Laser Isotope Separation Program

1980 Annual Reports (Secret & Unclass.)

Lawrence Livermore National Laboratory

OPTIMAL RESOURCE ALLOCATION AND DECISION ANALYSIS

Author

E.B. Rockower

Introduction. Any extensive R&D program requires numerous decisions as the program evolves. In our atomic-vapor laser isotope-separation (AVLIS) program, decisions span issues associated with process selection, systems design, operating and funding schedules, and the setting of performance goals and milestones. Usually we must choose among several processes and systems, but we would like to maintain alternatives until we are assured success in each required category. For example, we are considering a number of alternative pump-laser systems for the AVLIS process. If resources of money, manpower, and time were unlimited we would develop all available alternatives. Construction of the full-scale laser isotope-separation (LIS) plant would be based on the best alternatives available at the time of design freeze. In practice we must allocate limited resources (typically money) among a few competing alternatives. Thus as the program progresses we must eliminate alternatives as soon as it becomes clear that their contribution to the expected final payoff no longer justifies the resources they consume.

We have developed two computer codes for use as decision tools that systematize the timing of decision points, the setting of funding levels, and the dropping of alternative technologies. These codes aid the decision maker in formulating strategies that are consistent both with his preferences regarding outcomes and with his subjective judgments concerning the probabilities of unknown factors or future events.

The simpler model, the static nonlinear programming (NLP) code, makes the assumption that there is only one decision point, at which time we optimally allocate a fixed total budget among the various alternative technologies in each LIS program. We maximize the overall probability of success, defined as the successful completion of at least one of the LIS projects, using the numerical-optimization techniques of nonlinear programming. Our second code, the dynamic-resource allocation code SAGE (*S*equential *A*llocation *GE*nerator), is more complex and makes use of the methods of statistical decision theory. In SAGE we allow for up to five decision points at which we reallocate budgets and where we may drop alternate technologies or LIS programs to maximize the expected (average) payoff.

Such tools of decision analysis do not actually make decisions. Only someone cognizant of all quantifiable and qualitative factors in the project, and who is also maintaining the perspective of the project context, can make wise decisions. Rather, such tools provide a language and framework for clarifying issues, focusing discussion, and developing intuition regarding sensitivities to important parameters. Using these tools we can examine the consequences of various assumptions, scenarios, and possible alternate strategies.

One-Time Allocation Problem. Our NLP code makes the simplification that there is only one decision point, at which time we optimally distribute a fixed total budget among alternative competing laser drivers (D) and uranium-handling (U) technologies to maximize the probability of eventually achieving scientific, engineering, and economic success for a LIS process. Figure 3-16 illustrates an example of such a situation. Our code can also optimally allocate a total DOE/LIS budget among a number of competing LIS programs. As presently implemented, each of three LIS programs has four drivers and four uranium-handling technologies. Thus the allocation of a fixed overall DOE budget for LIS constitutes a 24-dimensional optimization problem constrained by the prior expenditures for each technology in each program.

Both of our decision codes use a two-parameter functional relationship for the dependence of the probability of eventual success of the *i*th technology [e.g., copper-vapor laser (CVL), rare-gas halide (RGH), and frequency-doubled neodymium YAG (FDNY) laser technologies] on the total dollars spent (D_i) on R&D for that technology,

$$P_i = P\phi_i \left[1 - \exp - \left(\frac{D_i}{SAT_i} \right) \right] . \qquad (23)$$

In this expression SAT_i is the level of funding that would begin to saturate the development of the *i*th technology. The other parameter $P\phi_i$ is the probability of achieving success if we had all the money we reasonably needed for development of the particular technology.

The various laser and uranium-handling technologies we investigated comprise a series/parallel circuit where the set of all laser technologies is in series with all uranium-handling technologies (both are necessary). However, CVL, RGH, and FDNY laser technologies are in parallel with each other (success in any one is sufficient). The logic of this type of circuit, shown in Fig. 3-16, also applies to each of the competing LIS programs. Additionally we assume that the three LIS programs are in a parallel development mode. In other words success in any one of the programs would constitute success (although with varying levels of benefit) of the overall LIS program.

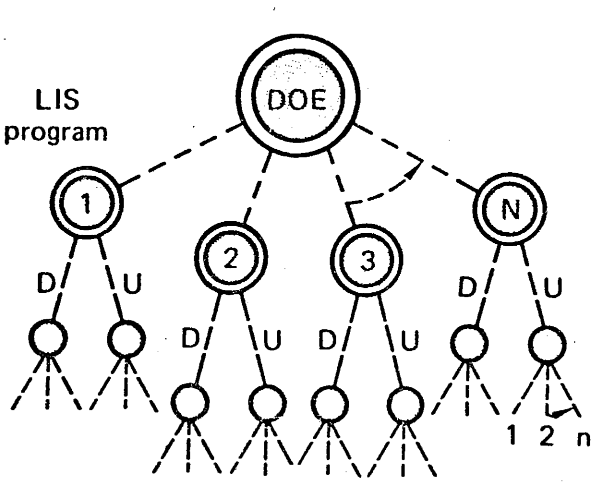

Fig. 3-16. LIS program structure.

Using the functional form of Eq. (23) for the dependence on funding of the probability of successful development of each technology, we can write the expression for the dependence of the overall probability of success P on the funding of each alternate technology D_i. Describing the funding allocation as a vector, we express our problem as:

maximize P (D)
subject to the constraints
(a) $D_i \geq 0$ and (b) $\sum D_i$ = budget.

The NLP code performs this optimization using a modified Fletcher–Reeves conjugate-gradient technique using penalty functions.[5] The code output gives the budgetary allocation to each of the alternative technologies, along with the probability of success for each program. We can thus exercise the model using the latest estimates of the input parameters and the expenditures to date and find the relative proportion of the next year's funding that should be spent on R&D in each technology area. This also identifies those technologies (given the finite remaining budget) that should not be funded any further.

We have exercised the NLP code using preliminary data representing rough estimates of $P\phi_i$ and SAT_i for sets of technologies in each of the three competing LIS programs separately. The results provide an initial estimate of the optimal resource allocation in each LIS program, subject to a given program budget. We also found a corresponding optimal probability of success for each project and for the LIS program as a whole. Using the same input data, but allowing the NLP code to allocate the total budget among the three programs, we found a significant improvement. Sensitivity studies varying the total budget provide the optimal number and identity of technologies in a program (and the optimal number of programs) at each level of funding. As we reduce the funding the code begins to allocate zero resources to alternates that previously had been funded.

Dynamic Resource-Allocation Methodology. The methodology we have developed for the sequential allocation of R&D dollars is an application and extension of the techniques of statistical decision theory. Certain concepts, problem structure, and calculations occur in any statistical decision-theory application.[6] Certain other problem elements are peculiar to the LIS R&D application where a number of alternate programs or technologies are developed in parallel. To aid in the exposition of possibly unfamiliar decision-theory principles, we will begin with a brief description of a simplified problem, which we analyze within the framework of statistical decision theory.

Statistical-Decision-Theory Example. Our example consists of the R&D of two pump lasers, laser 1 (L1) and laser 2 (L2), e.g., any two of FDNY, CVL, or RGH. Although we require only one pump-laser system for our development of an AVLIS process, we are not 100% certain at the outset that either laser system can be adequately developed for our needs. Therefore we hedge our bets by developing in parallel two laser systems, with the expectation that we have significantly increased our chances of success with at least one of the alternates. After a period of parallel development we must decide (presumably based on the outcome of the development effort) which laser to drop from consideration, and which to develop to (hopefully) eventual success. For simplicity we assume that the outcome of the initial development stage (in addition to an enhanced state of development of each technology) is either a positive or negative indication for each of the laser types (e.g., L1 looks good, L2 looks bad, etc.).

We can depict the sequence of development results and decisions in a decision tree as shown in Fig. 3-17. Two types of nodes occur on the decision tree: probability (P) nodes and decision (D) nodes. These alternate along any path. Probability nodes represent the outcome of some chance event (in this example the result of development). We have no control over which branch to the right is taken, but we do have certain probabilistic knowledge regarding the likelihood of each alternative. At decision nodes we do have complete control over which

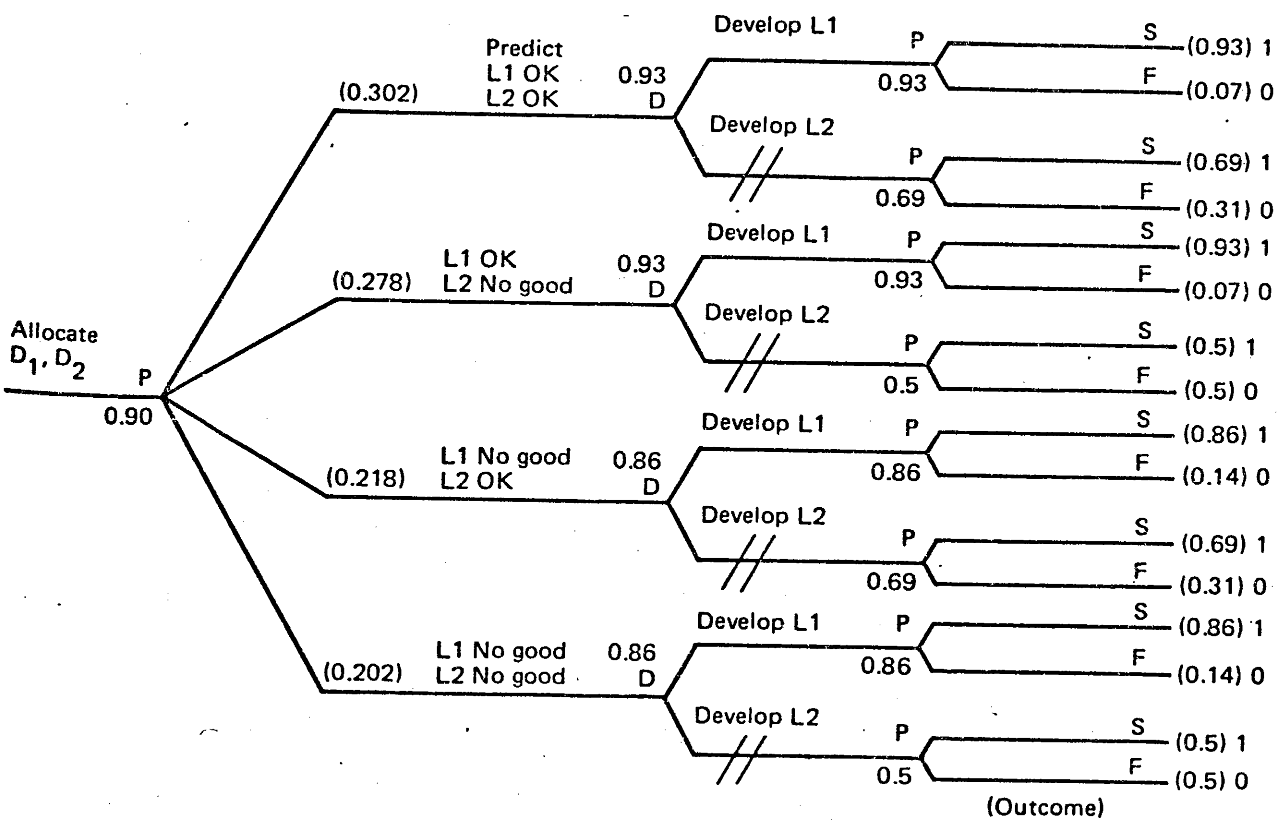

Fig. 3-17. Decision tree for L1 and L2 laser development.

path to take. In fact, it is assumed that we have some decision algorithm with which we make this choice. Our algorithm, of course, is a function of the particular path we have traversed to reach each D-node. The final P-node in our decision tree leads to either success or failure of the particular technology, with its associated benefit or penalty. In this example we have chosen all benefits equal to one and all penalties equal to zero. Since we will be maximizing the expected payoff, this choice corresponds to maximizing the probability of success. Note that even for this simple problem, with only one development phase and one decision to be made, the tree already has 29 branches representing all possible alternative sequences of decisions and outcomes.

The procedure for analyzing a problem using statistical decision theory consists of the following four steps.[6]

1. Set up the decision tree from left to right, defining the probability and decision nodes, hence specifying the logic of the program activities and the decision sequence.

2. Assign the payoffs at the terminal nodes, thus determining the figure of merit to be optimized.

3. Calculate the probabilities at the chance forks (P-nodes) from left to right. This consists of a Bayesian analysis,[6] for each possible branch, of the probabilities of success of each laser system that updates these probabilities to reflect each of the possible results of the preceding development stage. We also calculate the (predictive) probability that each of these branches will in fact be realized. These predictive probabilities appear in parentheses in Fig. 3-17.

4. Average out and fold back. Working from right to left we calculate the average payoff at each probability node as it is encountered. At each decision node we select the branch that leads to the P-node with the greatest expected payoff. We associate this value of expected payoff with the D-node. This branch then represents the decision made, e.g., drop laser L2. This constitutes the optimal use of our limited control at each decision point. Continuing from right to left we construct the complete set of optimal decisions.

This analysis yields the expected payoff written at the left-most node. Note that in our example this is the same as the probability of success. In Fig. 3-17 the numbers written in parentheses on each branch emanating from a P-node are the probabilities that the respective path will be traversed. The other number written at each node is the expected payoff corresponding to that position on the tree. We have crossed out the unfavored options from each D-node.

Behind our analysis of this decision tree lie calculations that require specification of a "likelihood function" representing the reliability of R&D information. An example of this function is the conditional probability P_r (predict L1 OK$|$L1 will be successful), which we use in the Bayesian probability analysis. For our example we have arbitrarily chosen this probability as 0.6 for both technologies. Later we will need to specify a model that determines this probability as a function of the level of R&D funding for each technology. Clearly, more dollars spent should yield more reliable R&D information. Other inputs required for this simple example include the *a priori* probabilities of success for each laser technology, here chosen to be 0.9 for L1 and 0.6 for L2. The right-most parentheses in Fig. 3-17 enclose the corresponding posterior (relative to the development stage) probability for each laser system.

Our analysis in Fig. 3-17 shows that whatever the outcome of the remaining development prior to a final decision, we will eventually terminate development of laser L2. This conclusion is a consequence of the relative weakness of our reliability-of-research factor 0.6 (0.5 corresponds to no usable information). We conclude that L2 should be dropped immediately, since its continuation adds nothing to the probability of success (0.9). The other choice is to provide enough additional funds during the current development stage so that the reliability-of-research factor becomes great enough to influence the final decision. This would definitely increase the probability of success.

Specific Model Assumptions. The statistical-decision-theory model just described treats only single decisions, i.e., which laser program to drop. The size of the tree increases almost exponentially with the number of decisions required. We also wish to decide how much money should be allocated to each technology at each research stage. If we were to represent the continuum of possible allocations, even divided into fairly large increments by alternate branches from the decision nodes, the problem would quickly become intractable. To deal with this problem we have derived a number of alternative budget-allocation algorithms. We use these algorithms to distribute at each D-node a fixed budget for each R&D stage among the surviving alternate programs or technologies. Depending on the option selected the budget at each stage is allocated proportional to:

- The initial allocations input for the first R&D stage.

- $$\frac{P\phi_i}{SAT_i} \exp\left[-\frac{D_i(\text{prior})}{SAT_i}\right]. \tag{24}$$

- The gradient of the probability of overall success with respect to the dollars allocated.

- The gradient of the expected payoff with respect to the dollars allocated.

The third and fourth options locally maximize the increase in probability of eventual success and expected payoff, respectively, as they are influenced by the dollars allocated. We can also include additional allocation algorithms.

The money spent on R&D gives rise to two separate types of gain. One type has more of the "research" character. Analogous to a market survey or statistical sampling, it garners more precise information to aid in better forecasts and decision making. The other type of gain has more of the "development" character, in which the program or technology is brought to a more mature and complete level of development. This would include construction of required facilities, attainment of milestones, defining a preferred process, and selection of operational options and procedures, etc.

Our methodology reflects the research gain by using the likelihood function. We apply this function in the Bayesian analysis for the calculation of both posterior probabilities of success and the probabilities of various research outcomes. The likelihood function, which describes the reliability R of the information derived from R&D during each stage, is a function of the dollar allocation to each technology during that stage. For each technology if one expected a positive or negative indication with a probability of 0.5 (irrespective of whether success is in fact eventually achieved), then clearly that would constitute no information, i.e.,

$$P_r\left(\begin{array}{c}\text{Positive}\\ \text{indication}\end{array}\middle|\begin{array}{c}\text{Eventual}\\ \text{success}\end{array}\right) = 0.5 . \tag{25}$$

Hence this should be the value of the likelihood function when no money is allocated to a technology, i.e., the result is no new information. The equation we have used to represent the dependence on dollar allocation is:

$$R(D_i) = 1 - \frac{\exp(-\beta D_i/SAT_i)}{2} . \tag{26}$$

We choose the parameter β (generally greater than or equal to 1) to rescale the saturation dollars. Large values of β correspond to very efficient research, i.e., research that generates reliable predictions.

We now briefly discuss the manner in which we treat the development aspect of R&D. Note that the *a priori* probabilities of success (and their subsequent Bayesian updates depending on results of prior R&D) are based on anticipated total expenditures. These have the same functional form as in the nonlinear programming model. Programmatic decisions at the D-nodes (dropping programs and budget reallocation) may result in greater (lesser) total expenditures and a consequent greater (lesser) chance of eventual success. Hence we update the probabilities of success for each program or technology both at each P-node (because of the current indications of the most recent research stage) and at each D-node (following the latest reallocation of funds).

Finally we specify the logical sequence of activities and decisions that we wish to analyze. In starting our analysis after an arbitrary amount of funds have already been expended, we begin with the next research stage and an initial allocation for three programs or technologies being developed in parallel. (Below we extend this analysis to the case in which there are up to four programs or technologies with an immediate decision at hand to drop one.)

Figure 3-18 shows the sequence of probability and decision nodes representing the results of research and reallocations of budgets, respectively. Note that only on every other D-node is one program or technology dropped. This is because no matter how negative the latest research results, a program or technology is generally phased down gradually, rather than abruptly terminated. This allows for the possibility that it may later be ramped up as a result of significant innovations during the

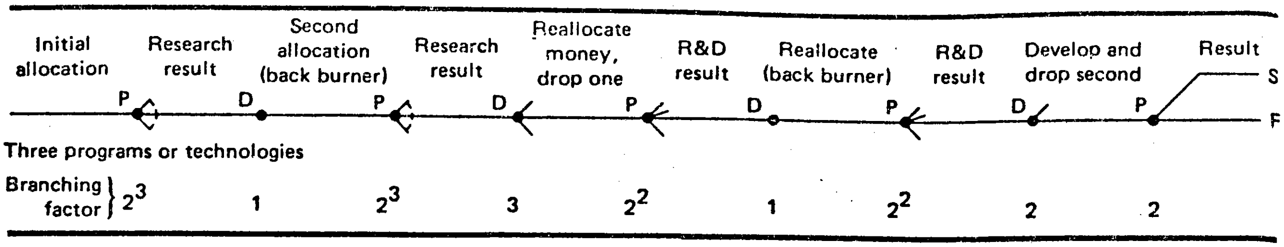

Fig. 3-18. Decision tree for three programs or technologies.

"back-burner" stage, along with possibly diminished expectations for the alternate programs. We indicate below each node the multiplicity of branching. This decision tree, if fully drawn, would have over 23,000 branches.

Description of SAGE Code. We implemented the methodology described above in a computer code SAGE for the CRAY computer. The code handles as many as three (four) programs or alternate technologies if the first branch point is a probability node (decision node at which one program or technology is dropped).

If the case being run requires the first node to be a decision node, at which one of the four programs is dropped, the code provides the option of some operator interaction. This is because the selection of the program or technology to be dropped first may correspond to an actual program decision at hand. The code provides the decision maker with the implications of each of the four options corresponding to dropping each of the four programs or technologies. In addition, although the SAGE code makes decisions based on maximizing expected payoff, other considerations and figures of merit may be of interest. The code first presents the decision maker with the expected payoff, standard deviation of payoff, and probability of success for each of the four options. For a number of reasons he may choose to drop technology P4, even though dropping technology P3 leads to a slightly higher expected payoff. The decision to drop technology P4 may lead to a substantially lower standard deviation of payoff or it may result in a higher probability of success (as is true in our example). Either of these decisions could be preferable to an insignificant increase in expected benefit. Political considerations could also easily outweigh a small advantage in expected payoff. By knowing the consequences of each of the alternatives, he is free to incorporate any qualitative considerations he feels are important.

After the decision maker chooses a decision option (in our example, dropping P3) the code prints out corresponding full numerical output data at the terminal and creates the graphical output file. The terminal output includes the expected value (7.86 arbitrary units) and standard deviation of the payoff as well as the probability of success that had been printed previously. In addition the output shows two useful measures of our R&D sequence, i.e., the expected value of perfect information (EVPI) and the expected value of sample information (EVSI). Lastly the output indicates that if a final decision were forced now, technology P1 would be selected, yielding an expected payoff of only 7.5911, even though all RD&D money is assumed allocated to P1.

In the case referred to here EVSI is positive. However, it is possible for it to become less than zero, and in fact this has occurred in some example cases we have run. A negative EVSI indicates that the R&D effort directed towards three parallel programs or technologies has less value than immediately reducing the program to one candidate. Such a result indicates that a selection decision was overdue. The value of EVPI printed out is a measure of the benefit one might realize from more precise forecasts of success and failure.

The detailed probability data printed at the terminal relate both to the probabilities of success and failure and to the statistics regarding the various decisions made at the decision nodes. The probability of survival and success for each program or alternate technology is equal to the product of the conditional probability of success given survival to the final stage times the probability of survival to the final stage (in our example, stage 4). From changes in the probability of survival to each stage we calculate the probability that each program or technology is dropped at each stage. These probabilities all appear in the output. The code computes these probabilities from the decision-tree data by summing the path probabilities corresponding to the appropriate event, such as survival of a specific technology to stage n.

The graphical output consists of the following for the selected option:

- Bar charts of the probability of being dropped for each program or technology at each stage.
- Bar chart of the probability of success for each program or technology given that it survives to the final stage, with the *a priori* probability of success presented for comparison.
- Bar chart of the probability of surviving to the final stage.
- The cumulative distribution of payoff.
- A bar chart displaying the contribution of each program or technology to the expected payoff.
- A histogram showing the distribution of expenditures for each program.

When the first node is a decision node the output also contains a bar chart showing the expected payoff and the probability of success for each option (corresponding to dropping each of the four programs at the first decision node). Figure 3-19 shows examples of these graphics.

We observe from Fig. 3-19 (a) and (c) that in this example alternate technology P2 contributes nothing to the optimal expected payoff of 7.86 (arbitrary) units. Figure 3-19 (d) confirms this by showing that only when one drops P4 or P1 is the

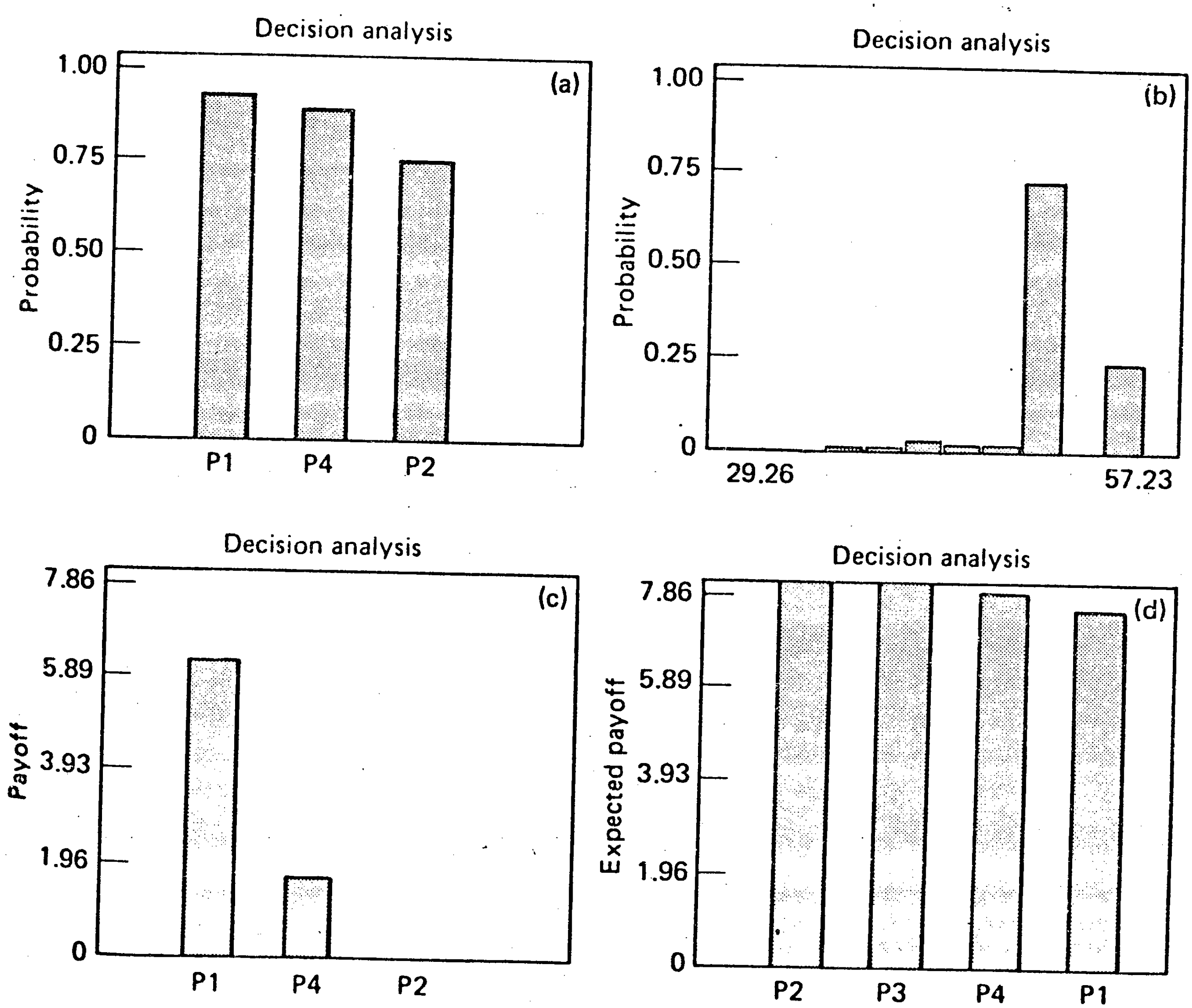

Fig. 3-19. SAGE graphical output. (a) Probability of success given survival; (b) expenditure distribution for P1; (c) contribution to expected payoff; (d) dropped program.

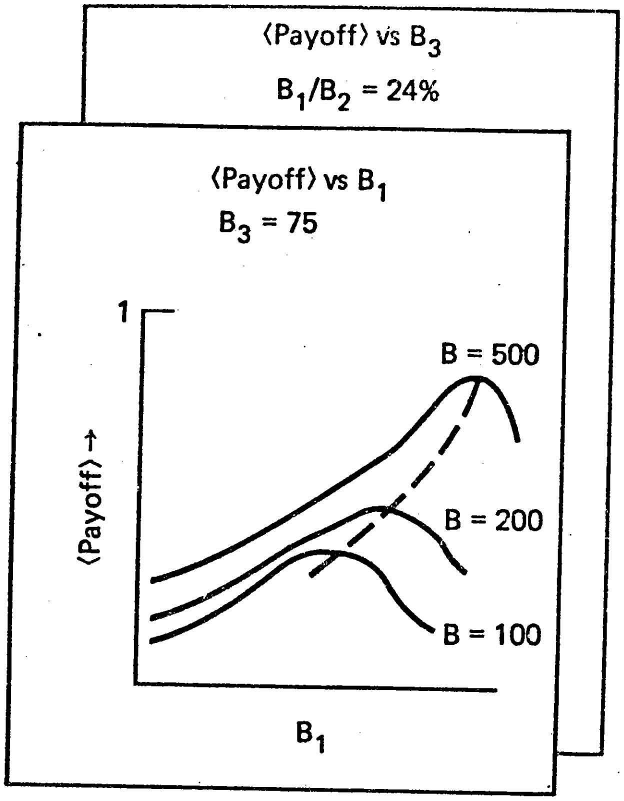

Fig. 3-20. Sensitivity studies, timing of decisions.

expected payoff significantly reduced. We should therefore consider dropping P2 and P3 immediately. Putting all the R&D funding into a development program comprising P1 and P4 only will result in an expected payoff greater than the 7.86 we obtained here since no funds are wasted on an option that had virtually no chance of final selection. By trying different combinations of program alternatives and parameter values we have gained numerous other insights that both confirm and extend our intuition regarding various possible tradeoffs. In particular, sensitivity studies with respect to magnitudes of the budget for each stage, keeping a fixed mission budget as in Fig. 3-20, can indicate near-optimal points in time at which to drop redundant alternatives.

We are now generating the data base for further validation and sensitivity studies. We will use the codes in the direct manner described above to derive the consequences of our assumptions regarding programmatic parameters and to determine optimal decision sequences and times. We will also use the codes to quantify the implications for programmatic variables of any postulated decision sequence and assumed probability of success. By using the code in this inverted mode we can help evaluate any proposed program-decision logic.

References

5. D.G. Luenberger, *Introduction to Linear and Nonlinear Programming* (Addison-Wesley Publishing Co., Inc., Reading, MA, 1973).
6. H. Raiffa, *Decision Analysis—Introductory Lectures on Choices Under Uncertainty* (Addison-Wesley Publishing Co., Inc., Reading, MA, 1968).