COWLES FOUNDATION FOR RESEARCH IN ECONOMICS

AT YALE UNIVERSITY

Box 2125, Yale Station
New Haven, Connecticut

COWLES FOUNDATION DISCUSSION PAPER NO. 243

QUIZ SHOW PROBLEMS

Joseph B. Kadane

February 8, 1968

QUIZ SHOW PROBLEMS*

by

Joseph B. Kadane

1. Introduction

A quiz show contestant may choose the category of his next question. Associated with each category a is a probability p_a of knowing the right answer to the question. If he answers the question correctly the contestant will be given a reward x_a and be required to choose a category not previously chosen. If he answers incorrectly, he will receive the consolation prize y_a and will leave the game with y_a plus his previous earnings. Suppose also that entering category a will require time t_a to recover and be ready to choose another question. Knowing a discount rate $\beta \geq 0$, and the parameters p_a, x_a, y_a and t_a, how should the contestant maximize his expected discounted winnings?

This question divides into two connected parts: Given r categories it has been decided to attempt, what is the optimal order in which to attempt them? Second, if there are n possible categories $(n \leq \infty)$ of which the contestant may choose r, which are the optimal categories to choose?

*In preparing this paper, the author benefited from discussions with Dr. Daniel Levine of the Center for Naval Analyses and Professor Herman Chernoff, L. J. Savage and Harvey Wagner. Mrs. Joan Odland Coker did helpful numerical computations during the summer of 1963.

This study was begun while the author was employed by the Center for Naval Analyses, Office of the Chief of Naval Operations. Its completion was supported by contracts Nonr 225(52) at Stanford and Nonr 3055(01) at Yale University. Reproduction in whole or part is permitted for any purpose of the United States Government. It does not necessarily represent the views of the U.S. Navy, nor of the Center for Naval Analyses.

Intuitively, the contestant seeks an appropriate balance between immediate rewards (x_a, y_a) and future rewards made less likely by the probability of failure $(1 - p_a)$ and less profitable by discounting $(e^{-\beta t_a})$.

At first, one might suppose that the ordering problem includes the choice problem because one could find the optimal order for all n categories and then choose the first r in that order. Consider, however, a situation in which there are several easy and not very valuable categories, and one difficult and valuable one. It should make sense to do the easy ones first and then try the hard one. However, if all but one can be tried perhaps it is optimal to eliminate one of the easy and less valuable categories. Thus further reflection leads one to suspect that the solutions to the choice problem and the order problem need not be the same, and this is indeed so.

By a strategy is meant a sequence of categories, finite or infinite, without repetitions, indicating which category is to be tried first, which second provided the contestant was successful on the first question, etc. Define the expected value of the category a as

$$V(a) = p_a x_a + (1 - p_a) y_a \qquad (V(a) > 0)$$

and the delay factor of the category a as

$$S(a) = p_a e^{-\beta t_a} \qquad (0 < S(a) < 1)$$

If b is a finite strategy and c is a strategy disjoint from b, then bc is a strategy. The functions V and S can be extended to strategies by the relations

$$V(bc) = V(b) + S(b)V(c) \quad \text{and}$$
(1)
$$S(bc) = S(b)S(c) .$$

This functional equation is basic to the discussions and proofs below. The theorems apply to any problem satisfying (1). At least two special cases of this model have been discussed by others: the gold-mining problem and the obstacle-course problem. A third problem, discrete search, is closely related but is not a quiz show problem.

In the gold-mining problem, a man possesses h gold mines and a delicate gold-mining machine which functions with probability $p_{j,k}$ if assigned to the k^{th} mine for the j^{th} time. If it does function, it will process a fraction $r_{j,k}$ of the amount of gold in mine k after $j-1$ excavations. If it does not function, the machine is broken and cannot be repaired. How should this man decide to which mine to assign the machine in each period if he wishes to maximize the expected amount of gold he will mine?

Let the j^{th} excavation of mine k correspond to a category. Then $p_{j,k}$ is the probability of success at the category (j,k), $x_{jk} = \prod_{j'<j} (1 - r_{j',k})r_{j,k}g_k$, where g_k is the amount of gold originally in mine k. Also the consolation prize $y_{j,k} = 0$

and the discount factor $\beta = 0$. In order to prevent making bad choices today to permit lucrative ones to become available in the future, two regularity conditions are imposed:

(i) the amount of gold mined does not increase:

$$r_{j,k} \geq (1 - r_{j,k})r_{j+1,k} \quad \text{for all} \quad j \geq 1 , \quad \text{and} \quad 1 \leq k \leq h .$$

(ii) the probability of success does not increase:

$$p_{j,k} \geq p_{j+1,k} \quad \text{for all} \quad j \geq 1 , \quad \text{and} \quad 1 \leq k \leq h$$

Notice that a constraint must be imposed so that the $(j+1)^{st}$ excavation will follow the j^{th} excavation in each mine and for each j for any strategy that is at all attractive. Bellman [1, pp. 66 ff] proposed this problem in the special case in which the success probability $p_{j,k}$ is a function only of k , and the machine mines a fixed fraction r_k of the gold in mine k each time, so $x_{j,k} = (1-r_k)^{j-1} r_k g_k$.

In the obstacle-course problem, each obstacle i will be overcome with probability p_i , $0 < p_i < 1$. There is a value x_i for doing so, and zero consolation prize. A runner is permitted to continue until he fails to overcome an obstacle.

Goodman [2] considers the special case of the obstacle course problem in which the x_i's are equal, and finds that the ordering putting the easiest obstacle first is optimal in the (strong) sense that the payoff is stochastically largest. He proves the same result where a failure diminishes, but does not necessarily eliminate, the probability of future success.

In the discrete-search problem (see [3] and the references cited there), an object is hidden in one of n boxes. Again a category corresponds to the j^{th} search of box k, and each category has a probability of success $p_{j,k}$ and a cost $c_{j,k}$. Superficially it might seem that the expected cost of finding the object satisfies (1). But this is not the case.

If $C(i_1\ i_2\ \ldots)$ is the expected cost of finding the object using the strategy $(i_1\ i_2\ \ldots)$ then

(2) $$C(i_1\ i_2\ \ldots) = c_{i_1} + (1 - p_{i_1})C^*(i_2\ \ldots)$$

where $C^*(i_2\ \ldots)$ is the expected cost of the strategy $(i_2\ \ldots)$ in a new problem with parameters

$$p^*_{i_k} = \frac{p_{i_k}}{1 - p_{i_1}} \qquad k > 1$$

$$c^*_{i_k} = c_{i_k}$$

Another way of understanding the distinction between the quiz show problem and the search problem is to examine the probability processes they generate. Suppose that the rules of both games are changed slightly, so that the quiz show and search go on indefinitely, regardless of whether the contestant has missed a question in the quiz show, or whether the object is found in the search. However the cost function is unchanged: the contestant receives only the sum of the x's for successes before his first failure, plus the consolation prize when he first fails (all appropriately discounted). Similarly the searcher is charged only for all unsuccessful searches before the object is found, and for the successful search. There may be any number of failures in this modified quiz show problem, but still the object can be found at most once by the searcher. Despite this difference, there is a connection between the ordering result for quiz shows and for search, which will be pursued in a later paper.

The results and methods of this paper are similar to those of dynamic programming. The most closely related material is found in [1], (especially Chapter II). Bellman's functional relations, however, are on the state space (amount of gold left in the mine), while (1) is a functional relation on strategy space.

Section 2 considers the ordering part of the quiz show problem, section 3 the choice part, and section 4 algorithms for computing optimal choices. Proofs of certain theorems are to be found in the appendix.

2. Optimal Ordering

In this section, assume that a choice has been made of which

r categories to attempt, and ask in what order they ought to be tried. For ease of exposition, assume $V(a) > 0$ for all categories a.

Define the postponability $P(b)$ of b as

$$P(b) = V(b)/(1 - S(b)) .$$

The natural idea in considering optimal ordering is to find out what happens if two adjacent categories or finite sequences of categories are reversed.

Lemma 1 If $P(c) > P(d)$ and $f = bcde$ and $g = bdce$, then $V(f) > V(g)$.

Proof
$$\begin{aligned} V(f) - V(g) &= S(b)[V(cd) - V(dc)] \\ &= S(b)[V(c)[1 - S(d)] - V(d)[1 - S(c)]] \\ &= S(b)[1 - S(d)][1 - S(c)][P(c) - P(d)] > 0 \end{aligned}$$

This leads immediately to

Theorem 1 Let $\sigma = (\sigma_1\ \sigma_2\ \ldots)$ be a strategy such that

(3) $$P(\sigma_i) \geq P(\sigma_j) \qquad \text{for } i \leq j$$

Then σ is optimally ordered among all strategies including $\{\sigma_1\ \sigma_2\ \ldots\}$. If $V(\sigma) < \infty$, then only strategies satisfying (3) are optimally ordered.

This property of $P(a)$ corresponds to Bellman's ([1], Chapter II) "decision regions."

Theorem 1 raises the question of existence of strategies satisfying (3). If sequences of any order type are allowed, such strategies certainly exist. Since a sequence of ordinality not ω would involve a difficulty of interpretation, it is of some interest to see when strategies of ordinality ω and satisfying (3) exist.

Let $\sigma = (a\ b \ldots.)$ be a strategy satisfying (3). Whether σ is of ordinality ω depends on the list $P(a), P(b), \ldots$. (In a set, repetitions of the same element can be eliminated; in a list they connot.) σ is of ordinality ω if and only if the list $L = P(a), P(b), \ldots$ satisfies the following conditions:

(*) L is bounded from above

(**) L has no more than one limit point, z

(***) If z exists, $P(a) \geq z$ for all categories a in σ .

It was noted in section 1 that the gold-mining problem is a quiz show problem subject to the special constraint that the $(j+1)^{st}$ mining of any mine must be subsequent to the j^{th} mining of that mine. More generally, let Q be a partial ordering on categories, and consider the problem of finding an optimal strategy σ subject to the condition that if σ includes categories a and b , and aQb , then σ is of the form $\sigma = cadbe$ (here c , d , or e may be empty, or sequences of categories).

The partial ordering Q is said to satisfy the regular order condition if

$$aQb \Longrightarrow P(a) \geq P(b)$$

Clearly if Q satisfies the regular order condition, the condition (3) is still necessary for σ to be optimal. Not all σ's satisfying (3) need be in accord with the partial ordering Q, but any σ satisfying (3) and in accord with Q is optimal. If Q satisfies the regular order condition, an optimal σ exists under the same necessary and sufficient conditions as before. A similar theorem for discrete search is given in [3, Theorem 3].

With the regularity assumptions (i) and (ii) on the gold-mining problem, the partial ordering that category $(j+1, k)$ must not preceed category (j, k) does satisfy the regular order condition, since

$$\frac{p_{j,k} \prod_{j'<j} (1 - r_{j',k}) r_{j,k} g_k}{1 - p_{j,k}} \geq \frac{p_{j+1,k} \prod_{j'<j} (1 - r_{j,k}) r_{j+1,k} g_k}{1 - p_{j+1,k}}$$

3. Optimal Choice of Categories

The problem in this section is to choose, from among n $(0 < n \leq \infty)$ possible categories, supposing that the contestant is permitted to choose no more than r of them $(r \leq n)$.

Without loss of generality, assume $r < n$ and hence $r < \infty$. Also continue to assume that $V(a) > 0$ for all a.

The problem of optimal choice of r categories is considerably more difficult than the ordering problem because the "one-by-one" approach which was successful in Theorem 1 does not apply. Consider the following obstacle-course example:

Let A and B have the following parameter values:

$$p_A = .2 \qquad x_A = 3$$

$$p_B = .5 \qquad x_B = 1$$

If the best two are to be chosen from $\{A, B, C\}$, where

$$p_C = .5 \qquad x_C = 2$$

direct computation shows that the best choice is $\{C, A\}$ (in that order, incidentally, according to Theorem 1).

However if the best two are to be chosen from $\{A, B, D\}$, where

$$p_D = .1 \qquad x_D = 7 ,$$

then the optimal choice is $\{B, D\}$ (in that order).

Hence which is preferable between A and B depends on whether it will be combined with C or with D. Therefore a theorem like Theorem 1 would be too strong for the problem of optimal choice.

The following theorem is true, however.

Theorem 2 [Fundamental Result] Every optimal choice of $r+1$ categories includes an optimal choice of r categories.

The proof of Theorem 2 is somewhat long, so it is given in the Appendix.

In the case in which the total number of categories, n, is finite, Theorem 2 suggests a computational procedure. Begin with all n categories, (clearly an optimal choice of n), choose the least costly single category to expel, which by Theorem 2 yields an optimal $n-1$, etc. This will be called Procedure I. Theorem 2 implies that Procedure I will find an optimal r.

Suppose instead the contestant uses Procedure II, which begins with the most valuable single category (highest V), adds to it the best second category to go with the one already chosen, etc. Were there no ties, Theorem 2 would prove that Procedure II finds an optimal r. However if there are ties, Theorem 2 proves only that some choice among the tied categories at whatever level the tie occurs leads to an optimal r. Theorem 3, however, shows that any choice in case of a tie leads to an optimal r.

Theorem 3 If R_1 and R_2 are two optimal choices of r categories, and if r_1 is the best $(r+1)^{st}$ choice to add to R_1, then

$$V[R_1 \cup \{r_1\}] = V[R^2 \cup \{r_2\}]$$

Where $V[C]$ is the expected value of the set C of categories, optimally ordered.

To apply this theory to the gold-mining problem, the effect of the partial order Q again is to be considered. Q satisfies the regular choice condition if

$$aQb \Longrightarrow V(a) \geq V(b) \text{ and } S(a) \geq S(b)$$

If the optimal choice problem is restricted so that if aQb and b is chosen, then a must also be chosen, application of (1) shows that there will be an unrestricted optimal r which satisfies this restriction. The conditions (i) and (ii) imply that the gold-mining problem satisfies the regular choice condition.

If negative V's were available, it would not be profitable to choose any of them. Hence the theory of this section could be used to choose the best set containing no more than r categories, and Theorem 1 shows the best order for that set. The main reason for presenting the theory in this fashion is that if negative V's are permitted, the natural sum representation for V of a strategy might not converge absolutely, and hence V might not exist in the sense necessary for this paper. If V were negative for a finite number of categories, it could still be extended to take a value for every strategy.

4. Three algorithms for optimal choice of categories

In section three an example was given to show that the problem of choosing the best r out of n possible categories is substantially more complex then the problem of optimal ordering. Additionally two procedures for choice were briefly discussed. This section gives more detail about Procedures I and II, and introduces Procedure III, which I think is of some practical importance.

The class of subsets of n categories is trivially a familiar distributive lattice under the inclusion relation, with $\binom{n}{r}$ nodes on the r^{th} level, $0 \leq r \leq n$. According to Theorem 2, there is at least one optimal chain and no descending optimal chains that stop, and according to Theorem 3, there are no ascending optimal chains that stop.

Procedure I starts at the top of this lattice, expelling the least profitable category at each stage as it descends an optimal chain. Procedure II starts at the bottom, picking the best single category, the best pair, etc., thus moving up an optimal chain. Procedure III starts at the appropriate level with any r categories and moves across the lattice changing one category at a time, until the optimum is reached. A fuller description, an upper bound on the number of iterations required, and a reference to the proof that each reaches an optimal choice is given below:

Procedure I (Start at the top).

1. Description: Start with all n categories. Expel the least profitable category, then the category least profitable among the $n-1$ remaining ones, etc.

   The loss in expelling category i between the sequences g and f in optimal order is given by

$$V(gif) - V(gf) = S(g)(1 - S(i))[P(i) - V(f)] \tag{4}$$

2. Bound for Procedure I: an upper bound on the number of comparisons required is given by

$$n + (n - 1) + \dots + (r + 1) = \frac{(n - r)(n + r + 1)}{2}$$

3. Termination at optimal r : Follows directly from theorem two.

Procedure II (Start at the bottom).

1. Description: Pick the best category (highest V), the best second category to go with it, etc. However it is not necessary to recompute the gain from adding each category each time. In going from an optimal r to an optimal $r + 1$, the ordering of the optimal r divides the unchosen $n - r$ categories into $r + 1$ groups, where any member of the j^{th} group would be ordered j^{th} were it chosen to be in the optimal choice for $r + 1$. The gain from adding a category i in group j is given by the expression in (4), which can be updated similarly.

2. Bound: An upper bound on the number of comparisons required is given by

$$n + (n - 1) + \dots + (n - r + 1) = \frac{r}{2}(2n - r + 1)$$

3. The procedure terminates only when the optimal r has been reached, according to theorems two and three.

Procedure III (Start with any r categories).

1. Description: IIIa (step-up, step down). Start with any set of r categories (preferably a carefully chosen r) and try to add each of the $n - r$ others, one by one. After adding each, yielding

a set of $r + 1$ categories, see which one is least valuable. If the one added is least valuable each time, terminate; if not, expel the category which is least valuable, yielding a set of r categories with greater expected value than before and continue.

IIIb (step-down, step-up). Start with any r categories, delete each one in sequence, yielding $r - 1$ categories. Now see which among the $n - r + 1$ not included would be most valuable to add. If it is the one deleted each time, stop; if not, take the most valuable addition and continue.

2. Bound: There are $\binom{n}{r}$ possible subsets of size r, and each can take as much as $r(n-r)$ comparisons to check, so the upper bound is $\frac{n!}{(r-1)!(n-r-1)!}$. But this bound will be far larger, in general, than the actual number of iterations required. Any prior information, such as a near optimal solution, can be used in Procedure III. In the absence of such prior information, I recommend the use of a convex function of $V(a)$ and $P(a)$ as criterion for the initial r, and for order in which to try new candidates.

3. Termination at the optimal r: See Theorem Four.

Theorem 4 Procedure III stops only when an optimal r has been reached. The proof is given in the appendix.

Example of the three Procedures.

Consider the three obstacles given below:

	p	x	px	P	
A	.85	1	.850	5.67	V(AB) = 1.564
B	.42	2	.840	1.45	V(CA) = 1.5225
C	.87	.9	.783	6.02	V(CB) = 1.5138

Procedure I:

Start at the top with all three chosen, and see which obstacle would cost least to expel. Let $q(a) = 1 - S(a)$. Then the cost of expelling

$$A \text{ is } p_C(p_A x_A - q_A V(B)) = .6298$$

$$B \text{ is } p_C p_A(p_B x_B) = .62118$$

$$C \text{ is } p_C v_C - q_C V(AB) = .57968$$

Hence obstacle C costs least to expel, and {A,B} is the optimal pair. The resultant expected value is $V(CAB) - .57968 = 1.564$ as before.

Procedure II:

Choose the first obstacle according to highest V , namely A . Then the gain from choosing B (which is in group 2 because were it chosen to be second with A it would be second in order) is $p_A(p_B x_B) = .714$; the gain from choosing C , which is in group

1, is $p_C x_C - q_C(V(A)) = .6725$. Hence B is chosen to be second, and thus the optimal pair is found.

Procedure III:

Suppose the function $f = \frac{1}{2}V + \frac{1}{2}P$ is used as the discriminator function. Then $f(A) = 3.26$, $f(B) = 1.145$, $f(C) = 3.4015$. Hence A and C would be chosen to be the initial solution. In Procedure IIIa, test to see whether to add B by going through exactly the same analysis as in Procedure I. In Procedure IIIb, test whether to expel A and add B (no), then test whether to expel C and add B (yes).

Discussion: Procedure I is clearly most useful for r/n large, and Procedure II for r/n small. The chief question is the usefulness of Procedure III. The advantage of Procedure III is that it can use knowledge of any near-optimal solution. I conjecture that with a suitable convex combination of V and P , Procedure III may be fastest in a middle range of r/n for large n .

5. Conclusion

The ordering problem for quiz shows is very simple, and the solution similar to Bellman's "decision regions." The choice problem is reminiscent of the problem of choosing the best r regressors of n possibilities. Procedure II is quite similar to stepwise regression. Perhaps something like Procedure III would find the best r regressors.

Also it may be noted that Procedure III is similar to the simplex method of linear programming. In an n variable problem with r linear equality constraints, according to a fundamental linear programming theorem there is an optimal solution with at most r variables strictly positive. Given which variables are allowed to be strictly positive, it is easy to find the value for the objective function. Thus the linear programming problem reduces to finding which r variables are allowed to be strictly positive. If there is a candidate set of r (a "basis") having the property that it is not profitable to expel any single variable and include any previously excluded variable, then according to another fundamental linear programming theorem, the candidate basis is optimal. Thus this latter theorem is similar to Theorem 4 of this paper.

There appears to be a complexity scale for problems in which ordering for quiz shows appears simplest, choice for quiz shows and linear programming next most complicated, and the choice of regressors more complicated still.

Appendix: Proofs of Theorems 2, 3 and 4

Notation

$\emptyset$ and R^c are the null set and the complement of R, respectively.

Theorem 2 [Fundamental Result] Every choice of $r + 1$ categories includes an optimal choice of r categories.

Proof

Suppose the contrary were so. Then for some $r + 1$ there is an optimal choice R^* which contains no optimal choice of r categories. Consider some set R which is an optimal choice of r categories. It will be shown that R^* can be improved (strictly) by the inclusion of R, which contradicts the hypothesis that R^* is optimal.

Since R has r elements and R^* has $r + 1$ elements, $R^* \cap R^c \neq \emptyset$. Order R^* optimally and select the first category f in this optimal ordering of R^* such that $f \notin R$. It will be shown that $V[R^*] < V[R \cup \{f\}]$.

Case I. If f occurs first in the optimal ordering of R^*, then

$$V[R^*] = V(f) + S(f)V[R^* - \{f\}] < V(f) + S(f)V(R) \leq V[R \cup \{f\}]$$

by the optimality of R and sub-optimality of $R^* - \{f\}$.

Case II. If f does not occur first in the optimal ordering, the same idea is used. Move f to first position R^* at some loss L

in expected value, insert R for $R^* - \{f\}$ at a strict gain, and then move f to its optimal position in $R \cup \{f\}$ at a gain G in expected value. If it can be shown that $G \geq L$, the theorem will be established.

Without loss of generality, R^* in optimal order can be written as $b_1 b_2 \ldots b_n fg$, and R in optimal order as $c_1 b_1 c_2 b_2 \ldots c_n b_n c_{n+1} fh$, where g, c_1, c_{n+1} and h may be empty. Then

$$L = V(b_1 \ldots b_n fg) - V(fb_1 \ldots b_n g)$$

$$= V(b_1 \ldots b_n f) - V(fb_1 \ldots b_n) \qquad \text{and}$$

$$G = V(c_1 b_1 \ldots c_n b_n c_{n+1} fh) - V(fc_1 b_1 \ldots c_n b_n c_{n+1} h)$$

$$= V(c_1 b_1 \ldots c_n b_n c_{n+1} f) - V(fc_1 b_1 \ldots c_n b_n c_{n+1})$$

The following lemma completes the proof of Theorem 2

Lemma 2

For any finite strategies $c_1, \ldots, c_{n+1}, b_1 \ldots b_n, f$ (c_1, c_{n+1} possibly empty), such that $P(c_i) \geq P(b_i) \geq P(c_{i+1}) \geq P(f)$ for all i and non-empty c,

$$V(b_1 \ldots b_n f) - V(fb_1 \ldots b_n) \leq V(c_1 b_1 \ldots c_n b_n f c_{n+1}) - V(fc_1 b_1 \ldots c_n b_n c_{n+1}) .$$

Proof

$$V(a_1 \ldots a_n f) - V(fa_1 \ldots a_n) = [1 - S(f)] \sum_{i=1}^{n} \left(\prod_{j<i} S(a_j)(1 - S(a_i))(P(a_i) - P(f)) \right)$$

for any finite strategies a by repeated use of lemma 1. Therefore

$$G = V(c_1b_1 \dots c_nb_nc_{n+1}f) - V(fc_1b_1 \dots c_nb_nc_{n+1})$$

$$\geq V(c_1b_1 \dots c_nb_nf) - V(fc_1b_1 \dots c_nb_n)$$

$$= [1 - S(f)] \sum_{i=1}^{n} \left(\prod_{j<i} S(c_jb_j)\right)(1 - S(c_ib_i))(P(c_ib_i) - P(f))$$

Since $P(cb)$ is a convex combination of $P(c)$ and $P(b)$, and $P(c_i) \geq P(b_i)$ for all i , $P(c_ib_i) \geq P(b_i)$. Then

$$G \geq [1 - S(f)] \sum_{i=1}^{n} \left(\prod_{j<i} S(c_jb_j)\right)(1 - S(c_ib_i))(P(b_i) - P(f))$$

Also note that $S(c_ib_i) \leq S(b_i)$ for all i . Let $X(y_1, y_2, \dots, y_n)$ be a real valued function of n real numbers y defined by

$$X(y_1, y_2, \dots, y_n) = [1 - S(f)] \sum_{i=1}^{n} \left(\prod_{j<i} y_j\right)(1 - y_i)(P(b_i) - P(f)) .$$

Then $G \geq X(S(c_1b_1), S(c_2b_2), \dots, S(c_nb_n))$ and $L = X(S(b_1), S(b_2), \dots, S(b_n))$.

Since $\frac{\partial X}{\partial y_k} \leq 0$, $G \geq L$. Lemma 2 and Theorem 2 are established.

<u>Theorem 3</u> <u>If</u> R_1 <u>and</u> R_2 <u>are two optimal choices of</u> r <u>obstacles, and if</u> r_i <u>is the best</u> $(r+1)^{st}$ <u>choices to add to</u> R_i , <u>then</u>

$$V[R_1 \cup \{r_1\}] = V[R_2 \cup \{r_2\}] .$$

Proof

Case I: $R_1 \cup \{r_1\} = R_2 \cup \{r_2\}$. Then the result is trivial.

Case II: $R_1 \cup \{r_1\} \neq R_2 \cup \{r_2\}$. Then let $\bar{r}_1$ be that number of $(R_1 \cup \{r_1\})(R_2 \cup \{r_2\})^c$ with the highest P (or one of then if more than one exists). Then

$$V[R_1 \quad \{r_1\}] = V[\{\bar{r}_1\} \cup (R_1 \cup \{r_1\} - \{\bar{r}_1\})] \leq V[\{r_1\} \cup R_2] \leq V[\{r_2\} \cup R_2]$$

by the proof of Theorem 2. Since similarly $V[R_2 \cup \{r_2\}] \leq V[R_1 \cup \{r_1\}]$, the result follows.

Theorem 4 Procedure III stops only when an optimal choice of r categories has been reached.

Proof:

Let U be a set of r categories for which procedure III terminates, i.e., there are no categories $a \in U$ and $b \notin U$ such that $V[U] < V[U \cup \{b\} - \{a\}]$. A proof that U is then an optimal choice of r categories sufficies for both Procedures IIIa and IIIb.

U must contain an optimal choice of one category, with highest V , as otherwise such a category could be substituted in last place in U , increasing expected value.

Suppose U contains K , an optimal set of k categories, but no $K^* \supset K$, K^* an optimal set of $k+1$ categories (for $k+1 < r$) which exists according to Theorems 2 and 3. Let a be that member of $U \cap K^c$ last in an optimal ordering of U . Then that optimal ordering of U can be written

$$U_1k_1U_2k_2 \dots U_m ak_m$$

where k_i is a finite (non-empty) sequence of elements of K and u_i is a finite sequence of elements of U, with u_1 and u_m possibly empty. Let

$$L = V(u_1k_1u_2k_2 \dots u_m ak_m) - V(u_1u_2 \dots u_m k_1 \dots k_{m-1}ak_m) .$$

Now $k_1 \dots k_{m-1}ak_m$ is a fortiori not an optimal choice of $k+1$ categories. Suppose that $K^* = K \cup \{b\}$, where $b \notin U$. The remainder of this proof shows that $V[U] < V[U \cup \{b\} - \{a\}]$, a contradiction. Then U contains (and hence is) an optimal choice of r categories by induction. To establish this, distinquish two cases according to where b falls in an optimal ordering of K^*.

Case I: An optimal ordering of K^* is $k_1 \dots k'_{m-1}bk''_m$ where $k'_mk''_m = k_m$ and one of k'_m, k''_m might be empty. Then

$$\begin{aligned} V[S] &= V(u_1k_1u_2k_2 \dots u_m ak_m) \\ &= L + V(u_1u_2 \dots u_m k_1 \dots k_{m-1}ak_m) \\ &< L + V(u_1u_2 \dots u_m k_1 \dots k_{m-1}k'_m bk_m) \\ &= V(u_1k_1u_2k_2 \dots u_m k'_m bk_m) \\ &= V[U \cup \{b\} - \{a\}] . \end{aligned}$$