AN ABSTRACT OF THE THESIS OF

Richard Roger Hagestedt for the degree of Master of Science

in Forest Management presented on ______________

Title: Spatial Allocation of Land Uses in Land Use Planning

Abstract approved: Signature redacted for privacy.

Pamela J. Case

The purpose of land use planning, as conducted by the Forest Service, is to allocate land uses. The techniques employed in the current planning process fail to take location of the land allocations into account in any systematic manner. The resulting solutions may be inconsistent with planning goals; the land use patterns produced may not provide the maximum value of goods and services possible while protecting long-term biological productivity of the Forest.

This study examines the impact of location on land allocation decisions, developing a strategy and set of techniques for incorporating spatial factors into the allocation process. Three spatial factors affect land allocation decisions: 1) the size of a land unit required to make management of a use practical, 2) conflicts caused by the adjacent location of specific uses, and 3) the need to organize uses across the landscape to take advantage

of certain characteristics of the planning unit.

Three promising strategies are investigated: 1) an optimizing algorithm, 2) an efficient solution algorithm, and 3) an assignment algorithm. The optimizing algorithm replaces the linear program currently employed in the planning process with an integer program able to consider location of land units in the allocation process. The efficient solution algorithm uses an integer program to create a land use pattern from the linear program acreage allocations. Computer core size limitations and the size and complexity of the planning problem prevent application of these strategies. The assignment approach overcomes these difficulties with a heuristic algorithm designed to locate linear program allocations on the planning unit.

The computer programs required to support the spatial allocation strategy include: 1) a computer mapping program, 2) a detail reduction program, 3) an adjacency program, 4) the heuristic program, and 5) a conflict detection program. The mapping program creates land units and keeps track of their location. The detail reduction program eliminates some complexity from the land base data. The adjacency program identifies adjacent land units. The conflict detection program detects conflicts caused by uses located adjacent to each other and violations of minimum land unit size.

The spatial allocation strategy and its associated tools are tested on the Clackamas Planning Unit of the Mt. Hood National Forest. Results of this case study indicate that the approach is workable with minor modifications.

Spatial Allocation of Land Uses in Land Use Planning

by

Richard Roger Hagestedt

A THESIS

submitted to

Oregon State University

in partial fulfillment of
the requirements for
the degree of

Master of Science

Commencement June 1980

APPROVED:

Signature redacted for privacy.

Professor of Forest Management in Charge of Major

Signature redacted for privacy.

Head of Department of Forest Management

Dean of Graduate School

Date thesis is presented

Typed by S. Hagestedt for Richard Roger Hagestedt

ACKNOWLEDGMENT

I would like to express my appreciation and gratitude to the following people. It was through their assistance, support, and understanding that this thesis was possible. My thanks to:

Pam Case,

Bill Ferrell,

Dave Butler

Jeff Arthur,

Doug Brodie,

John White and the Mt. Hood National Forest Planning Staff,

my fellow graduate students,

my parents, sisters, brother, and my dear wife, Sarah.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS

SPATIAL ALLOCATION OF LAND USES IN LAND USE PLANNING

I. INTRODUCTION

The U. S. Forest Service currently is engaged in comprehensive planning for use and management of National Forest lands. The plans produced for each Forest within the National Forest System are changing the pattern of land use throughout the System. Changes in land use will change the mixture of goods and services flowing to the public from these lands for the foreseeable future.

The significance of the changes in goods and services which might be made as a result of the planning process has led the Forest Service to design its planning system as carefully as possible. The system emerging from early trials is an assembly of highly technical components, each for analysis of some aspect of land capability: ecological effects of alternative management regimes, social consequences of shifts in land use patterns, or the achievement of budgetary and operational criteria under alternative plans.

The basic task of the planning process is to allocate land to uses. The plan produced must achieve two important objectives: it must supply a mixture of goods and services to the public and it must maintain the long-term biological productivity of forest land. The mixture of public goods and services is determined by maximizing the total value of the consumer products and recreational opportunities which the Forest could provide. Protection of long-term productivity is achieved by: a) identifying the kind of environmental impacts produced if a land use were to be imposed on a given area of land, b) identifying acceptable levels of these impacts and then, c) constraining the allocation process so that excessive impacts are avoided.

The Spatial Problem

The location of land uses on the planning unit can cause undesirable effects. For example, land uses, such as timber production or developed recreation, which are located on land types consistent with their physical and vegetative requirements, may sometimes have adverse effects on some other resource, such as when they intrude upon the movement of wildlife from summer to winter range. Location also affects the supply of forest products realized by the land use allocations. Timber production, for example, could be assigned to one area of the planning unit on the basis of compatibility with land type, but its location within the planning unit, in terms of road access or size of the unit, could make the assignment impractical or uneconomic. In a related example, land uses, such as timber production and developed recreation, which are consistent with the ecological requirements of the land types to which they are assigned, become incompatible if located next to each other.

The size of a land unit required to make management of a use practical, the problems of conflicts between adjacent land uses, and the need to organize uses across the landscape to take advantage of roads or protect streams are all examples of the importance of spatial factors in land allocation decisions.

Spatial factors are excluded from the most critical phase of the land allocation process in the current system of land use planning. Land allocation is performed in a single phase, in this system, with use of either a linear or goal program. Land use assignments, using this technique, are made on the basis of: 1) causing minimal disturbance to the land type to which they could be assigned and, 2) achieving the maximum product value possible. Spatial factors are treated only as a second step when the linear program output is transferred onto a map of the planning unit. These factors are treated only in cursory fashion: they are considered only if the individual doing the mapping recognizes that the location of a land use will cause some type of resource management problem.

Resolution of any recognized spatial problem usually is left to the professional discretion of the individual doing the mapping. In dealing with such problems, land use planners often are forced to modify the initial land allocation produced by the linear program. Modification of these results undercuts their "optimality" in ways that probably are not clearly understood and which probably vary from planner to planner.

The Present Study

The general purpose of this study is to develop a strategy and set of techniques for incorporating spatial factors into the allocative phase of the planning process.

The general problem of location and land allocation is discussed in Chapter II. Three primary spatial factors are isolated and identified during this discussion. The chapter concludes with a description of the effects of these factors on the land allocation process.

The third chapter contains a review of the most promising strategies and techniques which might be used for spatial allocation of land uses. Criteria for a useful technique are described and each tool is evaluated with respect to these criteria. One technique, a heuristic search algorithm originally developed by J. P. Ignizio (1978), satisfies most of the listed criteria.

The fourth chapter describes adaptation of this algorithm to the spatial allocation problem. The problem itself is broken down into discrete components and its overall structure is described.

The tools and procedures necessary to apply the heuristic algorithm are also described.

The fifth chapter describes application of these tools and procedures to a real problem. The Clackamas Planning Unit of the Mt. Hood National Forest possesses many of the features of a typical problem and is used to illustrate the application. This chapter concludes with a description and discussion of the practical difficulties encountered in the application of this strategy.

The final chapter summarizes the work completed in this thesis and concludes with a suggestion for the next research steps needed in managing the spatial allocation problem.

II. LOCATION AND LAND ALLOCATION

Only recently have spatial factors been excluded from the land allocation process while planning for use of the National Forests. Originally, site capability was the primary determinant in the location of a land use. Land use decisions were made by professional foresters who matched use to site. The decisions were based on professional understanding of resource management requirements and rules describing the "highest" or "best" use of the land.

Three social changes altered traditional decisionmaking. First, the absolute number of land use demands has increased dramatically since the end of World War II (Public Land Law Review Commission, 1970). Since the land base remains the same, increased demands have resulted in more intense competition for the limited resources of the forest.

Second, the scale of planning and decisionmaking expanded to include larger areas and longer time periods. Land use decisions in one area more and more frequently began to be affected by the decisions made on neighboring areas (Hirsch, 1970; Hufschmidt, 1969). These mutual effects increase the need for coordinated land use decisions between areas and the consideration of larger areas in the planning process. Similarly, longer planning horizons result from the realization that environmental impacts of many land uses extend far into the future.

The third change, probably an inevitable consequence of increased competition and expanded scale, was in the decision process itself. The first two changes require an individual to consider more information when making land use decisions. Processing this information in any systematic way exceeds human capabilities. Synoptic tools, such as mathematical programming algorithms, and small-scale analytical models, such as computer simulations, are more and more frequently replacing the individual's professional judgment (House, 1976).

The present land use decision is no longer made as a single choice, a result of the professional's judgment about the correspondence between land type and land use activity. Rather, land use decisions are aggregates of choices, many made with synoptic tools and models. The decision process itself is shaped more by the strengths and weaknesses of the available tools than by conscious structuring of choices and design of tools to assist with particular problems.

Linear Program Models of the Planning Problem

The allocative phase of current Forest Service planning processes is structured to a considerable extent by the use of linear programming. The purpose of the land allocation process is to create a land use pattern which maximizes the value of the consumer products and recreational opportunities provided by the land while minimizing the associated environmental impacts. A linear programming algorithm is used by most Forest planning teams to create this pattern.

Linear programming and the problems it addresses are best summarized by Hillier and Lieberman (1974).

> Briefly, linear programming typically deals with the problem of allocating limited resources among competing activities in the best possible (i.e., optimal) way. This problem of allocation can arise whenever one must select the level of certain activities that compete for certain scarce resources necessary to perform those activities.

In the case of land use planning, the "limited resource" is the productive capability of the forest land. The productive capability of a forest depends upon its physical and vegetative characteristics. Since the forest's physical and vegetative characteristics vary a great deal from place to place, forest land is sub-divided into land type units, each with unique physical and biological characteristics (Bell, 1976). These land type units become, in effect, the limited resources of the forest.

The "competing activities" in land use planning are management practices which will occur when the land is assigned to a certain use. Each land use will have a unique set of management practices associated with it. Since an area of land often can be used for more than one purpose at a time, the "competing activities" are often sets of management practices designed to produce certain land use combinations.

Management activities of any sort produce consumer products, recreational opportunities, and environmental effects. The allocation problem is regarded as one of assigning the limited acreage in each land type to the competing activities so that the maximum amount of consumer products and opportunities are produced while environmental disturbance is minimized.

An example of the land allocation problem formulated for the linear programming algorithm and the type of solution produced are shown in Tables I through III. As Tables I and II show, all the acreage of a given land type is grouped together to form a sum total of acreage for that type. In actuality, however, the acreage for each land type is scattered in different sized parcels throughout the planning unit (Figure 1). Therefore, the linear programming model is insensitive to location.

TABLE I. THE LINEAR PROGRAM MODEL OF THE ALLOCATION PROBLEM

Output	Land Types							Goals and Constraints
	LT 1			LT 2		LT 3		
	Management Activities							
	A1	A2	A3	A2	A3	A1	A3	
Variable	(1)	(2)	(3)	(4)	(5)	(6)	(7)	
	---------------- units per acre ------------------							
Goal 1		10	25	20	15		40	= maximum
Goal 2	5					10		= 5,000
Goal 3	15	20		10		20		= 12,000
Goal 4	1.0		0.5		1.0		2.0	= 2,000
LT 1 (acres	1	1	1					= 1,000
LT 2 (acres)				1	1			= 500
LT 3 (acres)						1	1	= 800

TABLE II

FORMULATION FOR THE LINEAR PROGRAM ALGORITHM

Maximize

$$Z = 10X_2 + 25X_3 + 20X_4 + 15X_5 + 40X_7$$

Subject to

$$5X_1 + 10X_2 + 10X_6 = 5,000$$

$$15X_1 + 20X_2 + 10X_4 + 20X_6 = 12,000$$

$$1.0X_1 + 0.5X_3 + 1.0X_5 + 2.0X_7 = 2,000$$

$$X_1 + X_2 + X_3 = 1,000$$

$$X_4 + X_5 = 500$$

$$X_6 + X_7 = 800$$

$$X_1, \ldots, X_7 = 0$$

TABLE III

LINEAR PROGRAM SOLUTION TO THE ALLOCATION PROBLEM

Acreage Allocations	Goal Levels
X_1 = 400 acres	Goal 1 = 44,000 units
X_2 = 0 acres	Goal 2 = 5,000 units
X_3 = 600 acres	Goal 3 = 15,000 units
X_4 = 300 acres	Goal 4 = 1,900 units
X_5 = 200 acres	
X_6 = 300 acres	
X_7 = 500 acres	

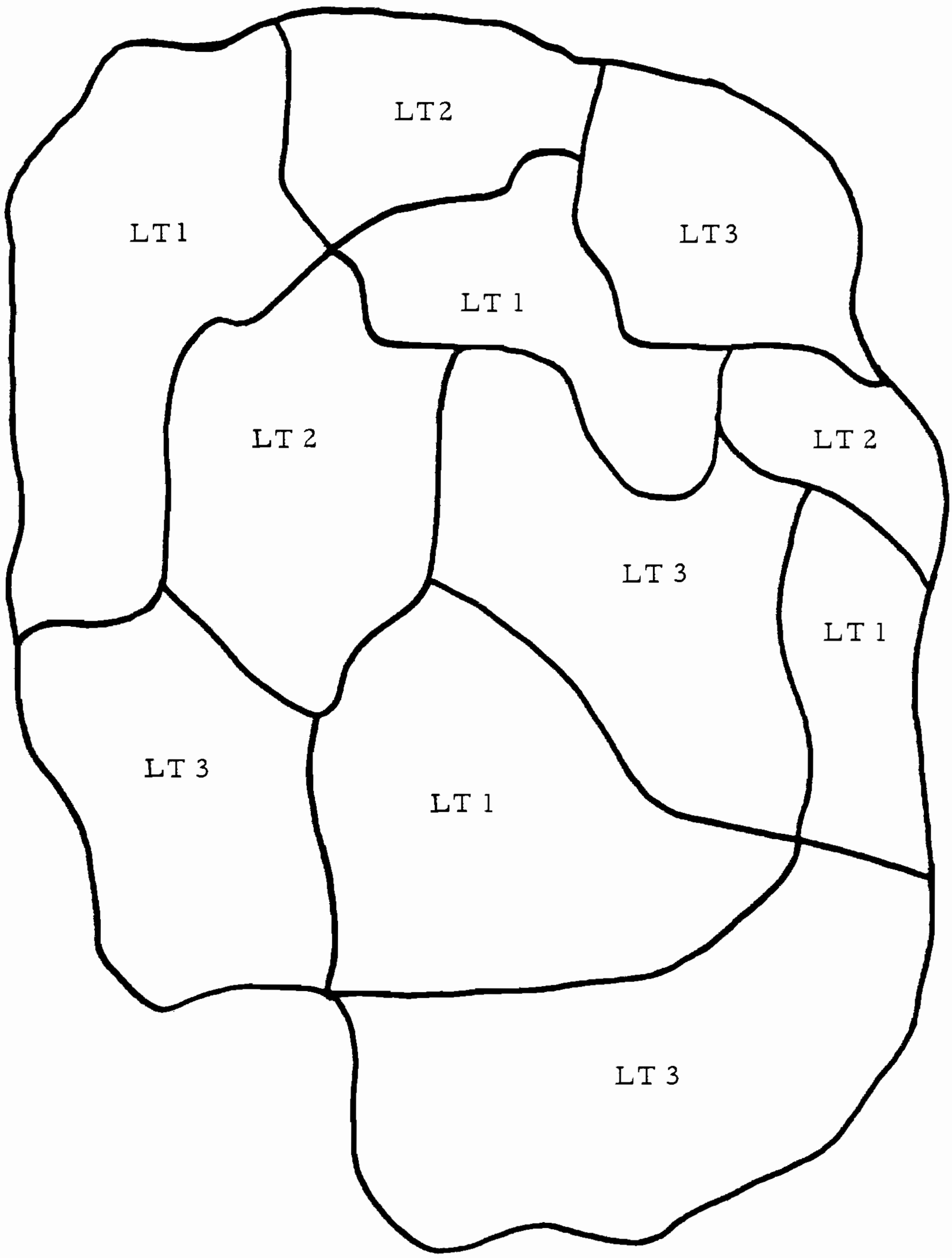

Figure 1. A Typical Land Type Map

Current Forms of Spatial Allocation

Current forms of spatial analysis in the land use planning process typically begin with the linear program output and a land type map of the planning unit. The land allocations produced by the linear programming algorithm must be transferred to the planning unit, creating a land use pattern. Since the linear program output includes the acreage allocations without reference to the location of this acreage, special procedures must be taken to locate the activities on the planning unit.

The process of mapping the activities onto the planning unit is done entirely by humans. A planner must locate acreage of a particular land type on the map and assign one of the allocated activities to it. He must do this for the entire planning unit, assigning every allocation while taking into consideration the acreage and location of each assignment. As the pattern of land uses develops, the planner must consider each new assignment in the context of those made previously. This consideration is crucial if the final pattern is expected to achieve the value of forest products and maintain the level of environmental impacts indicated by the linear program solution.

The basic problem with the procedure is that humans cannot optimize in problems of this complexity. They can't create an optimal pattern for three reasons. First, all of the needed information can't be considered by a single individual at once. Second, the individual often is selectively sensitive to the needs of land uses or restrictions on resources which are most familiar to him because of professional training. That is, selective perception often causes an individual to overlook some spatial conflicts among certain resources. Third, the individual has inadequate time to search every possible combination of activities and compare every potential solution. Optimization requires comprehensive consideration of all the factors in the decision problem.

The overall result of planners manually locating the land use assignments is to undercut the optimality attained by the linear programming algorithm. The assignments made by the planner may meet the acreage allocations specified by the linear program solution but, in many cases, will not produce the effects predicted by this solution.

The scope of the problem becomes most apparent after the pattern of activities is completed for the first time. A careful examination of the pattern will turn up areas where conflicts between activities will prevent achievement of the linear programming goal. This inspection compounds the problem for the planner; he must rearrange the assigned activities to eliminate this condition, not create additional location problems, and still correctly assign the acreage allocated by the linear program. In many cases, the planner may not be able to anticipate all the consequences of any possible rearrangement, forcing him to settle on a pattern which fails to achieve the optimal condition.

Types of Spatial Factors

To discover what kinds of spatial factors intrude upon land allocation decisions, a number of Forest Service planning documents were examined and the land use planning staff of several National Forests in Region Six were interviewed. Most of the staff members had never considered the spatial allocation problem in any depth and therefore had limited perception or knowledge of the spatial factors crucial to a land use allocation process. This fact eliminated any possibility of using a survey of planners to develop a description of the crucial spatial factors.

However, these discussions, combined with my study of the Environmental Impact Statements (the land use plans), did produce a number of examples of spatial problems in land allocation. Examination of their common characteristics indicate that they could be grouped into three basic types about which generalizations could be made.

The three underlying spatial variables affecting land allocation decisions are:

1) management unit size,

2) adjacent-use spillovers,

3) collocation patterns.

In the land allocation process, one of the critical spatial factors is the size of an individual management unit. Certain activities require a minimum number of acres grouped together before they can be imposed on the land.

Examples of activities which require a minimum management unit size include some wildlife habitats, timber harvest units, and wilderness areas. The wildlife habitat for some species, such as pileated woodpeckers, must have a minimal acreage of a particular land type to maintain the species (biological requirement). The timber harvest process requires management units of sufficient size to offset the fixed costs associated with access and management of the timber (administrative requirement). Originally, by Congressional designation, wilderness areas had to contain a minimum of 5,000 acres (administrative requirement).

The management unit must contain sufficient acreage so that the activity assigned to the site can meet the applicable biological, physical, or administrative requirements. Since the reason for the requirements varies from activity to activity, the minimum management unit size varies for each activity. In addition, the appropriate minimum unit size also depends on the land types involved in the allocations. Some land types are able to satisfy the requirements for a particular activity with fewer number of acres. Therefore, the minimum mangement unit size depends on the activity under consideration, the applicable requirements, and the land types involved.

Another critical spatial factor is the "spillover" caused by locating some activities adjacent to one another. Adjacent land uses may produce both positive and negative spillovers, depending on the activities involved. A planner must decide whether the aggregate impact of these spillovers produces a desirable or an undesirable result. If the total effect is desirable, then deliberate steps should be taken in the allocation process to locate these activities adjacent to each other. Undesirable spillover effects require relocation of the activities to avoid their adjacency.

An example of a desirable grouping arises in the establishment of wildlife habitat. Some wildlife species, such as elk, require a certain combination of vegetative habitats for forage, shelter, and reproductive functions. An example of an undesirable grouping is the placement of timber management next to developed campgrounds. Timber harvesting activities often create noise, hazards, and undesirable scenery. The result of such placement would produce a land use pattern which would fail to provide the recreational opportunities dictated by the linear programming solution.

The third prominent spatial factor in the allocation process is collocation. "Collocation" refers to the process of deliberately locating activities on the planning unit so that they form a particular pattern across the face of it. The pattern to be formed depends upon the special requirements of some activity or upon the spatial organization of the planning unit as a whole. For example, two prominent physiographic features frequently involved in the location of activities are road systems and streams. Developed campgrounds usually are assigned to the land next to the road system. In order to minimize the environmental effects created by automotive camping and protect the aesthetic attractions of the campgrounds, it is desirable to space the campgrounds some distance from each other. However, it is also important to consider travel distance and access in designing developed recreational opportunities for an urban public.

Collocation and adjacent-use conflicts obviously have much in common and may not actually be mutually exclusive variables. It may be that adjacent-use conflicts simply depend more on the nature of the activity than the land type while the reverse is true for collocation problems. Or, the two factors may vary only according to the scale of the spatial factors involved in a specific planning problem. For the purposes of this study, however, the magnitude of differences between these factors and the fact that they must be treated in different ways, seems to warrant treating them as two distinct spatial variables.

III. PROMISING ALGORITHMS

There are two major flaws in the current methods of translating linear program allocations onto maps of planning units. First, perception of location conflicts and judgment about their resolution vary from individual to individual. Second, individual planners are unable to carry in their minds all the information needed to make synoptic location decisions. The first flaw can be overcome if planners know what spatial factors to take into account, if they have a systematic procedure for making activity assignments and, if they have a uniform set of rules for resolving conflicts in the location of activities. Yet, the amount of detailed information to process in the spatial analysis problem still would prevent planners from finding the most efficient land use pattern.

The spatial allocation problem, even though extremely complex, is one of decisionmaking under the condition of certainty. That is, the effects of assigning any activity to any location can be known. The feature of certainty in the problem implies that there is an optimal or most efficient land use pattern for any set of activity-location assignment values. The troublesome problem is how to design a procedure or system of decision rules (i.e., an algorithm) for finding the most efficient land use pattern.

A computerized algorithm capable of searching for the optimal pattern would improve the present situation in several ways. Its use would decrease the selective perception and variability between people, since the technique would standardize treatment of the spatial factors in the assignment procedure. An algorithm increases the efficiency of the user in two ways. It drastically reduces the time required to solve highly detailed problems. More importantly, it allows the user to focus his attention on the value-judgment components of a problem, rather than on its computational aspects. Finally, because the decision rules are publicly available in the algorithm, they can be scrutinized over time, compared with others, and amended as knowledge of spatial influences on land use decisions grows.

Limiting Factors in the Search for an Algorithm

A great number of promising algorithms exist in the field of operations research (Phillips, et al., 1976; Ackoff and Sasieni, 1968). These include the specialized procedures of mathematical programming, which are designed for problems of resource allocation under conditions of certainty (Hillier and Lieberman, 1974; Simmons, 1972; Taha, 1971; Wagner, 1970). Two elements of spatial allocation problem greatly narrow the search for an algorithm among the mathematical programming techniques.

The first restricting element is the choice of principal criterion to be used in the algorithm. There are two broad options. One is to strive for the optimal solution to the total problem. In this case, the linear program would be replaced with some other optimizing technique. The algorithm would find the location for all activities which would maximize the value of consumer products and opportunities developed in the planning unit while meeting environmental constraints. The alternative is to retain the linear programming technique and use the results it produces as the criterion for location decisions. Some special algorithm could be devised for making location assignments: these assignments would have to sum so as to match the acreage allocations first provided by the linear program solution. The second option implies that the criterion of "efficiency" would be substituted for "optimality" and that "efficiency" would be judged in relation to the linear program solution.

The second limiting factor in the search for an algorithm is the size of the problem. Size is expressed by the number of units of analysis and the number of variables in the problem. Essentially, the greater the size of the problem, the more numerous the decisions to be made in assigning a land use activity to a land type.

Size of the problem determines whether it can be handled by a computerized algorithm or not. The more numerous the decisions to be made, the less likely the problem will fit within existing computer capabilities. The larger the problem, the less likely that an optimizing algorithm can be used (since these require simultaneous consideration of a great deal of information), and the more likely the problem must be broken into sub-parts in order to solve it. As indicated above, the larger the problem, the more likely that an efficiency solution algorithm criterion must be selected.

Promising Algorithms

An Optimizing Algorithm

A strong argument can be made for adopting optimality as the principal criterion for the proposed algorithm. The linear programming model of the planning unit does not take the spatial organization of land type units into account, yet spatial factors will affect the products and opportunities which can be developed in the unit. The linear program model may be unrealistic and, therefore, the "optimal solution" it produces may also be unrealistic. A genuinely optimal solution could be developed if the linear programming technique were replaced by another optimizing algorithm which could take location into account.

An integer programming algorithm in place of the linear program might overcome the difficulty associated with locating acreage allocations on the planning unit. The problem formulation for the integer program is similar to the linear programming formulation except that the acreage in the planning unit would be divided into "management units" rather than land type units. While the linear program allocates acreage from each land type to the various activities, the integer program allocates each management unit to a single activity.

Acreage in the planning unit could be divided along land type boundaries to create the management units. The larger blocks might be split into two or more management units in order to allow the algorithm to allocate this acreage to more than one activity when optimal. Another possibility would be to create a management unit out of the acreage of two or more land types. This tactic would be useful when there is insufficient acreage of any one type to support a specific allocation.

Since the allocation of a management unit to a particular activity is essentially a yes-or-no decision, a zero-one integer programming algorithm could be used. This type of algorithm examines the contribution which the assignment of each activity to a particular management unit would provide to the value of consumer products and recreational opportunities, then selects the assignment which meets all the constraints (environmental or spatial) and contributes the most value (Geoffrion and Marsten, 1972; Hillier and Lieberman, 1974).

Since the algorithm assigns an activity to a management unit and the location of that management unit is known, the operator can now consider the relative location of activities in the allocation process. That is, the operator can include additional constraints in the problem formulation to control the allocation of management units based on their location.

The most common use of this constraint would be to make the assignment of an activity to a particular management unit depend partially on the activities assigned to adjacent management units. Constraints of this form either strictly prohibit adjacent assignment of certain activities or place a penalty on the occurrence of this condition.

The zero-one integer programming formulation of the land allocation problem with location constraints is presented in Appendix A. This information can be compared with that provided for the linear programming solution technique in Tables I through III.

The major advantage of the integer programming algorithm is that it would consider both minimum management unit size and adjacent-use conflicts in producing an optimal assignment pattern. However, the algorithm in this form cannot consider the collocation factor. Unfortunately, the algorithm also runs into a number of practical disadvantages.

The problem formulation for most planning units would probably exceed the computer core size required to solve the problem. Any attempt to make the problem fit the computer core size, by reducing the number of management units and increasing their size, affects the precision of the solution and may make it invalid. A number of runs usually are required in the allocation procedure to determine solution bounds under various constraints and also to produce alternative allocations from which to select. The total cost of all these runs may be prohibitive. So, although a zero-one integer programming algorithm may be a good theoretical replacement for a linear program, practical limitations hinder its application.

An Efficient Solution Algorithm

The practicality of a pure optimization algorithm is limited by the size and complexity of the spatial allocation problem. The alternative is to seek an efficient, rather than an optimal solution.

The integer programming technique also might be used for this purpose. Most of the problem formulation would remain as above, but the integer program would contain only three types of constraints. These would be: a) the acreage assigned must be equal to the linear program acreage allocations, b) only one activity could be assigned to each management unit, and c) the assignment of one activity would depend on the assignment of other activities. This problem formulation is also presented in Appendix A.

Dividing the spatial allocation problem into two smaller problems permits both the linear and integer programming to fit within existing hardware capacity. However, a number of linear program runs are required to bound the allocation problem and create alternative solutions. Plus, the zero-one integer program must be run a number of times for each alternative linear program allocation. Each of these runs include a different composition of spatial constraints. Some constraints are relaxed and others are added until all important spatial factors have been considered in the final assignment pattern. The problems are still so large that the cost of making several runs of each algorithm is prohibitive. Few National Forests would find the solution costs for a normal problem to be a reasonable investment.

An Assignment Algorithm

A more promising possibility employing the efficiency criterion is to substitute an assignment algorithm for an optimizing algorithm. Ramalingam (1976) notes that assignment algorithms are particularly compatible with the efficiency criterion in resource allocation problems. They,

> . . . arise in the context of associating each of the N requirements to each of the N available means of satisfying them (requirements). . . . One of the important characteristics of the assignment problem is to allocate resources on a one-to-one basis. . . . Another characteristic is that the total number of requirements equals the total number of resources. The effectiveness coefficients indicate what might result from allocating each requirement to each resource. . . . The objective is to establish the assignment that minimizes the sum of effectiveness coefficients of the selected combinations.

Assignment algorithms overcome the size problem, but at the expense of finding an optimal solution. Assignment decisions are made in a step-by-step fashion. At each step, after the assignment is made, the size of the problem is reduced for all subsequent steps. The algorithms consider the current assignment and its affect on future options, but can't consider possible adjustments to earlier assignments which could improve the final solution. Essentially, assignment algorithms can overcome the size problem because they do not consider all possible land use patterns.

Ignizio (1978) has developed an assignment algorithm which can be adapted to the spatial allocation problem. His algorithm searches the array of options created by potential assignments and selects that assignment (land use activity location) which maintains the greatest number of future possibilities. Accordingly, his algorithm, "the heuristic search algorithm," can deal very well with the adjacent-use spatial factor. Although the algorithm cannot handle the collocation factor directly, the problem formulation could easily be structured so that the desired collocation pattern appears. The algorithm does not deal with the problem of minimum management unit size, but the procedure in which the algorithm is embedded also could be developed to take this factor into account.

Problem Formulation for the Heuristic Algorithm

If the heuristic technique were to be used, the formulation of the spatial allocation problem would have the following characteristics:

(1) There must be a finite number of management units in the planning unit.

(2) Each management unit can be assigned to only one of a finite number of activities.

(3) Because of the characteristics of adjacent activities, assignments will be promoted where a favorable condition arises or avoided where an adverse situation appears.

(4) The acreage of the management units assigned to each activity should equal the acreage allocation determined by the linear programming algorithm.

(5) The primary objective will be to assign activities to management units in such a way that the impact of each adjacency condition is considered and the acreate allocations are met.

The heuristic approach requires the computer to make a number of conditional checks; each of which narrows the selection process substantially and allows the activity assignments to be made. The procedure followed by this approach is briefly outlined below.

The planner must arrange the management units in some order. Since the computer algorithm will assign the management units in that order, the planners may wish to place certain units near the beginning or order them by decreasing acreage.

For all adjacent management units, the planners must indicate the occurrence of any favorable or unfavorable conditions which will result if certain activities are assigned. With this information, the computer algorithm starts the assignment process by considering the first management unit. The computer checks its land type, acreage, and all linear program acreage allocations of that land type. The acreage of that land type may be allocated to any number of the activities. The computer program compares the management unit acreage to the acreage allocations to determine which activities are eligible for assignment to this management unit. If the management unit acreage is the smaller acreage for a particular activity, then that activity is considered for possible assignment.

The computer repeats this comparison for each activity until it has identified all eligible activities. The computer program then checks the impact each possible assignment has on future assignments to adjacent management units. The favorable and unfavorable conditions created by each possible assignment are compared and the activity which creates the most desirable situation is selected. When the assignment of two or more activities are equally desirable, the activity with the largest remaining acreage allocation is selected. The acreage of this management unit is deducted from the appropriate acreage allocation.

The updated acreage allocations represent a running tally of the remaining acreage of each land type to be assigned to each activity. The algorithm must consult this tally to determine the eligible activities for each management unit and to assure that the algorithm is attempting to meet the linear program acreage allocations.

The same assignment procedure is repeated for each management unit, except now, the algorithm also must consider previous assignments made to adjacent management units. The favorable and unfavorable conditions created by these activity assignments will further restrict the selection process. The algorithm continues through the list of management units, attempting to assign an activity to each unit.

However, in some cases, no activity will be eligible for assignment. This occurs because of two constraints which any activity must satisfy before it can be assigned to a management unit. The assigned activity must not: 1) create a conflict with any previously assigned adjacent activity and 2) cause the acreage assigned to exceed the linear program acreage allocation by more than a specified tolerance. After assigning all management units possible with these constraints in effect, the algorithm stores the assignments and a matrix indicating how close to the acreage allocations the assignment process came. Then, the algorithm allows the user to terminate the assignment process or continue it with one or more of the constraints relaxed.

Subsequent runs assign the remaining management units based on the activities previously assigned. The constraints are relaxed one after another in the following order: 1) strict prevention of adjacency conflicts with previously assigned activities is eliminated and activities are selected which simply minimize all possible (current or anticipated) adjacency conflicts, 2) the tolerance level for achieving the acreage allocation is eliminated but an activity still cannot be assigned if that allocation was previously satisfied, and 3) all attempts to meet the acreage allocations are dropped and the activity is selected which minimizes all possible adjacency conflicts. Since the results of each run are stored, the user can examine the assignments at each step to get a feel for the location of conflicts or select one of the intermediate runs and assign the remaining management units manually.

A more detailed description of how the computer algorithm carries out the above procedure and an example of its operation is presented in the following chapters.

Summary

Inability to treat spatial influences in land allocation decisions in any systematic way has caused them to be ignored in the allocation process itself. A computer algorithm which could take these factors into account would be a useful tool for the land use planner and would help reduce the present variability in allocation decisions among National Forests. Unfortunately, the size of the spatial allocation problem prevents us from making use of an optimizing algorithm, such as integer programming. This is so even when a new unit of analysis, the management unit, is created to replace the land type unit and reduce the size of the problem.

The best alternative seems to be to sub-divide the problem into smaller components. Sub-division allows us to retain the linear programming tool and devise a specialized algorithm for making location assignments. Sub-division also implies that the objective of the spatial allocation problem is to find the most efficient approximation to the linear programming solution.

The collection of assignment algorithms is compatible with this objective. Ignizio's heuristic technique, in particular, provides a simple and effective way of taking the two most difficult spatial factors into account when assigning activities to management units. This algorithm has several advantages over the optimizing strategies explored. It is relatively simple to understand and inexpensive to use. Unlike the optimizing routines considered here, it can be modified to take the collocation factor into account. The major difficulties with the algorithm are that it does not produce optimal solutions and it may not produce consistently efficient solutions over a wide range of spatial allocation problems. Optimal solutions cannot be developed because the algorithm cannot search all possible improvements to the first assignment. The consistency of the algorithm is unknown at this time and will have to be tested before the algorithm could be widely used.

IV. THE SPATIAL ALLOCATION STRATEGY

A number of computer programs were needed to perform certain tasks in the spatial allocation process. The tools developed include: a) a computer mapping routine, b) a detail reduction program, c) an adjacency program, d) a spatial assignment procedure, and e) a conflict detection program. The mapping routine creates management units and keeps track of location. The detail reduction program eliminates some of the complexity in a land type map. The adjacency program detects adjacent management units. The assignment procedure utilized in the spatial allocation process is the heuristic approach presented in the previous chapter. The conflict detection program is a computer routine designed to detect activity assignments which produce detrimental spillover effects when they are located next to each other. It also assists the planner in identifying assignments which violate the minimum mangement unit size rule. This chapter describes the tools in greater detail. It focuses on their functions in the overall process and the linkages between them.

Tools Needed for the Procedure

The Computer Mapping Routine

The spatial arrangement of activities on a planning unit affects achievement of the goals described by the linear program acreage allocations. Some device is needed for keeping track of the locations of the management units, and later, of the activity assignments.

A number of computer mapping routines are available which are able to keep track of location. In most cases, they are unable to provide all of the mapping functions needed or handle the type of data used in land use planning. The E-ZMAP (Child and Rollin, 1976), IMGRID (Sinton, 1976), and SYMAP (Dougenik and Sheehan, 1976) computer mapping systems each contain many of the required mapping functions but they are unable to handle the amount of data needed. That is, the number of symbols needed to display land types, management units, or activities usually exceed the symbol list available in each of these mapping systems.

Some mapping systems may satisfy both requirements but were not available for use in the present research. The RAP mapping system, for example, is recommended for adoption throughout the Forest Service for use in land use planning, but is currently incompatible with Oregon State University's computer system. The Mt. Hood National Forest, the only Forest in the Pacific Northwest to make use of a computerized mapping system, is currently using a modified version of R3MAP. This program contains functions necessary to process and display planning unit data. The functions include display of maps and their legends, an ability to combine maps, and an ability to aggregate symbols. These functions are used to develop maps of land type units by combining maps of individual forest resources. The modified R3MAP program uses some computer system features not currently available on Oregon State University's computer.

The computer mapping routines discussed above all use cellular mapping techniques. The map information is displayed in square cells of a fixed acreage. The symbol assigned to each cell represents some characteristic of the acreage of that cell. All of the acreage inside the cell boundaries is assumed to have similar characteristics. Since the characteristics of the planning unit are not laid out in neat squares of fixed acreage, each cell symbol represents the dominant characteristic of the acreage in that cell.

Mapping routines, like R3MAP, have standard functions which are used by planners to create land type maps. The current land allocation process limits the mapping routine to this contribution and possibly to a display of the final land use pattern. However, the proposed spatial allocation procedure requires additional information and data processing which only an expanded computer mapping routine can provide. The activity assignment procedure requires the planning unit to be broken down into management units and information provided on their acreage, land type, and the adjacent location of other management units. The conflict detection program, which detects detrimental spillover effects and provides information on management unit size, also requires additional information not available from conventional mapping routines. This information includes each management unit's acreage, land type, and adjacent management units, as well as the activities assigned to it and the adjacent units.

In light of these requirements, a greatly expanded version of the R3MAP routine was built to support the spatial allocation procedure. The functions provided by this mapping routine are listed in Table IV along with a short explanation of their purpose. A program listing of the mapping routine is presented in Appendix B.

The Detail Reduction Program

The land type map of many planning areas may contain a great deal of complexity. That is, in addition to a large number of land types, each land type is broken into small parcels and these are scattered across the planning unit. Each parcel of each land type must be treated as an individual management unit when the mapping routine creates the management unit map. A large number of single-cell management units will result; pushing the total number of management units beyond that which the computer memory can contain or which can be handled at reasonable cost.

TABLE IV. MAPPING ROUTINE FUNCTIONS

SUBROUTINE	PURPOSE
DIRECT	The map directory. This mapping routine is an interactive computer program which allows up to twenty maps to be available to the operator during a single run. The directory lists and keeps track of the available maps, aiding in the selection of maps for other mapping functions.
EXCESS	Handles the map overflow. When the number of maps input or created exceeds the twenty allowed, this subroutine locates one map on the recoded data file.
COPY	Transfers maps onto the recoded data file.
RECODE	Recodes map symbols into a standardized numeric code and collects statistics on the occurrence of each symbol.
IN	Inputs maps in either of two formats.
SELECT	Selects and stores maps on the recoded data file at the completion of a run.
OUT	Displays maps at the terminal or outputs them on a line printer file.
COMBO	Combines the characteristics on several maps into a set of symbols on a single map. Each symbol on the new map represents a unique combination of the original characteristics. The legends of the original maps are also combined to create a legend which describes each unique combination.
AGGASS	Aggregates or assigns cells of each symbol into groups and documents those groups. Each group consists of any number of the original symbols.

Table IV. Mapping Routine Functions (Cont.)

SUBROUTINE	PURPOSE
REORDER	Reorders the symbol list and legend of a given map based on a selective order or by decreasing acreage size.
HILITE	Searches a map for symbols which share one or more common characteristics and lists the documentation associated with these symbols.
PARCEL	Detects and groups cells of the same symbol which are adjacent to one another.
UPDATE	Changes the titles and/or legend of any map.
MANIP	Manipulates map cells: moves individual cells from symbol to symbol, aggregates cells of one symbol with another, and splits cells of a single symbol into groups of several symbols.

The number of management units must be reduced to an acceptable level without reducing the accuracy of the final land use pattern. The easiest and most practical strategy is to group the isolated single cells of each land type with the adjacent group of cells exhibiting the most similar characteristics. The detail reduction program (REDUCE) is designed to perform this operation on a land type map before it is broken into management units by the mapping routine. A program listing of REDUCE is presented in Appendix C.

The Adjacency Program

The spatial assignment procedure and conflict detection program both require a list of adjacent management units. The assignment procedure uses the adjacency information to avoid assigning activities next to each other which produce detrimental spillovers when located on adjacent management units. The conflict detection program uses information about adjacent management units to identify any detrimental spillovers which appear in a land use pattern.

The adjacency program (ADJ) examines a map of management units, identifying the management units adjacent to each unit and recording the number of times each adjacent pair occurs. The program listing of ADJ is presented in Appendix D.

The Spatial Assignment Process

The heuristic approach described earlier in Chapter III is used here. The program listing of the heuristic (HEURIST) is presented in Appendix E.

The Conflict Detection Program

The conflict detection program is a computer program which provides information about the management unit size of assigned activities and detects detrimental spillovers created by the assignment of activities to adjacent locations on the planning unit. This program simply provides the planner with information which he can use to make a quick check for violations of minimum management unit size or the location of any adjacent activities producing detrimental spillovers.

The size of a management unit is checked after activity assignments are made. A single activity may be assigned to adjacent management units, not necessarily of the same land type, to produce a larger management unit for that activity. This is done by ignoring the old management unit boundaries and creating new management units based on adjacent cells assigned to the same activity. The mapping routine carries out this process. First, it assigns the activities to the land type map, erasing the management unit boundaries. Then, adjacent cells assigned to the same activity are grouped into a parcel or "new" management unit. The adjacency program detects the activities assigned to locations next to these management units.

This information is transferred to the conflict detection program. The computer program examines each management unit, checking its acreage against the minimum acreage figure provided for management units assigned this activity. If the acreage is smaller than the minimum required, the program lists the assigned activity, the management unit acreage, and the activities assigned to adjacent management units. This information, plus a map display of the assignments, allows the planner to detect violations of minimum management unit size.

Detection of detrimental spillovers can use either the "old" management units or the "new" ones developed in the above procedure. The "old" management units will pinpoint the location of those management units which should be reassigned by the spatial assignment procedure in the next run. However, the "new" management units indicate the true acreage assigned to an activity. The "true" acreage of a management unit is important since sufficient size often overcomes detrimental spillovers by providing a buffer for the activities. However, if a detrimental impact still exists (as shown by the "new" management unit), the planner must refer back to the "old" management unit map to locate those units needing reassignment. Having both options available allows the planner to select the information important in different circumstances.

The planner loads the conflict detection program with: a) information on the activities assigned, b) the management units, and c) a matrix indicating which activities will produce detrimental spillovers and the severity of these spillovers. For each management unit, the program checks the activities assigned to adjacent management units. The spillover matrix tells the program if a detrimental spillover exists between the activity assigned to the management unit being considered and those adjacent to it. If a detrimental spillover is found, the program lists the severity of the spillover, the activities involved, and information about the management units involved.

The conflict detection program provides the planner with information about the land use pattern which is difficult to detect with only a simple visual examination of an activity assignment map. The information can be used by the planner to improve the spatial assignments made in a subsequent run. A further explanation of this process is described in the next section of this chapter. A listing of the conflict detection program (DETECT) is located in Appendix F.

The Procedure

The procedure can be described most easily by referring to its major stages. These stages are shown in Table V.

Inputs to the Spatial Allocation Procedure

The spatial allocation procedure is only one step in the entire land use planning process. Rather than reconstruct the entire process here, the results of intermediate steps in the planning process will be treated as inputs to the spatial allocation procedure. The two inputs are: 1) a complete land type map with

TABLE V. THE SPATIAL ALLOCATION PROCEDURE

INPUTS TO THE SPATIAL ALLOCATION PROCEDURE

	Land type map. Linear program acreage allocations.

DEVELOPMENT OF MANAGEMENT UNITS

REDUCE	Eliminate isolated single cells.
PARCEL	Break into management units.
MANIP	*Modify size or shape of management units.

*DEVELOPMENT OF A SPILLOVER MATRIX

ACTIVITY ASSIGNMENTS

	*Preassign activities to some management units.
REORDER	*Place management units in order.
ADJ	Find adjacent management units.
	Input linear program allocations.
	Input spillover matrix.
HEURIST	Make activity assignments.
AGGASS	Transfer assignments to the map.
OUT	Display activity assignments.
	Check allocation achievement table.

DETRIMENTAL SPILLOVERS AND MINIMUM MANAGEMENT UNIT SIZE

	Input spillover matrix.
	Input adjacent management units.
	Input activity assignments.
	Input management unit map documentation.
DETECT	Detect detrimental spillovers.
	Input activity assignment maps.
PARCEL	Break assigned acreage into management units.
ADJ	Find adjacent management units.
DETECT	Detect detrimental spillovers.
	Input minimum management unit acreages.
DETECT	Detect minimum management unit size violations.

*These steps can be modified in a readjustment procedure.

legend and 2) the linear program acreage allocations.

The land type map is developed from inventory maps representing the characteristics of the planning unit. The characteristics contained in the final land type map are those which affect management for the selected group of activities. That is, the characteristics in the land type map indicate the acreage which is suitable and unsuitable for each activity. The linear program allocations indicate the acreage of each land type which is to be managed for each activity. The spatial allocation procedure attempts to locate the acreage allocations on the planning unit; using the locations indicated on the land type map as a guide.

Development of Management Units

Management units are areas of similar land type to which activities eventually will be assigned. They are composed of a number of adjacent cells in the digitized map which have the same or similar land type characteristics. Each management unit is identified by its: a) land type, b) acreage, c) shape, and d) location in the planning unit.

Three computer programs (or subroutines) were created for forming management units. These programs were designed to manipulate the cells of a land type map so that the management units developed could be assigned without serious complications.

There are almost always a large number of isolated single cells of each land type in a planning unit. For example, the original Clackamas Planning Unit land type map contains 1,944 such cells, or 11.72 percent of the total number of cells. If each isolated single cell were to be treated as an individual management unit, the number of units would exceed the information storage capacity (core size) of the computer. A special computer program, REDUCE, was developed to aggregate the map further. This routine groups the isolated single cells with the adjacent group of cells displaying the most similar characteristics.

The PARCEL subroutine is part of the computer mapping program. This subroutine simply groups adjacent map cells of the same land type into single units. For obvious reasons, PARCEL creates the bulk of the management units.

The third routine, MANIP, is another mapping program subroutine developed to make adjustments in the management units created by the PARCEL subroutine. MANIP allows the planner to make changes in management units which are too small, too large, or the wrong shape for the assignment procedure.

The number of management units needed in the planning unit presents the planner with a dilemma. Since each unit must be described by: a) its land type, b) its acreage, and c) the adjacent management units, a large number of units will exceed the core size of even a large computer. A large number of units will also increase computational time or the expense of each assignment run. On the other hand, the more management units there are in the problem, the more options are available for assigning activities to units. The planner must consider the balance between flexibility and expense of problem-solving when determining the number of management units with which to work.

Development of a Spillover Matrix

The spillover matrix reflects the desirable and undesirable impacts caused by assigning each pair of activities to adjacent management units. A spillover matrix is shown in Figure 2. The values in this matrix represent relative levels of desirable and undesirable impacts which are displayed by a range of numbers from positive to negative three. A spillover value of zero represents no spillovers or a neutral impact.

The development of the spillover matrix is important since creation of the final land use pattern is directly affected by the values it contains. The assignment procedure attempts to locate those activities with positive spillover values on adjacent management units. It also attempts to prevent the adjacent location of activities with negative spillover values. At this time, the assignment procedure is unable to distinguish the relative level of positive

ACTIVITIES

		1	2	3	4	5
ACTIVITIES	1	0	0	-1	0	+1
	2	0	0	0	-3	0
	3	-1	0	+2	0	-2
	4	0	-3	0	+1	+3
	5	+1	0	-2	+3	0

Figure 2. Spillover Matrix

or negative spillover values; concerning itself, instead, with the absolute differences represented by the positive, negative and zero values.

The conflict detection program, on the other hand, does distinguish between the relative levels of undesirable spillover impacts. It lists all adjacent assignments with undesirable spill-overs, classifying them according to the severity of the impact. This classification allows the planner to rapidly identify the most severe, therefore most unacceptable, spillovers and take steps to correct the assignment.

Activity Assignments

The heurisitc algorithm, HEURIST, assigns activities to the management units. This algorithm allows the planner to preassign activities to some management units and then assigns activities to the remaining units. The preassignment feature fixes activities to specific locations on the planning unit, forcing the algorithm to fit the other acreage allocations around these locations.

The order in which management units are assigned affects the final land use pattern. The heuristic algorithm assigns an activity to a management unit only if it: 1) helps to meet the acreage allocations produced by the linear program, 2) does not create an undesirable spillover condition with an activity which was previously assigned to an adjacent management unit, and 3) provides the maximum number of future assignment options by minimizing the potential spillovers with the adjacent unassigned management units. Each assignment restricts the activities which can be assigned to subsequent management units by: 1) meeting part of the acreage allocation and 2) blocking future assignments which produce undesirable spillovers on adjacent management units.

Subroutine REORDER in the mapping program permits arrangement of the management units in any selected order, will arrange the units by decreasing acreage, or will allow a select group to be placed first with the remainder arranged by decreasing acreage. Arrangement by decreasing acreage causes the larger, more critical management units to be assigned first, leaving the smaller units to adjust and fine-tune the final pattern.

The heuristic algorithm, as previously stated, is concerned with activities assigned to adjacent management units. Therefore, the computer program ADJ is used to detect all adjacent management units and this information is input to the heuristic algorithm.

Other inputs to the heuristic algorithm include the linear program acreage allocations and the spillover matrix. The function of both in the assignment procedure was also discussed earlier.

The heuristic program attempts to assign activities to management units in such a way as to meet the three conditions previously listed. The first two are constraints: 1) restricting each assignment on the basis of previous activity assignments to adjacent management units and 2) preventing the assignment of an activity which will exceed the acreage allocations by more than a specified tolerance. With these constraints in effect, the assignment procedure usually is unable to assign every management unit. Subsequent runs relax each of the constraints, one at a time, until activities can be assigned to all of the remaining management units.

The results produced by each run can be checked by the planner. The heuristic program stores the assignments made up to that point and updates their effect on the achievement of the acreage allocations. This last piece of information is presented in an allocation achievement table which the planner uses as one measure of the success of this assignment run. The assignments for each run also can be displayed on the planning unit map through the use of the mapping program. The subroutine AGGASS transfers the assignments to the management unit map and the subroutine OUT displays the assignment map at the terminal or places it on a line printer file.

Detrimental Spillovers and Minimum Management Unit Size

The conflict detection program, DETECT, provides two important functions to the spatial allocation procedure. It checks the map of activity assignments for undesirable spillovers and violations of minimum management unit size requirements. This program simply detects and lists the occurrences of these situations; it does not include procedure to correct them. However, this does not diminish its importance. This program detects those undesirable spillovers which could not be avoided in the assignment procedure, listing them so that the planner can take steps to eliminate the unacceptable spillovers in a later readjustment. It also provides the only systematic list of those assignments of insufficient acreage; the only other option being an individual inspection of the assignment map.

There are two strategies for the detection of undesirable spillovers. Both require the input of the spillover matrix, but differ in the rest of the required inputs. Both strategies list certain basic information: 1) the severity of the detrimental spillover, 2) the activities involved, and 3) location on the map. One strategy focuses on management units of a single land type and includes the land types of the management units involved in the spillover. The other strategy focuses on management units of a single activity and the acreages provided reflect the "true" size of the management units involved.

The first strategy, emphasizing land types, requires the input of activity assignments, the list of adjacent management units, and the documentation of the management unit map. The conflict detection program uses this information, plus the spillover matrix, to list every undesirable spillover on the planning unit.

The second strategy, emphasizing the "true" management unit size, requires a number of additional steps. The activity assignment map is input and subroutine PARCEL groups the assigned acreage into a "new" set of management units based on the activities. The adjacency program, ADJ, finds the adjacent management units in this new map. The conflict detection program, in this case, uses the spillover matrix, management unit map of the assigned activities, and list of adjacent management units to detect the undesirable spillovers.

The conflict detection program also uses the newly developed management unit map of assigned activities, the list of adjacent management units, plus a list of minimum acreages to identify all management units which violate minimum management unit size. The program screens each management unit against the minimum acreages, listing the location, activity, acreage, and adjacent activities of any management unit which is smaller than the minimum.

Readjustment Procedure

Major readjustments of the land use pattern can be carried out in additional runs of the spatial allocation procedure by changing certain inputs. The spillover matrix can be altered to promote or prevent a particular adjacent activity assignment in subsequent runs. The order in which management units are assigned can be changed or some of the management units can be preassigned activities. In some cases, a further step back in the process may be required. The management units developed from the land type map can be changed, with additional aggregations of smaller units or divisions of the larger units. All of these readjustments should cause changes in the final activity assignments.

The selection of which adjustments to make has to be based on experience with the assignment algorithm and an inspection of the assignments produced with the current set of inputs.

V. SPATIAL ALLOCATION OF THE CLACKAMAS PLANNING UNIT

This chapter describes the application of the spatial allocation procedure developed earlier to a representative planning unit of the Mt. Hood National Forest. Illustration of the procedure through a case study probably is not necessary to provide an understanding of its underlying strategy. However, the general complexity of the procedure seems to warrant the time and expense involved in the development of a case study. In any newly developed procedure with the complexity this one contains, there may be hidden flaws or unforeseen details in the planning unit which prevent the procedure from being fully utilized. It seemed wise to see whether the spatial allocation strategy outlined could survive the test of workability.

As anticipated, there were difficulties which required modification of the original spatial allocation procedure. The major complication was that the size of the problem, or the complexity of the land types in the planning unit, exceeded most expectations about them. This chapter, in addition to describing the planning unit and illustrating use of the procedure, describes the difficulties presented by such complexity in the planning unit.

The Clackamas Planning Unit

The Clackamas Planning Unit is located on the Mt. Hood National Forest in Oregon. It consists of 353,779 acres of forest land on the western slopes of the Cascade Range, south of Mt. Hood. There are three vegetative zones on the planning unit as designated by the dominant climax tree species: 1) western hemlock (with Douglas-fir as sub-climax species), 2) Pacific silver fir, and 3) mountain hemlock. Within these three zones are 51 distinct forest plant communities. The unit also contains 1,500 miles of rivers and streams, plus 160 ponds, lakes, and reservoirs. The diverse habitats found within the vegetative zones support a variety of wildlife including 73 species of mammals, 150 species of birds, and 25 species of reptiles and amphibians.

The current use pattern includes management for timber, developed recreation, dispersed recreation, and wildlife. Timber harvest is a dominant activity on this planning unit, providing 187 MMBF/year or fifty percent of the total output from the Mt. Hood National Forest. The biological potential for the unit with pre-commercial thinning, genetic stocking, and full stocking level control is 199 MMBF/year. The major commercial species include Douglas-fir, western hemlock, Pacific silver fir, noble fir, and grand fir. The Clackamas Planning Unit also contains 24 campgrounds with 417 camp units and 106 picnic areas. There are two roadless areas in the unit, Bull of the Woods (43,735 acres) and Olallie (8,673 acres). There is also a total of 180 miles of trail throughout the planning unit.

There are two reasons the Clackamas Unit was selected to illustrate the procedure. First, the Mt. Hood Planning Staff volunteered both the problem and their assistance in developing the case study.[1] Second, the planning unit is representative of both the size and diversity of the Forest-wide planning units to be used in the near future.[2] Physical characteristics and current use patterns are typical of the majority of National Forests in the Pacific Northwest Region.

The Spatial Allocation Procedure

The six stages of the spatial allocation procedure presented

---

[1]The Planning Staff provided all the data needed for this study and served as a sounding board whenever unforeseen difficulties arose.

[2]The Clackamas Planning Unit acreage is actually three times the size of the typical planning unit used during the last five years. However, recent changes in land use planning regulations instruct the Forest Service to regard National Forests, rather than their sub-divisions, as the proper unit of analysis. Most National Forests have begun the shift to Forest-wide planning. The Clackamas Unit is actually a composite of three earlier, unfinished units and is being treated by the staff as a prototype for the entire Forest.

in the previous chapter are:[3]

1) inputs to the spatial allocation procedure,
2) construction of management units within the planning unit,
3) construction of the adjacent-use spillover matrix,
4) assignment of activities,
5) detection of detrimental spillovers and violations of minimum management unit size,
6) readjustment of assignments.

Two variations were made from the procedure shown in Chapter IV. First, the information received from the Mt. Hood Planning Staff was not in the correct form to be input into the procedure. It contained some detail unnecessary to an illustration while lacking other data important to the demonstration. The information submitted was modified to an acceptable form. The second variation has to do with the last stage of the procedure. Readjustments to the land use pattern were not made in the case study as they might be in a real planning problem. The case study is designed to demonstrate the procedure; there is no need to correct any unrealistic results.

---

[3]Additional information on each of these stages is provided in Table V and the descriptions contained in Chapter IV.

Inputs to the Spatial Allocation Procedure

The information received from the Mt. Hood Planning Staff includes a digitized land type map of the Clackamas Planning Unit and a list of activities for which the unit can be managed. A portion of the digitized map is shown in Figure 3. The activity list describes twenty-seven activities: six of these were chosen for purposes of demonstration in this study. These representative activities are:

1) reserved sites,
2) developed recreation,
3) dispersed recreation,
4) commercial timber,
5) visual timber,
6) wildlife.

Each activity contains statements of the impacts it will have on the total Forest and products it will provide if assigned to the Forest. The physical characteristics of any land type unit which would limit or prohibit its assignment to that land type are also described. An example of an activity description and site identification is shown in Table VI.

A seventh activity, lakes and streams, was added to the list. This addition was made so that the lakes and streams would

```
MAP  1     OVERLAY

        ORIGINAL RAU MAP - RECODED

COLUMN     2

  1                                                                                      +
  2                                                                                      +
  3                             **********                                               +
  4                             **C9C1C9C9*********                                      +
  5                             **C9C9C9C9C9C9C9**                                       +
  6                             **C9C996C9C9C9C9A6C9**                                   +
  7                             **B4C9B6B6C9A6A6A6A6E8**                                 +
  8                             **B4B4C9B6B6B6A6A6A6A6C9**                               +
  9                             **C9B4B4C9B6B6B6C9A6A6A6E8****                           +
 10                             **C9C9B4B4B6C9B4C9A6A6A6A6A6A6**                         +
 11                             **C9C9B4B4B4B4C9C9C9C9C9C9C9**                           +
 12                             **B9C1C1E5E5A6A6B8B8A6C8C8A6**                           +
 13                             **E7B9C1E5E5A6C9C9B6A9C8B6C8A6**                         +
 14  ****                       **B9E7B9C1E5B400B4B4B4E5E5B6C9C9**                       +
 15  D6**                       **D6B9B9C1C1B1B4B4B4E5C1C1C9C9C9**                       +
 16  D6**                       **D6D6B9B9C1C1B1E5E5E5C1C1C9C1E6C1**                     +
 17  ****          ***************B9D6B9B9B9C1B1E5E5E5C1C1B1B1C1E6****                   +
 18                **D6D6D6D6D6D6B9D6E7E7C5C1C1B1E5E5B4C9C9C1C1C1C1C1**                  +
 19                **D6D6D6D6D6D6C8C8B9B9C5C1C9B1B4B4B4B4D3C1C1C1C1E8A6**                +
 20  *********     ***************B6B6C8C8C0C0C9C9C9B1B1B4D3C9C1C1C1A6A9**               +
 21  **D6D6**                   **B6B6B6B6C9B6C9C9C9D3C1B1B1B1C1A6A6E7A9A9**             +
 22  **D6D6*********            **B9B6B6B6B6B6B6C9C9D3D3B9B9B9B9B6B6A9C2A9A9**+
 23  ***********D6**            **B9B6B9B6B6C8B6C9C9C9C9B9B9B9D4D4A9A9A9E7**+
 24  ***************  *********  **B6B9C8C0C9C9C0E6E6E6E6C9C9C9C9D1C9A6C9C9B6E7+
 25  ********************D6D6*******B9B9B9C0C9C0C9E6E6E6C9C9C9C9C9C9C9C9C9E6E7**+
 26  B9B9D6D6D6D6D6D6D6B9B9B9B9B9B9B6B6C9C9C9C9E6C9E6D6D6D6D6C9B1B1C9E6E6E8**+
 27  B6B6D6D6D6D6D6D6D6D6B9B9B9B6B6B6B9B9C1C1C9C9C9E6C9E6D6D6D6D6C9C1C1C1A6E8E6**+
 28  B6B6B6B6D6D6D6D6D6D6B9E6B6B9B6B9C1C9C9C9C9C9C9D6D6D6D6E6C1C1C9A6E8E6**+
 29  B6B6B6B6D6D6D6D6D6D6B9E6B9B9B6B6B1C1D0E6C9F6C9C9C0C0E6E7E7A6C9E6*******  +
 30  B6B6B6B6D6D6D6D6D6B9B9B9B9B6B6B6EC1C1C1C1C9C9C9E6C9B9C5B9A6A6**          +
 31  B6B6B6B6D6D6D6D6D6B6B9B6B6B6B6C9C9C9B9C9B6B6B6B6B9C5C7C7E6**             +
 32  B6B6B6B6B6D0C1C1B9B9B6B6B6B6C9C9C9C9C9B6B6B6B6E4C6E9E6C0**               +
 33  D3D0C1C1C9C9C1B9B9B9B6B6C9C9C9D0C9C9C9C9B6B6B6C6C6D4E8D4**               +
 34  C9C1C1B9D0D0D0B9B9B9D0C9C9C9C9C9C9C9C9B6B6B6B9B6D4D4D4E7**               +
 35  B9B6B6B6B6D0D0D0B3D0B9C1C1D0D0C9B6B6B6B6B6B6B6B6B9B9B9D4B9**             +
 36  B6B6B6B6B6B6B6C1B6B6B9B9B9B9B9B9E6B6B6B6B6B6B6B9B9B6B6B6B9**             +
 37  D4D4C1B6B6B6B6B6C9D0B6E6B9D0B9B6D0D0B6B6B6B6D2B6B6B6B6B6B9**             +
 38  F6E6E8E8B6B6B6B6C9C1A0B2D0D0B9D0D0B6D2D2D2D2D2D2D2D2B4C9**               +
 39  E7E6E6E6E6C9B6B6B6C9B9E7D4B6B6D0D0D0D0B6B6D0D2D2D2D2B4**                 +
 40  D3E7E6E6D3E8B6B6B6B9B9E9B6B6B6B6D0D0B6D0D0D0D0B9D0B6B6B9**               +
 41  E4E4D4D3F8C1B6E0B9B9B9B9B9B6C9B9C9C9C9C9C1C1C1C1C1C1**                   +
 42  D3D3E7E7E6E8B6E0B9B9E0B9B9B9B9B9B9B6D6B9B6B6B5*********                  +
 43  E4D3D3E7E6E6B9E0B9B9B9B9D0D0D0B9B9B9B9B9B6B6**                           +
 44  B6D3F6E7E7E7B9D0B9B9D0B9B9B9B9B9B9E0D0B9B9**                             +
 45  E7E6F4E9D4E6B6B9B9B9B9F9B9B9B9B9B9B9B9B9B9**                             +
 46  E7F6D3B3B3E6C9B9B9B9B9C0B9B9B9B9C0D8989**                                +
 47  B6C9E7D3E9E6E7D3B9B9B9B9E6D4B9D9B9B9**                                   +
 48  B6C9B6F6E4E6D4E7D3B9B9C1E8E8E8E8B9B9**                                   +
 49  B6C9D0E7D3E7E8B6E6D3E6E6E8E6E8E8E9**                                     +
 50  B9C9B6E6E6D4E6D4B6E7E8E8F8E6E6E6E9**                                     +
 51  B9C1C9C9E6D7B3E6E6E8E6E6E9E7E6E6E6**                                     +
 52  B9C1C9C9B9E7D3E7E6E6E7D4D3D3E7E8E8**                                     +
 53  B9C1C9C9C1B9E6E4E6E7D4D3D3D3D3D4**                                       +
 54  B9C1C9C9A3B6E6B3E9D4D3D3B9E6D3C5*********                                +
 55  B6B9C9C9A3A0A0B3D4D3D3C5B9C7D3D3D4D3C3D3*****                            +
 56  B9B9C1A3A3B1A0E8E4D7D7C5B9C9D3D3D3E7E6E6D3D3****                         +
 57  C9B9B9C3A3A3E6E9E8D0C5E8B9B9C9B1D3D3E7E6D3D3D3D3*****                    +
 58  [illegible]                                                              +
```

Figure 3. A Digitized Map

TABLE VI. ACTIVITY INFORMATION FOR CLACKAMAS PLANNING UNIT

DEVELOPED RECREATION (D - LAND MANAGEMENT CATEGORY)

Site Identification

Identifiers: (Also RAU Criteria)	D-2
Unroaded Area	Not applicable
Soils	Slight erosion potential
Ecoclass	Excludes meadows, wetlands, & RARE communities
Geologic Conditions	Stable Conditions
RAM Site Class	Non-limiting
Physiography	Less the 30% slope
Accessibility	(see transportation)

Activity Description

Subactivities	D-2
Recreation Experience	Level 2
Facilities	Rustic (Pit toilets, shelters, Rock Fire Rings, Rustic Tables)
Timber Mgt.	Extended rotation; regulated, special
Visual Sensitivity	Level 2
Fire Mgt.	Less than 100% disposal (80%)
Transportation	Primitive roads limited maintenance, unsurfaced, level
Range Mgt.	Exclude
Wildlife	Harassment Level moderate

appear on the activity assignment map, rather than being lumped into an unassigned acreage category.

The land type map provided by the Planning Staff contained the physical characteristics which would limit or prohibit assignment of all twenty-seven activities. Since only six representative activities were selected, the land type map was simplified by aggregating those physical characteristics in the map which do not limit or prohibit the selected activities. The selection of six representative activities and subsequent modifications to the land type map simplifies the spatial allocation problem. This simplification makes observation of the procedure easier and the case study more clear.

Two activities, developed recreation and visual timber, are sensitive to the location of roads on the planning unit. However, the land type map did not contain any information about roads and access. A simple road system was added to the land type map so the spatial allocation procedure could demonstrate the way in which it assigns activities relative to the roads.

Both modifications mentioned were accomplished using the computer mapping system. After identifying the physical characteristics which did not limit assignment of the six activities, the subroutine AGGASS was used to aggregate those land types with others. The road system was added to the land type map with the subroutine

MANIP which singled out all the cells forming the road corridor.

The Mt. Hood Planning Staff was unable to provide the linear program acreage allocations because they had not reached that point in their own planning process. Instead of waiting until the allocations became available, a dummy set of acreage allocations was created. These dummy acreage allocations were devised so as to keep the case study simple while completely demonstrating the spatial allocation procedure. The acreate allocations used for this case study are presented in Table VII.

Construction of Management Units

The first step in the construction of management units is the elimination of isolated single cells by the detail reduction program. As stated before, the Clackamas Planning Unit contained 1,944 isolated single cells. The detail reduction program, REDUCE, grouped 1,916 of these cells with adjacent cells. The remaining 28 isolated single cells belong to lakes and streams which the program was instructed to ignore. The land type map produced at this point in the procedure is displayed as Map 1 in the accompanying map packet.

The next step involved breaking the acreage of each land type into management units. The subroutine PARCEL in the mapping program carried out this step automatically, creating a

TABLE VII. ACREAGE ALLOCATIONS

LAND TYPES	ACTIVITIES						
	RESERVED SITES	LAKES AND STREAMS	DEVELOPED RECREATION	DISPERSED RECREATION	COMMERCIAL TIMBER	VISUAL TIMBER	WILDLIFE
LAND TYPE 1	0.0	0.0	3178.0	0.0	26236.0	0.0	0.0
LAND TYPE 2	0.0	0.0	0.0	0.0	3327.0	0.0	0.0
LAND TYPE 3	0.0	0.0	0.0	0.0	14078.0	0.0	0.0
LAND TYPE 4	0.0	0.0	096.0	0.0	2133.0	0.0	0.0
LAND TYPE 5	0.0	0.0	0.0	0.0	533.0	0.0	0.0
LAND TYPE 6	0.0	0.0	1450.0	0.0	640.0	0.0	1493.0
LAND TYPE 7	0.0	0.0	5823.0	0.0	71456.0	0.0	27729.0
LAND TYPE 8	0.0	0.0	0.0	0.0	3754.0	0.0	0.0
LAND TYPE 9	0.0	0.0	0.0	0.0	88818.0	0.0	0.0
LAND TYPE 10	0.0	0.0	0.0	0.0	0.0	0.0	619.0
LAND TYPE 11	0.0	0.0	0.0	0.0	2026.0	0.0	0.0
LAND TYPE 12	0.0	0.0	0.0	0.0	747.0	0.0	0.0
LAND TYPE 13	0.0	0.0	0.0	0.0	363.0	0.0	0.0
LAND TYPE 14	0.0	0.0	0.0	0.0	0.0	0.0	8255.0
LAND TYPE 15	747.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 16	0.0	1130.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 17	6164.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 18	0.0	0.0	0.0	3882.0	0.0	0.0	0.0
LAND TYPE 19	0.0	0.0	0.0	3690.0	0.0	0.0	0.0
LAND TYPE 20	0.0	0.0	0.0	4458.0	0.0	0.0	0.0
LAND TYPE 21	0.0	0.0	0.0	11412.0	0.0	0.0	0.0
LAND TYPE 22	0.0	0.0	0.0	33168.0	0.0	0.0	0.0
LAND TYPE 23	0.0	0.0	0.0	12457.0	0.0	0.0	0.0
LAND TYPE 24	0.0	0.0	0.0	0.0	0.0	0.0	1173.0
LAND TYPE 25	0.0	0.0	107.0	0.0	0.0	107.0	0.0
LAND TYPE 28	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 29	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 30	0.0	0.0	3178.0	0.0	0.0	2133.0	0.0
LAND TYPE 31	0.0	0.0	0.0	0.0	0.0	64.0	0.0
LAND TYPE 32	0.0	0.0	0.0	0.0	0.0	3050.0	0.0
LAND TYPE 35	0.0	0.0	0.0	171.0	0.0	0.0	0.0
LAND TYPE 36	491.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 37	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 38	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 39	0.0	0.0	0.0	277.0	0.0	0.0	0.0

map with 1,292 management units. The legend of this map was run through subroutine UPDATE, reducing the original six lines of legend down to a single line indicating the land type number. This reduction of the legend saves storage space and processing cost in subsequent map operations. A portion of the original legend with six lines of description is shown in Figure 4. The management unit map with the reduced legend is Map 2 in the map packet.

No attempts were made to modify the size or shape of the resultant management units with the subroutine MANIP. The management unit map produced by the PARCEL subroutine was input directly into the activity assignment stage.

Construction of the Spillover Matrix

The activities being assigned by the spatial allocation procedure were examined and given the spillover values indicated in Figure 5. Examples of how to interpret the spillover matrix follow.

Select two activities, for example, commercial timber and developed recreation. The spillover value of -3 indicates the occurrence of a very severe undesirable impact when these activities are assigned to adjacent management units. The spillover value for developed recreation and lakes and streams is a +3,

MAP 2 PARCELS - MANAGEMENT UNIT MAP
LAND TYPES
ROAD MAP
ECOCLASS MAP
SOIL GROUP MAP
SLOPE MAP
SITE INDEX MAP

SYMBOL	NO. CELLS	ACREAGE	PCT.	LEGEND
A1	30	639.9	.2	LAND TYPE 1 UNROADED TRUE FIR SOIL GROUPS AA,BB,CC,DD,EE LESS THAN 30 PERCENT SLOPE SITE INDEX - NOT APPLICABLE OR AVAILABLE
A2	4	85.3	0.0	LAND TYPE 1 UNROADED TRUE FIR SOIL GROUPS AA,BB,CC,DD,EE LESS THAN 30 PERCENT SLOPE SITE INDEX - NOT APPLICABLE OR AVAILABLE
A3	5	106.7	0.0	LAND TYPE 1 UNROADED TRUE FIR SOIL GROUPS AA,BB,CC,DD,EE LESS THAN 30 PERCENT SLOPE SITE INDEX - NOT APPLICABLE OR AVAILABLE
A4	7	149.3	0.0	LAND TYPE 1 UNROADED TRUE FIR SOIL GROUPS AA,BB,CC,DD,EE LESS THAN 30 PERCENT SLOPE SITE INDEX - NOT APPLICABLE OR AVAILABLE
A5	11	234.6	.1	LAND TYPE 1 UNROADED TRUE FIR SOIL GROUPS AA,BB,CC,DD,EE LESS THAN 30 PERCENT SLOPE SITE INDEX - NOT APPLICABLE OR AVAILABLE
A6	3	64.0	0.0	LAND TYPE 1 UNROADED TRUE FIR SOIL GROUPS AA,BB,CC,DD,EE LESS THAN 30 PERCENT SLOPE SITE INDEX - NOT APPLICABLE OR AVAILABLE
A7	11	234.6	.1	LAND TYPE 1 UNROADED TRUE FIR SOIL GROUPS AA,BB,CC,DD,EE LESS THAN 30 PERCENT SLOPE SITE INDEX - NOT APPLICABLE OR AVAILABLE
A8	14	298.6	.1	LAND TYPE 1 UNROADED TRUE FIR SOIL GROUPS AA,BB,CC,DD,EE LESS THAN 30 PERCENT SLOPE SITE INDEX - NOT APPLICABLE OR AVAILABLE
A9	3	64.0	0.0	LAND TYPE 1 UNROADED TRUE FIR SOIL GROUPS AA,BB,CC,DD,EE LESS THAN 30 PERCENT SLOPE SITE INDEX - NOT APPLICABLE OR AVAILABLE
A0	14	298.6	.1	LAND TYPE 1 UNROADED TRUE FIR SOIL GROUPS AA,BB,CC,DD,EE

Figure 4. The Original Expanded Legend

indicating the beneficial impact produced when these activities are located adjacent to each other. Wildlife and dispersed recreation have a spillover value of 0. This value indicates a neutral impact when these activities are assigned to adjacent management units.

ACTIVITIES	1	2	3	4	5	6	7
1	0	0	0	0	0	0	0
2	0	0	+3	+3	-3	-2	0
3	0	+3	0	0	-3	-1	-3
4	0	+3	0	+1	-2	0	0
5	0	-3	-3	-2	+1	0	-1
6	0	-2	-1	0	0	+1	0
7	0	0	-3	0	-1	0	0

ACTIVITIES

1 - Reserved Sites

2 - Lakes and Streams

3 - Developed Recreation

4 - Dispersed Recreation

5 - Commercial Timber

6 - Visual Timber

7 - Wildlife

Figure 5. Spillover Matrix for Case Study

Assignment of Activities

In this case study, no activities were preassigned to management units. This allowed the heuristic program to assign activities to all of the management units without influencing the assignment procedure through an initialization process.

The first step of the assignment procedure, therefore, was to reorder the management units with the subroutine REORDER. Since every attempt is being made to avoid influencing the assignment procedure, the order of management units was based on decreasing acreage only with no single management unit picked out and placed in a special assignment position. The reordered management unit map is Map 3 in the map packet.

The list of adjacent management units is created by running the reordered management unit map through the adjacency program, ADJ.

The inputs to the heuristic assignment program, HEURIST, include: 1) the reordered management unit map, 2) the list of adjacent management units, 3) the linear program allocations, and 4) the spillover matrix. In the first run through the heuristic program, with all constraints in effect, 618 management units (or 47.8 percent of the management units) were assigned, accounting for 250,906 acres of the planning unit (or 72.6 percent of the acreage). The second run, which relaxed one constraint, assigned activities to all of the remaining management units but one. The third run, with two constraints relaxed, then assigned this last management unit. The allocation achievement tables for these three runs are shown in Tables VIII through X. The acreages presented in these tables represent how well that assignment meets the linear program acreage allocations. A positive number means the algorithm assigned that many acres over the allocation level. Negative numbers indicate the number of acres which need to be assigned to meet the allocations.

The subroutine AGGASS was used to transfer the activity assignments of each run to the management unit map. The first and third activity assignment maps were displayed by subroutine OUT and are shown in the map packet as Maps 4 and 5.

Detrimental Spillovers and Minimum Management Unit Size

Both strategies to detect detrimental spillovers were implemented. A portion of the information provided by each strategy is shown. The first strategy, which emphasizes information about the management units involved, requires no additional data manipulations, only an input of previous results. The information from this strategy is shown in Figure 6. The second strategy, emphasizing the "true" acreage assigned, requires

TABLE VIII. ALLOCATION ACHIEVEMENT TABLE
(No Constraints Relaxed)

LAND TYPES	ACTIVITIES						
	RESERVED SITES	LAKES AND STREAMS	DEVELOPED RECREATION	DISPERSED RECREATION	COMMERCIAL TIMBER	VISUAL TIMBER	WILDLIFE
LAND TYPE 1	0.0	0.0	64.2	0.0	-361.9	0.0	0.0
LAND TYPE 2	0.0	0.0	0.0	0.0	-255.0	0.0	0.0
LAND TYPE 3	0.0	0.0	0.0	0.0	-767.0	0.0	0.0
LAND TYPE 4	0.0	0.0	-277.3	0.0	21.5	0.0	0.0
LAND TYPE 5	0.0	0.0	0.0	0.0	-42.4	0.0	0.0
LAND TYPE 6	0.0	0.0	-1279.3	0.0	21.3	0.0	-1215.7
LAND TYPE 7	0.0	0.0	64.1	0.0	874.2	0.0	-23526.7
LAND TYPE 8	0.0	0.0	0.0	0.0	-511.4	0.0	0.0
LAND TYPE 9	0.0	0.0	0.0	0.0	-1363.5	0.0	0.0
LAND TYPE 10	0.0	0.0	0.0	0.0	0.0	0.0	-512.3
LAND TYPE 11	0.0	0.0	0.0	0.0	-810.0	0.0	0.0
LAND TYPE 12	0.0	0.0	0.0	0.0	-85.0	0.0	0.0
LAND TYPE 13	0.0	0.0	0.0	0.0	-.3	0.0	0.0
LAND TYPE 14	0.0	0.0	0.0	0.0	0.0	0.0	-7913.7
LAND TYPE 15	-.3	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 16	0.0	-447.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 17	.6	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 18	0.0	0.0	0.0	-1172.7	0.0	0.0	0.0
LAND TYPE 19	0.0	0.0	0.0	-2239.3	0.0	0.0	0.0
LAND TYPE 20	0.0	0.0	0.0	-4458.0	0.0	0.0	0.0
LAND TYPE 21	0.0	0.0	0.0	-6569.7	0.0	0.0	0.0
LAND TYPE 22	0.0	0.0	0.0	-30885.4	0.0	0.0	0.0
LAND TYPE 23	0.0	0.0	0.0	-6014.9	0.0	0.0	0.0
LAND TYPE 24	0.0	0.0	0.0	0.0	0.0	0.0	-1173.0
LAND TYPE 25	0.0	0.0	-107.0	0.0	0.0	-21.7	0.0
LAND TYPE 28	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 29	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 30	0.0	0.0	-3178.0	0.0	0.0	42.6	0.0
LAND TYPE 31	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 32	0.0	0.0	0.0	0.0	0.0	.5	0.0
LAND TYPE 35	0.0	0.0	0.0	-120.3	0.0	0.0	0.0
LAND TYPE 36	-.4	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 37	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 38	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 39	0.0	0.0	0.0	-277.0	0.0	0.0	0.0

TABLE IX. ALLOCATION ACHIEVEMENT TABLE
(One Constraint Relaxed)

LAND TYPES	RESERVED SITES	LAKES AND STREAMS	DEVELOPED RECREATION	DISPERSED RECREATION	COMMERCIAL TIMBER	VISUAL TIMBER	WILDLIFE
	ACTIVITIES						
LAND TYPE 1	0.0	0.0	64.2	0.0	-63.1	0.0	0.0
LAND TYPE 2	0.0	0.0	0.0	0.0	1.0	0.0	0.0
LAND TYPE 3	0.0	0.0	0.0	0.0	1.0	0.0	0.0
LAND TYPE 4	0.0	0.0	-21.1	0.0	21.5	0.0	0.0
LAND TYPE 5	0.0	0.0	0.0	0.0	.3	0.0	0.0
LAND TYPE 6	0.0	0.0	21.8	0.0	21.3	0.0	-42.3
LAND TYPE 7	0.0	0.0	64.1	0.0	874.2	0.0	-936.0
LAND TYPE 8	0.0	0.0	0.0	0.0	.5	0.0	0.0
LAND TYPE 9	0.0	0.0	0.0	0.0	1.0	0.0	0.0
LAND TYPE 10	0.0	0.0	0.0	0.0	0.0	0.0	-.4
LAND TYPE 11	0.0	0.0	0.0	0.0	.6	0.0	0.0
LAND TYPE 12	0.0	0.0	0.0	0.0	-.5	0.0	0.0
LAND TYPE 13	0.0	0.0	0.0	0.0	-.3	0.0	0.0
LAND TYPE 14	0.0	0.0	0.0	0.0	0.0	0.0	1.2
LAND TYPE 15	-.3	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 16	0.0	-.2	0.0	0.0	0.0	0.0	0.0
LAND TYPE 17	.6	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 18	0.0	0.0	0.0	.8	0.0	0.0	0.0
LAND TYPE 19	0.0	0.0	0.0	.3	0.0	0.0	0.0
LAND TYPE 20	0.0	0.0	0.0	.3	0.0	0.0	0.0
LAND TYPE 21	0.0	0.0	0.0	1.3	0.0	0.0	0.0
LAND TYPE 22	0.0	0.0	0.0	2.2	0.0	0.0	0.0
LAND TYPE 23	0.0	0.0	0.0	.9	0.0	0.0	0.0
LAND TYPE 24	0.0	0.0	0.0	0.0	0.0	0.0	.3
LAND TYPE 25	0.0	0.0	-107.0	128.0	0.0	-21.7	0.0
LAND TYPE 28	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 29	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 30	0.0	0.0	-42.2	0.0	0.0	42.6	0.0
LAND TYPE 31	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 32	0.0	0.0	0.0	0.0	0.0	.5	0.0
LAND TYPE 35	0.0	0.0	0.0	-.3	0.0	0.0	0.0
LAND TYPE 36	-.4	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 37	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 38	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 39	0.0	0.0	0.0	.3	0.0	0.0	0.0

TABLE X. ALLOCATION ACHIEVEMENT TABLE
(Two Constraints Relaxed)

LAND TYPES	ACTIVITIES						
	RESERVED SITES	LAKES AND STREAMS	DEVELOPED RECREATION	DISPERSED RECREATION	COMMERCIAL TIMBER	VISUAL TIMBER	WILDLIFE
LAND TYPE 1	0.0	0.0	64.2	0.0	-63.1	0.0	0.0
LAND TYPE 2	0.0	0.0	0.0	0.0	1.0	0.0	0.0
LAND TYPE 3	0.0	0.0	0.0	0.0	1.0	0.0	0.0
LAND TYPE 4	0.0	0.0	-21.1	0.0	21.5	0.0	0.0
LAND TYPE 5	0.0	0.0	0.0	0.0	.3	0.0	0.0
LAND TYPE 6	0.0	0.0	21.0	0.0	21.3	0.0	-42.3
LAND TYPE 7	0.0	0.0	64.1	0.0	874.2	0.0	-936.0
LAND TYPE 8	0.0	0.0	0.0	0.0	.5	0.0	0.0
LAND TYPE 9	0.0	0.0	0.0	0.0	1.0	0.0	0.0
LAND TYPE 10	0.0	0.0	0.0	0.0	0.0	0.0	-.4
LAND TYPE 11	0.0	0.0	0.0	0.0	.6	0.0	0.0
LAND TYPE 12	0.0	0.0	0.0	0.0	-.5	0.0	0.0
LAND TYPE 13	0.0	0.0	0.0	0.0	-.3	0.0	0.0
LAND TYPE 14	0.0	0.0	0.0	0.0	0.0	0.0	1.2
LAND TYPE 15	-.3	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 16	0.0	-.2	0.0	0.0	0.0	0.0	0.0
LAND TYPE 17	.6	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 18	0.0	0.0	0.0	.8	0.0	0.0	0.0
LAND TYPE 19	0.0	0.0	0.0	.3	0.0	0.0	0.0
LAND TYPE 20	0.0	0.0	0.0	.3	0.0	0.0	0.0
LAND TYPE 21	0.0	0.0	0.0	1.3	0.0	0.0	0.0
LAND TYPE 22	0.0	0.0	0.0	2.2	0.0	0.0	0.0
LAND TYPE 23	0.0	0.0	0.0	.9	0.0	0.0	0.0
LAND TYPE 24	0.0	0.0	0.0	0.0	0.0	0.0	.3
LAND TYPE 25	0.0	0.0	-107.0	0.0	0.0	-21.7	0.0
LAND TYPE 28	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 29	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 30	0.0	0.0	-42.2	0.0	0.0	42.6	0.0
LAND TYPE 31	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 32	0.0	0.0	0.0	0.0	0.0	.5	0.0
LAND TYPE 35	0.0	0.0	0.0	-.3	0.0	0.0	0.0
LAND TYPE 36	-.4	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 37	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 38	0.0	0.0	0.0	0.0	0.0	0.0	0.0
LAND TYPE 39	0.0	0.0	0.0	.3	0.0	0.0	0.0

```
DETECTION OF DETRIMENTAL SPILLOVERS

   ASSIGNMENT - REORDERED MANAGEMENT UNIT MAP
       LAND TYPES

DETRIMENTAL SPILLOVERS WITH SEVERITY 3

SEVERITY = 3      BOUNDARY LENGTH =      1
     MANAGEMENT UNIT AD (  10)     DEVELOPED RECREATION
                220 CELLS         4692.6 ACRES          1.3 PERCENT
                      LAND TYPE 7
     MANAGEMENT UNIT TE ( 759)     COMMERCIAL TIMBER
                  3 CELLS           64.0 ACRES          0.0 PERCENT
                      LAND TYPE 2

SEVERITY = 3      BOUNDARY LENGTH =     19
     MANAGEMENT UNIT AD (  10)     DEVELOPED RECREATION
                220 CELLS         4692.6 ACRES          1.3 PERCENT
                      LAND TYPE 7
     MANAGEMENT UNIT 3D (  20)     COMMERCIAL TIMBER
                133 CELLS         2836.9 ACRES           .8 PERCENT
                      LAND TYPE 1

SEVERITY = 3      BOUNDARY LENGTH =      1
     MANAGEMENT UNIT AD (  10)     DEVELOPED RECREATION
                220 CELLS         4692.6 ACRES          1.3 PERCENT
                      LAND TYPE 7
     MANAGEMENT UNIT 33 (  23)     COMMERCIAL TIMBER
                118 CELLS         2516.9 ACRES           .7 PERCENT
                      LAND TYPE 9

SEVERITY = 3      BOUNDARY LENGTH =      4
     MANAGEMENT UNIT AD (  10)     DEVELOPED RECREATION
                220 CELLS         4692.6 ACRES          1.3 PERCENT
                      LAND TYPE 7
     MANAGEMENT UNIT FC ( 393)     COMMERCIAL TIMBER
                  7 CELLS          149.3 ACRES          0.0 PERCENT
                      LAND TYPE 9

SEVERITY = 3      BOUNDARY LENGTH =      2
     MANAGEMENT UNIT AD (  10)     DEVELOPED RECREATION
                220 CELLS         4692.6 ACRES          1.3 PERCENT
                      LAND TYPE 7
     MANAGEMENT UNIT TW ( 777)     COMMERCIAL TIMBER
                  3 CELLS           64.0 ACRES          0.0 PERCENT
                      LAND TYPE 4
```

Figure 6. Detrimental Spillovers
(Strategy One)

additional data manipulation. The activity assignment map is input and subroutine PARCEL breaks it into a new set of management units. The program ADJ detects adjacent management units in this map and all this information is input into the conflict detection program, DETECT. Some of the information generated by this strategy is shown in Figure 7.

Violations of the minimum management unit size were detected in the same run with the second detrimental spillover strategy. Detection of minimum-size violations requires the inputs in that form plus minimum acreages against which the management units are screened. Some of the results of this procedure are shown in Figure 8.

Discussion of Results

The results of the case study can be examined to determine whether or not the spatial allocation technique developed is worthwhile. Tests can be constructed to determine: a) how close the acreage assignments come to the linear program allocations, b) whether or not the algorithm creates the pattern it is directed to in the spillover matrix, and c) how successful the heuristic is in preventing adjacent-use conflicts. Each of these indicators is examined for the purpose of measuring the performance of the spatial allocation strategy and of learning what value it may be

```
DETECTION OF DETRIMENTAL SPILLOVERS

   PARCELS - ACTIVITY ASSIGNMENTS - NO CONSTRAINTS RELAXED

DETRIMENTAL SPILLOVERS WITH SEVERITY 2

SEVERITY = 2      BOUNDARY LENGTH =       3

     MANAGEMENT UNIT F0 (   60)      DISPERSED RECREATION
                 20 CELLS            426.6 ACRES          .2 PERCENT

     MANAGEMENT UNIT J0 ( 100)       COMMERCIAL TIMBER
               2230 CELLS          47565.9 ACRES       18.4 PERCENT

SEVERITY = 2      BOUNDARY LENGTH =       2

     MANAGEMENT UNIT G1 (   61)      DISPERSED RECREATION
                  4 CELLS             85.3 ACRES         0.0 PERCENT

     MANAGEMENT UNIT J0 ( 100)       COMMERCIAL TIMBER
               2230 CELLS          47565.9 ACRES       18.4 PERCENT

SEVERITY = 2      BOUNDARY LENGTH =       1

     MANAGEMENT UNIT G3 (   63)      DISPERSED RECREATION
                 11 CELLS            234.6 ACRES          .1 PERCENT

     MANAGEMENT UNIT J9 (   99)      COMMERCIAL TIMBER
                822 CELLS          17533.3 ACRES        6.8 PERCENT

SEVERITY = 2      BOUNDARY LENGTH =       3

     MANAGEMENT UNIT H4 (   74)      DISPERSED RECREATION
                 15 CELLS            320.0 ACRES          .1 PERCENT

     MANAGEMENT UNIT O5 ( 145)       COMMERCIAL TIMBER
                  9 CELLS            192.0 ACRES          .1 PERCENT

SEVERITY = 2      BOUNDARY LENGTH =       8

     MANAGEMENT UNIT H6 (   76)      DISPERSED RECREATION
                 29 CELLS            618.6 ACRES          .2 PERCENT

     MANAGEMENT UNIT O6 ( 146)       COMMERCIAL TIMBER
                226 CELLS           4820.6 ACRES        1.9 PERCENT

SEVERITY = 2      BOUNDARY LENGTH =       8

     MANAGEMENT UNIT H9 (   79)      DISPERSED RECREATION
                 14 CELLS            298.6 ACRES          .1 PERCENT

     MANAGEMENT UNIT N5 ( 135)       COMMERCIAL TIMBER
               2139 CELLS          45624.9 ACRES       17.7 PERCENT
```

Figure 7. Detrimental Spillovers (Strategy Two)

```
DETECTION OF MANAGEMENT UNIT SIZE VIOLATIONS

   PARCELS - ACTIVITY ASSIGNMENTS - 2 CONSTRAINTS RELAXED

MINIMUM ACREAGE    ACTIVITY
      10.0          RESERVED SITES
      10.0          LAKES AND STREAMS
      25.0          DEVELOPED RECREATION
     250.0          DISPERSED RECREATION
     250.0          COMMERCIAL TIMBER
     100.0          VISUAL TIMBER
     200.0          WILDLIFE

    MANAGEMENT UNIT K5 ( 105)        DISPERSED RECREATION
                 2 CELLS             42.7 ACRES          0.0 PERCENT
      ADJACENT MANAGEMENT UNITS AND THEIR ASSIGNED ACTIVITIES
        DA ( 339)  COMMERCIAL TIMBER

    MANAGEMENT UNIT K6 ( 106)        DISPERSED RECREATION
                 2 CELLS             42.7 ACRES          0.0 PERCENT
      ADJACENT MANAGEMENT UNITS AND THEIR ASSIGNED ACTIVITIES
        HH ( 450)  WILDLIFE
        CW ( 335)  COMMERCIAL TIMBER

    MANAGEMENT UNIT CX ( 336)        COMMERCIAL TIMBER
                 3 CELLS             64.0 ACRES          0.0 PERCENT
      ADJACENT MANAGEMENT UNITS AND THEIR ASSIGNED ACTIVITIES
        K2 ( 102)  DISPERSED RECREATION
        A3 (   3)  RESERVED SITES
        A4 (   4)  RESERVED SITES
        K3 ( 103)  DISPERSED RECREATION

    MANAGEMENT UNIT CY ( 337)        COMMERCIAL TIMBER
                 3 CELLS             64.0 ACRES          0.0 PERCENT
      ADJACENT MANAGEMENT UNITS AND THEIR ASSIGNED ACTIVITIES
        K3 ( 103)  DISPERSED RECREATION

    MANAGEMENT UNIT GA ( 417)        COMMERCIAL TIMBER
                 6 CELLS            128.0 ACRES          0.0 PERCENT
      ADJACENT MANAGEMENT UNITS AND THEIR ASSIGNED ACTIVITIES
        CP ( 328)  DISPERSED RECREATION
        CU ( 333)  DISPERSED RECREATION

    MANAGEMENT UNIT GE ( 421)        VISUAL TIMBER
                 3 CELLS             64.0 ACRES          0.0 PERCENT
      ADJACENT MANAGEMENT UNITS AND THEIR ASSIGNED ACTIVITIES
        D9 ( 149)  DISPERSED RECREATION
        DO ( 353)  COMMERCIAL TIMBER
        DS ( 357)  COMMERCIAL TIMBER
        G9 (  69)  DEVELOPED RECREATION

    MANAGEMENT UNIT GH ( 424)        VISUAL TIMBER
                 2 CELLS             42.7 ACRES          0.0 PERCENT
      ADJACENT MANAGEMENT UNITS AND THEIR ASSIGNED ACTIVITIES
        RO ( 180)  DISPERSED RECREATION
        KQ ( 537)  WILDLIFE

    MANAGEMENT UNIT HD ( 446)        WILDLIFE
                 2 CELLS             42.7 ACRES          0.0 PERCENT
      ADJACENT MANAGEMENT UNITS AND THEIR ASSIGNED ACTIVITIES
        CV ( 334)  COMMERCIAL TIMBER

    MANAGEMENT UNIT HE ( 447)        WILDLIFE
                 6 CELLS            128.0 ACRES          0.0 PERCENT
      ADJACENT MANAGEMENT UNITS AND THEIR ASSIGNED ACTIVITIES
        K3 ( 103)  DISPERSED RECREATION
        CW ( 335)  COMMERCIAL TIMBER

    MANAGEMENT UNIT HF ( 448)        WILDLIFE
                 7 CELLS            149.3 ACRES          0.0 PERCENT
      ADJACENT MANAGEMENT UNITS AND THEIR ASSIGNED ACTIVITIES
        CW ( 335)  COMMERCIAL TIMBER
```

Figure 8. Violations of Minimum Management Unit Size

to a planner.

The heuristic program produces an "Allocation Achievement Table." This table allows comparisons to be made between the acreage assigned by the heuristic algorithm and the acreage allocations produced by the linear program.

After completion of the activity assignments (with two constraints relaxed) in the Clackamas Planning Unit, the heuristic algorithm had assigned an activity to every management unit. Table X shows how close the assignments came to the original allocation levels. The assignments were 1,123.6 acres over and 1,235.8 acres under the linear program acreage allocations. (The difference occurs because the allocation acreages are not exactly equal to the planning unit acreage.) The combined acreage variations represent 0.667 percent of the total acreage to be assigned. There is, therefore, an extremely small difference between the linear program allocations and the heuristic allocations.

Development of the spillover matrix requires the planner to specify which land uses should be brought together and which kept apart from each other. These specifications can be treated as predictions for the purposes of testing the effectiveness of the heuristic algorithm. That is, one can compare the expected and actual locations to test the performance of the algorithm. Inspection of the adjacent conditions shown in the maps can be summarized as follows.

The assignment algorithm avoided placing timber activities on management units adjacent to lakes and streams when all of the constraints were in effect. However, in two cases, visual timber was assigned next to a stream which followed the road corridor. Fifteen adjacent commercial timber assignments were also made after the constraints were relaxed.

The algorithm grouped 69 percent of the total developed recreation allocation into four management units, each containing more than 1,000 acres. This action was taken by the algorithm in an attempt to keep developed recreation away from timber activities and wildlife. This was an unexpected result, but satisfied the instructions given to the algorithm.

The dispersed recreation assignments all were made as expected. Large areas were created near lakes and streams, away from roads and timber activities.

The algorithm assigned the majority of the commercial timber units as expected with all the constraints in effect. The linear program acreage allocations fixed visual timber activities to the road corridor as expected. No visual timber assignments appeared elsewhere in the map.

Wildlife areas were scattered across the planning unit adjacent to roads, developed recreation, and commercial timber.

Wildlife management units generally were not grouped together, as might be necessary to create range for large game animals.

Comparison of the expected pattern with the pattern produced by the heuristic algorithm produces two major differences. These differences are associated with developed recreation and wildlife. First, the acreage assigned to developed recreation appeared in large blocks away from the road system. The algorithm grouped developed recreation units together to offset the conflicts resulting when these are assigned next to commercial timber, visual timber, or wildlife units. The conflict with visual timber (which is associated with the road corridor) forced developed recreation units to be located away from the road.

This result could be avoided by changing the value assigned to visual timber and developed recreation in the spillover matrix or by preassigning the most suitable areas of the planning unit to developed recreation. The algorithm would then fill in around these assignments.

Wildlife units were scattered around the planning unit and made adjacent, in many places, to activities which created negative spillovers. There seem to be two reasons for this result. One reason may be that commercial timber assignments were made first by the algorithm since these units were to be allocated to large management units. This left smaller acreage openings into which wildlife could be fit to satisfy the allocation goal.

Wildlife assignments also were made without consideration of roads. Wildlife could be made sensitive to road location by placing roads in a special category similar to lakes and streams. A negative value then could be given to the adjacent location of roads and wildlife in the spillover matrix. An alternative tactic would be to assign negative spillover values to the adjacent location of wildlife and those activities associated with the road corridor (i.e., visual timber and developed recreation).

Success of the heuristic algorithm in preventing adjacent-use conflicts also can be determined by comparing the number of expected conflicts in a random assignment process with the number actually produced. This comparison was made for assignments with no constraints relaxed (Map 1) and assignments with two constraints relaxed (Map 3). The actual number of adjacent-use conflicts produced in each case was determined from a tally of the detrimental spillovers listed by the conflict detection program. Map 1 contained 41 detrimental spillovers while Map 3 contained 1,551.

The expected number of conflicts produced in a random assignment process was determined by weighting the values in the spillover matrix with the percentage of acreage assigned to each activity (Tables XI through XIII show the computations for this process). Table XI shows the percentage of acreage assigned to each activity in Maps 1 and 3. Using these percentages as weights on the expected occurrence of adjacent locations, computations of these expected values were constructed for Maps 1 and 3 (Tables XII and XIII). The values in parentheses are linked to the negative values in the spillover matrix (Figure 5) and, when totalled, represent the total percentage of adjacent conditions resulting in conflicts.

The calculated percentages are 11.37 for Map 1 and 44.31 for Map 3. The total number of adjacent conditions (6,120) is obtained from the adjacency program. Multiplication of the total number of adjacent conditions by the percentages produces the number of conflicts which can be expected from random assignment of acreage equivalent to that shown in Maps 1 and 3. The expected number of conflicts are 696 for Map 1 and 2,712 for Map 3. Comparison of these values with the actual number of conflicts (41 and 1,551) indicate that the spatial allocation strategy performs better than a random assignment procedure.

TABLE XI

ACREAGE BREAKDOWN BETWEEN ACTIVITIES

Activities	Percentages	
	Map 1	Map 2
1 - Reserved Sites	2.1	2.1
2 - Lakes and Streams	0.2	0.3
3 - Developed Recreation	2.8	4.1
4 - Dispersed Recreation	5.0	20.4
5 - Commercial Timber	59.6	60.8
6 - Visual Timber	1.5	1.5
7 - Wildlife	1.4	10.8

TABLE XII. EXPECTED VALUES FOR MAP 1

ACTIVITIES		1	2	3	4	5	6	7
ACTIVITIES	1	0.044	0.004	0.059	0.105	1.252	0.032	0.029
	2	0.004	0.001	0.006	0.010	(0.119)	(0.003)	0.003
	3	0.059	0.006	0.078	0.140	(1.669)	(0.042)	(0.039)
	4	0.105	0.010	0.140	0.250	(2.980)	0.075	0.070
	5	1.252	(0.119)	(1.669)	(2.980)	35.522	0.894	(0.834)
	6	0.032	(0.003)	(0.042)	0.075	0.894	0.023	0.021
	7	0.029	0.003	(0.039)	0.070	(0.834)	0.021	0.020

TABLE XIII. EXPECTED VALUES FOR MAP 3

ACTIVITIES		1	2	3	4	5	6	7
ACTIVITIES	1	0.044	0.006	0.086	0.428	1.277	0.032	0.227
	2	0.006	0.001	0.012	0.061	(0.182)	(0.005)	0.032
	3	0.086	0.012	0.168	0.836	(2.493)	(0.062)	(0.443)
	4	0.428	0.061	0.836	4.162	(12.403)	0.306	2.203
	5	1.277	(0.182)	(2.493)	(12.403)	36.966	0.912	(6.566)
	6	0.032	(0.005)	(0.062)	0.306	0.912	0.023	0.162
	7	0.227	0.032	(0.443)	2.203	(6.566)	0.162	1.166

VI. FUTURE RESEARCH STEPS

The spatial allocation problem involves three important spatial factors: 1) management unit size, 2) adjacent-use conflicts, and 3) collocation patterns. Construction of strategies which optimize land allocation locations, when these factors are taken into account, generally are prohibited by the size of the allocation problem. Even when the problem is sub-divided and an efficiency criterion substituted for optimization, optimizing algorithms either exceed computer memory size or are prohibitively expensive. Assignment algorithms are more useful and are suitable for the objective of finding the most efficient approximation of the linear program solution.

Ignizio's (1978) heuristic technique, a search and assignment algorithm, can be modified so as to treat the three spatial factors as variables in the assignment problem. Before the technique can be used, the problem must be broken into several components, then tools developed to handle each component. The tools and procedures developed through this research are not completely satisfactory, although they are workable and make it possible to deal with spatial factors in a systematic manner.

Difficulties in Application

A number of difficulties are encountered when applying the spatial allocation procedure to an actual planning unit. These difficulties include: 1) the amount of detail, 2) computer core limitations, 3) algorithm efficiency, 4) the amount of information, 5) the number of interactions, and 6) the lack of exact rules for making tradeoffs. Each of the difficulties will be discussed in a general manner.

The amount of detail refers to the complexity of the planning unit. The diversity of land types plus their location creates a complex and detailed pattern. This complexity forces solution techniques away from mathematical programming algorithms (which can only handle a limited number of variables) toward algorithms with more flexible limits. It also strains computerized data manipulation routines, such as the mapping routine developing management units.

The computer core limitations refer to the amount of information which the computer can load and process at any one time. These limitations become important in problems where a large number of factors must be considered at the same time. Any mathematical programming technique attempting to include location in its allocations encounters this limitation. Several operations in the mapping program also face this problem. However, in the mapping program, this limitation is overcome by sub-dividing the problem. This strategy also can be applied to mathematical programming problems, but formulation is difficult and multiple runs make a final solution quite expensive.

Algorithm efficiency is reflected in the cost of obtaining a solution. The solution cost is affected by: 1) the amount of information considered by the algorithm, 2) the required precision of the solution, and 3) the number of runs necessary to calculate a solution. Mathematical programming algorithms considering location and factors associated with location in their allocations must sub-divide the problem into a number of smaller problems, each of which must be solved. This, coupled with the requirement of an "optimal" solution for each sub-problem, usually creates a solution with high costs. Cost also may be prohibitive for other types of algorithms depending on their design, their solution requirements, and the amount of information they process.

The amount of information included in an allocation or assignment algorithm must be examined in light of computer core limitations and solution cost. These factors sometimes require a simplification of one or more aspects of a particular problem. In the spatial allocation problem, the linear program acreage allocations and adjacent activity assignments were considered directly in the assignment process while minimum management unit size and collocation requirements were encouraged with the spillover matrix but only checked after assignments were made.

The number of interactions between the information examined when making assignments adds to the complexity of any assignment algorithm. When a tradeoff exists between two factors affecting the solution, the algorithm must consider this special case (and every other special case) or must have rules to avoid these complications. Consideration of every case will increase the amount of information to be examined, possibly causing computer core or cost problems.

The lack of exact rules occur in the case of interactions and tradeoffs. The problem arises because of the complexity of the spatial allocation process and the number of factors involved in development of a land use pattern. Attempts to anticipate every tradeoff and develop a rule for each are virtually impossible and create additional complexity in solution procedures. The best approach may be a simplification of the problem with a standard procedure which ignores tradeoffs.

Tool Improvements and Subjects for Further Investigation

This study has exposed the problem of spatial allocation in forest land use planning. There is room for improvement in the tools developed. The problem requires greater study in its own right and the heuristic technique is in need of reliability and efficiency testing.

The adjacency program, for example, is limited to working with only those management units sharing a common boundary. Examination of spillover impacts are restricted to those located along a boundary. The adjacency program, as a result, cannot account for the effects of a single activity assignment on the overall land use pattern. A useful improvement to the procedure would be a program which kept track of the relative location of all management unit assignments. This program would allow the proximity of other activities to be considered in each assignment and, thereby, allow the collocation factor to be treated directly in the procedure.

The spillover matrix, or the values in it, require further study. The values in the matrix represent the spillover effects expected if two activities were to be located on adjacent management units. The matrix is a useful device for making identification of spillovers and the judgments about their importance explicit. However, the values necessarily are a simplification of several factors. These include: 1) the type of effect expected from proximity of two activities, 2) the impact of this effect on production of consumer goods and services, 3) the sensitivity of each activity to the type of spillover, and 4) the relative importance of the types of spillovers identified for the matrix. Very little is known about these spillovers and their effects on production of goods and services from a planning unit. Even though explicit in the allocation process developed here, the value judgments in the matrix remain subjective.

A related task is to more thoroughly investigate the spatial factors involved in land use allocation on National Forests. Three factors were identified here. It seems unlikely that these are the only spatial factors or that they are the most important ones. It is not even certain that the three identified here are mutually exclusive factors. Further research into the identity of spatial factors and their influence on land use allocations seems warranted.

On another tack, planning problems vary in size and complexity. It would be useful to know how size of the problem affects: a) computational efficiency, b) the flexibility of assignment options mentioned earlier, and c) the sensitivity of the solution given by the heuristic technique. If the problem is very large, the planner will make simplifications in variables and reduce the level of detail in the units of analysis. Similarly, it will be important to know how such reductions, when made, are likely to affect the accuracy of the final solution.

The most troublesome shortcomings are in the heuristic technique. Since the allocation algorithm (or strategy) seems relatively independent of the kind of spatial factors involved in the problem, high priority should be placed on further study of this strategy.

The heuristic search algorithm looks to future assignment options and considers previous assignments when selecting an activity to assign to a management unit. However, the algorithm is unable to reassign management units even if such an action were determined to be the most beneficial to the final land use pattern.

The heuristic search algorithm could be modified so that it is able to reassign management units. This modification should permit an improvement in the land use patterns which is currently prevented by the algorithm.

The heuristic algorithm assigns activities to as many management units as possible without violating the values in the spillover matrix. To assign the remaining management units, the spillover constraint is subsequently relaxed and activities are assigned which fulfill the acreage allocations and minimize the detrimental spillovers. However, there is no assurance that a land use pattern would not have been created with fewer detrimental spillovers had this relaxation always been in effect.

Detrimental spillovers block assignment options in the heuristic search algorithm. There is probably a range of detrimental spillovers which is acceptable in the heuristic algorithm.

Outside of this range the spillovers disrupt the algorithm, permitting assignments to only a few isolated management units. The acceptable range of these spillovers would be useful information to a planner who may wish to select only the most severe conflicts in order to create a land use pattern.

The tradeoff between computational efficiency and assignment options caused by the number of management units should be investigated. There is a need to establish a range in the number of management units which fits the computer core limitations, can be manipulated with some degree of efficiency, and provides adequate flexibility in the assignment of activities to management units. The range would assist planners in determining the amount of detail the land type and management unit maps could contain, , if the heuristic assignment algorithm was being used in the spatial allocation procedure.

Finally, the reliability of the heuristic algorithm is unknown. It seems likely that the ability of the technique to produce the most efficient possible solution is affected by the number of variables in the problem and by the range in these variables. Tests should be conducted to determine how sensitive the algorithm is to: a) a change from the use of positive, negative, and zero values from the spillover matrix to the use of a full range of values from +3 to -3, b) variation in the spillover matrix values, and c) the order in which management units are scheduled for assignment.

BIBLIOGRAPHY

Ackoff, R. L., and M. W. Sasieni. 1968. Fundamentals of Operations Research. Wiley & Sons, Inc., New York. 455 p.

Bell, Enoch F. 1976. Goal Programming for Land Use Planning. USDA Forest Service General Technical Report PNW-53, Portland, Oregon. 12 p.

Child, Dennis R., and Linda S. Rollin. 1976. E-ZMAP: Computerized Map Overlay Procedure. Range Science Series No. 23. Colorado State University, Fort Collins, Colorado. 43 p.

Dougenik, James A., and David E. Sheehan. 1975. SYMAP User's Reference Manual. Harvard University, Cambridge, Mass. 187 p.

Geoffrion, A. R., and R. E. Marsten. 1972. Integer programming algorithms: a framework and state-of-the-art survey. Management Science 18:465-491

Hillier, Frederick S., and Gerald J. Lieberman. 1974. Operations Research. Holden-Day, Inc., San Francisco, Calif. 800 p.

Hirsch, W. Z. and S. Sonenblum. 1970. Selecting Regional Information for Government Planning and Decision-Making. Praeger Special Studies, N. Y. 198 p.

House, Peter W. 1976. The Quest for Completeness. D. C. Heath & Co., Lexington, Mass. 236 p.

Hufschmidt, Maynard M., ed. 1969. Regional Planning: Challenge and Prospects. Praeger Special Studies, New York. 396 p.

Ignizio, James P. 1978. Solving large scheduling problems by minimizing conflict. Simulation. March, 1978. pp. 75-79.

Phillips, D. T., A. Ravindran, and J. J. Solberg. 1976. Operations Research: Principles and Practice. Wiley & Sons, Inc., New York. 585 p.

Public Land Law Review Commission. 1970. One Third of the Nation's Land. Government Printing Office, Washington, D. C. 342 p.

Ramalingam, P. 1976. Systems Analysis for Managerial Decisions. Wiley & Sons, Inc., New York. 607 p.

Simmons, Donald M. 1972. Linear Programming for Operations Research. Holden-Day, Inc., San Francisco, Calif. 288 p.

Sinton, David F. 1976. An introduction to I. M. G. R. I. D.: an information manipulation system for grid cell data structures. Harvard Graduate School of Design, Dept. of Landscape Architecture, Cambridge, Mass. 23 p.

Taha, H. A. 1971. Operations Research: An Introduction. Macmillian Press, New York. 626 p.

Wagner, H. M. 1970. Principles of Management Science. Prentice-Hall, Inc., Englewood Cliffs, New Jersey. 562 p.

APPENDICES

APPENDIX A

USE OF INTEGER PROGRAMMING FOR MAKING ALLOCATIONS

Problem Formulation

i = activity

j = management unit

Objective Function[1]

$$\text{MAX } Z = WT_1(c_{11}X_{11} + c_{1j}X_{1j} + c_{21}X_{21} + \ldots + c_{ij}X_{ij}) - WT_2(d_1^+ + \ldots + d_m^+ + d_{m+1}^- + d_{m+1}^+ + \ldots + d_n^- + d_n^+)$$

subject to

Allocation Constraints[2]

$$a_{11}X_{11} + a_{12}X_{12} + \ldots + a_{1j}X_{1j} = b_1$$
$$a_{21}X_{21} + a_{22}X_{22} + \ldots + a_{2j}X_{2j} = b_2$$
$$\vdots$$
$$a_{i1}X_{i1} + a_{i2}X_{i2} + \ldots + a_{ij}X_{ij} = b_i$$

Adjacent Conflict Constraints[3]

$$X_{11} + X_{22} + d_1^- - d_1^+ = 1$$
$$\vdots$$
$$X_{..} + X_{..} + d_m^- - d_m^+ = 1$$

Management Unit Size Constraints[4]

$$X_{11} - X_{12} + d_{m+1}^- - d_{m+1}^+ = 0$$
$$\vdots$$
$$X_{..} - X_{..} + d_n^- - d_n^+ = 0$$

Multiple Choice Constraints[5]

$$X_{11} + X_{21} + \ldots + X_{i1} = 1$$
$$X_{12} + X_{22} + \ldots + X_{i2} = 1$$
$$\vdots$$
$$X_{1j} + X_{2j} + \ldots + X_{ij} = 1$$

Zero-one Constraints[6]

$$X_{fg} = 0 \text{ or } 1 \text{ for every } f = 1, \ldots, i \text{ and } g = 1, \ldots, j$$

[1] The objective function maximizes one goal or product while minimizing the adjacency conflicts and management unit size violations. The weights are used to indicate the relative importance of these two objectives.

[2] The allocation constraints are similar to those in the linear program formulation except each management unit is assigned in its entirety to a single activity.

[3] The conflict constraints prevent two adjacent management units from being assigned to conflicting activities. One constraint is required for each pair of management units where the conflicting situation may arise. The possible situations which can arise are:

1) 1 + 1 + 0 - 1 penalty
2) 0 + 0 + 1 - 0
3) 1 + 0 + 0 - 0
4) 0 + 1 + 0 - 0

Since only the d^+ variable is penalized in the objective function, only the first situation (where the conflicting activities are assigned to both management units) produces a penalty and is, therefore, avoided.

[4] The management unit size constraints require two adjacent management units to be assigned to the same activity. The possible situations are:

1) 1 - 1 + 0 - 0
2) 0 - 0 + 0 - 0
3) 1 - 0 + 0 - 1 penalty
4) 0 - 1 + 1 - 0 penalty

Since the same activity is to be assigned to both management units or neither, a penalty is attached to the situations where the activity is assigned to only one of the management units. This situation arises in the last two cases when either the d^- or d^+ variable takes on a value other than zero.

[5]The multiple choice constraints assure that each management unit will be assigned to one and only one activity.

[6]The zero-one constraints prevent the variables from taking on any values other than zero or one.

USE OF INTEGER PROGRAMMING TO ASSIGN ALLOCATIONS

Objective Function[1]

$$\text{MIN } Z = WT(d_1^+ + \ldots + d_m^- + d_{m+1}^- + d_{m+1}^+ + \ldots + d_n^- + d_n^+)$$

Allocation Constraints[2]

$$a_{i1}X_{i1} + a_{i2}X_{i2} + \ldots + a_{ij}X_{ij} = b_i$$

Conflict Constraint[3]

Management Unit Size Constraints[3]

Multiple Choice Constraints[3]

Zero-one Constraints[3]

[1]In this use of the objective function, the algorithm must simply minimize the spatial assignment penalties.

[2]The allocation constraints require the management units assigned to each activity to meet the acreage allocations. These constraints can also be formulated so that the allocations do not have to be met exactly but a penalty will be paid for deviations from the allocation levels.

[3]The other constraints are identical to their previous use in the integer program making actual allocations.

APPENDIX B

```
      OVERLAY(MAPS,0,0)
      PROGRAM MAIN(INPUT=65,OUTPUT=65,TAPE49=INPUT,TAPE50=OUTPUT,
     1TAPE1=65,TAPE2=65,TAPE3=65,TAPE4=65,TAPE5=65,
     2TAPE6=65,TAPE7=65,TAPE8=65,TAPE9=65,TAPE10=65,
     3TAPE11=65,TAPE12=65,TAPE13=65,TAPE14=65,TAPE15=65,
     4TAPE16=65,TAPE17=65,TAPE18=65,TAPE19=65,TAPE20=65,
     5TAPE21=65,TAPE22=65,TAPE23=65,TAPE24=65,TAPE25=65,
     6TAPE26=65,TAPE27=65,TAPE28=65,TAPE29=65,TAPE30=65,
     7TAPE31=65,TAPE32=65,TAPE33=65,TAPE34=65,TAPE35=65,
     8TAPE36=65,TAPE37=65,TAPE38=65,TAPE39=65,TAPE40=65,
     9TAPE43=65,TAPE44=65,TAPE45=65,
     ATAPE46=65,TAPE47=65,TAPE48=65)
      COMMON/A/MFILE,MF,LF
      COMMON/D/ISYMBL(2496)
      COMMON/E/KODE
      DATA MFILE/0/
C
C*****************************************************************
C
C   THIS PROGRAM IS A MAPPING ROUTINE WHICH CARRIES OUT A NUMBER OF
C   SPECIALIZED FUNCTIONS.  MAPS AND THEIR ASSOCIATED DOCUMENTATION
C   ARE INPUT INTO THE PROGRAM AND THE MAP SYMBOLS ARE RECODED INTO
C   A NUMERICAL CODE WHICH ALLOWS FURTHER OPERATIONS ON THE MAPS.
C
C   THESE OPERATIONS INCLUDE:
C       1) OUTPUT IN A STANDARDIZED FORM AT THE TERMINAL OR ON A
C          LINE PRINTER FILE,
C       2) COMBINATION OF MAPS TO CREATE A MAP WITH A NEW SET OF
C          SYMBOLS REPRESENTING EACH UNIQUE MAP COMBINATION,
C       3) AGGREGATION OF MAP SYMBOLS INTO ASSOCIATED GROUPS,
C       4) SYMBOL CHANGES FROM A MAP ASSIGNMENT,
C       5) REORDER MAP SYMBOLS AND LEGEND,
C       6) A DIRECTORY OF MAPS AVAILABLE TO THE USER,
C       7) HIGHLIGHT OF SOME CHARACTERISTICS OF THE DOCUMENTATION,
C       8) AUTOMATIC GROUPING OF ADJACENT CELLS OF THE SAME SYMBOL
C          INTO PARCELS,
C       9) UPDATE OF MAP TITLES AND LEGENDS,
C      10) MANIPULATION OF MAP CELLS.
C
C   THESE OPERATIONS WILL BE BETTER DESCRIBED IN EACH SUBROUTINE.
C   THIS PROGRAM OPERATES ON AN INTERACTIVE BASIS; THE USER RESPONDS
C   TO QUESTIONS AND IN MOST CASES CAN SELECT ANY MAP OPERATION
C   AFTER COMPLETION OF THE PREVIOUS OPERATION.
C
C   THE TAPE USEAGE FOR THIS PROGRAM IS:
C       TAPES 1-20     TITLES AND LEGENDS FOR 20 MAPS
C       TAPES 21-40    ACTUAL MAPS ASSOCIATED WITH DOCUMENTATION ON
C                         TAPES 1-20
C       TAPE 43        LINE PRINTER OUTPUT FILE
C       TAPE 44        RECODED DATA OUTPUT FILE
C       TAPE 45        RECODED DATA INPUT FILE, ASSIGNMENT INPUT FILE
C       TAPE 46        RAW DATA INPUT FILE, ASSIGNMENT INPUT FILE,
C                         WORKING FILE
C       TAPE 47        SYMBOL LIST INPUT FILE, WORKING FILE
C       TAPE 48        WORKING FILE
C       TAPE 49        TERMINAL INPUT FILE
C       TAPE 50        TERMINAL OUTPUT FILE
C
C*****************************************************************
C
      WRITE(50,1001)
 1001 FORMAT(/4X,*MAPPING ROUTINE*/*CREATED BY R. HAGESTEDT*/
     16X,*SEPT.  1978*)
C
      WRITE(50,1002)
 1002 FORMAT(/*THE FOLLOWING FILE NUMBERS ARE SET.*/
     1*THE USER MUST BE CONSISTENT WITH THEIR DESIGNATION.*/
     23X,*LINE PRINTER OUTPUT FILE (43)*/
     33X,*RECODED DATA OUTPUT FILE (44)*/
     43X,*RECODED DATA INPUT FILE (45)*/
     53X,*RAW DATA INPUT FILE (46)*//
     6*WHEN A QUESTION IS ASKED, ANSWER WITH A YES OR NO.*/
     7*WHEN MORE THAN ONE NUMBER IS TO BE PLACED ON A SINGLE *,
     8*LINE,*/*SEPARATE THEM WITH BLANKS OR COMMAS.*)
C
C*****************************************************************
C
C   THE MAPS WHICH ARE OUTPUT FROM THE PROGRAM CONTAIN A STANDARDIZED
C   SET OF SYMBOLS.  THE FOLLOWING CODE INPUTS THIS SYMBOL LIST FROM
```

```
C  TAPE47.  THIS FILE IS LATER USED BY THE PROGRAM AS A WORKING FILE.
C
C*****************************************************************
C
      REWIND 47
      READ(47,1003) (ISYMBL(I),I=1,2496)
1003  FORMAT(10A2)
C
C*****************************************************************
C
C  THE FOLLOWING CODE SETS UP AND CALLS THE INPUT SUBROUTINE.  THIS
C  SUBROUTINE WILL INPUT MAPS IN EITHER THE RAW OR RECODED DATA FORM.
C  IN SOME CASES WHERE A PREVIOUS RUN WAS JUST MADE AND THE MAPS ARE
C  ALREADY PLACED ON LOCAL FILES 1-40, THE NUMBER OF AVAILABLE MAPS
C  IS SUPPLIED AND THE INPUT PROCEDURE CAN BE SKIPPED.
C
C*****************************************************************
C
      WRITE(50,1004)
1004  FORMAT(/*IS THERE ANY MAP DATA TO BE INPUT+ *)
      READ(49,1005) KODE
1005  FORMAT(A1)
      IF(KODE.EQ.1HY) GO TO 100
C
      WRITE(50,1006)
1006  FORMAT(/*HOW MANY MAPS ARE AVAILABLE+ *)
      READ(49,*) MFILE
      GO TO 110
C
100   CALL OVERLAY(4HMAPS,1,0)
C
C*****************************************************************
C
C  THIS CODE ALLOWS SELECTION OF A MAP OPERATION AND CALLS THE
C  APPROPRIATE SUBROUTINE.
C
C*****************************************************************
C
110   WRITE(50,1007)
1007  FORMAT(/*LIST OF MAP OPERATIONS AND THEIR KEYS.*/
     A*TO SELECT A GIVEN OPERATION - TYPE IN THE KEY.*//
     B*LIST - TO GET THIS LIST*/
     C*TERMINATE - TO TERMINATE MAP OPERATIONS*/
     D*OUTPUT - TO OUTPUT MAPS*/
     E*COMBINE - TO COMBINE MAPS AND DOCUMENTATION*/
     F*AGGREGATE - TO AGGREGATE MAPS AND UPDATE DOCUMENTATION*/
     G*ASSIGN - TO MAKE MAP ASSIGNMENTS AND UPDATE DOCUMENTATION*/
     H*REORDER - TO REORDER THE MAP SYMBOLS AND LEGEND*/
     I*DIRECTORY - TO EXAMINE THE DIRECTORY*/
     J*HIGHLIGHT - TO HIGHLIGHT CHARACTERISTIC(S) OF DOCUMENTATION*/
     K*PARCEL - TO GROUP ADJACENT CELLS INTO PARCELS*/
     L*UPDATE - TO UPDATE MAP TITLE OR LEGEND*/
     M*MANIPULATE - TO MANIPULATE PARCEL MAPS*)
C
120   WRITE(50,1008)
1008  FORMAT(//*SELECT A MAP OPERATION*)
      READ(49,1009) KODE
1009  FORMAT(A3)
C
      IF(KODE.EQ.3HLIS) GO TO 110
      IF(KODE.EQ.3HOUT) GO TO 130
      IF(KODE.EQ.3HCOM) GO TO 140
      IF(KODE.EQ.3HAGG) GO TO 150
      IF(KODE.EQ.3HASS) GO TO 150
      IF(KODE.EQ.3HREO) GO TO 150
      IF(KODE.EQ.3HDIR) GO TO 160
      IF(KODE.EQ.3HHIG) GO TO 170
      IF(KODE.EQ.3HPAR) GO TO 180
      IF(KODE.EQ.3HUPD) GO TO 190
      IF(KODE.EQ.3HMAN) GO TO 200
      IF(KODE.EQ.3HTER) GO TO 210
C
      WRITE(50,1010)
1010  FORMAT(/*THAT*S NOT ONE OF THE CHOICES*)
      GO TO 110
130   CALL OVERLAY(4HMAPS,3,0)
      GO TO 120
140   CALL OVERLAY(4HMAPS,4,0)
      GO TO 120
```

```
  150   CALL OVERLAY(4HMAPS,5,0)
        GO TO 120
  160   CALL DIRECT
        GO TO 120
  170   CALL OVERLAY(4HMAPS,6,0)
        GO TO 120
  180   CALL OVERLAY(4HMAPS,7,0)
        GO TO 120
  190   CALL OVERLAY(4HMAPS,8,0)
        GO TO 120
  200   CALL OVERLAY(4HMAPS,9,0)
        GO TO 120
C
C**********************************************************************
C
C    AFTER ALL MAP OPERATIONS HAVE BEEN COMPLETED BUT BEFORE PROGRAM
C    TERMINATION, THE SUBROUTINE SELECT IS CALLED.  THIS SUBROUTINE
C    ALLOWS THE USER TO SELECT RECODED MAPS TO BE SAVED IN THAT FORM
C    FOR INPUT ON SUBSEQUENT RUNS.  ALL THE SELECTED MAPS ARE PLACED
C    ON TAPE44.
C
C**********************************************************************
C
  210   CALL OVERLAY(4HMAPS,2,0)
C
        WRITE(50,1011)
 1011   FORMAT(/*RECODED MAPS HAVE BEEN PLACED ON FILE 44*/
       1*REMEMBER TO SAVE THAT FILE*/*THE LINE PRINTER OUTPUT HAS *
       2*BEEN PLACED ON FILE 43*/*REMEMBER TO ROUTE THAT FILE*)
C
        STOP
        END
```

```
      SUBROUTINE DIRECT
      DIMENSION TITLE(6)
      COMMON/A/MFILE,MF,LF
C
C*******************************************************************
C
C  SINCE THE PROGRAM IS INTERACTIVE, IT IS IMPORTANT THAT AT ANY POINT
C  IN A RUN THE USER KNOW WHICH MAPS ARE AVAILABLE AND WHICH MAP
C  NUMBER CONTAINS WHAT INFORMATION.  THIS IS ACHIEVED BY SUBROUTINE
C  DIRECT.  IT PROVIDES THE MAP NUMBER AND TITLE OF EACH AVAILABLE MAP.
C
C*******************************************************************
C
      WRITE(50,1001)
 1001 FORMAT(/≠DIRECTORY FOR OVERLAYS, AGGREGATIONS, AND COMBINATIONS≠/)
C
      DO 100 J=1,MFILE
      REWIND J
C
      READ(J,1002) NT,(TITLE(I),I=1,6)
 1002 FORMAT(I2,6X,6A10)
      WRITE(50,1003) J,(TITLE(I),I=1,6)
 1003 FORMAT(/≠MAP ≠,I2,4X,6A10)
C
      DO 100 K=1,NT
      READ(J,1004) (TITLE(I),I=1,6)
  100 WRITE(50,1004) (TITLE(I),I=1,6)
 1004 FORMAT(8X,6A10)
C
      RETURN
      END
```

```
      SUBROUTINE EXCESS
      COMMON/A/MFILE,MF,LF
      COMMON/D/TITLE(6)
C
C*********************************************************************
C
C   THIS SUBROUTINE ALLOWS FOR MORE THAN TWENTY MAPS BY REPLACING ONE OF
C   THE TWENTY WITH THE EXCESS MAP.  THE REPLACED MAP CAN BE ELIMINATED
C   OR MAY BE PLACED ON A RECODED MAP OUTPUT FILE.  THIS FILE MAKES
C   THESE MAPS ACCESSIBLE FOR LATER RUNS.
C
C   THE TITLE OF THE INCOMING MAP AND THE DIRECTORY OF MAPS ELIGIBLE
C   FOR REPLACEMENT ARE GIVEN.
C
C*********************************************************************
C
      WRITE(50,1001) (TITLE(I),I=1,6)
 1001 FORMAT(/*THIS MAP: *,6A10/*MUST BE WRITTEN OVER AN *
     1*EXISTING MAP.*)
C
      MFILE=20
      CALL DIRECT
C
      WRITE(50,1002)
 1002 FORMAT(/*SELECT THE APPROPRIATE MAP NUMBER *)
      READ(49,*) MF
C
C*********************************************************************
C
C   IF THE EXISTING MAP IS TO BE SAVED, SUBROUTINE COPY IS CALLED.
C   IT WILL PROCESS THE EXISTING MAP BEFORE THE NEW MAP IS PLACED
C   ON THE FILE.
C
C*********************************************************************
C
      WRITE(50,1003)
 1003 FORMAT(/*DO YOU WISH TO PLACE THE EXISTING MAP ON THE RECODED *,
     1*MAP OUTPUT FILE+*)
      READ(49,1004) KA
 1004 FORMAT(A1)
      IF(KA.EQ.1HY) CALL COPY
C
      LF=MF+20
      REWIND MF
      REWIND LF
C
      RETURN
      END
```

```
      SUBROUTINE COPY
      DIMENSION TRANS(12)
      COMMON/A/MFILE,MF,LF
C
C*****************************************************************************
C
C  THIS SUBROUTINE TRANSFERS THE SPECIFIED FILES ONTO THE RECODED
C  MAP OUTPUT FILE (TAPE44).
C
C*****************************************************************************
C
      ML=MF
100   REWIND ML
C
110   READ(ML,1001) (TRANS(K),K=1,12)
1001  FORMAT(12A10)
      IF(EOF(ML)) 130,120
C
120   WRITE(44,1001) (TRANS(K),K=1,12)
      GO TO 110
C
130   ML=ML+20
      IF(ML.LE.40) GO TO 100
C
      RETURN
      END
```

```
      SUBROUTINE RECODE
      DIMENSION MAP(36),MARGIN(250),ICT(1296),DOC(4),TITLE(6)
      COMMON/A/MFILE,MF,LF
      COMMON/B/ICODE(1296),NS,NC,NT
      COMMON/E/KODE
      DATA AREA/21.33/,ICT/1296*0/
C
C*****************************************************************
C
C   THIS SUBROUTINE IS CALLED BY OTHER SUBROUTINES WHEN THEY WANT
C   THE SYMBOLS ON A MAP RECODED ACCORDING TO THE ORDER OF THEIR
C   OCCURRENCE IN THE LEGEND.  THIS SUBROUTINE ALSO RECORDS THE
C   INCIDENCE OF EACH SYMBOL IN THE MAP, SO STATISTICS SUCH AS
C   NUMBER OF CELLS, ACREAGE, AND PERCENTAGE OF ACREAGE CAN BE
C   OUTPUT FOR EACH SYMBOL.
C
C*****************************************************************
C
      TOTAC=0.
      REWIND LF
C
C*****************************************************************
C
C   THE FOLLOWING CODE CHECKS EACH CELL IN EVERY MAP ROW AGAINST THE
C   LEGEND ORDERING INFORMATION, RECODES THAT CELL ACCORDINGLY, AND
C   WRITES THE RECODED ROW ON THE MAP FILE.  A TALLY IS ALSO MADE OF
C   EACH SYMBOL OCCURRENCE.
C
C*****************************************************************
C
100   READ(46,1001) ICOL,IROW,(MAP(I),I=1,36)
1001  FORMAT(2I4,36A2)
      IF(ICOL.EQ.9999) GO TO 190
      IF(ICOL.GT.1.AND.MAP(1).EQ.2H  .AND.MARGIN(IROW).NE.2H**)
     1 MAP(1)=MARGIN(IROW)
      IMAP=2H
C
      DO 180 I=1,36
      IF(MAP(I).EQ.2H  ) GO TO 110
      IF(MAP(I).EQ.2H**) GO TO 170
      IMAP=MAP(I)
      GO TO 120
110   IF(KODE.NE.1HY) GO TO 160
      IF(IMAP.EQ.2H  ) GO TO 160
      MAP(I)=IMAP
120   DO 130 L=1,NS
130   IF(MAP(I).EQ.ICODE(L)) GO TO 140
      MAP(I)=-998
      GO TO 150
140   MAP(I)=L
      ICT(L)=ICT(L)+1
150   TOTAC=TOTAC+1.
      GO TO 180
160   MAP(I)=0
      GO TO 180
170   IMAP=2H
      MAP(I)=-999
180   CONTINUE
C
      IF(MAP(36).GT.0) MARGIN(IROW)=ICODE(MAP(36))
      IF(MAP(36).EQ.-999) MARGIN(IROW)=2H**
      IF(MAP(36).EQ.-998) MARGIN(IROW)=2H++
      IF(MAP(36).EQ.0) MARGIN(IROW)=2H
C
      WRITE(LF,1002) ICOL,IROW,(MAP(I),I=1,36)
1002  FORMAT(2I4/8X,18I4)
      GO TO 100
C
190   TOTAC=TOTAC*AREA
      WRITE(LF,1003) NS
1003  FORMAT(4H9999,I4)
C
C*****************************************************************
C
C   THE FOLLOWING CODE COMPLETES THE DOCUMENTATION FILE FOR THE MAP.
C   FIRST THE TITLES ARE TRANSFERRED, THEN SYMBOL BY SYMBOL, THE
C   LEGEND IS READ, ACREAGE AND PERCENTAGE ACREAGE IS CALCULATED,
C   AND THE EXPANDED DOCUMENTATION IS WRITTEN ON THE MAP DOCUMENTATION
C   FILE.
```

```
C
C********************************************************************
C
      REWIND 47
      REWIND MF
C
      READ(47,1004) NT,NC,(TITLE(K),K=1,6)
1004  FORMAT(2I2,4X,6A10)
      WRITE(MF,1005) NT,NC,NS,(TITLE(K),K=1,6)
1005  FORMAT(2I2,I4,6A10)
C
      DO 200 I=1,NT
      READ(47,1006) (TITLE(K),K=1,6)
200   WRITE(MF,1006) (TITLE(K),K=1,6)
1006  FORMAT(8X,6A10)
C
      DO 220 I=1,NS
      READ(47,1007) M,(DOC(L),L=1,4)
1007  FORMAT(29X,I4,4A10)
      CAREA=ICT(I)*AREA
      PCAREA=CAREA/TOTAC*100.0
      WRITE(MF,1008) I,ICT(I),CAREA,PCAREA,M,(DOC(L),L=1,4)
1008  FORMAT(I4,I8,F11.1,F5.1,I4,4A10)
      IF(NC.EQ.1) GO TO 220
C
      DO 210 J=2,NC
      READ(47,1007) M,(DOC(L),L=1,4)
210   WRITE(MF,1007) M,(DOC(L),L=1,4)
220   ICT(I)=0
C
      WRITE(MF,1009) TOTAC
1009  FORMAT(F11.1)
C
      RETURN
      END
```

```
      OVERLAY(MAPS,1,0)
      PROGRAM IN
      DIMENSION TRANS(12),DOC(4)
      COMMON/A/MFILE,MF,LF
      COMMON/B/ICODE(1296),NS,NC,NT
      COMMON/D/TITLE(6)
C
C*********************************************************************
C
C     THIS IS THE FIRST SUBROUTINE CALLED WHEN THE PROGRAM IS EXECUTED.
C     IT READS MAPS IN BOTH RECODED AND RAW DATA FORM FROM FILES 45 AND 46.
C     THE DOCUMENTATION FOR THESE MAPS IS PLACED ON FILES 1-20 WHILE THE
C     MAPS THEMSELVES ARE PLACED ON FILES 21-40.
C
C     THE RECODED MAPS ARE INPUT FIRST.  SINCE THE RECODED FORM IS USED IN
C     THE PROGRAM, THIS CODE SIMPLY TRANSFERS THE MAPS FROM THE RECODED
C     DATA INPUT FILE TO THE APPROPRIATE FILES (TAPES 1-20 AND 21-40),
C     MAKING THEM AVAILABLE TO THE USER.
C
C*********************************************************************
C
100   READ(45,1001) NT,NC,NS,(TITLE(I),I=1,6)
1001  FORMAT(2I2,I4,6A10)
      IF(EOF(45)) 140,110
C
C*********************************************************************
C
C     THIS CODE IS USED WHENEVER THE NUMBER OF AVAILABLE MAPS IS
C     INCREASED (EITHER A NEW MAP IS INPUT OR ONE IS CREATED BY A
C     MAP OPERATION).  THE NUMBER OF AVAILABLE MAPS IS INCREASED,
C     THE FILE NUMBERS FOR THE NEW MAP ARE DESIGNATED, AND A CHECK
C     IS MADE TO SEE IF THE NEW MAP PUSHES THE NUMBER OF AVAILABLE
C     MAPS BEYOND THE FILE SPACE FOR TWENTY MAPS.  IF THE NEW MAP
C     IS NUMBER TWENTY-ONE, SUBROUTINE EXCESS IS CALLED TO HANDLE
C     THE PROBLEM.
C
C*********************************************************************
C
110   MFILE=MFILE+1
      MF=MFILE
      LF=MF+20
      IF(MFILE.GT.20) CALL EXCESS
C
C*********************************************************************
C
      WRITE(MF,1001) NT,NC,NS,(TITLE(I),I=1,6)
      NQ=NT+NS*NC+1
      DO 120 J=1,NQ
      READ(45,1002) (TRANS(I),I=1,8)
120   WRITE(MF,1002) (TRANS(I),I=1,8)
1002  FORMAT(8A10)
C
130   READ(45,1003) ICOL,(TRANS(I),I=1,12)
      WRITE(LF,1003) ICOL,(TRANS(I),I=1,12)
1003  FORMAT(I4,12A10)
      IF(ICOL.EQ.9999) GO TO 100
      GO TO 130
C
C*********************************************************************
C
C     THE MAPS IN RAW DATA FORM ARE READ IN, THEIR SYMBOL LIST AND LEGEND
C     ARE RECODED, AND THIS INFORMATION IS CARRIED TO SUBROUTINE RECODE
C     WHERE THE RECODING PROCESS IS COMPLETED.  THESE MAPS ARE PLACED ON
C     THE UNUSED FILES AMONG TAPES 1-20 AND 21-40.
C
C*********************************************************************
C
140   READ(46,1004) (TITLE(I),I=1,6)
1004  FORMAT(6A10)
      IF(EOF(46)) 180,150
C
C*********************************************************************
C
C     THIS CODE IS USED WHENEVER THE NUMBER OF AVAILABLE MAPS IS
C     INCREASED (EITHER A NEW MAP IS INPUT OR ONE IS CREATED BY A
C     MAP OPERATION).  THE NUMBER OF AVAILABLE MAPS IS INCREASED,
C     THE FILE NUMBERS FOR THE NEW MAP ARE DESIGNATED, AND A CHECK
C     IS MADE TO SEE IF THE NEW MAP PUSHES THE NUMBER OF AVAILABLE
C     MAPS BEYOND THE FILE SPACE FOR TWENTY MAPS.  IF THE NEW MAP
```

```
C   IS NUMBER TWENTY-ONE, SUBROUTINE EXCESS IS CALLED TO HANDLE
C   THE PROBLEM.
C
C*********************************************************************
C
150     MFILE=MFILE+1
        MF=MFILE
        LF=MF+20
        IF(MFILE.GT.20) CALL EXCESS
C
C*********************************************************************
C
        REWIND 47
        NS=0
C
        WRITE(47,1005) (TITLE(I),I=1,6)
1005    FORMAT(*1010101,4X,*OVERLAY*/8X,6A10)
C
160     READ(46,1006) ITYPE,END,(DOC(L),L=1,4)
1006    FORMAT(A2,2X,A3,4A10)
        IF(END.EQ.3HEND) GO TO 170
C
        NS=NS+1
        ICODE(NS)=ITYPE
        WRITE(47,1007) NS,(DOC(L),L=1,4)
1007    FORMAT(23X,I4,4A10)
        GO TO 160
C
170     NC=1
        CALL RECODE
        GO TO 140
C
180     RETURN
        END
```

```
      OVERLAY(MAPS,2,0)
      PROGRAM SELECT
      DIMENSION NMAP(20)
      COMMON/A/MFILE,MF,LF
C
C*****************************************************************
C
C     THIS SUBROUTINE ASKS FOR THOSE MAPS WHICH THE OPERATOR WISHES
C     TO SAVE ON THE RECODED MAP OUTPUT FILE.  THIS FILE CAN BE INPUT
C     ON LATER RUNS.  SUBROUTINE COPY IS CALLED TO MAKE THE ACTUAL
C     TRANSFER.
C
C*****************************************************************
C
      WRITE(50,1001)
 1001 FORMAT(/≠DO YOU WISH TO SAVE ANY OF THE RECODED MAPS FOR ≠
     1≠FUTURE USE+≠)
      READ(49,1002) KA
 1002 FORMAT(A1)
      IF(KA.EQ.1HN) GO TO 110
C
      WRITE(50,1003)
 1003 FORMAT(/≠DO YOU WISH TO SEE THE DIRECTORY TO AID IN ≠
     1≠YOUR SELECTION+≠)
      READ(49,1002) KB
      IF(KB.EQ.1HY) CALL DIRECT
C
      WRITE(50,1004)
 1004 FORMAT(/≠HOW MANY RECODED MAPS DO YOU WISH TO SAVE+≠)
      READ(49,*) KC
      IF(KC.EQ.0) GO TO 110
C
      WRITE(50,1005)
 1005 FORMAT(/≠LIST EACH MAP YOU WANT SAVED ON A SINGLE LINE≠/
     1≠WITH EACH MAP NUMBER SEPARATED BY A COMMA.≠)
      READ(49,*) (NMAP(I),I=1,KC)
C
      DO 100 I=1,KC
      MF=NMAP(I)
  100 CALL COPY
C
  110 RETURN
      END
```

```
      OVERLAY(MAPS,3,0)
      PROGRAM OUT
      DIMENSION TITL(20,6),MAP(36),DOC(4),TITLE(6)
      COMMON/D/ISYMBL(2496)
C
C*************************************************************
C
C   THIS SUBROUTINE OUTPUTS MAPS IN A STANDARDIZED FORM TO EITHER THE
C   TERMINAL OR TO A LINE PRINTER FILE.  THE MAPS ARE SPLIT VERTICALLY
C   INTO COLUMNS OF 36 CELLS PER ROW WITH CONTINUATION IN THE NEXT
C   COLUMN.  THIS ALLOWS DISPLAY AT A TERMINAL WITHOUT WRAP-AROUND.
C
C*************************************************************
C
      WRITE(43,1000)
1000  FORMAT(*1*)
C
100   IERR=0
      ICOL2=0
      KI=50
C
      WRITE(50,1001)
1001  FORMAT(/*DO YOU WISH TO SEE THE DIRECTORY+*)
      READ(49,1002) IS
1002  FORMAT(A1)
      IF(IS.EQ.1HY) CALL DIRECT
C
110   WRITE(50,1003)
1003  FORMAT(/*WHICH MAP DO YOU WISH TO SEE+*)
      READ(49,*) IR
      IF(IR.EQ.0) GO TO 230
C
      JR=IR+20
      REWIND IR
      REWIND JR
C
      WRITE(50,1004)
1004  FORMAT(/*WHERE IS THE OUTPUT TO BE DISPLAYED+*/
     1*(1)IMMEDIATE DISPLAY (2)LINE PRINTER DISPLAY*)
      READ(49,*) IT
      IF(IT.EQ.2) KI=43
C
C*************************************************************
C
C   THE FOLLOWING CODE PROVIDES HEADINGS AT THE TOP OF EACH COLUMN OF
C   MAP OUTPUT AND AT THE TOP OF THE MAP LEGEND OUTPUT.
C
C*************************************************************
C
      READ(IR,1005) NT,NC,NS,(TITLE(I),I=1,6)
1005  FORMAT(2I2,I4,6A10)
      DO 120 I=1,NT
120   READ(IR,1006) (TITL(I,J),J=1,6)
1006  FORMAT(9X,6A10)
C
130   READ(JR,1007) ICOL,IROW,(MAP(I),I=1,36)
1007  FORMAT(20I4/8X,18I4)
      IF(ICOL2.EQ.ICOL) GO TO 150
C
      WRITE(KI,1008) IR,(TITLE(I),I=1,6)
1008  FORMAT(*1MAP *,I2,4X,6A10)
      DO 140 I=1,NT
140   WRITE(KI,1009) (TITL(I,J),J=1,6)
1009  FORMAT(1X/10X,6A10)
      ICOL2=ICOL
      IF(ICOL.EQ.9999) GO TO 200
      WRITE(KI,1010) ICOL
1010  FORMAT(1X//1X,*COLUMN *,I4/)
C
C*************************************************************
C
C   THE FOLLOWING CODE CONVERTS THE MAP CELLS TO SYMBOLS USING THE
C   SYMBOL LIST INPUT AT THE BEGINNING OF THE PROGRAM, THEN WRITES
C   THE MAP ON THE SELECTED OUTPUT DEVICE.
C
C*************************************************************
C
150   DO 190 I=1,36
      IF(MAP(I).EQ.0) GO TO 160
```

```
      IF(MAP(I).EQ.-999) GO TO 170
      IF(MAP(I).EQ.-998) GO TO 180
      MAP(I)=ISYMBL(MAP(I))
      GO TO 190
160   MAP(I)=2H
      GO TO 190
170   MAP(I)=2H**
      GO TO 190
180   MAP(I)=2H++
      IERR=IERR+1
190   CONTINUE
C
      WRITE(KI,1011) IROW,(MAP(I),I=1,36)
1011  FORMAT(1X,I4,2X,36A2,≠+≠)
      GO TO 130
C
C*******************************************************************
C
C  THE FOLLOWING CODE WRITES THE LEGEND AND ASSOCIATED DOCUMENTATION
C  ON THE SELECTED OUTPUT DEVICE.
C
C*******************************************************************
C
200   WRITE(KI,1012)
1012  FORMAT(1X//1X,≠SYMBOL≠,4X,≠NO. CELLS≠,4X,≠ACREAGE≠,5X,≠PCT.≠,
     15X,≠LEGEND≠)
C
      DO 220 I=1,NS
      READ(IR,1013) J,ITC,CACR,PCACR,(DOC(L),L=1,4)
1013  FORMAT(I4,I8,F11.1,F5.1,4X,4A10)
      WRITE(KI,1014) ISYMBL(J),ITC,CACR,PCACR,(DOC(L),L=1,4)
1014  FORMAT(1X/3X,A2,6X,I8,1X,F11.1,4X,F5.1,5X,4A10)
      IF(NC.EQ.1) GO TO 220
C
      DO 210 M=2,NC
      READ(IR,1015) (DOC(L),L=1,4)
1015  FORMAT(32X,4A10)
210   WRITE(KI,1016) (DOC(L),L=1,4)
1016  FORMAT(45X,4A10)
220   CONTINUE
C
      IF(IERR.GE.1) WRITE(KI,1017) IERR
1017  FORMAT(1X/3X,2H++,6X,I8,26X,≠**UNDOCUMENTED ACREAGE**≠)
C
      READ(IR,1018) TOTAC
1018  FORMAT(F11.1)
      WRITE(KI,1019) TOTAC
1019  FORMAT(1X/3X,2H**,40X,≠BOUNDARY≠//8X,≠TOTAL ACREAGE = ≠,
     1F11.1,≠ ACRES≠)
C
C*******************************************************************
C
      WRITE(50,1020)
1020  FORMAT(/≠OTHER OUTPUT*≠)
      READ(49,1002) IU
      IF(IU.EQ.1HY) GO TO 100
C
230   RETURN
      END
```

```
      OVERLAY(MAPS,4,0)
      PROGRAM COMBO
      DIMENSION NMAP(20),MAP1(36),MAP2(36),MAP3(36),
     1ICODE3(1296),DOC1(20,4),DOC2(20,4),TITL(6),
     2MOC1(20),MOC2(20)
      COMMON/A/MFILE,MF,LF
      COMMON/B/ICODE(1296),NS,NC,NT
      COMMON/C/ISYMBL(2496)
      COMMON/D/TITLE(6)
C
C*****************************************************************
C
C  THIS SUBROUTINE COMBINES SEVERAL MAPS INTO A SINGLE MAP BY
C  ASSIGNING A NEW SYMBOL TO EACH UNIQUE COMBINATION CREATED DURING
C  THE PROCESS.  THE LEGENDS OF THE MAPS ARE ALSO COMBINED, RESULTING
C  IN A NEW LEGEND WHICH CONTAINS MULTI-LINED DESCRIPTIONS FOR EACH
C  SYMBOL.
C
C*****************************************************************
C
100   WRITE(50,1001)
1001  FORMAT(/≠HOW MANY MAPS DO YOU WISH TO COMBINE+≠)
      READ(49,*) JA
      IF(JA.EQ.0) GO TO 400
C
      WRITE(50,1002)
1002  FORMAT(/≠DO YOU WISH TO SEE THE DIRECTORY TO AID IN ≠
     1≠YOUR SELECTION+≠)
      READ(49,1003) JB
1003  FORMAT(A1)
      IF(JB.EQ.1HY) CALL DIRECT
C
      WRITE(50,1004)
1004  FORMAT(/≠LIST EACH MAP IN THE ORDER OF COMBINATION.≠/
     1≠PLACE THE MAP NUMBERS ON A SINGLE LINE WITH EACH ≠/
     2≠MAP NUMBER SEPARATED BY A COMMA.≠)
      READ(49,*) (NMAP(I),I=1,JA)
C
      WRITE(50,1005)
1005  FORMAT(/≠ENTER NAME OF THE COMBINATION (45 CHARACTERS)≠)
      READ(49,1006) (TITLE(K),K=1,5)
1006  FORMAT(5A10)
C
C*****************************************************************
C
C  THIS CODE IS USED WHENEVER THE NUMBER OF AVAILABLE MAPS IS
C  INCREASED (EITHER A NEW MAP IS INPUT OR ONE IS CREATED BY A
C  MAP OPERATION).  THE NUMBER OF AVAILABLE MAPS IS INCREASED,
C  THE FILE NUMBERS FOR THE NEW MAP ARE DESIGNATED, AND A CHECK
C  IS MADE TO SEE IF THE NEW MAP PUSHES THE NUMBER OF AVAILABLE
C  MAPS BEYOND THE FILE SPACE FOR TWENTY MAPS.  IF THE NEW MAP
C  IS NUMBER TWENTY-ONE, SUBROUTINE EXCESS IS CALLED TO HANDLE
C  THE PROBLEM.
C
C*****************************************************************
C
      MFILE=MFILE+1
      JC=MFILE
      MF=MFILE
      LF=MF+20
      IF(MFILE.GT.20) CALL EXCESS
C
C*****************************************************************
C
C  ONLY TWO MAPS ARE COMBINED IN ONE PASS THROUGH THE SUBROUTINE, SO
C  THE FOLLOWING CODE INITIALIZES CONDITIONS FOR THE TWO MAPS BEING
C  COMBINED IN EACH PASS.
C
C*****************************************************************
C
      ML1=NMAP(1)
      ML2=NMAP(2)
      NL1=ML1+20
      NL2=ML2+20
      JE=2
C
110   REWIND ML1
      REWIND ML2
      REWIND NL1
```

```
      REWIND NL2
      REWIND 46
      REWIND 47
C
      READ(ML1,1007) NT1,NC1
      READ(ML2,1007) NT2,NC2
1007  FORMAT(2I2)
      NT=NT1+NT2
      NC=NC1+NC2
      NS=0
C
C*****************************************************************
C
C   THE FOLLOWING CODE COMBINES THE CELLS OF TWO MAPS, CHECKS THAT
C   COMBINATION AGAINST A LIST OF PREVIOUS CELL COMBINATIONS (ADDING
C   TO THE LIST IF IT IS NEW), CODES THE CELL ACCORDING TO ITS PLACE
C   ON THE LIST, AND WRITES THE MAP INFORMATION IN A FORM WHICH CAN
C   BE USED BY THE SUBROUTINE RECODE.
C
C*****************************************************************
C
120   READ(NL1,1008) ICOL1,IROW1,(MAP1(I),I=1,36)
      READ(NL2,1008) ICOL2,IROW2,(MAP2(I),I=1,36)
1008  FORMAT(2UI4/8X,18I4)
      IF(ICOL1.EQ.9999.AND.ICOL2.EQ.9999) GO TO 200
      IF(ICOL1.NE.9999.AND.ICOL2.NE.9999) GO TO 130
      GO TO 390
C
130   DO 190 I=1,36
      IF(MAP1(I).EQ.-999.AND.MAP2(I).EQ.-999) GO TO 170
      IF(MAP1(I).EQ.-999.OR.MAP2(I).EQ.-999) GO TO 390
      IF(MAP1(I).EQ.0.AND.MAP2(I).EQ.0) GO TO 180
      MAP3(I)=MAP1(I)*10000+MAP2(I)
      IF(NS.EQ.0) GO TO 150
      DO 140 L=1,NS
140   IF(MAP3(I).EQ.ICODE3(L)) GO TO 160
150   NS=NS+1
      ICODE3(NS)=MAP3(I)
      MAP3(I)=ISYMBL(NS)
      GO TO 190
160   MAP3(I)=ISYMBL(L)
      GO TO 190
170   MAP3(I)=2H**
      GO TO 190
180   MAP3(I)=2H
190   CONTINUE
C
      WRITE(46,1009) ICOL1,IROW1,(MAP3(I),I=1,36)
1009  FORMAT(2I4,36A2)
      GO TO 120
C
200   WRITE(46,1010) NS
1010  FORMAT(+49999,I4)
C
C*****************************************************************
C
C   AFTER CREATING THE TITLE FOR THIS NEW MAP, THE LEGEND IS DEVELOPED.
C   THE CODE IS SET UP TO ARRANGE THE LEGENDS IN ORDER WITH THE LEGEND
C   OF THE SECOND MAP NESTED WITHIN THE FIRST.  SINCE NOT EVERY
C   COMBINATION OF LEGENDS MAY EXIST IN THE MAP, THE LEGEND IS COMPARED
C   TO THE LIST OF COMBINATIONS CREATED FROM THE MAP CELLS AND THOSE
C   MISSING ARE SKIPPED OVER.  THE REMAINDER OF THE LEGEND IS WRITTEN
C   ON A WORKING FILE, THE INFORMATION TO RECODE THE MAP CELLS
C   CONSISTENT WITH THE LEGEND IS STORED, AND SUBROUTINE RECODE IS
C   CALLED.
C
C*****************************************************************
C
      WRITE(47,1011) NT,NC,(TITLE(K),K=1,5)
1011  FORMAT(2I2,4X,≠COMBINATION - ≠,5A10)
      DO 210 J=1,NT1
      READ(ML1,1012) (TITL(I),I=1,6)
210   WRITE(47,1012) (TITL(I),I=1,6)
1012  FORMAT(8X,6A10)
      DO 220 J=1,NT2
      READ(ML2,1012) (TITL(I),I=1,6)
220   WRITE(47,1012) (TITL(I),I=1,6)
C
      REWIND 43
```

```
      DO 230 IB=1,IROW2
      DO 230 I=1,NC2
      READ(ML2,1013) MOC2(I),(DOC2(I,J),J=1,4)
1013  FORMAT(23X,I4,4A10)
230   WRITE(48,1014) IB,MOC2(I),(DOC2(I,J),J=1,4)
1014  FORMAT(I4,24X,I4,4A10)
C
      IP=0
      JP=0
C
      DO 380 IJ=1,NS
      IM=ICODE3(1)
C
      DO 240 IN=1,NS
      IF(ICODE3(IN).GT.IM) GO TO 240
      IM=ICODE3(IN)
      IQ=IN
240   CONTINUE
      ICODE(IJ)=ISYMBL(IQ)
      ICODE3(IQ)=100000000
C
      IA=IM/10000
      IF(IA.EQ.0) GO TO 270
250   IF(IA.EQ.IP) GO TO 290
      READ(ML1,1014) IP,MOC1(1),(DOC1(1,J),J=1,4)
      IF(NC1.EQ.1) GO TO 250
      DO 260 I=2,NC1
260   READ(ML1,1013) MOC1(I),(DOC1(I,J),J=1,4)
      GO TO 250
C
270   DO 280 I=1,NC1
280   WRITE(47,1015) IQ
1015  FORMAT(I4,24X,≠0000******≠)
      GO TO 310
C
290   DO 300 I=1,NC1
300   WRITE(47,1014) IQ,MOC1(I),(DOC1(I,J),J=1,4)
C
310   REWIND 48
      IB=IM-IA*10000
      IF(IB.EQ.0) GO TO 340
320   IF(IB.EQ.JP) GO TO 360
      DO 330 I=1,NC2
330   READ(48,1014) JP,MOC2(I),(DOC2(I,J),J=1,4)
      GO TO 320
C
340   DO 350 I=1,NC2
350   WRITE(47,1015) IQ
      GO TO 330
C
360   DO 370 I=1,NC2
370   WRITE(47,1014) IQ,MOC2(I),(DOC2(I,J),J=1,4)
380   CONTINUE
C
      REWIND 45
      CALL RECODE
C
C**********************************************************************
C
C  IF THE NUMBER OF MAPS TO BE COMBINED EXCEEDS TWO, THIS SUBROUTINE
C  IS PASSED THROUGH AN ADDITIONAL TIME FOR EACH ADDITIONAL MAP.  THE
C  PREVIOUSLY CREATED COMBINATION IS TREATED AS THE FIRST MAP AND THE
C  NEXT MAP TO BE COMBINED IS TREATED AS THE SECOND MAP.  THE FOLLOWING
C  CODE TESTS FOR A TERMINATION OF THE COMBINATION PROCESS AND WILL SET
C  UP CONDITIONS FOR ANY OF THE ADDITIONAL RUNS THROUGH THE SUBROUTINE.
C
C**********************************************************************
C
      IF(ML2.EQ.NMAP(JA)) GO TO 400
      JE=JE+1
      ML1=MF
      ML2=NMAP(JE)
      NL1=LF
      NL2=ML2+20
      GO TO 110
C
C**********************************************************************
C
390   WRITE(50,1016) ML1,ML2
```

```
240        1016  FORMAT(///≠COMBINATION TERMINATED≠/≠MAPS ≠,I2,
                1≠ AND ≠,I2,≠ ARE NOT IDENTICAL IN SIZE≠)
                 IF(JC.LE.20) MFILE=MFILE-1
           C
           400   RETURN
                 END
```

```
      OVERLAY(MAPS,5,0)
      PROGRAM AGGASS
      DIMENSION MAP(50),DOC(4),IGRP(1296),NAP(1296),ICT(1296)
      COMMON/A/MFILE,MF,LF
      COMMON/B/ISYMBL(2496)
      COMMON/D/TITLE(6)
      COMMON/E/KODE
      COMMON/F/IR
      DATA AREA/21.33/,ICT/1296*0/
C
C*********************************************************************
C
C  THIS SUBROUTINE ALLOWS A USER TO AGGREGATE OR ASSIGN THE CELLS OF
C  EACH SYMBOL INTO GROUPS AND DOCUMENT THOSE GROUPS.  THIS INFORMATION
C  CAN BE PROVIDED EITHER INTERACTIVELY FROM THE TERMINAL OR FROM TWO
C  FILES.  IF THE INFORMATION IS ON FILES, TAPE45 CONTAINS THE ACTUAL
C  ASSIGNMENTS AND TAPE46 CONTAINS THE GROUP DOCUMENTATION.  IT IS
C  IMPORTANT IN CASES WHERE THE INFORMATION IS ON FILE THAT THE MAPS
C  BE INPUT ON A PREVIOUS RUN AND THIS SUBROUTINE BE THE FIRST
C  OPERATION IN THE CURRENT RUN.
C
C*********************************************************************
C
      TOTAC=0.
      MNAP=0
      KD=2
C
      WRITE(50,1001)
 1001 FORMAT(/*DO YOU WISH TO SEE THE DIRECTORY+*)
      READ(49,1002) IS
 1002 FORMAT(A1)
      IF(IS.EQ.1HY) CALL DIRECT
C
      WRITE(50,1003)
 1003 FORMAT(/*WHICH MAP DO YOU WISH TO SELECT+*)
      READ(49,*) IR
      IF(IR.EQ.0) GO TO 310
C
C*********************************************************************
C
C  THIS CODE IS USED WHENEVER THE NUMBER OF AVAILABLE MAPS IS
C  INCREASED (EITHER A NEW MAP IS INPUT OR ONE IS CREATED BY A
C  MAP OPERATION).  THE NUMBER OF AVAILABLE MAPS IS INCREASED,
C  THE FILE NUMBERS FOR THE NEW MAP ARE DESIGNATED, AND A CHECK
C  IS MADE TO SEE IF THE NEW MAP PUSHES THE NUMBER OF AVAILABLE
C  MAPS BEYOND THE FILE SPACE FOR TWENTY MAPS.  IF THE NEW MAP
C  IS NUMBER TWENTY-ONE, SUBROUTINE EXCESS IS CALLED TO HANDLE
C  THE PROBLEM.
C
C*********************************************************************
C
      MFILE=MFILE+1
      MF=MFILE
      LF=MF+20
      IF(MFILE.GT.20) CALL EXCESS
C
C*********************************************************************
C
C  THE FOLLOWING CODE CALLS THE SUBROUTINE WHICH WILL REORDER THE
C  MAP SYMBOLS IN THE LEGEND ACCORDING TO USER PREFERENCE OR BY
C  LARGEST ACREAGE FIRST.  THE USER THEN INPUTS INFORMATION
C  DIRECTING THE ASSIGNMENT/AGGREGATION PROCESS AND THE NEW
C  MAP TITLE.
C
C*********************************************************************
C
      IF(KODE.EQ.3HREO) CALL REORDER
      JR=IR+20
      REWIND IR
      REWIND JR
      REWIND 47
      READ(IR,1004) NT,NC
 1004 FORMAT(2I2)
      IF(KODE.EQ.3HAGG) KD=1
C
      WRITE(50,1005)
 1005 FORMAT(/*SELECT THE INPUT DEVICE  (1) TERMINAL  (2) FILES*/
     1*AND INPUT THE NAME OF THE NEW MAP (45 CHARACTERS).*/
     2*PLACE THIS INFORMATION ON ONE LINE.*/*SEPARATE WITH *,
```

```
     3≠A COMMA OR BLANK.≠)
      READ(49,1006) LOC,(TITLE(K),K=1,5)
1006  FORMAT(I1,1X,5A10)
      IF(LOC.EQ.1) GO TO 120
C
C*********************************************************************
C
C  THE FOLLOWING CODE READS THE MAP ASSIGNMENTS/AGGREGATIONS AND THE
C  NEW LEGEND FROM TAPES 45 & 46.
C
C*********************************************************************
C
      READ(45,1007) MNAP,NS
1007  FORMAT(2I4)
      DO 100 K=1,NS
100   READ(45,1007) IGRP(K),NAP(K)
C
      READ(46,1007) MNAP,NC
      NQ=MNAP*NC
      DO 110 I=1,NQ
      READ(46,1008) M,(DOC(J),J=1,4)
110   WRITE(47,1008) M,(DOC(J),J=1,4)
1008  FORMAT(I4,4A10)
      GO TO 180
C
C*********************************************************************
C
C  THE FOLLOWING CODE ALLOWS THE MAP ASSIGNMENTS/AGGREGATIONS AND
C  THE NEW LEGEND TO BE INPUT FROM THE TERMINAL.
C
C*********************************************************************
C
120   WRITE(50,1009) NC
1009  FORMAT(/≠ENTER EACH GROUP NUMBER, ITS DOCUMENTATION, ≠,
     1≠AND THE ASSOCIATED PLOT SYMBOLS.≠/≠THE LEGEND FOR EACH ≠,
     2≠GROUP MUST BE ≠,I2,≠ LINE(S) - 40 CHARACTERS PER LINE.≠/
     3≠THE PLOT SYMBOLS SHOULD BE PLACED IN QUOTATION MARKS≠/
     4≠WITH EACH SEPARATED BY A COMMA OR BLANK.≠//
     5≠HOW MANY GROUPS DO YOU WISH TO CREATE?≠)
      READ(49,*) NG
C
      K=0
      DO 170 IJ=1,NG
      WRITE(50,1010)
1010  FORMAT(/≠ENTER GROUP NUMBER AND NUMBER OF ASSOCIATED SYMBOLS.≠)
      READ(49,*) NP,MP
C
      WRITE(50,1011) NP
1011  FORMAT(/≠ENTER LEGEND FOR GROUP ≠,I3)
      DO 130 I=1,NC
      READ(49,1012) (DOC(J),J=1,4)
1012  FORMAT(4A10)
130   WRITE(47,1008) NP,(DOC(J),J=1,4)
C
      WRITE(50,1013) NP
1013  FORMAT(/≠ENTER PLOT SYMBOLS ASSOCIATED WITH GROUP ≠,I3)
      READ(49,*) (MAP(I),I=1,MP)
C
      DO 160 I=1,MP
      DO 140 J=1,1296
140   IF(MAP(I).EQ.ISYMBL(J)) GO TO 150
150   K=K+1
      IGRP(K)=J
160   NAP(K)=NP
      IF(NAP(K).GT.MNAP) MNAP=NAP(K)
170   CONTINUE
C
C*********************************************************************
C
C  THE FOLLOWING CODE ASSIGNS EACH CELL IN THE MAP TO THE GROUP
C  WHICH WAS PREVIOUSLY DESIGNATED.  IF ANY CELL SYMBOLS WERE
C  MISSED IN THE INPUT PROCESS, GROUP ASSIGNMENTS AND DOCUMENTATION
C  WILL BE REQUESTED AND MUST BE PROVIDED IMMEDIATELY AT THE
C  TERMINAL.
C
C*********************************************************************
C
180   READ(JR,1014) ICOL,IROW,(MAP(I),I=1,36)
1014  FORMAT(20I4/8X,18I4)
```

```
      IF(ICOL.EQ.9999) GO TO 250
C
      DO 240 I=1,36
      IF(MAP(I).EQ.0) GO TO 240
      IF(MAP(I).EQ.-999) GO TO 240
      DO 190 L=1,K
190   IF(MAP(I).EQ.IGRP(L)) GO TO 220
      K=K+1
      IGRP(K)=MAP(I)
      WRITE(50,1015) ISYMBL(MAP(I)),MAP(I)
1015  FORMAT(/*WHICH GROUP DOES THE PLOT SYMBOL *,A2,* (*,I4,*)*,
     1* BELONG TO?*)
      READ(49,*) NAP(K)
      IF(NAP(K).LE.MNAP) GO TO 210
      MNAP=NAP(K)
      WRITE(50,1011) NAP(K)
      DO 200 II=1,NC
      READ(49,1012) (DOC(J),J=1,4)
200   WRITE(47,1008) NAP(K),(DOC(J),J=1,4)
210   MAP(I)=NAP(K)
      ICT(NAP(K))=ICT(NAP(K))+1
      GO TO 230
220   MAP(I)=NAP(L)
      ICT(NAP(L))=ICT(NAP(L))+1
230   TOTAC=TOTAC+1.
240   CONTINUE
C
      WRITE(LF,1014) ICOL,IROW,(MAP(I),I=1,36)
      GO TO 130
C
250   WRITE(LF,1016) MNAP
1016  FORMAT(4H9999,I4)
      TOTAC=TOTAC*AREA
C
C***************************************************************
C
C  THE FOLLOWING CODE TITLES THE ASSIGNMENT OR AGGREGATION,
C  CALCULATES STATISTICS FOR THE NEW MAP, AND WRITES THE
C  UPDATED DOCUMENTATION ON FILE.
C
C***************************************************************
C
      IF(KD.EQ.1) GO TO 260
      WRITE(MF,1017) NT,NC,MNAP,(TITLE(K),K=1,5)
1017  FORMAT(2I2,I4,*ASSIGNMENT - *,5A10)
      GO TO 270
C
260   WRITE(MF,1018) NT,NC,MNAP,(TITLE(K),K=1,5)
1018  FORMAT(2I2,I4,*AGGREGATION - *,5A10)
C
270   DO 280 J=1,NT
      READ(IR,1019) (TITLE(I),I=1,6)
280   WRITE(MF,1019) (TITLE(I),I=1,6)
1019  FORMAT(3X,6A10)
C
      REWIND 47
      DO 300 I=1,MNAP
      CAREA=ICT(I)*AREA
      PCAREA=CAREA/TOTAC*100.0
      READ(47,1008) M,(DOC(J),J=1,4)
      WRITE(MF,1020) I,ICT(I),CAREA,PCAREA,M,(DOC(J),J=1,4)
1020  FORMAT(I4,I6,F11.1,F5.1,I4,4A10)
      ICT(I)=0
      IF(NC.EQ.1) GO TO 300
      DO 290 N=2,NC
      READ(47,1008) MM,(DOC(J),J=1,4)
290   WRITE(MF,1021) MM,(DOC(J),J=1,4)
1021  FORMAT(25X,I4,4A10)
300   CONTINUE
C
      WRITE(MF,1022) TOTAC
1022  FORMAT(F11.1)
C
310   RETURN
      END
```

```
      SUBROUTINE REORDER
      DIMENSION ISYM(250),INUM(250),IORDER(1296),CAREA(1296),
     1DOC(4),DOCUM(1296,4),ITYPE(1296)
      COMMON/C/ISYMBL(2496)
      COMMON/F/IR
C
      REWIND IR
      REWIND 45
      REWIND 46
C
      WRITE(50,1001)
1001  FORMAT(/*FOR HOW MANY PARCELS DO YOU WISH TO PREDETERMINE *,
     1*THEIR ORDER IN THE LEGEND?*)
      READ(49,*) MORDER
      IF(MORDER.EQ.0) GO TO 120
C
      WRITE(50,1002)
1002  FORMAT(/*ENTER THE PLOT SYMBOLS (EACH ONE IN PARENTHESES) OF *,
     1*THE PARCELS TO BE ORDERED FIRST.*/*SEPARATE EACH PLOT SYMBOL *,
     2*WITH A COMMA OR BLANK.*)
      READ(49,*) (ISYM(I),I=1,MORDER)
C
      DO 110 I=1,MORDER
      DO 100 J=1,1296
100   IF(ISYM(I).EQ.ISYMBL(J)) GO TO 110
110   INUM(I)=J
C
120   READ(IR,1003) NT,NC,NS
1003  FORMAT(2I2,I4)
      DO 130 I=1,NT
130   READ(IR,1004) IDUM
1004  FORMAT(A1)
C
      IF(NC.GT.1) GO TO 210
      DO 140 I=1,NS
      READ(IR,1005) IORDER(I),CAREA(I),ITYPE(I),(DOC(J),J=1,4)
1005  FORMAT(I4,6X,F11.1,5X,I4,4A10)
      N=ITYPE(I)
      DO 140 J=1,4
140   DOCUM(N,J)=DOC(J)
C
      IF(MORDER.EQ.0) GO TO 160
      DO 150 I=1,MORDER
      N=INUM(I)
      JORD=IORDER(I)
      TAREA=CAREA(I)
      JTYPE=ITYPE(I)
      IORDER(I)=IORDER(N)
      CAREA(I)=CAREA(N)
      ITYPE(I)=ITYPE(N)
      IORDER(N)=JORD
      CAREA(N)=TAREA
150   ITYPE(N)=JTYPE
C
160   NORDER=MORDER+2
      IF(MORDER.EQ.NS.OR.NORDER.GT.NS) GO TO 190
C
170   K=0
      DO 180 J=NORDER,NS
      I=J-1
      IF(CAREA(I).GE.CAREA(J)) GO TO 180
      JORD=IORDER(I)
      TAREA=CAREA(I)
      JTYPE=ITYPE(I)
      IORDER(I)=IORDER(J)
      CAREA(I)=CAREA(J)
      ITYPE(I)=ITYPE(J)
      IORDER(J)=JORD
      CAREA(J)=TAREA
      ITYPE(J)=JTYPE
      K=K+1
180   CONTINUE
      IF(K.NE.0) GO TO 170
C
190   WRITE(45,1006) NS,NS
      WRITE(46,1006) NS,NC
1006  FORMAT(2I4)
C
      DO 200 I=1,NS
```

```
      M=IORDER(I)
      N=ITYPE(I)
      WRITE(45,1006) M,I
200   WRITE(45,1007) N,(DOCUM(N,J),J=1,4)
1007  FORMAT(I4,4A10)
C
      REWIND 45
      REWIND 46
      GO TO 220
C
210   WRITE(50,1008) NC
1008  FORMAT(/≠THIS MAP CONTAINS ≠,I2,≠ LINES OF LEGEND FOR EACH ≠,
     1≠SYMBOL.≠/≠THIS REORDERING ROUTINE CAN ONLY OPERATE WITH 1 LINE ≠,
     2≠PER SYMBOL.≠)
C
220   RETURN
      END
```

```
      OVERLAY(MAPS,6,0)
      PROGRAM HILITE
      DIMENSION TITLE(6),TITL(20,6),JFIX(20),KFIX(20),IDOC(20),
     1JFLT(20),ODC(20,4),FDOC(20,4)
      COMMON/3/ISYMBL(2496)
C
C****************************************************************
C
C   THIS SUBROUTINE WILL HIGHLIGHT CERTAIN ASPECTS OF A MAP'S
C   DOCUMENTATION.  IN A COMBINATION MAP, THE USER CAN FIX ONE OR
C   MORE CHARACTERISTICS WHICH MAKE UP THAT MAP AND LET ONE OR MORE
C   CHARACTERISTICS VARY.  THIS WILL PRODUCE A LEGEND OF MAP CELLS
C   WHICH HAVE THE FIRST SET OF CHARACTERISTICS IN COMMON BUT A
C   DIFFERENT SECOND SET.
C
C****************************************************************
C
100   WRITE(50,1001)
1001  FORMAT(/#DO YOU WISH TO SEE THE DIRECTORY TO AID IN SELECTION #
     1#OF A MAP+#)
      READ(49,1002) JA
1002  FORMAT(A1)
      IF(JA.EQ.1HY) CALL DIRECT
C
      WRITE(50,1003)
1003  FORMAT(/#WHICH MAP DO YOU WISH TO SELECT+#)
      READ(49,*) JB
      IF(JB.EQ.0) GO TO 310
C
110   NR=0
      TACR=0.
      JFX=0
      JFT=0
C
      REWIND 47
      REWIND JB
C
C****************************************************************
C
C   THE FOLLOWING CODE PROVIDES FOR THE SELECTION OF THOSE
C   CHARACTERISTICS WHICH WILL BE FIXED AND THOSE WHICH WILL VARY.
C
C****************************************************************
C
      READ(JB,1004) NT,NC,NS,(TITLE(I),I=1,6)
1004  FORMAT(2I2,I4,6A10)
      DO 120 I=1,NT
      READ(JB,1005) (TITL(I,J),J=1,6)
1005  FORMAT(8X,6A10)
120   WRITE(50,1006) I,(TITL(I,J),J=1,6)
1006  FORMAT(I2,6X,6A10)
C
      WRITE(50,1007)
1007  FORMAT(/#HOW MANY MAP CHARACTERISTICS DO YOU WISH TO FIX+#)
      READ(49,*) JFX
      IF(JFX.EQ.0) GO TO 300
C
      WRITE(50,1008)
1008  FORMAT(/#TO SELECT EACH CHARACTERISTIC, TYPE THE MAP NUMBER #/
     1#FOLLOWED BY THE ORIGINAL MAP SYMBOL IN QUOTATION MARKS.#/
     2#SEPARATE THESE WITH A COMMA OR BLANK,#)
      DO 140 J=1,JFX
      READ(49,*) JC,JD
      JFIX(J)=JC
      DO 130 I=1,300
130   IF(JD.EQ.ISYMBL(I)) KFIX(J)=I
140   CONTINUE
C
150   DO 170 I=1,NC
      DO 160 K=1,JFX
160   IF(I.EQ.JFIX(K)) GO TO 170
      JFT=JFT+1
      JFLT(JFT)=I
170   CONTINUE
C
C****************************************************************
C
C   THE FOLLOWING CODE CHECKS THE ENTIRE MAP LEGEND AGAINST THE
C   LIST OF FIXED CHARACTERISTICS, SELECTING OUT THOSE WHICH MATCH.
```

```
C   SINCE ADDITIONAL STATISTICS ON THIS PARTIAL AREA ARE WANTED,
C   THE ACREAGE IS TALLIED AND THE SELECTED LEGEND IS WRITTEN ON
C   A WORKING FILE.
C
C*****************************************************************
C
      DO 240 IA=1,NS
      READ(JB,1009) J,ITC,GACR,PCACR,IDOC(1),(DOC(1,L),L=1,4)
 1009 FORMAT(I4,I3,F11.1,F5.1,I4,4A10)
      IF(NC.EQ.1) GO TO 185
C
      DO 180 I=2,NC
 180  READ(JB,1010) IDOC(I),(DOC(I,L),L=1,4)
 1010 FORMAT(23X,I4,4A10)
C
 185  DO 190 I=1,JFX
 190  IF(KFIX(I).NE.IDOC(JFIX(I))) GO TO 240
      WRITE(47,1011) J,ITC,GACR,PCACR
 1011 FORMAT(I4,I3,F11.1,F5.1)
      IF(JFT.EQ.0) GO TO 210
C
      DO 200 I=1,JFT
 200  WRITE(47,1012) (DOC(JFLT(I),L),L=1,4)
 1012 FORMAT(4A10)
      GO TO 220
C
 210  WRITE(47,1013)
 1013 FORMAT(1X)
C
 220  TACR=TACR+GACR
      NR=NR+1
      IF(NR.GT.1) GO TO 240
C
      DO 230 I=1,JFX
      DO 230 K=1,4
 230  FDOC(I,K)=DOC(JFIX(I),K)
 240  CONTINUE
C
      IF(NR.EQ.0) GO TO 290
      READ(JB,1014) TOTAC
 1014 FORMAT(F11.1)
      PTACR=TACR/TOTAC*100.0
      KI=50
C
C*****************************************************************
C
C   THE FOLLOWING CODE ALLOWS THE USER TO SELECT OUTPUT TO THE
C   TERMINAL OR LINE PRINTER OUTPUT FILE.  HEADINGS ARE THEN
C   WRITTEN ON THE SELECTED DEVICE.
C
C*****************************************************************
C
      WRITE(50,1015)
 1015 FORMAT(/≠WHERE IS THE OUTPUT TO BE DISPLAYED+≠/
     1≠(1)IMMEDIATE DISPLAY (2)LINE PRINTER OUTPUT FILE≠)
      READ(49,*) KJ
      IF(KJ.EQ.2) KI=43
      REWIND 47
C
      WRITE(KI,1016) JB,(TITLE(I),I=1,6)
 1016 FORMAT(≠1≠/1X,≠DOCUMENTATION HIGHLIGHT FOR MAP ≠,I2/1X,6A10)
C
      DO 250 I=1,NT
 250  WRITE(KI,1017) (TITL(I,J),J=1,6)
 1017 FORMAT(8X,6A10)
      WRITE(KI,1018)
 1018 FORMAT(1X/1X,≠DOCUMENTATION WHICH IS FIXED≠)
      DO 260 I=1,JFX
 260  WRITE(KI,1019) (FDOC(I,J),J=1,4)
 1019 FORMAT(8X,4A10)
C
      WRITE(KI,1020)
 1020 FORMAT(1X//1X,≠SYMBOL≠,2X,≠NO. OF≠,5X,≠ACREAGE≠,2X,≠TOTAL≠,2X,
     1≠PARTIAL≠,2X,≠LEGEND≠/9X,≠CELLS≠,16X,≠PCT.≠,4X,≠PCT.≠)
C
C*****************************************************************
C
C   THE FOLLOWING CODE READS THE STORED LEGEND OFF THE WORKING FILE,
C   CALCULATES SOME ADDITIONAL STATISTICS, AND WRITES ALL OF THIS
```

```
C   INFORMATION ON THE APPROPRIATE OUTPUT DEVICE.
C
C***********************************************************************
C
      DO 280 IA=1,NR
      READ(47,1011)  J,ITC,CACR,PCACR
      QCACR=CACR/TACR*100.0
      READ(47,1012)  (DOC(1,L),L=1,4)
      WRITE(KI,1021)  ISYMBL(J),ITC,CACR,PCACR,QCACR,(DOC(1,L),L=1,4)
 1021 FORMAT(1X/3A,A2,2X,I8,1X,F11.1,2X,F5.1,4X,F5.1,2X,4A10)
C
      IF(JFT.LE.1) GO TO 280
      DO 270 I=2,JFT
      READ(47,1012)  (DOC(I,L),L=1,4)
  270 WRITE(KI,1022)  (DOC(I,L),L=1,4)
 1022 FORMAT(45X,4A10)
  280 CONTINUE
C
      WRITE(KI,1023)  TOTAC,TACR,PTACR
 1023 FORMAT(1X/1X,≠TOTAL ACREAGE OF THE ENTIRE AREA = ≠,F11.1/
     11X,≠TOTAL ACREAGE OF THE PARTIAL AREA = ≠,F11.1/
     21X,≠PERCENTAGE OF THE ENTIRE AREA IN THE PARTIAL AREA = ≠,F5.1)
      GO TO 300
C
C***********************************************************************
C
  290 WRITE(50,1024)
 1024 FORMAT(/≠THERE WERE NO CELLS IN THE COMBINATION WITH THAT ≠
     1≠FIXED DOCUMENTATION.≠)
C
  300 WRITE(50,1025)
 1025 FORMAT(/≠DO YOU WISH TO MAKE ADDITIONAL RUNS OF THIS OPERATION+≠/
     1≠(0)NO (1)YES, FROM THE SAME MAP (2)YES, FROM A DIFFERENT MAP≠)
      READ(49,*) JE
      IF(JE-1)  310,110,100
C
  310 RETURN
      END
```

```
      OVERLAY(MAPS,7,0)
      PROGRAM PARCEL
      DIMENSION MAP1(37),MAP2(37),NAP1(37),NAP2(37),MARGIN(250),
     1IPARCEL(2500),ITYPE(2500),DOC(20,4),IDOC(20),NARGIN(250),
     2JCODE(1296)
      COMMON/A/MFILE,MF,LF
      COMMON/B/ICODE(1296),NS,NC,NT
      COMMON/C/ISYMBL(2496)
      COMMON/D/TITLE(6)
C
C**********************************************************************
C
C  THIS SUBROUTINE DETECTS AND THEN GROUPS THOSE CELLS WHICH ARE
C  ADJACENT TO ONE ANOTHER (NOT DIAGONAL) AND ARE OF THE SAME
C  SYMBOL (HAVE THE SAME CHARACTERISTICS).  IF TWO CELLS SHARE A
C  COMMON SYMBOL BUT ARE NOT CONNECTED BY ADJACENT CELLS OF THE
C  SAME SYMBOL, THESE TWO CELLS WILL BE PLACED INTO DIFFERENT
C  GROUPS (PARCELS).  THIS PROCESS WILL BREAK THE ENTIRE MAP INTO
C  PARCELS, SOME OF WHICH MAY SHARE COMMON CHARACTERISTICS BUT
C  ARE SPATIALLY SEPARATED.
C
C**********************************************************************
C
      N=0
      NS=0
C
      WRITE(50,1001)
 1001 FORMAT(/*DO YOU WISH TO SEE THE DIRECTORY+*)
      READ(49,1002) IS
 1002 FORMAT(A1)
      IF(IS.EQ.1HY) CALL DIRECT
C
      WRITE(50,1003)
 1003 FORMAT(/*WHICH MAP DO YOU WISH TO GROUP INTO PARCELS+*)
      READ(49,*) IR
      IF(IR.EQ.0) GO TO 370
C
      JR=IR+20
      REWIND IR
      REWIND JR
      REWIND 46
      REWIND 47
C
C**********************************************************************
C
C  THIS CODE IS USED WHENEVER THE NUMBER OF AVAILABLE MAPS IS
C  INCREASED (EITHER A NEW MAP IS INPUT OR ONE IS CREATED BY A
C  MAP OPERATION).  THE NUMBER OF AVAILABLE MAPS IS INCREASED,
C  THE FILE NUMBERS FOR THE NEW MAP ARE DESIGNATED, AND A CHECK
C  IS MADE TO SEE IF THE NEW MAP PUSHES THE NUMBER OF AVAILABLE
C  MAPS BEYOND THE FILE SPACE FOR TWENTY MAPS.  IF THE NEW MAP
C  IS NUMBER TWENTY-ONE, SUBROUTINE EXCESS IS CALLED TO HANDLE
C  THE PROBLEM.
C
C**********************************************************************
C
      MFILE=MFILE+1
      MF=MFILE
      LF=MF+20
      IF(MFILE.GT.20) CALL EXCESS
C
C**********************************************************************
C
C  THE FOLLOWING CODE CHECKS EACH MAP CELL AGAINST THE CELLS ABOVE
C  AND TO THE LEFT.  IF IT MATCHES EITHER ONE IT IS PLACED IN THAT
C  PARCEL.  IF IT MATCHES BOTH AND THEY ARE IN DIFFERENT PARCELS,
C  THE CONNECTION IS NOTED AND THE CELL IS PLACED IN ONE OF THE
C  PARCELS.  THE PARCEL NUMBER OF EACH CELL IS WRITTEN OUT.
C
C**********************************************************************
C
  100 READ(JR,1004) ICOL,IROW,(MAP2(I),I=2,37)
 1004 FORMAT(2I4/6X,18I4)
      IF(ICOL.EQ.9999) GO TO 220
      IF(ICOL.GT.1) GO TO 105
      MARGIN(IROW)=0
      NARGIN(IROW)=0
C
  105 MAP2(1)=MARGIN(IROW)
```

```
      NAP2(1)=NARGIN(IROW)
C
      DO 200 I=2,37
      IF(MAP2(I).EQ.0) GO TO 170
      IF(MAP2(I).EQ.-998) GO TO 180
      IF(MAP2(I).EQ.-999) GO TO 190
C
      K=0
      IF(IROW.EQ.1) GO TO 110
      IF(MAP2(I).NE.MAP1(I)) GO TO 110
      K=K+1
110   IF(MAP2(I).NE.MAP2(I-1)) GO TO 120
      K=K+2
C
120   IF(K.EQ.0) GO TO 130
      IF(K.EQ.1) NAP2(I)=IPARCEL(NAP1(I))
      IF(K.EQ.2) NAP2(I)=IPARCEL(NAP2(I-1))
      IF(K.EQ.3) GO TO 135
      GO TO 200
C
130   N=N+1
      ITYPE(N)=MAP2(I)
      IPARCEL(N)=N
      NAP2(I)=N
      GO TO 200
135   NP1=NAP1(I)
      NP2=NAP2(I-1)
140   IF(IPARCEL(NP1).EQ.NP1) GO TO 145
      NP1=IPARCEL(NP1)
      GO TO 140
145   IF(IPARCEL(NP2).EQ.NP2) GO TO 150
      NP2=IPARCEL(NP2)
      GO TO 145
150   IF(NP1-NP2) 155,160,165
155   NAP2(I)=NP1
      IPARCEL(NP2)=NP1
      GO TO 200
160   NAP2(I)=NP1
      GO TO 200
165   NAP2(I)=NP2
      IPARCEL(NP1)=NP2
      GO TO 200
170   NAP2(I)=0
      GO TO 200
180   NAP2(I)=-998
      GO TO 200
190   NAP2(I)=-999
200   CONTINUE
C
      MARGIN(IROW)=MAP2(37)
      NARGIN(IROW)=NAP2(37)
      DO 210 I=1,37
      MAP1(I)=MAP2(I)
210   NAP1(I)=NAP2(I)
C
      WRITE(LF,1004) ICOL,IROW,(NAP2(I),I=2,37)
      GO TO 100
C
220   WRITE(LF,1005) N
1005  FORMAT(' 99999',I4)
C
C*****************************************************************
C
C   THE FOLLOWING CODE CHECKS FOR ALL CONNECTIONS PREVIOUSLY DETECTED,
C   READS THE PARCEL NUMBERS, AND WRITES THE MAP ONTO A WORKING FILE
C   GIVING ANY CONNECTED PARCELS THE SAME SYMBOL.  THIS SETS THE MAP
C   UP TO BE RECODED BY SUBROUTINE RECODE ACCORDING TO THE ORDER OF
C   THE LEGEND.
C
C*****************************************************************
C
      DO 230 I=1,N
      IF(IPARCEL(I).NE.I) GO TO 230
      NS=NS+1
      JCODE(NS)=I
230   CONTINUE
C
      REWIND LF
240   READ(LF,1004) ICOL,IROW,(MAP1(I),I=1,36)
```

```
      IF(ICOL.EQ.9999) GO TO 310
C
      DO 300 I=1,36
      IF(MAP1(I).EQ.0) GO TO 270
      IF(MAP1(I).EQ.-998) GO TO 280
      IF(MAP1(I).EQ.-999) GO TO 290
      MP1=MAP1(I)
250   IF(IPARCEL(MP1).EQ.MP1) GO TO 260
      MP1=IPARCEL(MP1)
      GO TO 250
260   MAP1(I)=ISYMBL(MP1)
      GO TO 300
270   MAP1(I)=2H
      GO TO 300
280   MAP1(I)=2H++
      GO TO 300
290   MAP1(I)=2H**
300   CONTINUE
C
      WRITE(46,1006) ICOL,IROW,(MAP1(I),I=1,36)
1006  FORMAT(2I4,36A2)
      GO TO 240
C
310   WRITE(45,1005) NS
C
C*****************************************************************
C
C   THE FOLLOWING CODE SETS UP THE TITLES, WRITES THE LEGEND FOR EACH
C   PARCEL IN ORDER ON A WORKING FILE, STORES THE INFORMATION TO PROPERLY
C   RECODE THE MAP, AND CALLS SUBROUTINE RECODE TO COMPLETE THE PARCEL
C   CREATION PROCESS.
C
C*****************************************************************
C
      READ(IR,1007) NT,NC,ND
1007  FORMAT(2I2,I4)
      WRITE(50,1008)
1008  FORMAT(/*ENTER THE NAME OF THIS NEW MAP (50 CHARACTERS)*)
      READ(49,1009) (TITLE(K),K=1,5)
1009  FORMAT(5A10)
      WRITE(47,1010) NT,NC,NS,(TITLE(K),K=1,5)
1010  FORMAT(2I2,I4,*PARCELS - *,5A10)
C
      DO 320 I=1,NT
      READ(IR,1011) (TITLE(K),K=1,6)
320   WRITE(47,1011) (TITLE(K),K=1,6)
1011  FORMAT(3X,6A10)
C
      M=0
      DO 360 I=1,ND
      READ(IR,1012) IC,IDOC(1),(DOC(1,L),L=1,4)
1012  FORMAT(I4,24X,I4,4A10)
      IF(NC.EQ.1) GO TO 340
      DO 330 J=2,NC
330   READ(IR,1013) IDOC(J),(DOC(J,L),L=1,4)
1013  FORMAT(28X,I4,4A10)
C
340   DO 360 K=1,NS
      JCD=JCODE(K)
      IF(IC.NE.ITYPE(JCD)) GO TO 360
C
      DO 350 J=1,NC
350   WRITE(47,1012) JCD,IDOC(J),(DOC(J,L),L=1,4)
      M=M+1
      ICODE(M)=ISYMBL(JCD)
360   CONTINUE
C
      REWIND 46
      CALL RECODE
C
370   RETURN
      END
```

```
      OVERLAY(MAPS,10,0)
      PROGRAM UPDATE
      DIMENSION TITLE(6),DOC(20,4),TRANS(8),ID(20),ISYM(50),IPOS(20)
      COMMON/A/MFILE,MF,LF
      COMMON/C/ISYMBL(2496)
C
C*****************************************************************
C
C  THIS SUBROUTINE ALLOWS THE USER TO UPDATE THE TITLES AND/OR LEGENDS
C  OF ANY MAP.  THE UPDATE MAY INCLUDE AN INCREASE OR DECREASE IN THE
C  NUMBER OF LINES OF EACH AS WELL AS A CHANGE IN THEIR CONTENTS.
C
C*****************************************************************
C
100   WRITE(50,1001)
1001  FORMAT(/≠DO YOU WISH TO SEE THE DIRECTORY+≠)
      READ(49,1002) IS
1002  FORMAT(A1)
      IF(IS.EQ.1HY) CALL DIRECT
C
      WRITE(50,1003)
1003  FORMAT(/≠WHICH MAP DO YOU WISH TO SELECT+≠)
      READ(49,*) IR
      IF(IR.EQ.0) GO TO 350
C
110   REWIND 46
      REWIND IR
C
C*****************************************************************
C
C  THE FOLLOWING CODE ALLOWS THE USER TO SELECT AN UPDATE OPTION AND
C  ASKS TO HAVE THE APPROPRIATE INFORMATION INPUT FROM THE TERMINAL.
C  ANY DOCUMENTATION WHICH IS NOT TO BE CHANGED IS SIMPLY MOVED INTACT
C  TO THE WORKING FILE.
C
C*****************************************************************
C
      READ(IR,1004) NT,NC,NS,(TITLE(I),I=1,6)
1004  FORMAT(2I2,I4,6A10)
      WRITE(50,1005) NT,NC
1005  FORMAT(/≠THERE ARE CURRENTLY ≠,I2,≠ TITLES AND ≠,I2,
     1≠ LINES OF DOCUMENTATION.≠/≠WHICH DO YOU WISH TO UPDATE+≠/
     2≠(0)NEITHER (1)TITLES (2)ENTIRE LEGEND (3)PART OF LEGEND≠/
     3≠AFTER CURRENT INFORMATION IS LISTED, ENTER UPDATED ≠,
     4≠INFORMATION.≠)
      READ(49,*) IT
      IF(IT.EQ.0) GO TO 350
C
      NP=1
      NQ=NT
      NR=NC
C
      DO 120 I=1,20
120   IPOS(I)=I
C
      IF(IT.NE.1) GO TO 130
      WRITE(50,1006)
1006  FORMAT(/≠HOW MANY UPDATED TITLES DO YOU WANT+≠)
      READ(49,*) NQ
      NP=NC*NS+1
C
130   IF(IT.NE.2) GO TO 140
      WRITE(50,1007)
1007  FORMAT(/≠DO YOU WISH SIMPLY TO REDUCE THE NUMBER OF LINES ≠/
     1≠OF LEGEND WITHOUT ENTERING NEW INFORMATION+≠)
      READ(49,1002) IU
      WRITE(50,1008)
1008  FORMAT(/≠HOW MANY UPDATED LINES OF LEGEND DO YOU WANT+≠)
      READ(49,*) NR
      IF(IU.EQ.1HN) GO TO 140
C
      WRITE(50,1009)
1009  FORMAT(/≠ARE THE LINE(S) ALWAYS TO BE SELECTED ≠/
     1≠FROM THE SAME POSITION IN THE LEGEND+≠)
      READ(49,1002) IW
C
140   WRITE(45,1004) NQ,NR,NS,(TITLE(I),I=1,6)
      WRITE(50,1010)
1010  FORMAT(/≠TITLES≠)
```

```
      DO 150 J=1,NT
      READ(IR,1011) (TITLE(I),I=1,6)
      IF(IT.NE.1) WRITE(46,1011) (TITLE(I),I=1,6)
150   WRITE(50,1011) (TITLE(I),I=1,6)
1011  FORMAT(6X,6A10)
C
      IF(IT.EQ.1) GO TO 290
      IF(IT.EQ.2) GO TO 160
      WRITE(50,1012)
1012  FORMAT(/*HOW MANY LEGENDS DO YOU WISH TO UPDATE+*)
      READ(49,*) NL
      IF(NL.EQ.0) GO TO 340
C
      WRITE(50,1013)
1013  FORMAT(/*LIST THE SYMBOLS ASSOCIATED WITH THE LEGENDS TO *,
     1*BE UPDATED.*/*PLACE EACH SYMBOL IN QUOTATION MARKS*/
     2*AND SEPARATE THE SYMBOLS WITH A COMMA OR BLANK.*)
      READ(49,*) (ISYM(I),I=1,NL)
      GO TO 180
C
160   IF(IU.EQ.1HN) GO TO 180
      IF(IW.EQ.1HN) GO TO 170
      WRITE(50,1014) NR
1014  FORMAT(/*SELECT THE *,I2,* LINE(S) OF LEGEND TO BE RETAINED.*/
     1*TYPE IN THEIR POSITION IN THE CURRENT LEGEND.*/
     2*SEPARATE THE NUMBERS WITH A COMMA.*)
      READ(49,*) (IPOS(I),I=1,NR)
      GO TO 180
C
170   WRITE(50,1015) NR
1015  FORMAT(/*WHEN SELECTING THE *,I2,* LINE(S) OF LEGEND,*/
     1*TYPE IN THEIR POSITION AFTER EACH CURRENT LEGEND IS LISTED.*/
     2*SEPARATE THE NUMBERS WITH A COMMA.*)
C
180   DO 280 K=1,NS
      READ(IR,1016) J,ITC,CACR,PCACR,ID(1),(DOC(1,L),L=1,4)
1016  FORMAT(I4,I8,F11.1,F5.1,I4,4A10)
      IF(NC.EQ.1) GO TO 200
      DO 190 I=2,NC
190   READ(IR,1017) ID(I),(DOC(I,L),L=1,4)
1017  FORMAT(28X,I4,4A10)
C
200   IF(IT.EQ.2) GO TO 220
      DO 210 I=1,NL
210   IF(ISYMB_(J).EQ.ISYM(I)) GO TO 220
      GO TO 280
C
220   IF(IU.EQ.1HY.AND.IW.EQ.1HY) GO TO 260
      DO 230 I=1,NC
230   WRITE(50,1018) (DOC(I,L),L=1,4)
1018  FORMAT(4A10)
C
      IF(IU.EQ.1HN) GO TO 240
      READ(49,*) (IPOS(I),I=1,NR)
      GO TO 260
C
240   DO 250 I=1,NR
      ID(I)=K
250   READ(49,1018) (DOC(I,L),L=1,4)
C
260   M=IPOS(1)
      WRITE(46,1016) J,ITC,CACR,PCACR,ID(M),(DOC(M,L),L=1,4)
      IF(NR.EQ.1) GO TO 280
      DO 270 I=2,NR
      M=IPOS(I)
270   WRITE(46,1017) ID(M),(DOC(M,L),L=1,4)
280   CONTINUE
      GO TO 310
C
290   DO 300 J=1,NQ
      READ(49,1019) (TITLE(I),I=1,6)
1019  FORMAT(6A10)
300   WRITE(46,1011) (TITLE(I),I=1,6)
C
310   DO 320 J=1,NP
      READ(IR,1020) (TRANS(I),I=1,8)
320   WRITE(46,1020) (TRANS(I),I=1,8)
1020  FORMAT(8A10)
C
```

```
     1*PLACING THE PARCEL TO BE ELIMINATED FIRST,*/*FOLLOWED *,
     2*BY THE PARCEL IT IS TO BE INCLUDED IN.*/
     3*SEPARATE THESE WITH A COMMA OR BLANK.*)
      DO 230 I=1,NPAR
      READ(49,*) IPAR(I),JPAR(I)
      DO 230 J=1,1296
      IF(IPAR(I).EQ.ISYMBL(J)) IPAR(I)=J
230   IF(JPAR(I).EQ.ISYMBL(J)) JPAR(I)=J
C
C*****************************************************************
C
C   THE FOLLOWING CODE READS THE MAP, LOCATES CELLS IN THE PARCELS TO
C   BE ELIMINATED, CHANGES THOSE CELL SYMBOLS, AND WRITES THE MAP OUT
C   ON A WORKING FILE.  THE DOCUMENTATION, MINUS THAT ASSOCIATED WITH
C   THE ELIMINATED PARCELS, IS ALSO TRANSFERRED TO A WORKING FILE.
C   THIS INFORMATION IS PROVIDED TO SUBROUTINE RECODE SO THE MAP AND
C   DOCUMENTATION CAN BE UPDATED.
C
C*****************************************************************
C
240   READ(JR,1012) ICOL,IROW,(MAP(I),I=1,36)
      IF(ICOL.EQ.9999) GO TO 300
C
      DO 290 I=1,36
      IF(MAP(I).EQ.0) GO TO 270
      IF(MAP(I).EQ.-999) GO TO 280
      DO 250 J=1,NPAR
250   IF(MAP(I).EQ.IPAR(J)) GO TO 260
      MAP(I)=ISYMBL(MAP(I))
      GO TO 290
260   MAP(I)=ISYMBL(JPAR(J))
      GO TO 290
270   MAP(I)=2H
      GO TO 290
280   MAP(I)=2H**
290   CONTINUE
C
      WRITE(46,1013) ICOL,IROW,(MAP(I),I=1,36)
      GO TO 240
C
300   NR=NS
      NS=NS-NPAR
      WRITE(46,1014) NS
      WRITE(47,1015) NT,NC,NS,(TITLE(I),I=1,5)
      DO 310 J=1,NT
      READ(IR,1016) (TRANS(I),I=1,8)
310   WRITE(47,1016) (TRANS(I),I=1,8)
C
      DO 360 J=1,NR
      READ(IR,1020) I,IA(1),(DOC(1,K),K=1,4)
1020  FORMAT(I4,24X,I4,4A10)
      IF(NC.EQ.1) GO TO 330
      DO 320 L=2,NC
320   READ(IR,1021) IA(L),(DOC(L,K),K=1,4)
1021  FORMAT(28X,I4,4A10)
C
330   DO 340 L=1,NPAR
340   IF(I.EQ.IPAR(L)) GO TO 360
      DO 350 L=1,NC
350   WRITE(47,1021) IA(L),(DOC(L,K),K=1,4)
      M=M+1
      ICODE(M)=ISYMBL(I)
360   CONTINUE
      GO TO 640
C
C*****************************************************************
C
C   THE FOLLOWING CODE ALLOWS THE USER TO INDICATE THE PARCEL TO BE
C   SPLIT, THE NUMBER OF NEW PARCELS WANTED, AND THE DIRECTION OF THE
C   SPLIT.  THE MAP IS THEN READ IN TO FIND THE LOCATION OF EACH CELL
C   IN THE PARCEL.
C
C*****************************************************************
C
370   WRITE(50,1022)
1022  FORMAT(/*WHICH PARCEL DO YOU WANT TO DIVIDE, INTO HOW *,
     1*MANY PIECES, AND IN WHAT DIRECTION+*//
     2*LIST THE PARCEL SYMBOL IN QUOTATION MARKS, THE NUMBER OF PIECES,*
     3/*AND THE DIRECTION OF DIVISION    (1) NORTH-SOUTH (2) EAST-WEST.*/
```

```
     4≠SEPARATE EACH OF THESE WITH A COMMA OR BLANK.≠)
      READ(49,*) ISYM,NPC,IDIR
      DO 380 J=1,1296
380   IF(ISYM.EQ.ISYMBL(J)) GO TO 390
390   KPAR=J
C
400   READ(JR,1012) ICOL,IROW,(MAP(I),I=1,36)
      IF(ICOL.EQ.9999) GO TO 430
C
      DO 420 I=1,36
      IF(MAP(I).NE.KPAR) GO TO 420
      IF(IDIR.EQ.2) GO TO 410
      J=(ICOL-1)*36+I
      LOC(J)=LOC(J)+1
      GO TO 420
410   LOC(IROW)=LOC(IROW)+1
420   CONTINUE
      GO TO 400
C
C*****************************************************************
C
C  THE FOLLOWING CODE TRANSFERS THE DOCUMENTATION OF THE MAP ADDING
C  TO IT THE ADDITIONAL LINES OF LEGEND FOR THOSE PARCELS CREATED BY
C  THE SPLIT.  INFORMATION IS ALSO STORED FOR USE IN SUBROUTINE RECODE.
C
C*****************************************************************
C
430    NR=NS
       NS=NS+NPC-1
       WRITE(47,1015) NT,NC,NS,(TITLE(I),I=1,5)
       DO 440 J=1,NT
       READ(IR,1016) (TRANS(I),I=1,8)
440    WRITE(47,1016) (TRANS(I),I=1,8)
C
       DO 510 J=1,NR
       READ(IR,1020) I,IA(1),(DOC(1,K),K=1,4)
       IF(NC.EQ.1) GO TO 460
       DO 450 L=2,NC
450    READ(IR,1021) IA(L),(DOC(L,K),K=1,4)
C
460    DO 470 L=1,NC
470    WRITE(47,1021) IA(L),(DOC(L,K),K=1,4)
       M=M+1
       ICODE(M)=ISYMBL(I)
C
       IF(I.NE.KPAR) GO TO 510
480    DO 500 LL=2,NPC
       DO 490 L=1,NC
490    WRITE(47,1021) IA(L),(DOC(L,K),K=1,4)
       NQ=NR+LL-1
       M=M+1
500    ICODE(M)=ISYMBL(NQ)
510    CONTINUE
C
C*****************************************************************
C
C  THE FOLLOWING CODE USES THE CELL LOCATIONS PREVIOUSLY COLLECTED TO
C  CALCULATE WHERE THE PARCEL SPLITS WILL BE MADE.  IT THEN READS THE
C  MAP AND REASSIGNS THE CELLS OF THE INDICATED PARCEL USING THE
C  CALCULATIONS OF THE SPLITS.
C
C*****************************************************************
C
       I=1
       DO 520 J=2,250
       LOC(J)=LOC(J)+LOC(I)
       IF(LOC(J).GT.0.AND.LOC(J).EQ.LOC(I)) GO TO 530
520    I=J
C
530    NDIV=LOC(J)/NPC
       REWIND JR
C
540    READ(JR,1012) ICOL,IROW,(MAP(I),I=1,36)
       IF(ICOL.EQ.9999) GO TO 620
C
       DO 610 I=1,36
       IF(MAP(I).EQ.0) GO TO 560
       IF(MAP(I).EQ.-999) GO TO 570
       IF(MAP(I).EQ.KPAR) GO TO 580
```

```
 550    MAP(I)=ISYMBL(MAP(I))
        GO TO 610
 560    MAP(I)=2H
        GO TO 610
 570    MAP(I)=2H**
        GO TO 610
 580    IF(IDIR.EQ.2) GO TO 590
        J=(ICOL-1)*36+I
        LDIV=LOC(J)/NDIV
        GO TO 600
 590    LDIV=LOC(IROW)/NDIV
 600    IF(LDIV.EQ.0) GO TO 550
        NQ=NR+LDIV
        IF(NQ.GT.NS) NQ=NS
        MAP(I)=ISYMBL(NQ)
 610    CONTINUE
C
        WRITE(45,1013) ICOL,IROW,(MAP(I),I=1,36)
        GO TO 540
C
 620    WRITE(46,1014) NS
        DO 630 I=1,250
 630    LOC(I)=0
C
C*********************************************************************
C
C   THE SUBROUTINE RECODE IS CALLED FOR ALL THREE OPERATIONS.  IT
C   CHANGES ACREAGES CAUSED BY CELLS MOVING FROM PARCEL TO PARCEL,
C   ELIMINATES DOCUMENTATION FOR THOSE PARCELS AGGREGATED WITH OTHERS,
C   AND EXPANDS DOCUMENTATION FOR THOSE PARCELS CREATED BY A SPLIT.
C
C*********************************************************************
C
 640    REWIND 46
        CALL RECODE
C
        IR=MF
        JR=LF
C
        WRITE(50,1024)
 1024   FORMAT(/*DO YOU WISH TO PERFORM ANY MORE OPERATIONS *,
       1*ON THE PARCELS OF THIS MAP+*)
        READ(49,1002) IS
        IF(IS.EQ.1HY) GO TO 100
C
 650    RETURN
        END
```

```
      OVERLAY(MAPS,11,0)
      PROGRAM MANIP
      DIMENSION JCOL(20),JROW(20),JSPOT(20),JSYM(20),JCELL(20),
     1MAP(36),TRANS(8),IPAR(20),JPAR(20),IA(20),DOC(20,4),LOC(250)
      COMMON/A/MFILE,MF,LF
      COMMON/B/ICODE(1296),NS,NC,NT
      COMMON/C/ISYMBL(2496)
      COMMON/D/TITLE(6)
      DATA LOC/500*0/
C
C***********************************************************************
C
C  THIS SUBROUTINE ALLOWS THE USER TO MANIPULATE MAP CELLS BUT DOES
C  NOT ALLOW CONTENT CHANGES TO THE LEGEND.  THERE ARE THREE TYPES
C  OF OPERATIONS PROVIDED:  INDIVIDUAL CELLS CAN BE MOVED FROM ONE
C  PARCEL TO ANOTHER, CELLS OF ONE PARCEL CAN BE AGGREGATED WITH
C  THOSE OF ANOTHER, AND THE CELLS IN ONE PARCEL CAN BE SPLIT INTO
C  TWO OR MORE PARCELS.
C
C***********************************************************************
C
      WRITE(50,1001)
 1001 FORMAT(/≠DO YOU WISH TO SEE THE DIRECTORY+≠)
      READ(49,1002) IS
 1002 FORMAT(A1)
      IF(IS.EQ.1HY) CALL DIRECT
C
      WRITE(50,1003)
 1003 FORMAT(/≠WHICH MAP DO YOU WISH TO SELECT+≠)
      READ(49,*) IR
      IF(IR.EQ.0) GO TO 550
C
C***********************************************************************
C
C  THIS CODE IS USED WHENEVER THE NUMBER OF AVAILABLE MAPS IS
C  INCREASED (EITHER A NEW MAP IS INPUT OR ONE IS CREATED BY A
C  MAP OPERATION).   THE NUMBER OF AVAILABLE MAPS IS INCREASED,
C  THE FILE NUMBERS FOR THE NEW MAP ARE DESIGNATED, AND A CHECK
C  IS MADE TO SEE IF THE NEW MAP PUSHES THE NUMBER OF AVAILABLE
C  MAPS BEYOND THE FILE SPACE FOR TWENTY MAPS.  IF THE NEW MAP
C  IS NUMBER TWENTY-ONE, SUBROUTINE EXCESS IS CALLED TO HANDLE
C  THE PROBLEM.
C
C***********************************************************************
C
      MFILE=MFILE+1
      MF=MFILE
      LF=MF+20
      IF(MFILE.GT.20) CALL EXCESS
C
C***********************************************************************
C
      WRITE(50,1005)
 1005 FORMAT(/≠ENTER THE NAME OF THE NEW MAP (45 CHARACTERS)≠)
      READ(49,1006) (TITLE(K),K=1,5)
 1006 FORMAT(5A10)
C
      JR=IR+20
  100 REWIND IR
      REWIND JR
      REWIND 46
      REWIND 47
      M=0
C
      READ(IR,1007) NT,NC,NS
 1007 FORMAT(2I2,I4)
C
      WRITE(50,1008)
 1008 FORMAT(/≠WHICH OPERATION DO YOU WISH TO USE+≠/
     1≠(1) MOVE INDIVIDUAL CELLS FROM ONE PARCEL TO ANOTHER≠/
     2≠(2) AGGREGATE SOME PARCELS WITH OTHERS≠/
     3≠(3) DIVIDE A SINGLE PARCEL INTO TWO OR MORE PARCELS≠)
      READ(49,*) IO
      IF(IO-2) 110,220,370
C
C***********************************************************************
C
C  THE FOLLOWING CODE ALLOWS THE USER TO LOCATE INDIVIDUAL CELLS IN A
C  MAP THEN DESIGNATE THE SYMBOLS FOR THOSE CELLS.
```

```
C
C*********************************************************************
C
110   WRITE(50,1009)
1009  FORMAT(/≠HOW MANY INDIVIDUAL CELLS DO YOU WANT TO BE ≠,
     1≠MOVED FROM PARCEL TO PARCEL+≠)
      READ(49,*) NCELL
C
      WRITE(50,1010)
1010  FORMAT(/≠LIST THE COLUMN NUMBER, ROW NUMBER, LOCATION ≠,
     1≠IN THE ROW,≠/≠AND THE NEW PARCEL SYMBOL (IN QUOTATION MARKS) ≠,
     2≠FOR EACH CELL BEING MOVED.≠/
     3≠SEPARATE EACH OF THESE WITH A COMMA OR BLANK.≠)
      DO 130 I=1,NCELL
      READ(49,*) JCOL(I),JROW(I),JSPOT(I),JSYM(I)
      DO 120 J=1,1296
120   IF(JSYM(I).EQ.ISYMBL(J)) GO TO 130
130   JCELL(I)=J
C
C*********************************************************************
C
C  THE FOLLOWING CODE READS THE MAP, LOCATES THE INDICATED CELLS,
C  CHANGES THOSE CELL SYMBOLS, AND WRITES THE MAP OUT ON A WORKING
C  FILE.  THE DOCUMENTATION IS ALSO TRANSFERRED TO A WORKING FILE
C  AND INFORMATION IS PROVIDED TO SUBROUTINE RECODE SO CHANGES IN
C  THE ACREAGE FOR EACH SYMBOL CAN BE UPDATED.
C
C*********************************************************************
C
140   READ(JR,1012) ICOL,IROW,(MAP(I),I=1,36)
1012  FORMAT(2I4/8X,18I4)
      IF(ICOL.EQ.9999) GO TO 190
C
      DO 150 J=1,NCELL
150   IF(JCOL(J).EQ.ICOL.AND.JROW(J).EQ.IROW) MAP(JSPOT(J))=JCELL(J)
C
      DO 180 I=1,36
      IF(MAP(I).EQ.0) GO TO 160
      IF(MAP(I).EQ.-999) GO TO 170
      MAP(I)=ISYMBL(MAP(I))
      GO TO 180
160   MAP(I)=2H
      GO TO 180
170   MAP(I)=2H**
180   CONTINUE
C
      WRITE(45,1013) ICOL,IROW,(MAP(I),I=1,36)
1013  FORMAT(2I4,36A2)
      GO TO 140
C
190   WRITE(45,1014) NS
1014  FORMAT(4H9999,I4)
C
      WRITE(47,1015) NT,NC,NS,(TITLE(I),I=1,5)
1015  FORMAT(2I2,I4,≠MANIPULATION - ≠,5A10)
      NN=NT+NC*NS
      DO 200 J=1,NN
      READ(IR,1016) (TRANS(I),I=1,8)
200   WRITE(47,1016) (TRANS(I),I=1,8)
1016  FORMAT(8A10)
C
      DO 210 I=1,NS
210   ICODE(I)=ISYMBL(I)
      GO TO 640
C
C*********************************************************************
C
C  THE FOLLOWING CODE ALLOWS THE USER TO INDICATE PARCELS TO BE
C  AGGREGATED WITH OTHER PARCELS.
C
C*********************************************************************
C
220   WRITE(50,1017)
1017  FORMAT(/≠HOW MANY PARCELS DO YOU WANT TO AGGREGATE WITH ≠,
     1≠OTHERS+≠)
      READ(49,*) NPAR
C
      WRITE(50,1018)
1018  FORMAT(/≠LIST EACH PAIR OF PARCEL SYMBOLS IN QUOTATION MARKS,≠/
```

```
C*********************************************************************
C
C   THE FOLLOWING CODE SIMPLY TRANSFERS EVERY LINE OF THE WORKING FILE
C   BACK ONTO THE ORIGINAL MAP DOCUMENTATION FILE.
C
C*********************************************************************
C
      REWIND 46
      REWIND IR
C
      NP=NR*NS+NQ+2
      DO 330 J=1,NP
      READ(46,1020) (TRANS(I),I=1,8)
330   WRITE(IR,1020) (TRANS(I),I=1,8)
C
C*********************************************************************
C
340   WRITE(50,1021)
1021  FORMAT(/≠DO YOU WISH TO MAKE ADDITIONAL RUNS OF THIS OPERATION+≠/
     1≠(0)NO (1)YES, FROM THE SAME MAP (2)YES, FROM A DIFFERENT MAP≠)
      READ(49,*) JE
      IF(JE-1) 350,110,100
C
350   RETURN
      END
```

APPENDIX C

```
      PROGRAM REDUCE(INPUT,OUTPUT,TAPE10,TAPE11,TAPE12,TAPE13,
     1TAPE14=OUTPUT,TAPE15=INPUT,TAPE16)
      DIMENSION MAP1(38),MAP2(38),MAP3(38),K1(38),K2(38),K3(38),
     1IT(2,5,250),NC(2,5,250),L(4),M(4),ISYMBL(1296),NCELL(1296),
     2ISYM(1296),TRANS(8),TITLE(8),DOC(20,4)
      DATA AREA/21.33/
C
      TC=0.
      SC=0.
      RC=0.
      EC=0.
      UC=0.
      ND=0
      NE=0
C
      READ(16,1001) (ISYMBL(I),I=1,1296)
1001  FORMAT(10A2)
      REWIND 16
C
      READ(10,1002) NT,NCC,NS
1002  FORMAT(2I2,I4)
      DO 100 I=1,NT
100   READ(10,1003) IDUM
1003  FORMAT(A1)
C
      DO 120 I=1,NS
      READ(10,1004) J,NCELL(J)
1004  FORMAT(I4,I8)
      IF(J.GT.NE) NE=J
      IF(NCC.EQ.1) GO TO 120
      DO 110 K=2,NCC
110   READ(10,1003) IDUM
120   CONTINUE
C
      READ(10,1005) TOTAC
1005  FORMAT(F11.1)
C
      WRITE(14,1006)
1006  FORMAT(/*ARE THERE ANY SYMBOLS WHICH SHOULD NOT BE *,
     1*REDUCED IN DETAIL.*)
      READ(15,1007) IA
1007  FORMAT(A1)
      IF(IA.EQ.1HN) GO TO 150
C
      WRITE(14,1008)
1008  FORMAT(/*HOW MANY SYMBOLS SHOULD BE REMOVED FROM*,
     1* CONSIDERATION.*)
      READ(15,*) NSYM
C
      WRITE(14,1009)
1009  FORMAT(/*LIST EACH SYMBOL IN QUOTATION MARKS.*/
     1*SEPARATE EACH SYMBOL WITH A COMMA.*)
      READ(15,*) (ISYM(I),I=1,NSYM)
C
      DO 140 I=1,NSYM
      DO 130 J=1,1296
130   IF(ISYM(I).EQ.ISYMBL(J)) GO TO 140
140   ISYM(I)=J
C
150   READ(11,1010) ICOL2,IROW2,(MAP2(I),I=2,37)
1010  FORMAT(2OI4/8X,18I4)
      IF(ICOL2.EQ.9999) GO TO 190
      IF(ICOL2.GT.1) GO TO 160
C
      IT(1,1,IROW2)=0
      NC(1,1,IROW2)=0
      IT(2,1,IROW2)=0
      NC(2,1,IROW2)=0
C
160   MAP2(1)=IT(1,ICOL2,IROW2)
      K2(1)=NC(1,ICOL2,IROW2)
C
      DO 180 I=2,37
      K2(I)=0
      IF(MAP2(I).EQ.0) GO TO 180
      IF(MAP2(I).EQ.-998) GO TO 180
      IF(MAP2(I).EQ.-999) GO TO 180
C
      IF(IROW2.EQ.1) GO TO 170
```

```
      IF(MAP2(I).NE.MAP1(I)) GO TO 170
      K1(I)=K1(I)+1
      K2(I)=K2(I)+1
C
170   IF(MAP2(I).NE.MAP2(I-1)) GO TO 180
      K2(I)=K2(I)+1
      K2(I-1)=K2(I-1)+1
180   CONTINUE
C
190   IF(IROW2.EQ.1.AND.ICOL2.EQ.1) GO TO 210
      IF(ICOL1.EQ.1) GO TO 200
      IT(1,ICOL1,IROW1)=MAP1(1)
      NC(1,ICOL1,IROW1)=K1(1)
      IT(2,ICOL1,IROW1)=MAP1(2)
      NC(2,ICOL1,IROW1)=K1(2)
C
200   IC=ICOL1+1
      IT(1,IC,IROW1)=MAP1(37)
      NC(1,IC,IROW1)=K1(37)
C
      WRITE(15,1011) ICOL1,IROW1,(K1(I),I=2,37)
1011  FORMAT(2I4,36I2)
      IF(ICOL2.EQ.9999) GO TO 230
C
210   DO 220 I=1,37
      MAP1(I)=MAP2(I)
220   K1(I)=K2(I)
      ICOL1=ICOL2
      IROW1=IROW2
      GO TO 190
C
230   REWIND 10
      REWIND 11
      REWIND 15
C
240   READ(11,1010) ICOL3,IROW3,(MAP3(I),I=2,37)
      IF(ICOL3.EQ.9999) GO TO 250
      READ(16,1011) ICOL3,IROW3,(K3(I),I=2,37)
C
      MAP3(1)=IT(1,ICOL3,IROW3)
      K3(1)=NC(1,ICOL3,IROW3)
C
      IC=ICOL3+1
      K3(37)=NC(1,IC,IROW3)
      MAP3(38)=IT(2,IC,IROW3)
      K3(38)=NC(2,IC,IROW3)
C
      IF(ICOL3.EQ.1.AND.IROW3.EQ.1) GO TO 360
C
250   DO 340 I=2,37
      IF(MAP2(I).EQ.0) GO TO 340
      IF(MAP2(I).EQ.-998) GO TO 340
      IF(MAP2(I).EQ.-999) GO TO 340
      TC=TC+1.
C
      IF(K2(I).GT.0) GO TO 340
      SC=SC+1.
      IF(IA.EQ.1HN) GO TO 270
      DO 260 J=1,NSYM
260   IF(MAP2(I).EQ.ISYM(J)) GO TO 340
270   EC=EC+1.
      DO 280 J=1,4
280   L(J)=100
C
      IF(IROW3.EQ.2) GO TO 290
      IF(K1(I).EQ.0) GO TO 290
      L(1)=IABS(MAP2(I)-MAP1(I))
      M(1)=MAP1(I)
C
290   IF(K2(I+1).EQ.0) GO TO 300
      L(2)=IABS(MAP2(I)-MAP2(I+1))
      M(2)=MAP2(I+1)
C
300   IF(ICOL3.EQ.9999) GO TO 310
      IF(IROW3.EQ.1) GO TO 310
      IF(K3(I).EQ.0) GO TO 310
      L(3)=IABS(MAP2(I)-MAP3(I))
      M(3)=MAP3(I)
C
```

```
310   IF(K2(I-1).EQ.0) GO TO 320
      L(4)=IABS(MAP2(I)-MAP2(I-1))
      M(4)=MAP2(I-1)
C
320   LM=L(1)
      MM=M(1)
      DO 330 J=2,4
      IF(L(J).GE.LM) GO TO 330
      LM=L(J)
      MM=M(J)
330   CONTINUE
C
      IF(LM.EQ.100) GO TO 340
      MP=MAP2(I)
      MAP2(I)=MM
      NCELL(MP)=NCELL(MP)-1
      NCELL(MM)=NCELL(MM)+1
      RC=RC+1
340   CONTINUE
C
      WRITE(13,1010) ICOL2,IROW2,(MAP2(I),I=2,37)
      IF(ICOL3.EQ.9999) GO TO 380
C
      DO 350 I=1,38
      MAP1(I)=MAP2(I)
350   K1(I)=K2(I)
      ICOL1=ICOL2
      IROW1=IROW2
360   DO 370 I=1,38
      MAP2(I)=MAP3(I)
370   K2(I)=K3(I)
      ICOL2=ICOL3
      IROW2=IROW3
      GO TO 240
C
380   WRITE(13,1012) ICOL3,IROW3
1012  FORMAT(2I4)
C
      DO 390 I=1,NE
      IF(NCELL(I).EQ.0) GO TO 390
      ND=ND+1
390   CONTINUE
C
      READ(10,1013) NT,NCC,NS,(TITLE(I),I=1,6)
      WRITE(12,1013) NT,NCC,ND,(TITLE(I),I=1,6)
1013  FORMAT(2I2,I4,6A10)
      DO 400 K=1,NT
      READ(10,1014) (TRANS(I),I=1,8)
400   WRITE(12,1014) (TRANS(I),I=1,8)
1014  FORMAT(8A10)
C
      DO 440 I=1,NS
      READ(10,1015) J,K1(1),(DOC(1,N),N=1,4)
1015  FORMAT(I4,24X,I4,4A10)
      IF(NCC.EQ.1) GO TO 420
      DO 410 NN=2,NCC
410   READ(10,1016) K1(NN),(DOC(NN,N),N=1,4)
1016  FORMAT(28X,I4,4A10)
C
420   IF(NCELL(J).EQ.0) GO TO 440
      ACR=FLOAT(NCELL(J))*AREA
      PCACR=ACR/TOTAC*100.
      WRITE(12,1017) J,NCELL(J),ACR,PCACR,K1(1),(DOC(1,N),N=1,4)
1017  FORMAT(I4,I8,F11.1,F5.1,I4,4A10)
      IF(NCC.EQ.1) GO TO 440
      DO 430 NN=2,NCC
430   WRITE(12,1016) K1(NN),(DOC(NN,N),N=1,4)
440   CONTINUE
C
      READ(10,1014) (TRANS(I),I=1,8)
      WRITE(12,1014) (TRANS(I),I=1,8)
C
      UC=SC-EC
      PC1=SC/TC*100.
      PC2=EC/TC*100.
      PC3=UC/TC*100.
      PC4=RC/TC*100.
      PC5=EC/SC*100.
      PC6=UC/SC*100.
```

```
      PC7=RC/SC*100.
      PC8=RC/EC*100.
C
      WRITE(14,1018) TC,SC,EC,UC,RC,PC1,PC2,PC3,PC4,PC5,PC6,PC7,PC8
1018  FORMAT(/≠TOTAL NUMBER OF CELLS = ≠,F7.1/
     1≠TOTAL NUMBER OF SINGLE CELLS = ≠,F7.1/
     2≠NUMBER OF SINGLE CELLS ELIGIBLE FOR REASSIGNMENT = ≠,F7.1/
     3≠NUMBER OF SINGLE CELLS REMOVED FROM REASSIGNMENT = ≠,F7.1/
     4≠NUMBER OF REASSIGNED CELLS = ≠,F7.1/
     5≠PERCENTAGE OF CELLS WHICH ARE SINGLE = ≠,F6.2/
     6≠PERCENTAGE OF CELLS ELIGIBLE FOR REASSIGMENT = ≠,F6.2/
     7≠PERCENTAGE OF CELLS REMOVED FROM REASSIGNMENT = ≠,F6.2/
     8≠PERCENTAGE OF CELLS WHICH WERE REASSIGNED = ≠,F6.2/
     9≠PERCENTAGE OF SINGLE CELLS ELIGIBLE FOR REASSIGNMENT = ≠,F6.2/
     A≠PERCENTAGE OF SINGLE CELLS REMOVED FROM REASSIGNMENT = ≠,F6.2/
     B≠PERCENTAGE OF SINGLE CELLS WHICH WERE REASSIGNED = ≠,F6.2/
     C≠PERCENTAGE OF ELIGIBLE CELLS WHICH WERE REASSIGNED = ≠,F6.2)
C
      STOP
      END
```

APPENDIX D

```
      PROGRAM ADJ(TAPE1,TAPE21,TAPE45)
      DIMENSION MARGIN(250),MAP1(37),MAP2(37),TITLE(6),
     1IADJ(1296,10),JADJ(1296,10),IOUT(100),JOUT(100)
      COMMON/A/NOVER,IOVER(125),NEXC(125),IEXC(125,10),
     1JEXC(125,10),IA(1296)
      DATA JADJ/12960*0/,NOVER/0/,IA/1296*0/
C
      REWIND 1
      REWIND 21
      REWIND 45
C
      WRITE(45,1001)
 1001 FORMAT(≠ADJACENT CELLS≠)
      READ(1,1002) NT,NC,NS,(TITLE(I),I=1,6)
 1002 FORMAT(2I2,I4,6A10)
      WRITE(45,1003) NT,NC,NS,(TITLE(I),I=1,6)
 1003 FORMAT(1X,I2,1X,I2,1X,I4,1X,6A10)
      DO 100 J=1,NT
      READ(1,1004) (TITLE(I),I=1,6)
  100 WRITE(45,1004) (TITLE(I),I=1,6)
 1004 FORMAT(8X,6A10)
C
  110 READ(21,1005) ICOL,IROW,(MAP2(I),I=2,37)
 1005 FORMAT(20I4/6X,16I4)
C
      IF(ICOL.EQ.9999) GO TO 230
      IF(ICOL.EQ.1) MARGIN(IROW)=0
      MAP2(1)=MARGIN(IROW)
C
      DO 210 I=2,37
      M1=MAP2(I)
      I1=IA(M1)
      IF(M1.EQ.0.OR.M1.EQ.-998.OR.M1.EQ.-999) GO TO 210
      ITEST=1
      IF(IROW.EQ.1) GO TO 200
      M2=MAP1(I)
      I2=IA(M2)
C
  120 IF(M2.EQ.0.OR.M2.EQ.-998.OR.M2.EQ.-999) GO TO 200
      IF(M1.EQ.M2) GO TO 200
      IF(I1.EQ.0) GO TO 140
      IF(I1.GT.10) I1=10
      DO 130 J=1,I1
  130 IF(IADJ(M1,J).EQ.M2) GO TO 150
      IF(I1.LT.10) GO TO 140
C
      CALL SPILL(M1,M2)
      GO TO 150
C
  140 IA(M1)=IA(M1)+1
      I1=IA(M1)
      IADJ(M1,I1)=M2
      J=I1
C
  150 JADJ(M1,J)=JADJ(M1,J)+1
C
  160 IF(I2.EQ.0) GO TO 180
      IF(I2.GT.10) I2=10
      DO 170 J=1,I2
  170 IF(IADJ(M2,J).EQ.M1) GO TO 190
      IF(I2.LT.10) GO TO 180
C
      CALL SPILL(M2,M1)
      GO TO 200
C
  180 IA(M2)=IA(M2)+1
      I2=IA(M2)
      IADJ(M2,I2)=M1
      J=I2
C
  190 JADJ(M2,J)=JADJ(M2,J)+1
C
  200 IF(ITEST.EQ.2) GO TO 210
      M2=MAP2(I-1)
      I2=IA(M2)
      ITEST=2
      GO TO 120
  210 CONTINUE
C
```

```
      MARGIN(IROW)=MAP2(37)
      DO 220 I=1,37
  220 MAP1(I)=MAP2(I)
      GO TO 110
C
  230 DO 290 I=1,NS
      N=IA(I)
      IF(N.LE.10) GO TO 270
      L=10
C
      DO 250 J=1,NOVER
      IF(IOVER(J).NE.I) GO TO 250
      NEX=NEXC(J)
      DO 240 K=1,NEX
      L=L+1
      IOUT(L)=IEXC(J,K)
      JOUT(L)=JEXC(J,K)
  240 IF(L.EQ.IA(I)) GO TO 260
  250 CONTINUE
C
  260 N=10
  270 DO 280 J=1,N
      IOUT(J)=IADJ(I,J)
  280 JOUT(J)=JADJ(I,J)
C
      N=IA(I)
  290 WRITE(6,1006) I,N,(IOUT(J),JOUT(J),J=1,N)
 1006 FORMAT(32I4)
C
      STOP
      END
```

```
      SUBROUTINE SPILL(M1,M2)
      COMMON/A/NOVER,IOVER(125),NEXC(125),IEXC(125,10),
     1JEXC(125,10),IA(1296)
C
      IF(NOVER.EQ.0) GO TO 120
      DO 110 I=1,NOVER
      NEX=NEXC(I)
      IF(IOVER(I).NE.M1) GO TO 110
C
      DO 100 J=1,NEX
100   IF(IEXC(I,J).EQ.M2) GO TO 140
      IF(NEX.LT.10) GO TO 130
110   CONTINUE
C
120   NOVER=NOVER+1
      IOVER(NOVER)=M1
      NEXC(NOVER)=1
      IEXC(NOVER,1)=M2
      JEXC(NOVER,1)=1
      IA(M1)=IA(M1)+1
      RETURN
C
130   NEX=NEX+1
      NEXC(I)=NEX
      IEXC(I,NEX)=M2
      JEXC(I,NEX)=1
      IA(M1)=IA(M1)+1
      RETURN
C
140   JEXC(I,J)=JEXC(I,J)+1
      RETURN
      END
```

APPENDIX E

```
      PROGRAM HEURIST(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,
     1TAPE4,TAPE5,TAPE6,TAPE7,TAPE8,TAPE9,TAPE10,
     2TAPE11,TAPE12,TAPE13,TAPE14=INPUT,TAPE15=OUTPUT)
      DIMENSION ACRE(1296),ITYPE(1296),ALLOC(40,8),TOLER(40,8),
     1ICNFLT(8,8),IN(100),JN(100),IADJ(1296),IASSIGN(1296),
     2IDELETE(1296,8)
      DATA TOL/.05/,IWT/1/,JWT/10/,KWT/100/,IDELETE/10368*0/,
     1IASSIGN/1296*0/,IADJ/1296*0/
C
      IR=6
      JR=7
      KS=5
      ICHK=0
C
      READ(1,1001) NT,NPAR
1001  FORMAT(I2,2X,I4)
      DO 100 I=1,NT
100   READ(1,1002) IDUM
1002  FORMAT(A1)
      DO 110 I=1,NPAR
110   READ(1,1003) ACRE(I),ITYPE(I)
1003  FORMAT(12X,F11.1,5X,I4)
C
      READ(2,*) NTYPE,NLU
      DO 120 I=1,NTYPE
      READ(2,*) (ALLOC(I,J),J=1,NLU)
      DO 120 J=1,NLU
120   TOLER(I,J)=TOL*ALLOC(I,J)
C
      READ(3,*) NLU
      DO 160 I=1,NLU
      READ(3,*) (ICNFLT(I,J),J=1,NLU)
      DO 160 J=1,NLU
      IF(ICNFLT(I,J)) 130,140,150
130   ICNFLT(I,J)=-1
      GO TO 160
140   ICNFLT(I,J)=0
      GO TO 160
150   ICNFLT(I,J)=1
160   CONTINUE
C
170   REWIND 4
      REWIND KS
      ICNT=0
      TACRE=0.
      IPAR=0
C
      READ(4,1004) NT,NPAR
1004  FORMAT(1X/1X,I2,4X,I4)
      DO 180 I=1,NT
180   READ(4,1002) IDUM
C
      READ(KS,1005) NLU,NPREDET
1005  FORMAT(2I4)
C
      WRITE(IR,1005) NLU,NPAR
C
190   IPAR=IPAR+1
      JPAR=IPAR+1
      ILU=0
      IFLEX=0
      TDIF=0.
      N=ITYPE(IPAR)
C
      READ(4,1006) IA,IB,(IN(J),JN(J),J=1,IB)
1006  FORMAT(32I4)
      DO 200 J=1,IB
      KN=IN(J)
200   IADJ(KN)=JN(J)
C
      IF(IPAR.GT.NPREDET) GO TO 210
      READ(KS,*) IASG,IAS
      IF(IAS.EQ.0) GO TO 210
      ILU=IAS
      IF(ICHK.GT.0) GO TO 350
      GO TO 320
C
210   DO 310 I=1,NLU
      M3=0
```

```
      JFLEX=0
      IF(ICHK.GE.1) GO TO 220
      IF(IDELETE(IPAR,I).LT.0) GO TO 310
220   IF(ICHK.GE.3) GO TO 230
      IF(ALLOC(N,I).LT.0) GO TO 310
230   DIFF=ALLOC(N,I)+TOLER(N,I)-ACRE(IPAR)
      DIF=ALLOC(N,I)-ACRE(IPAR)
      IF(ICHK.GE.2) GO TO 240
      IF(DIFF.LT.0) GO TO 310
240   IF(IPAR.EQ.NPAR) GO TO 290
C
      IF(ICHK.GT.0) JPAR=1
      DO 280 J=JPAR,NPAR
      M1=0
      M2=0
      DO 270 K=1,NLU
      IF(IDELETE(J,K).LT.0) GO TO 270
      IF(IADJ(J).EQ.0) GO TO 250
      IF(ICNFLT(I,K)) 270,250,260
250   M1=M1+1
      GO TO 270
260   M2=M2+1
270   CONTINUE
      IF(M1.GT.0.OR.M2.GT.0) M3=M3+1
280   JFLEX=JFLEX+IWT*M1+JWT*M2
C
      JFLEX=JFLEX+KWT*M3
      IF(JFLEX-IFLEX) 310,290,300
290   IF(ILU.EQ.0) GO TO 300
      IF(TDIF.GT.DIF) GO TO 310
300   ILU=I
      IFLEX=JFLEX
      TDIF=DIF
310   CONTINUE
C
      IF(ILU.EQ.0) GO TO 360
320   DO 340 J=JPAR,NPAR
      IF(IADJ(J).EQ.0) GO TO 340
      DO 330 K=1,NLU
330   IDELETE(J,K)=IDELETE(J,K)+ICNFLT(ILU,K)
340   CONTINUE
      ALLOC(N,ILU)=ALLOC(N,ILU)-ACRE(IPAR)
C
350   ICNT=ICNT+1
      TACRE=TACRE+ACRE(IPAR)
C
360   IASSIGN(IPAR)=ILU
      DO 370 J=1,NPAR
370   IADJ(J)=0
C
      WRITE(IR,1005) IPAR,IASSIGN(IPAR)
      IF(IPAR.LT.NPAR) GO TO 190
C
      WRITE(JR,1005) NLU,NTYPE
      DO 380 I=1,NTYPE
380   WRITE(JR,1007) (ALLOC(I,J),J=1,NLU)
1007  FORMAT(8F12.1)
C
      JCNT=NPAR-ICNT
      WRITE(15,1008) ICHK,ICNT,JCNT,TACRE
1008  FORMAT(/≠NUMBER OF RELAXED CONSTRAINTS = ≠,I1/
     1≠NUMBER OF ASSIGNED MANAGEMENT UNITS = ≠,I4/
     2≠NUMBER OF UNASSIGNED MANAGEMENT UNITS = ≠,I4/
     3≠TOTAL ACREAGE ASSIGNED = ≠,F11.1)
C
      KS=IR
      IR=IR+2
      JR=JR+2
      IF(ICHK.EQ.3) GO TO 390
C
      WRITE(15,1009)
1009  FORMAT(/≠DO YOU WISH TO RELAX SOME OF THE ASSIGNMENT ≠,
     1≠CONSTRAINTS≠/≠SO THAT MORE ASSIGNMENTS WILL BE MADE+≠)
      READ(14,1002) NY
      ICHK=ICHK+1
      IF(NY.EQ.1HY) GO TO 170
C
390   STOP
      END
```

APPENDIX F

```
      PROGRAM DETECT(INPUT=65,OUTPUT=65,TAPE9=INPUT,
     1TAPE10=OUTPUT,TAPE1=513,TAPE2=513,TAPE3=513,
     2TAPE4=513,TAPE5=513,TAPE6=513,TAPE7=513,TAPE8=513,
     3TAPE11=513,TAPE12=513,TAPE13=513,TAPE14=513,TAPE15=513,
     4TAPE16=513,TAPE17=513,TAPE18=513,TAPE19=513,TAPE20=513,
     5TAPE21=513,TAPE22=513,TAPE23=513,TAPE24=513,TAPE25=513)
      INTEGER ASN,UB
      DIMENSION ISN(2),TRANS(10),IADJ(200),JADJ(200),LUC(20,20),
     1TITLE(6)
      COMMON/A/ISYMBL(1296),ASN(1296),USES(20,4),ITC(1296),
     1ACR(1296),PCACR(1296),DOC(1296,4)
C
      WRITE(10,1001)
 1001 FORMAT(/≠CONFLICT DETECTION ROUTINE≠/1X,≠CREATED BY R. HAGESTEDT≠/
     18X,≠DEC. 1978≠)
C
      WRITE(10,1002)
 1002 FORMAT(/≠THE FOLLOWING FILE NUMBERS ARE SET.≠/
     1≠THE USER MUST BE CONSISTENT WITH THEIR DESIGNATION.≠/
     23X,≠SYMBOL LIST (1)≠/3X,≠ACTIVITY ASSIGNMENTS (2)≠/
     33X,≠ACTIVITY NAME LIST (3)≠/3X,≠ACTIVITY SPILLOVER MATRIX (4)≠/
     43X,≠ADJACENT MANAGEMENT UNITS (5)≠/
     53X,≠MANAGEMENT UNIT MAP DOCUMENTATION (6)≠/
     63X,≠MINIMUM MANAGEMENT UNIT ACREGES (7)≠)
C
      REWIND 1
      REWIND 2
      REWIND 3
      REWIND 4
      REWIND 5
      REWIND 6
      REWIND 7
C
      WRITE(10,1003)
 1003 FORMAT(/≠DO YOU WISH TO INPUT ACTIVITY NAMES, MINIMUM ACREAGES,≠/
     1≠OR SPILLOVERS ONTO A FILE+≠)
      READ(9,1004) IA
 1004 FORMAT(A1)
      IF(IA.EQ.1HY) CALL IN
C
      READ(1,1005) (ISYMBL(I),I=1,1296)
 1005 FORMAT(18A2)
C
      READ(4,1006) NLU
 1006 FORMAT(I4)
      READ(4,*) ((LUC(I,J),J=1,NLU),I=1,NLU)
C
      READ(6,1007) NT,NO,NP
 1007 FORMAT(2I2,I4)
      IF(NO.GE.2) GO TO 270
      DO 100 J=1,NT
  100 READ(6,1004) IDUM
C
      DO 110 I=1,NP
  110 READ(6,1008) ITC(I),ACR(I),PCACR(I),ASN(I),(DOC(I,J),J=1,4)
 1008 FORMAT(4X,I8,F11.1,F5.1,I4,4A10)
C
      WRITE(10,1009)
 1009 FORMAT(/≠WHICH SPILLOVER STRATEGY IS BEING USED+ (1 OR 2)≠)
      READ(9,*) IZ
      IF(IZ.EQ.2) GO TO 125
C
      READ(2,1010) NLU,NPAR
 1010 FORMAT(2I4)
      DO 120 I=1,NPAR
  120 READ(2,1010) IPAR,ASN(IPAR)
C
  125 READ(3,*) NLU
      DO 130 K=1,NLU
  130 READ(3,1011) I,(USES(I,L),L=1,4)
 1011 FORMAT(I4,4A10)
      IF(IZ.EQ.1) GO TO 140
C
      WRITE(10,1012)
 1012 FORMAT(/≠DO YOU WISH TO DETECT MINIMUM MANAGEMENT UNIT ≠,
     1≠SIZE VIOLATIONS+≠)
      READ(9,1004) IA
      IF(IA.EQ.1HY) CALL MINIMUM
C
```

```
140     READ(5,1013) NTITLE,NDOC,NPAR,(TITLE(I),I=1,6)
1013    FORMAT(1X/1X,I2,1X,I2,1X,I4,1X,6A10)
        NF=15+NLU
        DO 150 LF=13,NF
150     WRITE(LF,1014) (TITLE(I),I=1,6)
1014    FORMAT(≠1DETECTION OF DETRIMENTAL SPILLOVERS≠/≠-≠,3X,6A10)
        DO 160 J=1,NTITLE
        READ(5,1015) (TITLE(I),I=1,6)
        DO 160 LF=13,NF
160     WRITE(LF,1015) (TITLE(I),I=1,6)
1015    FORMAT(8X,6A10)
C
        DO 170 I=1,3
        LF=12+I
170     WRITE(LF,1016) I
1016    FORMAT(≠-DETRIMENTAL SPILLOVERS WITH SEVERITY ≠,I1)
        DO 180 I=1,NLU
        LF=15+I
180     WRITE(LF,1017) I,(USES(I,J),J=1,4)
1017    FORMAT(≠-DETRIMENTAL SPILLOVERS ASSOCIATED WITH ACTIVITY ≠,
       1I2/5X,4A10)
C
        DO 230 K=1,NPAR
        READ(5,1018) IPAR,NADJ,(IADJ(J),JADJ(J),J=1,NADJ)
1018    FORMAT(32I4)
C
        DO 230 J=1,NADJ
        IF(IPAR.GT.IADJ(J)) GO TO 230
        ISN(1)=ASN(K)
        ISN(2)=ASN(IADJ(J))
C
        ISVR=LUC(ISN(1),ISN(2))
        IF(ISVR.GE.0) GO TO 230
        ISVR=IABS(ISVR)
        DO 220 N=1,3
        IF(N.EQ.3) GO TO 190
        LF=15+ISN(N)
        GO TO 200
190     LF=12+ISVR
C
200     WRITE(LF,1019) ISVR,JADJ(J)
1019    FORMAT(≠-SEVERITY = ≠,I1,5X,≠BOUNDARY LENGTH = ≠,I5)
C
        M=IPAR
        DO 220 I=1,2
        IF(I.EQ.2) M=IADJ(J)
        IF(IZ.EQ.2) GO TO 210
C
        NN=ISN(I)
        WRITE(LF,1020) ISYMBL(M),M,(USES(NN,L),L=1,4)
1020    FORMAT(≠0≠,5X,≠MANAGEMENT UNIT ≠,A2,≠ (≠,I4,≠)≠,5X,4A10)
        WRITE(LF,1021) ITC(M),ACR(M),PCACR(M)
1021    FORMAT(11X,I8,≠ CELLS≠,5X,F11.1,≠ ACRES≠,5X,F5.1,≠ PERCENT≠)
        WRITE(LF,1022) (DOC(M,L),L=1,4)
1022    FORMAT(22X,4A10)
        GO TO 220
C
210     WRITE(LF,1020) ISYMBL(M),M,(DOC(M,L),L=1,4)
        WRITE(LF,1021) ITC(M),ACR(M),PCACR(M)
220     CONTINUE
230     CONTINUE
C
        WRITE(10,1023)
1023    FORMAT(/≠WHICH SPILLOVER LISTS DO YOU WANT ON OUTPUT FILE 8+≠/
       1≠(0)NONE≠/≠(1)SEVERITY LISTS ONLY≠/≠(2)ACTIVITY LISTS ONLY≠/
       2≠(3)SEVERITY AND ACTIVITY LISTS≠)
        READ(9,*) IA
        IF(IA.EQ.0) GO TO 280
C
        LB=13
        UB=15+NLU
        IF(IA.EQ.1) UB=15
        IF(IA.EQ.2) LB=16
C
        DO 260 I=LB,UB
        REWIND I
240     READ(I,1024) (TRANS(J),J=1,10)
1024    FORMAT(10A10)
        IF(EOF(I)) 260,250
```

```
250   WRITE(8,1024) (TRANS(J),J=1,10)
      GO TO 240
260   CONTINUE
C
      WRITE(10,1025)
1025  FORMAT(/≠SPILLOVERS HAVE BEEN WRITTEN ON FILE 8.≠/
     1≠REMEMBER TO ROUTE THAT FILE.≠)
      GO TO 230
C
270   WRITE(10,1026) ND
1026  FORMAT(/≠THERE ARE ≠,I2,≠ LINES OF LEGEND IN THIS MAP.≠/
     1≠THIS ROUTINE CAN HANDLE ONLY 1 LINE OF LEGEND.≠)
C
280   STOP
      END
```

```
      SUBROUTINE IN
      DIMENSION USE(4),LUCFLT(20,20)
      DATA LUCFLT/400*0/
C
      WRITE(10,1000)
 1000 FORMAT(/≠ENTER THE NUMBER OF ACTIVITIES≠)
      READ(9,*) NLU
C
      WRITE(10,1001)
 1001 FORMAT(/≠DO YOU WISH TO INPUT LAND-USE NAMES+≠)
      READ(9,1002) IA
 1002 FORMAT(A1)
      IF(IA.E2.1HN) GO TO 110
C
      WRITE(3,1003) NLU
 1003 FORMAT(I+,3X,≠1≠)
C
      WRITE(10,1004)
 1004 FORMAT(/≠ENTER THE NAME OF EACH ACTIVITY AFTER THE NUMBER ≠,
     1≠IS LISTED.≠/≠LIMIT IS 40 CHARACTERS.≠)
      DO 100 I=1,NLU
      WRITE(10,1005) I
 1005 FORMAT(I+)
      READ(9,1006) (USE(J),J=1,4)
 1006 FORMAT(4A10)
  100 WRITE(3,1007) I,(USE(J),J=1,4)
 1007 FORMAT(I+,4A10)
      REWIND 3
C
  110 WRITE(10,1008)
 1008 FORMAT(/≠DO YOU WISH TO INPUT MINIMUM MANAGEMENT UNIT ≠,
     1≠ACREAGES+≠)
      READ(9,1002) IA
      IF(IA.EQ.1HN) GO TO 130
C
      WRITE(7,1005) NLU
      WRITE(10,1009)
 1009 FORMAT(/≠ENTER THE MINIMUM ACREAGE OF MANAGEMENT UNITS AFTER ≠/
     1≠EACH ACTIVITY NUMBER IS LISTED.≠)
      DO 120 I=1,NLU
      WRITE(10,1005) I
      READ(9,*) MINACR
      ACRMIN=FLOAT(MINACR)
  120 WRITE(7,1010) ACRMIN
 1010 FORMAT(F11.1)
      REWIND 7
C
  130 WRITE(10,1011)
 1011 FORMAT(/≠DO YOU WISH TO INPUT LAND-USE SPILLOVERS+≠)
      READ(9,1002) IA
      IF(IA.E2.1HN) GO TO 170
      WRITE(4,1005) NLU
C
      WRITE(10,1012)
 1012 FORMAT(/≠ENTER EACH SPILLOVER BY LISTING THE TWO ACTIVITY ≠,
     1≠NUMBERS≠/≠FOLLOWED BY THE TYPE (+,-) AND LEVEL OF THE SPILLOVER.≠
     2/≠SEPARATE EACH OF THE NUMBERS WITH A COMMA.≠/≠TERMINATE THE ≠,
     3≠ENTRY SEQUENCE WITH A SERIES OF ZEROS.≠)
  140 READ(9,*) I,J,K
      IF(I.EQ.0) GO TO 150
C
      LUCFLT(I,J)=K
      LUCFLT(J,I)=K
      GO TO 140
C
  150 DO 160 I=1,NLU
  160 WRITE(4,1013) (LUCFLT(I,J),J=1,NLU)
 1013 FORMAT(20I3)
      REWIND 4
C
  170 RETURN
      END
```

```
      SUBROUTINE MINIMUM
      INTEGER ASN
      DIMENSION TITLE(6),ACRMIN(20),IADJ(200),JADJ(200)
      COMMON/A/ISYMBL(1296),ASN(1296),USES(20,4),ITC(1296),
     1ACR(1296),PCACR(1296),ODC(1296,4)
C
      READ(5,1001) NTITLE,NODC,NPAR,(TITLE(I),I=1,6)
1001  FORMAT(1X/1X,I2,1X,I2,1X,I4,1X,6A10)
      WRITE(11,1002) (TITLE(I),I=1,6)
1002  FORMAT(≠1DETECTION OF MANAGEMENT UNIT SIZE VIOLATIONS≠/
     1≠-≠,3X,6A10)
C
      DO 100 J=1,NTITLE
      READ(5,1003) (TITLE(I),I=1,6)
100   WRITE(11,1003) (TITLE(I),I=1,6)
1003  FORMAT(8X,6A10)
C
      WRITE(10,1004)
1004  FORMAT(/≠DO YOU WISH TO SCREEN THE MANAGEMENT UNITS WITH ≠,
     1≠MINIMUM ACREAGES+≠)
      READ(9,1005) IA
1005  FORMAT(A1)
      IF(IA.EQ.1HN) GO TO 130
C
      READ(7,*) NLU
      DO 110 I=1,NLU
110   READ(7,*) ACRMIN(I)
C
      WRITE(11,1006)
1006  FORMAT(≠-MINIMUM ACREAGE≠,3X,≠ACTIVITY≠)
      DO 120 I=1,NLU
120   WRITE(11,1007) ACRMIN(I),(USES(I,J),J=1,4)
1007  FORMAT(3X,F11.1,6X,4A10)
C
130   DO 160 I=1,NPAR
      READ(5,1008) IPAR,NADJ,(IADJ(J),JADJ(J),J=1,NADJ)
1008  FORMAT(32I4)
C
      IF(IA.EQ.1HN) GO TO 140
      J=ASN(I)
      IF(ACR(I).GT.ACRMIN(J)) GO TO 160
C
140   WRITE(11,1009) ISYMBL(I),I,(ODC(I,K),K=1,4)
1009  FORMAT(≠0≠,5X,≠MANAGEMENT UNIT ≠,A2,≠ (≠,I4,≠) ≠,5X,4A10)
      WRITE(11,1010) ITC(I),ACR(I),PCACR(I)
1010  FORMAT(11X,I8,≠ CELLS≠,5X,F11.1,≠ ACRES≠,5X,F5.1,≠ PERCENT≠/
     18X,≠ADJACENT MANAGEMENT UNITS AND THEIR ASSIGNED ACTIVITIES≠)
C
      IF(NADJ.EQ.0) GO TO 160
      DO 150 L=1,NADJ
      J=IADJ(L)
150   WRITE(11,1011) ISYMBL(J),J,(ODC(J,K),K=1,4)
1011  FORMAT(10X,A2,≠ (≠,I4,≠)≠,2X,4A10)
160   CONTINUE
C
      WRITE(10,1012)
1012  FORMAT(/≠MINIMUM MANAGEMENT UNIT SIZE VIOLATIONS ARE ≠,
     1≠ON FILE 11.≠)
      REWIND 5
      RETURN
      END
```