WRC RESEARCH REPORT NO. 163

GENERATING ALTERNATIVE SOLUTIONS FOR DYNAMIC PROGRAMMING MODELS OF WATER RESOURCES PROBLEMS

Lewis D. Hopkins

Institute for Environmental Studies
and
Departments of Landscape Architecture
and Urban and Regional Planning

and

E. Downey Brill, Jr.
Benedict Wong

Departments of Civil Engineering
and Institute for Environmental Studies

FINAL REPORT

Project No. S-078-ILL

UNIVERSITY OF ILLINOIS
WATER RESOURCES CENTER
2535 Hydrosystems Laboratory
208 North Romine
Urbana, Illinois 61801

August 1981

ABSTRACT

GENERATING ALTERNATIVE SOLUTIONS FOR DYNAMIC PROGRAMMING MODELS OF WATER RESOURCES PROBLEMS

A technique is presented to generate alternatives that are different from each other, but good with respect to modeled objectives, for problems that can be modeled by dynamic programming. The technique is compared to other possible approaches, and relevant concepts of difference among alternatives are discussed. Application to a floodplain management model shows that the technique can produce sets of different alternatives for water resources problems.

Hopkins, Lewis D.; Brill, E. Downey, Jr.; and Wong, Benedict.
GENERATING ALTERNATIVE SOLUTIONS FOR DYNAMIC PROGRAMMING MODELS OF WATER RESOURCES PROBLEMS
Completion Report for Project S-078-ILL supported under the University of Illinois Water Resources Center.
KEYWORDS--*generating alternatives/*optimization models/floodplain management/*planning

ACKNOWLEDGEMENTS

Part of the work reported in this paper was conducted while the first author was a visitor at the Department of Town and Regional Planning at the University of Sheffield. The development of the floodplain example was supported by grant S-078-ILL from the Water Resources Center at the University of Illinois.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. Modeling to Generate Alternatives

Many applications of optimum-seeking models require a combination of human and computer effort. The arguments are familiar: models are incomplete; utility functions are difficult to specify; the range of alternatives is not known; exact algorithmic solution is not possible; the understanding of the problem changes as potential solutions are considered. It has been argued elsewhere that one appropriate response to such 'ill-defined' problems is 'Modeling to Generate Alternatives' (MGA). This approach is characterized by using models to generate sets of alternatives that are different from each other, but feasible given the physical structure of the system of concern and good with respect to target values for well-understood objectives (Brill, 1979; Hopkins, 1975). The argument, in simplest terms, is that in order for the human to bring implicit knowledge to bear on the problem, that is knowledge that cannot be incorporated in the model, he must be confronted with alternatives that are perceivable as different with respect to this implicit knowledge. The implicit knowledge cannot be incorporated into the model for the reasons listed above.

Experiments have been conducted to show that optimization techniques can be used to generate such sets of alternatives. Brill (1979) has suggested a technique (HSJ) for linear and mixed integer programming problems. Brill *et al*. (1981) and Chang *et al*. (1981) have demonstrated this technique and a random generation technique on a land use allocation problem and on a water resource problem. Hopkins (1975, 1977) has demonstrated some relevant properties of heuristic location-allocation algorithms for land use allocation problems. Church and Huber (1979) have incorporated difference as one of a set of objectives for location-allocation problems.

This paper has two major purposes: 1) to present a method for identifying different, feasible alternatives in dynamic programming problems, and 2) to discuss some conceptual issues of defining differences between dynamic programming solutions. Section 2 considers possible methods and illustrates a preferred method with a simple numerical shortest path example. Section 3 discusses the concept of difference, including difference between decisions versus difference between states, interval measures of difference, difference between variances across decisions within one alternative, difference in spatial organization of decisions, and uncertainty. Section 4 describes a prototypical application to a dynamic programming model for allocating land uses for floodplain management.

2. MGA Methods for Dynamic Programming

A model for generating alternatives should have four characteristics. First, it should retain the structural relations of the system being modeled. In modeling a river system, for example, the properties of moving water should be retained in the model; in a transportation network, the connectivity of links should be retained. Simple random generation of links is, therefore, not an effective approach because it does not retain connectivity. Second, the model should take into account the value for the known objective(s). Generation of alternatives that are relatively good given what is known is, in general, preferable, even though expectations must be relaxed from the best that could be obtained if that objective alone were optimized. Third, the model should identify a small set of solutions that are maximally different from each other, subject to the feasibility and quality conditions just noted, because a human analyst can only cope with a small number of alternatives. Finally, a model for generating alternatives should be efficient not only in the number of alternatives generated per unit of computing resources, but also in the ease with which an analyst can specify inputs and interpret outputs.

For dynamic programming problems, these characteristics suggest that 1) the state and decision variable representation of the problem could be retained in order to specify the physical structure, and that 2) the problem could be solved first for a known objective in order to identify target values for this objective and to establish an initial solution from which to compute differences. Three methods which differ in the procedure for identifying different alternatives are considered: 1) adapting the k^{th} best policy method based on a difference constraint, 2) generalizing this method by treating difference and the known objective as two components to be weighted, and 3) searching a modified traceback for solutions of maximum difference. Although any final choice among these methods should await more operational experience, the second approach has apparent operational advantages over the first and third and conceptual advantages over the third.

2.1 k^{th} best policies

Bellman and Kalaba (1960) identified the problem of finding the next best solution to a dynamic programming problem as the finding of the 'k^{th} best policy.' Indeed, they suggested that one of the uses of the algorithm might be to look for solutions that, though not the best solutions to the model, are better solutions of the real physical problem of which the model is only an approximate representation. The Modeling to Generate

Alternatives (MGA) concept is very similar, but more extreme. MGA asserts that the alternatives most useful for consideration by analysts will be as different as possible from each other subject to satisfactory performance for the included objective. In contrast, the k^{th} best policy idea seeks the best solution for the included objective subject to a constraint that at least one decision variable be different from the previously generated solutions.

It is evident that the k^{th} best policy approach can be adapted to MGA purposes, but operational difficulties arise. To understand these, the algorithm must be described. The terminology of a shortest path problem will be used for clarity, but the implications hold equally for dynamic programming in general. The first step is to solve the problem in the traditional manner. The cost of the first link (decision) that is included in the solution is then set to infinity, that is, constrained out of the solution. The resulting new subproblem is solved and its minimum cost solution stored. All subproblems, in which in turn one of the links included in the original path is set to infinity, are solved and the resulting minimum costs stored. The minimum of the minimum cost solutions from all these subproblems is then the 'next best' path. The algorithm can be iterated to obtain the k^{th} best solution by successively setting to infinity the cost for each link in the previous ($k\text{-}1^{th}$) solution. Any link costs that were set to infinity to obtain the $k\text{-}1^{th}$ alternative are kept at infinity. The k^{th} best path is then the minimum of all the subproblems remaining from the previous iterations and all the subproblems from the current iteration.

If the k^{th} best policy approach were to be adapted for MGA purposes, it would probably make sense to choose a number of links to be different, say three, rather than to start with one link different and enumerate each successive next best solution. The best solution with three links different would be obtained by choosing the best of the solutions obtained by setting to infinity the cost for each possible combination of three links in the original solution. If the best such solution was deemed too costly, then one could obtain the best solution with two links different; if the solution was not too costly, then one could obtain the best solution with four links different.

The algorithm in this adapted form is relatively efficient for small or large numbers of links constrained to be different because the number of combinations of n things taken r at a time is smallest when r is small or n-r is small. The intent of MGA is to generate alternatives that are as different as possible, subject to a target on the modeled objective. The case of a small r is thus not likely to be of interest. Although the case in which n-r is small (most links different) is of interest, it is not reasonable to expect that all links can be different from all links chosen in previous alternatives if the algorithm is iterated to generate a set of, say six, alternatives. One might even argue that if unmodeled objectives are valued as much as the modeled objective and are as sensitive to the choice of links, then approximately half of the links could be expected to be different. In any case MGA is likely to require investigation of levels of difference for which the 'k^{th} best' policy method, even in its adapted form directed at a specific level of difference, is very inefficient. This expectation might be wrong for some probelms because forcing one link to be different can change the entire path. It is not reasonable, however, to choose an MGA technique hoping that forcing out one link will yield solutions with several links different.

A second operational difficulty with the k^{th} best policy approach is that there is no direct way of setting a target for the included objective, for example, total path cost. A search might be required among the best solutions for several degrees of difference before finding a different solution that is still within an acceptable range for the included objective. These operations further increase the computational load. Although it is conceptually possible to introduce a second state variable in a dynamic programming problem to specify a constraint on the included objective, this procedure would also increase the computational load. These computational problems are further exacerbated by the need not only to generate a solution different from the first, but also to generate a third solution different from both the first and the second, and so on.

2.2 Weighted difference

For Modeling to Generate Alternatives it is advantageous to generalize the k^{th} best policy approach so that the tradeoff between cost for the known objective and difference between solutions can be expressed as a ratio rather than as a constraint. Transforming a constraint into a penalty function is a standard computational technique for handling constraints in a dynamic programming problem without increasing the number of state variables. Consider, for example, a cost-minimization program. After solving the problem as a traditional dynamic programming problem, choose a ratio (i.e., weights) for the number of units of cost you are willing to forego in order to get a solution with one link different from any link in the original solution. Then weight the links that have been used and add the resulting values to the weighted link costs. Solve the new problem. The process can be iterated to obtain a set of alternatives that are different from each other.

Note that in a minimization problem, the number of links that are the same is minimized, rather than the number of links that are different maximized. (These objectives are equivalent only if the number of links in each alternative path is the same.) For MGA the weights and measure of difference are used only to generate solutions that are different; imprecision in the choice of weights and difference measure is, therefore, acceptable. Experimentation with different weights may be beneficial in providing solutions that could not be generated from only one set of weights.

The major advantage of the weighted method is that it eliminates the need for solving large numbers of subproblems at each MGA iteration to obtain solutions with desirable degrees of difference and acceptable levels of performance on the known objective. At worst, it may be necessary to try several different weights, but each weight requires the solution of only one dynamic programming problem, regardless of the number of links that are different in the solution. This computational requirement is a significant improvement over the original k^{th} best policy method even in its adapted form. Hereafter, the k^{th} best policy method will be referred to as the constrained difference method and the generalized form as the weighted difference method.

The weighted difference method is illustrated in Figures 1 and 2. Iteration 0 is the initial minimum cost solution, which is shown in Figure 1. Nodes are identified by letters. The boxes above each node indicate 1) the minimum cost by which that node can be attained and 2) the immediate predecessor node in the associated shortest path. In the example, the inclusion of a link that was in the previously generated path is assessed a value of 5, which is the cost of the most costly link in the network.

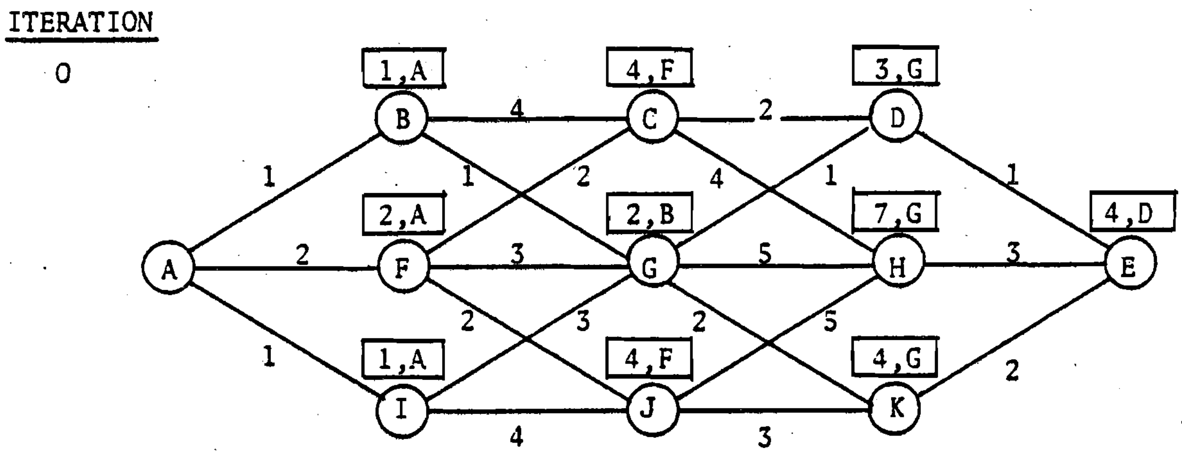
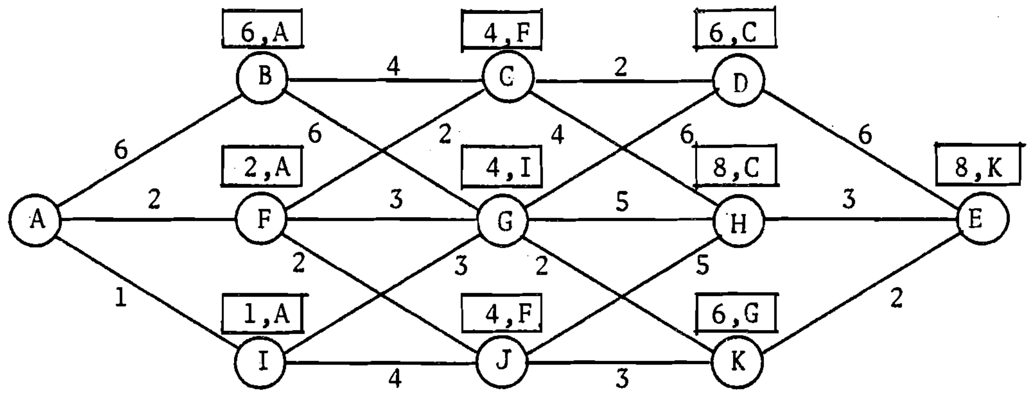


FIGURE 1 : Optimal solution with known objective

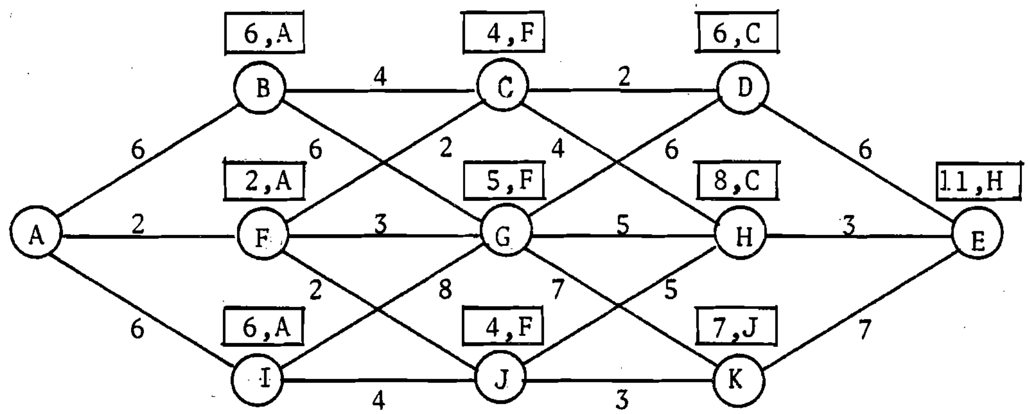
ITERATION

1



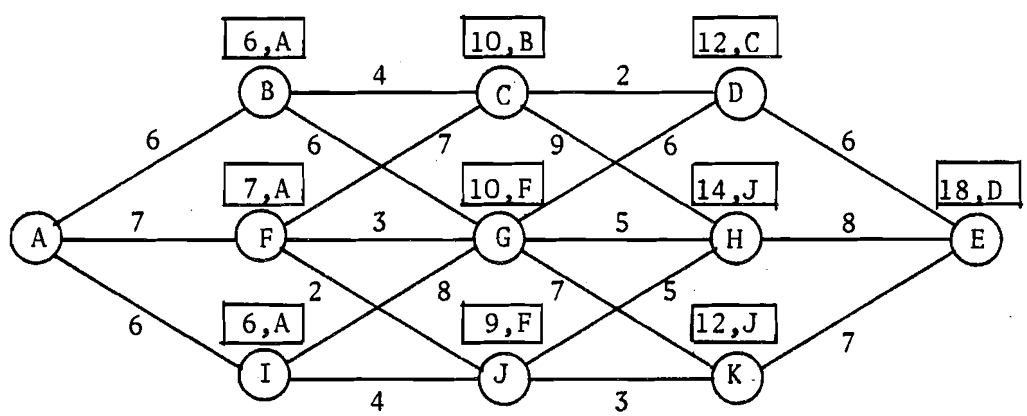
OPTIMUM PATH : AIGKE COST : 8

2



OPTIMUM PATH : AFCHE COST : 11

3



OPTIMUM PATH : ABCDE COST : 8 [+ 2(5)]

FIGURE 2 : Weighted difference method; weight = 5
(Iterations 1 - 3)

This choice of ratio implies indifference between using the most costly link and using a link that has already been used in a previously generated link.

Iteration 1 is the first MGA iteration. The cost for each link (decision) included in the solution at iteration 0 has been increased by 5 and the problem solved again. The total cost is now 8, consisting entirely of link costs because no links are reused. Iteration 2 is computed after adding 5 to all the links in the solution from iteration 1. The resulting solution now has a total cost of 11, again consisting entirely of link costs. Iteration 3 yields a solution of total cost 18, consisting of link costs of 8 and similarity (penalty) costs of 10; two links included previously are included again.

This process could continue until no different solutions can be found. An analyst might well choose to stop the iterations before this point on the premise that in terms of the modeled objective the cost was too high to yield any additional alternatives of interest. This stopping rule is valid because the sum of link costs and similarity costs is monotonically non-decreasing from iteration to iteration; that is, additional alternatives must either be more costly or more similar, or both.

In general, the weighted difference approach would be expected to accomplish the purposes of MGA at least as effectively as the constrained difference approach and more efficiently. The weighted difference approach requires no modifications to a standard dynamic programming code. A relatively straightforward outer loop in the code to do the bookkeeping of changing link values between iterations is helpful; it can be coded independently of the dynamic programming code, knowing only the form of the inputs and outputs.

2.3 Maximizing the difference in the traceback

A third possible method has been considered and should be mentioned, even though it appears to have conceptual and operational limitations for general use. Again the first step is to solve the problem in the traditional manner. The crux of this method is then to solve the problem again, this time saving pointers at each node to all previous nodes from which links yield costs within a specified range of the minimum cost for that node; in contrast, in the normal computation only the minimum is saved. The procedure thus yields a tree for the traceback that has options available at each node if more than one arriving link meets the cutoff for cost. The traceback tree can then be searched, using the dynamic programming algorithm, so as to minimize the number of links that are the same as those chosen in the original solution. For example, in the traceback, set all links that were in the previously generated solution to a value of 1 and all other links to a value of 0. Then choose a path through the traceback that minimizes the sum of these 0-1 variables. The traceback tree assures link connectivity (feasible state transitions). The saving of all pointers that meet the targets for the known objective assures that only relatively good solutions are considered.

The major conceptual limitation of this method is that it is difficult to determine appropriate target levels for the known objective because these targets must be applied to each node. Almost any rule will either allow a very wide range of total costs for the complete path or will impose an 'equity' standard across nodes (stages). For example, if the total acceptable relaxation of the known objective is divided proportionally among the nodes, then new solutions are constrained to be similar in cost at each node, rather than only in the aggregate. Although there may be specific instances in which such constraints would be appropriate, they violate the essence of dynamic programming in which cost at one node can be traded off against cost at another.

In addition to this conceptual limitation, this method requires modifications of a dynamic programming code to save more than one pointer at each node and to search the traceback. Given these conceptual and operational disadvantages, the traceback method does not appear to be worth pursuing for general use when compared to the weighted difference method.

3. The Concept of Difference between Dynamic Programming Solutions

In the preceding discussion of methods, the number of links remaining the same (or the number different) was taken to be an appropriate measure of difference between solutions. It is important, however, to consider some issues of assessing the difference between solutions for dynamic programming problems. There are technical questions, such as the distinction between the number of links different and the number of links the same if the paths consist of different numbers of links. There is also the distinction between different links in a shortest path problem and different decisions for a given stage of a dynamic programming problem. These operational questions are best answered in the context of a particular application, such as the case considered in Section 4. There are several more general questions, however, that should also be considered: differences between decisions versus differences between states, interval measures of difference, the sum of different decisions versus differences in the variances across decisions, and difference in real geographic space to which the problem may pertain.

Differences in state variables can be considered as well as differences in decision variables. In a shortest path problem differences in nodes, which represent intersections, may be as important as differences in links; a concern with congestion, for example, should apply to links and nodes. In the dynamic programming model for floodplain management described in Section 4, in which decisions are land uses and states are levels of flow in a stream, it may be appropriate to consider differences in stream flows as well as differences in land use pattern. The methods described above can be extended easily to cover nodes or states. If a node has been included in a previously generated solution, then the cost for reusing that node could be added to the cost for each link that arrives at that node. This cost assignment is equivalent to assigning the cost to the node itself. It may even be appropriate to choose weights for cost for the modeled objective, difference in links, and difference in nodes.

If the decision variables or states can be expressed in interval terms, then difference can be measured more precisely. In a model for the operation of a reservoir, for example, a decision to release 200 cubic feet per second is more different from a decision to release 400 cubic feet per second than it is from a decision to release 300 cubic feet per second. Rather than minimizing the number of decisions that are the same as a 0-1 variable (1 if same, 0 if different), the decision variable interval differences could be optimized. For example, one might wish to maximize the absolute differences between decisions in a previous alternative and decisions in the new alternative. If the modeled objective were cost minimization, then the difference objective would be to minimize the negative of the absolute differences.

For MGA it is difficult to decide what sorts of differences among alternatives are pertinent. Is a different alternative one in which more links are different, or one in which links are different from each other in different patterns within each alternative? For example, if states are levels of flow in a stream and decisions are reservoir releases, then it may be of interest to consider solutions that have different variances of stream flows over time. Such differences in solutions will not necessarily result from minimizing the number of states that are the same. One solution could have all low flows and another have all high flows; both solutions would, however, have the same variance. The likelihood that physical constraints on the problem would not permit both an all low flows and an all high flows solution suggests that the simpler measure may be sufficient for computing alternatives that are different with regard to variance for certain kinds of problems. Variance measures of difference can describe stability over time and equity objectives, both of which are frequently excluded from formal models, or at best, poorly understood.

The decisions and states may be spatially organized rather than merely temporally organized, and therefore have relationships not immediately apparent from the dynamic programming model itself. In the shortest path problem, for example, the difference between two paths can be measured as the area between the resulting spatial plot of the two paths. Such differences are especially pertinent because human analysts are likely to assess alternatives by directly comparing visual images (Hopkins, 1973). As discussed in Section 4, the two dimensional pattern of decisions may be of interest. Spatial patterns can affect other objectives beyond those considered in a model.

These questions about what types of differences among alternatives are pertinent for MGA can only be answered through experience with particular problems. The next section describes initial work with a dynamic programming model for floodplain management.

4. Floodplain Management Example

4.1 The Floodplain Management Model

The problem of allocating land efficiently relative to flooding interactions has been formulated as a dynamic programming model in a previous paper (Hopkins *et al.*, 1978). The model is explained here only to the extent necessary to illustrate the application of the dynamic programming algorithm to the generation of alternatives. In the model, the watershed is divided into subbasins, each of which is associated with a reach of the stream. Possible land uses are defined with respect to hydrologic characteristics and economic rent. A dynamic programming alogrithm can then be used to find the assignment of land uses to subbasins that maximizes economic rent to land for the entire watershed, while accounting for flood damages. Each reach of the watershed is a stage in the dynamic program. The decision variable at each stage is the land use to be located there. At each stage the following computations are made. For each combination of inflow hydrograph and land use, determine 1) the outflow hydrograph at the base of the reach, 2) the area flooded and average depth, and 3) the economic rent for the land use in the subbasin after subtracting flood damages for the depth and area flooded. For each outflow peak and time of peak the land use decision that yields the highest economic rent cumulated for all stages through the present one is saved as the best means for reaching that outflow and time of peak. (The pair of states, outflow peak and time of peak, is analogous to a node in a shortest path problem.) When all stages (reaches of the stream) have been processed in this manner, the final outflow and time of peak with the highest cumulative economic rent is the starting point for the traceback through the set of land use decisions and associated inflow levels that led to that outflow. The process is thus analogous to the solution of the shortest path problems discussed above.

The model is slightly more complex than described here in that it treats the land use in the flooded area and the land use in the rest of the watershed as separate, and therefore potentially different. An explanation of the modifications to the search algorithm to handle this and more detailed discussions of other aspects of the model can be found in Hopkins, et al., (1978, 1980, and 1981).

Figure 3 shows a map of the subbasins and the stream network for the example problem, which is based on the Hickory Creek watershed in Will County, Illinois. Table 1 lists the land use types considered for assignment to the subbasins. Note that there are four residential/urban land uses, which may be assigned with or without provision for detention of storm runoff. The two additional land uses, agriculture and recreation, can be assigned only without special provision for storm runoff. Figure 4 shows the assignment of land uses to subbasins that results from solving the dynamic programming model to maximize total economic rent.

4.2 Application of the Weighted Difference Procedure

In order to demonstrate and explore the implications of Modeling to Generate Alternatives for this problem, additional solutions were obtained

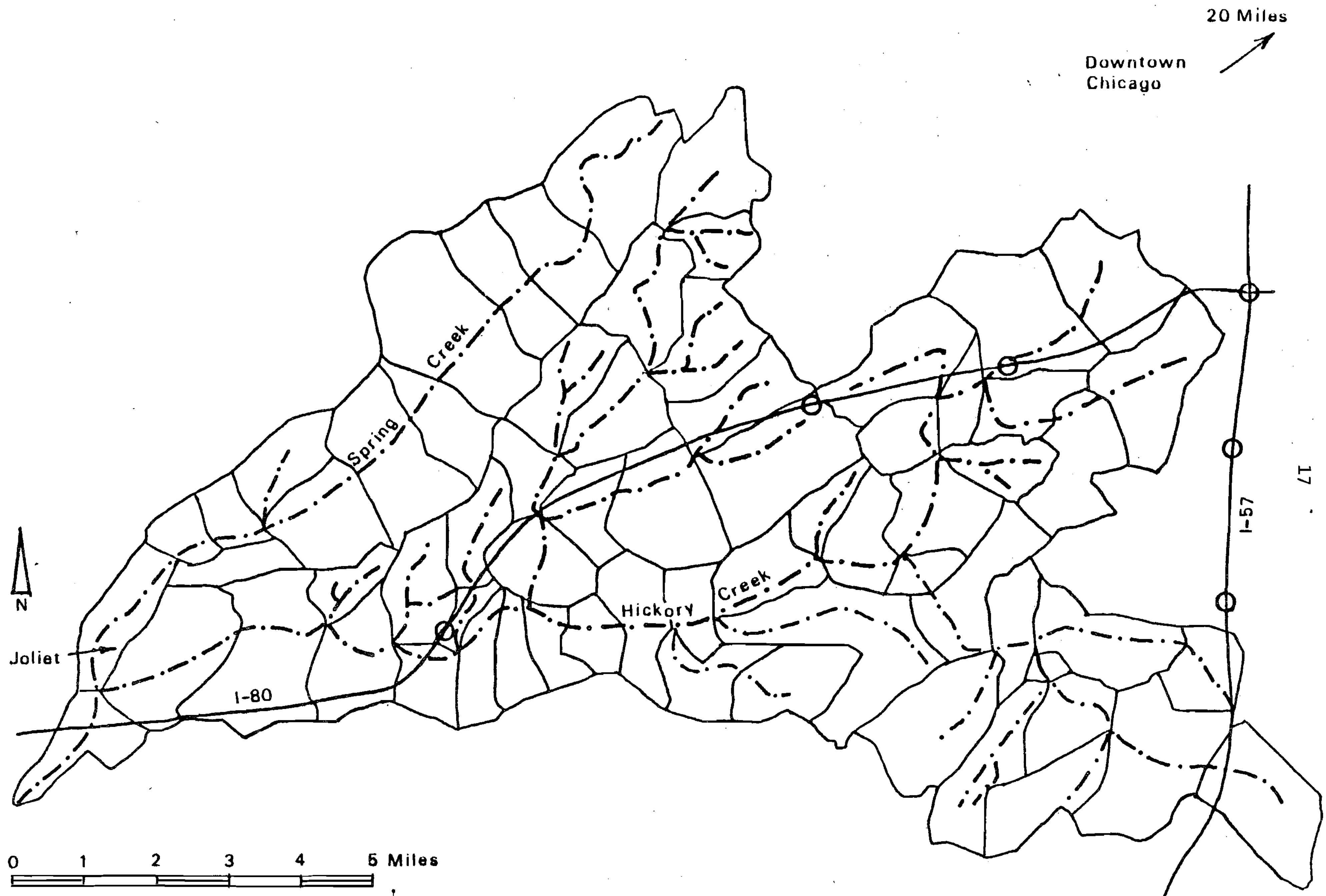


FIGURE 3 : Hickory Creek Watershed

TABLE 1

Land Use Categories

Category Number	Description
1	Urban 1: 0.2 dwelling units per acre
2	Urban 2: 1.0 dwelling units per acre
3	Urban 3: 2.75 dwelling units per acre
4	Urban 4: greater than 6 dwelling units per acre and/or industrial use
5	Urban 1 with detention
6	Urban 2 with detention
7	Urban 3 with detention
8	Urban 4 with detention
9	Agriculture
10	Recreation

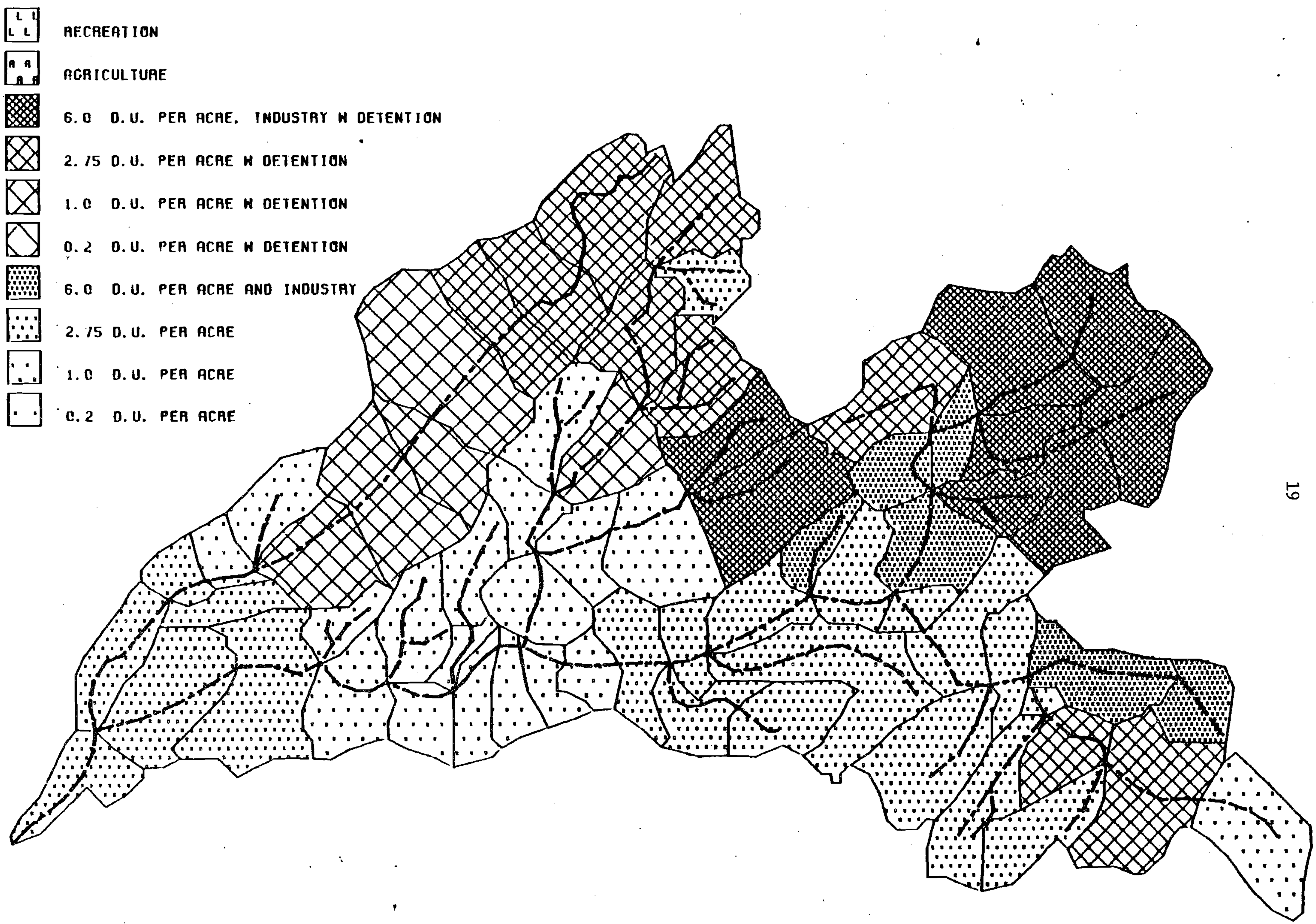


FIGURE 4 : Original Solution (Optimal Solution to Model)

using the weighted difference algorithm described above. Carrying out such an exploration requires not only choosing weights, but also choosing to which variables the weight should be applied. For this exploration, difference was focused on the decision variables (land use type) and on the number of different decisions rather than the variance of decisions within a single alternative. In contrast, the focus could have been on variance in peak flows among subbasins, for example, if adjacent landowners made extensive use of riparian water rights.

To apply the weighted difference procedure it is necessary to relate the value of having a different decision to a change in the economic rent objective function. First, some unit of measure must be established for a difference in a decision so that a weight can be applied. For each subbasin the maximum economic rent among all land uses was chosen as the reference unit, which thus varies from subbasin to subbasin. The original premise was that if this quantity were subtracted from the economic rent of the land use assigned in the original solution that, if feasible, some other land use decision would be assigned. Any land use not previously assigned would have to have an economic rent higher than that for any use previously assigned after the maximum economic rent for any use in that subbasin had been subtracted. This premise is not strictly true because economic rents can be negative (i.e. a particular land use cannot cover the costs at a particular location), and some are. It is, in any case, an easily computed indicator of the likelihood of changing a land use decision relative to changes in the economic rent objective function. This difference indicator was then weighted to yield solutions with varying degrees of difference as described below.

The economic rents for floodplain land uses are affected by flood damage levels. The actual economic rent net of flood damages for each use in each reach is computed in the model and cannot be determined before a run of the algorithm. Therefore, the difference indicator for floodplain land uses was taken as the absolute value of the minimum (most negative) economic rent among all the floodplain land uses among all the reaches in the original solution. For each reach this minimum value was subtracted from the economic rent for each floodplain land use decision included in the original solution. In the present example, this minimum value was a large negative value and, therefore, subtracting its absolute value from the economic rents for land use decisions included in the original solution tended to force most such decisions out of any new solutions generated. The choice of this particular difference indicator was somewhat _ad hoc_, but it worked well. An indicator based on the range of economic rent was also examined, as described at the end of the following section.

4.3 Results

First, weights of 1.0 were applied to both the non-floodplain and floodplain difference indicators described above, and the weighted difference procedure was run. This run resulted in a solution in which all land use decisions, except one non-floodplain land use and one floodplain land use, were different from the decisions in the original solution. The economic rent objective was 11 percent lower than in the original solution. The outflow at the base of the watershed was approximately 26,000 cubic feet per second, compared to 20,000 cubic feet per second in the original solution. The weighted difference procedure was repeated, modifying the economic rent data for decisions included in the second solution, as well as retaining the adjusted data for decisions included in the original solution. In this run, the economic rent was 41 percent less than in the original, and many of the land use decisions were the same as those in one of the previous two solutions.

In order to obtain a larger number of alternative solutions with total economic rent closer to that for the original solution, the weights on the difference indicators were decreased. The results from using weights of 0.1 on each of the difference indicators are summarized in Table 2 and displayed for the five alternatives in Figures 5 through 9. In order to present maps that are easily readable, these figures show only the non-floodplain land uses (with the exception of one subbasin in the city of Joliet as explained below). All five alternatives are within 10 percent of the economic rent for the original solution.

Outflows at the base of the watershed vary from 18,000 to 30,000 cubic feet per second. The final outflow is not valued in the objective function of the model. Preference for low final outflows is, therefore, external to the model and could be a meaningful basis for comparing alternatives if down-stream flooding is important.

Various features of the original solution and five alternatives can be discerned from Figures 4 through 9. For example, if there is detention in most subbasins on Spring Creek, then urban development can occur in Joliet as in Figures 4 and 5. Otherwise, urban development in Joliet must be removed to attain a similar (within 10 percent or less) level of economic rent (see Figures 6-9). This comparison is apparent in the maps because the entire Joliet subbasin is flooded in each alternative and the map, therefore, shows the floodplain use in the Joliet subbasin. That land use patterns with different configurations of detention and land use density can be

TABLE 2

Comparison of Alternative Solutions

Alternative	Total Economic Rent, $\$10^6$	Discharge at outlet of watershed, 10^3 cfs	% difference in total econ. rent
Original	853	19.84	---
1	833	17.97	2.4
2	809	29.96	5.2
3	803	17.55	5.9
4	798	19.91	6.5
5	787	17.93	7.7

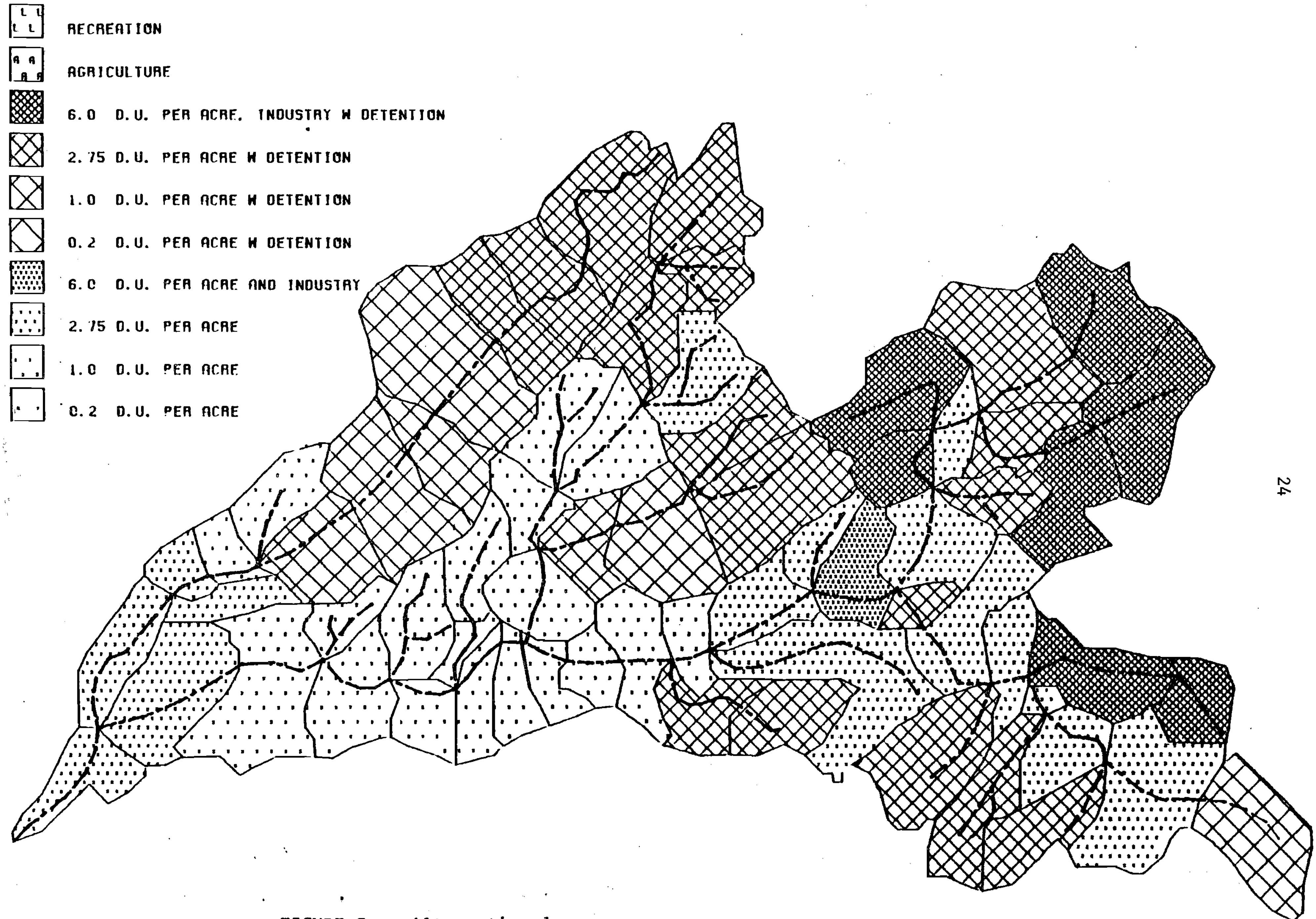


FIGURE 5 : Alternative 1

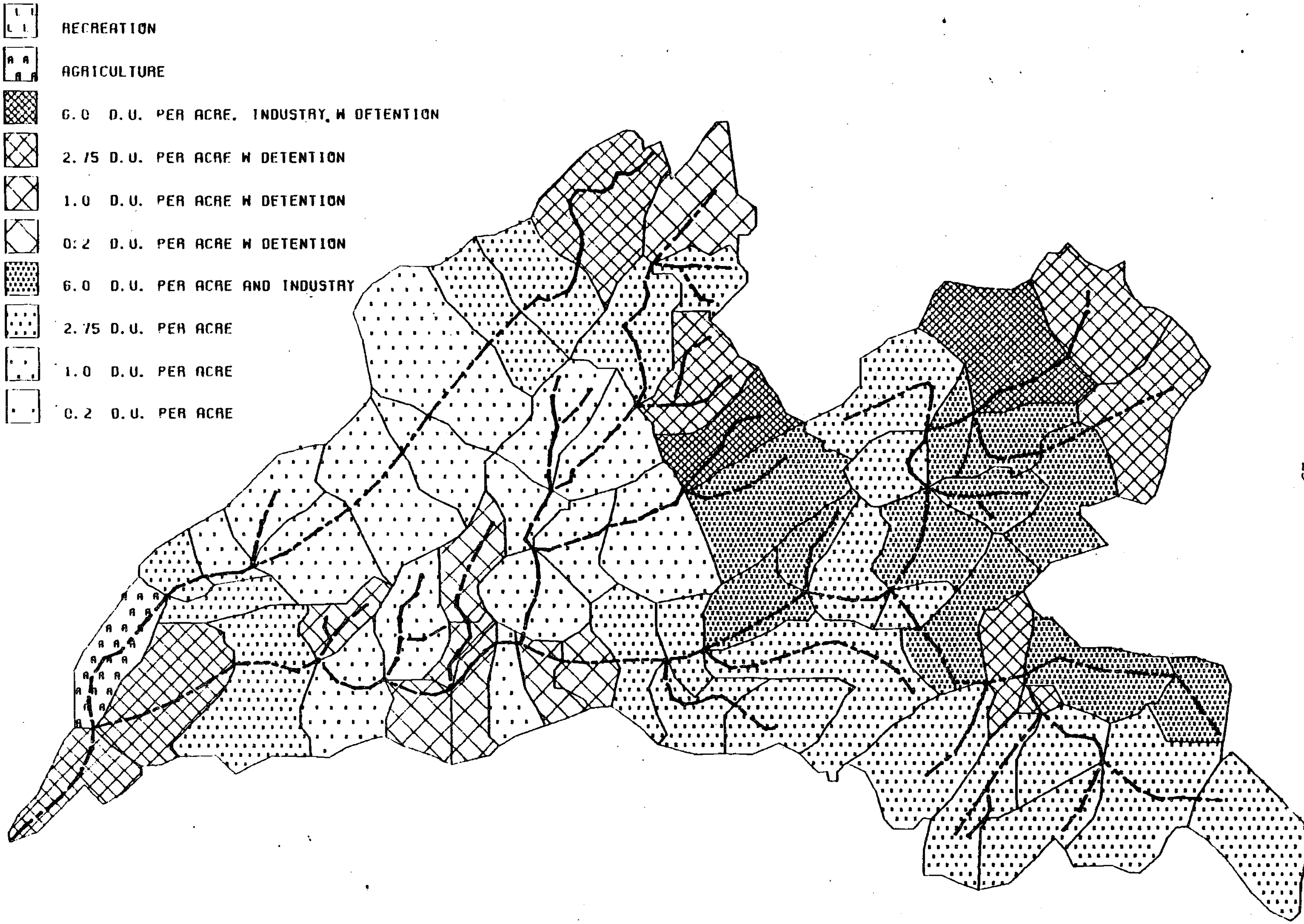


FIGURE 6 : Alternative 2

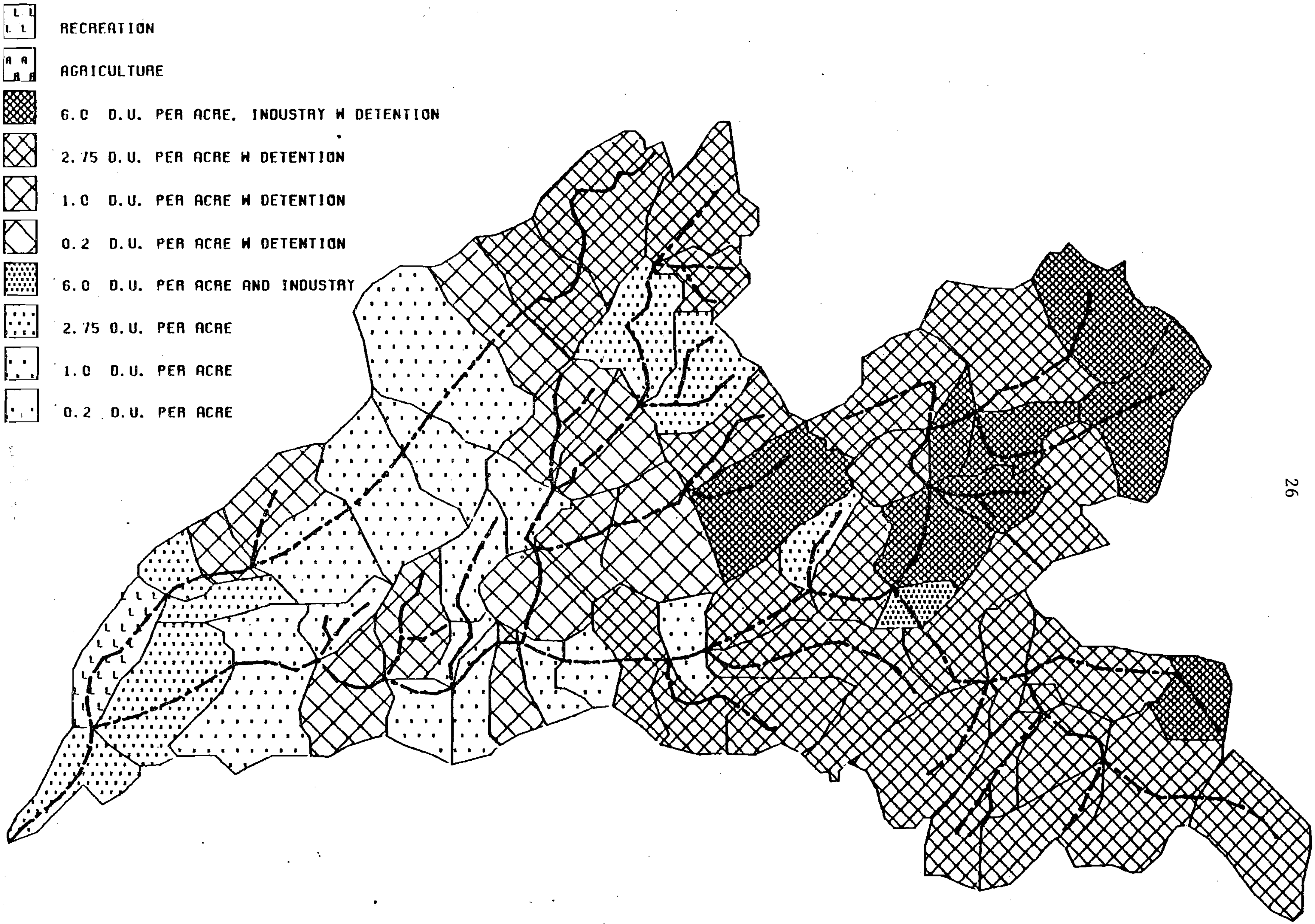


FIGURE 7 : Alternative 3

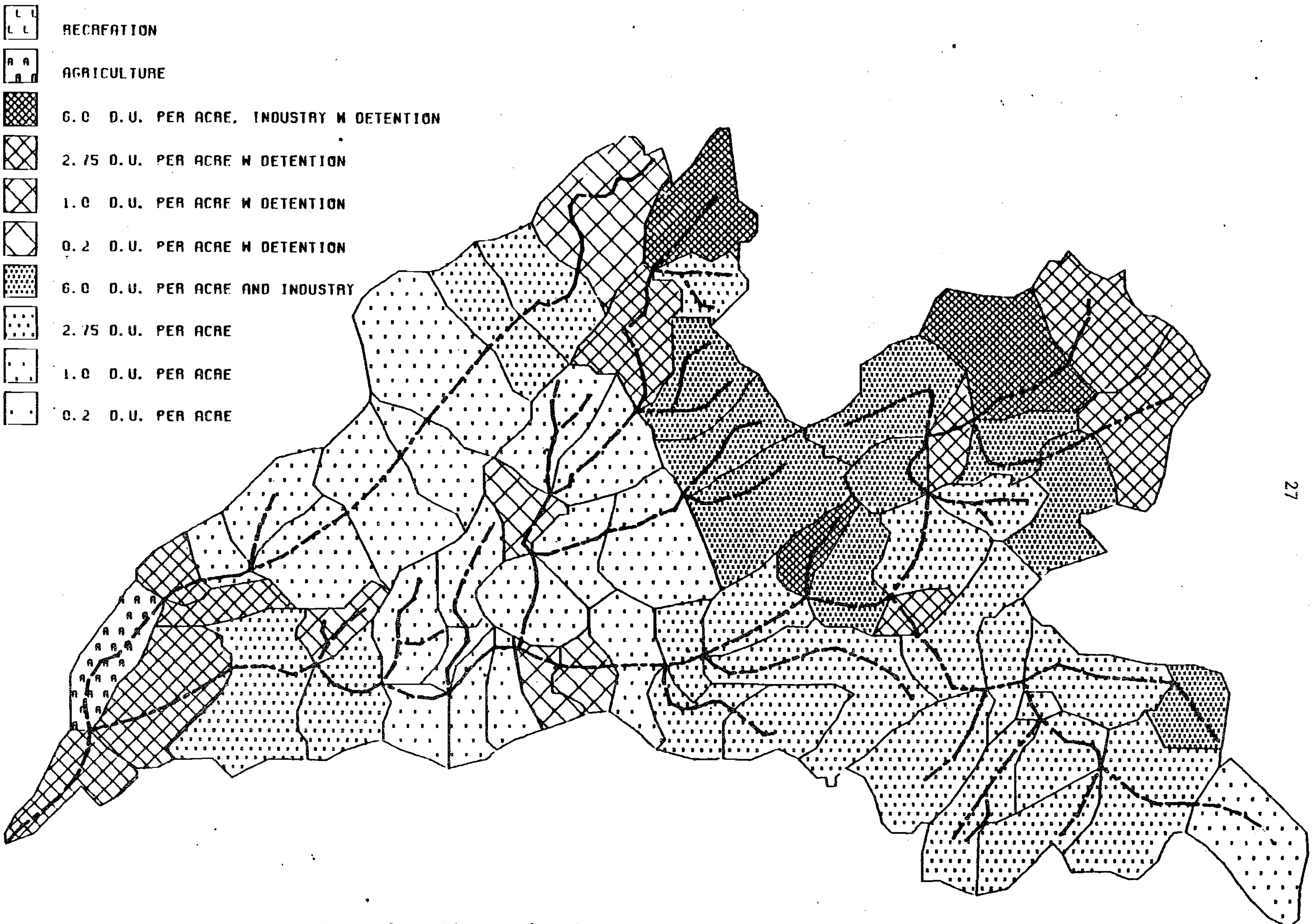


FIGURE 8 : Alternative 4

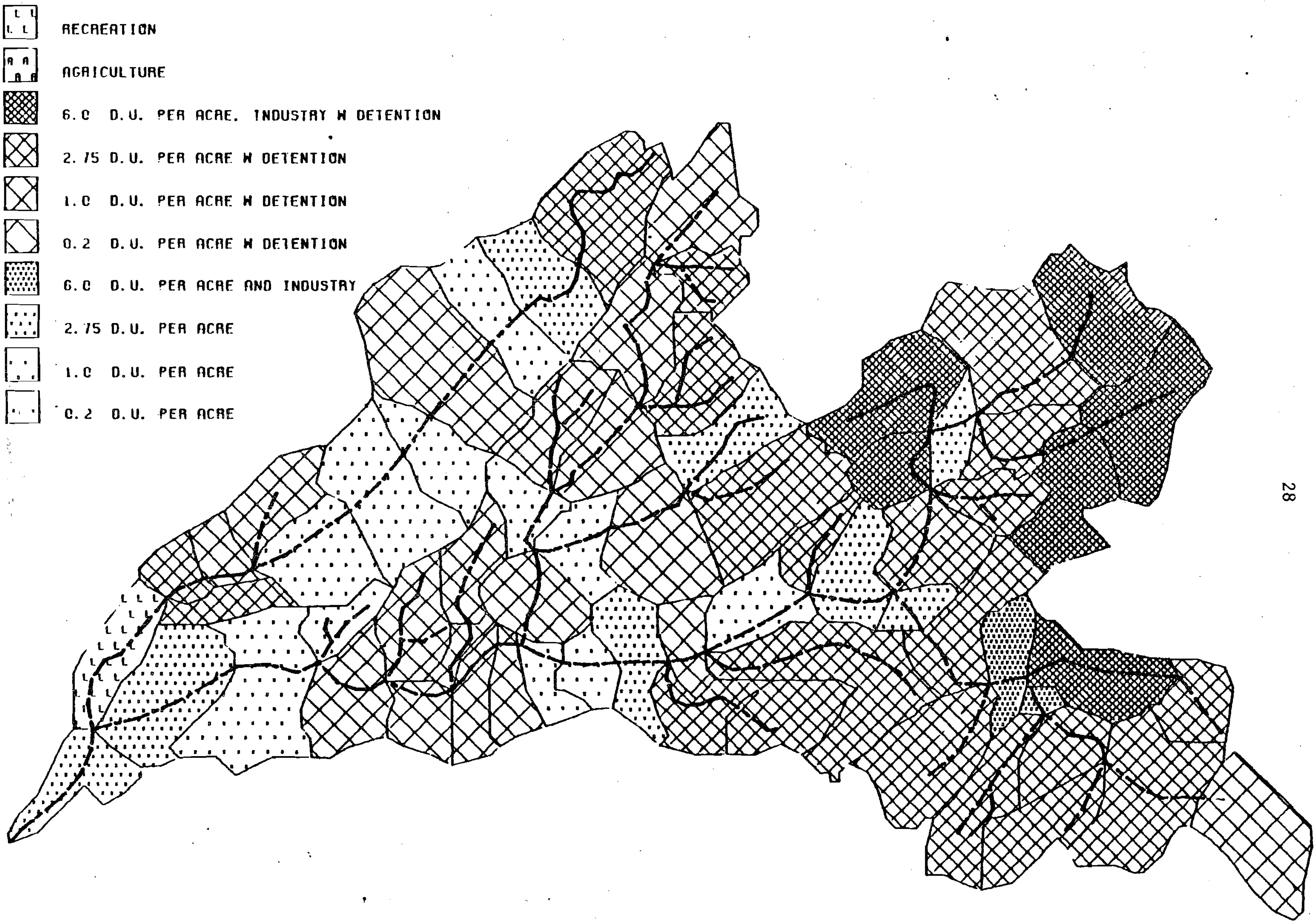


FIGURE 9 : Alternative 5

obtained is demonstrated by a comparison of the maps. Typically, land uses change from one land use to the next lower or higher density; or, the change is from the land use without (with) detention to the same land use with (without) detention. Such changes are not surprising because such land uses tend to be relatively similar in economic rent. Although no earth-shaking policy implications may jump out at the uninvolved observer of these different patterns, real decision makers have a tremendous amount of additional information, some of it political, about their particular environment. There are likely to be important differences among the patterns from the perspective of communities located in the watershed. For example, the variation in location of 6 dwelling unit per acre/industrial zones (see figures 5-9) would affect tax revenues.

As discussed in Section 4.2, in generating these alternative solutions, the difference indicator for the non-floodplain uses in each subbasin was the maximum economic rent among all land uses. Since some of the economic rents are negative in value, another possible indicator is the range between the maximum and minimum economic rents for each subbasin. A second set of five solutions was generated using this indicator within the same overall procedure. The floodplain uses were handled in the same way as in the first case. Weights of 0.1 were applied to both the floodplain and the new non-floodplain indicators. The number of new land uses in each alternative generated in each case was examined, and it was judged that the differences among the alternatives in the two sets were similar in magnitude when measured by the number of land use decisions changed. (A more detailed discussion is provided in the appendix.)

4.4 Implications of Example

The floodplain management example illustrates that for a realistic problem a variety of alternatives can be obtained, without reducing greatly the economic rent from the optimal solution to a model of the problem. The modeling to generate alternatives intent of obtaining alternatives that are perceivable as different with respect to implicit knowledge is, therefore, satisfied. This type of analysis can also be viewed as a kind of sensitivity analysis. Quite different solutions can be identified that may be equivalent in terms of economic rent if the likely error in data is considered. Finally, this analysis could be used as a mental "blockbuster" to confront the analyst with policy options that are suggested by consideration of alternatives, but are not suggested by the original solution or past experience.

5. Conclusion

The weighted difference approach for generating alternatives for dynamic programming problems has been argued deductively to be the most promising among the approaches considered. It has been demonstrated to be operational for a relatively complex and realistic dynamic programming problem. Whether the Modeling to Generate Alternatives idea in general or this approach in particular will prove useful for exploring ill-defined problems can only be determined through attempts at application with real or simulated human problem-solvers.

BIBLIOGRAPHY

Bellman, R. and R. Kalaba, "On k^{th} best policies," Journal of the Society for Industrial and Applied Mathematics, 8, 582-588, 1960.

Brill, E. D., Jr., "The use of optimization models in public sector planning," Management Science, 25:4, 1979.

Brill, E. D., Jr., S. Y. Chang, and L. D. Hopkins, "Modeling to generate alternatives: the HSJ approach and an illustration using a problem in land use planning," Management Science, forthcoming, 1981.

Chang, S. Y., E. D. Brill, Jr., and L. D. Hopkins, "Use of mathematical models to generate alternative solutions to water resources planning problems," Water Resources Research, forthcoming, 1981.

Church, R. L. and D. L. Huber, "On determining many close to optimal configurations for single and multiple objective location problems," University of Tennessee, Department of Civil Engineering, Research Series No. 34, 1979.

Hopkins, L. D., "Design method evaluation: an experiment with corridor selection," Socio-Economic Planning Sciences, 7, 423-436, 1973.

Hopkins, L. D., "Optimum-seeking models for design of suburban land use plans," unpublished Ph.D. Dissertation, University of Pennsylvania, 1975.

Hopkins, L. D., "Land-use plan design--quadratic assignment and central facility models," Environment and Planning A, 9, 625-642, 1977.

Hopkins, L. D., E. D. Brill, Jr., J. C. Liebman, and H. G. Wenzel, Jr., "Land use allocation model for flood control," Journal of the Water Resources Planning and Management Division, ASCE, WR1, 93-104, 1978.

Hopkins, L. D., I. C. Goulter, K. B. Kurtz, H. G. Wenzel, Jr., and E. D. Brill, Jr., "A model for floodplain management in urbanizing areas," Water Resources Center, Research Report 146, University of Illinois, Urbana, Illinois, 1980.

Hopkins, L. D., E. D. Brill, Jr., K. B. Kurtz, H. G. Wenzel, Jr., "Analyzing floodplain policies using an interdependent land use allocation model," Water Resources Research, 17, 467-477, 1981.

APPENDIX

Comparison of Alternatives Generated Using Two Difference Indicators

Table A-1 lists the number of new land uses in comparison to those specified in the preceding solution(s) for two sets of alternatives. The first set was obtained using the maximum economic rent for each reach as the non-floodplain difference indicator (Method I), and the second set was obtained using the range between the maximum and minimum economic rent for each reach (Method II).

Of the ten land use categories, four pairs of them specify two uses that are the same except that one provides detention of storm runoff while the other does not. In the table, the number inside the parentheses refers to the number of new land uses if adding or deleting detention is not counted as a change in land use.

For visual interpretation of the results, the number of new land uses for each alternative is plotted in Figs. A-1 to A-4. In Figs. A-3 and A-4 the number of new land uses does not include additions or deletions of detention. In Figs. A-1 and A-3 the abscissa is the number of new non-floodplain land uses, whereas in Figs. A-2 and A-4. the abscissa is the number of new floodplain land uses. These four figures show the results from these two sets of alternatives are similar. For example, in Fig. A-2 the maximum difference between the number of new floodplain land uses for the first, second, and fifth solutions for Methods I and II was only 3. The third solution from Method I yielded a slightly higher number of new land uses than that of Method II. In general, the results of the two methods are quite similar with respect to the number of new land uses associated with each successive alternative.

TABLE A-1. Number of New Land Uses Specified in Alternative Solutions

Alternative	Method I				Method II			
	Number of New Land Uses*				Number of New Land Uses*			
	Non-flood plain		Floodplain		Non-flood plain		Floodplain	
1	30	(13)+	58	(50)+	41	(10)+	60	(57)+
2	30	(7)	30	(30)	25	(7)	33	(27)
3	20	(4)	28	(17)	19	(10)	23	(14)
4	13	(4)	14	(10)	12	(2)	23	(12)
5	10	(5)	20	(3)	12	(8)	17	(9)

* Maximum possible number of new land uses is 67.

+ Adding or deleting detention is not counted as a change in land use.

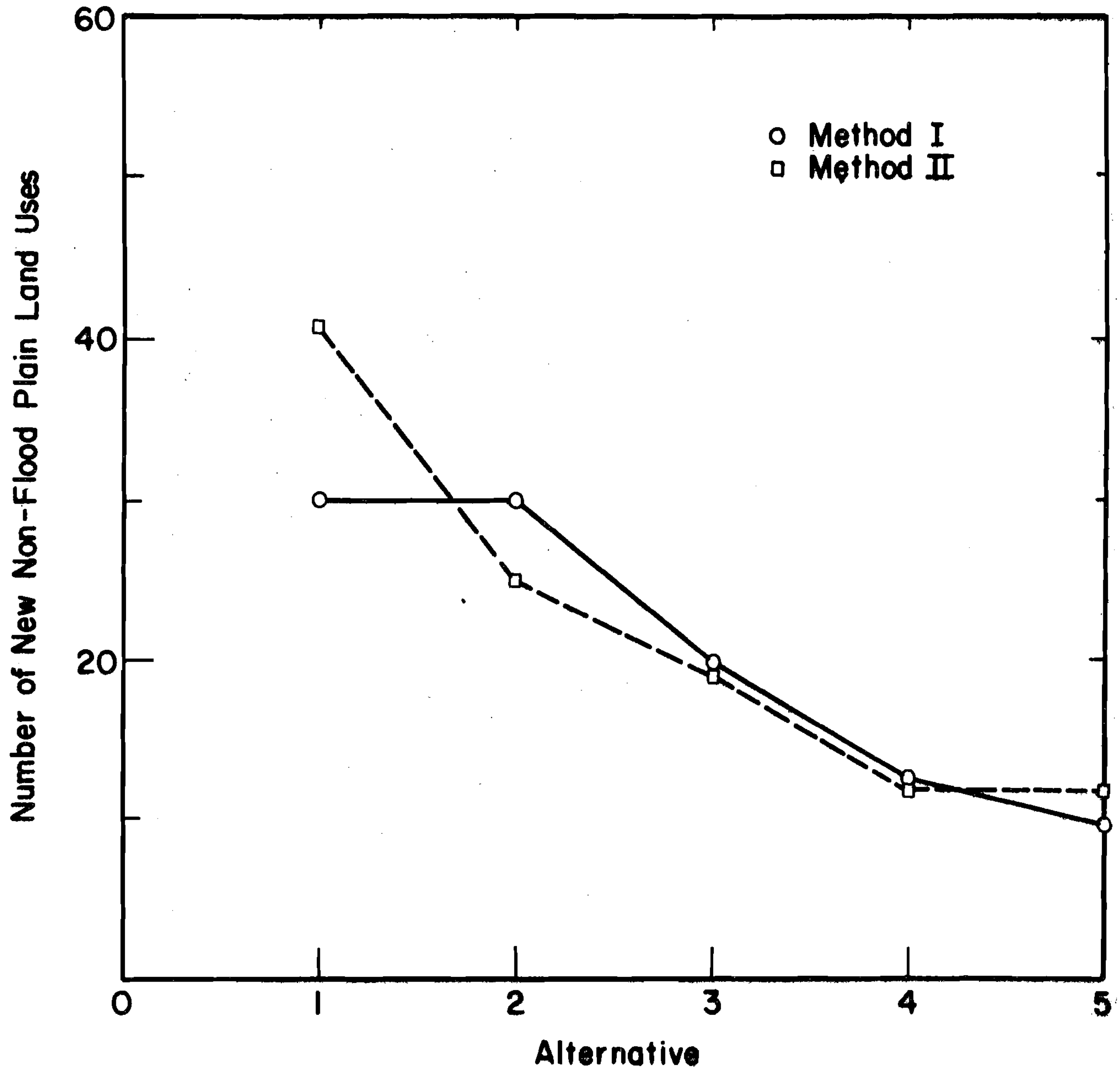


Figure A.1 Comparison of the Number of New Non-floodplain Uses for Methods I and II - Provision for Detention is Considered as a Different Land Use

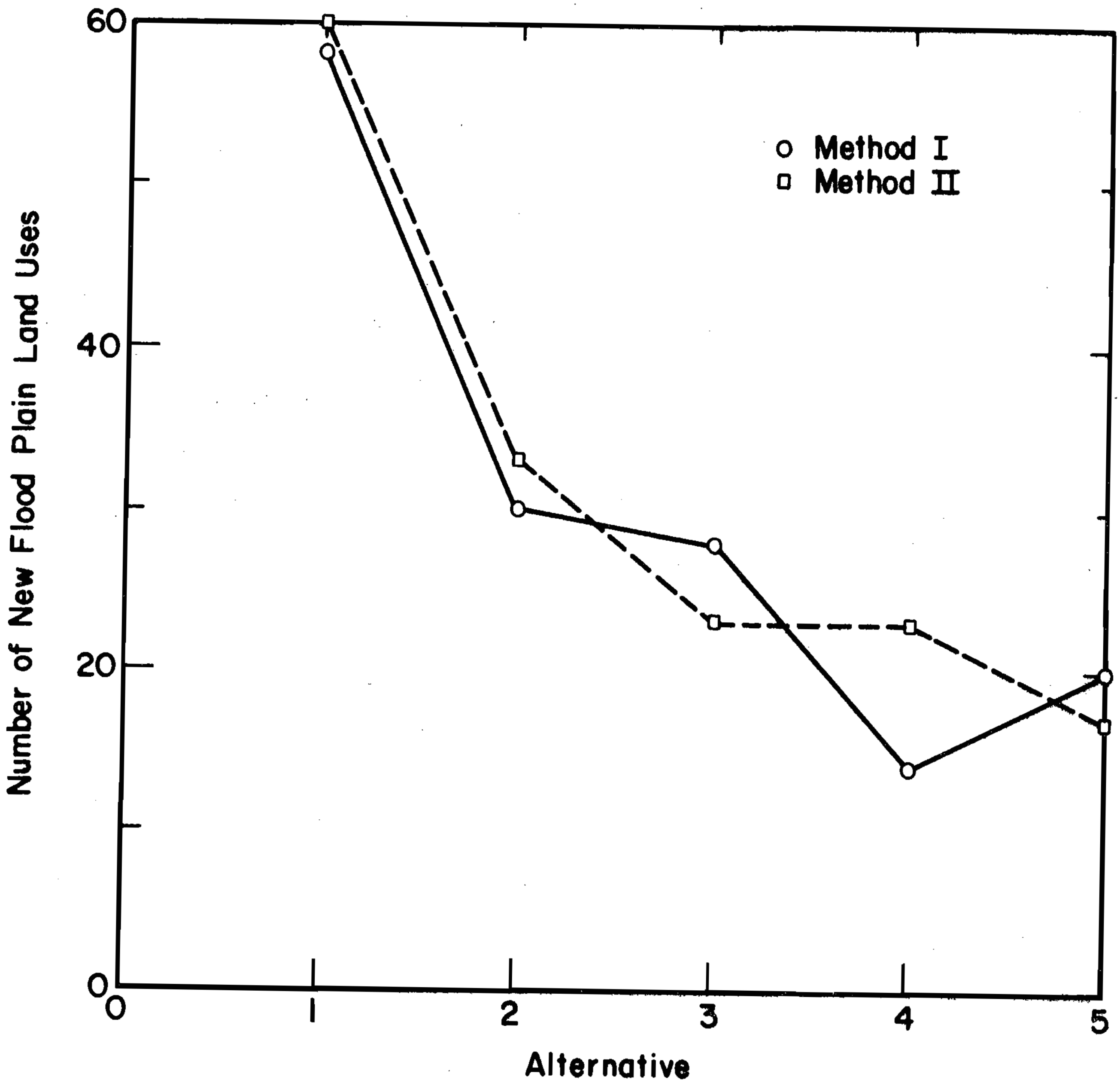


Figure A.2 Comparison of the Number of New Floodplain Land Uses for Methods I and II - Provision for Detention is Considered as a Different Land Use

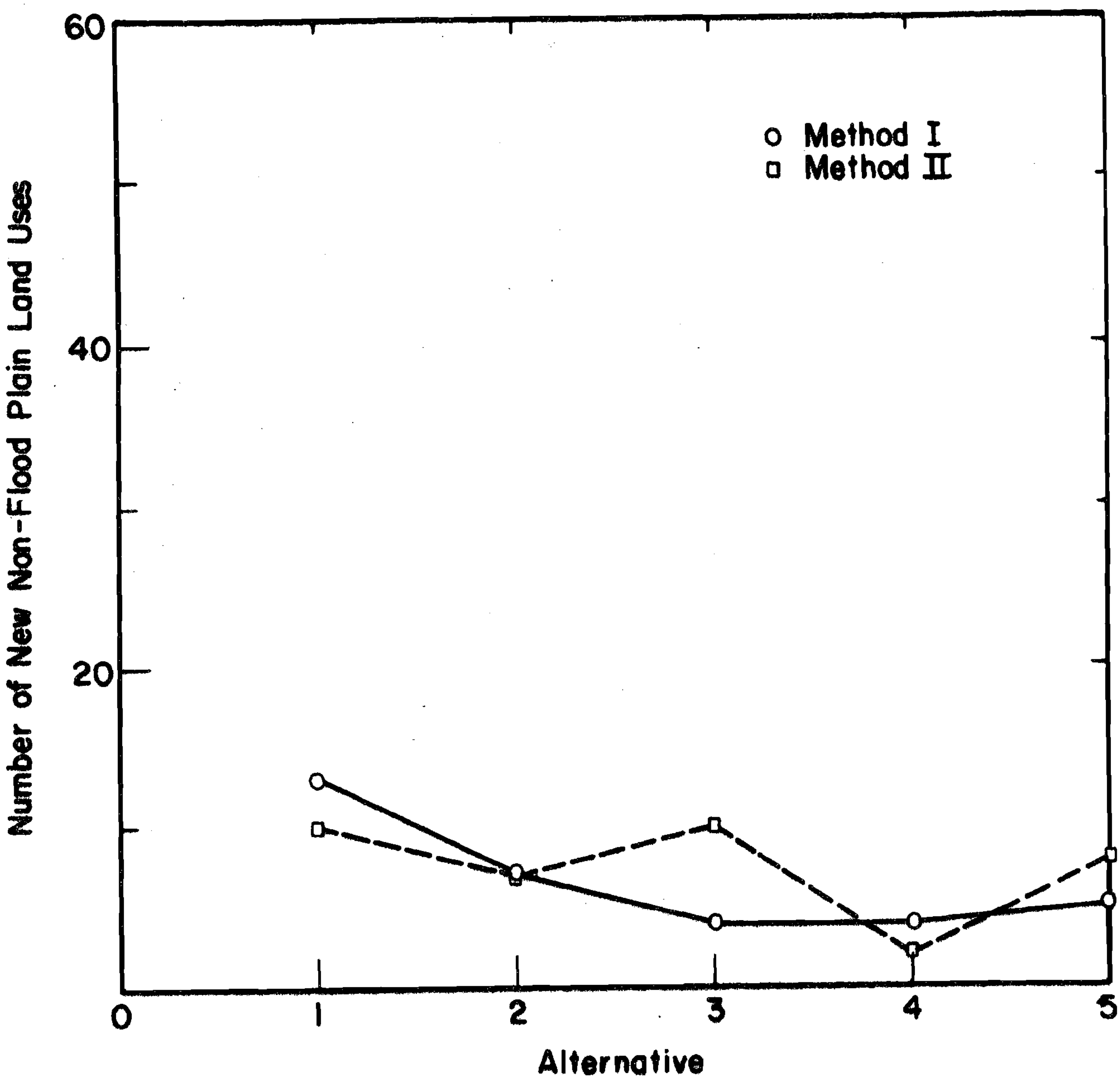


Figure A.3 Comparison of the Number of New Non-floodplain Land Uses for Methods I and II - Provision for Detention is Not Considered as a Different Land Use

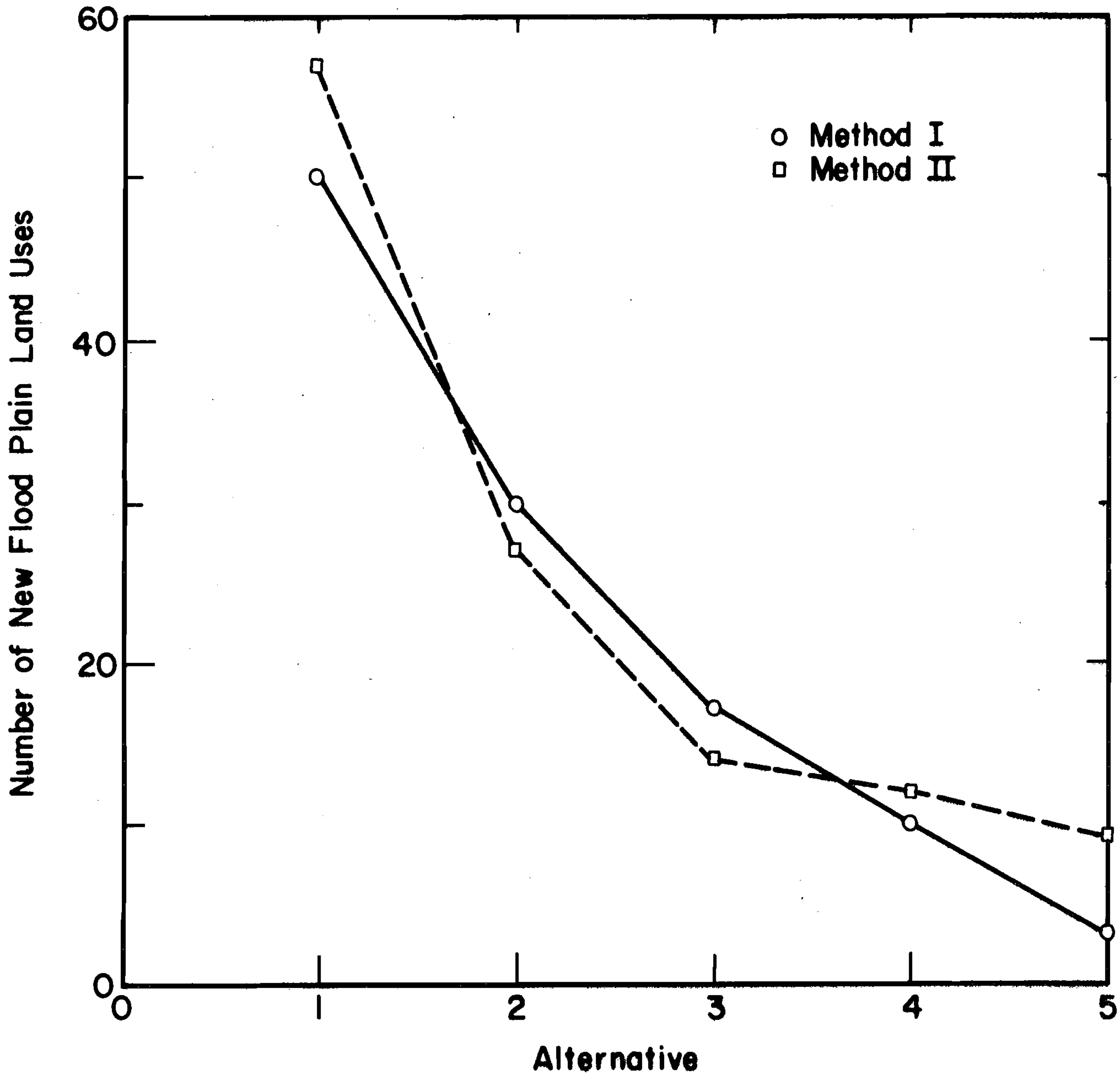


Figure A.4 Comparison of the Number of New Floodplain Land Uses for Methods I and II - Provision for Detention is Not Considered as a Different Land Use