MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0366494 2

ORNL/TM-12151

ornl

OAK RIDGE NATIONAL LABORATORY

MARTIN MARIETTA

Sustainability and Profitability in Ecological Systems with Harvesting

S. J. Gaff
V. Protopopescu

OAK RIDGE NATIONAL LABORATORY
CENTRAL RESEARCH LIBRARY
CIRCULATION SECTION
4500N ROOM 175
LIBRARY LOAN COPY
DO NOT TRANSFER TO ANOTHER PERSON
If you wish someone else to see this report, send in name with report and the library will arrange a loan.

MANAGED BY
MARTIN MARIETTA ENERGY SYSTEMS, INC.
FOR THE UNITED STATES
DEPARTMENT OF ENERGY

This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from the Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, TN 37831; prices available from (615) 576-8401, FTS 626-8401.

Available to the public from the National Technical Information Service, U.S. Department of Commerce, 5285 Port Royal Rd., Springfield, VA 22161.

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

ORNL/TM-12151

Sustainability and Profitability in Ecological Systems with Harvesting

S. J. Gaff
Taylor University
Upland, Indiana 46989

V. Protopopescu
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831

DATE PUBLISHED—AUGUST 1992

Prepared by the
Oak Ridge National Laboratory
Oak Ridge, Tennessee 37831
managed by
MARTIN MARIETTA ENERGY SYSTEMS, INC
for the
U.S. DEPARTMENT OF ENERGY
under contract DE-AC05-84OR21400

MARTIN MARIETTA ENERGY SYSTEMS LIBRARIES

3 4456 0366494 2

CONTENTS

Page

LIST OF FIGURES v

ACKNOWLEDGMENTS ix

ABSTRACT xi

1. INTRODUCTION AND STATEMENT OF THE PROBLEM 1

2. THE MODEL 3

3. ANALYSIS 10

4. DISCUSSION 43

REFERENCES 47

APPENDIX A - Optimal Control for Discrete-Time Systems A-1

APPENDIX B - Description of Program B-1

APPENDIX C - Printout of Program C-1

APPENDICES REFERENCES C-18

LIST OF FIGURES

Figure		Page
1.	Economic specific profit $\tilde{g}$ as a function of harvest size.	6
2.	Ecological specific profit $\tilde{g}$ as a function of reset value.	8
3.	Total profit as a function of harvest percentage. $a_1=0.25$, $b_1=0.24$	14
4a.	Economic and ecological profit as a function of harvest percentage. Base case	15
4b.	Total profit as a function of harvest percentage. Base case	15
5a.	Economic and ecological profit as a function of harvest percentage. $a_1=0.25$, $b_1=0.2$	16
5b.	Total profit as a function of harvest percentage. $a_1=0.25$, $b_1=0.2$	16
6a.	Economic and ecological profit as a function of harvest percentage. $a_1=0.25$, $b_1=0.25$	17
6b.	Total profit as a function of harvest percentage. $a_1=0.25$, $b_1=0.25$	17
7.	Total profit as a function of harvest percentage. Strategy #2	18
8.	Overall performance of the system as a function of the harvest percentages. Strategy #2	19
9.	Total profit as a function of harvest size. Strategy #3	20
10a.	Economic and ecological profit as a function of reset value. Strategy #4	21
10b.	Total profit as a function of reset value. Strategy #4	21

11a. Economic and ecological profit as a function of reset value. Strategy #5 22

11b. Total profit as a function of reset value. Strategy #5 22

12. Overall performance of the system as a function of the reset values. Strategy #5 23

13. Total profit as a function of harvest percentage. Strategy #6 24

14. Overall performance of the system as a function of the harvest percentage. Strategy #6 25

15. Total profit as a function of harvest percentage. a_1=0.5, b_1=0.45 26

16. Total profit as a function of harvest percentage. a_1=0.5, b_1=0.45, a_2=-0.00001, b_2=-0.00001 27

17. Total profit as a function of harvest percentage. a_1=0.5, b_1=0.45, a_2=0.00001, b_2=0.00001 28

18. Total profit as a function of harvest percentage. a_1=0.5, b_1=0.45, a_2=0.0, b_2=0.00001 29

19. Total profit as a function of harvest percentage. a_1=0.5, b_1=0.45, a_2=0.00001, b_2=-0.00001 30

20. Total profit as a function of harvest percentage. a_1=0.5, b_1=0.45, a_2=-0.00001, b_2=0.00001 31

21. Total profit as a function of harvest percentage. K=L=10,000 32

22. Total profit as a function of harvest percentage. Delay=1 33

23. Total profit as a function of harvest percentage. Delay=2 34

24. Total profit as a function of harvest percentage. Delay=4 35

25. Total profit as a function of harvest percentage. Delay=5 36

26. Overall performance of the system as a function of the harvest percentages. Base case 37

27. Overall performance of the system as a function of the harvest percentages. Delay = 1 38

28. Overall performance of the system as a function of the harvest percentages. Delay = 2 39

29. Overall performance of the system as a function of the harvest percentages. Delay = 5 40

30. Total profit as a function of harvest percentage. m=1 41

31. Total profit as a function of harvest percentage. m=1/5 42

32. Total profit as a function of harvest percentage with smaller time steps. $a_1 = 0.01$, $b_1 = 0.0095$ 44

33. Persistence of the populations as a function of reset values. Strategy #3, $a_1 = b_1 = 0.2$, $a_2 = b_2 = 0.00011$ 46

ACKNOWLEDGMENTS

This research was sponsored in part by the U.S. Department of Energy at ORNL under contract DE-AC05-84OR21400 with Martin Marietta Energy Systems, Inc. S. J. Gaff's contribution to this paper is part of her activity as a participant in the Science and Engineering Research Semester, sponsored by ORAU. The authors thank Mr. R.T. Santoro for his continuing support and interest and Mrs. S. P. Shriner for her help in preparing this manuscript.

ABSTRACT

A simple model of economic and ecological interplay for a system of two interacting populations grown in a closed environment and harvested periodically for economic purposes was analyzed. The analysis was carried out by exploring the parameter space of the model, defined by a discrete map, a harvesting strategy, and an objective functional. Results showed nonmonotonicities of the outcome and sharp sensitivities that depend on the values of the parameters and that are caused by the discrete nature of the system. This approach may prove useful for solving problems that cannot be solved analytically and for providing some guidance in the management of complex systems.

1. INTRODUCTION AND STATEMENT OF THE PROBLEM

Awareness of resource depletion and need for sustainable development has increased greatly in recent years [1]. One of the most important aspects which both decision-making organizations and the general public have come to realize is that in social systems the coupling of individual, apparently rational decisions can produce huge, often catastrophic, global effects. Over-exploitation, careless waste disposal, abusive deregulation, or one-dimensional socioeconomic policies may lead, and often have led, to severe upsets and sometimes irreversible destruction of natural or even man-made ecological entities. It has thus become evident that understanding the internal mechanisms and triggers interwoven in the structure of complex systems, like the economy or the environment, is a necessary step toward their more rational, profitable, and sustainable management.

The undeniable successes scored by physical sciences in the last several centuries within the hypothetical-deductive paradigm, the newly-added exploratory dimension offered by fast computers, together with the complexity and urgency of the present world challenges, have naturally led to various attempts at introducing more scientific methods into social sciences such as economics, sociology, operations research, psychology, and political science. Usually these sciences deal with extremely large systems and very intricate issues related to non-repetitive, one-time events. Moreover, the analysis is complicated by the scarcity of data, lack of scaling properties, imperfect knowledge about the present, and vague expectations about the future.

Mathematical modeling of very large systems will, almost by default, deprive them of their complexity and reduce them to a bare skeleton. Models — no matter how sophisticated — will never predict the exact evolution of the actual system; instead, their usefulness is to be evaluated in a different context. According to their main purpose (e.g. comparison, training, prediction, reinforcement, verification, prescription, etc.), the main functions of models in social sciences are:

(i) to guide and educate our perception and grasp of complex situations via computer experimentation, analytic results, or graphical output;

(ii) to allow extensive and rapid sensitivity analyses to eliminate undesirable alternatives;

(iii) to illustrate that complex, hierarchical, and decision-driven systems more often than not lead to counter-intuitive behaviors, emergent properties, and self-organizations that result from subtle interplays and counter-balancing effects;

(iv) to isolate the common causes and universal features of such phenomena, within the dynamics of the system itself;

(v) to provide heuristic insights and educated guesses for better control strategies and alternative approaches.

Through these functions, mathematical models can contribute to increased public awareness of the intricate causal relationships and limits on predictability in the global results of human activities. This in turn could lead to a more sensible and thoughtful approach to environmental problems and resource utilization. Coming to realize — on very simple examples — that stability is fragile, that extinction is possible, and that sustainability is not unconditional, will hopefully build a stronger consensus and lead to more careful monitoring and utilization of our resources.

The purpose of this paper is to present the results of our analysis of a simple model of economic and ecological interplay for a system of two interacting populations grown in a closed environment and harvested periodically for economic purposes. The analysis is geared towards indicating the range of things that <u>may</u> happen, rather than accurately predicting <u>when</u> they <u>will</u> happen.

In the continuous time version, the time evolution of the two populations is described by a system of coupled nonlinear ordinary differential equations:

$$\begin{aligned} \frac{dx}{dt} &= F_1(x,y) - H_1(t) \\ \frac{dy}{dt} &= F_2(x,y) - H_2(t) \end{aligned} \tag{1.1}$$

where $x(t)$, $y(t)$ represent the populations at time t and $H_1(t)$ and $H_2(t)$ are the harvesting rates (see Refs. [2-5] and references therein). The functions F_1 and F_2 represent the net growth rate of the populations in the absence of harvesting and include the effects of interactions that may range from purely competitive to mutualistic.

In general, economic profitability, dependent on the amount harvested, and ecological reward, related to the amount left to ensure sustainability and/or quality of the environment, have proven to represent conflicting interests. Thus evaluating the relative and total benefits provides an appropriate measure of the overall performance of the system.

2. THE MODEL

To define the model, we have to specify: (i) an evolution rule, (ii) a harvesting strategy including termination conditions, and (iii) an evaluation function.

(i) The time evolution of the system is described by discrete-time evolution models (maps). Using discrete, instead of continuous, dynamics is more appropriate since the entire life history of plants, animals, and crops has a naturally periodic or seasonal character. The population is assumed to be metered (i.e. measured) once during each cycle, usually a year. Thus the intrinsic evolution of the system can be written as:

$$\begin{aligned} x_{k+1} &= x_k + F_k^1 (x_k, y_k) \\ y_{k+1} &= y_k + F_k^2 (x_k, y_k) \end{aligned} \tag{2.1}$$

where

$$\begin{aligned} F_k^1 (x_k, y_k) &= a_1 x_k(1 - x_k/K)^m + a_2 x_k y_k(1 - x_k/K)^m \\ F_k^2 (x_k, y_k) &= b_1 y_k(1 - y_k/L)^m + b_2 y_k x_k (1 - y_k/L)^m. \end{aligned} \tag{2.2}$$

The values x_k, y_k represent the sizes of the two populations at time step k. The parameters a_1, b_1 represent the intrinsic growth coefficients, while the signs and sizes of a_2, b_2 determine the type and strength of interaction between the populations, namely: competitive (a_2, $b_2 < 0$), predator-prey ($a_2 b_2 < 0$), and mutualistic (a_2, $b_2 > 0$). Examples of each type abound in natural and man-made environments [2-6] and occasional switches from one type to another are not impossible [7]. All members of the population are considered to have the same growth rate, i.e. there is no age group differentiation. When the interaction coefficients (a_2, b_2) are set equal to zero, the two populations may still be coupled via external conditions. In Eqs. (2.2) the carrying capacities (K, L) represent the limits of the system to support the two populations, and the factors $(1-x/K)^m$, $(1-y/L)^m$ act as self-repression (negative feedback), ensuring a finite sized population. Indeed, if the population exceeds its carrying capacity and $m = 1$ (which is usually used in modeling self-repression terms), the growth rate becomes negative. By modifying the self repression terms to: $sign(K - x_k)\ |1 - x_k/K|^m$, $sign(L - y_k)\ |1 - y_k/L|^m$, we are able to consider other values of m, $m > 0$. The growth rates considered in Eqs. (2.2) are concave functions in x and y respectively (compensation models). Depensation or critical depensation models [5], leading to either slow growth or extinction at low densities, have not been considered.

(ii) Another important part of the model is represented by the harvesting strategies. We experimented with several different combinations of harvesting conditions and methods in order to see how harvesting policies affect the system. One condition for initiating a harvest, used in all but one of the strategies, is that whenever the population size reaches or exceeds a harvest threshold (*xthold* or *ythold*), the population is harvested. In the other strategy, the harvest occurs after a certain number of iterations (n). The harvesting method can either be selective (each population is harvested separately when its threshold condition is satisfied) or nonselective (the two populations are always harvested together when either one of the threshold conditions is satisfied. This situation arises when the populations are strongly intermingled and their separation is viewed as costly or unnecessary.) The harvest size is another possible factor. One option is to harvest a certain percentage of the population when the harvesting conditions are met. Another method is to take a constant sized harvest every time and leave the rest to continue growing. Yet another possibility is to leave a constant (reset) amount after each harvest.

The following chart summarizes the characteristics of each harvest strategy:

Harvest Strategy	Harvest technique	Size of harvest
1	nonselective	(harvest percentage × population)
2	selective	(harvest percentage × population)
3	selective	constant harvest size
4	nonselective	(population - reset)
5	selective	(population - reset)
6	after n iterations	(harvest percentage × population)

Each harvest strategy is supplemented with termination conditions. In our case they are fairly simple: whenever a certain number of harvests, N, have occurred, the harvest sequence (the evolutions with one set of parameters) is finished. This amounts to acknowledging that one can extend a reasonable prognosis for only a limited future time (e.g. N years).

(iii) To evaluate the results of a whole harvest sequence, we use an objective functional that quantifies the total economic plus ecological benefit of that sequence and compare the value of our functional at various values of the parameters. The objective functional, G, is defined as the sum of two terms, $\tilde{G}$ and $\ddot{G}$, accounting for the economic profit and the ecological reward respectively:

$$G(x,y,\lambda) = \tilde{G}(x,y,\lambda) + \tilde{\tilde{G}}(x,y,\lambda) =$$

$$\sum_{k=1}^{N} \tilde{G}_k(x_k, y_k, \lambda_k) + \sum_{k=1}^{N} \tilde{\tilde{G}}_k(x_k, y_k, \lambda_k). \tag{2.3}$$

In Eq. (2.3), the objective functional G and its two terms $\tilde{G}$ and $\tilde{\tilde{G}}$ depend on the variables, x, y, and on the set of varying parameters and controls in the problem that we denoted, in short, by λ.

The total economic profit, $\tilde{G}$, is calculated as the sum of the profits from each harvest, G_k, $k = 1,2,...,N$. The profit from one nonselective harvest is calculated as the product between the harvest sizes, H_k^1, H_k^2, and a specific profit function that is assumed to depend only on the harvest size. In a selective harvest the profit has a similar form, but for any one harvest, one of the harvest sizes, H_k^1 or H_k^2 , is zero.

$$\tilde{G}_k = \tilde{g}_1(H_k^1)\, H_k^1 + \tilde{g}_2(H_k^2)\, H_k^2$$

$$k = 1,2,...,N. \tag{2.4}$$

The specific profits, $\tilde{g}_1$, $\tilde{g}_2$, could have different forms, but we used the one represented in Fig. 1 and modeled by the analytic formula:

$$\tilde{g}_i\,(H_k^i) = \exp\,(-(A/H_k^i)^4)\,(B + (C/H_k^i)^4)\,, \qquad i = 1, 2$$

where $A = 2000$, $B = 0.1$, $C = 1778$. The specific profit is equal to the total revenue from one unit of that population minus the cost to produce it. With small harvest

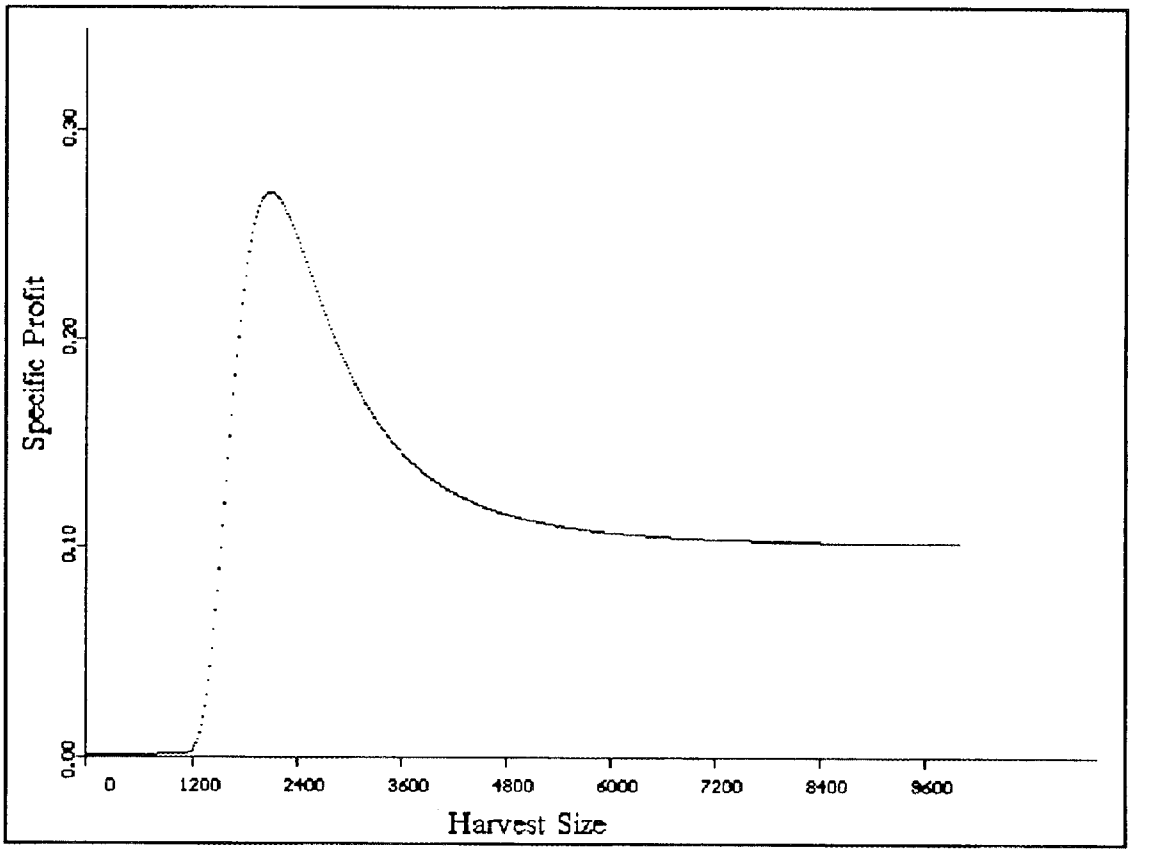

Figure 1. Economic specific profit $\tilde{g}$ as a function of harvest size.

sizes, the fixed costs are large enough that the cost per unit will make the specific profit zero. At large harvest sizes, the efficiency of the process decreases, because of additional storage and equipment costs, and again the specific profit decreases. We are assuming throughout that the market price for the populations is constant. Other factors we do not consider are the elasticity of the market, the type of market (monopoly or pure competition), and the effects of time discounting [5].

The total ecological reward, $\overset{*}{G}$, is calculated as the sum of individual rewards, depending only on the size of the population (or reset value) left after harvest k (r_k^i, $i=1,2$). The k-th contribution to ecological reward is calculated as:

$$\overset{*}{G}_k = \overset{*}{g}_1 (r_k^1)\, r_k^1 + \overset{*}{g}_2 (r_k^2)\, r_k^2$$

$$k = 1,2,\ldots, N \tag{2.5}$$

where $\overset{*}{g}_1$, $\overset{*}{g}_2$ are the specific reward functions (see Fig. 2). The specific reward is negative if too little is left, peaks for some moderate-sized remaining amount, and decreases for larger amounts because overpopulation may be detrimental. Again many actual forms would be possible for this function, but we used

$$\overset{*}{g}_i (r_k^i) = \frac{(r_k^i/1000 - 1)}{(r_k^i/1000)^2} + 0.01 , \qquad i = 1, 2.$$

Overall scaling factors multiplying $\tilde{G}$ and $\overset{*}{G}$ are used to adjust the relative importance of the two terms in G.

The total value of the functional, G, is used to appraise the system's performance in different regions of the parameter space. These comparisons allow us to evaluate the effects of various parameters and to adjust decision-making policies and management toward an optimal performance. Thus the discrete evolution system (2.1) with periodic harvesting bears strong similarities with an optimal control problem and sometimes might be solved that way [8-9].

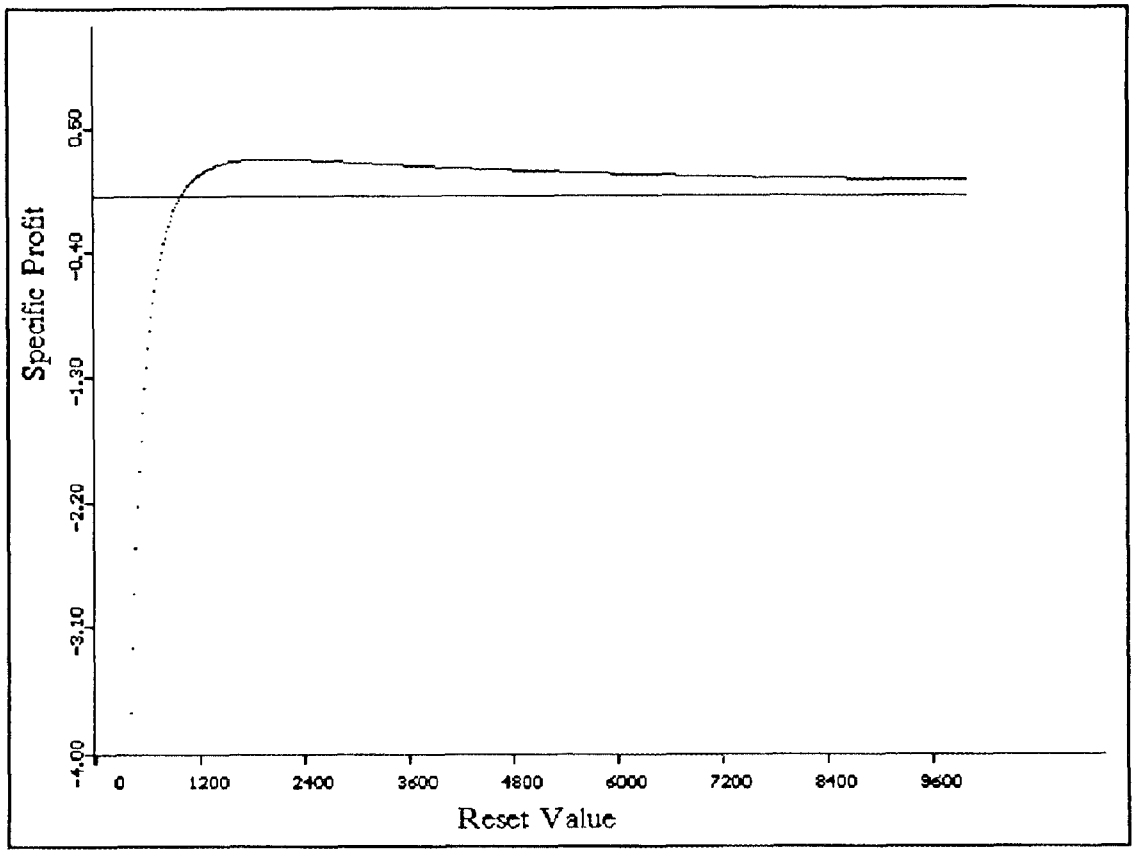

Figure 2. Ecological specific profit $\overset{*}{g}$ as a function of reset value.

Yet there are two main reasons for which we preferred a different approach, based on the more or less exhaustive exploration of the parameter space:

- in many instances the solution of the optimal control problem is: (i) not known to exist, (ii) known to fail, (iii) difficult to obtain, or (iv) inconclusive (see Appendix A for a description and limitations of the optimal control approach).

- even when the solution is known to exist and to yield usable answers, no indication is given whatsoever about the stability of that solution with respect to parametric changes. Our analysis will provide preliminary answers and indications on this aspect, that will turn out to be rather important for decision making and establishing policies.

In face of model-related uncertainties, various unsolved aspects of the mathematical analysis, and occasional inapplicability of the maximum principle, the study carried out in this paper is aimed at building a computer-based decision aid to support the analyst and the decision maker on the possible consequences of the different economic policies within a given ecological environment. Experimental mathematics and the associated computer graphics output could be of great help in assessing a situation and formulating a plan or, when still in doubt, exploring further conjectures.

3. ANALYSIS

The model described in Section 2 contains a relatively large number of parameters and options. An exhaustive analysis of all the options is at this point both irrelevant and time-consuming. In the absence of a validated model or a particular reason to explore a certain region in the parameter space, the main scope of the analysis presented here is more to illustrate and alert, than to demonstrate or predict.

Starting from a base case — to be described below — we analyzed the effect of the different harvesting strategies and parameters described in Section 2 by varying them in turn, in a somewhat hierarchical order of importance. The hierarchy was determined after a large number of preliminary computer experiments, and while neither perfect nor absolute, it serves as a reasonable taxonomic starting point. Our parameter hierarchy is organized in the following way:

1. Intrinsic growth rate;
2. Harvest strategy;
3. Type and size of interaction;
4. Carrying capacity compared to harvest threshold;
5. Delay between order for harvest and actual harvest;
6. Exponent of the self repression term.

The different parameters were varied one at a time, either as a singleton (for one population) or as a pair (for both populations). The results of these variations were plotted in one dimensional (1-D) and two dimensional (2-D) graphs, respectively. In the 1-D graphs we were interested in the variation of the economic, ecological, and total benefit over the whole harvest sequence as a function of the respective varying parameter. In the 2-D graphs we represented the regions of superior, acceptable, and unacceptable overall performance as a function of two similar varying parameters (e.g. the harvest size of the two populations).

The base case models a situation where the two populations evolve independently and the harvesting strategy is nonselective. The amount harvested is a percentage of the population at the time of the harvest (harvest strategy #1). Other parameters were set to the following values: $a_1 = 1.0$, $b_1 = 0.95$, $a_2 = b_2 = 0.0$, $K = L = 20{,}000$, $xthold = ythold = 10{,}000$, $m = 1/2$, $N = 40$, $\tilde{G}$ factor = 1, $\dot{G}$ factor = 0.12.

1. One factor that has a large effect on the size of the oscillations is the value of the intrinsic growth coefficients, a_1 and b_1. As these coefficients increase, so does the size of the oscillations (Fig. 3 & 4b). The difference between the intrinsic growth coefficients of the two populations also affects the behavior of the total profit in

nonselective situations. If the two populations grow at significantly different rates, the ecological factor, and therefore the total profit, will become negative quickly for larger harvests (Fig. 5a, 5b, 6a & 6b).

2. The next factor we analyzed was the harvest strategy. This factor affects the shape of the total functional curve significantly. Taking the growth coefficients close to one, we looked at six different strategies. The first was strategy #1, the base case (Fig. 4b). The results for strategy #2 are similar, but differences in the growth coefficients do not have as much effect in the latter case (Fig. 7). The two dimensional graph for this strategy, with the harvest percentages varying separately, shows somewhat fractal boundaries (Fig. 8). For strategy #3 (Fig. 9), the results are surprisingly different than for the previous ones. This difference is due to the discrete nature of the system, which causes the population size to differ from harvest to harvest. Growth occurs in discrete lumps, so in most strategies the harvest will occur at a point past the threshold. Within strategy #3, the harvest is exactly the same size for each harvest in a sequence while the amount remaining varies. When harvesting a percentage, both the amount that is harvested and what remains changes for each harvest. The results for strategy #4 and #5 are given in Fig. 10b and Fig. 11b, respectively. The shapes of these graphs are again significantly different from the previous graphs. They show a total profit with fewer but larger oscillations. Because the populations are always harvested to a certain level, the ecological reward is monotonic (Fig. 10a, 11a). Only the economic profit causes oscillations in the total profit. In the graph of strategy #5 with the reset values varying separately, boundaries appear that represent limits in the system (Fig. 12). The last strategy, #6, shows sections of connected behavior and then jumps (Fig. 13). These jumps result from the fact that after a small number of steps the population is oscillating around the carrying capacity. The timing of the harvest can greatly influence the value of the economic conditions. These oscillations are especially evident in the two dimensional graph (Fig. 14).

3. In order to test the effect of the interactions, we took a moderate value for the intrinsic growth and used our original parameters and harvest strategy (Fig. 15). Using a high value for the intrinsic growth may lead to unrealistic time steps when a mutualistic interaction is added, so the highest values were not used. For purely competitive situations (a_2 , $b_2 < 0$), the total profit becomes negative more quickly for large harvesting percentages (Fig. 16). In the mutualistic setting (a_2 , $b_2 > 0$), the value of the total functional increases for most harvest percentages and becomes negative only at very large percentages (Fig. 17). When $a_2 > 0$, $b_2 = 0$ or $a_2 = 0$, $b_2 > 0$, it matters which side is benefiting. If the slower-growing population benefits ($b_2 > 0$), then the total profit is higher at high values of the harvest percentage, and the total becomes negative much later (Fig. 18). With predator-prey interactions, the predator has a positive interaction and the prey has a negative interaction. The results depend on which population is given the positive interaction coefficient. When the slower-growing population is the prey ($a_2 > 0$, $b_2 < 0$), the total profit

becomes negative much sooner (Fig. 19). When the slower population is the predator ($a_2 < 0$, $b_2 > 0$), the total profit stays positive longer than with no interaction (Fig. 20). The analysis of these cases show that the interaction acts like it is increasing or decreasing the growth rate and so can increase or decrease the oscillations. These examples also demonstrate that if the difference between the two populations' growth rates is increased, it causes the total profit to become negative at lower harvest percentages. The size of the interaction causes the results to be more or less pronounced.

4. Another factor that alters the total profit is the relative size of the carrying capacity compared to the harvest threshold. If they are equal, a pattern of rounded oscillations will result (Fig. 21). If the carrying capacity is somewhat larger than the threshold, the oscillations become steeper and the peak at lower percentages becomes more noticeable. The combination with the largest oscillations (Fig. 4b) occurs when the carrying capacity is approximately twice the threshold which corresponds to the maximum sustainable yield [5]. The size of the oscillations in Fig. 4b is directly related to the populations' growth rates at the time of the harvest.

5. Another parameter that plays an important role is the delay between the moment the harvest conditions are met and the actual harvest time. This delay, accounting for inherent communication and preparation time in real situations, changes the original results significantly. With just a one iteration delay, the whole pattern is shifted to the right (Fig. 22). With a delay of two iterations, the shape of the oscillations is already altered (Fig. 23). These changes are, at least in part, a result of the population size oscillating around the carrying capacity. When the population grows two steps past the harvest threshold, it often has begun oscillating around the carrying capacity, while after only one step past the threshold, it is not yet oscillating. For longer delays, the pattern has more short continuous steps with large jumps between (Fig. 24). These continuous parts become longer as the length of the delay increases (Fig. 25). Another trend with these delays is that the first jump will be down if the delay is for an even length of time and up if the delay is odd. After a delay of two steps, the populations are oscillating around the carrying capacity and where they are when the harvest occurs determines the behavior. A series of two dimensional graphs show more about the effect of a delay. Fig. 26 shows the base case with no delay. The addition of a one and two time step delay appear in Fig. 27 and 28, respectively. The change from an orderly pattern to something more random is obvious. To show a time delay of five it is necessary to change the boundaries that determine the acceptable range of values. This delay causes large random patches (Fig. 29).

6. By varying the exponent in the self repression term, it is possible to look at how different maps change the total profit. It turns out that if the exponent is close to one, the oscillations are large at larger percentages of harvest (Fig. 30). With smaller exponents, the initial peak is increased and the following oscillations are reduced (Fig. 31). Our original value of 1/2 provides a judicious balance between the two trends.

Many other factors could be varied such as: (i) the number of harvests considered in one sequence (N), (ii) initial values of the populations (x_0, y_0), and (iii) the functional form of the nonlinear maps. Compared to the ones we analyzed before, these factors either produce small changes or would be difficult to control in a real system.

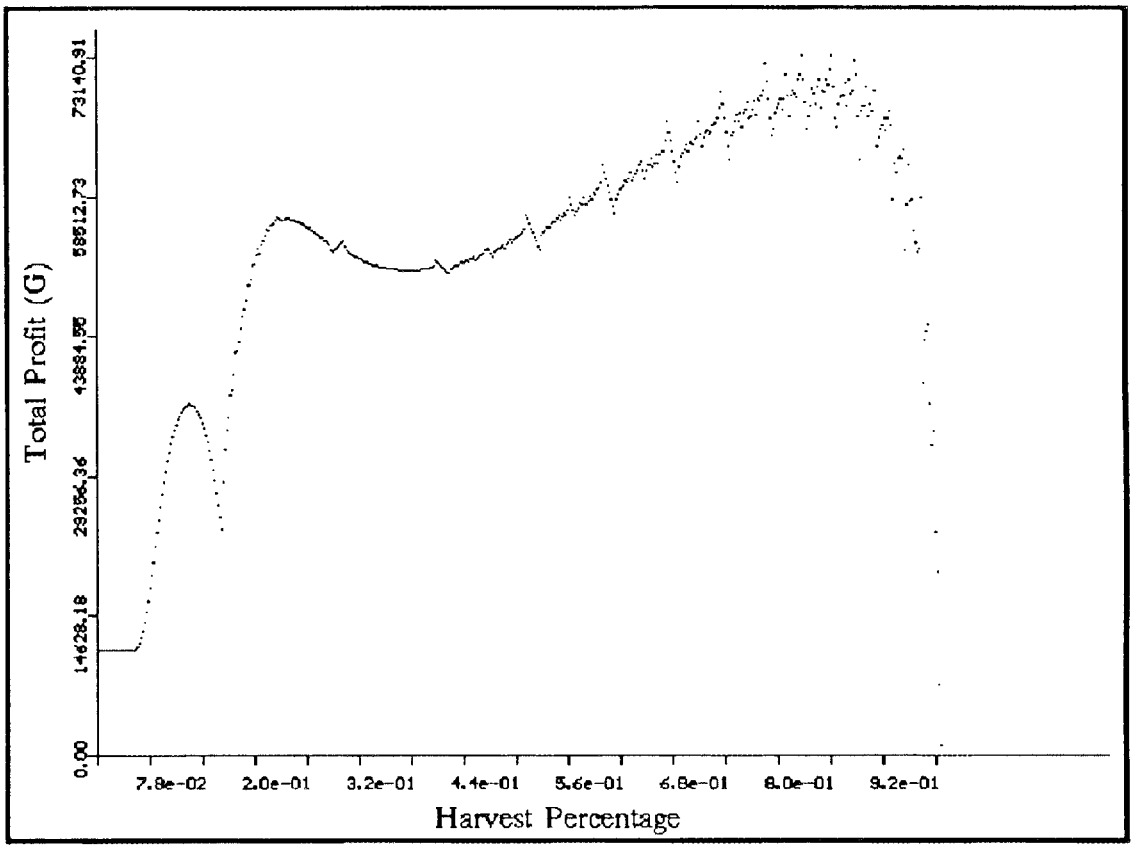

Figure 3. Total profit as a function of harvest percentage. $a_1=0.25$, $b_1=0.24$

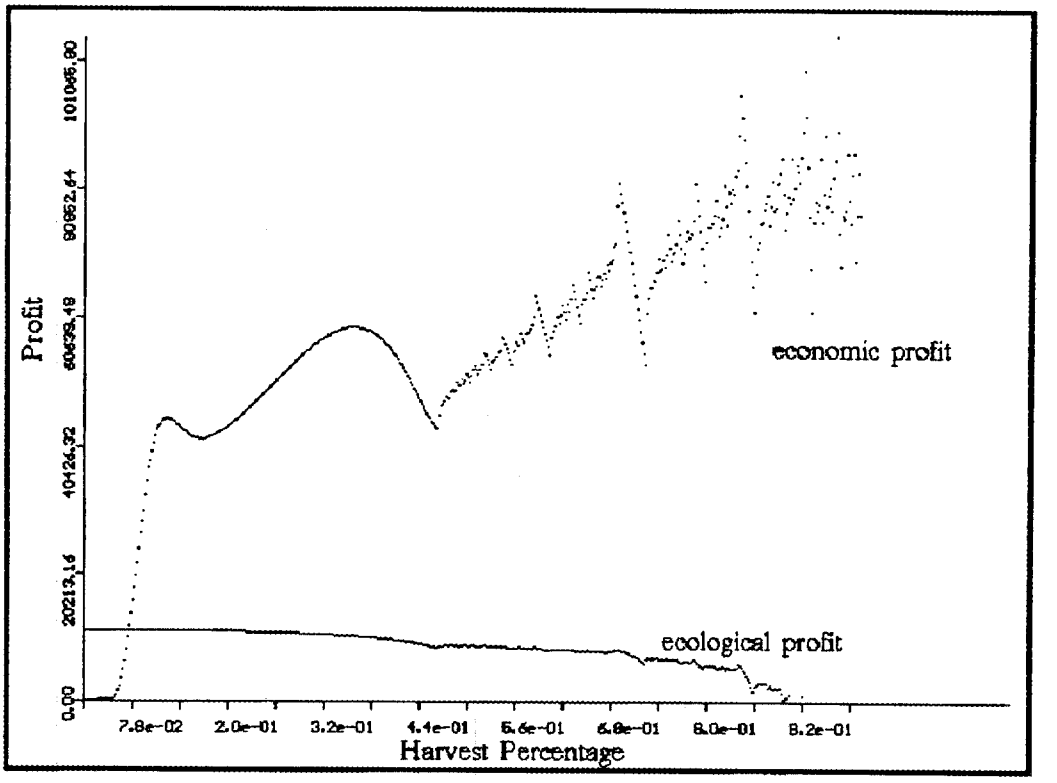

Figure 4a. Economic and ecological profit as a function of harvest percentage. Base case

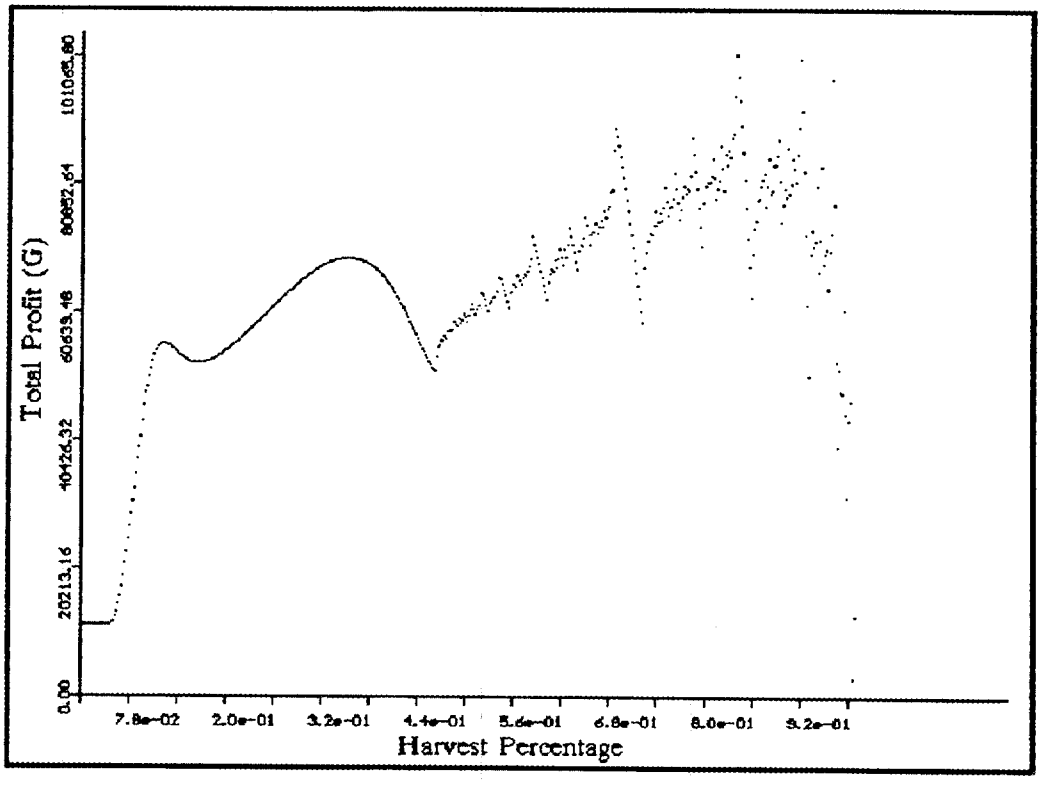

Figure 4b. Total profit as a function of harvest percentage. Base case

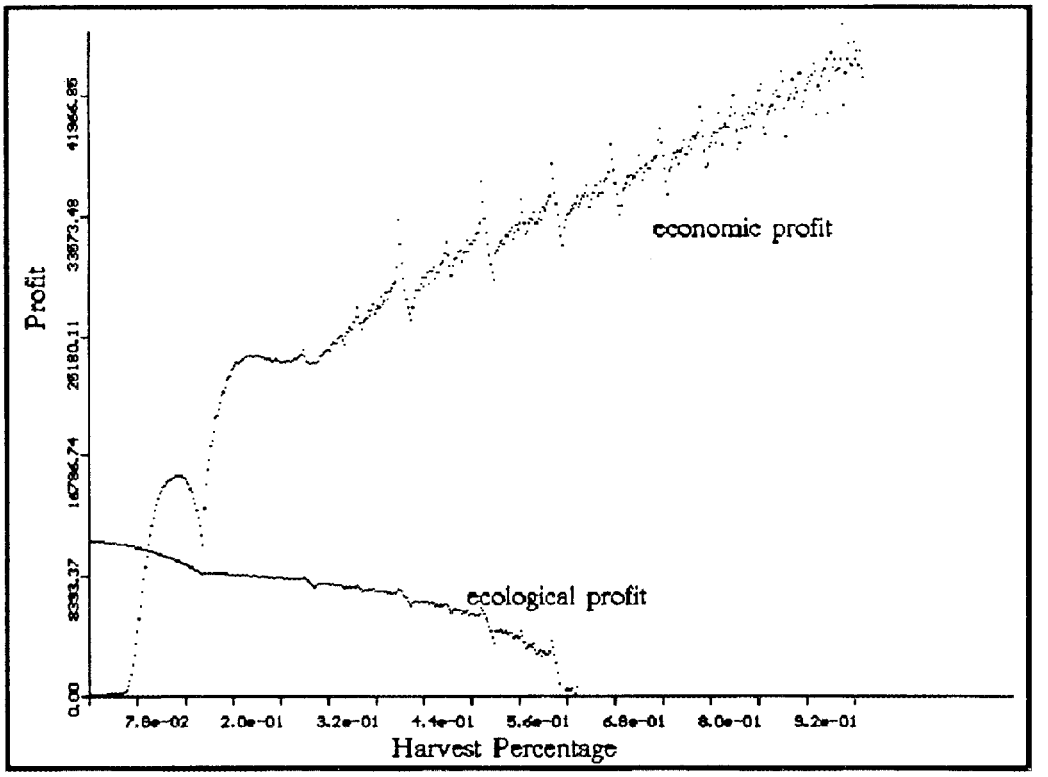

Figure 5a. Economic and ecological profit as a function of harvest percentage. $a_1=0.25$, $b_1=0.2$

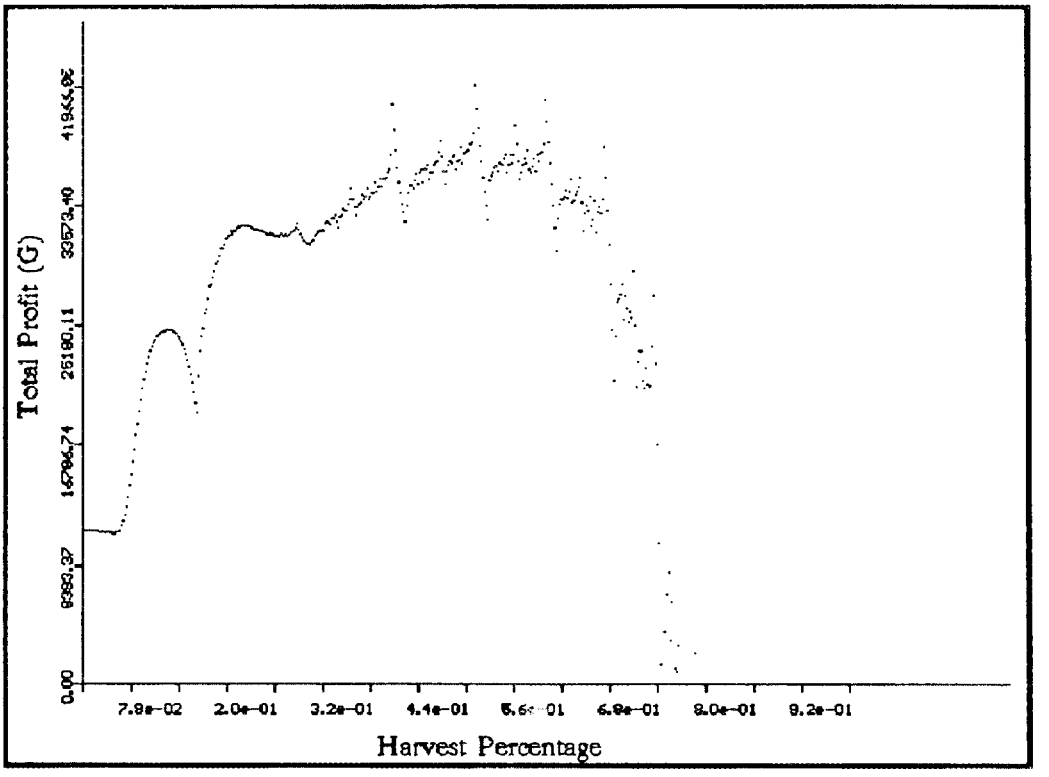

Figure 5b. Total profit as a function of harvest percentage. $a_1=0.25$, $b_1=0.2$

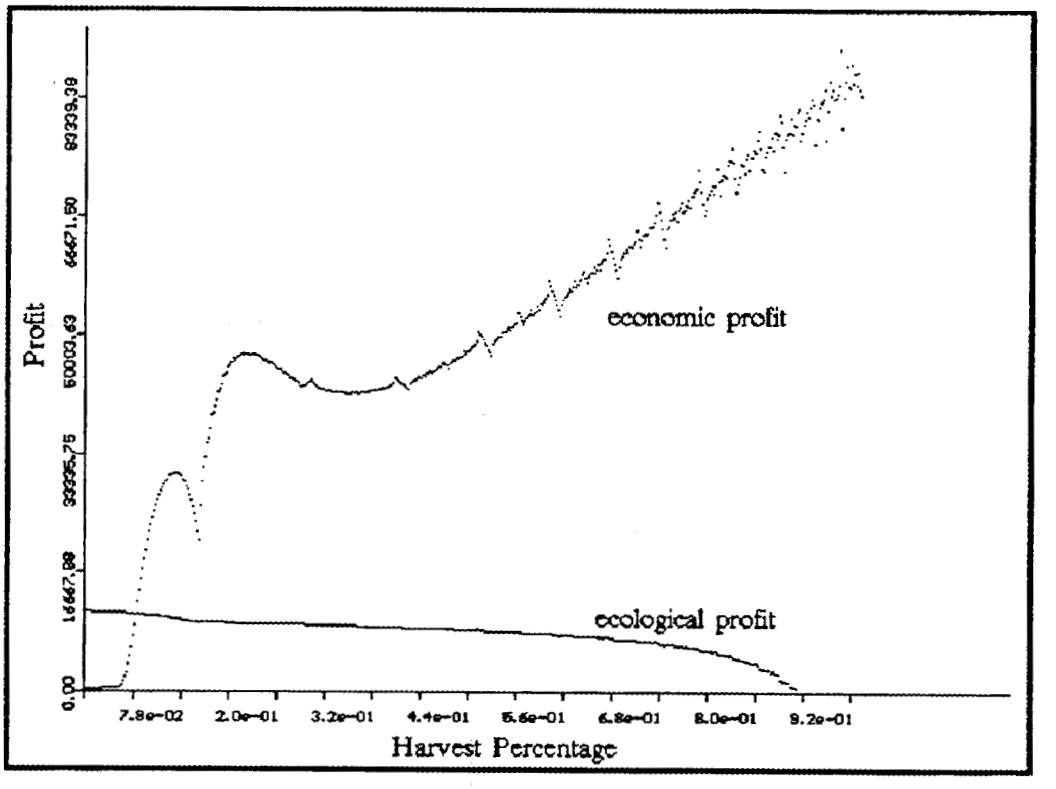

Figure 6a. Economic and ecological profit as a function of harvest percentage. $a_1=0.25$, $b_1=0.25$

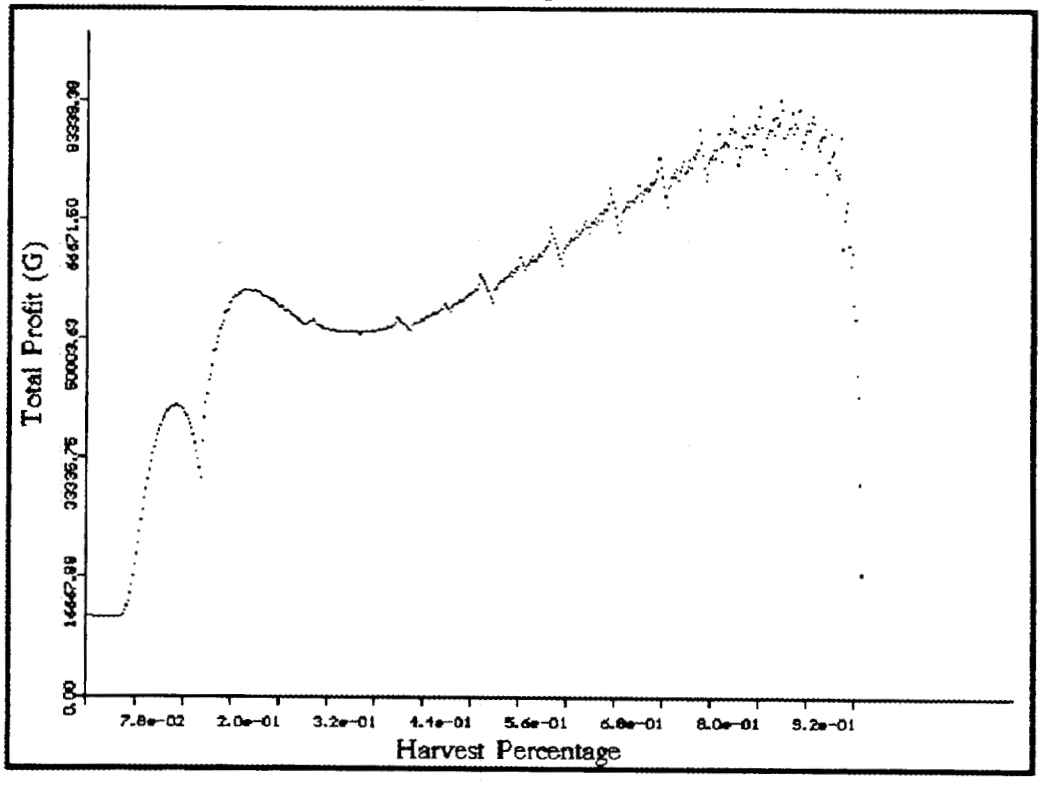

Figure 6b. Total profit as a function of harvest percentage. $a_1=0.25$, $b_1=0.25$

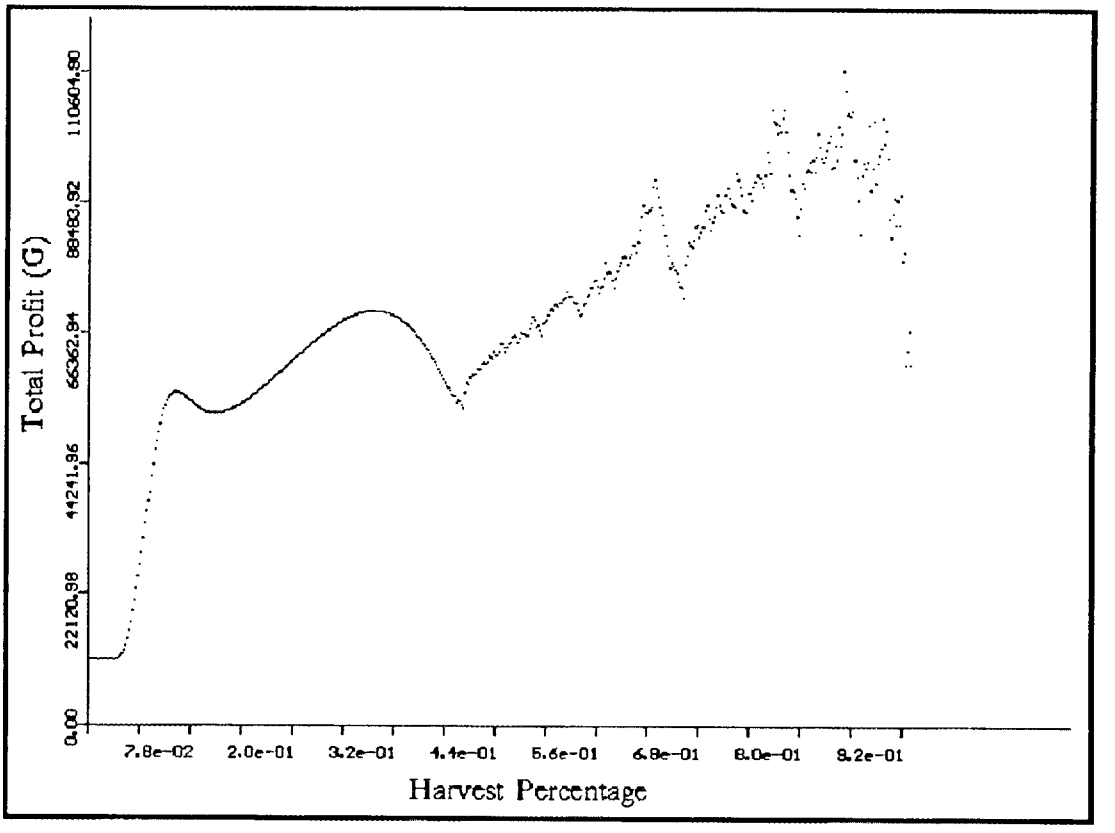

Figure 7. Total profit as a function of harvest percentage. Strategy #2

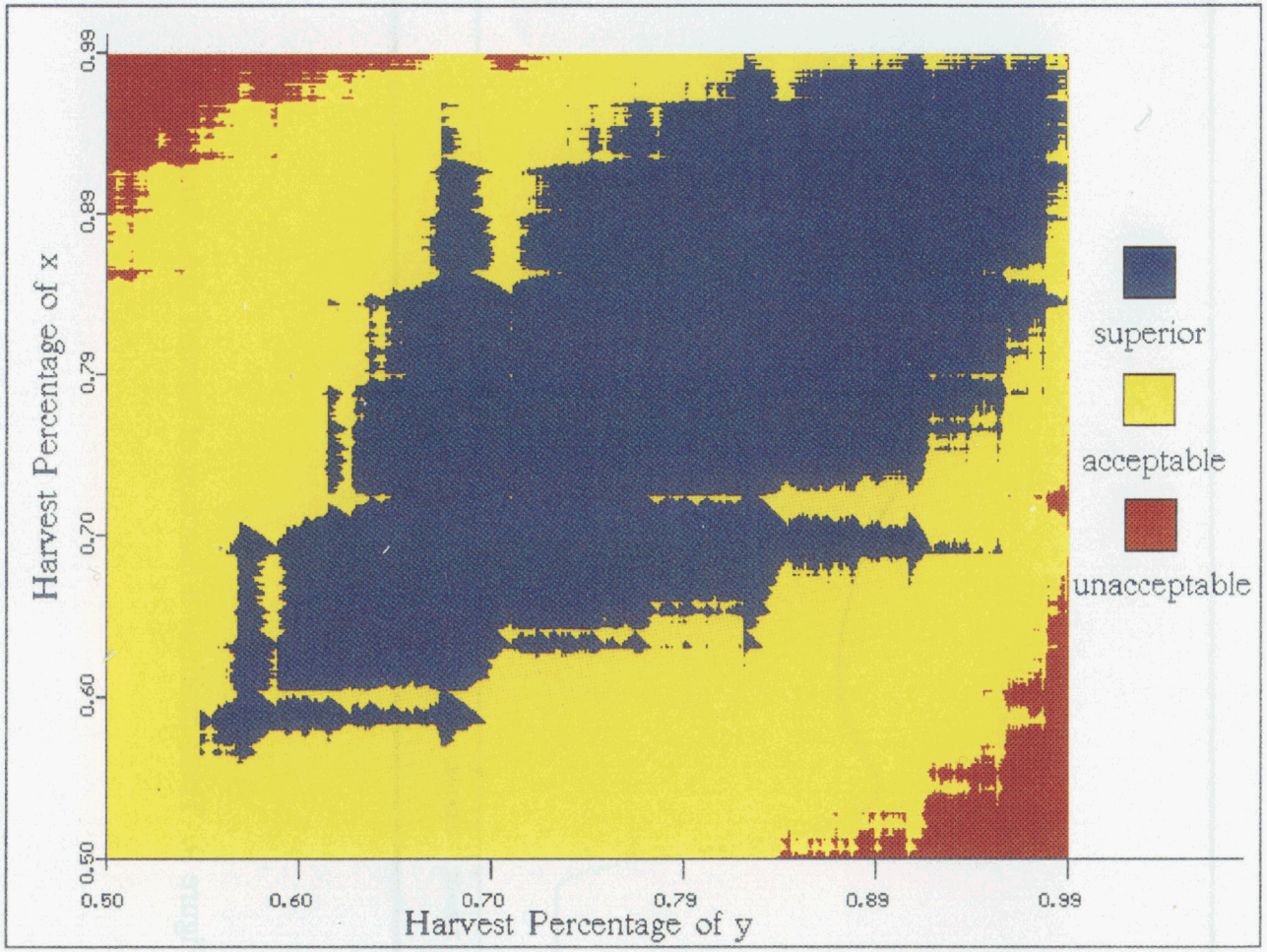

Figure 8. Overall performance of the system as a function of the harvest percentages. Strategy #2

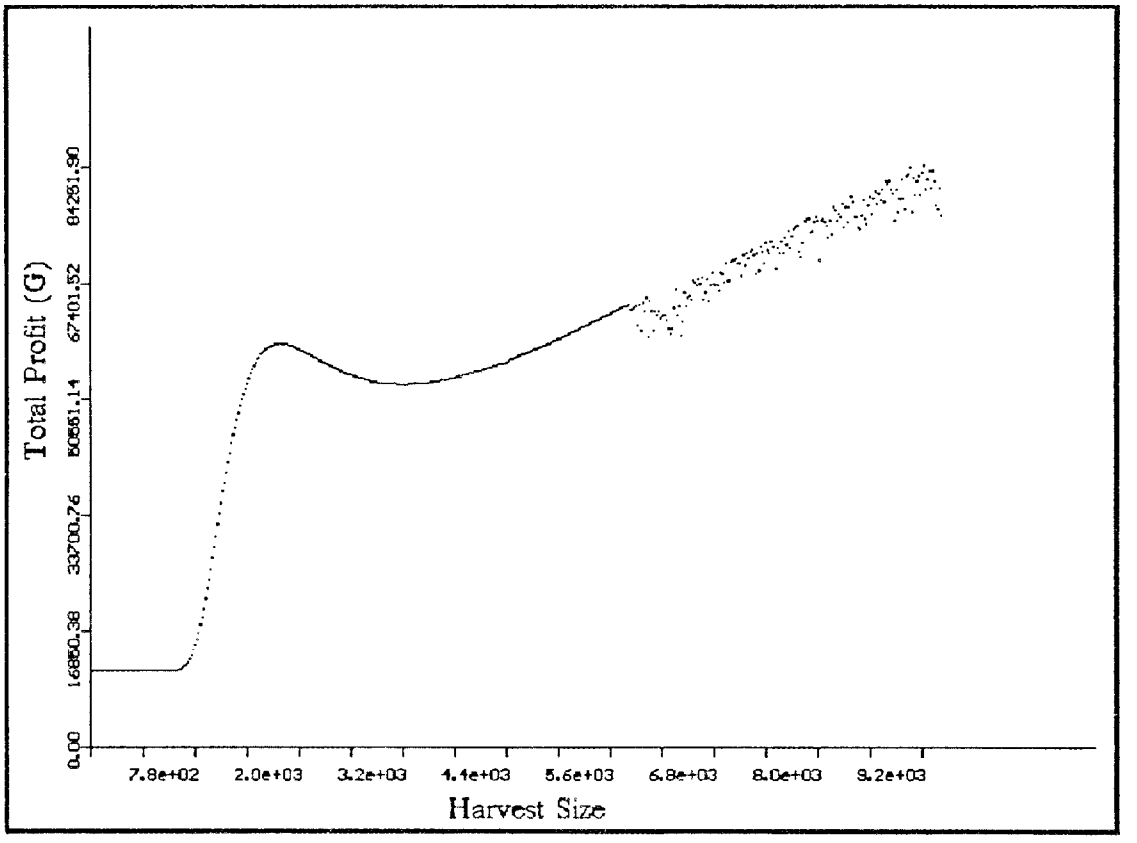

Figure 9. Total profit as a function of harvest size. Strategy #3

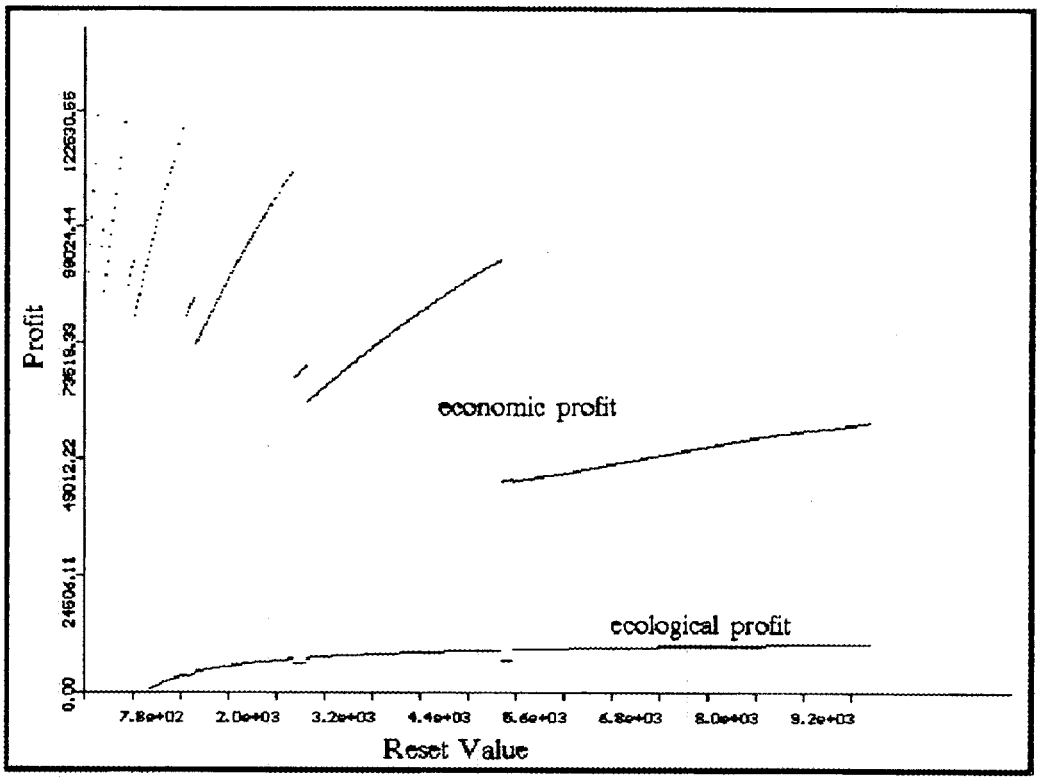

Figure 10a. Economic and ecological profit as a function of reset value. Strategy #4

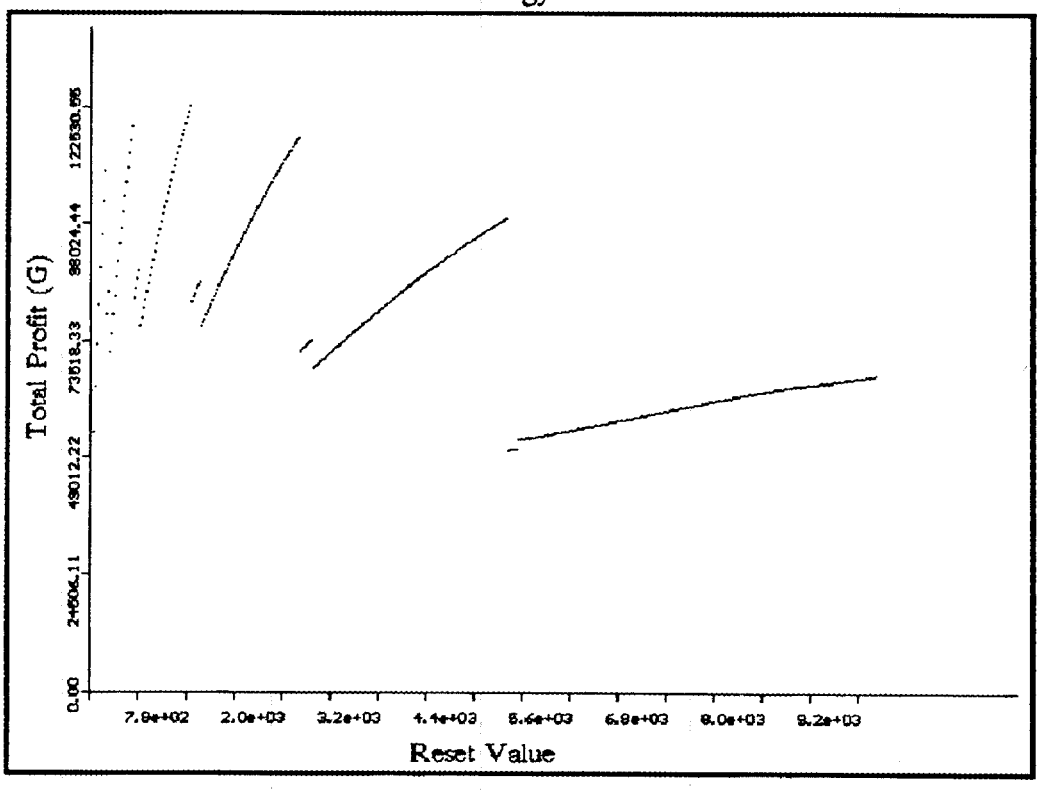

Figure 10b. Total profit as a function of reset value. Strategy #4

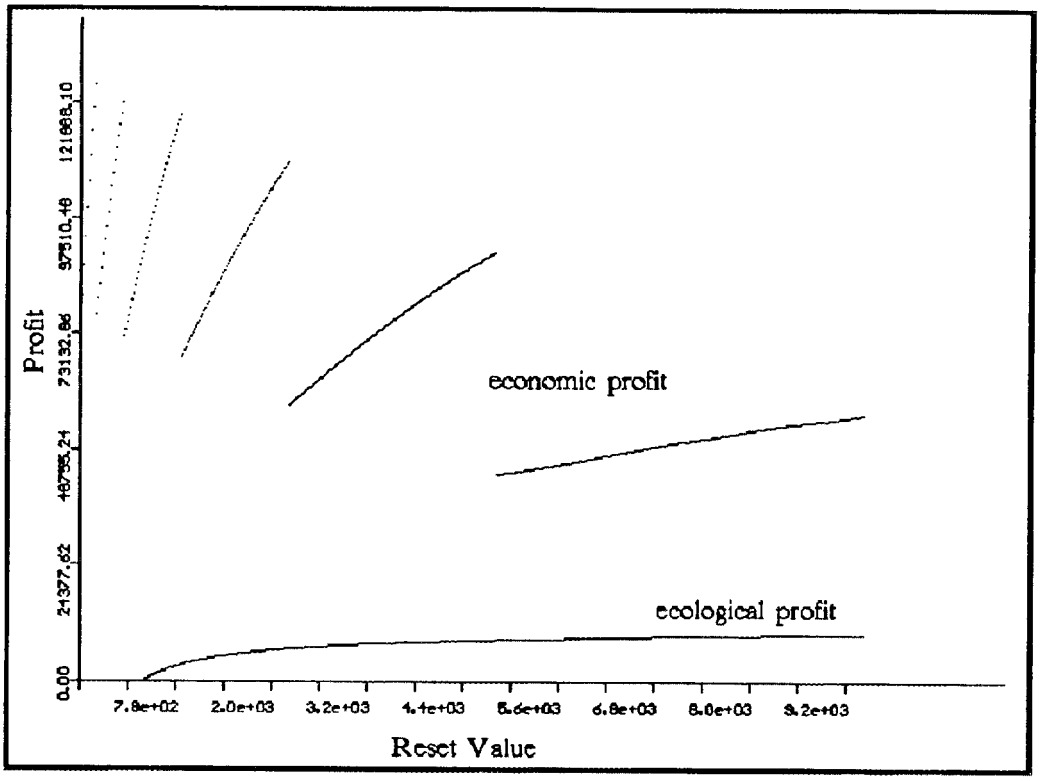

Figure 11a. Economic and ecological profit as a function of reset value. Strategy #5

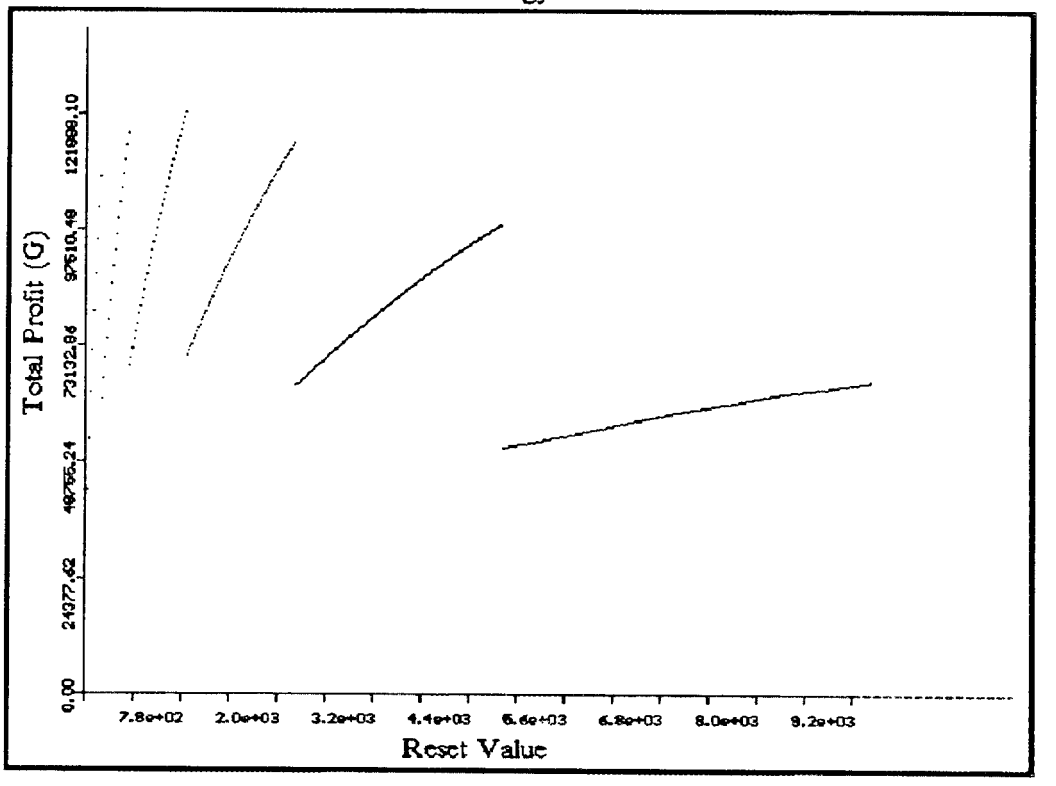

Figure 11b. Total profit as a function of reset value. Strategy #5

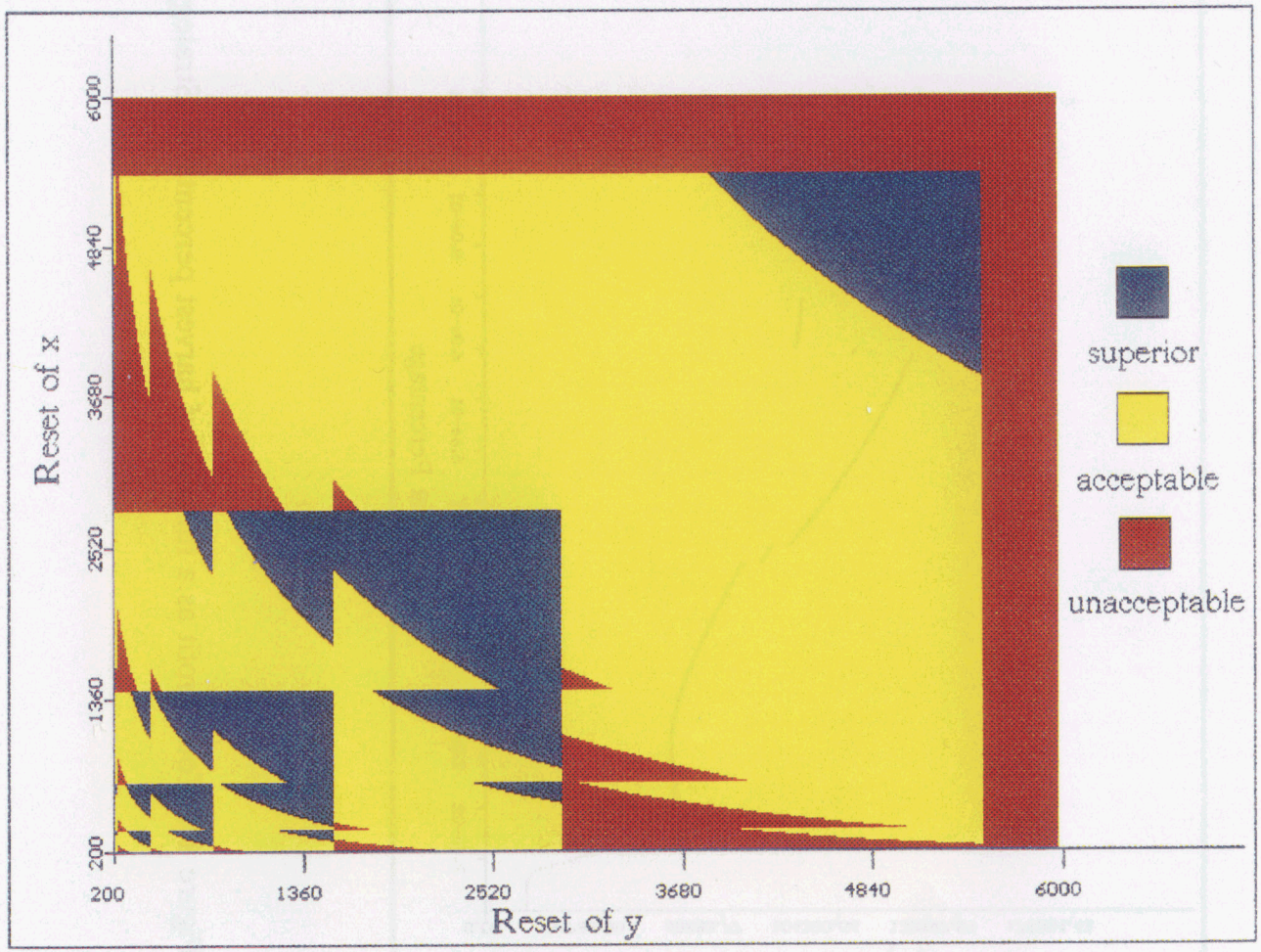

Figure 12. Overall performance of the system as a function of the reset values. Strategy #5

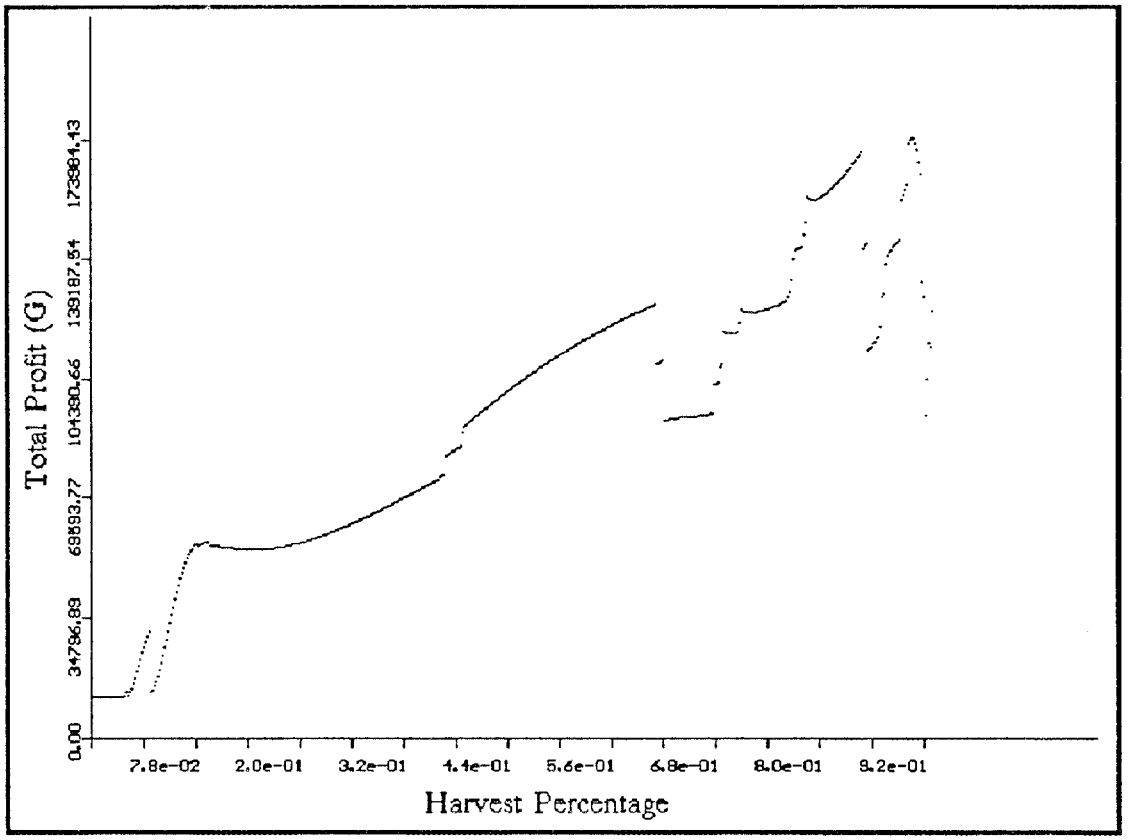

Figure 13. Total profit as a function of harvest percentage. Strategy #6

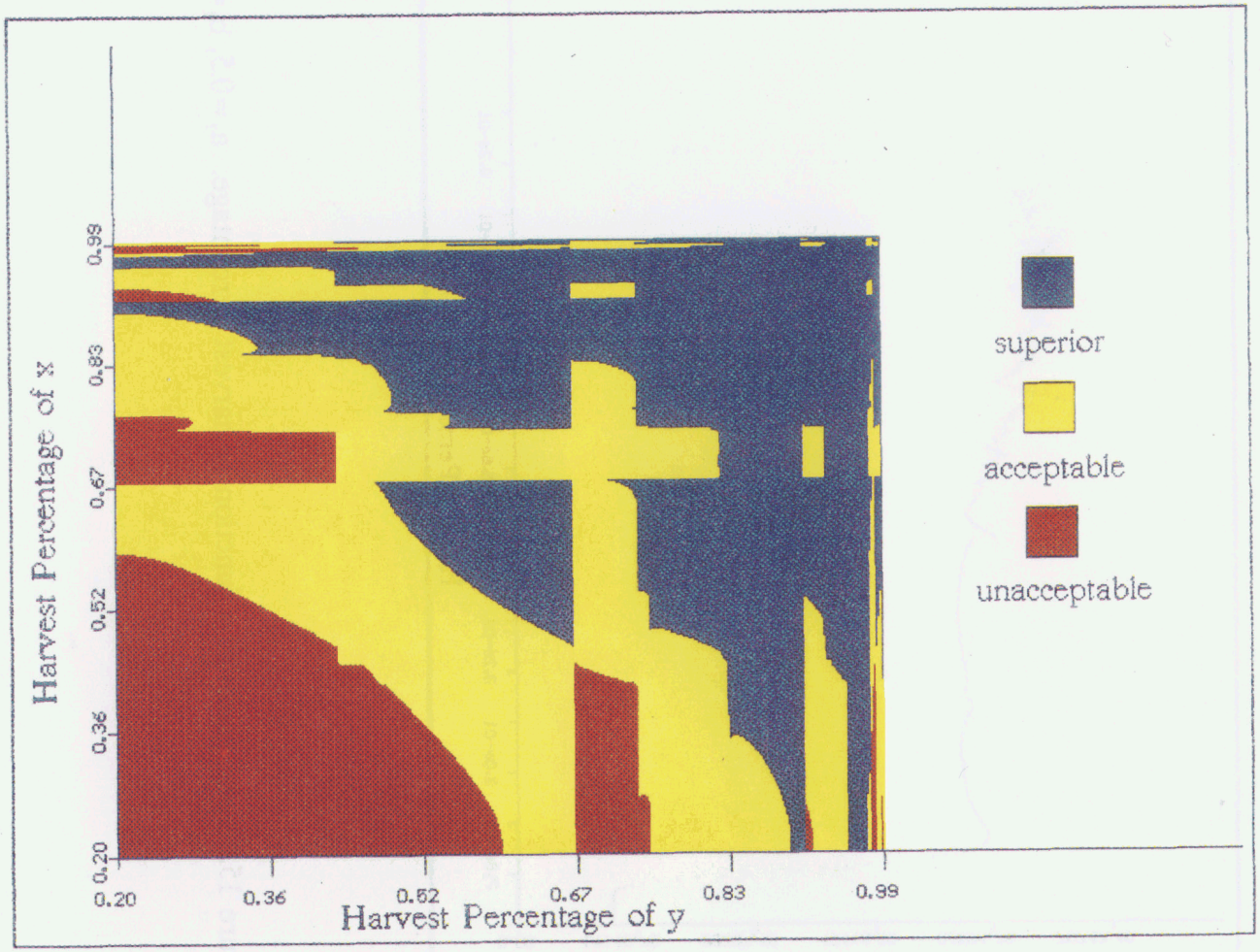

Figure 14. Overall performance of the system as a function of the harvest percentages. Strategy #6

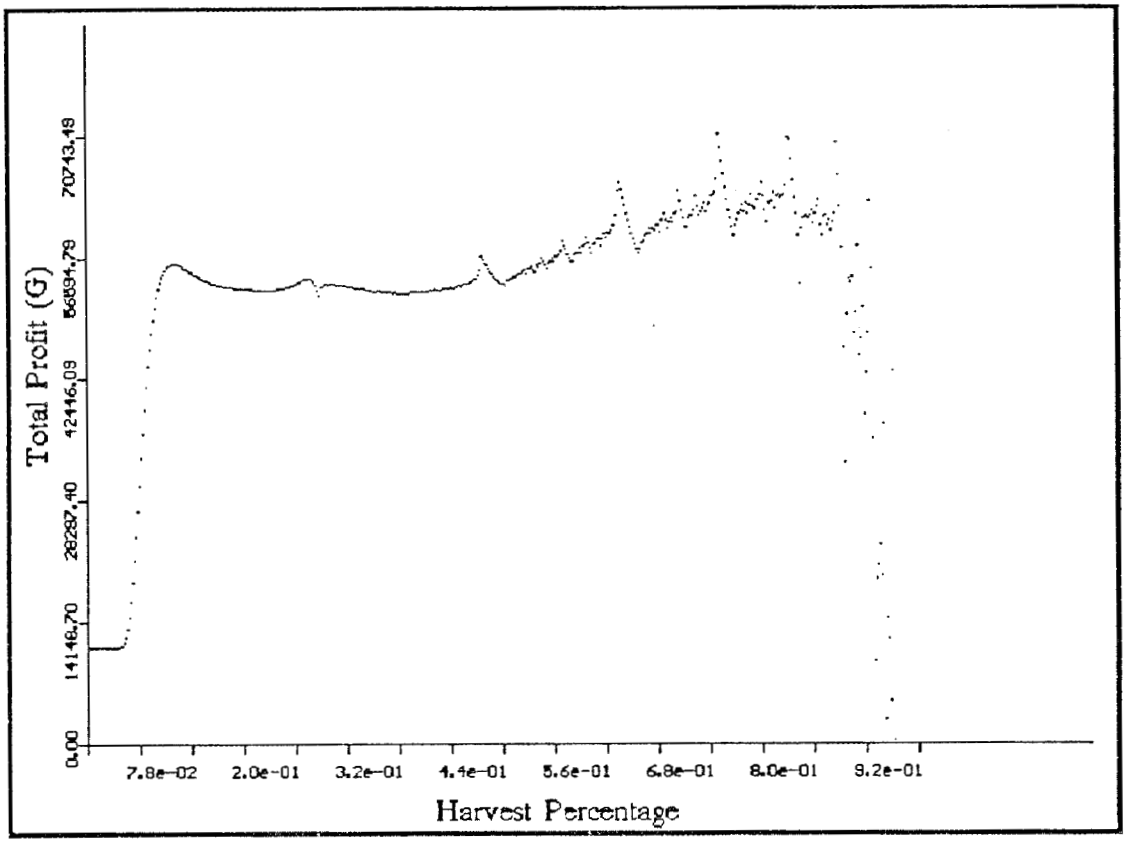

Figure 15. Total profit as a function of harvest percentage. $a_1=0.5$, $b_1=0.45$

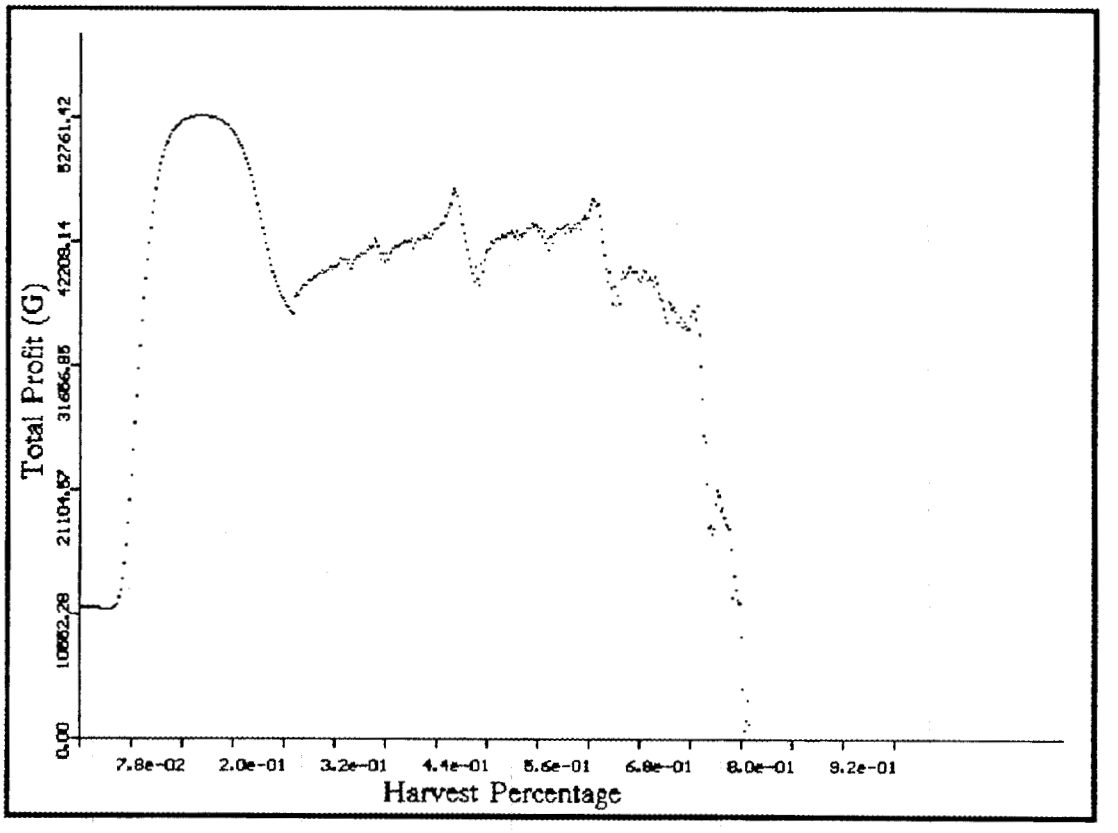

Figure 16. Total profit as a function of harvest percentage.
a_1=0.5, b_1=0.45, a_2=-0.00001, b_2=-0.00001

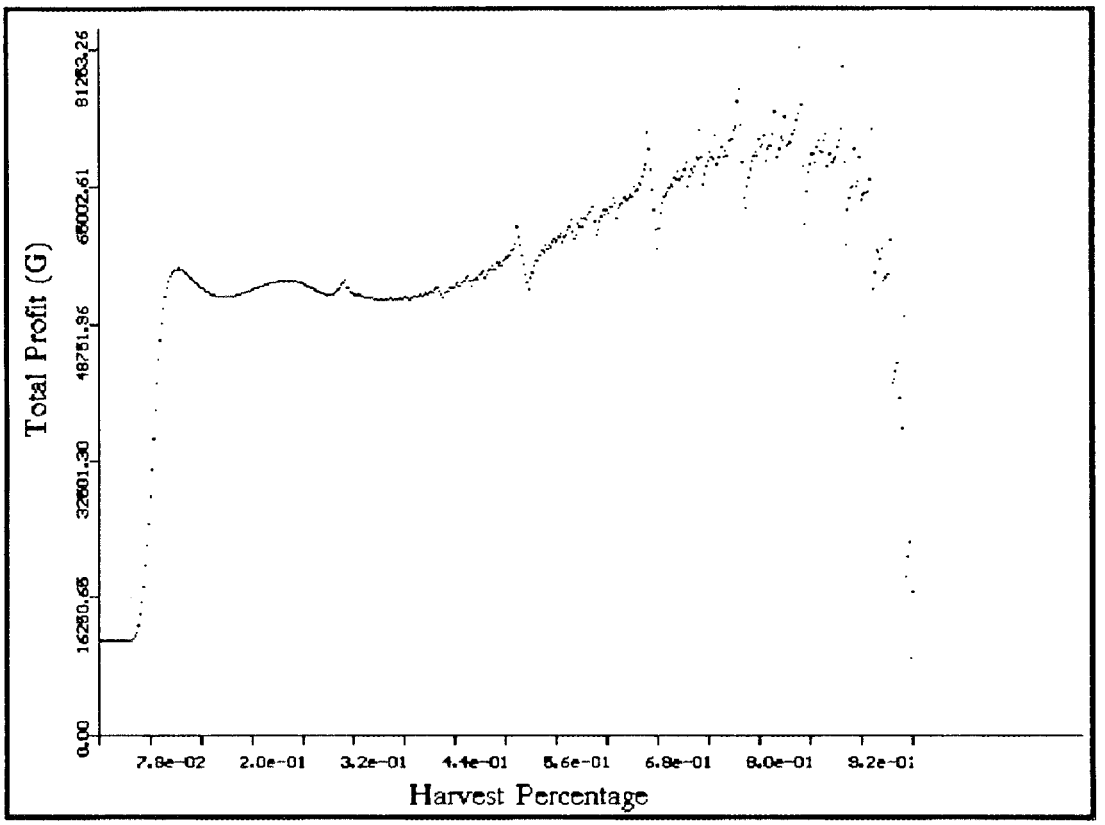

Figure 17. Total profit as a function of harvest percentage.
a_1=0.5, b_1=0.45, a_2=0.00001, b_2=0.00001

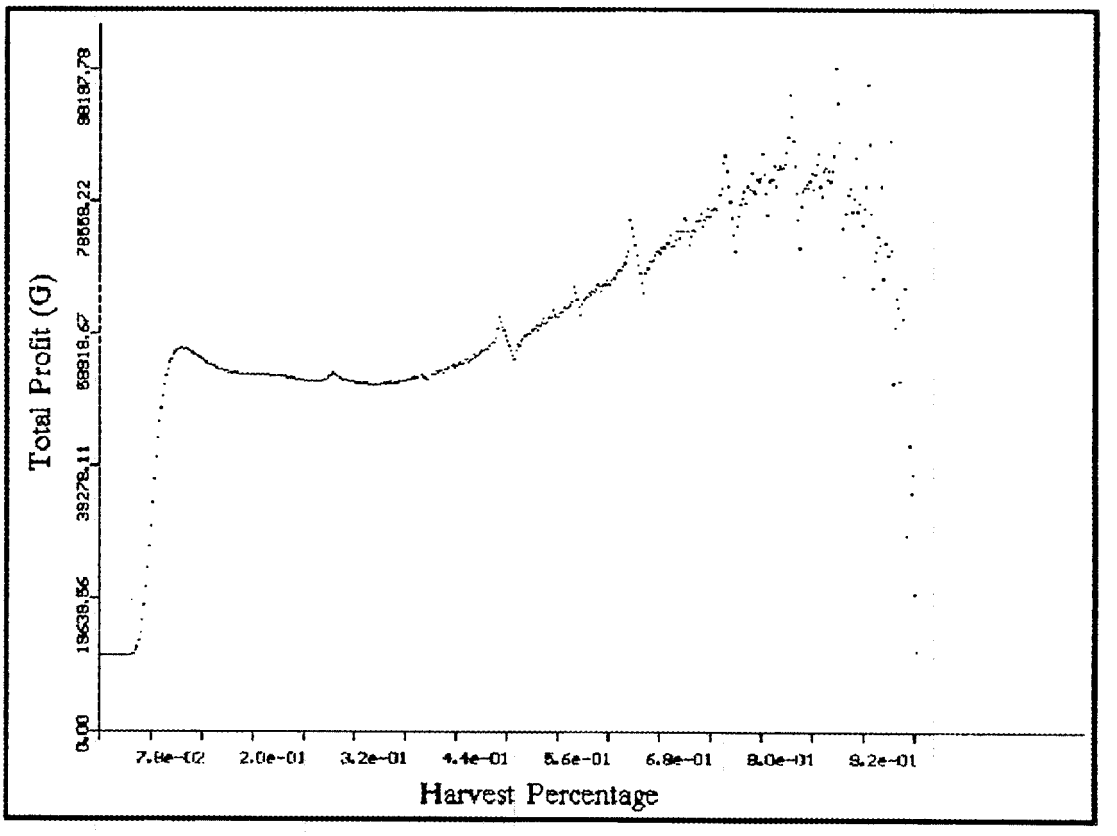

Figure 18. Total profit as a function of harvest percentage.
$a_1=0.5$, $b_1=0.45$, $a_2=0.0$, $b_2=0.00001$

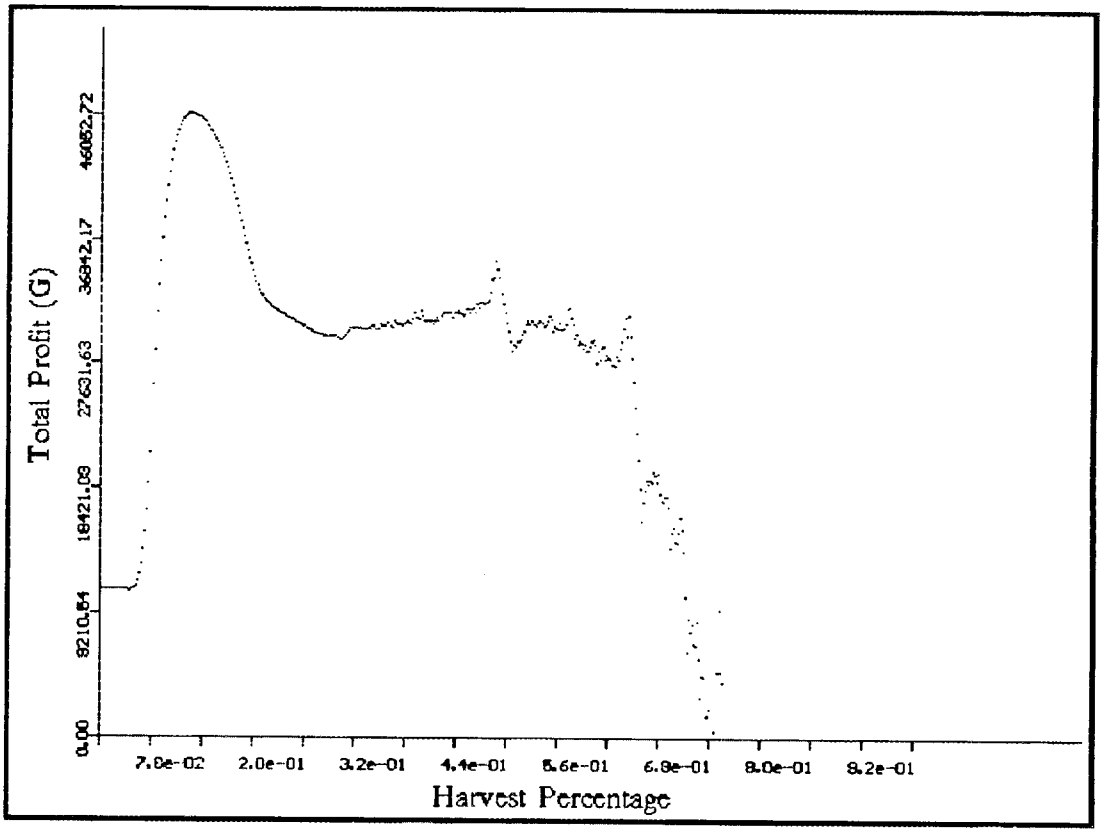

Figure 19. Total profit as a function of harvest percentage.
a_1=0.5, b_1=0.45, a_2=0.00001, b_2=-0.00001

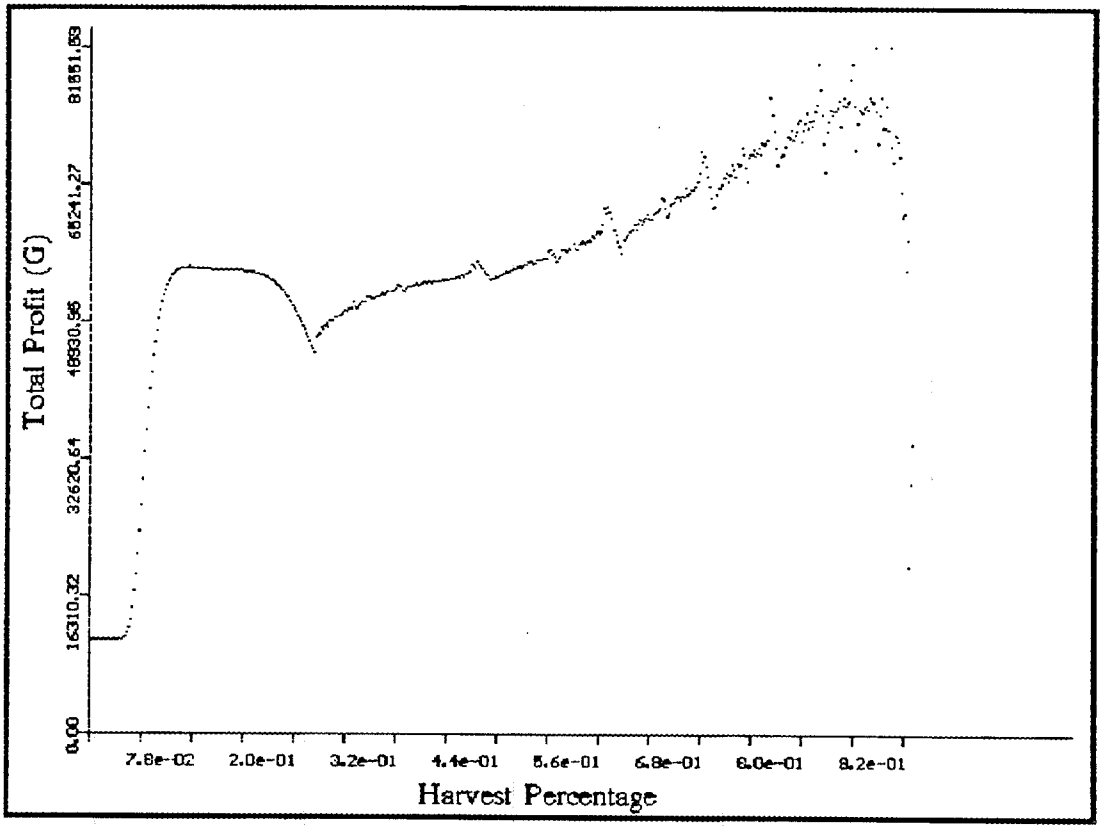

Figure 20. Total profit as a function of harvest percentage.
a_1=0.5, b_1=0.45, a_2=-0.00001, b_2=0.00001

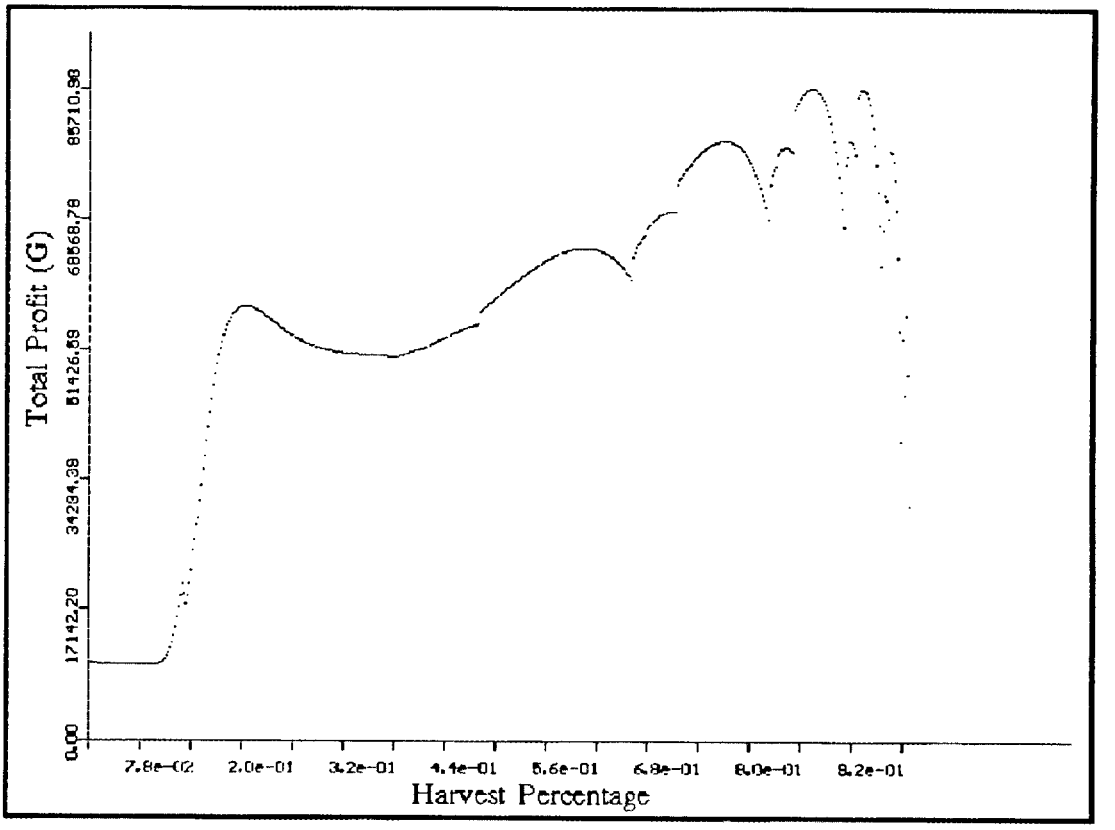

Figure 21. Total profit as a function of harvest percentage. K=L=10,000

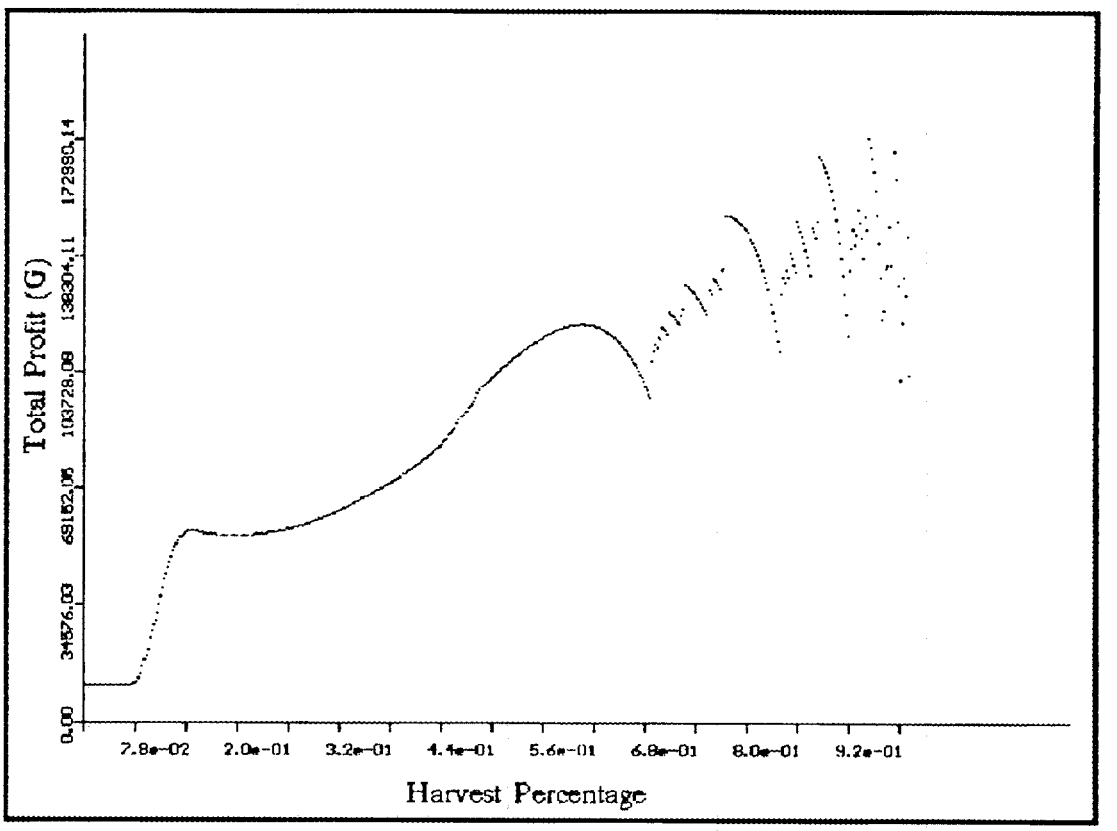

Figure 22. Total profit as a function of harvest percentage. Delay=1

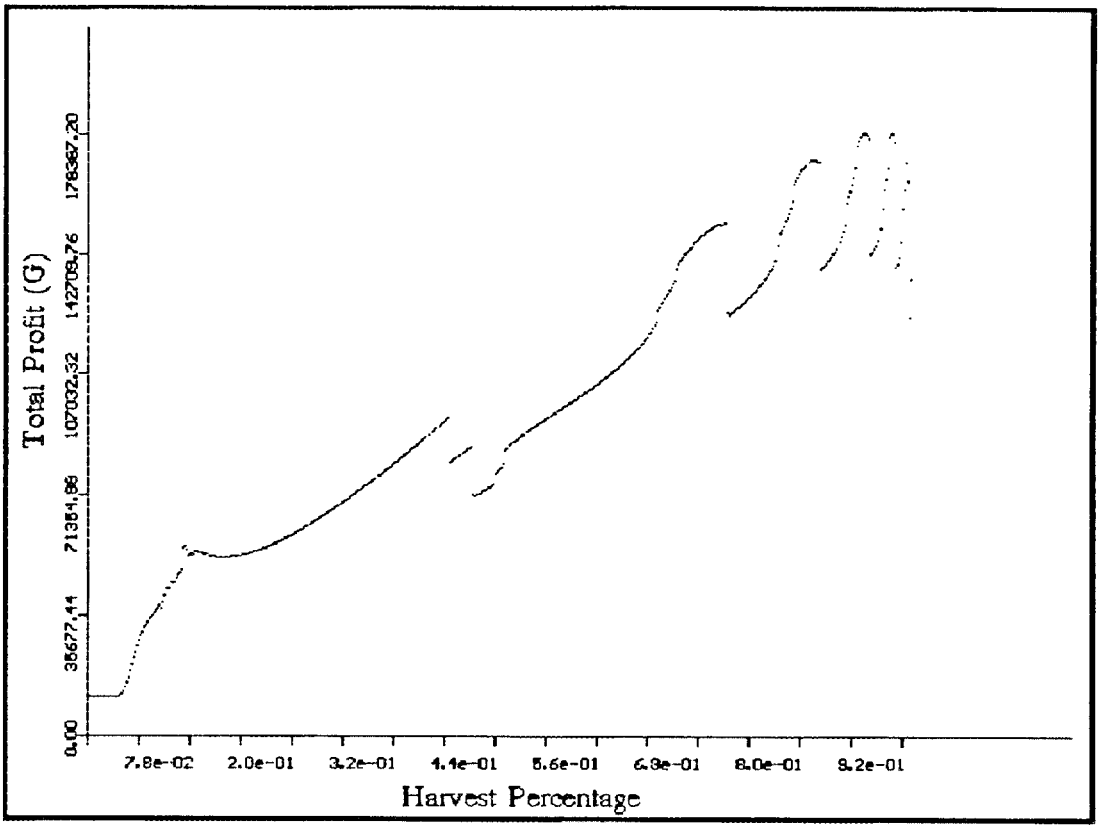

Figure 23. Total profit as a function of harvest percentage. Delay=2

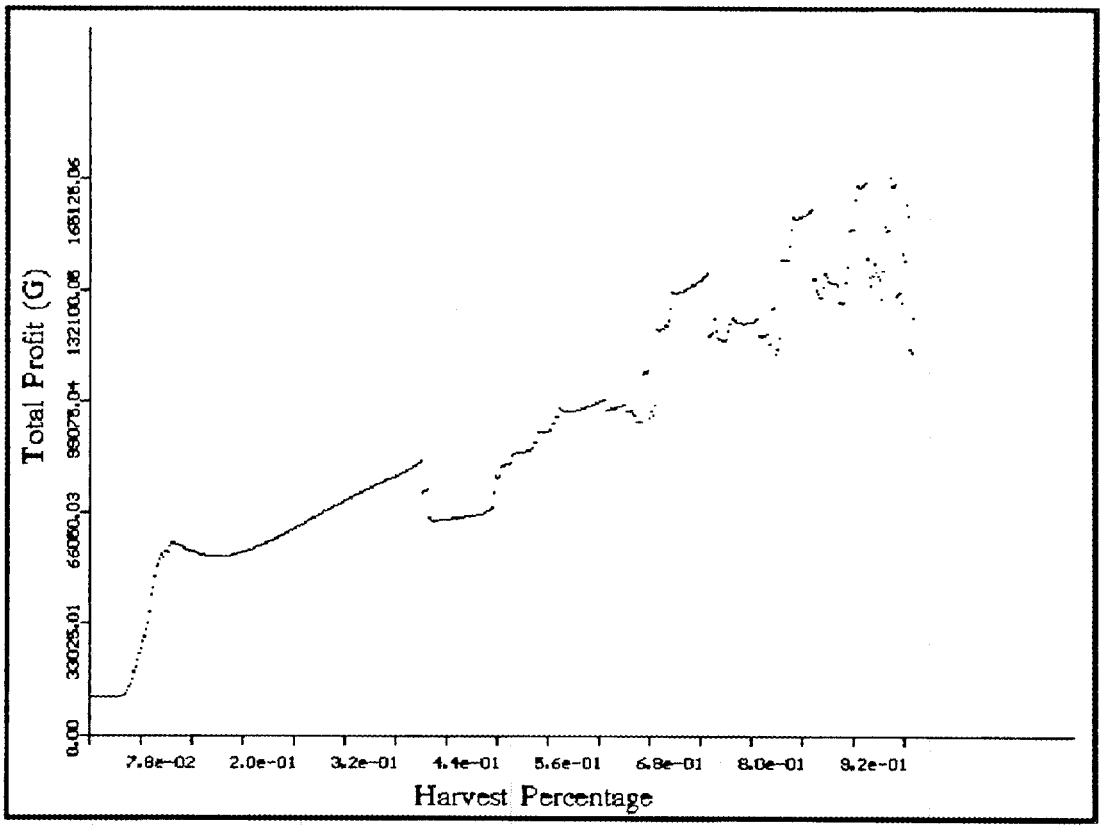

Figure 24. Total profit as a function of harvest percentage. Delay=4

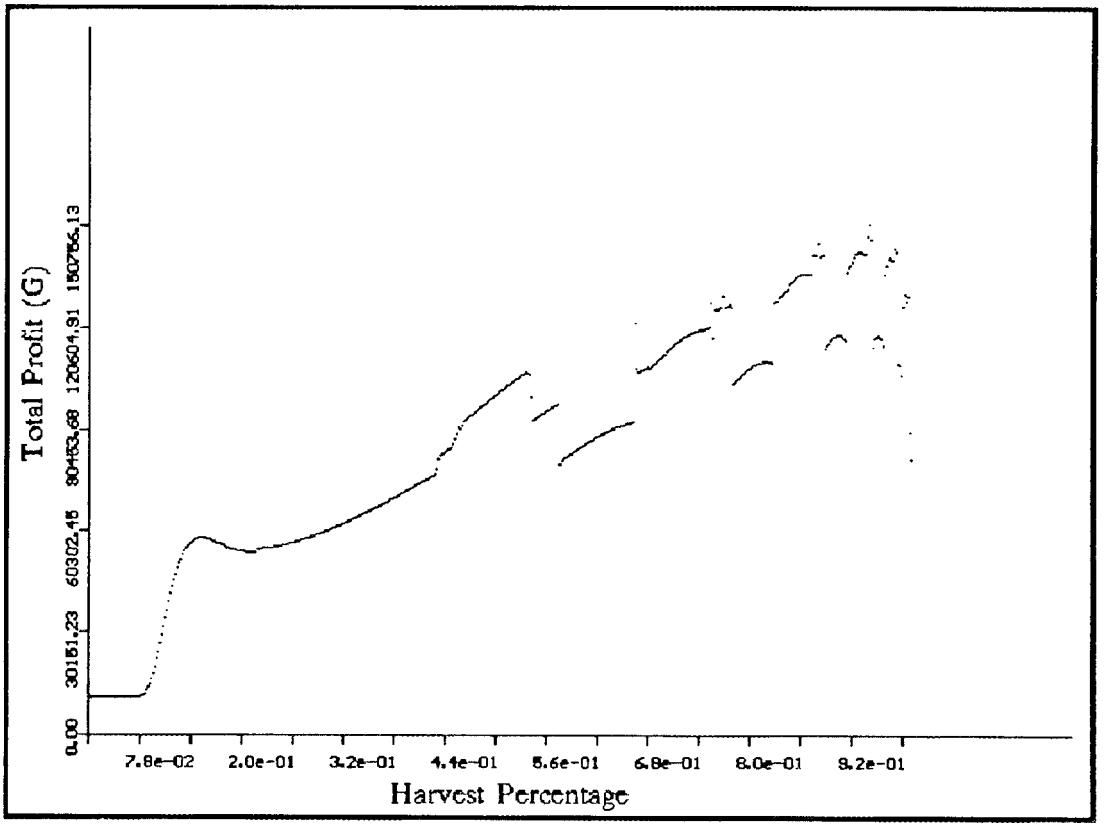

Figure 25. Total profit as a function of harvest percentage. Delay=5

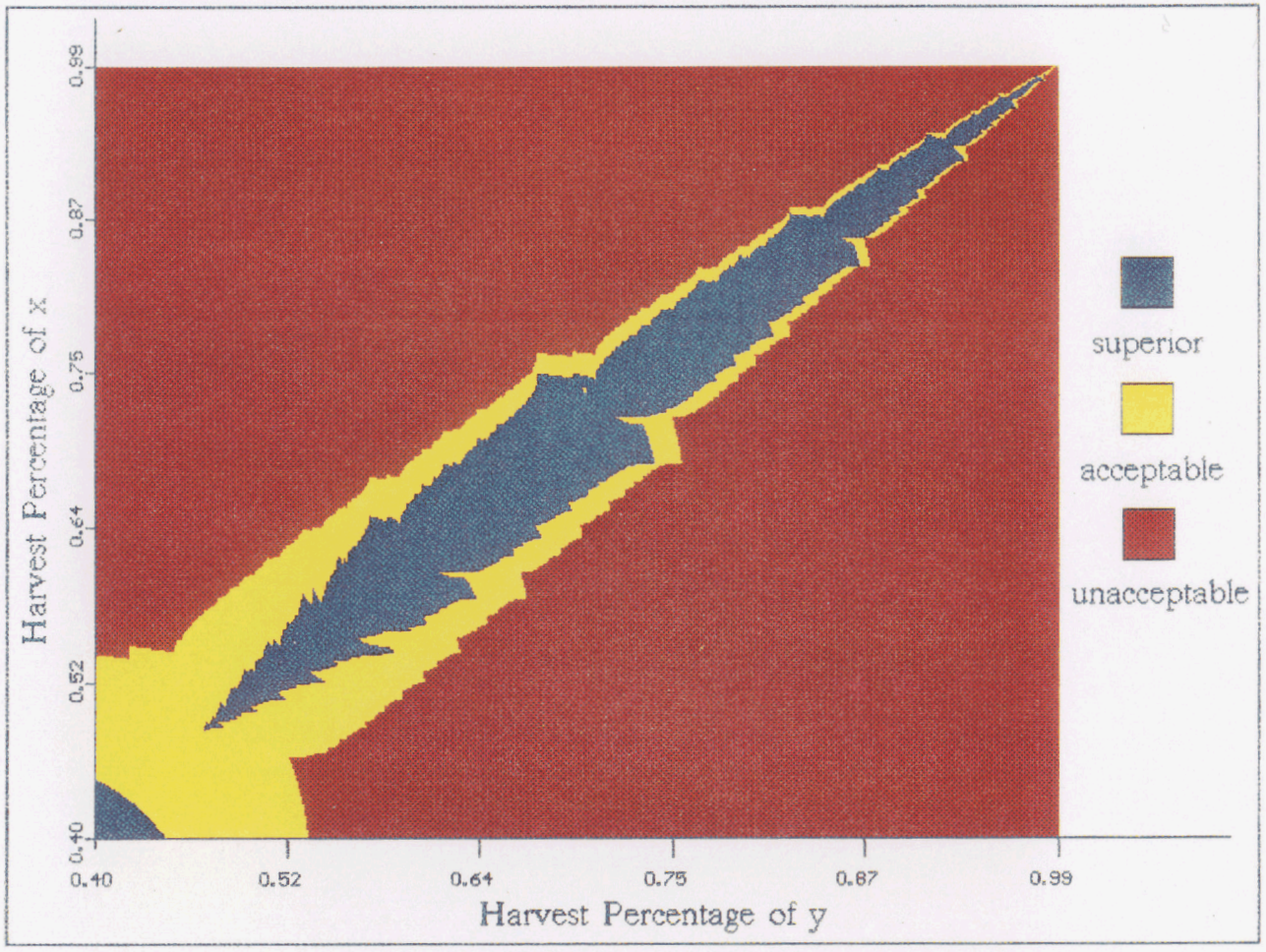

Figure 26. Overall performance of the system as a function of the harvest percentages. Base case

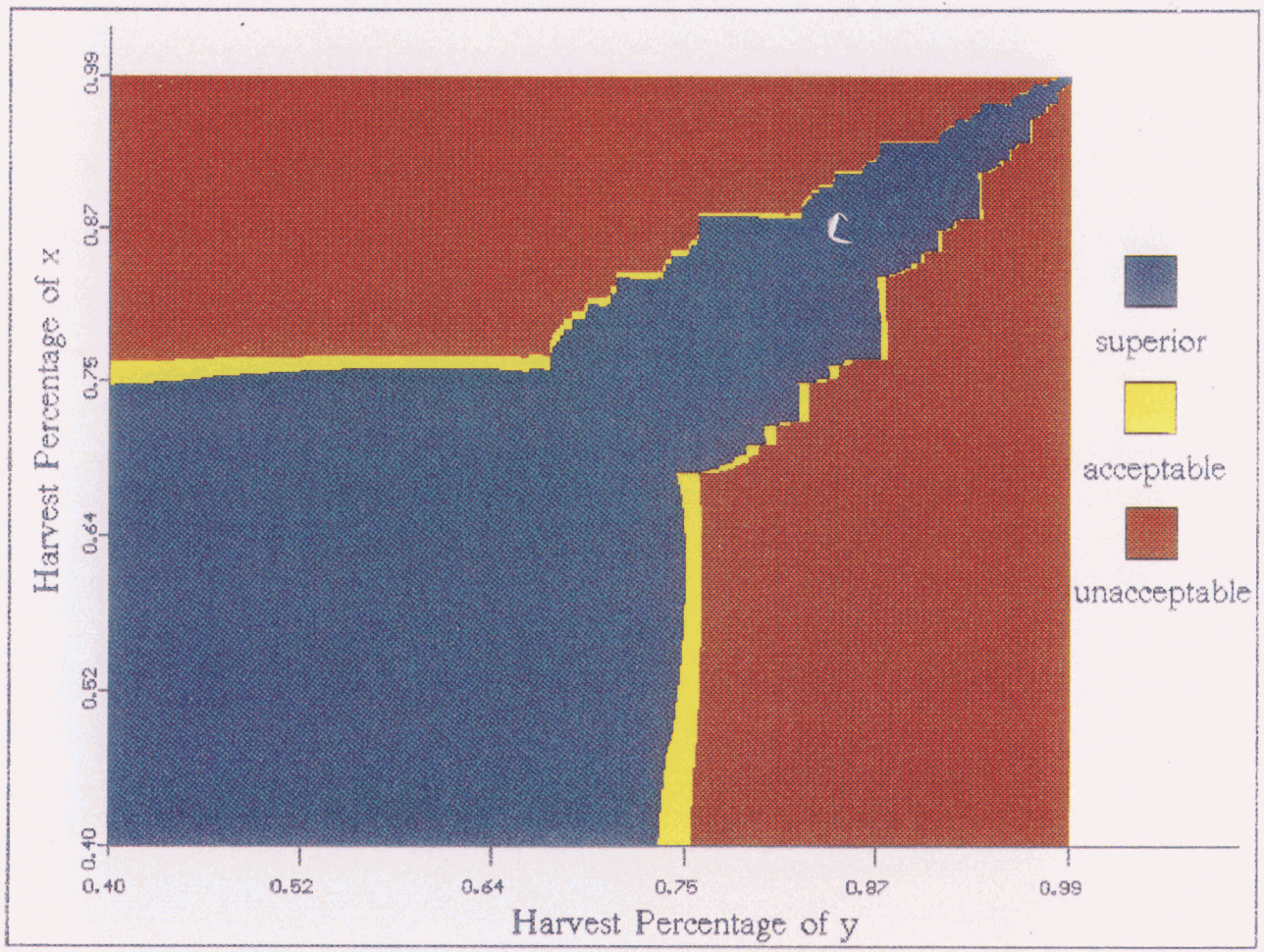

Figure 27. Overall performance of the system as a function of the harvest percentages. Delay = 1

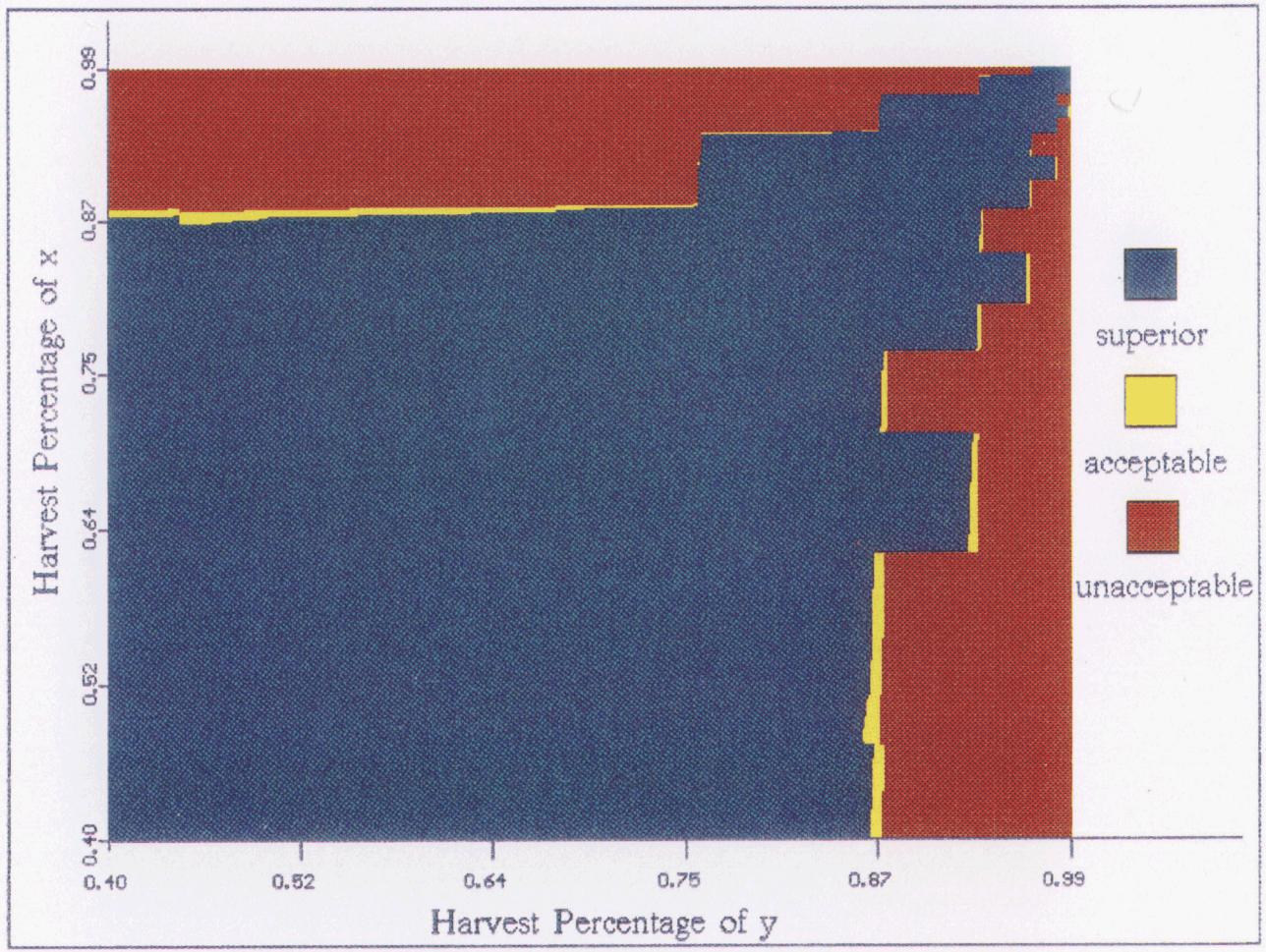

Figure 28. Overall performance of the system as a function of the harvest percentages. Delay = 2

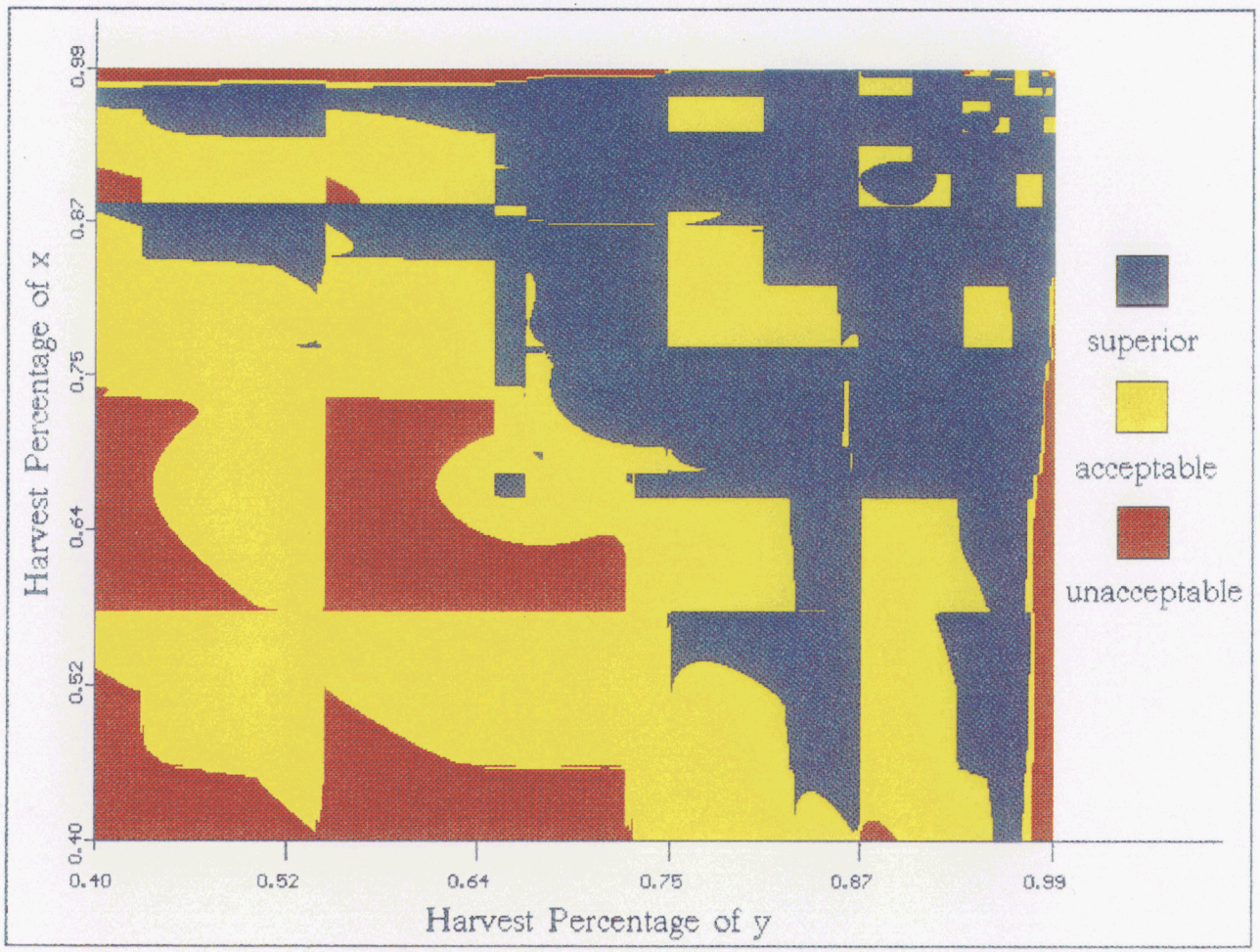

Figure 29. Overall performance of the system as a function of the harvest percentages. Delay = 5

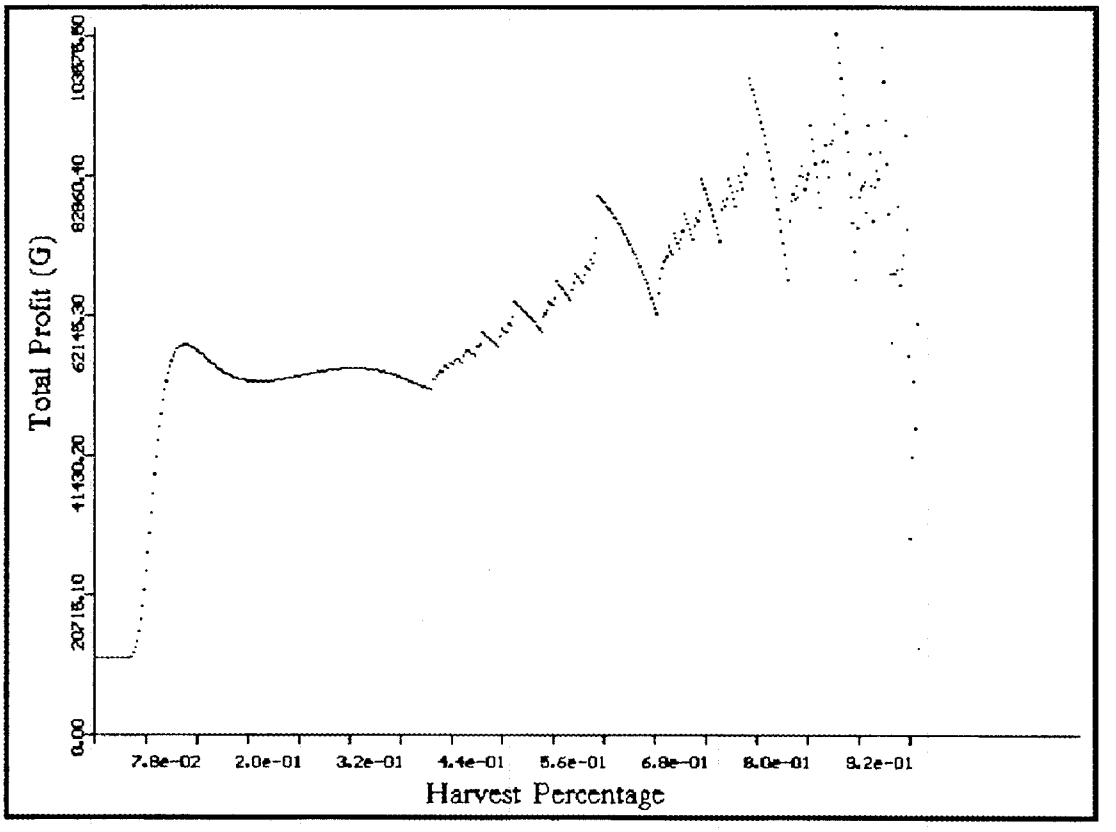

Figure 30. Total profit as a function of harvest percentage. m=1

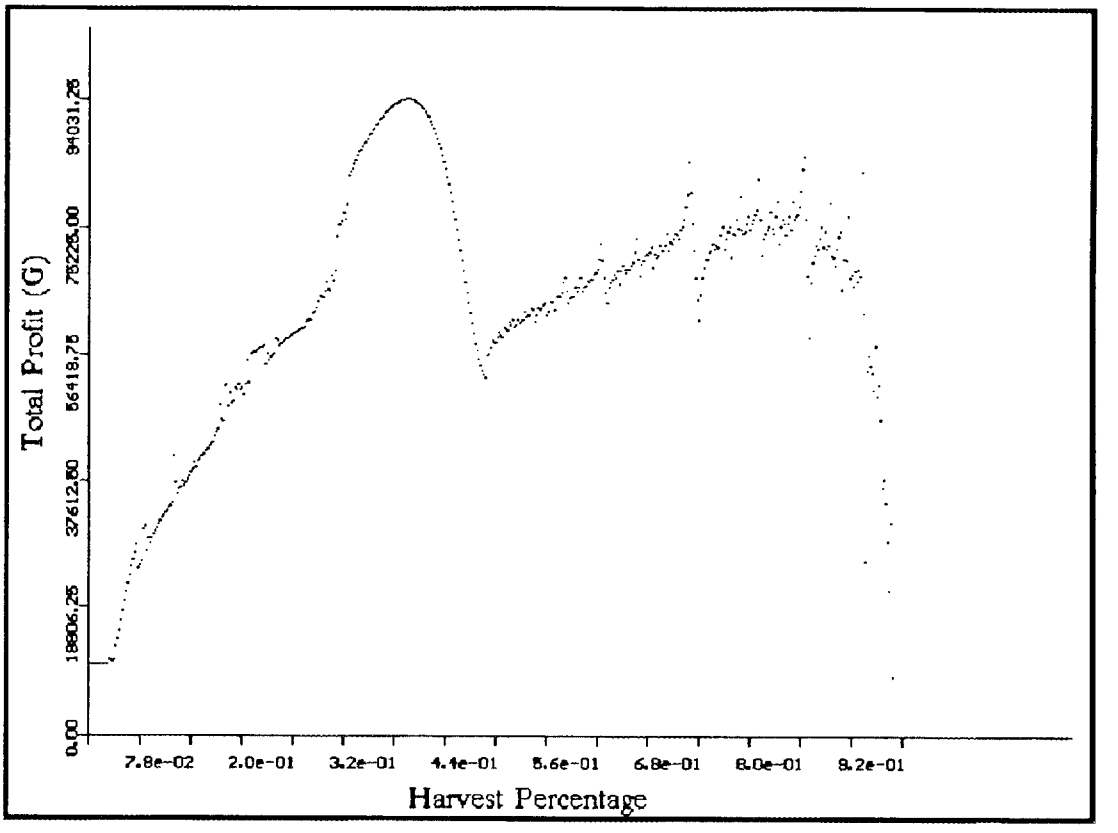

Figure 31. Total profit as a function of harvest percentage. m=1/5

4. DISCUSSION

We have analyzed the sustainability and profitability of an ecological system of two populations harvested periodically for economic reasons. The applications we envisage range from forestry, fishing, greenhouse cultures, and ecological management to — more remotely — economic competition and social texture stability.

A simple dynamical model and its variants have been used to understand the consequences of various parametric changes that would be related to a decision-making policy.

There are two main reasons for using this type of approach:

(i) Ecological interactions can result in widely varying and extremely unstable harvested populations. For multi-population systems subject to nonselective harvesting there are not many rigorous results concerning the existence of optimal harvest policies, and it may very well be that, in general, there is no solution for this type of problem.

(ii) Despite their limitations, mathematical models of complex environmental, political, and economic issues are our only tools to understand, predict, and ultimately modify the outside world. They are crucial to coping with resource management, sustainability, and coexistence. Mathematics and computer-aided agriculture, banking, fishing, investing, arms negotiations, etc., are features of our present world that are here to stay and to develop.

The results of the analysis indicate strong nonmonotonicities of the outcome as a function of the varying parameters as well as large regions of "blurred," non-clear-cut alternating zones in the parameter space. To ensure smooth operations, non-catastrophic behavior, and manageable sensitivities, these zones are to be avoided.

To some extent we should also be able to understand the cause of large sensitivities and catastrophe-like behavior. In particular, we should be able to recognize whether those are due — at least in part — to chaotic behavior in the underlying dynamics. For the bioeconomic model analyzed in this paper this does not seem to be the case: instead, the nonmonotonicities are caused by the discrete nature of the evolution and especially of the harvesting. These nonmonotonicities become continuous if the time step is decreased (Fig. 32).

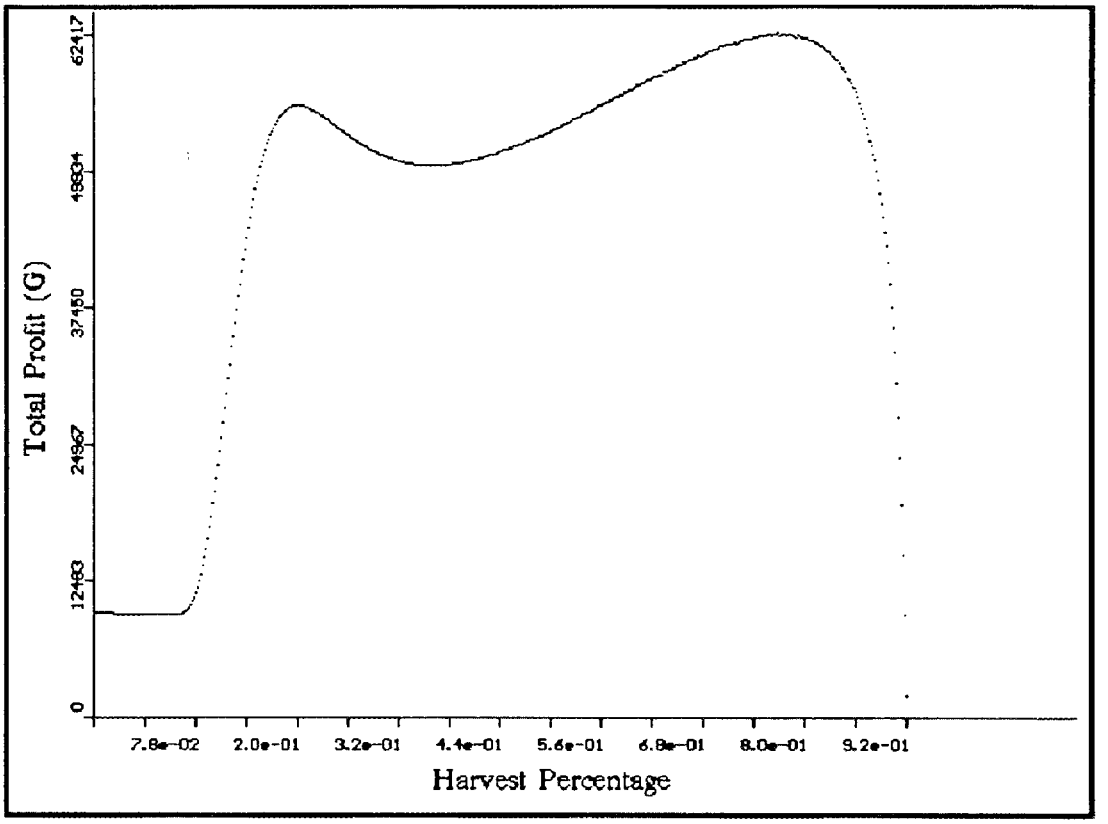

Figure 32. Total profit as a function of harvest percentage with smaller time steps. $a_1 = 0.01$, $b_1 = 0.0095$

Discretization has been recognized to be the cause of similar nonmonotonicities in simpler competitive [10] or mutualistic systems [11].

The work in reference [11] considers only the ecological effects of harvesting on a mutualistic system. Thus a different objective functional is studied which captures the persistence quality of the two-population system after N harvests within strategy #3. A population persists if the ratio of its size to the size of the other population is larger than the assigned limit, so x persists if $\frac{x_n}{y_n} \geq x.\text{ratio}$ and y persists if $\frac{y_n}{x_n} \geq y.\text{ratio}$ ($x.ratio = 0.22, y.ratio = 0.28$). A graph then represents which persists with red for x, blue for y, and white for both. The 2-D graph in Fig. 33 shows the ecological results of varying the reset value of the two populations for $a_1 = b_1 = 0.2$ and $a_2 = b_2 = 0.00011$.

From the viewpoint of direct applications, the principal weakness of bioeconomic models, like the one analyzed in this paper, is related to the identification problem, namely estimating the model's parameters in order to test and use its, albeit limited, predictive or prescriptive power. These aspects must be further analyzed in future work.

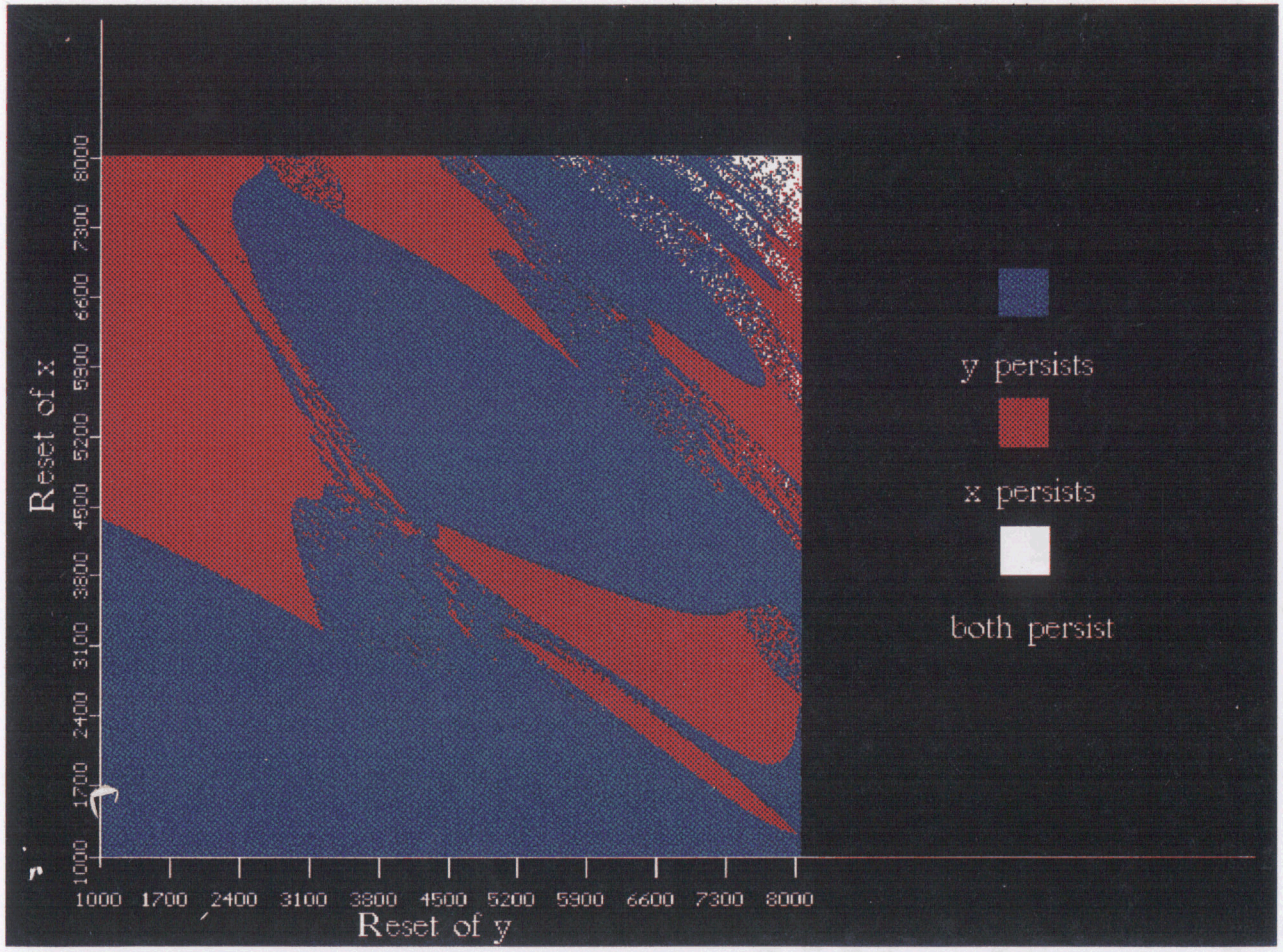

Figure 33. Persistence of the populations as a function of reset values. Strategy #3, $a_1 = b_1 = 0.2$, $a_2 = b_2 = 0.00011$

REFERENCES

1. R. Goodland and H. Daly, Approaching Global Environmental Sustainability, Preprint, 1992.

2. J. Roughgarden, Theory of Population Genetics and Evolutionary Ecology: An Introduction, Macmillan Publishing Company, Inc., New York, 1979.

3. E. Beltrami, Mathematics for Dynamic Modeling, Academic Press, Inc., Boston, 1987.

4. Perspectives in Ecological Theory, J. Roughgarden, R. M. May, and S. A. Levin, Eds., Princeton University Press, Princeton, NJ, 1989.

5. C. W. Clark, Mathematical Bioeconomics, The Optimal Management of Renewable Resources, John Wiley & Sons, Inc., New York, 1990.

6. D. L. DeAngelis, W. M. Post, and C. C. Travis, "Positive Feedback in Natural Systems," Biomathematics 15, Springer Verlag, Berlin, 1986.

7. M. D. Bertness, "The Ecology of a New England Salt Marsh," Amer. Scientist **80**, 260 (1992).

8. L. S. Pontryagin, V. S. Boltyanskii, R. V. Gamkrelidze, and E. F. Mischenko, The Mathematical Theory of Optimal Processes, Wiley-Interscience, New York, 1962.

9. M. D. Canon, C. D. Cullum, and E. Polak, Theory of Optimal Control and Mathematical Programming, McGraw-Hill, New York, 1970.

10. J. A. Dewar, J. J. Gillogly, and M. L. Juncosa, Non-Monotonicity, Chaos, and Combat Models, RAND Report R-3995-RC, 1991.

11. V. Protopopescu and R. L. Melancon, Persistence and Sustainability in Mutualistic Systems with Harvesting, unpublished results, 1991.

APPENDIX A - OPTIMAL CONTROL FOR DISCRETE-TIME SYSTEMS

In an optimal control problem, the state variables (x_k, y_k) describe the state of the system at time step k, $k=1,2,...,N$. Equations (2.1) are the state equations and describe the evolution of the system from its initial state (x_0, y_0), as a result of the application of the controls (H^1, H^2). The terminal time step, N, is called the time horizon and usually is finite.

Given a set of controls (H_k^1, H_k^2), $k=0,1,...,N$, the corresponding solution (x_{k+1}, y_{k+1}) is referred to as the response. The controls are assumed to belong to a specified control set, or a class of admissible controls. Any admissible control that determines a response satisfying the initial condition is called feasible control.

Usually one wants to control a system in order to achieve a certain goal or preassigned task. The measure of achieving the goal (task) is quantified by the objective functional:

$$G(H^1,H^2) = \sum_{k=1}^{N} G_k(x_k, y_k, H_k^1, H_k^2)$$

The fundamental problem in optimal control theory is to determine the feasible control that maximizes G. If it exists, such a control is called the optimal control. Proving the existence of optimal controls is, in general, difficult, but necessary conditions are provided by Pontryagin's maximum principle (see references [8,9]). This principle works in both continuous and discrete forms, but due to the characteristics of our problem, we shall give here a short description of the discrete time version.

As mentioned before, the objective functional is constructed as the sum over all harvests of the ecological reward plus the economic profit. We note that the final controls (H_N^1, H_N^2) do not influence the dynamics but do affect the value of the objective functional.

To be specific, we consider that nonselective harvesting occurs after n steps of autonomous evolution and is performed N times. We can rewrite system (2.1) as

$$\begin{aligned} x_{k+1} &= x_k + F_k^1 (x_k, y_k, H_k^1, H_k^2) \\ y_{k+1} &= y_k + F_k^2 (x_k, y_k, H_k^1, H_k^2) \end{aligned}$$

$$k = 0,1,...,N-1 \qquad \text{(A.1)}$$

where the effects of the autonomous evolution between successive harvestings are contained in the new nonlinear maps F^1 and F^2.

The optimal control problem for this new discrete evolution system consists in finding the controls (H_k^1, H_k^2), $k=0,1,\ldots,N$, that maximize the value of the objective functional G.

As a necessary condition for optimality we apply the maximum principle in its Lagrangian form [8,9]. In analogy with mechanical systems we define the Lagrangian [5,8]:

$$\begin{aligned}\mathcal{L}(x, y, H^1, H^2) = \sum_{k=1}^{N} \{&G_k(x_k, y_k, H_k^1, H_k^2) - \\ &\lambda_k^1 (x_{k+1} - x_k - F_k^1 (x_k, y_k, H_k^1, H_k^2)) - \\ &\lambda_k^2 (y_{k+1} - y_k - F_k^2 (x_k, y_k, H_k^1, H_k^2))\}.\end{aligned} \tag{A.2}$$

The additional unknown functions λ_k^1, λ_k^2 are called adjoint variables and have somewhat the significance of a "price" associated with the state variables x_k, y_k respectively. Since the state equations do not apply any longer for $k=N$, we can take $\lambda_N^1 = \lambda_N^2 = 0$ (transversality conditions) [8].

If (H_k^1, H_k^2) is the optimal control and (x_k, y_k) the corresponding response, the maximum principle asserts the existence of adjoint variables $(\lambda_k^1, \lambda_k^2)$ such that the following equations are satisfied:

$$\frac{d\mathcal{L}}{dx_k} = 0, \quad \frac{d\mathcal{L}}{dy_k} = 0 \quad k=1,2,\ldots,N-1 \tag{A.3}$$

$$\frac{d\mathcal{L}}{dH_k^1} = 0, \quad \frac{d\mathcal{L}}{dH_k^2} = 0 \quad k= 0,1,\ldots,N. \tag{A.4}$$

$$\frac{d\mathcal{L}}{d\lambda_k^1} = 0, \quad \frac{d\mathcal{L}}{d\lambda_k^2} = 0 \quad k = 0,1,\ldots,N-1. \tag{A.5}$$

Eqs. (A.3)-(A.5) are nothing but the necessary conditions of extremality for the functional $\mathcal{L}$ depending on the set of variables x, y, λ^1, λ^2, H^1, H^2. These conditions provide $6N$ equations to determine $6N$ unknowns: $x_1,\ldots,x_{N-1}$; $y_1,\ldots,y_{N-1}$; $H_0^1,\ldots,H_N^1$; $H_0^2,\ldots,H_N^2$; $\lambda_0^1,\ldots,\lambda_{N-1}^1$; $\lambda_0^2,\ldots,\lambda_{N-1}^2$. Equations (A.3) and (A.4) give:

$$\frac{dG_k}{dx_k} + \lambda_k^1 + \lambda_k^1 \frac{dF_k^1}{dx_k} + \lambda_k^2 \frac{dF_k^2}{dx_k} - \lambda_{k-1}^1 = 0$$

$$k = 1,2,\ldots,N-1 \tag{A.6}$$

$$\frac{dG_k}{dy_k} + \lambda_k^2 + \lambda_k^2 \frac{dF_k^2}{dy_k} + \lambda_k^1 \frac{dF_k^1}{dy_k} - \lambda_{k-1}^2 = 0$$

$$k = 1,2,\ldots,N-1$$

$$\frac{dG_k}{dH_k^1} + \lambda_k^1 \frac{dF_k^1}{dH_k^1} + \lambda_k^2 \frac{dF_k^2}{dH_k^1} = 0$$

$$k = 0,1,\ldots,N \tag{A.7}$$

$$\frac{dG_k}{dH_k^2} + \lambda_k^1 \frac{dF_k^1}{dH_k^2} + \lambda_k^2 \frac{dF_k^2}{dH_k^2} = 0$$

$$k = 0,1,\ldots,N,$$

while Eqs. (A.5) merely restate the evolution equations (A.1).

As mentioned before, the existence and uniqueness of optimal controls are not guaranteed by Eqs. (A.3)-(A.5) which only represent necessary (but not sufficient) conditions. Moreover, system (A.3)-(A.5) is a two-point boundary-value problem (the state variables evolve forward in time starting from the initial condition (x_0, y_0), while the adjoint variables evolve backward in time starting from the final condition $(\lambda_N^1, \lambda_N^2))$ which renders its numerical solution unstable and computationally intensive.

There are many situations in which: (i) either it is known that an optimal control fails to exist, or (ii) it is not known whether an optimal control exists. Lack of or uncertainty about an optimal solution may sometimes indicate failures in modeling and can be corrected by altering the concept of admissible control or by revising the model, which is by no means an easy task. Actually, there is always the possibility that severe dynamic instabilities may be introduced into nonlinear bioeconomical models by otherwise benign-looking modifications.

Factors that have the potential of inducing instabilities in bioeconomic models include [5]: demand inelasticity, depensation effects in the biological growth function, nonlinear cost effects, convexity of the objective functional as a function of the control, and character of harvesting (selective vs. nonselective). In turn, such dynamical instabilities are responsible for nonexistence of optimal controls, chattering controls, hysteresis effects, and pulsing.

In addition to all these caveats, we may encounter situations when we would like to have the control set depend on the state variable and thus invalidate the maximum principle altogether.

APPENDIX B - DESCRIPTION OF PROGRAM

B.1 STRUCTURE

The results described in this paper were obtained using the following algorithm:

```
Initialize variables and graphics screen.
For x_varying = x_start to x_end by x_increment.
        For y_varying = y_start to y_end by y_increment.
                Until termination condition is met.
                        Grow until a harvest condition is met.
                        Harvest; calculate profit and reward.
                End until termination condition.
                Graph the total functional.
        End for y_varying.
End x_varying.
Finish graph and wait for user response.
```

The parameters x_varying and y_varying are varying quantities in the 2-D graphs. In the 1-D graphs, x_varying = y_varying and both x and y vary together. What this parameter actually is depends on the situation, but in our cases it relates to the size of the harvest. The start, end, and increment variables are determined before the program is run and identify the region to be studied.

The harvest condition is specified by the user as either after the population has increased to at least a certain size, or when a certain number of iterations have been completed. One harvest includes resetting the population sizes to reflect how much of the population is left and calculating the benefit of the harvest. The conditions to determine these values depend on the harvest strategy. The economic profit for this harvest is then calculated from the given formula and added to a running total for the population being harvested. Similarly, the ecological reward is calculated and added to its running total. The process of growth and harvests is concluded after N harvests; that is the termination condition in these cases.

After the harvesting sequence has been completed, the economic and ecological totals for the separate populations are added for a total economic profit and a total ecological reward for the whole sequence. Then the objective functional is found as the sum of these totals. This final result is then ready to be graphed. (See Appendix B.2.)

The present algorithm is implemented in the C++ program [A1], combined.c, given in Appendix C. Some of the commands, especially those for graphics are C++ specific. The program includes two supplemental files. The first, eqmap.c, contains all the equations that define the problem, such as the growth map, the profit function, and the ecological reward function. The values for the parameters are in the second included file, params.c. The separate file allows easier access and allows a user without much programming experience to make some changes without looking through the code. The program must be compiled after

each change, but it has many options and does not require much user input during a run.

Because this program was set up to allow the user to choose different harvest strategies by setting the values of a few variables in params.c, there are several conditional statements that would not be needed in a program specific to one strategy.

In the program instead of having "x" and "y" populations, we used red and blue, respectively. This allowed us to refer to the coordinate axes as "x" and "y" without confusion.

B.2 DATA DISPLAY

The results of our program are displayed on the screen using the graphic capabilities of C++. The graphics library easily allows us to draw the axes, plot points, and print text for titles and labels.

Two types of graphs are available in this program. The 1-D graph plots the profit for variations in one parameter. This graph can show the economic and ecological profit while also displaying the total. The 2-D graph evaluates the value of the objective functional as a function of two parameters. The value of the functional is classified as either above an upper bound (superior performance), below a lower bound (unacceptable performance), or between the two bounds (acceptable performance). Then the appropriate color for each category is used to plot that point on the screen. We consistently used blue, red, and yellow for the three respective situations. The upper and lower bounds can be modified to fit the situation.

The output from the programs was saved using PC Paintbrush IV Plus [A2]. This program offers a screen capture routine, Frieze, that saves the graphs. Printouts can be made later using the Paintbrush software.

APPENDIX C - PRINTOUT OF PROGRAM

```
/*************************************************************/
/* combined.c                                                */
/*                                                           */
/*     This program simulates the growth of two populations. */
/* These populations grow according to the equations in      */
/* eqmap.c. When one fulfills the harvesting conditions a    */
/* harvest occurs. The harvest is either selective or        */
/* nonselective depending on the settings in params.c. The   */
/* harvest can be done in any of three ways: harvest all     */
/* but a reset amount, harvest a certain amount, or harvest a */
/* percentage of the population. After a number of harvests, */
/* harvest_max, the harvest sequence is terminated.          */
/*     The simulation involves two cycles. The first is the  */
/* growth cycle, and it ends with a harvest. The second is   */
/* the harvesting sequence, and it ends with termination     */
/* after a certain number of harvests. After a harvest the   */
/* economic profit from that population is calculated and    */
/* added to a total. Also the ecological reward is           */
/* calculated in a similar way. At the end of a harvesting   */
/* sequence the totals of the economic profit, ecological    */
/* reward, and their total, the objective functional, can be */
/* graphed. The parameter values are changed and the initial */
/* values reset as a new harvesting sequence begins.         */
/*************************************************************/

#include <graphics.h>
#include <stdio.h>
#include <conio.h>
#include <stdlib.h>
#include <stdarg.h>
#include <math.h>
#define MIN_VAL 1e-299
#define PI 3.141592654
#include "eqmap.c"

int  gprintf(int *xloc, int *yloc, char *fmt, ... );

int MAXY;      //Holds the largest y position for the graphics screen



typedef struct thresh {
        double harv_thresh;          /* Harvesting thresholds                  */
        int harv_time;               /* How many iterations between harvests*/
```

```
        double initial;             /* Initial populations (R0&B0)        */
        double growth_1;            /* Growth rates (a1 & b1)             */
        double growth_2;            /* Interaction terms (a2 & b2)        */
        double reset;               /* Reset values after harvest         */
        double harvsize;            /* Size of harvest to collect         */
        double harvperc;            /* Percent of population to harvest   */
        long delay;                 /* Delay between ordering harvest &   */
                                    /*    actually executing it           */
        double carry;               /* Carrying capacity for this         */
                                    /*    population (K&L)                */
        int ordered_harv;           /* Flag if harvest has been ordered   */
        int harv_count;             /* Counts number of harvests that     */
                                    /*    have been completed             */
        long gets_harv;             /* Holds time pop. is to be harvested */
        long double prev;           /* Population size in past step       */
        long double next;           /* Population size calculated for the */
                                    /*    next step                       */
        double profit;              /* Profit from this harvest           */
        double tot_prof;            /* Total profit during this harvesting */
                                    /*    sequence for this population    */
        double ecology;             /* Ecological reward for this harvest */
        double tot_ecol;            /* Total ecol. reward for this harvest */
                                    /*    sequence for this population    */
           } thold;

/*************************************************************/
/*   initial_graph                                           */
/*                                                           */
/*   Sets the screen for graphics output.                    */
/*************************************************************/


void initial_graph()
{

 int gerror;
 int gdriver = DETECT;
 int gmode;
 int maxx;

   initgraph(&gdriver, &gmode, "C:\\TC\\BGI");
   gerror = graphresult();
   if (gerror < 0)
   {
      printf("Graphics initialization error: ");
      printf("%s \n", grapherrormsg(gerror));
      exit(1);
```

```
    }

    setbkcolor(EGA_LIGHTGRAY);
    maxx = getmaxx();
    MAXY = getmaxy();
    setcolor(EGA_BLUE);
    settextstyle(2,HORIZ_DIR,4);

    /* Outlines the screen and draws the axes  */

    rectangle(0,0,maxx,MAXY);
    line(30,30,30,(MAXY-25));
    line(25,(MAXY-30),(maxx-30),(MAXY-30));

}

/*************************************************************/
/*  graphp                                                   */
/*                                                           */
/*  Graphs the profit as it varies with a parameter. Three   */
/* colors are available for plotting economic profit,        */
/* ecological reward and their total separately. Is also     */
/* used for 2-D graphs where the color represents the size   */
/* of the total objective functional and both axes represent */
/* values of parameters.                                     */
/*************************************************************/

void graphp(double xpt, double ypt, int y_axis,
      double yf, double yoff, int color, int x_axis, int graph)
{
 double xval;
 double yval;
 int maxy2;
 int j =1;

  /* Plots only points above the yaxis in the profit graph */

    if (!((ypt < yoff) && (graph ==1)))
    {
          maxy2 = MAXY-20;

          xval = x_axis;
          yval = y_axis;

  /* Labels the x axis as points are plotted in profit graph */

          if (graph == 1)
```

```
        {
          (x_axis % 60) ? (j=j):
               gprintf(&x_axis,&maxy2,"%2.1e",xpt);
          (x_axis % 30) ? (j=j) :
               line(x_axis,MAXY-25,x_axis,MAXY-30);
          yval = MAXY - 31 - (ypt * yf) + yoff;
        }

        switch (color) {
            case 0 : putpixel(xval,yval,EGA_RED);
                     break;
            case 1 : putpixel(xval,yval,EGA_YELLOW);
                     break;
            case 2 : putpixel(xval,yval,EGA_BLUE);
                     break;
                     }

   } //end if
}

/*************************************************************/
/*  plot_maxy                                                */
/*                                                           */
/*  Labels y-axis for profit plot  & 2-D graph.              */
/*                                                           */
/*************************************************************/

void plot_maxy (double start, double end, double scale_factor)
{

 int x, y;
 double yval, y_scale;
 double interval;
 int yoff;

   settextstyle(2,VERT_DIR,4);

   yoff = start*scale_factor;
   yval = end * scale_factor - yoff;
   interval =  end - start;
   y_scale = 0.2;

//Label the starting point

   x = 21;
   y = MAXY - 40;
   gprintf(&x,&y,"%3.2f",start);
```

```
//Label the rest of the axis

   while (y_scale <= 1)
   {
     y = MAXY - 30 - (yval * y_scale);
     line(25,y,30,y);
     y = y - 13;
     gprintf(&x,&y,"%3.2f",(start + (interval * y_scale)));
     y_scale = y_scale + 0.2;
   }

   settextstyle(2,HORIZ_DIR,4);

}

/*************************************************************/
/*   plot_maxx                                               */
/*                                                           */
/*    This function labels the x-axis for the 2D plot.       */
/*                                                           */
/*************************************************************/

void plot_maxx (double start, double end, double scale_factor,
          int xoff)
{

 int x, y;
 double xval, x_scale;
 double interval;

   xval = end * scale_factor - xoff;
   interval = end - start;
   x_scale = 0.2;

 //label beginning of axis

   y = MAXY-15;
   x = 25;
   gprintf(&x,&y,"%3.2f",start);

 //label the rest of the axis

   while (x_scale <= 1)
   {
     x = 30 + (xval * x_scale);
     line(x,MAXY - 30,x,MAXY - 20);
```

```
      x = x - 7;
      gprintf(&x,&y,"%3.2f",(start + (interval * x_scale)));
      x_scale = x_scale + 0.2;
    }

}

/*************************************************************/
/*   harvest                                                 */
/*                                                           */
/*   This function does the calculations for a harvest of    */
/*  either population                                        */
/*************************************************************/
void harvest (thold *population, double factor, int measure)
{
 double harvested,
       remains;

    switch (measure){
      case 1:  harvested = (*population).prev -
                                    (*population).reset;
                   remains =  (*population).reset;
                   break;
      case 2:  harvested = (*population).harvsize;
                   remains = (*population).prev - harvested;
                   break;
      case 3:  harvested = (*population).prev *
                                    (*population).harvperc;
                   remains = (*population).prev - harvested;
                   break;
        }

    (*population).harv_count++;
    (*population).profit = prof(harvested);
    (*population).tot_prof += (*population).profit;
    (*population).prev = remains;
    (*population).ecology = ecology(remains, factor);
    (*population).tot_ecol += (*population).ecology;
    (*population).ordered_harv = 0;
}


/*************************************************************/
/*   mainline                                                */
/*                                                           */
/*************************************************************/
```

```
main()
{

 thold red, blue;                /* Red and Blue are the two populations */
                                 /* Red = x, Blue = y in other documentation*/
 long time = 0;
 int graph = 0,  done = 0,
    x_axis = 31, y_axis;
 long harvest_max;
 double rvary, bvary;
 double r_reset_start, r_reset_end, r_reset_interval,
        b_reset_start, b_reset_end, b_reset_interval;
 double r_harvsize_start, r_harvsize_end, r_harvsize_interval,
        b_harvsize_start, b_harvsize_end, b_harvsize_interval;
 double r_harvperc_start, r_harvperc_end, r_harvperc_interval,
        b_harvperc_start, b_harvperc_end, b_harvperc_interval;
 double rfactor, bfactor;
 double r_start, b_start,
        r_end, b_end,
        r_interval, b_interval;
 int roff, boff;
 double harv_size;
 double prof_factor, prof_off,
        tot_prof=0, max_profit=0, max_param;
 int x, y;
 double power;
 double prof_upper, prof_lower;
 double tot_ecol, functional, factor;
 int condition, selective, measure;


   #include "params.c";

   /* Determine from user which map to graph */

   printf("1. Profit plot\n");
   printf("2. Two Dimensional Profit Plot\n\n");
   printf("Enter Choice - ");
   scanf("%d",&graph);

   initial_graph();

    /*Initialize parameters */

   y_axis = MAXY - 31;
   switch(measure){
        case 1:  /* reset*/
```

```
            r_start = r_reset_start;
            b_start = b_reset_start;
            r_end = r_reset_end;
            b_end = b_reset_end;
            r_interval = r_reset_interval;
            b_interval = b_reset_interval;
            break;
    case 2: /* harvsize*/
            r_start = r_harvsize_start;
            b_start = b_harvsize_start;
            r_end = r_harvsize_end;
            b_end = b_harvsize_end;
            r_interval = r_harvsize_interval;
            b_interval = b_harvsize_interval;
            break;
    case 3: /* harvperc*/
            r_start = r_harvperc_start;
            b_start = b_harvperc_start;
            r_end = r_harvperc_end;
            b_end = b_harvperc_end;
            r_interval = r_harvperc_interval;
            b_interval = b_harvperc_interval;
            break;
        }

 /* label x and y axis for 2-D graph  */

 if (graph == 2)
 {
      plot_maxy(r_start, r_end, rfactor);
      plot_maxx(b_start, b_end, bfactor, boff);
 }


 bvary = b_start;
 rvary = r_start;
/* Begin the loops that try each value of rvary and bvary */


 while (rvary <= r_end)
 {
     while (bvary <= b_end)
     {
       time = 0;
       blue.ordered_harv = 0; /*Reinitialize values after */
       red.ordered_harv = 0;  /* each harvesting sequence */
       blue.harv_count = 0;
```

```
red.harv_count = 0;
blue.prev = blue.initial;
red.prev = red.initial;
red.tot_ecol = 0;
blue.tot_ecol = 0;
red.tot_prof =0;
blue.tot_prof = 0;

/* Assign all of these because others not used anyway */

blue.harvsize = bvary;
red.harvsize = rvary;
blue.reset = bvary;
red.reset = rvary;
blue.harvperc = bvary;
red.harvperc = rvary;

while (!done)     /* Main Loop - harvesting sequence */
{

 /*  Populations grow */

   eqmap(&blue.next, &red.next, blue.prev, red.prev,
          blue.growth_1, blue.growth_2, red.growth_1,
          red.growth_2, blue.carry, red.carry, power);
   if ((blue.prev > 1e320) || (red.prev > 1e320))
   {
        break;  /* numbers are getting out of bounds */
   }
   blue.prev = blue.next;
   red.prev = red.next;
   time++;

/* Check if a harvest has already been ordered.  If      */
/* not and one is needed order harvest.                  */

   if (!blue.ordered_harv)
   {
        if (((condition == 1) && (blue.prev >=
           blue.harv_thresh))
         ||((condition == 2) && (time % blue.harv_time ==
           0)))
        {
         blue.ordered_harv = 1;
         blue.gets_harv = time + blue.delay;
        }
   }
```

```
        if (!red.ordered_harv)
        {
            if (((condition == 1) && (red.prev >= red.harv_thresh))
             ||((condition == 2) && (time % red.harv_time == 0)))
            {
              red.ordered_harv = 1;
              red.gets_harv = time + red.delay;
            }
        }

  /* Check if harvest has been ordered and if delay has          */
  /* passed. If so, harvest.                                     */

        if ( (blue.ordered_harv) && (time == blue.gets_harv) )
        {
            if (blue.harv_count >= harvest_max)
            {
              break;
            }
            harvest(&blue, factor, measure);
            if (!selective)
            {
              harvest(&red, factor, measure);
            }
        }

        if ( (red.ordered_harv) && (time == red.gets_harv) )
        {
            if (red.harv_count >= harvest_max)
            {
              break;
            }
            harvest(&red, factor, measure);
            if (!selective)
            {
              harvest(&blue, factor, measure);
            }
        }

      } /* EndWhile (Main Loop) - Termination Reached - */

/* Calculate grand totals */

      tot_prof = red.tot_prof + blue.tot_prof;
      tot_ecol = red.tot_ecol + blue.tot_ecol;
      functional = tot_prof + tot_ecol;
```

```
/* Graph appropriate results */

        switch(graph) {
        case 1 :   /* Profit graph  */
                graphp(bvary, tot_prof, y_axis, prof_factor,
                     prof_off*prof_factor, 0, x_axis, graph);
                graphp(bvary, tot_ecol, y_axis, prof_factor,
                     prof_off*prof_factor, 1, x_axis, graph);
                graphp(bvary, functional, y_axis, prof_factor,
                     prof_off* prof_factor, 2, x_axis, graph);
                break;
        case 2 :  /*  Two Dimensional Graph  */
                if (tot_prof >= prof_upper)
                  graphp(bvary, rvary, y_axis, rfactor, roff, 2,
                            x_axis, graph);
                else
                {
                  if (tot_prof <= prof_lower)
                       graphp(bvary, rvary, y_axis, rfactor, roff, 0,
                                x_axis, graph);
                  else
                       graphp(bvary, rvary, y_axis, rfactor, roff, 1,
                                x_axis, graph);
                }
                break;
        } /*end switch*/

/* determines maximum of functional so it can be used in          */
/* labeling the y axis, and printed as the maximum.               */

        if (functional > max_profit)
        {
          max_profit = functional;
          max_param = bvary;
        }

        x_axis++;
        bvary = bvary + b_interval;

  } /* EndWhile (Blue Loop) */

  rvary = rvary + r_interval;
  bvary = b_start;
  x_axis = 31;
  y_axis--;
```

```
    } /* EndWhile (Red Loop) */

    /*prints out the maximum profit and where it is found */

    if (graph == 1)
    {
      x=275;
      y=20;
      plot_maxy (prof_off, max_profit, prof_factor);
      outtextxy(150,20,"Maximum profit =");
      gprintf(&x, &y, "%1.6g", max_profit);
      outtextxy(350,20,"at");
      x= 380;
      gprintf(&x,&y, "%1.4g", max_param);
    }

  /*indications that the program has finished */

    outtextxy(7,5,"(Press any key ...)");
    putchar(7);                          //Beeps when done
    getch();
    getch();

    closegraph();
    return (0);

}

/*************************************************************/
/* GPRINTF: Used like PRINTF except the output is sent to the     */
/* screen in graphics mode at the specified co-ordinate.          */
/*                                                                 */
/*************************************************************/

int gprintf( int *xloc, int *yloc, char *fmt, ... )
{
 va_list  argptr;                    /* Argument list pointer          */
 char str[140];                      /* Buffer to build string into    */
 int cnt;                            /* Result of SPRINTF for return   */

    va_start( argptr, fmt );         /* Initialize va_ functions       */
    cnt = vsprintf( str, fmt, argptr );  /* prints string to buffer    */
    outtextxy( *xloc - 7, *yloc, str );  /* Send string in graphics mode */
    va_end( argptr );                /* Close va_ functions            */
    return( cnt );                   /* Return the conversion count    */

}
```

```
//params.c for combined.c

    harvest_max = 40;                   //termination condition

    red.growth_1 = 1;                   //intrinsic growth rates
    blue.growth_1 = 0.95;

    blue.initial = 200.0;               //initial population sizes
    red.initial = 200.0;

    blue.delay = 5;                     //how long to delay between reaching
    red.delay = 5;                      //  harvest condition and actually harvesting.

    blue.carry = 20000;                 //carrying capacities
    red.carry = 20000;

    power = 1./2.;                      //power in the self repression terms

    prof_factor = 0.002;                //Scaling factors for 1D graph
    prof_off = 0;

    blue.growth_2 = 0.0;                //Interaction coefficients
    red.growth_2 = 0.0;

    factor = 0.12;                      //factor determining effect of ecology
                                        //  compared to economy

    prof_upper = 125000;                //limits for categories in 2D graph
    prof_lower = 100000;


/*  Choose the harvest strategy and give the appropriate information */

/*  Condition for harvest       */
/*  1  = threshold              */
/*  2  = # iterations           */

    condition = 1;

    if (condition == 1)
    {
      blue.harv_thresh = 10000;
      red.harv_thresh = 10000;
    }
```

```
    if (condition == 2)
    {
      blue.harv_time = 10;
      red.harv_time = 10;
    }

/* Selective harvest?          */
/* 0 = nonselective            */
/* 1 = selective               */

    selective = 0;

/* Measure of harvest          */
/* 1 = reset                   */
/* 2 = harvest size            */
/* 3 = harvest percent         */

    measure = 3;

    if (measure == 1)
    {
      r_reset_start = 200;
      r_reset_end  = 6000;
      r_reset_interval = 15;

      b_reset_start = 200;
      b_reset_end = 6000;
      b_reset_interval = 10;

      rfactor = 1.0 / r_reset_interval;
      roff = r_reset_start * rfactor;
      bfactor = 1.0 / b_reset_interval;
      boff = b_reset_start * rfactor;
    }

    if (measure == 2)
    {
      r_harvsize_start = 2000;
      r_harvsize_end  = 10000;
      r_harvsize_interval = 20;

      b_harvsize_start = 2000;
      b_harvsize_end = 10000;
      b_harvsize_interval = 20;

      rfactor = 1.0 / r_harvsize_interval;
      roff = r_harvsize_start * rfactor;
```

```
      bfactor = 1.0 / b_harvsize_interval;
      boff = b_harvsize_start * rfactor;
    }

    if (measure == 3)
    {
      r_harvperc_start = 0.5;
      r_harvperc_end  = 0.99;
      r_harvperc_interval = 0.00125;

      b_harvperc_start = 0.5;
      b_harvperc_end = 0.99;
      b_harvperc_interval = 0.001;

      rfactor = 1.0 / r_harvperc_interval;
      roff = r_harvperc_start * rfactor;
      bfactor = 1.0 / b_harvperc_interval;
      boff = b_harvperc_start * bfactor;
    }


//eqmap.c for combined.c

/*************************************************************/
/* sgn                                                       */
/*    Returns -1 if parameter is negative and +1 if parameter */
/* is positive. Allows use of exponents less than one        */
/* without the program blowing up when the population passes */
/* the carrying capacity.                                    */
/*************************************************************/

int sgn (double number)
{
 int sign;

    if (number > 0)      sign = 1;
    else
    {
     if (number < 0)    sign = -1;
     else               sign = 0;
    }
    return sign;
}

/*************************************************************/
/* eqmap                                                     */
/*    The evolution equations for the populations. In        */
```

```
/*  documentation these equations have x = red, y = blue,          */
/*  a1 = r1_grow, a2 = r2_grow, b1 = b1_grow, b2 = b2_grow.        */
/*************************************************************/

void eqmap(long double *b_next, long double *r_next,
          long double b_prev, long double r_prev, double b1_grow,
          double b2_grow, double r1_grow, double r2_grow,
          double b_carry, double r_carry, double power)
{
 int b_sign, r_sign;

    b_sign = sgn(1-b_prev/b_carry);
    r_sign = sgn(1-r_prev/r_carry);

    *b_next = b_prev          + b_sign*b1_grow*b_prev*
                        pow(fabs(1-b_prev/b_carry),power)
                        + b_sign*b2_grow*(r_prev*b_prev)*
                        pow(fabs(1-b_prev/b_carry), power);
    if (*b_next < MIN_VAL)
          *b_next = 0.0;

    *r_next = r_prev  +r_sign*r1_grow*r_prev*
                        pow(fabs(1-r_prev/r_carry), power)
                        + r_sign*r2_grow*(b_prev*r_prev)*
                        pow(fabs(1-r_prev/r_carry), power);
    if (*r_next < MIN_VAL)
          *r_next = 0.0;
}

/*************************************************************/
/*  prof                                                     */
/*     The equation for the profit function.                 */
/*************************************************************/
double prof(double harv_size)
{
 double profit;

        profit = harv_size*(exp(-pow((2000/harv_size),4))*
                (.1 + pow((1778/harv_size), 4)));

    return profit;
}

/*************************************************************/
/*  ecology                                                  */
/*     Function to calculate a reward for the amount of the  */
/*  population not harvested.                                */
```

```
/*************************************************************/

double ecology(double remains, double factor)
{
 double reward;

   reward = factor*remains*((remains/1000 -1)
                        /pow((remains/1000),2) + 0.01);
   return reward;
}
```

APPENDICES REFERENCES

A1. Turbo C++: Borland International, Inc. 1800 Green Hill Road, P.O. Box 660001, Scotts Valley, CA 95067-0001, 1991.

A2. PC Paintbrush IV Plus: ZSoft Corporation, 450 Franklin Road, Suite 100, Marietta, GA 30067, 1990.

INTERNAL DISTRIBUTION

1. B. R. Appleton
2. Y. Y. Azmy
3. T. J. Burns
4. V. H. Dale
5. D. L. DeAngelis
6. C. M. Haaland
7. D. T. Ingersoll
8. J. O. Johnson
9. S. M. Lenhart
10. H. S. Payne

11-15. V. Protopopescu
16-20. R. T. Santoro
21. D. S. Shriner
22. R. C. Ward
23. EPMD Reports Office
24-25. Laboratory Records Department
26. Laboratory Records, ORNL-RC
27. Document Reference Section
28. Central Research Library
29. ORNL Patent Section

EXTERNAL DISTRIBUTION

30. Prof. R. W. Brockett, Harvard University, Pierce Hall, 29 Oxford Street, Cambridge, Massachusetts 02138

31. Prof. J. J. Dorning, Department of Nuclear Engineering and Engineering Physics Reactor Facility, University of Virginia, Charlottesville, Virginia 22903

32-36. S. J. Gaff, 5883 Francis St., Kalamazoo, Michigan 49001

37. Dr. J. E. Leiss, Route 2, Box 142C, Broadway, Virginia 22815

38. Prof. N. Moray, University of Illinois, Department of Mechanical and Industrial Engineering, 1206 West Green Street, Urbana, Illinois 61801

39. Prof. M. F. Wheeler, Department of Mathematics, Rice University, P.O. Box 1892, Houston, Texas 77251

40. Office of the Assistant Manager for Energy Research and Development, Department of Energy, Oak Ridge Operations, P.O. Box 2001, Oak Ridge, Tennessee 37831

41-50. Office of Scientific and Technical Information, P.O. Box 62, Oak Ridge, Tennessee 37830