CALL LKTCSI(FILE, DGN50, NC, O, IRC)
      IF (IRC .NE. O) STOP 'LKTCSI CONVERSION ERROR'
C
C     REGION NAME CONVERSION
C     ----------------------
      CALL ASC2RD(REGNA, REGN50, 6, IRC)
      IF (IRC .NE. O) STOP 'ASC2RD CONVERSION ERROR'
C
C     DETERMINE POSITIONAL UNIT FACTOR
C     --------------------------------
   40 CONTINUE
      WRITE (5,50)
   50 FORMAT (///, '$NUMBER OF PU''S PER SU= ')
      READ (5,*,ERR=40) PUSU
      WRITE (5,60)
   60 FORMAT (/, '$NUMBER OF S     PER MU= ')
      READ (5,*,ERR=40) SUMU
C
      PUFA = PUSU * SUMU
C
C     INITIALIZE FOR DFPI
C     -------------------
      CALL INDFPI(REGN50, DGN50, O, O, O, O, IRC, 'EX')
```

```
      IF (IRC .NE. 0) THEN
        WRITE (5,70) IRC
   70   FORMAT (' INDFPI ERROR IRC= ', I3)
        CALL ERROR9(IRC)
        STOP
      END IF
   80 WRITE (5,90)
   90 FORMAT (/, ' *** INITIALIZATION  COMPLETED ***')
C
C     KEY-IN DATA FILE NAME
C     ---------------------
      WRITE (5,100)
  100 FORMAT (//, '$DATA FILE NAME= ')
      READ (5,110) NC, INFILE
  110 FORMAT (Q, 31A1)
      IF (NC .EQ. 0) GO TO 140
      IF (NC .GT. 30) NC = 30
      INFILE(NC + 1) = 0
      OPEN (UNIT=2,NAME=INFILE,TYPE='OLD',READONLY,ERR=82)
      GO TO 150
C
  120 WRITE (5,130)
  130 FORMAT (/, ' PROBLEM OPENING DATA FILE')
      GO TO 140
C
C
  140 CONTINUE
      STOP
C
C     READ THE NUMBER OF ZONES
C     ------------------------
  150 CONTINUE
      READ (2,*,ERR=260) N
C
C     READ COORDINATES
C     ----------------
      COUNT = 1
      DO 170 I = 1, N
        READ (2,160) PTS(COUNT), PTS(COUNT + 1)
  160   FORMAT (4X, 2F12.5)
        PTS(COUNT) = PTS(COUNT) * PUFA
        PTS(COUNT + 1) = PTS(COUNT + 1) * PUFA
        COUNT = COUNT + 2
  170 CONTINUE
C
      CALL CONVER(PTS, ORIG, COUNT - 1, IRC)
C
C     PLACE PROPORTIONAL CIRCLES
C     ------------  ----------
      WRITE (5,180)
  180 FORMAT (///, '$PLACE PROPORTIONAL CIRCLES?(Y/N) ')
```

```
      READ (5,190) ANSER1
  190 FORMAT (A1)
C
C
C     READ VALUES
C     -----------
      DO 210 I = 1, N
        READ (2,200) ZVAL(I)
  200   FORMAT (F10.3)
  210 CONTINUE
      IF (ANSER1 .EQ. 'N') GO TO 220
C
      CALL P1CIR
C
C     PLACE PROPORTIONAL FLOW LINES
C     -----------------------------
  220 CONTINUE
      WRITE (5,230)
  230 FORMAT (///, '$PLACE PROPORTIONAL FLOW LINES?(Y/N) ')
      READ (5,190) ANSER2
      IF (ANSER2 .EQ. 'N') GO TO 280
C
C     READ THE NUMBER OF PERCENTAGES
C     ------------------------------
      READ (2,*,ERR=260) NPER
C
C     READ PERCENTAGE AND ZONES
C     -------------------------
      J = 1
      DO 250 I = 1, NPER
        READ (2,240) PER(I), IAD(J), IAD(J + 1)
  240   FORMAT (F6.2, 2I5)
        J = J + 2
  250 CONTINUE
C
      CALL P1LIN
      GO TO 280
C
  260 WRITE (5,270)
  270 FORMAT (/, ' * PROBLEM READING DATA FILE *')
      STOP
C
  280 CONTINUE
      CALL P1END
C
      CLOSE (UNIT=2,DISPOSE='SAVE')
      RETURN
      END
C
C     .....................................................
C
```

```
C       SUBROUTINE P1CIR
C
C       LANGUAGE      : FORTRAN
C       SYSTEM        : PDP11/70 RSX-11M-PLUS
C       AUTHOR        : LISE ALLARD
C       UIC           : [266,10]
C       LAST UPDATED  : JULY 1984
C
C       COMPILE       : >F77 ILAP1CIR.OBJ,ILAP1CIR.LST/-SP=ILAP1CIR.FTN/TR:NONE
C
C       PURPOSE       : ASSOCIATE A RADIUS TO EACH VALUE
C                       AND PLACE THE CIRCLES IN THE DESIGN FILE.
C
C       DFPI SUBROUTINES USED :
C       ELDFPI          PLACE ELLIPSE, 2-D ONLY.
C       ERROR9          RETURNS ERROR MESSAGES TO LUN6 FOR
C                       DFPI INTERFACE ERROR.
C
C       SUBROUTINES AND FUNCTIONS REQUIRED :
C       RADIUS
C
C       ..............................................
C
        SUBROUTINE P1CIR
C
        COMMON /BLK1/ ZL(10), ZS(10), PERL(10), PERS(10), K1, K2
        COMMON /BLK2/ IAD(200), PTS(200), ORIG(200), PER(100), ZVAL(100),
       1        NORAD(100), PUFA, N, NPER
        COMMON /BLK3/ ANSER1
        COMMON /IREQ/ IREQ
        COMMON /INFLAG/ INFLAG
        COMMON /WAITFL/ WAITFL
        COMMON /APREGN/ APREGN
        COMMON /DFPI/ DFPI
C
        INTEGER*2 IAD, ELSPE(5), ATLK(4), GG(2)
        INTEGER*2 K, K1, K2, N, NPER, LV, OPTION
        INTEGER*4 ORIG
C
        REAL PER, ZVAL, NORAD, RAD(10)
        REAL ZL, ZS, PERL, PERS
        REAL*8 PTS, AXES(2), AS(2), WPTS(2)
        REAL*8 AA, ANTIL, DBRAD, LOGA, PUFA, UNVA
        CHARACTER*1 ANSER1
C
        DATA AS /0.0, 0.0/, ATLK /4*0/, GG /2*0/
        DATA ELSPE /5*0/, AA /0.0/
C
        K = K1
C
C       DEFINE THE ACTIVE LEVEL
```

```
C      -----------------------
       WRITE (5,10)
    10 FORMAT (/, '$ACTIVE LEVEL (1-63) = ')
       READ (5,20) LV
    20 FORMAT (I2)
C
C      CALCULATE RADIUS
C      ----------------
    30 CONTINUE
       WRITE (5,40)
    40 FORMAT (/, '$RADIUS FOR LARGEST CIRCLE = ')
       READ (5,*,ERR=30) RAD(K)
       ZMID = ((ZL(K) - ZS(K))/2) + ZS(K)
       LOGA = ALOG10(ZMID)
       LOGA = LOGA * 0.5718
       ANTIL = 10 ** LOGA
       UNVA = RAD(K) / ANTIL
       KK = K - 1
       DO 50 I = 1, KK
         ZMID = ((ZL(I) - ZS(I))/2) + ZS(I)
         RAD(I) = RADIUS(ZMID,UNVA)
    50 CONTINUE
C
C      PRINT RADIUS
C      ------------
       WRITE (5,60)
    60 FORMAT (//, ' RADIUS ASSOCIATED TO EACH CLASS')
       DO 80 I = 1, K
         WRITE (5,70) RAD(I)
    70   FORMAT (10X, F10.6)
    80 CONTINUE
C
       WRITE (5,90)
    90 FORMAT (//, '$MODIFY THE RADIUS OR CONTINUE?', ' (M/C) ')
       READ (5,100) OPTION
   100 FORMAT (A1)
       IF (OPTION .EQ. 'M') GO TO 30
C
C      PLACE CIRCLES
C      -------------
   110 CONTINUE
       J = 1
       DO 150 I = 1, N
         DO 120 L = K, 1, -1
           IF (ZVAL(I) .GE. ZS(L)) THEN
             DBRAD = DBLE(RAD(L))
             AXES(1) = DBRAD * PUFA
             AXES(2) = AXES(1)
             GO TO 130
           ELSE
             GO TO 120
```

```
         END IF
  120  CONTINUE
  130  NORAD(I) = RAD(L)
       WPTS(1) = ORIG(J)
       WPTS(2) = ORIG(J + 1)
       CALL ELDFPI(GG, LV, ELSPE, AXES, WPTS, AA, IRC, ATLK)
       IF (IRC .NE. 0) THEN
         WRITE (5,140) IRC
  140    FORMAT (' ERROR IN ELDFPI IRC= ', I3)
         CALL ERROR9(IRC)
         STOP
       END IF
       J = J + 2
  150 CONTINUE
C
      WRITE (5,160)
  160 FORMAT (/, ' ** PROPORTIONAL CIRCLES COMPLETED **')
C
      RETURN
      END
C
C     ..................................................
C
C     FUNCTION RADIUS
C
C     LANGUAGE     : FORTRAN
C     SYSTEM       : PDP11/70 RSX-11M-PLUS
C     AUTHOR       : LISE ALLARD
C     UIC          : [266,10]
C     LAST UPDATED : JULY 1984
C
C     PURPOSE      : CALCULATE A PROPORTIONAL RADIUS
C                    VALUE ACCORDING TO THE PSYCHOLOGICAL
C                    METHOD.
C
C     REFERENCE    : ROBINSON A. ET AL.(1978), ELEMENTS
C                    OF CARTOGRAPHY 4TH EDITION.
C
C     ..................................................
C
C
      FUNCTION RADIUS(ZMID, UNVA)
C
C     ARGUMENTS DEFINITION
C     --------------------
C     ZMID           MIDPOINT OF ONE CLASS INTERVAL
C     UNVA           UNIT VALUE
C
      REAL ZMID
      REAL*8 LOGA, ANTIL, UNVA
C
```

```
      LOGA = ALOG10(ZMID)
      LOGA = LOGA * 0.5718
      ANTIL = 10 ** LOGA
      RADIUS = ANTIL * UNVA
C
      RETURN
      END
C
C     ..............................................
C
C     SUBROUTINE P1LIN
C
C     LANGUAGE       : FORTRAN
C     SYSTEM         : PDP11/70 RSX-11M-PLUS
C     AUTHOR         : LISE ALLARD
C     UIC            : [266,10]
C     LAST UPDATED   : JULY 1984
C
C     COMPILE        : >F77 ILAP1LIN.OBJ,ILAP1LIN.LST/-SP=ILAP1LIN.FTN/TR:NONE
C
C     PURPOSE        : ASSOCIATE A LINE WIDTH TO EACH PERCENTAGE
C                      AND PLACE THE LINES IN THE DESIGN FILE.
C
C     DFPI SUBROUTINES USED :
C     CONVER           CONVERTS FROM FLOATING POINT DOUBLE-
C                      PRECISION NUMBER WITH RANGE OF 0 TO
C                      +4294967296 TO I*4 FORTRAN FORMAT
C                      WITH RANGE OF -2147483648 TO +2147483647.
C     ERROR9           RETURNS ERROR MESSAGES TO LUN6 FOR DFPI
C                      INTERFACE ERROR.
C     LNDFPI           PLACE LINE BY DEFINING THE ENDPOINTS.
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     CTLINE
C     IWIDTH
C
C     ..............................................
C
      SUBROUTINE P1LIN
C
      COMMON /BLK1/ ZL(10), ZS(10), PERL(10), PERS(10), K1, K2
      COMMON /BLK2/ IAD(200), PTS(200), ORIG(200), PER(100), ZVAL(100),
     1       NORAD(100), PUFA, N, NPER
      COMMON /BLK3/ ANSER1
      COMMON /IREQ/ IREQ
      COMMON /INFLAG/ INFLAG
      COMMON /WAITFL/ WAITFL
      COMMON /APREGN/ APREGN
      COMMON /DFPI/ DFPI
C
      INTEGER*2 IAD, WIT(10), LNSPE(5), ATLK(4)
```

```
      INTEGER*2 GG(2), K, K1, K2, N, NPER, NLINE
      INTEGER*4 ORIG, LINSTR(4)
      CHARACTER*1 ANSER1, OPTION
C
      REAL PER, ZVAL, NORAD, ZL, ZS, PERL, PERS, PMID(10)
      REAL*8 PTS, COOR(4), RD(2), RUFA
C
      DATA NLINE /4/, LNSPE /5*0/
      DATA GG /2*0/, ATLK /4*0/
C
      K = K2
C
C     DEFINE THE ACTIVE LEVEL
C     -----------------------
   10 CONTINUE
      WRITE (5,20)
   20 FORMAT (/, '$ACTIVE LEVEL (1-63) = ')
      READ (5,30) LV
   30 FORMAT (I2)
      LILVS = LV + K - 1
      WRITE (5,40) K, LV, LILVS
   40 FORMAT (//, ' LEVELS TO BE USED FOR A', I3, ' CLASS MAP = ', I3,
     1        ' TO ', I3, //, '$MODIFY OR CONTINUE?(M/C) ')
      READ (5,120) OPTION
      IF (OPTION .EQ. 'M') GO TO 10
C
C     CALCULATE LINE THICKNESS
C     ------------------------
   50 CONTINUE
      WRITE (5,60)
   60 FORMAT (/, '$THICKNESS FOR BIGGEST LINE (0-31) = ')
      READ (5,*,ERR=50) WIT(K)
      PMID(K) = ((PERL(K) - PERS(K))/2) + PERS(K)
      UNVA = WIT(K) / PMID(K)
      KK = K - 1
      DO 70 I = 1, KK
        PEMID = ((PERL(I) - PERS(I))/2) + PERS(I)
        WIT(I) = IWIDTH(PEMID,UNVA)
   70 CONTINUE
C
C     PRINT LINE THICKNESS
C     --------------------
      WRITE (5,80)
   80 FORMAT (//, ' LINE THICKNESS ASSOCIATED TO EACH CLASS')
      DO 100 I = 1, K
        WRITE (5,90) WIT(I)
   90   FORMAT (10X, I3)
  100 CONTINUE
C
      WRITE (5,110)
  110 FORMAT (//, '$MODIFY THE LINE THICKNESS OR ', 'CONTINUE?(M/C) ')
```

```
      READ (5,120) OPTION
  120 FORMAT (A1)
      IF (OPTION .EQ. 'M') GO TO 50
C
C     PLACE LINES
C     -----------
  130 CONTINUE
      DO 180 I = 1, NPER
        DO 140 L = K, 1, -1
          IF (PER(I) .GE. PERS(L)) THEN
            LNSPE(4) = WIT(L)
            GO TO 150
          ELSE
            GO TO 140
          END IF
  140   CONTINUE
  150   CONTINUE
        IA1 = IAD((2*I) - 1)
        IA2 = IAD(2*I)
        COOR(1) = PTS((2*IA1) - 1) / PUFA
        COOR(2) = PTS(2*IA1) / PUFA
        COOR(3) = PTS((2*IA2) - 1) / PUFA
        COOR(4) = PTS(2*IA2) / PUFA
        IF (ANSER1 .EQ. 'N') GO TO 160
        IF (IA1 .EQ. IA2) THEN
          COOR(3) = COOR(1) + (0.5*NORAD(IA1))
          COOR(1) = COOR(1) - (0.5*NORAD(IA1))
          COOR(4) = COOR(2)
          GO TO 160
        END IF
        RD(1) = NORAD(IA1)
        RD(2) = NORAD(IA2)
        CALL CTLINE(COOR, RD)
  160   CONTINUE
        COOR(1) = COOR(1) * PUFA
        COOR(2) = COOR(2) * PUFA
        COOR(3) = COOR(3) * PUFA
        COOR(4) = COOR(4) * PUFA
        CALL CONVER(COOR, LINSTR, NLINE, IRC)
        LV = LV + L - 1
        CALL LNDFPI(GG, LV, LNSPE, LINSTR, IRC, ATLK)
        LV = LV - L + 1
        IF (IRC .NE. 0) THEN
          WRITE (5,170) IRC
  170     FORMAT (' ERROR IN LNDFPI IRC=', I3)
          CALL ERROR9(IRC)
          STOP
        END IF
  180 CONTINUE
C
      WRITE (5,190)
```

```
  190 FORMAT (/, ' ** PROPORTIONAL LINES COMPLETED **')
C
      RETURN
      END
C
C     .............................................................
C
C     FUNCTION IWIDTH
C
C     PURPOSE        : CALCULATES A PROPORTIONAL LINE
C                      WIDTH.
C
C     .............................................................
C
      FUNCTION IWIDTH(PEMID, UNVA)
C
C     ARGUMENTS DEFINITION
C     --------------------
C     PEMID            MIDPOINT OF ONE CLASS INTERVAL
C     UNVA             UNIT VALUE
C
      REAL PEMID, UNVA
C
      IWIDTH = ININT(PEMID*UNVA)
C
      RETURN
      END
C
C     .............................................................
C
C     SUBROUTINE CTLINE
C
C     LANGUAGE       : FORTRAN
C     SYSTEM         : PDP11/70 RSX-11M-PLUS
C     AUTHOR         : LISE ALLARD
C     UIC            : [266,10]
C     LAST UPDATED   : JULY 1984
C
C     COMPILE        : >F77 ILAP1LIN.OBJ,ILAP1LIN.LST/-SP=ILAP1LIN.FTN/TR:NONE
C
C     PURPOSE        : CALCULATES TWO PAIRS OF COORDINATES
C                      TO PLACE A LINE BETWEEN TWO CIRCLES.
C
C     DFPI SUBROUTINES USED :
C     NONE
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     NONE
C
C     .............................................................
C
```

```
      SUBROUTINE CTLINE(COOR, RD)
C
C     ARGUMENTS DEFINITION
C     --------------------
C     COOR          ARRAY CONTAINING TWO PAIRS OF COORDINATES
C     RD            ARRAY OF RADII OF TWO CIRCLES
C
C     LOCAL VARIABLES
C     ---------------
C     M             SLOPE
C     XA            X COORDINATE (ORIGIN OF CIRCLE)
C     YA            Y COORDINATE (ORIGIN OF CIRCLE)
C     XC            X COORDINATE (EDGE OF CIRCLE)
C     YC            Y COORDINATE (EDGE OF CIRCLE)
C     RR            RADIUS
C
      REAL*8 COOR(4), RD(2), M, XA, YA, XC, YC, ROOT1, ROOT2, R(2)
      REAL*8 DY, DX
C
      DY = COOR(4) - COOR(2)
      DX = COOR(3) - COOR(1)
      IF (DY .EQ. 0) DY = 1E-10
      IF (DX .EQ. 0) DX = 1E-10
C
      M = DY / DX
C
      DO 10 I = 1, 2
        RR = RD(I)
        II = I
        IF (I .EQ. 2) II = 3
        XA = COOR(II)
        YA = COOR(II + 1)
        ROOT1 = SQRT(RR**2/(1 + (M**2)))
C
        IF (COOR(3) .GT. COOR(1)) THEN
          XC = XA - ((-1)**I) * ROOT1
        ELSE
          XC = XA + ((-1)**I) * ROOT1
        END IF
C
        ROOT2 = SQRT(RR**2 - ((XC - XA)**2))
C
        IF (COOR(4) .GT. COOR(2)) THEN
          YC = YA - ((-1)**I) * ROOT2
        ELSE
          YC = YA + ((-1)**I) * ROOT2
        END IF
C
        COOR(II) = XC
        COOR(II + 1) = YC
   10 CONTINUE
```

```
C
      RETURN
      END
C
C     ..........................................................
C
C     SUBROUTINE P1END
C
C     LANGUAGE      : FORTRAN
C     SYSTEM        : PDP11/70 RSX-11M-PLUS
C     AUTHOR        : LISE ALLARD
C     UIC           : [266,10]
C     LAST UPDATED  : JULY 1984
C
C     COMPILE       : >F77 ILAP1END.OBJ,ILAP1END.LST/-SP=ILAP1END.FTN
C
C     PURPOSE       : TERMINATES DFPI
C
C     DFPI SUBROUTINES USED :
C     DEDFPI          DETACH FROM DFPI
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     NONE
C
C     ..........................................................
C
      SUBROUTINE P1END
C
      COMMON /IREQ/ IREQ
      COMMON /INFLAG/ INFLAG
      COMMON /WAITFL/ WAITFL
      COMMON /APREGN/ APREGN
      COMMON /DFPI/ DFPI
C
C     TERMINATE DFPI
C     --------------
      IARG = 1
      IF (IRC .NE. 0) IARG = 0
      CALL DEDFPI(IARG)
C
      RETURN
      END
C
C     ********************************************************
C
C     O V E R L A Y   S E G M E N T   F O U R
C
C     ********************************************************
C
```

```
C     VARIABLES         DESCRIPTION
C     ---------         -----------
C     AA                ACTIVE ANGLE
C     ANTIL             ANTILOGARITHM
C     AS                ACTIVE SCALE
C     ATLK              ATTRIBUTE LINKAGE
C     AXES              ARRAY OF CIRCLE'S AXIS
C     COOR              COORDINATES(R*8)
C     DBRAD             RADIUS DOUBLE-PRECISION
C     ELSPE             ELLIPSE SPECIFICATIONS
C     FILE              DESIGN FILE NAME
C     GG                GRAPHIC GROUP
C     IAD               ARRAY CONTAINING ADDRESSES FOR LINES
C     INFILE            DATA FILE NAME
C     IRC               ERROR CODE
C     K1                NUMBER OF CLASSES FOR CIRCLES
C     K2                NUMBER OF CLASSES FOR LINES
C     LILVS             LEVELS USED FOR THE LINES
C     LINSTR            COORDINATES(I*4)
C     LNSPE             LINE SPECIFICATIONS
C     LOGA              LOGARITHM
C     LV                LEVEL
C     N                 NUMBER OF ZONES
C     NC                NUMBER OF CHARACTERS
C     NORAD             ARRAY OF RADII
C     NPER              NUMBER OF PERCENTAGES
C     ORIG              ARRAY OF COORDINATES(I*4)
C     PEMID             MIDPOINT OF ONE CLASS INTERVAL
C     PMID              ARRAY OF MIDPOINTS
C     PER               ARRAY OF PERCENTAGES
C     PERL              ARRAY OF LARGEST VALUES
C     PERS              ARRAY OF SMALLEST VALUES
C     PTS               ARRAY OF COORDINATES(R*8)
C     PUFA              POSITIONAL UNIT FACTOR
C     PUSU              NUMBER OF PU'S BY SU'S
C     RAD               RADIUS
C     REGNA             REGION NAME
C     SUMU              NUMBER OR SU'S BY MU'S
C     UNVA              UNIT VALUE
C     WIT               LINE WIDTH
C     ZL                ARRAY OF LARGEST VALUES
C     ZMID              MIDPOINT OF ONE CLASS INTERVAL
C     ZONE              ARRAY OF ZONE NUMBERS
C     ZS                ARRAY OF SMALLEST VALUES
C     ZVAL              ARRAY OF WEIGHTS
C
C     ..................................................
C
C     SUBROUTINE P2SET
C
C     LANGUAGE      : FORTRAN
```

```
C     SYSTEM        : PDP11/70 RSX-11M-PLUS
C     AUTHOR        : LISE ALLARD
C     UIC           : [266,10]
C     LAST UPDATED  : JULY 1984
C
C     COMPILE       : >F77 ILAP2SET.OBJ,ILAP2SET.LST/-SP=ILAP2SET.FTN/TR:NONE
C
C     PURPOSE       : INITIALIZE DFPI, READ THE DATA AND
C                     CALL THE SUBROUTINES P2CIR, P2LIN, P2END.
C
C     REMARK        : IT PLAYS THE ROLE OF A MAIN PROGRAM
C                     IN THE FOURTH SEGMENT OF THE OVERLAY
C                     STRUCTURE.
C
C     DFPI SUBROUTINES USED :
C     ASC2RD          CONVERTS A STRING OF ASCII CHARACTERS
C                     INTO AN EQUIVALENT STRING OF PACKED
C                     RAD50 CHARACTERS.
C     CONVER          CONVERTS FROM FLOATING POINT DOUBLE-
C                     PRECISION NUMBER WITH RANGE OF 0 TO
C                     +4294967296 TO I*4 FORTRAN FORMAT
C                     WITH RANGE OF -2147483648 TO +2147483647.
C     ERROR9          RETURNS ERROR MESSAGES TO LUN6 FOR DFPI
C                     INTERFACE ERROR.
C     INDFPI          INITIALIZE DFPI AND SET UP DESIGN FILE.
C     LKTCSI          FILE NAME CONVERSION
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     P2LIN
C     P2CIR
C     P2END
C
C     ..................................................
C
      SUBROUTINE P2SET
C
      COMMON /BLK1/ ZL(10), ZS(10), PERL(10), PERS(10), K1, K2
      COMMON /BLK2/ IAD(300), PTS(200), ORIG(200), PER(150), ZVAL(100),
     1        NORAD(100), PUFA, N, NPER
      COMMON /BLK3/ ANSER1
      COMMON /BLK4/ IOP(4)
      COMMON /IREQ/ IREQ
      COMMON /INFLAG/ INFLAG
      COMMON /WAITFL/ WAITFL
      COMMON /APREGN/ APREGN
      COMMON /DFPI/ DFPI
C
      INTEGER*2 INFILE(31), FILE(15), DGN50(7)
      INTEGER*2 REGNA(3), N, NPER, K1, K2, IOP
      INTEGER*4 ORIG
C
```

```
      REAL PER, ZVAL, NORAD, ZL, ZS, PERL, PERS
      REAL*8 PTS, PUFA, PUSU, SUMU
      CHARACTER*1 ANSER1, ANSER2
      CHARACTER*20 FMT
C
      DATA REGNA /'DF', 'PI', 'RG'/
C
C     PROGRAM IDENTIFICATION
C     ----------------------
      WRITE (5,10)
   10 FORMAT (/////, 10X, ' * PART 2 : CREATE A MAP ',
     1        ' WITH YOUR OWN CLASS INTERVALS *')
C
C     KEY-IN DESIGN FILE NAME
C     -----------------------
      WRITE (5,20)
   20 FORMAT (///, '$DESIGN FILE NAME-   ')
      READ (5,30) NC, FILE
   30 FORMAT (Q, 15A2)
      CALL LKTCSI(FILE, DGN50, NC, O, IRC)
      IF (IRC  NE  O) STOP 'LKTCSI CONVERSION ERROR'
C
C     REGION NAME CONVERSION
C     ----------------------
      CALL ASC2RD(REGNA, REGN50, 6, IRC)
      IF (IRC  NE  O) STOP 'ASC2RD CONVERSION ERROR'
C
C     DETERMINE POSITIONAL UNIT FACTOR
C     --------------------------------
   40 CONTINUE
      WRITE (5,50)
   50 FORMAT (///, '$NUMBER OF PU''S PER SU-  ')
      READ (5,*,ERR=40) PUSU
      WRITE (5,60)
   60 FORMAT (/, '$NUMBER OF SU''S PER MU=  ')
      READ (5,*,ERR=40) SUMU
C
      PUFA = PUSU * SUMU
C
C     INITIALIZE FOR DFPI
C     -------------------
      CALL INDFPI(REGN50, DGN50, O, O, O, O, IRC, 'EX')
      IF (IRC  NE  O) THEN
        WRITE (5,70) IRC
   70   FORMAT (' INDFPI ERROR IRC=', I3)
        CALL ERROR9(IRC)
        STOP
      END IF
   80 WRITE (5,90)
   90 FORMAT (/, ' *** INITIALIZATION  COMPLETED ***')
C
```

```
C      KEY-IN DATA FILE NAME
C      ---------------------
       WRITE (5,100)
   100 FORMAT (//, '$DATA FILE NAME= ')
       READ (5,110) NC, INFILE
   110 FORMAT (Q, 31A1)
       IF (NC .EQ. 0) GO TO 140
       IF (NC .GT. 30) NC = 30
       INFILE(NC + 1) = 0
       OPEN (UNIT=2,NAME=INFILE,TYPE='OLD',READONLY,ERR=82)
       GO TO 150
C
   120 WRITE (5,130)
   130 FORMAT (/, ' PROBLEM OPENING DATA FILE')
       GO TO 140
C
   140 CONTINUE
       STOP
C
C      READ THE NUMBER OF ZONES
C      ------------------------
   150 CONTINUE
       READ (2,*,ERR=340) N
C
C      READ COORDINATE FORMAT
C      ----------------------
       READ (2,160) FMT
   160 FORMAT (20A)
C
C      READ  COORDINATES
C      -----------------
       COUNT = 1
       DO 170 I = 1, N
         READ (2,FMT) PTS(COUNT), PTS(COUNT + 1)
         PTS(COUNT) = PTS(COUNT) * PUFA
         PTS(COUNT + 1) = PTS(COUNT + 1) * PUFA
         COUNT = COUNT + 2
   170 CONTINUE
C
       CALL CONVER(PTS, ORIG, COUNT - 1, IRC)
C
C      PLACE PROPORTIONAL CIRCLES
C      --------------------------
       WRITE (5,180)
   180 FORMAT (///, '$PLACE PROPORTIONAL CIRCLES?(Y/N) ')
       READ (5,190) ANSER1
   190 FORMAT (A1)
C
C      READ DATA FORMAT
C      ----------------
       READ (2,160) FMT
```

```
C
C     READ THE DATA VALUES
C     --------------------
      DO 200 I = 1, N
        READ (2,FMT) ZVAL(I)
  200 CONTINUE
      IF (ANSER1 .EQ. 'N') GO TO 270
      WRITE (5,210)
  210 FORMAT (//, '$NUMBER OF CLASSES (1-10) = ')
      READ (5,220) K1
  220 FORMAT (I3)
  230 CONTINUE
      WRITE (5,240)
  240 FORMAT (//, ' ENTER LOWER LIMITS : ')
      READ (5,*,ERR=230) (ZS(I),I=1,K1)
  250 CONTINUE
      WRITE (5,260)
  260 FORMAT (//, ' ENTER UPPER LIMITS : ')
      READ (5,*,ERR=250) (ZL(I),I=1,K1)
C
      CALL P2CIR
C
C     PLACE PROPORTIONAL FLOW LINES
C     -----------------------------
  270 CONTINUE
      WRITE (5,280)
  280 FORMAT (///, '$PLACE PROPORTIONAL FLOW LINES?(Y/N) ')
      READ (5,190) ANSER2
      IF (ANSER2 .EQ. 'N') GO TO 330
C
C     READ THE NUMBER OF PERCENTAGES
C     ------------------------------
      READ (2,*,ERR=340) NPER
C
C     READ THE PERCENTAGE FORMAT
C     --------------------------
      READ (2,160) FMT
C
C     READ PERCENTAGE AND ZONES
C     -------------------------
      J = 1
      DO 290 I = 1, NPER
        READ (2,FMT) PER(I), IAD(J), IAD(J + 1)
        J = J + 2
  290 CONTINUE
C
      WRITE (5,300)
  300 FORMAT (//, '$NUMBER OF CLASSES (1-10) = ')
      READ (5,220) K2
  310 CONTINUE
      WRITE (5,240)
```

```
      READ (5,*,ERR=310) (PERS(I),I=1,K2)
  320 CONTINUE
      WRITE (5,260)
      READ (5,*,ERR=320) (PERL(I),I=1,K2)
C
      CALL P2LIN
C
  330 CONTINUE
      CALL P2FND
      GO TO 360
C
  340 WRITE (5,350)
  350 FORMAT (/, ' * PROBLEM READING DATA FILE *')
      STOP
C
  360 CONTINUE
      CLOSE (UNIT=2,DISPOSE='SAVE')
      RETURN
      END
C
C     ...............................................
C
C     SUBROUTINE P2CIR
C
C     LANGUAGE      : FORTRAN
C     SYSTEM        : PDP11/70 RSX-11M-PLUS
C     AUTHOR        : LISE ALLARD
C     UIC           : [266,10]
C     LAST UPDATED  : JULY 1984
C
C     COMPILE       : >F77 ILAP2CIR.OBJ,ILAP2CIR.LST/-SP=ILAP2CIR.FTN/TR:NONE
C
C     PURPOSE       : ASSOCIATE A RADIUS TO EACH VALUE
C                     AND PLACE THE CIRCLES IN THE DESIGN FILE.
C
C     DFPI SUBROUTINES USED :
C     ELDFPI          PLACE ELLIPSE, 2-D ONLY.
C     ERROR9          RETURNS ERROR MESSAGES TO LUN6 FOR
C                     DFPI INTERFACE ERROR.
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     RADIUS
C
C     ...............................................
C
      SUBROUTINE P2CIR
C
      COMMON /BLK1/ ZL(10), ZS(10), PERL(10), PERS(10), K1, K2
      COMMON /BLK2/ IAD(300), PTS(200), ORIG(200), PER(150), ZVAL(100),
     1       NORAD(100), PUFA, N, NPER
      COMMON /BLK3/ ANSER1
```

```
      COMMON /IREQ/ IREQ
      COMMON /INFLAG/ INFLAG
      COMMON /WAITFL/ WAITFL
      COMMON /APREGN/ APREGN
      COMMON /DFPI/ DFPI
C
      INTEGER*2 IAD, ELSPE(5), ATLK(4), GG(2)
      INTEGER*2 K, K1, K2, N, NPER, LV, OPTION
      INTEGER*4 ORIG
C
      REAL PER, ZVAL, NORAD, RAD(10)
      REAL ZL, ZS, PERL, PERS
      REAL*8 PTS, AXES(2), AS(2), WPTS(2)
      REAL*8 AA, ANTIL, DBRAD, LOGA, PUFA, UNVA
      CHARACTER*1 ANSER1
C
      DATA AS /0.0, 0.0/, ATLK /4*0/, GG /2*0/
      DATA ELSPE /5*0/, AA /0.0/
C
      K = K1
C
C     DEFINE THE ACTIVE LEVEL
C     -----------------------
      WRITE (5,10)
   10 FORMAT (/, '$ACTIVE LEVEL (1-63) = ')
      READ (5,20) LV
   20 FORMAT (I2)
C
C     CALCULATE RADIUS
C     ----------------
   30 CONTINUE
      WRITE (5,40)
   40 FORMAT (/, '$RADIUS FOR LARGEST CIRCLE = ')
      READ (5,*,ERR=30) RAD(K)
      ZMID = ((ZL(K) - ZS(K))/2) + ZS(K)
      LOGA = ALOG10(ZMID)
      LOGA = LOGA * 0.5718
      ANTIL = 10 ** LOGA
      UNVA = RAD(K) / ANTIL
      KK = K - 1
      DO 50 I = 1, KK
        ZMID = ((ZL(I) - ZS(I))/2) + ZS(I)
        RAD(I) = RADIUS(ZMID,UNVA)
   50 CONTINUE
C
C     PRINT RADIUS
C     ------------
      WRITE (5,60)
   60 FORMAT (//, ' RADIUS ASSOCIATED TO EACH CLASS')
      DO 80 I = 1, K
        WRITE (5,70) RAD(I)
```

```
   70   FORMAT (10X, F10.6)
   80 CONTINUE
C
      WRITE (5,90)
   90 FORMAT (//, '$MODIFY THE RADIUS OR CONTINUE?', ' (M/C) ')
      READ (5,100) OPTION
  100 FORMAT (A1)
      IF (OPTION .EQ. 'M') GO TO 30
C
C     PLACE CIRCLES
C     -------------
  110 CONTINUE
      J = 1
      DO 150 I = 1, N
        DO 120 L = K, 1, -1
          IF (ZVAL(I) .GE. ZS(L)) THEN
            DBRAD = DBLE(RAD(L))
            AXES(1) = DBRAD * PUFA
            AXES(2) = AXES(1)
            GO TO 130
          ELSE
            GO TO 120
          END IF
  120   CONTINUE
  130   NORAD(I) = RAD(L)
        WPTS(1) = ORIG(J)
        WPTS(2) = ORIG(J + 1)
        CALL ELDFPI(GG, LV, ELSPE, AXES, WPTS, AA, IRC, ATLK)
        IF (IRC .NE. 0) THEN
          WRITE (5,140) IRC
  140     FORMAT (' ERROR IN ELDFPI IRC= ', I3)
          CALL ERROR9(IRC)
          STOP
        END IF
        J = J + 2
  150 CONTINUE
C
      WRITE (5,160)
  160 FORMAT (/, ' ** PROPORTIONAL CIRCLES COMPLETED **')
C
      RETURN
      END
C
C     ..............................................
C
C     FUNCTION RADIUS
C
C     LANGUAGE        : FORTRAN
C     SYSTEM          : PDP11/70 RSX-11M-PLUS
C     AUTHOR          : LISE ALLARD
C     UIC             : [266,10]
```

```
C      LAST UPDATED : JULY 1984
C
C      PURPOSE        : CALCULATE A PROPORTIONAL RADIUS
C                       VALUF ACCORDING TO THE PSYCHOLOGICAL
C                       METHOD.
C
C      REFERENCF      : ROBINSON A. ET AL.(1978), ELEMENTS
C                       OF CARTOGRAPHY 4TH EDITION.
C
C      ..........................................................
C
C
       FUNCTION RADIUS(ZMID, UNVA)
C
C      ARGUMENTS DEFINITION
C      --------------------
C      ZMID            MIDPOINT IN ONE CLASS INTERVAL
C      UNVA            UNIT VALUE
C
       REAL ZMID
       REAL*8 LOGA, ANTIL, UNVA
C
       LOGA = ALOG10(ZMID)
       LOGA = LOGA * 0.5718
       ANTIL = 10 ** LOGA
       RADIUS = ANTIL * UNVA
C
       RETURN
       END
C
C      ..........................................................
C
C      SUBROUTINE P2LIN
C
C      LANGUAGE       : FORTRAN
C      SYSTEM         : PDP11/70 RSX-11M-PLUS
C      AUTHOR         : LISE ALLARD
C      UIC            : [266,10]
C      LAST UPDATED   : JULY 1984
C
C      COMPILE        : >F77 ILAP2LIN.OBJ,ILAP2LIN.LST/-SP=ILAP2LIN.FTN/TR:NONE
C
C      PURPOSE        : ASSOCIATE A LINE WIDTH TO EACH PERCENTAGE
C                       AND PLACE THE LINES IN THE DESIGN FILE.
C
C      DFPI SUBROUTINES USED :
C      CONVER           CONVERTS FROM FLOATING POINT DOUBLE-
C                       PRECISION NUMBER WITH RANGE OF 0 TO
C                       +4294967296 TO I*4 FORTRAN FORMAT
C                       WITH RANGE OF -2147483648 TO +2147483647.
C      ERROR9           RETURNS ERROR MESSAGES TO LUN6 FOR DFPI
```

```
C                    INTERFACE ERROR.
C     LSDFPI         PLACE LINE STRINGS.
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     CTLINE
C     IWIDTH
C
C     ..............................................................
C
      SUBROUTINE P2LIN
C
      COMMON /BLK1/ ZL(10), ZS(10), PERL(10), PERS(10), K1, K2
      COMMON /BLK2/ IAD(300), PTS(200), ORIG(200), PER(150), ZVAL(100),
     1       NORAD(100), PUFA, N, NPER
      COMMON /BLK3/ ANSER1
      COMMON /IREQ/ IREQ
      COMMON /INFLAG/ INFLAG
      COMMON /WAITFL/ WAITFL
      COMMON /APREGN/ APREGN
      COMMON /DFPI/ DFPI
C
      INTEGER*2 IAD, WIT(10), LNSPE(5), ATLK(4)
      INTEGER*2 GG(2), K, K1, K2, N, NPER, NLINE
      INTEGER*4 ORIG, LINSTR(4)
      CHARACTER*1 ANSER1, OPTION
C
      REAL PER, ZVAL, NORAD, ZL, ZS, PERL, PERS, PMID(10)
      REAL*8 PTS, COOR(4), RD(2), PUFA
C
      DATA NLINE /4/, LNSPE /5*0/
      DATA GG /2*0/, ATLK /4*0/
C
      K = K2
C
C     DEFINE THE ACTIVE LEVEL
C     -----------------------
   10 CONTINUE
      WRITE (5,20)
   20 FORMAT (/, '$ACTIVE LEVEL (1-63) = ')
      READ (5,30) LV
   30 FORMAT (I2)
      LILVS = LV + K - 1
      WRITE (5,40) K, LV, LILVS
   40 FORMAT (//, ' LEVELS TO BE USED FOR A', I3, '      P = ', I3,
     1       ' TO ', I3, //, '$MODIFY OR CONTINUE?(M
      READ (5,120) OPTION
      IF (OPTION .EQ. 'M') GO TO 10
C
C     CALCULATE LINE THICKNESS
C     ------------------------
   50 CONTINUE
```

```
      WRITE (5,60)
   60 FORMAT (/, '$THICKNESS FOR BIGGEST LINE (0-31) = ')
      READ (5,*,ERR=50) WIT(K)
      PMID(K) = ((PERL(K) - PERS(K))/2) + PERS(K)
      UNVA = WIT(K) / PMID(K)
      KK = K - 1
      DO 70 I = 1, KK
        PEMID = ((PERL(I) - PERS(I))/2) + PERS(I)
        WIT(I) = IWIDTH(PEMID,UNVA)
   70 CONTINUE
C
C     PRINT LINE THICKNESS
C     --------------------
      WRITE (5,80)
   80 FORMAT (//, ' LINE THICKNESS ASSOCIATED TO EACH CLASS')
      DO 100 I = 1, K
        WRITE (5,90) WIT(I)
   90   FORMAT (10X, I3)
  100 CONTINUE
C
      WRITE (5,110)
  110 FORMAT (//, '$MODIFY THE LINE THICKNESS OR ', 'CONTINUE?(M/C) ')
      READ (5,120) OPTION
  120 FORMAT (A1)
      IF (OPTION .EQ. 'M') GO TO 50
C
C     PLACE LINES
C     -----------
  130 CONTINUE
      DO 180 I = 1, NPER
        DO 140 L = K, 1, -1
          IF (PER(I) .GE. PERS(L)) THEN
            LNSPE(4) = WIT(L)
            GO TO 150
          ELSE
            GO TO 140
          END IF
  140   CONTINUE
  150   CONTINUE
        IA1 = IAD((2*I) - 1)
        IA2 = IAD(2*I)
        COOR(1) = PTS((2*IA1) - 1) / PUFA
        COOR(2) = PTS(2*IA1) / PUFA
        COOR(3) = PTS((2*IA2) - 1) / PUFA
        COOR(4) = PTS(2*IA2) / PUFA
        IF (ANSER1 .EQ. 'N') GO TO 160
        IF (IA1 .EQ. IA2) THEN
          COOR(3) = COOR(1) + (0.5*NORAD(IA1))
          COOR(1) = COOR(1) - (0.5*NORAD(IA1))
          COOR(4) = COOR(2)
          GO TO 160
```

```
      END IF
      RD(1) = NORAD(IA1)
      RD(2) = NORAD(IA2)
      CALL CTLINE(COOR, RD)
  160 CONTINUE
      COOR(1) = COOR(1) * PUFA
      COOR(2) = COOR(2) * PUFA
      COOR(3) = COOR(3) * PUFA
      COOR(4) = COOR(4) * PUFA
      CALL CONVER(COOR, LINSTR, NLINE, IRC)
      LV = LV + L - 1
      CALL LSDFPI(GG, LV, LNSPE, LINSTR, 2, IRC, ATLK)
      LV = LV - L + 1
      IF (IRC .NE. 0) THEN
        WRITE (5,170) IRC
  170   FORMAT (' ERROR IN LSDFPI IRC=', I3)
        CALL ERROR9(IRC)
        STOP
      END IF
  180 CONTINUE
C
  190 CONTINUE
      WRITE (5,200)
  200 FORMAT (/, ' ** PROPORTIONAL LINES COMPLETED **')
C
      RETURN
      END
C
C     ...............................................
C
C     FUNCTION IWIDTH
C
C     PURPOSE      : CALCULATES A PROPORTIONAL LINE
C                    WIDTH.
C
C     ...............................................
C
      FUNCTION IWIDTH(PEMID, UNVA)
C
C     ARGUMENTS DEFINITION
C     --------------------
C     PEMID          MIDPOINT OF ONE CLASS INTERVAL
C     UNVA           UNIT VALUE
C
      REAL PEMID, UNVA
C
      IWIDTH = ININT(PEMID*UNVA)
C
      RETURN
      END
C
```

```
C     ..................................................
C
C     SUBROUTINE CTLINE
C
C     LANGUAGE      : FORTRAN
C     SYSTEM        : PDP11/70 RSX-11M-PLUS
C     AUTHOR        : LISE ALLARD
C     UIC           : [266,10]
C     LAST UPDATED  : JULY 1984
C
C     COMPILE       : >F77 ILAP2LIN.OBJ,ILAP2LIN.LST/-SP=ILAP2LIN.FTN/TR:NONE
C
C     PURPOSE       : CALCULATES TWO PAIRS OF COORDINATES
C                     TO PLACE A LINE BETWEEN TWO CIRCLES.
C
C     DFPI SUBROUTINES USED
C     NONE
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     NONE
C
C     ..................................................
C
      SUBROUTINE CTLINE(COOR, RD)
C
C     ARGUMENTS DEFINITION
C     --------------------
C     COOR         ARRAY CONTAINING TWO PAIRS OF COORDINATES
C     RD           ARRAY OF RADII OF TWO CIRCLES
C
C     LOCAL VARIABLES
C     ---------------
C     M            SLOPE
C     XA           X COORDINATE (ORIGIN OF CIRCLE)
C     YA           Y COORDINATE (ORIGIN OF CIRCLE)
C     XC           X CCORDINATE (EDGE OF CIRCLE)
C     YC           Y COORDIANTE (EDGE OF CIRCLE)
C     RR           RADIUS
C
      REAL*8 COOR(4), RD(2), M, XA, YA, XC, YC, ROOT1, ROOT2, R(2)
      REAL*8 DY, DX
C
      DY = COOR(4) - COOR(2)
      DX = COOR(3) - COOR(1)
      IF (DY .EQ. 0) DY = 1E-10
      IF (DX .EQ. 0) DX = 1E-10
C
      M = DY / DX
C
      DO 10 I = 1, 2
        RR = RD(I)
```

```
      II = I
      IF (I .EQ. 2) II = 3
      XA = COOR(II)
      YA = COOR(II + 1)
      ROOT1 = SQRT(RR**2/(1 + (M**2)))
C
      IF (COOR(3) .GT. COOR(1)) THEN
        XC = XA - ((-1)**I) * RC
      ELSE
        XC = XA + ((-1)**I) * ROOT1
      END IF
C
      ROOT2 = SQRT(RR**2 - ((XC - XA)**2))
C
      IF (COOR(4) .GT. COOR(2)) THEN
        YC = YA - ((-1)**I) * ROOT2
      ELSE
        YC = YA + ((-1)**I) * ROOT2
      END IF
C
      COOR(II) = XC
      COOR(II + 1) = YC
   10 CONTINUE
C
      RETURN
      END
C
C     ..........................................................
C
C     SUBROUTINE P2END
C
C     LANGUAGE     : FORTRAN
C     SYSTEM       : PDP11/70 RSX-11M-PLUS
C     AUTHOR       : LISE ALLARD
C     UIC          : [266,10]
C     LAST UPDATED : JULY 1984
C
C     COMPILE      : >F77 ILAP2END.OBJ,ILAP2END.LST/-SP=ILAP2END.FTN/TR:NONE
C
C     PURPOSE      : TERMINATES DFPI
C
C     DFPI SUBROUTINES USED :
C     DEDFPI        . DETACH FROM DFPI
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     NONE
C
C     ..........................................................
C
      SUBROUTINE P2END
C
```

```
      COMMON /IREQ/ IREQ
      COMMON /INFLAG/ INFLAG
      COMMON /WAITFL/ WAITFL
      COMMON /APREGN/ APREGN
      COMMON /DFPI/ DFPI
C
C     TERMINATE DFPI
C     --------------
      IARG = 1
      IF (IRC .NE. 0) IARG = 0
      CALL DEDFPI(IARG)
C
      RETURN
      END
C
C     ************************************************
C
C
C     O V E R L A Y   S E G M E N T   F I V E
C
C
C     ************************************************
C
C     VARIABLES        DESCRIPTION
C     ---------        -----------
C     AA               ACTIVE ANGLE
C     ANTIL            ANTILOGARITHM
C     AS               ACTIVE SCALE
C     ATLK             ATTRIBUTE LINKAGE
C     AXES             ARRAY OF CIRCLE'S AXIS
C     COOR             COORDINATES(R*8)
C     DBRAD            RADIUS DOUBLE-PRECISION
C     ELSPE            ELLIPSE SPECIFICATIONS
C     FILE             DESIGN FILE NAME
C     GG               GRAPHIC GROUP
C     IAD              ARRAY CONTAINING ADDRESSES FOR LINES
C     INFILE           DATA FILE NAME
C     IRC              ERROR CODE
C     K1               NUMBER OF CLASSES FOR CIRCLES
C     K2               NUMBER OF CLASSES FOR LINES
C     LILVS            LEVELS USED FOR THE LINES
C     LINSTR           COORDINATES(I*4)
C     LNSPE            LINE SPECIFICATIONS
C     LOGA             LOGARITHM
C     LV               LEVEL
C     N                NUMBER OF ZONES
C     NC               NUMBER OF CHARACTERS
C     NORAD            ARRAY OF RADII
C     NPER             NUMBER OF PERCENTAGES
C     ORIG             ARRAY OF COORDINATES(I*4)
C     PEMID            MIDPOINT OF ONE CLASS INTERVAL
```

```
C     PER              ARRAY OF PERCENTAGES
C     PERL             ARRAY OF LARGEST VALUES
C     PERMAX           PERCENTAGE MAXIMUM
C     PERMIN           PERCENTAGE MINIMUM
C     PERS             ARRAY OF SMALLEST VALUES
C     PMID             ARRAY OF MIDPOINTS.
C     PTS              ARRAY OF COORDINATES(R*8)
C     PUFA             POSITIONAL UNIT FACTOR
C     PUSU             NUMBER OF PU'S BY SU'S
C     RAD              RADIUS
C     REGNA            REGION NAME
C     SUMU             NUMBER OR SU'S BY MU'S
C     UNVA             UNIT VALUE
C     WIT              LINE WIDTH
C     ZL               ARRAY OF LARGEST VALUES
C     ZMID             MIDPOINT OF ONE CLASS INTERVAL.
C     ZONE             ARRAY OF ZONE NUMBERS
C     ZS               ARRAY OF SMALLEST VALUES
C     ZVAL             ARRAY OF WEIGHTS
C
C     ..........................................................
C
C     SUBROUTINE P3SET
C
C     LANGUAGE       : FORTRAN
C     SYSTEM         : PDP11/70 RSX-11M-PLUS
C     AUTHOR         : LISE ALLARD
C     UIC            : [266,10]
C     LAST UPDATED   : JULY 1984
C
C     COMPILE        : >F77 ILAP3SET.OBJ,ILAP3SET.LST/-SP=ILAP3SET.FTN/TR:NONE
C
C     PURPOSE        : INITIALIZE DFPI, READ THE DATA AND
C                      CALL THE SUBROUTINES P3CIR, P3LIN, P3END.
C
C     REMARK         : IT PLAYS THE ROLE OF A MAIN PROGRAM
C                      IN THE FIFTH SEGMENT OF THE OVERLAY
C                      STRUCTURE.
C
C     DFPI SUBROUTINES USED :
C     ASC2RD           CONVERTS A STRING OF ASCII CHARACTERS,
C                      INTO AN EQUIVALENT STRING OF PACKED
C                      RAD50 CHARACTERS.
C     CONVER           CONVERTS FROM FLOATING POINT DOUBLE-
C                      PRECISION NUMBER WITH RANGE OF 0 TO
C                      +4294967296 TO I*4 FORTRAN FORMAT
C                      WITH RANGE OF -2147483648 TO +2147483647.
C     ERROR9           RETURNS ERROR MESSAGES TO LUN6 FOR DFPI
C                      INTERFACE ERROR.
C     INDFPI           INITIALIZE DFPI AND SET UP DESIGN FILE.
C     LKTCSI           FILE NAME CONVERSION
```

```
C
C     SUBR(    ... ... UNCTIONS REQUIRED :
C     P3CIR
C     P3LIN
C     P3FND
C
C
C     . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .
            NE P3SFT
C
      COMMO   BLK2/ IAD(300), PTS(200), ORIG(200), PER(150), ZVAL(100),
            NORAD(100), PUFA, N, NPER
      (     /BLK3/ ANSER1
      COMMON /BLK4/ IOP(4)
      COMMON /IREQ/ IREQ
      COMMON /INFLAG/ INFLAG
      COMMON /WAITFL/ WAITFL
      COMMON /APREGN/ APREGN
      COMMON /DFPI/ DFPI
C
      INTEGER*2 INFILE(31), FILE(15), DGN50(7)
      INTEGER*2 REGNA(3), N, NPER, IOP
      INTEGER*4 ORIG
C
      REAL PER, ZVAL, NORAD
      REAL*8 PTS, PUFA, PUSU, SUMU
      CHARACTER*1 ANSER1, ANSER2
      CHARACTER*20 FMT
C
      DATA REGNA /'DF', 'PI', 'RG'/
C
C     PROGRAM IDENTIFICATION
C     ----------------------
      WRITE (5,10)
   10 FORMAT (/////, 10X, ' * PART 4 : CREATE A ', 'NON-CLASSED MAP *')
C
C     KEY-IN DESIGN FILE NAME
C     -----------------------
      WRITE (5,20)
   20 FORMAT (///, '$DESIGN FILE NAME= ')
      READ (5,30) NC, FILE
   30 FORMAT (Q, 15A2)
      CALL LKTCSI(FILE, DGN50, NC, 0, IRC)
      IF (IRC .NE. 0) STOP 'LKTCSI CONVERSION ERROR'
C
C     REGION NAME CONVERSION
C     ----------------------
      CALL ASC2RD(REGNA, REGN50, 6, IRC)
      IF (IRC .NE. 0) STOP 'ASC2RD CONVERSION ERROR'
C
C     DETERMINE POSITIONAL UNIT FACTOR
```

```
C       ------------------------------
   40 CONTINUE
      WRITE (5,50)
   50 FORMAT (///, '$NUMBER OF PU''S PER SU= ')
      READ (5,*,ERR=40) PUSU
      WRITE (5,60)
   60 FORMAT (/, '$NUMBER OF SU''S PER MU= ')
      READ (5,*,ERR=40) SUMU
C
      PUFA = PUSU * SUMU
C
C     INITIALIZE FOR DFPI
C     -------------------
      CALL INDFPI(REGN50, DGN50, O, O, O, O, IRC, 'EX')
      IF (IRC .NE. O) THEN
        WRITE (5,70) IRC
   70   FORMAT (' INDFPI ERROR IRC=', I3)
        CALL ERROR9(IRC)
        STOP
      END IF
   80 WRITE (5,90)
   90 FORMAT (/, ' *** INITIALIZATION  COMPLETED ***')
C
C     KEY-IN DATA FILE NAME
C     ---------------------
      WRITE (5,100)
  100 FORMAT (//, '$DATA FILE NAME= ')
      READ (5,110) NC, INFILE
  110 FORMAT (Q, 31A1)
      IF (NC .EQ. O) GO TO 140
      IF (NC .GT. 30) NC = 30
      INFILE(NC + 1) = O
      OPEN (UNIT=2,NAME=INFILE,TYPE='OLD',READONLY,ERR=82)
      GO TO 150
C
  120 WRITE (5,130)
  130 FORMAT (/, ' PROBLEM OPENING DATA FILE')
      GO TO 140
C
  140 CONTINUE
      STOP
C
C     READ THE NUMBER OF ZONES
C     ------------------------
  150 CONTINUE
      READ (2,*,ERR=250) N
C
C     READ COORDINATES FORMAT
C     -----------------------
      READ (2,160) FMT
  160 FORMAT (20A)
```

```
C
C     READ COORDINATES
C     ----------------
      COUNT = 1
      DO 170 I = 1, N
        READ (2,FMT) PTS(COUNT), PTS(COUNT + 1)
        PTS(COUNT) = PTS(COUNT) * PUFA
        PTS(COUNT + 1) = PTS(COUNT + 1) * PUFA
        COUNT = COUNT + 2
  170 CONTINUE
C
      CALL CONVER(PTS, ORIG, COUNT - 1, IRC)
C
C     PLACE PROPORTIONAL CIRCLES
C     --------------------------
      WRITE (5,180)
  180 FORMAT (///, '$PLACE PROPORTIONAL CIRCLES?(Y/N) ')
      READ (5,190) ANSER1
  190 FORMAT (A1)
C
C     READ DATA FORMAT
C     ----------------
      READ (2,160) FMT
C
C     READ THE DATA VALUES
C     --------------------
      DO 200 I = 1, N
        READ (2,FMT) ZVAL(I)
  200 CONTINUE
      IF (ANSER1 .EQ. 'N') GO TO 210
C
      CALL P3CIR
C
C     PLACE PROPORTIONAL FLOW L  ES
C     -----------------------------
  210 CONTINUE
      WRITE (5,220)
  220 FORMAT (///, '$PLACE PROPORTIONAL FLOW LINES?(Y/N) ')
      READ (5,190) ANSER2
      IF (ANSER2 .EQ. 'N') GO TO 240
C
C     READ THE NUMBER OF PERCENTAGES
C     ------------------------------
      READ (2,*,ERR=250) NPER
C
C     READ THE PERCENTAGE FORMAT
C     --------------------------
      READ (2,160) FMT
C
C     READ THE PERCENTAGE AND ZONES
C     -----------------------------
```

```
      J = 1
      DO 230 I = 1, NPER
        READ (2,FMT) PER(I), IAD(J), IAD(J + 1)
        J = J + 2
  230 CONTINUE
C
      CALL P3LIN
C
  240 CONTINUE
      CALL P3END
      GO TO 270
C
  250 WRITE (5,260)
  260 FORMAT (/, ' * PROBLEM READING DATA FILE *')
      STOP
C
C
  270 CONTINUE
      RETURN
      END
C
C     .............................................................
C
C     SUBROUTINE P3CIR
C
C     LANGUAGE      : FORTRAN
C     SYSTEM        : PDP11/70 RSX-11M-PLUS
C     AUTHOR        : LISE ALLARD
C     UIC           : [266,10]
C     LAST UPDATED  : JULY 1984
C
C     COMPILE       : >F77 ILAP3CIR.OBJ,ILAP3CIR.LST/-SP=ILAP3CIR.FTN/TR:NONE
C
C     PURPOSE       : ASSOCIATE A RADIUS TO EACH VALUE
C                     AND PLACE THE CIRCLES IN THE DESIGN FILE.
C
C     DFPI SUBROUTINES USED :
C     ELDFPI          PLACE ELLIPSE, 2-D ONLY.
C     ERROR9          RETURNS ERROR MESSAGES TO LUN6 FOR
C                     DFPI INTERFACE ERROR.
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     RADIUS
C
C     .............................................................
C
      SUBROUTINE P3CIR
C
      COMMON /BLK2/ IAD(300), PTS(200), ORIG(200), PER(150), ZVAL(100),
     1       NORAD(100), PUFA, N, NPER
      COMMON /BLK3/ ANSER1
```

```
      COMMON /BLK4/ IOP(4)
      COMMON /IREQ/ IREQ
      COMMON /INFLAG/ INFLAG
      COMMON /WAITFL/ WAITFL
      COMMON /APREGN/ APREGN
      COMMON /DEPI/ DEPI
C
      INTEGER*2 IAD, ELSPE(5), ATLK(4), GG(2)
      INTEGER*2 IOP, N, NPER, LV, OPTION
      INTEGER*4 ORIG
C
      REAL ZMIN, ZVALI
      REAL PER, ZVAL, NORAD, RAD
      REAL*8 PTS, AXES(2), AS(2), WPTS(2)
      REAL*8 AA, ANTIL, DBRAD, LOGA, PUFA, UNVA
      CHARACTER*1 ANSER1
C
      DATA AS /0.0, 0.0/, ATLK /4*0/, GG /2*0/
      DATA ELSPE /5*0/, AA /0.0/
C
C     DEFINE THE ACTIVE LEVEL
C
      WRITE (5,10)
   10 FORMAT (/, '$ACTIVE LEVEL (1-63) = ')
      READ (5,20) LV
   20 FORMAT (I2)
C
C     FIND THE SMALLEST VALUE
C     -----------------------
      ZMIN = ZVAL(1)
      DO 30 I = 2, N
        IF (ZVAL(I) .LT. ZMIN) ZMIN = ZVAL(I)
   30 CONTINUE
C
C     CALCULATE RADIUS
C     ----------------
   40 CONTINUE
      WRITE (5,50)
   50 FORMAT (/, '$RADIUS FOR SMALLEST CIRCLE= ')
      READ (5,*,ERR=40) RAD
      LOGA = ALOG10(ZMIN)
      LOGA = LOGA * 0.5718
      ANTIL = 10 ** LOGA
      UNVA = RAD / ANTIL
C
C     PLACE CIRCLES
C     -------------
   60 CONTINUE
      J = 1
      DO 90 I = 1, N
        ZVALI = ZVAL(I)
```

```
      RAD = RADIUS(ZVALI,UNVA)
      DBRAD = DBLE(RAD)
      AXES(1) = DBRAD * PUFA
      AXES(2) = AXES(1)
   70 NORAD(I) = RAD
      WPTS(1) = ORIG(J)
      WPTS(2) = ORIG(J + 1)
      CALL ELDFPI(GG, LV, ELSPE, AXES, WPTS, AA, IRC, ATLK)
      IF (IRC .NE. 0) THEN
        WRITE (5,80) IRC
   80   FORMAT (' ERROR IN ELDFPI IRC= ', I3)
        CALL ERROR9(IRC)
        STOP
      END IF
      J = J + 2
   90 CONTINUE
C
      WRITE (5,100)
  100 FORMAT (/, ' ** PROPORTIONAL CIRCLES COMPLETED **')
C
      RETURN
      END
C
C     ..................................................
C
C     FUNCTION RADIUS
C
C     LANGUAGE     : FORTRAN
C     SYSTEM       : PDP11/70 RSX-11MPLUS
C     AUTHOR       : LISE ALLARD
C     UIC          : [266,10]
C     LAST UPDATED : FEBRUARY 1984
C
C     PURPOSE      : CALCULATE A PROPORTIONAL RADIUS
C                    VALUE ACCORDING TO THE PSYCHOLOGICAL
C                    METHOD.
C
C     REFERENCE    : ROBINSON A. ET AL.(1978), ELEMENTS
C                    OF CARTOGRAPHY 4TH EDITION.
C
C     ................................................
C
C
      FUNCTION RADIUS(ZLI, UNVA)
C
C     ARGUMENTS DEFINITION
C     --------------------
C     ZLI            LARGER LIMIT OF ONE CLASS INTERVAL
C     UNVA           UNIT VALUE
C
      REAL ZLI
```

```
      REAL*8 LOGA, ANTIL, UNVA
C
      LOGA = ALOG10(ZLI)
      LOGA = LOGA * 0.5718
      ANTIL = 10 ** LOGA
      RADIUS = ANTIL * UNVA
C
      RETURN
      END
C
C
C
C     SUBROUTINE P3LIN
C
C     LANGUAGE        FORTRAN
C     SYSTEM          PDP11/70 RSX-11M PLUS
C     AUTHOR          LISE ALLARD
C     UIC             [266,10]
C     LAST UPDATED    JULY 1984
C
C     COMPILE         >F77 IL  IN OBJ.ILAP3LIN LST/-SP=ILAP3LIN.FTN/TR:NONE
C
C     PURPOSE         ASSOCIATE A LINE WIDTH TO EACH PERCENTAGE
C                     AND PLACE THE LINES IN THE DESIGN FILE
C
C     DFPI SUBROUTINES USED
C     CONVER          CONVERTS FROM FLOATING POINT DOUBLE-
C                     PRECISION NUMBER WITH RANGE OF 0 TO
C                     +4294967296 TO I*4 FORTRAN FORMAT
C                     WITH RANGE OF -2147483648 TO +2147483647.
C     ERROR9          RETURNS ERROR MESSAGES TO LUN6 FOR DFPI
C                     INTERFACE ERROR.
C     LSDFPI          PLACE LINE STRINGS.
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     CTLINE
C
C     ..............................................................
C
      SUBROUTINE P3LIN
C
      COMMON /BLK2/ IAD(300), PTS(200), ORIG(200), PER(150), ZVAL(100),
     1       NORAD(100), PUFA, N, NPER
      COMMON /BLK3/ ANSER1
      COMMON /BLK4/ IOP(4)
      COMMON /IREQ/ IREQ
      COMMON /INFLAG/ INFLAG
      COMMON /WAITFL/ WAITFL
      COMMON /APREGN/ APREGN
      COMMON /DFPI/ DFPI
C
```

```
      INTEGER*2 IAD, LNSPE(5), ATLK(4)
      INTEGER*2 IOP, GG(2), N, NPER, NLINE
      INTEGER*4 ORIG, LINSTR(4)
      CHARACTER*1 ANSER1, OPTION
C
      REAL PER, ZVAL, NORAD, SCALE, PERMIN, PERMAX, DZ
      REAL*8 PTS, COOR(4), RD(2), PUFA
C
C
      DATA NLINE /4/, LNSPE /5*0/
      DATA GG /2*0/, ATLK /4*0/
C
C     DEFINE THE ACTIVE LEVEL
C     ----------------------
   10 CONTINUE
      WRITE (5,20)
   20 FORMAT (/, '$ACTIVE LEVEL (1-63) = ')
      READ (5,30) LV
   30 FORMAT (I2)
C
C     CALCULATE LINE THICKNESS AND PLACE LINES
C     ---------------------------------------
   40 CONTINUE
      PERMIN = PER(1)
      PERMAX = PER(2)
      DO 50 I = 2, NPER
        IF (PER(I) .LT. PERMIN) PERMIN = PER(I)
        IF (PER(I) .GT. PERMAX) PERMAX = PER(I)
   50 CONTINUE
C
      DZ = PERMAX - PERMIN
C
      DO 80 I = 1, NPER
        SCALE = (PER(I) - PERMIN) / DZ
        LNSPE(4) = ININT(SCALE*31)
        IA1 = IAD((2*I) - 1)
        IA2 = IAD(2*I)
        COOR(1) = PTS((2*IA1) - 1) / PUFA
        COOR(2) = PTS(2*IA1) / PUFA
        COOR(3) = PTS((2*IA2) - 1) / PUFA
        COOR(4) = PTS(2*IA2) / PUFA
        IF (ANSER1 .EQ. 'N') GO TO 60
        IF (IA1 .EQ. IA2) THEN
          COOR(3) = COOR(1) + (0.5*NORAD(IA1))
          COOR(1) = COOR(1) - (0.5*NORAD(IA1))
          COOR(4) = COOR(2)
          GO TO 60
        END IF
        RD(1) = NORAD(IA1)
        RD(2) = NORAD(IA2)
        CALL CTLINE(COOR, RD)
```

```
   60   CONTINUE
        COOR(1) = COOR(1) * PUFA
        COOR(2) = COOR(2) * PUFA
        COOR(3) = COOR(3) * PUFA
        COOR(4) = COOR(4) * PUFA
        CALL CONVER(COOR, LINSTR, NLINE, IRC)
        LV = LV + LNSPE(4)
        CALL LSDFPI(GG, LV, LNSPE, LINSTR, 2, IRC, ATLK)
        LV = LV - LNSPE(4)
        IF (IRC .NE. 0) THEN
          WRITE (5,70) IRC
   70     FORMAT (' ERROR IN LSDFPI IRC=', I3)
          CALL ERROR9(IRC)
          STOP
        END IF
   80 CONTINUE
C
      WRITE (5,90)
   90 FORMAT (/, ' ** PROPORTIONAL LINES COMPLETED **')
C
      RETURN
      END
C
C     ...............................................
C
C     SUBROUTINE CTLINE
C
C     LANGUAGE     : FORTRAN
C     SYSTEM       : PDP11/70 RSX-11M-PLUS
C     AUTHOR       : LISE ALLARD
C     UIC          : [266,10]
C     LAST UPDATED : JULY 1984
C
C     COMPILE      : >F77 ILAP3LIN.OBJ,ILAP3LIN.LST/-SP=ILAP3LIN.FTN
C
C     PURPOSE      : CALCULATES TWO PAIRS OF COORDINATES
C                    TO PLACE A LINE BETWEEN TWO CIRCLES.
C
C     DFPI SUBROUTINES USED :
C     NONE
C
C     SUBROUTINES AND FUNCTIONS REQUIRED :
C     NONE
C
C     ...............................................
C
      SUBROUTINE CTLINE(COOR, RD)
C
C     ARGUMENTS DEFINITION
C     --------------------
C     COOR        ARRAY CONTAINING TWO PAIRS OF COORDINATES
```

```
C      RD            ARRAY OF RADII OF TWO CIRCLES
C
C      LOCAL VARIABLES
C      ---------------
C      M             SLOPE
C      XA            X COORDINATE (ORIGIN OF CIRCLE)
C      YA            Y COORDINATE (ORIGIN OF CIRCLE)
C      XC            X CCORDINATE (EDGE OF CIRCLE)
C      YC            Y COORDIANTE (EDGE OF CIRCLE)
C      RR            RADIUS
C
       REAL*8 COOR(4), RD(2), M, XA, YA, XC, YC, ROOT1, ROOT2, R(2)
       REAL*8 DY, DX
C
       DY = COOR(4) - COOR(2)
       DX = COOR(3) - COOR(1)
       IF (DY .EQ. O) DY = 1E-10
       IF (DX .EQ. O) DX = 1E-10
C
       M = DY / DX
C
       DO 10 I = 1, 2
         RR = RD(I)
         II = I
         IF (I .EQ. 2) II = 3
         XA = COOR(II)
         YA = COOR(II + 1)
         ROOT1 = SQRT(RR**2/(1 + (M**2)))
C
         IF (COOR(3) .GT. COOR(1)) THEN
           XC = XA - ((-1)**I) * ROOT1
         ELSE
           XC = XA + ((-1)**I) * ROOT1
         END IF
C
         ROOT2 = SQRT(RR**2 - ((XC - XA)**2))
C
         IF (COOR(4) .GT. COOR(2)) THEN
           YC = YA - ((-1)**I) * ROOT2
         ELSE
           YC = YA + ((-1)**I) * ROOT2
         END IF
C
         COOR(II) = XC
         COOR(II + 1) = YC
    10 CONTINUE
C
       RETURN
       END
C
C
```

```
C
C     SUBROUTINE P3END
C
C     LANGUAGE        FORTRAN
C     SYSTEM          PDP11/70 RSX-11M-PLUS
C     AUTHOR          LISE ALLARD
C     UIC             [266,10]
C     LAST UPDATED    JULY 1984
C
C     COMPILE         >F77 ILAP3END OBJ,ILAP3END LST/-SP=ILAP3END FTN/TR NONE
C
C     PURPOSE         TERMINATES DFPI
C
C     DFPI SUBROUTINES USED
C     DEDFPI          DETACH FROM DFPI
C
C     SUBROUTINES AND FUNCTIONS REQUIRED
C     NONE
C
C
C
      SUBROUTINE P3END
C
      COMMON /IREQ/ IREQ
      COMMON /INFLAG/ INFLAG
      COMMON /WAITFL/ WAITFL
      COMMON /APREGN/ APREGN
      COMMON /DFPI/ DFPI
C
C     TERMINATE DFPI
C     --------------
      IARG = 1
      IF (IRC .NE. O) IARG = O
      CALL DEDFPI(IARG)
C
      RETURN
      END
C
C     ********************************************************
C
C
C     O V E R L A Y   S E G M E N T   S I X
C
C
C     ********************************************************
C
C     VARIABLES       DESCRIPTION
C     ---------       -----------
C     AA              ACTIVE ANGLE
C     AC              ACTIVE CELL
C     ANTIL           ANTILOGARITHM
```

```
C     AS          ACTIVE SCALE
C     ATLK        ATTRIBUTE LINKAGE
C     AXES        ARRAY OF CIRCLE'S AXIS
C     CLORIG      CELL ORIGIN COORDINATES
C     CLSPE       CELL SPECIFICATIONS
C     COOR        COORDINATES(R*8)
C     DBRAD       RADIUS DOUBLE PRECISION
C     ELSPE       ELLIPSE SPECIFICATIONS
C     FAC         FACILITY NODE NUMBER
C     FILE        DESIGN FILE NAME
C     GG          GRAPHIC GROUP
C     IAD         ARRAY CONTAINING ADDRESSES FOR LINES
C     INFILE      DATA FILE NAME
C     IRC         ERROR CODE
C     K1          NUMBER OF CLASSES FOR CIRCLES
C     K2          NUMBER OF CLASSES FOR LINES
C     LIFILE      CELL LIBRARY NAME
C     LILVS       LEVELS USED FOR THE LINES
C     LINSTR      COORDINATES(I*4)
C     LNSPE       LINE SPECIFICATIONS
C     LOCA        TITLE LOCATION
C     LOGA        LOGARITHM
C     LV          LEVEL
C     N           NUMBER OF ZONES
C     NC          NUMBER OF CHARACTERS
C     NOFAC       NUMBER OF FACILITIES
C     NORAD       ARRAY OF RADII
C     NPAR        TEXT NODE PARAMETERS
C     NPER        NUMBER OF PERCENTAGES
C     ORIG        ARRAY OF COORDINATES(I*4)
C     PEMID       MIDPOINT OF ONE CLASS INTERVAL
C     PMID        ARRAY OF MIDPOINTS
C     PER         ARRAY OF PERCENTAGES
C     PERL        ARRAY OF LARGEST VALUES
C     PERS        ARRAY OF SMALLEST VALUES
C     PTS         ARRAY OF COORDINATES(R*8)
C     PUFA        POSITIONAL UNIT FACTOR
C     PUSU        NUMBER OF PU'S BY SU'S
C     RAD         RADIUS
C     REGNA       REGION NAME
C     SUMU        NUMBER OR SU'S BY MU'S
C     TEXT        ZONE NUMBER OR NAME
C     TITLE       MAP TITLE
C     TNSPE       TEXT NODE SPECIFICATIONS
C     TNXY        TEXT NODE LOCATION
C     TPAR        TEXT NODE PARAMETERS
C     TXH         TEXT NODE HEIGHT
C     TXPAR       TITLE PARAMETERS
C     TXSPE       TITLE SPECIFICATIONS
C     TXTH        TITLE HEIGHT
C     UNVA        UNIT VALUE
```

```
C      VCN             VIEW CODE NUMBER
C      WIT             LINE WIDTH
C      ZL              ARRAY OF LARGEST VALUES
C      ZMID            MIDPOINT OF ONE CLASS INTERVAL
C      ZONE            ARRAY OF ZONE NUMBERS
C      ZS              ARRAY OF SMALLEST VALUES
C      ZVAL            ARRAY OF WEIGHTS
C
C      ................................................
C
C      SUBROUTINE SYMBOL
C
C      LANGUAGE      : FORTRAN
C      SYSTEM        : PDP11/70 RSX-11M-PLUS
C      AUTHOR        : LISE ALLARD
C      UIC           : [266,10]
C      LAST UPDATED  : JULY 1984
C
C      COMPILE       : >F77 ILASYMBOL.OBJ,ILASYMBOL.LST/-SP=ILASYMBOL.FTN/TR:NONE
C
C      PURPOSE       : CREATE A SYMBOL MAP.
C                      READ THE DATA, PLACE TITLE, CELLS
C                      AND TEXT NODES.
C
C      DFPI SUBROUTINES USED :
C      ASC2RD          CONVERTS A STRING OF ASCII CHARACTERS
C                      INTO AN EQUIVALENT STRING OF PACKED
C                      RAD50 CHARACTERS.
C      CLDFPI          PLACE CELLS
C      CONVER          CONVERTS FROM FLOATING POINT DOUBLE-
C                      PRECISION NUMBER WITH RANGE OF 0 TO
C                      +4294967296 TO I*4 FORTRAN FORMAT
C                      WITH RANGE OF -2147483648 TO +2147483647.
C      DEDFPI          DETACH FROM DFPI
C      ERROR9          RETURNS ERROR MESSAGES TO LUNG FOR DFPI
C                      INTERFACE ERROR.
C      INDFPI          INITIALIZE DFPI AND SET UP DESIGN FILE.
C      LKTCSI          FILE NAME CONVERSION
C      TNDFPI          PLACE TEXT NODES
C      TXDFPI          PLACE TEXT
C
C      SUBROUTINES AND FUNCTIONS REQUIRED :
C      NONE
C
C      ................................................
C
       SUBROUTINE SYMBOL
C
       COMMON /IREQ/ IREQ
       COMMON /INFLAG/ INFLAG
       COMMON /WAITFL/ WAITFL
```

```
      COMMON /APREGN/ APREGN
      COMMON /DFPI/ DFPI
C
      REAL*8 PTS(200), AXES(2), AS(2), AA, TXH, PUFA, PUSU, SUMU
      REAL*8 LOCA(2), TXTH
C
      INTEGER*4 ORIG(200), CLORIG(2), TNXY(2), ITXH
      INTEGER*4 ILOCA(2), ITXTH
      INTEGER*2 INFILE(31), LIFILE(15)
      INTEGER*2 FILE(15), CLIB50(7), DGN50(7), TNSPE(7)
      INTEGER*2 TXPAR(7), TXSPE(7)
      INTEGER*2 TPAR(6), CLSPE(5), NPAR(5), ATLK(4)
      INTEGER*2 AC(3), REGNA(3), VCN(2), GG(2), COUNT, LV
      INTEGER*2 FAC(10), NO
C
      CHARACTER*1 TITLE(50)
      CHARACTER*20 FMT, TEXT
C
      EQUIVALENCE (ITXH,TPAR(1))
      EQUIVALENCE (ITXTH,TXPAR(1))
C
      DATA AS /0.0, 0.0/, ATLK /4*0/, GG /2*0/, VCN /2*0/
      DATA NPAR /5*0/, TPAR /4*0, 1, 20/
      DATA CLSPE /5*0/, TNSPE /7*0/
      DATA TXPAR /4*0, 1, 50, 0/, TXSPE /7*0/
      DATA REGNA /'DF', 'PI', 'RG'/
C
C     PROGRAM IDENTIFICATION
C     ----------------------
      WRITE (5,10)
   10 FORMAT (/////,10X, ' * PART 4 - CREATE A ', 'SYMBOL MAP *')
C
C     KEY-IN DESIGN FILE NAME
C     -----------------------
      WRITE (5,20)
   20 FORMAT (///, '$DESIGN FILE NAME = ')
      READ (5,30) NC, FILE
   30 FORMAT (Q, 15A2)
      CALL LKTCSI(FILE, DGN50, NC, 0, IRC)
      IF (IRC .NE. 0) STOP 'LKTCSI CONVERSION ERROR'
C
C     DETERMINE POSITIONAL UNIT FACTOR
C     --------------------------------
   40 CONTINUE
      WRITE (5,50)
   50 FORMAT (///, '$NUMBER OF PU''S PER SU = ')
      READ (5,*,ERR=40) PUSU
      WRITE (5,60)
   60 FORMAT (/, '$NUMBER OF SU''S PER MU = ')
      READ (5,*,ERR=40) SUMU
C
```

```
      PUEA = PUSU * SUMU
C
C     KEY IN CELL LIBRARY NAME
C
      WRITE (5,70)
   70 FORMAT (//, '$CELL LIBRARY NAME = ')
      READ (5,30) NC, LIFILE
      CALL LKICSI(LIFILE, CLIB50, NC, O, IRC)
      IF (IRC .NE. 0) STOP 'LKICSI CONVERSION ERROR'
C
C     REGION NAME CONVERSION
C
      CALL ASC2RD(REGNA, REGN50, 6, IRC)
      IF (IRC .NE. 0) STOP 'ASC2RD CONVERSION ERROR'
C
C     INITIALIZE FOR DEPI
C
      CALL INDEPI(REGN50, DGN50, 1, CLIB50, O, O, IRC, 'EX')
      IF (IRC .EQ. 0) GO TO 90
      WRITE (5,80) IRC
   80 FORMAT (' INDEPI ERROR IRC=', I3)
      GO TO 590
   90 WRITE (5,100)
  100 FORMAT (/, ' *** INITIALIZATION COMPLETED ***')
C
C     KEY IN DATA FILE NAME
C
      WRITE (5,110)
  110 FORMAT (//, '$DATA FILE NAME = ')
      READ (5,120) NC, INFILE
  120 FORMAT (Q, 31A1)
      IF (NC .EQ. 0) GO TO 630
      IF (NC .GT. 30) NC = 30
      INFILE(NC + 1) = 0
      OPEN (UNIT=2,NAME=INFILE,TYPE='OLD',READONLY,ERR=82)
      GO TO 150
C
  130 WRITE (5,140)
  140 FORMAT (' PROBLEM OPENING DATA FILE')
      GO TO 63
C
C     READ THE NUMBER OF NODES
C
  150 CONTINUE
      READ (2,*,ERR=600) ND
C
C     READ COORDINATE FORMAT
C
      READ (2,160) FMT
  160 FORMAT (20A)
C
```

```
C     READ COORDINATES
C
      COUNT = 1
      DO 170 I = 1, ND
        READ (2,FMT) PTS(COUNT), PTS(COUNT + 1)
        PTS(COUNT) = PTS(COUNT) * PUFA
        PTS(COUNT + 1) = PTS(COUNT + 1) * PUFA
        COUNT = COUNT + 2
  170 CONTINUE
C
      CALL CONVER(PTS, ORIG, COUNT - 1, IRC)
C
C     **** DEPT PLACE ELEMENTS SUBROUTINES ****
C
C     PLACE TITLE (TYPE 17)
C
      WRITE (5,180)
  180 FORMAT (//, '$DO YOU WISH TO WRITE A MAP TITLE?(Y/N) ')
      READ (5,300) RESP
      IF (RESP .EQ. 'N') GO TO 280
C
      WRITE (5,190)
  190 FORMAT (///, ' MAP TITLE (MAX. 50 CHAR.) = ')
      READ (5,200) TITLE
  200 FORMAT (80A1)
C
  210 CONTINUE
      WRITE (5,220)
  220 FORMAT (//, '$ENTER (X,Y) LOCATION = ')
      READ (5,*,ERR=210) (LOCA(I),I=1,2)
C
      LOCA(1) = LOCA(1) * PUFA
      LOCA(2) = LOCA(2) * PUFA
C
      CALL CONVER(LOCA, ILOCA, 2, IRC)
C
  230 CONTINUE
      WRITE (5,240)
  240 FORMAT (//, '$ACTIVE LEVEL = ')
      READ (5,*) LV
C
      WRITE (5,250)
  250 FORMAT (//, '$TEXT HEIGHT = ')
      READ (5,*) TXTH
      TXTH = TXTH * PUFA
      ITXTH = JIDNNT(TXTH)
      TXPAR(3) = TXPAR(1)
C
      CALL TXDEPI(GG, AA, LV, TXPAR, TXSPE, ILOCA, IRC, TITLE, ATLK)
      IF (IRC .NE. 0) THEN
        WRITE (5,260) IRC
```

```
 260   FORMAT (/, ' ERROR IN TXDEPI IRC = ', I3)
         CALL ERROR9
         STOP
       END IF
C
       WRITE (5,270)
 270   FORMAT (//, ' ** TITLE COMPLETED ** ')
C
C      PLACE CELL (TYPE 2)
C
 280   CONTINUE
       WRITE (5,290)
 290   FORMAT (///, '$PLACE SYMBOL AT FACILITY LOCATION?(Y/N) ')
       READ (5,300) RESP
 300   FORMAT (A1)
       IF (RESP .EQ. 'N') GO TO 450
C
 310   CONTINUE
       WRITE (5,320)
 320   FORMAT (/, '$NUMBER OF FACILITIES = ')
       READ (5,*) NOFAC
       WRITE (5,330)
 330   FORMAT (/, '$ENTER THE FACILITIES'' NODE NUMBERS = ')
       READ (5,*,ERR=310) (FAC(I),I=1,NOFAC)
C
       WRITE (5,340)
 340   FORMAT (/, '$SYMBOL NAME (6 CHARACTERS) = ')
       READ (5,350) AC
 350   FORMAT (3A2)
C
       WRITE (5,360)
 360   FORMAT (/, '$ACTIVE LEVEL (1-63) = ')
       READ (5,370) LV
 370   FORMAT (I2)
C
 380   CONTINUE
       WRITE (5,390)
 390   FORMAT (/, '$ACTIVE ANGLE = ')
       READ (5,*,ERR=380) AA
C
 400   CONTINUE
       WRITE (5,410)
 410   FORMAT (/, '$ACTIVE SCALE (X,Y) = ')
       READ (5,*,ERR=400) (AS(I),I=1,2)
C
       DO 420 I = 1, NOFAC
         CLORIG(1) = ORIG((2*FAC(I)) - 1)
         CLORIG(2) = ORIG(2*FAC(I))
         CALL CLDEPI(LV, GG, AA, AS, AC, CLORIG, VCN, IRC, ATLK, CLSPE)
         IF (IRC .NE. 0) GO TO 530
 420   CONTINUE
```

```
      WRITE (5,430)
  430 FORMAT (//,  '** SYMBOLS COMPLETED **')
C
      WRITE (5,440)
  440 FORMAT (/, '$DO YOU HAVE AN OTHER SET OF FACILITIES?(Y/N)')
      READ (5,300) RESP
      IF (RESP.EQ.'Y') THEN
        GO TO 110
      END IF
C
C     PLACE TEXT NODE (TYPE 7)
C
  450 CONTINUE
      WRITE (5,460)
  460 FORMAT (///, '$PLACE TEXT NODE?(Y/N) ')
      READ (5,300) RESP
      IF (RESP.EQ.'N') GO TO 620
C
      WRITE (5,470)
  470 FORMAT (/, '$ATTACH TEXT TO THE TEXT NODE?(Y/N) ')
      READ (5,300) RESP
C
      WRITE (5,360)
      READ (5,370) LV
C
      WRITE (5,390)
      READ (5,*) AA
C
  480 CONTINUE
      WRITE (5,490)
  490 FORMAT (/, '$TEXT NODE''S HEIGHT = ')
      READ (5,*,ERR=480) TXH
      TXH = TXH * PUFA
      ITXH = JIDNNT(TXH)
      TPAR(3) = TPAR(1)
C
      DO 510 I = 1, ND
        TNXY(1) = ORIG((2*I) - 1)
        TNXY(2) = ORIG(2*I)
        IF (RESP.EQ.'Y') THEN
          READ (2,500) TEXT
  500     FORMAT (A20)
        END IF
        CALL INDEPI(GG, AA, LV, NPAR, TPAR, INSPE, TNXY, IRC, TEXT,
     1       ATLK)
        IF (IRC.NE.0) GO TO 570
  510 CONTINUE
      WRITE (5,520)
  520 FORMAT (//, ' ** TEXT NODE COMPLETED **')
C
      GO TO 620
```

```
C
C       ERROR MESSAGES
C
    530 WRITE (5,540) IRC
    540 FORMAT (' ERROR IN CLDFPI IRC=', I3)
        GO TO 590
C
    550 WRITE (5,560) IRC
    560 FORMAT (' ERROR IN INDFPI IRC=', I3)
        GO TO 590
C
    570 WRITE (5,580) IRC
    580 FORMAT (' ERROR IN INDFPI IRC=', I3)
C
    590 CONTINUE
        CALL ERROR9(IRC)
C
    600 WRITE (5,610)
    610 FORMAT (/, ' * PROBLEM READING DATA FILE *')
        STOP
C
    620 CONTINUE
C
C       TERMINATE DFPI
C       --------------
        IARG = 1
        IF (IRC .NE. 0) IARG = 0
        CALL DEDFPI(IARG)
        GO TO 640
C
    630 STOP ' *** PROGRAM TERMINATED *** '
C
    640 CONTINUE
        CLOSE (UNIT=2,DISPOSE='SAVE')
        RETURN
        END
```

APPENDIX 2. ILAMAR User's Manual

Table of Contents

List of Figures

FIGURE

Introduction

ILAMAP (Interactive Location-Allocation Mapping Program) is especially oriented toward the creation of flow maps to represent solutions of location-allocation problems. The program can also be used in a broad range of applications in order to generate proportional circle maps and symbol maps.

The purpose of this manual is to provide the user with the information required to achieve the creation of a map using ILAMAP on the INTERGRAPH System. It is designed for the PDP 11/70 and the version 8.5 of the INTERGRAPH System. We recommend a basic knowledge of IGDS, although it is not mandatory, as well as familiarity with the Intergraph line width system.

The information is procedurally oriented. The manual contains the log and design file creation procedures, the digitizing process, the data input requirements and the operation sequence. Examples of data files and runs are presented to assist the user in achieving a successful application of ILAMAP.

1. AN INTRODUCTION TO THE INTERGRAPH SYSTEM

The Intergraph system is an automated drafting and mapping system. It is a sophisticated tool which supports 2-D and 3-D capabilities for interactive graphics operations. Conceived as a design digitizing system, the input features consist of a floating menu tablet used as a function selection device as well as for the indication of x, y and z coordinates on the screen, a keyboard used for command data entries, a digitizing table and a cursor.

The colour graphics Intergraph workstation has dual monitors, one in colour and the other monochromatic. Each monitor supports a full screen or four view quadrant display mode for a total of eight independent views.

Two types of files resident on disk are associated with a graphic session: the design file which stores the drawings and the cell file or symbol file. The cell file is merely a library of standard design elements generated by the user, stored and retrieved by name whenever a symbol is to be placed on a map.

The maximum design area available is a cube of 4 294 967 296 units of resolution. The design file is composed of 63 identically sized overlaid planes, commonly referred to as levels. One can think of these as 63 clear acetates which can be displayed in any combination on the graphics screen. To facilitate work with large files, the system provides the capability to associate reference files of background data with the active drawing. As many as three reference files may be associated at any time with the working design file.

2. LOG ON PROCEDURE

-Type HELLO IGGEOG
-Enter Password

3. DESIGN FILE CREATION

A. Create the design file

Once the log-in or sign-on procedure is completed the *Drawing Management* utilities must be initiated to create a new design file with given name dimension and size specifications.

- Type **UTI** (to choose the Utilities program)
- Type **CREATE** (to create a new file)
- Type the name of the new design file ______________ **.DGN**
- Type **2D**
- Enter the number of blocks
  * If you don't know then start at 100. This can be increased or decreased later.

B. Access the design file

- Type **GRA**
- Type the file name ______________ .DGN

* *Refer to Figure 1.*

C. Set up the design file parameters

At this stage the command menu becomes active. The following operations are strongly recommended:

- Set the level to 1. Key in **LV=1**
- Turn off the locks LV(level), AN(angle), SC(scale) (Use the menu, Press C)
- Turn on the lock SN(snap) (Press C)
- Turn off VIEW CONTROL DELAY (Press C)

The establishment of the design working parameters is the first step in any map creation. The selection of the DESIGN OPTIONS tutorial display from the command menu, allows direct access to the working units definition process. The definition of working units provides a means of relating the system internal measurement of units of resolution (UORs) to any desired measurable units that a specific map would require. The standard format for all readout data is given according to the defined units in master unit, subunit and positional unit.

```
>HELLO IGGEOG
PASSWORD

 RSX-11M-PLUS V01 BL6     MULTI-USER [14,5] SYSTEM

GOOD MORNING
13-AUG-84 11:00 LOGGED ON TERMINAL TT2:  AS IGS26

This computer was booted (started) from QS0    That means we are
running a QS0 [321,54] data base.

>@LOGIN CMD
 *** INTERGRAPH Command Environment ***.   2 1 0/PDP
>* GRAPHICS [S]  UTI
>* UTILITIES [S]  CREATE
>* CREATE -- new file name [S]  FILEA DGN
>* CREATE -- dimension (2D/3D) [S]  2D
>* CREATE -- size [S]  100
SUCCESSFUL COMPLETION
>* UTILITIES [S]  GRA
>* GRAPHICS [S]  _
```

FIGURE 1. Log-on create and access a design file

Define the working units:

1. Use the menu command DESIGN OPTIONS (Press C)

-Select on the screen DEFINE DESIGN WORKING UN... : Press D)
* *Note: The units can be any measurable units.*

For example, key in:

1	km	as your MU (master unit)
1000	m	as your SU (subunit)
100		as your PU (positional unit)

* By doing so the PU indicates the degree of accuracy for the drawing and is the determining factor for the design file space.

2. Select RETURN TO PRIOR LEVEL @ (Press D)

3. Select BEGIN / CONTINUE DESIGN @ (Press D)

4. PROCESS OF GETTING A BASE MAP

In order to get the base map of the study area into the design file, the user may choose between two methods: the digitizing process or the use of the BLDDGN program.

A. The digitizing process

The digitizing process involves seven steps to be executed while in graphics mode. The user needs a map of the study area on which a ... ars a graphic scale.

Step 1. Display the entire file on the screen.

1. Use the menu command VIEW SCALE FIT (Press C)
   - Select view on the right screen (Press D twice)
   - Will say Operation Complete.

2. Select VIEW CONTROL:COPY to set left screen (Press C)
   - Select view on the right screen (Press D)
   - Select view on the left screen (Press D)

3. Using the menu command MEASURE DISTANCE:CUM (cumulative distance), measure the length of the right screen (Press C)
   - Enter two data points (by pressing D) to see this distance on the screen.
   - Reset (Press R)

Step 2 Define the working area

Note: Your working area should be of sufficient size to include the map to be digitized. It should be located near the center of the design file in order to eliminate possible problems when working close to the edges of the design plane.

1. Use the menu command VIEW SCALE WINDOW AREA (Press C)
   - Define on the left screen your working area by entering two data points (Press D) - One on either diagonal
   - Select view on the right screen

2. Set the menu command VIEW CONTROL OVERVIEW ON / OFF (Press C)
   - Turn it on
   - This will indicate by a dashed outline on the left screen that area which is showing on the right screen

Step 3 Attach the map to your design file

1. Tape the map on the table at 0 degree. (North is pointing away from you.)

2. Use the menu commands

   a. DIGITIZE ORIGIN DEF (Press C)
   - Place cursor on the map on lower left corner (at your selected origin) and Press D
   - Identify a point on the lower left corner of the right screen (Press D)

   b. ROTATION ANG (Press C)
   - Press D over two points in a straight horizontal line on the map
   - Key in **0** (Press RETURN)

   c. FIXED SCALE (Press C)
   - Press D on the beginning point of the graphic scale on the map
   - Press D on the ending point of the graphic scale on the map
   - Key in the scale in MU SU PU (Not all are needed)

Step 4 Start the digitizing

1. Make sure you are on the right level
   - Key in **LV=$** the level will appear on the screen
   - To change the level, key in LV= the number you want (1-63)
   - Pick a colour **CO**= 0-7 (see matching colours on the board)

2. Digitize elements by selecting points
Note: Any elements from the menu can be digitized. (Lines, circles, shapes, line strings etc.)
   - Select the element needed on the menu (Press C)
   - Enter required data points on your map with the cursor (Press D)

Step 5. Terminate the digitizing session

When done, use the menu command FILE DESIGN. Press [illegible] [illegible] get back to the same picture [illegible] screens the next day.

[illegible] Figure [illegible]

Step 6. Record coordinates

The centroids of each zone or demand points of the study area (e.g. census tracts) must be recorded in the geographic base file.

- Press the [illegible] button on the cursor over each zone of the map at the approximate centroid or wherever you wish symbols to be placed and systematically note the x, y position appearing on the bottom right corner of the right screen.
- Example:
  - on the screen [illegible]
  - becomes [illegible]

Step 7. Exit the design file

- Press **CTRL Z** and then the **RETURN** key.
- Press **CTRL Z** to terminate the graphics session.

B. Use of the BLDDGN program

The BLDDGN program (author: Wade Johnson) can be used to place line strings in a design file from coordinate strings in an input data file. It requires a base map in digital form. The destination design file must be predefined as 2-D, contiguous and of sufficient size to contain all the elements to be placed. Single point placements are considered illegal. Only lines and lines strings are placed.

The data file must contain a unique I.D. for each line string, followed by the coordinate pairs pertaining to the same feature:

Example:

1 x y
1 x y
2 x y
2 x y

FIGURE 2. The base map

The required format is 'EA2,2F20.5'

To run enter the command >RUN BLDDGN

Then enter the required information when prompted for by the program.

5. DATA FILE INPUT REQUIREMENTS

The data required consist of

1. The centroids of each zone of the study area as recorded in the geographic base file and corresponding to the demand points.

2. The weight attached to these demand points according to the initial data used by the location-allocation model.

3. The model's solution in percentage of flows of patrons from each zone attending each facility.

The data files format required is explained in the description of the four major parts of ILAMAP (refer to section 6).

Create an empty file >EDT filename.DAT

Insert the data *INSERT

Terminate CTRL Z

6. EXECUTE ILAMAP

ILAMAP has been designed to perform a number of operations which could adequately map location-allocation solutions. This expresses the need to portray the initial data, such as the number of patrons in each zone, the facilities' location and attributes, and the attendance patterns. Since we are dealing with absolute values gathered for areas, proportional circles have been selected to map the initial data. Proportional flow lines are used to illustrate the attendance patterns in terms of the percentage of patrons from each demand zone attending each facility. The facilities' location and attributes are represented by various point symbols stored in a cell library.

The program is presented in four major modules or parts. The program's options allow the creation of up to four maps in any combination in th[illegible]me run. The four maps can be stored in a single design file using the multi-level capability for display or stored in four individual design files using the reference files system for simultaneous display.

- To run the program enter the command : >RUN ILAMAP
- Then enter the required information when prompted for by the program.
- Every distance or measure to be keyed in must be expressed in master units.

PART 1. Create a map using Jenks data classification.

This part makes use of Jenks optimization program for data classification and allows you to interactively create the map. However an important restriction must be carefully noted: no more than 50 values can be processed. This means that the data file should contain less or be equal to 50 zones and less or equal to 50 flow lines. The data file must be formatted as illustrated in Figure 3. Figure 4 shows an example of run. Figure 5 presents the map generated.

PART 2. Create a map using your own classification.

When more than 50 zones or 50 flow lines have to be processed and a map with discrete classes generated, part two should be used. The class limits must be predefined by a method of your choice. The upper and lower limits of each class must be entered during execution.

The data file must include the FORMAT statements (for the data) as illustrated in Figure 6. Figure 7 shows an example of run, and Figure 8 presents the map generated.

PART 3. Create a non-classed map.

The non-classed or unquantized map should be created when no grouping of data is desired. Relative circle sizes are assigned to each individual value and relative line width assigned to each individual percentage. However the range of line thickness available on the Intergraph falls between 0 and 31. As a consequence, the proportionality is normalised within that range. Thirty-two levels are used in the design file in order to group all the lines with a same thickness on the same level.

```
 30
N001 12835 25906 11528 49562
N002 12832 26167 11525 36270
N003 12832 69434 11529 12899
N004 12829 55099 11529 11335
N005 12836 61180 11530 13246
:
:
N025 12834 82379 11522 22717
N026 12835 44412 11517 11075
N027 12831 10963 11512 59902
N028 12831 18782 11515 78668
N029 12833 96367 11512 73195
N030 12836 60398 11512 59381
   937 700
   879 200
   838 100
  1638 400
     7 700
:
:
   634 500
     2 100
  1314 200
   922 800
    35 200
 39
 99 98     1     4
 99 14     2     4
100 00     3     4
100 00     4     4
100 00     5     4
100 00     6     4
 99 86     7    13
:
:
100 00    26    24
 93 07    27    22
  6 91    27    24
 99 94    28    24
 93 07    29    22
  6 90    29    24
 99 99    30    24
>
```

— number of zones *(I3)*

zone identification and (x,y) location *(AX, 2F12.5)*

patrons in each zone *(F10.3)*

— number of percentage flows *(I3)*

percentage, zone number, facility number *(F6.2, 2I5)*

FIGURE 3. Example of data file for PART 1.

```
>RUN ILAMAP

*** ILAMAP  INTERACTIVE LOCATION-ALLOCATION MAPPING PROGRAM ***

        by Lise A. [illegible]
        UNIVERSITY OF ALBERTA

        This program is set up in 4 parts
        PART 1 - Create a map using Jenks data classification
        PART 2 - Create a map with your own class intervals
        PART 3 - Create a non-classed map
        PART 4 - Create a symbol map

ENTER OPTIONS :
    0 = NO
    1 = YES

USE OF JENKS' CLASSIFICATION = 1
SET YOUR OWN CLASS INTERVALS = 0
CREATE A NON-CLASSED MAP = 0
CREATE A SYMBOL MAP = 0


        * PART 1-A   JENKS' DATA CLASSIFICATION *

DATA FILE NAME= DIAJE DAT

CALCULATE CLASS INTERVALS FOR CIRCLES?(Y/N) Y
NUMBER OF CLASSES (1-10) = 4
JENKS' REPORT FILE NAME= J1 DAT

A  4 CLASS MAP
        2 100     245 300
      493 200     838 100
      879 200    1232 600
     1314 200    1638 400

MODIFY THE NUMBER OF CLASSES OR CONTINUE?(M/C) M
```

Continued...

... Continued

```
NUMBER OF CLASSES (1-10) = 5
JENKS' REPORT FILE NAME= J1.DAT

A 5 CLASS MAP
       2.100     245.300
     493.200     729.900
     798.600    1041.900
    1115.100    1314.200
    1486.200    1638.400

MODIFY THE NUMBER OF CLASSES OR CONTINUE?(M/C) C


CALCULATE CLASS INTERVALS FOR FLOW LINES?(Y/N) Y
NUMBER OF CLASSES (1-10) = 5
JENKS' REPORT FILE NAME= J2.DAT

A 5 CLASS MAP
       1.330      10.880
      37.660      45.010
      49.980      62.100
      88.830      93.070
      96.940     100.000

MODIFY THE NUMBER OF CLASSES OR CONTINUE?(M/C) C


        * PART 1-B   CREATE A MAP WITH OPTIMAL CLASS INTERVALS *

DESIGN FILE NAME=  DIAP1.DGN

NUMBER OF PU'S PER SU= 100
NUMBER OF SU'S PER MU= 1000
*** INITIALIZATION  COMPLETED ***

DATA FILE NAME= DIAJE.DAT
```

Continued ...

... Continued

```
PLACE PROPORTIONAL CIRCLES?(Y/N) Y
ACTIVE LEVEL (1-63) = 10
RADIUS FOR LARGEST CIRCLE = 0.95

RADIUS ASSOCIATED TO EACH CLASS
          0.222816
          0.555662
          0.701925
          0.822555
          0.950000

MODIFY THE RADIUS OR CONTINUE?(M/C) C

** PROPORTIONAL CIRCLES COMPLETED **

PLACE PROPORTIONAL FLOW LINES?(Y/N) Y
ACTIVE LEVEL (1-63) = 11

LEVELS TO BE USED FOR A  5 CLASS MAP =  11 TO  15
MODIFY OR CONTINUE?(M/C) C
THICKNESS FOR BIGGEST LINE (0-31) = 20

LINE THICKNESS ASSOCIATED TO EACH CLASS
          1
          8
         11
         18
         20

MODIFY THE LINE THICKNESS OR CONTINUE?(M/C) C

** PROPORTIONAL LINES COMPLETED **
TT2  --  STOP  * PROGRAM COMPLETED *_

>_
```

FIGURE 4. PART 1 : Example of a run.

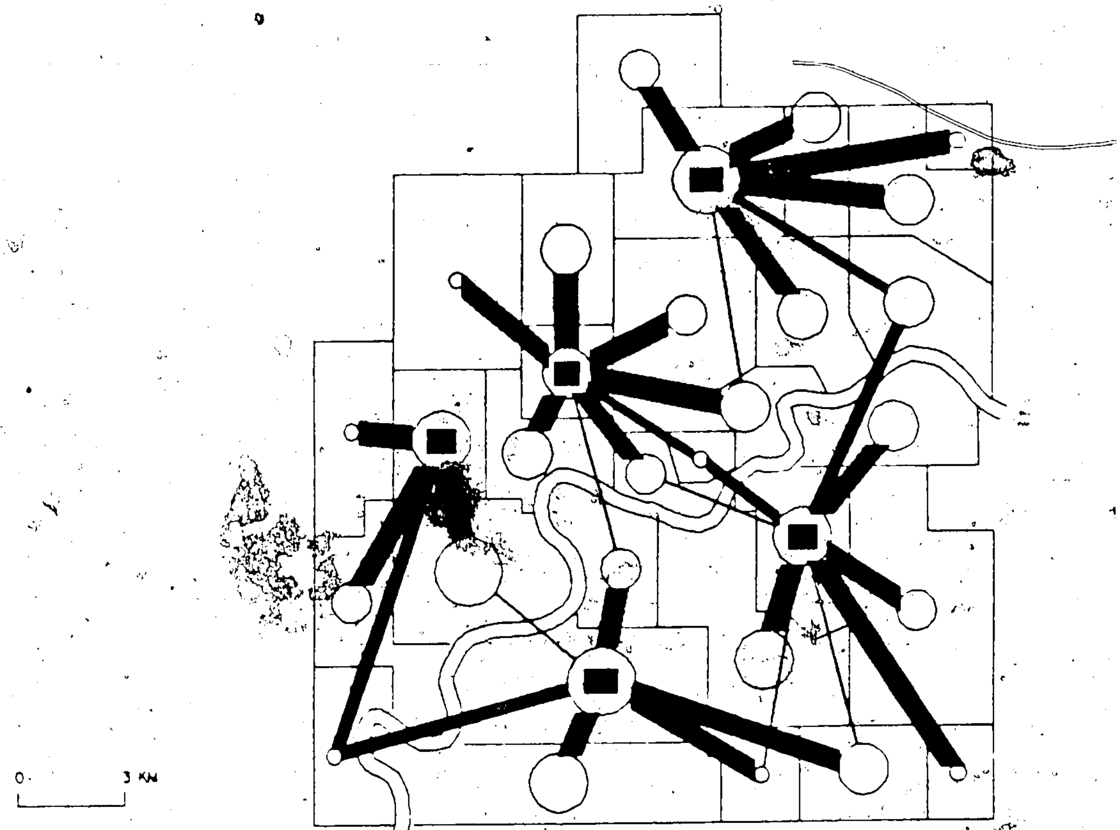


FIGURE 5. Example of a map produced using PART 1.

```
 30                                         — number of zones (FREE FORMAT)
(4X 2F12 5)                                 — format statement for the coordinates
N001 12835 25906 11528 49562
N002 12832 26167 11525 36270
N003 12632 69434 11529 12899
N004 12629 55099 11529 11335
N005 12836 61180 11530 13246
:
:                                             zone identification and (x y) location
:                                                (format as specified above)
:
N025 12834 62379 11522 22717
N026 12835 44412 11517 11075
N027 12831 10963 11512 59902
N028 12831 18752 11515 78666
N029 12833 36367 11512 73195
N030 12836 60396 11512 59381
(F10 3)                                     — format statement for the patrons
   937 700
   879 200
   838 100
  1636 400
     7 700
:
:                                             patrons in each zone
:                                             (format as specified above)
:
   634 500
     2 100
  1314 200
   922 800
    35 200
 73                                         — number of percentage flows (FREE FORMAT)
(F6 2 2I5)                                  — format for the percentages
 99 98     1     4
  0 02     1    24
 99 14     2     4
  0 74     2    13
  0 12     2    24
100 00     3     4
:
:                                             percentage zone number
:                                                facility number
:                                             (format as specified above)
:
 99 94    28    24
  0 02    29    16
 93 07    29    22
  6 90    29    24
  0 01    30    22
 99 99    30    24
>_
```

FIGURE 6. Example of data file for PART 2 and PART 3.

```
>RUN ILAMAP

*** ILAMAP   INTERACTIVE LOCATION-ALLOCATION MAPPING PROGRAM ***

         by Lise Allard
         UNIVERSITY OF ALBERTA

         This program is set up in 4 parts
         PART 1 - Create a map using Jenks' data classification
         PART 2 - Create a map with your own class intervals
         PART 3 - Create a non-classed map
         PART 4 - Create a symbol map

ENTER OPTIONS
     0 = NO
     1 = YES

USE OF JENKS' CLASSIFICATION = 0
SET YOUR OWN CLASS INTERVALS = 1
CREATE A NON-CLASSED MAP = 0
CREATE A SYMBOL MAP = 0

        * PART 2 , CREATE A MAP  WITH YOUR OWN CLASS INTERVALS *

DESIGN FILE NAME=  EXA DGN

NUMBER OF PU'S PER SU= 100
NUMBER OF SU'S PER MU= 1000
***.INITIALIZATION  COMPLETED ***

DATA FILE NAME= LAMB10 DAT

PLACE PROPORTIONAL CIRCLES?(Y/N) Y

NUMBER OF CLASSES (1-10) = 5
```

Continued ...

...Continued

```
ENTER LOWER LIMITS
2 1 493 2 798 6 1115 1 1486 2

ENTER UPPER LIMITS
245 3 729 9 1041 9 1314 2 1638 4

ACTIVE LEVEL (1-63) = 10

RADIUS FOR LARGEST CIRCLE = 0 95

RADIUS ASSOCIATED TO EACH CLASS
          0 222816
          0 555662
          0 701925
          0 822655
          0 950000

MODIFY THE RADIUS OR CONTINUE? (M/C) C

** PROPORTIONAL CIRCLES COMPLETED **

PLACE PROPORTIONAL FLOW LINES? (Y/N) Y

NUMBER OF CLASSES (1-10) = 7

ENTER LOWER LIMITS
0 01 1 00 5 00 10 00 25 00 50 00 75 00

ENTER UPPER LIMITS
0 99 4 99 9 99 24 99 49 99 74 99 100 00

ACTIVE LEVEL (1-63) = 11

LEVELS TO BE USED FOR A 7 CLASS MAP = 11 TO 17
MODIFY OR CONTINUE? (M/C) C
THICKNESS FOR BIGGEST LINE (0-31) = 20

LINE THICKNESS ASSOCIATED TO EACH CLASS
          0
          1
          2
          4
          9
          14
          20

MODIFY THE LINE THICKNESS OR CONTINUE? (M/C) C

** PROPORTIONAL LINES COMPLETED **
TT2 -- STOP * PROGRAM COMPLETED *
>_
```

FIGURE 7. PART 2 . Example of a run.

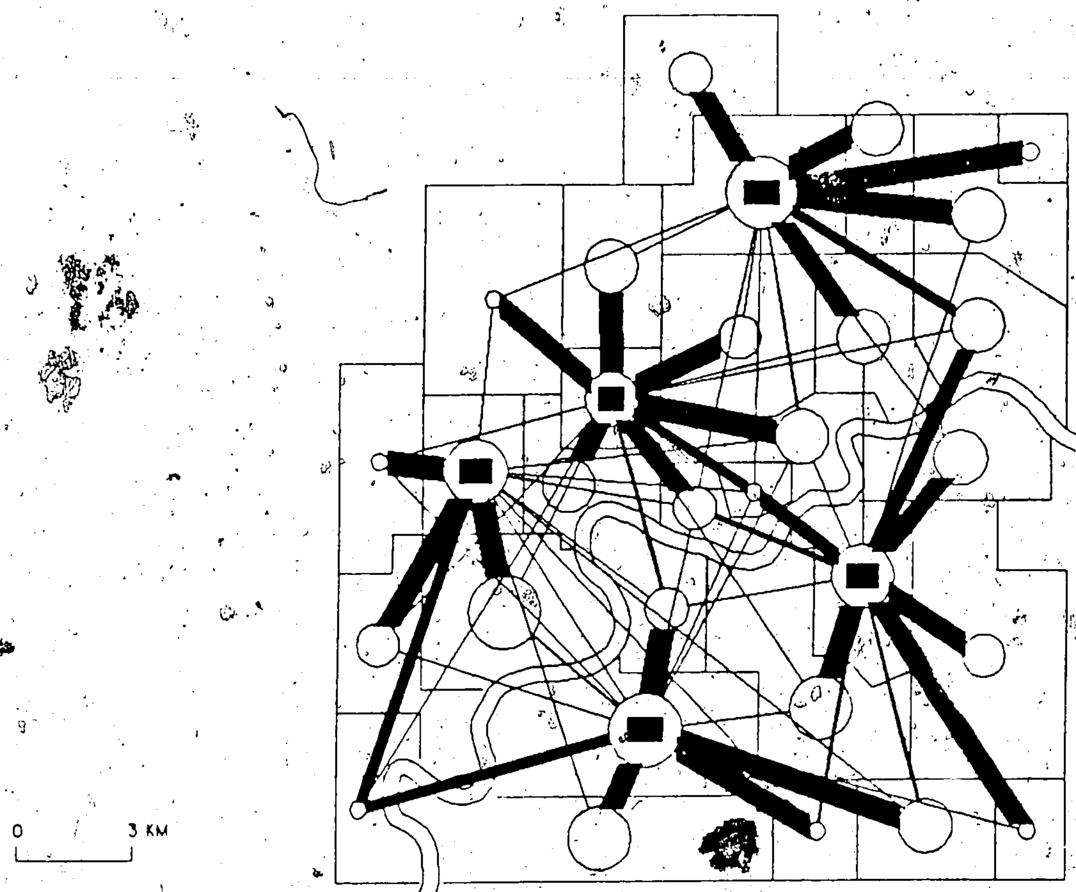


FIGURE 8. Example of a map produced using PART 2.

The data file required is the same as the one for PART 2 (refer to Figure 6). Figure 9 shows an example of a run and Figure 10 presents the map generated.

PART 4. Create a symbol map.

The last module generates a symbol map which can be used to add supplementary information on the final map. A title can be placed the facilities location and attributes symbolized and numbers or names associated to each demand point. The symbols are retrieved from a cell library. The cell library named *SYMBOL.CEL* contains a good selection of symbols that just need to be scaled to meet different scale maps. Any special symbol can be created and attached to the cell library for further use by the program.

The data file required includes only the demand zones locations and their names or numbers as illustrated in Figure 11. Figure 12 shows an example of a run, and Figure 13 presents the map generated.

7. RESULTS EXAMINATION

A. List the active tasks

Prior to accessing the design file for examination of the results it is recommended to list, at the terminal, the active tasks and wait until the *.A/DMP* and *DFP/* tasks are both terminated.

The command is. >ACT / ALL

If the tasks are terminated, they are not listed, as shown in Figure 14.

B. In the design file

Once in the design file, the base map should be displayed on the screen. Then, every level used by the program must be turned on in order to appear on the screen.

- Key-in **LV=** (1-63)
- Select view (Press D)

```
>RUN ILAMAP

*** ILAMAP   INTERACTIVE LOCATION-ALLOCATION MAPPING PROGRAM ***

          by Lise Allard
          UNIVERSITY OF ALBERTA

          This program is set up in 4 parts

          PART 1 - Create a map using Jenks' data classification

          PART 2 - Create a map with your own class intervals

          PART 3 - Create a non-classed map

          PART 4 - Create a symbol map

ENTER OPTIONS
     0 = NO
     1 = YES

USE OF JENKS' CLASSIFICATION = 0

SET YOUR OWN CLASS INTERVALS = 0

CREATE A NON-CLASSED MAP = 1

CREATE A SYMBOL MAP = 0


           * PART 3   CREATE A NON-CLASSED MAP *

DESIGN FILE NAME=  EXAMPLE DGN

NUMBER OF PU'S PER SU= 100
NUMBER OF SU'S PER MU= 1000
*** INITIALIZATION  COMPLETED ***
```

Continued ...

...Continued

```
DATA FILE NAME= LAMB10.DAT

PLACE PROPORTIONAL CIRCLES?(Y/N) Y
ACTIVE LEVEL (1-63) = 20
RADIUS FOR SMALLEST CIRCLE= 0 03
** PROPORTIONAL CIRCLES COMPLETED **

PLACE PROPORTIONAL FLOW LINES?(Y/N) Y
ACTIVE LEVEL (1-63) = 30
** PROPORTIONAL LINES COMPLETED **
TT2  --  STOP  * PROGRAM COMPLETED *
>_
```

FIGURE 9. PART 3 Example of a run.

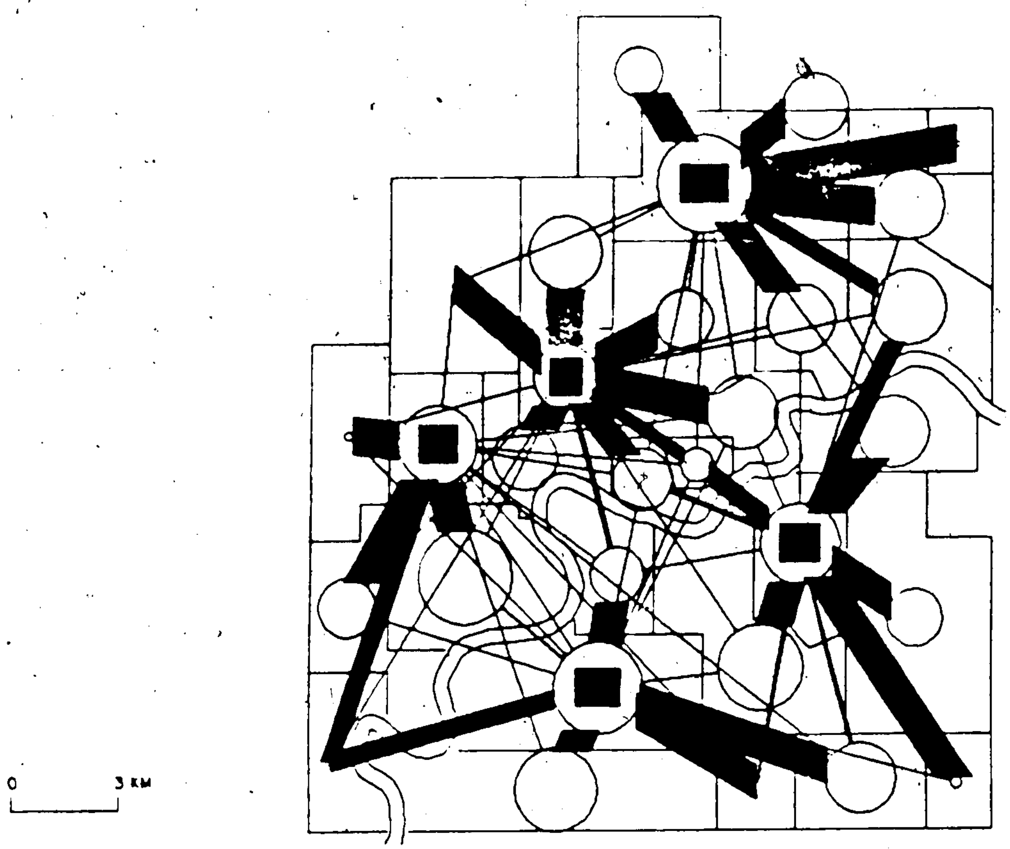


FIGURE 10. Example of a map produced using PART 3.

```
 30                                      — number of zones (FREE FORMAT)
(4X, 2F12 5)                             — format statement for the coordinates
N001 12835 25906 11528 49562  ┐
N002 12832 26167 11525 36270  │
N003 12832 69434 11529 12899  │
N004 12829 55099 11529 11336  │
N005 12836 61180 11530 13246  │
N006 12827 69000 11532 18893  │
N007 12828 98539 11525 36530  │
N008 12835 26167 11525 65201  │
N009 12830 64569 11522 79015  │
N010 12829 37896 11521 36923  │
N011 12827 84117 11520 96044  │
N012 12825 61267 11527 21848  │
N013 12825 67523 11523 78581  │
N014 12822 53448 11526 37660  │
N015 12819 62050 11522 19328  │  zone identification and (x,y) location
N016 12622 11845 11521 93525  │       (format as specified above)
N017 12824 53882 11521 56513  │
N018 12822 88635 11518 29667  │
N019 12819 61528 11517 47043  │
N020 12819 15394 11513 21935  │
N021 12827 14525 11518 34619  │
N022 12826 63179 11515 23673  │
N023 12825 44326 11512 44263  │
N024 12832 26167 11519 18025  │
N025 12834 82379 11522 22717  │
N026 12835 44412 11517 11075  │
N027 12631 10963 11512 59902  │
N028 12831 18782 11515 78668  │
N029 12833 96367 11512 73195  │
N030 12636 60398 11512 59381  ┘
ZONE 01  ┐
ZONE 02  │
ZONE 03  │
ZONE 04  │
ZONE 05  │
ZONE 06  │
ZONE 07  │
ZONE 08  │
ZONE 09  │
ZONE 10  │
ZONE 11  │
ZONE 12  │
ZONE 13  │
ZONE 14  │  number or name associated to each zone
ZONE 15  │                 (20A1)
ZONE 16  │
ZONE 17  │
ZONE 18  │
ZONE 19  │
ZONE 20  │
ZONE 21  │
ZONE 22  │
ZONE 23  │
ZONE 24  │
ZONE 25  │
ZONE 26  │
ZONE 27  │
ZONE 28  │
ZONE 29  │
ZONE 30  ┘
>_
```

FIGURE 11. Example of data file for PART 4.

```
>RUN ILAMAP

*** ILAMAP   INTERACTIVE LOCATION-ALLOCATION MAPPING PROGRAM ***

          by Lise Allard
          UNIVERSITY OF ALBERTA

          This program is set up in 4 parts

          PART 1 - Create a map using Jenks' data classification

          PART 2 - Create a map with your own class intervals

          PART 3 - Create a non-classed map

          PART 4 - Create a symbol map

ENTER OPTIONS
      0 = NO
      1 = YES

USE OF JENKS' CLASSIFICATION = 0

SET YOUR OWN CLASS INTERVALS = 0

CREATE A NON-CLASSED MAP = 0

CREATE A SYMBOL MAP = 1


          * PART 4 - CREATE A SYMBOL MAP *


DESIGN FILE NAME = EXFILE DGN


NUMBER OF PU'S PER SU = 100

NUMBER OF SU'S PER MU = 1000

CELL LIBRARY NAME = LASYMBOL CEL

*** INITIALIZATION COMPLETED ***

DATA FILE NAME =TEXT DAT
```

Continued ...

...Continued

```
DO YOU WISH TO WRITE A MAP TITLE?(Y/N) Y

MAP TITLE (MAX  50 CHAR ) =
ATTENDANCE PATTERNS IN EDMONTON

ENTER (X.Y) LOCATION = 12813 95349 11532 99763

ACTIVE LEVEL = 18

TEXT HEIGHT = 0 4

** TITLE COMPLETED **

PLACE SYMBOL AT FACILITY LOCATION?(Y/N) Y
NUMBER OF FACILITIES = 5
ENTER THE FACILITIES' NODE NUMBERS = 4 13 16 22 24
SYMBOL NAME (6 CHARACTERS) = TRIA
ACTIVE LEVEL (1-63) = 18
ACTIVE ANGLE = 0 0
ACTIVE SCALE (X.Y) = 0 001 0.001

** SYMBOLS COMPLETED **
DO YOU HAVE AN OTHER SET OF FACILITIES?(Y/N)N

PLACE TEXT NODE?(Y/N) Y
ATTACH TEXT TO THE TEXT NODE?(Y/N) Y
ACTIVE LEVEL (1-63) = 19
ACTIVE ANGLE = 0 0
TEXT NODE S HEIGHT = 0 28

** TEXT NODE COMPLETED **
TT2  --  STOP  * PROGRAM COMPLETED *
>_
```

FIGURE 12. PART 4 Example of a run.

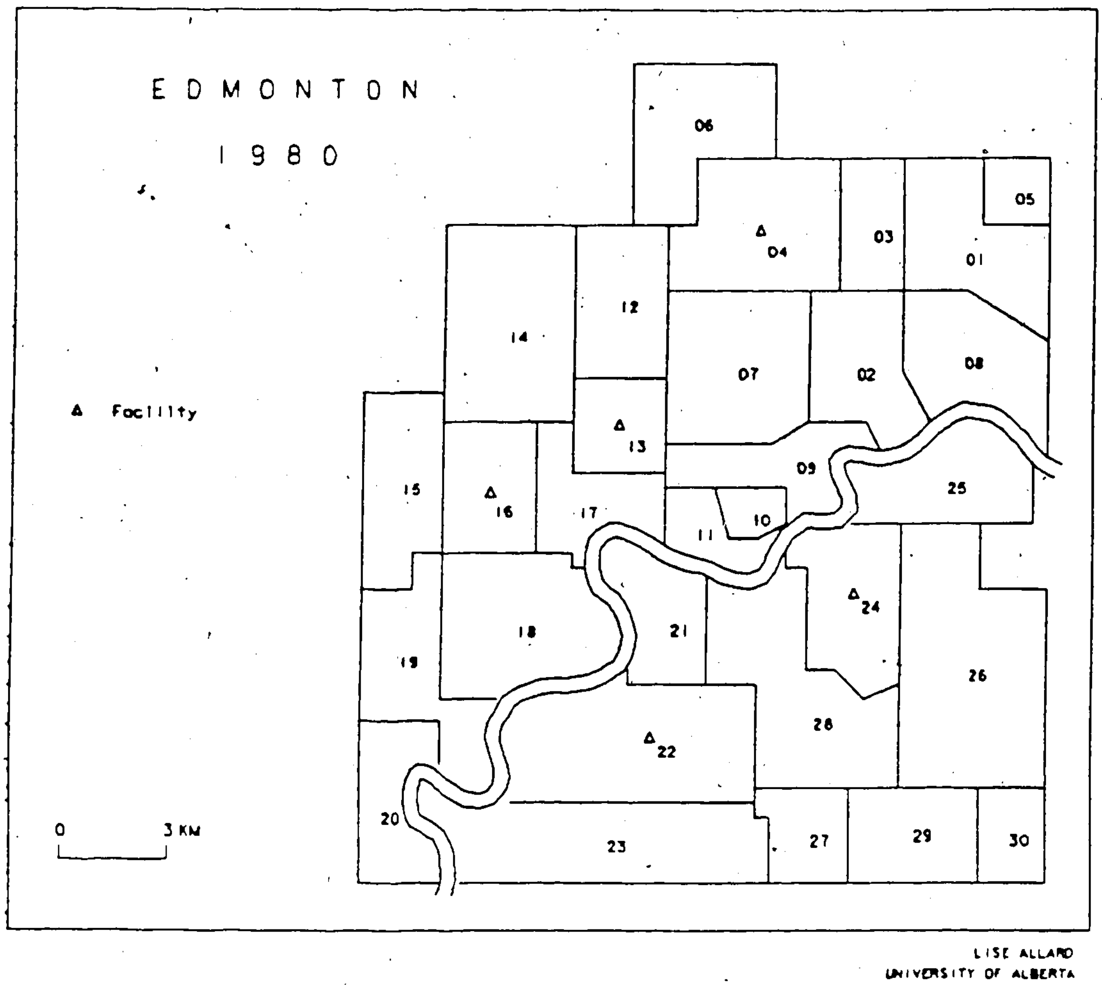


FIGURE 13. Example of a map produced using PART 4.

```
>ACT /ALL
   LDR (CD0 )
EGNOTE  (TT0 )
ACTT2   (TT2 )
MCR     (TT2 )
F11ACP  (CD0 )
DC      (TT0 )
 AIDMP  (TT2 )  ←
HRC     (CD0 )
DFPI    (TT2 )  ←
SYSLOG  (CD0 )
BAP0    (CD0 )
SYSMON  (TT0 )
QMG     (CD0 )
ALARMS  (TT0 )
LPP0    (CD0 )
CONLOG  (TT0 )
CN1T16  (TT16 )
DSPR02  (TT0 )
>_
```

```
>ACT /ALL
   LDR (CD0 )
EGNOTE  (TT0 )
ACTT2   (TT2 )
MCR     (TT2 )
F11ACP  (CD0 )
DC      (TT0 )
HRC     (CD0 )
SYSLOG  (CD0 )
BAP0    (CD0 )
SYSMON  (TT0 )
QMG     (CD0 )
ALARMS  (TT0 )
LPR0    (CD0 )
CONLOG  (TT0 )
CN1T16  (TT16 )
DSPR02  (TT0 )
>_
```

FIGURE 14. Listing of the active tasks.

Because the graphic information is stored on different levels, it is possible to display selectively the information and consequently manipulate it graphically in order to create a more suitable map for a specific purpose.

8. MAP FINALISATION

The final step involves the creation of a legend. This can be achieved interactively on the screen by using menu commands. It is also possible at this point, to make any essential correction to the map. Colours can be changed, elements can be moved or deleted, the map can be scaled and so on. Every graphic element becomes a single element which can be manipulated independently. Figure 15 illustrates an example of a final map.

If more information is required about how using the menu commands, please refer to the *Intergraph Operator Manual*.

9. LOG OFF PROCEDURE

- In the design file, press **CTRL Z** and then the **RETURN** key
- Press **CTRL Z** to terminate the graphics session
- Key in **BYE**

* *Do not forget to power off the workstation and the modem.*

10. PROGRAM LIMITATIONS

The most important limitation of ILAMAP is the size of the cartographic problem which can be looked at. In fact, to use ILAMAP PART 1, the number of zones in the study area, as well as the number of percentage flows of the attendance pattern, must be less or equal to 50. For the other parts, a more reasonable problem size can be handled. The maximum number of zones or demand points can then reach 100 and the maximum number of percentage flows 150. The number of facilities is not critical. Up to 10 facilities can be processed at once. The symbols are placed within a loop which allows the possibility of placing different sets of symbols by looping as many times as desired.

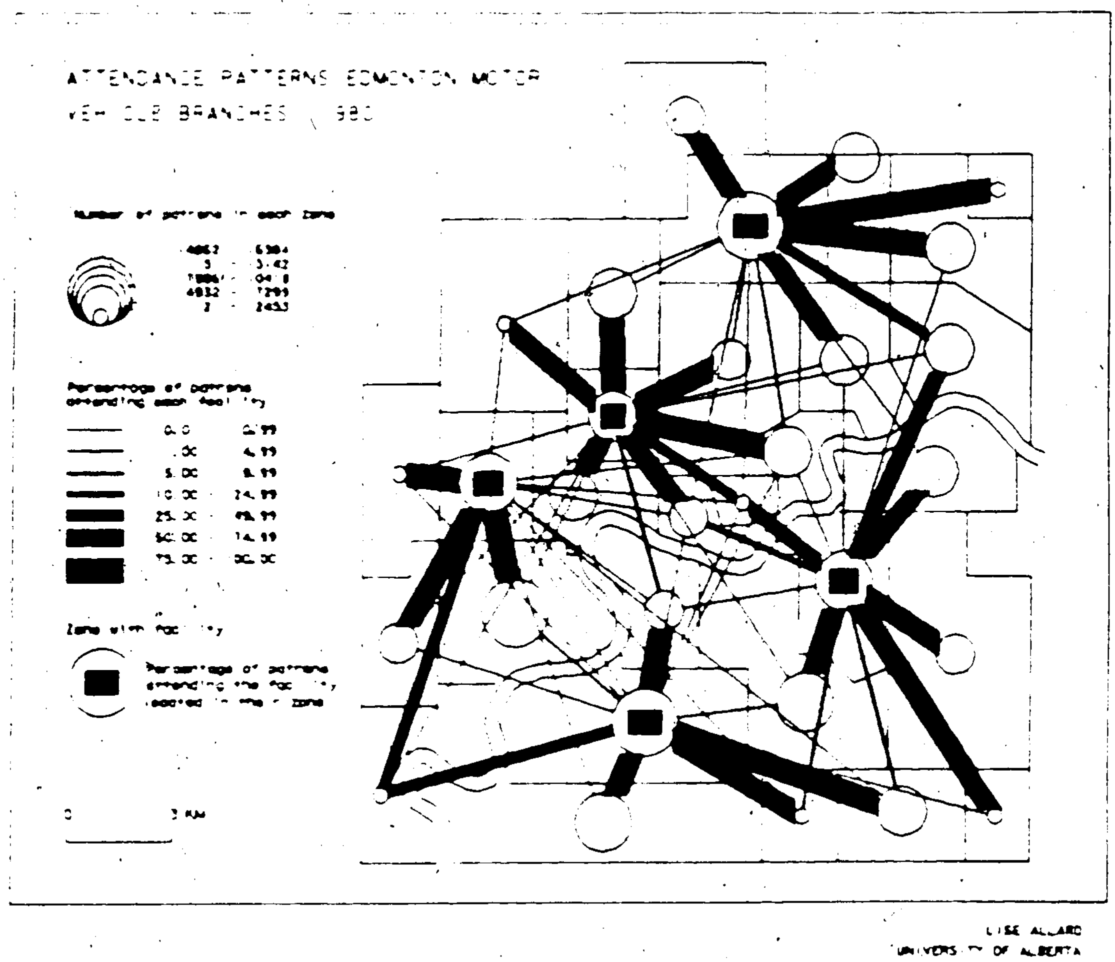


FIGURE 15. Example of a final map

11. RECOMMENDATIONS AND COMMON ERRORS

Recommendations

- Save enough free blocks in the design file to contain the elements to be placed by the program.

- Keep track of the levels being used in order to avoid information overlaps.

- The base map should be placed on a separate level and should not interfere with the rest of information.

- Take note of the radius assigned to each class of circles as well as the line width assigned to each class of flow lines in order to reproduce them for the legend.

- Familiarity with the Intergraph line width system is recommended. It ranges from 0 to 31.

- If a symbol map is created, it is essential to make a few trials in the design file to help decide on the proper scale to be used for each symbol.

Common errors

- *GRAPHICS:* filename.DGN
*** DESIGN FILE IN USE ***

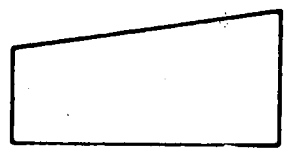
- List the tasks and make sure that *.AIDMP* and *DFPI* are both terminated before accessing the design file.

- *CONVERSION ROUTINE MISSING*

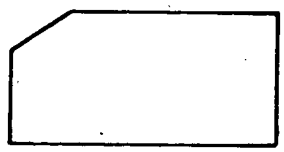
- If the design file is too small to contain the elements to be placed by the program, the IGDS system erroneously reports the condition as *"conversion routine missing"*. Should this error occur during the execution, the user should delete those elements that were successfully placed, increase the allocated design file space and rerun the program.

APPENDIX 3. Flowchart Key

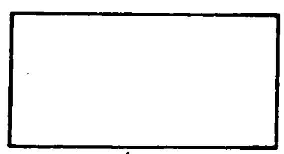
FLOWCHART KEY



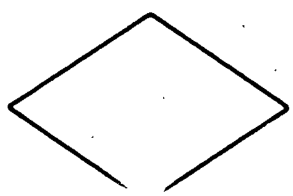
Input at the terminal



Input

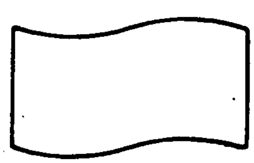


Program segment

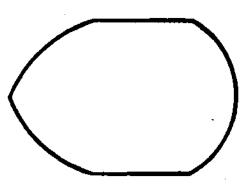


Decision

Document : printer output



Plotter output



Graphic terminal display

APPENDIX 4. Letter of permission

The British Cartographic Society

Lise Allard,
Department of Geography,
The University of Alberta,
Edmonton, Canada T6G 2H4

5th December 1984

Dear Miss Allard,

Thank you for your letter of 21st November concerning your wish to use material from The Cartographic Journal in your thesis. My apologies in not getting a reply off to you sooner; I hope this doesn't get too much delayed in the Christmas post.

There is no objection to your using the flow map illustrated in the paper by R Kern and G Rushton (Figure 2 on page 133 of Volume 6, No.2, December 1969) for the purposes of your thesis.

Should you subsequently wish to use this diagram in any article for publication, you should make an approach to the authors of the paper and of course also make a further request to me.

Yours sincerely,

G R P Lawrence.

Hon asurer
P S Hodson
64 Warren Close
Shirley
Southampton
SO1 6BJ

Hon. Secretary
Patrick E. Sorrell
Department of Land Surveying
North East London Polytechnic
Forest Road
London E17 4JB

Editor: The Cartographic Journal
G. R. P. Lawrence
Department of Geography
University of London King's College
Strand
London WC2R 2LS