GENERALIZED LAGRANGE MULTIPLIER METHOD FOR SOLVING PROBLEMS OF OPTIMUM ALLOCATION OF RESOURCES

Hugh Everett III

Weapons Systems Evaluation Division, Institute for Defense Analyses, Washington, D C

(Received August 20, 1962)

The usefulness of Lagrange multipliers for optimization in the presence of constraints is not limited to differentiable functions They can be applied to problems of maximizing an arbitrary real valued objective function over any set whatever, subject to bounds on the values of any other finite collection of real valued functions defined on the same set While the use of the Lagrange multipliers does not guarantee that a solution will necessarily be found for all problems, it is 'fail-safe' in the sense that any solution found by their use is a true solution Since the method is so simple compared to other available methods it is often worth trying first, and succeeds in a surprising fraction of cases They are particularly well suited to the solution of problems of allocating limited resources among a set of independent activities

IN MOST textbook treatments, Lagrange multipliers are introduced in a context of differentiable functions, and are used to produce constrained stationary points Their validity or usefulness often appears to be connected with differentiation of the functions to be optimized Many typical operations-research problems, however, involve discontinuous or nondifferentiable functions (integral valued functions, for example), which must be optimized subject to constraints

We shall show that with a different viewpoint the use of Lagrange multipliers constitutes a technique whose goal is *maximization* (rather than location of stationary points) of a function with constraints, and that in this light there are no restrictions (such as continuity or differentiability) on the functions to be maximized Indeed, the domain of the function to be maximized can be any set (of any cardinal number) whatever

The basic theorems upon which the techniques to be presented depend are quite simple and elementary, and it seems likely that some of them may have been employed previously However, their generality and applicability do not seem to be well understood at present (to operations analysts at least) The presentation will consequently place primary emphasis on the implications and applications of the basic theorems, as well as

discussion of a number of techniques for extending the usefulness of the methods

FORMULATION

For clarity of presentation, we shall develop the subject in a language of problems concerning the optimal allocation of resources Other applications of the theorems will suggest themselves

Let us suppose that there is a set S (completely arbitrary) that is interpreted as the set of possible strategies or actions Defined on this strategy set is a real valued function H, called a *payoff function* $H(x)$ is interpreted as the payoff (or utility) which accrues from employing the strategy $x \epsilon S$ In addition, there are n real valued functions $C^k(k=1 \quad n)$ defined on S, which are called *Resource* functions The interpretation of these functions is that employment of the strategy $x \epsilon S$ will require the expenditure of an amount $C^k(x)$ of the kth resource

The problem to be solved is the maximization of the payoff subject to given constraints $c^k, k=1 \quad n$, on each resource, i e , to find

$$\max_{x \epsilon S} H(x)$$

subject to $C^k(x) \leqq c^k$, all k

A particular subclass of this general problem with wide application is what will be called a *cell problem* (or separable problem) in which there are a number, m, of independent areas into which the resources may be committed, and for which the over-all payoff that accrues is simply the sum of the payoffs that accrue from each independent venture (cell) In this type of problem we have as before, for each cell, a strategy S_i, a payoff function H_i defined on S_i, and n resource functions C_i^k defined on S_i $H_i(x_i)$ is the payoff in the ith cell for employing strategy $x_i \epsilon S_i$, and for each k, $C_i^k(x_i)$ is the amount of the kth resource expended in the ith cell by employing strategy x_i in that cell In this case the problem to be solved is to find a strategy set, one element for each cell, which maximizes the total payoff subject to constraints c^k on the total resources expended, i e ,

$$\max_{\substack{\text{all choices of } \{x_i\} \\ x_i \epsilon S_i}} \sum_{i=1}^{i=m} H_i(x_i)$$

subject to $$\sum_{i=1}^{i=m} C_i^k(x_i) \leqq c^k \quad \text{for all } k$$

This type of problem is simply a subclass of the previous general problem since it can be translated to the previous problem by the following identifications

$$S = \prod_{i=1}^{i=m} S_i \text{ (direct product set)},$$

[where a strategy $x \epsilon S$ consists of an ordered m-tuple $(x_1, \quad , x_n)$ of strategies, one for each S_i]

$$H(x) = \sum_{i=1}^{i=m} H_i(x_i),$$

$$C^k(x) = \sum_{i=1}^{i=m} C_i^k(x_i), \quad \text{all } k$$

MAIN THEOREM AND SOME OF ITS IMPLICATIONS

WE NOW present the main theorem concerning the use of Lagrange multipliers, and discuss its meaning and implications The proof will be supplied in a later section

THEOREM 1

1 λ^k, $k=1, n$ *are nonnegative real numbers,*

2 $x^* \epsilon S$ *maximizes the function*

$$H(x) - \sum_{k=1}^{k=n} \lambda^k C^k(x) \text{ over all } x \epsilon S,$$

→3 x^* *maximizes* $H(x)$ *over all those* $x \epsilon S$ *such that* $C^k \leqq C^k(x^*)$ *for all* k

Discussion

This theorem says, for any choice of nonnegative λ^k, $k=1, n$, that if an *unconstrained* maximum of the new (Lagrangian) function

$$H(x) - \sum_{k=1}^{k=n} \lambda^k C^k(x)$$

can be found (were x^*, say, is a strategy which produces the maximum), then this solution is a solution to that *constrained* maximization problem whose constraints are, in fact, the amount of each resource expended in achieving the unconstrained solution Thus if x^* produced the unconstrained maximum, and required resources $C^k(x^*)$, then x^* itself produces the greatest payoff which can be achieved without using more of any resource than x^* does

According to Theorem 1, one can simply choose an arbitrary set of nonnegative λ's, find an unconstrained maximum of the modified function, $H(x) - \sum_{k=1}^{k=n} \lambda^k C^k(x)$, and one has as a result a solution to a constrained problem Notice, however, that the particular constrained problem which is solved is not known in advance, but arises in the course of solution and is, in fact, the problem whose constraints equal the resources expended by the strategy that solved the unconstrained problem

In general, different choices of the λ^k's lead to different resource levels, and it may be necessary to adjust them by trial and error to achieve any given set of constraints stated in advance

However, it is noteworthy that in most operations-research work one is not simply interested in achieving the optimum payoff for some given resource levels, but rather in exploring the entire range of what can be

obtained as a function of the resource commitments In this case it matters little whether this function is produced by solving a spectrum of problems with constraints stated in advance, or by simply sweeping through the λ^k's to solve a spectrum of problems whose constraint levels are produced in the course of solution The method when applicable is therefore quite efficient if the whole spectrum of constraints is to be investigated Even in the case where only a single constraint set is of interest the use of this method, and adjustment of the λ^k's until the constraint set is achieved, is often more efficient than alternative procedures

A limitation of the Lagrange multiplier method arises from the fact that it does not guarantee that an answer can be found in every case It simply asserts that if an answer can be found it will indeed be optimum

In cases where multiple constraints are involved that are not completely independent it may not be possible to simultaneously utilize all resources to the full allowance of the constraints This can happen if the utilization of one resource requires the utilization of others, or equivalently in cases where some constraints may involve various combinations of others These cases are analogous to problems in linear programming where certain constraints prove to be irrelevant in the optimum solution

In such cases one might actually find the optimum solution but be unable to establish the optimality of the result because of incompletely utilized resources Nevertheless, there is a large class of allocation problems in which the constraints really are independent (i e , the resources can be consumed independently in the region of interest) In such cases solutions can usually be obtained that give consumption values adequately close to the constraint values The existence of optimum solutions that can be found by this method actually depends upon an approximate concavity requirement in the region of the solution that will be discussed more carefully later

At this point we wish to remind the reader of the generality of Theorem 1 *There are no restrictions whatever on the nature of the strategy set* S, *nor on the functions H and* C^k *other than real-valuedness* The strategy set may therefore be a discrete finite set, or an infinite set of any cardinality Furthermore, the payoff function and the resource functions can take on negative as well as positive values [$C^k(x)$ negative may be interpreted as *production* rather than expenditure of the kth resource]

Application to Cell Problem

One of the most important applications of Theorem 1 is in the solution of cell problems As shown in the Formulation Section, these problems are a subclass of the general problem to which Theorem 1 is applicable In this case, maximizing the unconstrained Lagrangian function

$$H(x) - \sum_{k=1}^{k=m} \lambda^k C^k(x)$$

is equivalent to finding

$$\max_{x \in \Pi_{i=1}^{i=m}} \mathcal{S}_i [\sum_{i=1}^{i=m} H_i(x_i)] - \sum_{k=1}^{k=n} \lambda^k [\sum_{i=1}^{i=m} C_i^k(x_i)],$$

which (interchanging summation order) is the same as

$$\max_{x \in \Pi_{i=1}^{i=m}} \mathcal{S}_i \sum_{i=1}^{i=m} [H_i(x_i) - \sum_{k=1}^{k=n} \lambda^k C_i^k(x_i)]$$

But, since the choices x_i may be made independently in each cell as a consequence of $\mathcal{S} \equiv \prod_{i=1}^{i=m} \mathcal{S}_i$, the sum is obviously maximized by simply maximizing

$$H_i(x_i) - \sum_{k=1}^{k=n} \lambda^k C_i^k(x_i)$$

in each cell independently of strategy choices in other cells, and summing the payoffs and resources expended for each cell (for the strategy that maximized the Lagrangian for that cell) to get the total payoff and resource expenditures Theorem 1 then assures us that the result of this process is a solution to the over-all constrained problem with constraints equal to the total resources expended by the strategy produced by this procedure

Observe that there is no possibility that just a local maximum to the over-all problem has been obtained If the Lagrangian in each cell has been correctly maximized (i e, is not itself merely locally maximized), then theorem 1 guarantees that the result is a *global* maximum to the over-all problem

Theorem 1 says nothing about the manner in which one obtains the maxima of the unconstrained Lagrangian functions, but simply asserts that if one can find them, then one can also have maxima of a problem with constraints The Lagrange multipliers therefore are not a way in themselves of finding maxima, but a technique for converting optimization problems with constrained resources into unconstrained maximization problems

This conversion is especially crucial for cell problems with constraints on total resource expenditures, where the conversion to unconstrained maximization of the Lagrangian function uncouples what was an essentially combinatorial problem (because of the interaction of choices in each cell through total resource constraints) into a vastly simpler problem involving independent strategy selections in each cell

The present treatment of Lagrange multipliers was motivated, in fact, by a cell problem involving continuous, differentiable payoff functions, the solution of which was attempted by a classical Lagrange multiplier approach In this case, the resulting (transcendental) equations had in many circumstances a multiplicity of solutions, and the embarrassing problem arose as to which of several solutions to select for each cell It appeared as though it might be necessary to try all combinations of choices of solutions—an impossible task in this case which involved several hun-

dred cells As a result of this difficulty, a closer look was taken at the role of Lagrange multipliers, and the present treatment is the result The original problem of multiple solutions is, of course, easily solved by simply selecting that solution in each cell which gives the largest value for the Lagrangian

It is the recognition that the objective is to maximize the Lagrangian, by whatever means, not to zero its derivative, which is decisive In many cases it *is* expeditious to maximize the Lagrangian by finding zeroes of its derivative One can then easily select a final value by testing each solution (if there is more than one) to find which gives the largest (global) maximum This procedure automatically excludes any solutions that correspond to minima or saddle values, and also facilitates taking into account any boundary conditions (such as nonnegative resource constraints) by testing the boundary cases as well †

In other cases (particularly cases of nonnumerical strategies, or discrete strategy sets such as integers), the Lagrangian may best be maximized by trial and error procedures, or even direct computer scanning of all possibilities

Another possibility is illustrated by cases wherein resources may be applied only in integral numbers Often in such cases one can define a continuous differentiable payoff function that attains its correct value on the integers A useful trick applicable to many such cases is to maximize analytically the Lagrangian based upon the continuous function, and then test the integer on each side of the solution, selecting the one that maximizes the Lagrangian

PROOF OF MAIN THEOREM

THE PROOF of the main theorem presented and discussed in the previous section is quite elementary and direct

Proof of Main Theorem By assumptions (1) and (2) of Theorem 1, λ^k, $k=1 \quad n$, are nonnegative real numbers, and $x^* \epsilon S$ maximizes

$$H(x) - \sum_{k=1}^{k=n} \lambda^k C^k(x)$$

over all $x \epsilon S$ (the x^* producing the maximum may very well not be unique—all that we require is that x^* be *some* element that maximizes the Lagrangian) This means that, for all $x \epsilon S$,

$$H(x^*) - \sum_{k=1}^{k=n} \lambda^k C^k(x^*) \geqq H(x) - \sum_{k=1}^{k=n} \lambda^k C^k(x),$$

† This type of constraint (e g , nonnegativity of resources), which holds independently for each cell rather than over-all as with total resources, is handled by simply restricting the strategy set for the cell appropriately The Lagrange multipliers are reserved for over-all constraints

and hence, that

$$H(x^*) \geqq H(x) + \sum_{k=1}^{k=n} \lambda^k [C^k(x^*) - C^k(x)]$$

for all $x \epsilon \mathcal{S}$ But if the latter inequality is true for all $x \epsilon \mathcal{S}$, it is necessarily true for any subset of $\mathcal{S}$, and hence true on that subset $\mathcal{S}^*$ of $\mathcal{S}$ for which the resources never exceed the resources $C^k(x^*)$ Notationally $x \epsilon \mathcal{S}^* \Leftrightarrow$ for all k, $C^k(x) \leqq C^k(x^*)$ However, on the subset $\mathcal{S}^*$ the term

$$\sum_{k=1}^{k=n} \lambda^k [C^k(x^*) - C^k(x)]$$

is nonnegative by definition of the subset and the nonnegativity of the λ^k's, hence our inequality reduces to $H(x^*) \geqq H(x)$ for all $x \epsilon \mathcal{S}^*$, and the theorem is proved

LAMBDA THEOREM

THEOREM 2

1 Let $\{\lambda_1^k\}, \{\lambda_2^k\} k = 1 \quad n$ *be two sets of* λ^k's *that produce solutions* x_1^* *and* x_2^*, *respectively Furthermore, assume that the resource expenditures of these two solutions differ in only the* j*th resource*

$$C^k(x_1^*) = C^k(x_2^*) \text{ for } k \neq j$$

and that $C^j(x_1^*) > C^j(x_2^*)$

2 Then $\lambda_2^j \geqq [H(x_1^*) - H(x_2^*)]/[C^j(x_1^*) - C^j(x_2^*)] \geqq \lambda_1^j$

This theorem states that, given two optimum solutions produced by Lagrange multipliers for which only one resource expenditure differs, the ratio of the change in optimum payoff to the change in that resource expenditure is bounded between the two multipliers that correspond to the changed resource

Thus the Lagrange multipliers, which were introduced in order to constrain the resource expenditures, in fact give some information concerning the effect of relaxing the constraints

In particular, if the set of solutions produced by Lagrange multipliers results in an optimum payoff that is a differentiable function of the resources expended at some point, then it follows from Theorem 2 that the λ^k's at this point are in fact the partial derivatives (or total derivative in case of one resource) of the optimum payoff with respect to each resource (all other resources kept constant)

$$[\partial H^*/\partial C^j]_{\substack{C_k \text{ constant} \\ k \neq j}} = \lambda^j$$

Proof The proof of Theorem 2 is also quite elementary By hypothesis x_1^* is the solution produced by $\{\lambda_1^k\}$, hence x_1^* maximizes the Lagrangian for $\{\lambda_1^k\}$, which implies

$$H(x_1^*) \geqq H(x) + \lambda_1^j [C^j(x_1^*) - C^j(x)] + \sum_{k \neq j} \lambda_1^k [C^k(x_1^*) - C^k(x)]$$

holds for all $x \epsilon S$, and hence in particular holds for x_2^* But since by hypothesis $C^k(x_1^*) = C^k(x_2^*)$ for $k \neq j$, we can deduce that

$$H(x_1^*) \geqq H(x_2^*) + \lambda_1^j[C^j(x_1^*) - C^j(x_2^*)],$$

which, since by hypothesis $C^j(x_1^*) > C^j(x_2^*)$, implies that

$$[H(x_1^*) - H(x_2^*)]/[C^j(x_1^*) - C^j(x_2^*)] \geqq \lambda_1^j,$$

which proves one side of the conclusion of Theorem 2 Interchanging the roles of x_1^* and x_2^* [and observing the reversal of the sign of

$$C^j(x_1^*) - C^j(x_2^*)]$$

produces the other side of the inequality to complete the proof of Theorem 2

An obvious consequence of Theorem 2 is the fact that, if all but one resource level is held constant, the resource that changes is a monotone *decreasing* function of its associated multiplier This fact indicates the direction to make changes when employing a trial and error method of adjusting the multipliers in order to achieve some given constraints on the resources

The Lambda Theorem also suggests a potentially useful technique for choosing a starting set of multipliers for such a trial-and-error method of achieving given constraint levels in a cell problem Beginning with any reasonably good allocation of the given resources, one can often calculate easily what the effect on the payoff is for a small additional increment of each resource, optimally placed within the cells The differential payoff divided by the increment of resource is then taken as the starting λ for that resource The λ's are then adjusted by trial and error until the Lagrange solution corresponds to the given constraints, producing the optimum allocation

THE EPSILON THEOREM

A NATURAL question with respect to the practical application of the Lagrange method concerns its stability—supposing that as a result of methods of calculation or approximation one cannot precisely maximize the Lagrangian, but can only guarantee to achieve a value close to the maximum Such a solution can very well be at a drastically different resource level and payoff than that which actually achieves the maximum, and yet produce a value of the Lagrangian very near to the maximum For the method to be practical, it is required that in this situation a solution that nearly maximizes the Lagrangian must be a solution that also nearly maximizes the payoff for the resource levels that *it itself* produces (which may be quite different than those of the solution that actually

maximizes the Lagrangian) Only in such a circumstance would it be safe to assert that the solutions produced by any nonexact procedures (such as numerical computation with finite accuracy, or methods based upon approximations) were in fact approximately optimal solutions to the constrained problem Such required assurance of insensitivity is supplied by the following ('epsilon') theorem

THEOREM 3

1 $\bar{x}$ comes within ϵ of maximizing the Lagrangian, i e , for all $x \epsilon S$

$$H(\bar{x}) - \sum \lambda^k C^k(\bar{x}) > H(x) - \sum \lambda^k C^k(x) - \epsilon$$

→2 $\bar{x}$ is a solution of the constrained problem with constraints $c^k = C^k(\bar{x})$ that is itself within ϵ of the maximum for those constraints

The proof of this theorem, which is a simple extension of Theorem 1, exactly parallels the proof of Theorem 1 (with an added ϵ) and will not be repeated

ADDITIONAL REMARKS, CONCLUSIONS, AND COMPUTATIONAL PLOYS

Gaps or Inaccessible Regions

Theorem 1 assures us that any maximum of the Lagrangian necessarily is a solution of the constrained maximum problem for constraints equal to the resource levels expended in maximizing the Lagrangian

The Lagrange multiplier method therefore generates a mapping of the space of lambda vectors (components λ^k, $k=1, \quad , n$) into the space of constraint vectors (components c^k, $k=1 \quad n$) There is no a priori guarantee, however, that this mapping is onto—for a given problem there may be inaccessible regions (called *gaps*) consisting of constraint vectors that are not generated by any λ vectors Optimum payoffs for constraints inside such inaccessible regions can therefore not be discovered by straightforward application of the Lagrange multiplier method, and must hence be sought by other means

The basic cause of an inaccessible region is nonconcavity in the function of optimum payoff vs resource constraints (convexities in the envelope of the set of achievable payoff points in the space of payoff vs constraint levels) This possibility, and several methods for dealing with it, will now be investigated

Before beginning this investigation, however, we wish to point out that even though the Lagrange multiplier method is not certain to obtain the desired solutions in all cases, any solutions that it does yield are guaranteed by Theorem 1 to be true solutions The procedure is therefore 'fail-safe,' a very reassuring property It has been our experience over the last several years, which includes application of this method to a variety

of production and military allocation problems, that the method has been extremely successful, and nearly always has directly yielded all solutions of interest The few situations in which the direct method failed were readily solved by simple modifications to the procedure, some of which will now be mentioned

Source of Gaps

Consider the $(n+1)$ dimensional space of payoff vs resource expenditures This space will be called PR space for brevity Every strategy $x \epsilon S$ maps into a point in this space corresponding to $H(x), C^k(x)$ $(k=1 \quad n)$ The entire problem is therefore represented by this set of accessible points in PR space The problem of finding the maximum of H subject to constraints c^k, $k=1 \quad n$, is simply the problem of selecting that point of our set in PR space of maximum H that is contained in the subspace of PR space where the resources are bounded by the c^k's The set of all such points (corresponding to all sets of values in the c^k's) will be called the *envelope*, and constitutes the entire set of solutions for all possible constraint levels

Consider now any solution x^* produced by a set of Lagrange multipliers (λ^k) By definition x^* maximizes the Lagrangian, consequently we have that

$$H(x^*) - \sum \lambda^k C^k(x^*) \geqq H(x) - \sum \lambda^k C^k(x)$$

for all $x \epsilon S$ Rearranging terms slightly, we have

$$H(x) \leqq H(x^*) - \sum \lambda^k C^k(x^*) + \sum \lambda^k C^k(x)$$

for all $x \epsilon S$ If we consider now the hyperplane in PR space defined by $H = \sum \lambda^k C^k + \alpha$ where $\alpha = H(x^*) - \sum \lambda^k C^k(x^*)$, we see that, because of the previous inequality, none of the accessible points in PR space lies above this hyperplane, and at least one point, $H(x^*), C^k(x^*)$ $k=1 \quad n$, lies on it

Each solution produced by Lagrange multipliers therefore defines a bounding hyperplane that is tangent to the set of accessible points in PR space at the point corresponding to the solution (hence tangent to the envelope), and which constitutes an upper bound to the entire set of accessible points It is clear that, since no such tangent bounding hyperplanes exist in regions where the envelope of accessible points in PR space is not concave, the Lagrange multiplier method cannot produce solutions in such a region Conversely, for any point on the envelope (solution) where a tangent bounding hyperplane *does* exist (envelope concave at the point), it is obvious that there exists a set of multipliers (namely the slopes of the hyperplane) for which the strategy corresponding to the point in question maximizes the Lagrangian

Thus the Lagrange method will succeed in producing all solutions that correspond to concave regions of the envelope (optimized payoff vs constraint level), and fail in all nonconcave regions

A fortunate feature of cell problems with many cells is the fact that, even though there may be large convexities in the envelope in the PR space for each cell, the result of over-all optimization is an envelope in the PR space for the total problem in which the convexities are vastly reduced in significance † This property is the major reason for the general success of the Lagrange method in solving cell problems

Some Methods for Handling Gaps

Despite the general success of Lagrange multipliers (at least for the problems we have encountered), occasions may arise where gaps occur in regions of critical interest Under such circumstances there are several useful techniques that can be attempted before abandoning the procedure altogether

First, all solutions that can be obtained outside the gaps contribute a good deal of information and can be used to bound the solution in the gap region As was previously shown, each solution that can be obtained by Lagrange multipliers defines a bounding hyperplane that gives an upper bound to the maximum payoff at all points, and hence inside the gap as well For any point inside a gap, therefore, an upper bound can be obtained by finding the minimum payoff for that point over the set of bounding hyperplanes corresponding to the solutions that one could calculate

On the other hand, every solution that can be obtained that has the property that none of its resource expenditures exceeds the resources of a point in a gap for which one is seeking bounds, obviously constitutes a *lower bound* to the optimum payoff at the point in question, and the maximum of these lower bounds can be selected as a lower bound to the payoff in question Thus the set of solutions that *can* be obtained by Lagrange multipliers can be used to obtain bounds on the optimum payoff for inaccessible regions

There is another technique that is often successful in reducing gaps in instances where the bounds one can compute leave too large a region of uncertainty, and where the gap is caused by degeneracy in which a number of cells have gaps corresponding to the same multiplier A gap is char-

† In fact, the gap structure for the over-all problem obviously simply reflects faithfully the gap structure in the individual cells, with each gap in a cell corresponding to a given multiplier value occurring with the same magnitude (same jump in payoff and resources) in the over-all optimization at precisely the same multiplier value Only degeneracies in which several cells have gaps corresponding to the same multiplier can cause a larger gap in the over-all problem, and such degeneracy is easily removed by techniques to be discussed in the following section

acterized by the behavior that, as the λ's are continuously varied, there are abrupt discontinuities in the resource levels generated These discontinuities can often be filled in cell problems by the following technique

Given two sets of λ's, (λ_1^k), (λ_2^k), which are very close, but for which the generated resource levels markedly differ, one can make a *mixed* calculation in a cell problem using the set (λ_1^k) in some cells and the set (λ_2^k) in the others If the two sets of λ's are close together, maximizing the Lagrangian in any cell for one set will necessarily result in a solution that nearly maximizes the Lagrangian for the other set, hence by the Epsilon Theorem will yield a result that is guaranteed to be nearly optimum

Somewhat more generally, one can simply exploit the Epsilon Theorem directly in a cell problem, working with a given set of λ's but deliberately modifying the choices in some or all cells in a way which moves in the direction of the desired expenditure of resources By summing the deviations from maximum of the Lagrangian in each cell (epsilons) in which the strategies are so modified, a bound on the error of the result is obtained (which can be kept quite small in most cases by judicious choice of deviations). This appears to be a quite powerful strategem

Generalization of Lagrange Method

We conclude by mentioning a generalization of the Lagrange method that could prove useful for some problems Instead of a Lagrangian of the form $H(x) - \sum \lambda^k C^k(x)$, one can use a more general form

$$H(x) - G[C^1(x), \quad , C^n(x), \lambda^1, \quad , \lambda^n]$$

where G is any function satisfying

$$C_1^k \leqq C_2^k (\text{all } k) \rightarrow G[\{C_1^k\}, \{\lambda^k\}] \leqq G[\{C_2^k\}, \{\lambda^k\}]$$

for all $\{\lambda^k\}$, but is otherwise completely arbitrary Thus the only requirement on G is that it be monotone on the directed set of constraint vectors partially ordered by inclusion (one constraint vector *includes* a second if no component of the second exceeds the corresponding component of the first)

It is easily verified that the proof of Theorem 1 carries over to supply a proof for this general form of Lagrangian, and we can therefore conclude that an unconstrained maximum of a Lagrangian of this form is in fact a solution to the problem of optimizing the payoff subject to those constraints that are the resources expended by this solution.

This more general formulation is somewhat more powerful than the standard form for some classes of problems, since the solutions that can be found by it are no longer restricted to concave points in the PR space envelope It is, however, of little use in the solution of cell problems,

where the decisive decoupling of the problem into independent choices in each cell depends upon the linear form of the Lagrangian

EXAMPLE

AN INSTRUCTIVE example that illustrates a number of the points heretofore described is the application of Lagrange multiplier techniques to a problem of least-cost allocation of reliability investment presented previously in OPERATIONS RESEARCH[1] by JOHN D KETTELLE, JR , and solved there by a dynamic programming algorithm

The problem is to optimize the redundancy of an m-stage system, each stage of which consists of a number n_i of parallel (redundant) components of cost c_i and reliability (availability) a_i The separate stages are taken to be in series, so that the system is operable if, and only if, every stage contains at least one operable component The allocation problem is then to choose the stage redundancies (n_i's) in such a manner as to minimize the cost of achieving some stated system reliability (or alternately, to maximize the system reliability subject to constrained total cost)

The system reliability is given by

$$A = \prod_{i=1}^{i=m} [1-(1-a_i)^{n_i}]$$

Since maximizing the logarithm of a function maximizes the function, we take our payoff to be the log of the reliability

$$H = \ln A = \sum_{i=1}^{i=m} \ln[1-(1-a_i)^{n_i}],$$

and we have the problem formulated as a standard cell problem of the type previously discussed

In accord with the generalized Lagrange multiplier technique presented earlier, in order to produce an optimum solution for a given $\lambda \geqq 0$ we have only to maximize, independently for each stage (cell), the quantity

$$H_i'(n_i) = \ln[1-(1-a_i)^{n_i}] - \lambda\, c_i\, n_i$$

over the integers $n_i \geqq 1$ The allocation so produced is then guaranteed by the main theorem to be optimum for its cost

Since the functions $H_i'(n_i)$ are concave, they can be maximized by determining first analytically (by differentiation) which *real* value of n_i maximizes, then testing the integer on each side to find which *integer* maximizes Thus

$$dH_i'/dn_i = -(1-a_i)^{n_i}\ln(1-a_i)/[1-(1-a_i)^{n_i}] - \lambda c_i = 0$$

leads to the solution (real) for n_i

$$n_i = \ln\{1/[1-\ln(1-a_i)/\lambda c_i]\}/\ln(1-a_i)$$

This formula is then applied to each stage, the nearest integers (not less than one), $[n_i]$ and $[n_i+1]$, are tested to determine which maximizes $H_i'(n_i)$, and the payoffs and costs summed to produce an optimum solution

The entire procedure is repeated for a series of values of λ to produce a series of optimum solutions

Notice that this procedure leads to a direct calculation of the optimum number of components in each stage (for a given λ), which is independent of the actual numerical values and of the number of stages The amount of computation involved in producing a solution for a given λ is thus *linear* with the number of stages—a considerable advantage over other methods for large-scale problems involving many stages (As a test case,

TABLE I

Stage	Cost	Reliability
1	1 2	0 8
2	2 3	0 7
3	3 4	0 75
4	4 5	0 85

an artificial example of 10,000 stages, with randomly chosen component reliabilities and costs, was programmed for the CDC-1604 computer, leading to a computation time of about 40 seconds to produce the solution for each λ)

As previously mentioned, the advantages of the Lagrange multiplier method are most pronounced for large-scale cell problems, in which the importance of gaps is smallest, and in which it is most important that computation time be linear with the number of cells Consequently any small-scale numerical examples chosen cannot properly convey the value of the method, since for very small problems other methods are more competitive and the gaps between solutions produced by the Lagrange method can have much greater significance

With this observation in mind, it is nevertheless instructive to illustrate the concepts by studying in detail the structure of a small-scale numerical example To this end we shall treat the same numerical example treated in reference 1, which is a four-stage reliability allocation problem with component costs and reliabilities given by Table I

Application of the Lagrange multiplier method as previously developed for this problem for a series of values of λ produces the solutions shown in Table II

Inspection of these results shows that in all but one case the changes

in allocation from one solution to the next consist of at most one additional component in at most one stage It is therefore clear in this case, where payoff and cost are monotone with number of components, that there are no λ values which could produce new solutions between these solutions

TABLE II

λ	Cost	System unreliability	Allocation			
			Stage 1	Stage 2	Stage 3	Stage 4
0 0009	44 6	0 009997	5	5	4	3
0 0008	48 0	0 007086	5	5	5	3
0 0007	50 3	0 005392	5	6	5	3
0 0006	54 8	0 002530	5	6	5	4
0 0005	54 8	0 002530	5	6	5	4
0 0004	54 8	0 002530	5	6	5	4
0 0003	54 8	0 002530	5	6	5	4
0 0002	61 7	0 001033	6	7	6	4

However, the transition from λ=0 0003 to 0 0002 produced a change in three stages, and we can expect further solutions in this interval for intermediate λ values Additional exploration of this region indeed yields two new solutions, as given in Table III

TABLE III

λ	Cost	System unreliability	Allocation			
			Stage 1	Stage 2	Stage 3	Stage 4
0 000225	54 8	0 002530	5	6	5	4
0 000220	57 1	0 002020	5	7	5	4
0 000215	60 5	0 001288	5	7	6	4
0 000210	61 7	0 001033	6	7	6	4

Since there are no longer any changes by more than one component between successive solutions, the list of solutions that can be produced by Lagrange multipliers is complete for this range of cost (reasonably efficient procedures for automatic selection of successive λ values to produce all solutions that can be produced by this method are obviously easy to construct)

Let us now examine the complete structure of the problem to determine any solutions we may have overlooked, and what might be done to produce them

Figure 1 depicts a portion of the PR space for this problem An allocation is represented as a dot whose abscissa is the total cost of that allocation, and whose ordinate is the system reliability (payoff) produced by that allocation Every possible allocation that falls in the range of Fig 1 is represented in Fig 1 The points enclosed by squares are those optimum solutions produced by the Lagrange multiplier method (those that lie on the convex hull of the point set), and the circled points are the remaining optimum solutions. The solid stepped line joining both types of points is the complete optimum reliability vs investable cost function

The density of its solutions even in this small-scale problem is quite adequate for many applications, particularly preliminary explorative studies (It succeeded, in fact, in producing the particular solution sought in the original presentation of this example by Kettelle)

As mentioned earlier, the epsilon theorem can often be exploited to produce additional nearly optimal solutions by deliberately deviating slightly from maximizing the Lagrangian in each cell, and keeping track of the sum of the deviations (epsilon) An even more powerful strategem, which we shall employ now, is based upon generating these deviations in order of increasing epsilon In a discrete problem such as this, one can, within each cell, order the strategies in order of increasing deviation of the value of the Lagrangian from its maximum value One can furthermore construct simple algorithms to select, based on these ordered deviations within the cells, the sequence of combinations of deviations among the cells with the property that each successive combination has the smallest total deviation from the Lagrange solution excluding the preceding combinations Such a sequence has the property that every member, which is not dominated by a preceding member (in the sense of more payoff for equal or less cost), is itself an optimum solution, since it maximizes the over-all Lagrangian when the previous members of the sequence are removed

Application of this technique to the present problem produces the additional solutions shown in Table IV

In addition to the new solutions, Table IV gives the value of λ which produced that solution, and how far in the ordered sequence of deviations (ϵ-Depth) the solution was produced Note that all solutions in the range studied (including one, at a cost of 46 9, not reported by Kettelle) have been produced by examining at most the first six deviations from each of the solutions produced by the initial set of λ values

While generation of the ordered sequence of deviations becomes more difficult (time consuming) the farther one progresses, it is quite fast for the first members One might expect that for many applications the density of solutions would be adequately increased by extending such

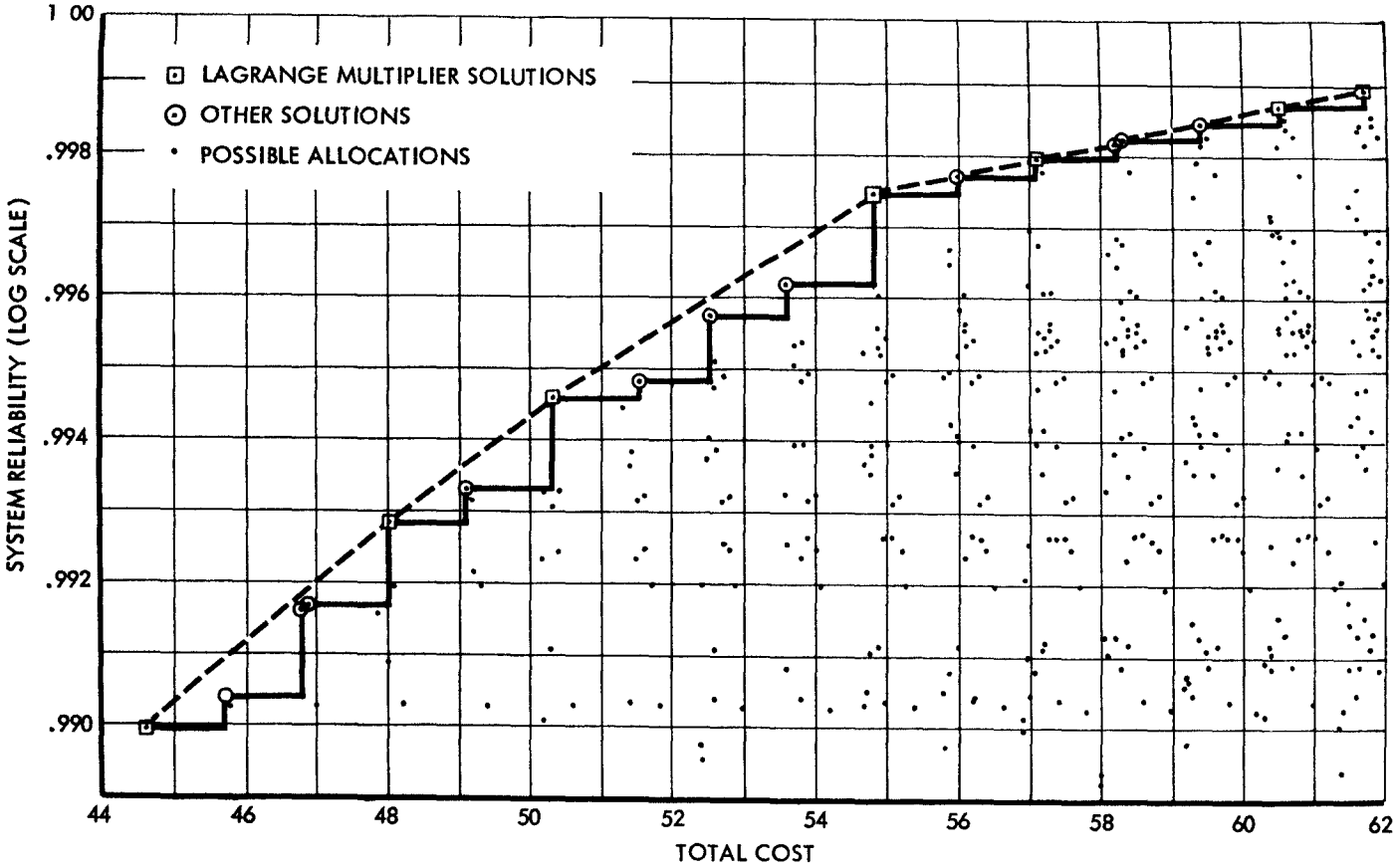

FIGURE 1 Reliability vs Cost for All Allocations

calculations only to the first several deviations when the basic Lagrange solutions were deemed of insufficient density

A general advantage of the techniques herein described lies in the fact that they allow one to approach a problem first with a simple technique (the basic Lagrange multiplier method) and then to produce additional solutions only when actually desirable or necessary (either by using a finer mesh of λ's, or if that fails by employing gap filling techniques), in contrast to techniques like dynamic programming that are 'all or nothing,' and by their nature produce a complete solution

TABLE IV

Cost	System unreliability	Allocation				λ	ε-Depth
		Stage 1	Stage 2	Stage 3	Stage 4		
45 7	0 009579	4	6	4	3	0 0009	6
46 8	0 008357	4	5	5	3	0 0009	3
46 9	0 008309	5	6	4	3	0 0009	4
49 1	0 006666	4	6	5	3	0 0007	4
51 5	0 005138	6	6	5	3	0 0006	6
52 5	0 004227	5	5	5	4	0 0006	2
53 6	0 003807	4	6	5	4	0 0006	5
56 0	0 002274	6	6	5	4	0 0004	1
58 2	0 001798	5	6	6	4	0 0004	3
58 3	0 001765	6	7	5	4	0 0002	3
59 4	0 001543	6	6	6	4	0 0002	2

SUMMARY

THE LAGRANGE multiplier method of solving constrained maximum problems has considerably greater power than is generally realized It is not limited to differentiable functions, but may often be profitably applied in situations involving maximization of any type of function over any set of strategies, discrete or continuous, numerical or nonnumerical, with constraints that can be represented as bounds on real valued functions over the same strategy set

The method is especially useful for solving problems of optimal allocation of a number of resources to a number of independent ventures where the total payoff is the sum of the payoffs that accrue from each venture (cell problem) In such a cell problem the Lagrange method reduces the problem to a series of independent unconstrained maximization problems, one for each cell

One example of such a problem that often arises in military operations research is the optimum allocation of given stocks of several weapons

types of differing characteristics to a diverse set of independent targets For such problems it is often crucial to account for the fact that weapons can be delivered only in integral numbers We have heard of at least one case where this problem has been attacked by nonlinear integer programming

The solution of this problem is, of course, quite simple and straightforward with the present technique of Lagrange multipliers In this case, the cells are the individual targets A strategy for a cell is an n-tuple of integers, one for each weapon type, representing the number of that type of weapon to allocate to the target The payoff in a cell is the expected destruction to the target of the given weapon allocation, and the resource functions are simply the numbers of weapons allocated themselves

The technique is not in general certain to produce solutions for all interesting constraint levels It is, however, 'fail-safe' in the sense that any solution that it does produce is a true optimum Furthermore, there are a number of additional techniques that often succeed in regions where the basic method fails

The Lagrange multiplier techniques presented here are particularly well suited to use with computers, where the sweeping (or trial-and-error variation) of the multipliers, as well as the maximizations within the individual cells of a cell problem, can be programmed to be rapidly and automatically executed

As was previously mentioned, the method has been employed in WSEG for several years for solving both production and military allocation problems, and has been quite successful

As a final note, the reader is cautioned against indiscriminately applying this method to min-max problems (where there are two sides allocating resources—an attacker and a defender, for example—with opposing interests) It is tempting in such cases to introduce multipliers for both sides and then carry out a min-max operation on the resulting Lagrangian, the analogy of the pure maximization case However, there is no analogue of Theorem 1 in the min-max case, and the procedure is *not* 'fail-safe,' but can and does in many instances produce erroneous results

Methods of handling the min-max case will be the subject of another paper

REFERENCE

1 John D Kettelle, Jr, "Least-Cost Allocations of Reliability Investment," *Opns Res* **10**, 249–265 (1962)

Copyright 1963, by INFORMS, all rights reserved. Copyright of Operations Research is the property of INFORMS: Institute for Operations Research and its content may not be copied or emailed to multiple sites or posted to a listserv without the copyright holder's express written permission. However, users may print, download, or email articles for individual use.