DOE/RA/50259--T2

DOE/RA/50259--T2

DE83 017033

A MATHEMATICAL MODEL OF A UTILITY FIRM

FINAL TECHNICAL REPORT,

on

Contract No. DE-AC01-80R50259, Entitled

INDUSTRY FUNCTIONAL MODELING

PART II A

Submitted to the

Office of Electric Energy Systems

US department of Energy

MASTER

by the

Polysystems Analysis Corporation

PO Box 846, Huntington, NY 11743

August 21, 1983

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

DISCLAIMER

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

DISCLAIMER

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

DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.

TABLE OF CONTENTS

PART I

	Page
1. INTRODUCTION	1
2. CHARACTERISTICS OF AN ORGANIZATION MEMBER	1
2.0 Overview	1
2.1 Primary Characteristics	7
2.11 Efficiency and Overload	7
2.12 Fallibility	12
2.13 Resistance to Change	19
2.1A Appendix A	24
2.14 Comments	30
2.2 Secondary Socio-Psychological Characteristics	32
2.21 Personal Preferences	32
2.22 Desire for Autonomy	36
2.2A Appendix A	44
2.2B Appendix B	45
2.23 Motivation	50
2.24 Comments	60
2.3 Information-Processing Characteristics	62
2.31 The Adaptive Information Channel	62
2.32 The Human Information Processor as an Adaptive Channel	77
2.33 Experimental Evidence	92
2.34 Comments	101
2.4 References	104

PART II

	Page
3. ORGANIZATIONAL DYNAMICS	1
3.0 Overview	1
3.1 Multilinear Programming	7

3.11	Primal Problems and Their Duals	7
3.12	Solutions "Classwise"	16
3.13	A "Curvilinear" Simplex Algorithm	27
3.14	Comments	36
3.2	Many-Person Symmetric Games	40
3.21	Problem and Solution Concepts	40
3.22	Solution Algorithm	49
3.23	Games and Multilinear Programs	55
3.24	Simultaneous Solution	73
3.25	Comments	80
3.3	Organizational Dynamics and Control	84
3.31	Control Options	84
3.32	Optimizing Control of Organizational Procedures	89
3.33	Allowance for Resistance to Change	103
3.34	Allowance for Desire for Autonomy	108
3.35	Comments	118
3.4	Planning Under Uncertainty	121
3.41	Uncertainty Modeling and the Planning Problem	121
3.42	Solution by Generalized Dynamic Programming ...	128
3.43	Stability and Robustness of Solution Algorithms	132
3.44	Application: Control of Resistance to Change	136
3.45	Computational Aspects and Supoptimal Procedures	139
3.46	Some Special Cases	143
3.47	Application: Utility Planning	148
3.48	Comments	157
3.4A	Appendix A	159
3.4B	Appendix B	163
3.4C	Appendix C	167

PART III

		Page
4.	DYNAMICS OF ECONOMIC SYSTEMS	1
4.0	Overview	1
4.1	Economic Equilibrium Theory - A Review	5
4.2	Discrete Economies	16

4.21	Exchange Economies	16
4.22	Production Economies	23
4.23	Approach to an Equilibrium	31
4.24	Comments	39
4.2A	Appendix A	43
4.2B	Appendix B	46
4.3	References	48
5.	PUBLICATIONS	1

3. ORGANIZATIONAL DYNAMICS

3.0 Overview

In an organization, one can expect two kinds of dynamical effects, those which are traceable to the characteristics of the members and those which are contributed by the management of change. The first kind is basically non-negotiable. The second, however, is largely a matter of organizational design.

In this project we identified one, and possibly two, of the non-negotiable variety. One is the phenomenon of resistance to change which is discussed in Sect. 2.13. The other is the motivation of individuals and that is reviewed in Sect. 2.22. We say that the latter is "possibly" one of the pertinent member characteristics because there is reason to believe that it can be partly or wholly treated by the same mathematics as the first one.

This chapter in any case deals with the second type of dynamic effect, namely the ones that are intentionally made part of the operation of an organization. In our opinion these effects may constitute the part of our work with the most promising applications to practice.

We have identified essentially only one such effect, namely that due to the processes by which the functioning of the organization is controlled. We discovered no others. The need for control, however, can have many causes. It arises whenever a divergence is detected between the actual and desired performances of an organization. Such

divergences may be due to an unexpected change in the environment in which an organization operates. They may also be due to the fact that the characteristics of the members may differ from anticipated ones, or that errors may have been committed in the original blueprint for the internal functioning of the organization, or any number of others.

If these factors were non-existent or perfectly predictable, control would be unnecessary. The operations of the organization could be pre-programmed to accommodate all events. It is the uncertainties in those events which establish the need for controls.

We have taken two approaches to the problem of organizational control. One is aimed at establishing procedures for what might be called "optimizing" control. It is desired to find a mode of operation with a performance that is optimal or at least satisfactory in the long run, as well as a procedure for attaining this mode. This approach is described in Sect.'s 3.1 to 3.3 of the report. The second approach is developed in Sect. 3.4. It leads to "optimal" control in the usual sense of the word. This is a control process in which not so much the performance in the long run is optimized but as transitional performance over a given period of time. We visualize the first approach to be appropriate to organizations which function in a more or less stationary environment. The second one is designed for situations in which plans must be drawn up for an effective

transition from one operation to another that is substantially different.

As mentioned above, control processes in organizations, and for that matter in any system, are intended to cope with uncertain events. The concept of uncertainty, therefore, pervades both of our approaches. The common practice in system theory, operations research, and related fields, is to equate uncertainty with probability. There are good arguments for this practice but also several good ones against it. In the work to be reported here, the arguments against seem the stronger ones, for several reasons. One reason is that the processes within an organization are often best modeled probabilistically to begin with. Representing the uncertainties in them by probabilities means working with probabilities of probabilities, a procedure that is frequently self-defeating. The uncertainty representation we use in what follows is of the kind we relied on in our work on the degree of a person's autonomy which we reported in Sect. 2.22. We in effect assumed there that an event that is uncertain is known only to belong to a certain set of possible events. This is also the interpretation to be given to it in this chapter. We realize that other interpretations (by fuzzy sets, for instance) are possible as well but we were unable to consider those in the time available to us.

Both of our approaches required much new ground to be broken, mathematically as well as conceptually. Optimizing

control is essentially an algorithm for the iterative (feedback) solution of a mathematical programming problem of a special kind which we call "multilinear." Sect. 3.1 develops a theory for such problems which appears to be a generalization of the well-known one for linear programming problems. It is, however, a generalization in a new, and so far unexplored, direction. The discussion in that section leads up to solution algorithm which is perhaps of general interest because it is an extension of the familiar Simplex method. It is, however, unattractive in an organizational setting because it is not iterative.

In order to derive an algorithm that is of the desired kind, a rather major conceptual detour had to be taken. A parallel had to be developed between multilinear programs and certain many-person games. An iterative algorithm for the solution could then be devised for such games and, through them, for multilinear programs as well. This work is reported in Sect. 3.2.

Sect. 3.3 finally reports on the applications of this algorithm to organization control problems. One application is to the control of the operational procedures on an output producing division (the "Line"), and the second to the problem of how to generate such procedure for the Staff of an Executive when the latter is conscious of his desire for autonomy. The last application is to an organization whose members offer resistance to change.

In our opinion, the mathematics we developed for these

control processes points to some applications of considerable potential in practice. Organizations seem to be fairly well designed, by and large. Several millennia of experience with them apparently have eliminated many mistakes and malfunctions. The control processes in them, however, seem to have received much less attention than they deserve, perhaps because the pace of change in organizational environments in the past has not been as rapid as today. Modern business and other organizations, however, need to have greater flexibility and adaptability. Their control mechanisms are the ones which must endow them with these properties, and our theory may point to ways in which they can do this in which case our work could lead to very useful practical procedures. We say more about this in Sect. 3.35.

Sect. 3.4 is perhaps the most important one, from the point of view of potential application to the utility industry, and more particularly to the problem that precipitated the idea for this and its companion projects. It is the problem of long-range planning under uncertainty. Our formulation starts from the assumption that the management of a utility firm has set targets which it hopes to achieve over a period of time, in particular perhaps over a period of major change in its operation. Such periods typically require the most carefully laid plans, yet present the largest uncertainties in their execution. These uncertainties are reflected in the targets. The targets are

taken to be sets which are to contain the variables that characterize the performance of the firm during the periods of interest. The object of the planning process is to manage the firm in such a way that the variables actually do lie in the targets or, if that is not possible, that they come as close to them as can be arranged.

The difficulty in this problem formulation arises from the uncertainties in the operation of the firm and in its environment. With the set-valued interpretation of these uncertainties, the firm is represented mathematically as an infinite-dimensional system with feedback control. It is shown in Sect. 3.34 that the planning problem is a generalized dynamic programming problem. However, because of its infinite dimensionality, the conventional solution procedures are not sufficient and special approximation techniques are needed. These too are discussed. Several applications are described, most importantly one of the planning for a utility firm.

3.1 MULTILINEAR PROGRAMMING

3.1.1 Primal Problems and Their Duals

Multilinear programming problems, in the sense in which the term is used in this section, are a special kind of nonlinear programming problems which appear to be natural and direct generalizations of the linear kind. In particular they possess duals which differ from the ones to which one is led from nonlinear programming theory but which are patent analogs of these linear counterparts. The main purpose of this section is the definition of a multilinear programming problem and of its new dual.

A mathematical programming problem will be called "multilinear" of degree D if it has the following three properties:

1. The objective function is a homogeneous polynomial of degree D , at most, in which the variables are raised to the powers of 0 or 1 , only.
2. The same is true of the non-constant terms of all constraints.
3. The objective function and/or the non-constant portions of at least one of the constraints is of degree D .

In such problems one can always partition the set of variables into $M \geq D$ classes, in such a way that the objective function and constraints are linear in the variables of one class when those in the others are fixed. In many applications in fact $M=D$ but it is possible in principle that there are as many classes as there are variables.

An example of a multilinear programming problem is the

following trilinear one. It involves three classes of non-negative variables x_i , y_j , and z_k whose subscripts range from 1 to n_x , n_y , n_z , respectively. The objective function

$$(3.1.1a) \quad s(x, y, z) = \sum c_{ij} x_i y_j + \sum c_{jk} y_j z_k$$

is of degree $S = 2$. It is to be minimized subject to the constraints

$$(3.1.1b) \quad \sum a_{ijk} x_i y_j z_k - a \geq 0, \quad \sum b_{ijk} x_i y_j z_k \geq b$$

$$(3.1.1c) \quad \sum a_i x_i - a_0 \geq 0, \quad \sum b_j y_j - b_0 \geq 0, \quad \sum c_k z_k - c_0 \geq 0$$

(Here and in what follows subscripts will be omitted on a summation sign when the sum ranges over all indices and all values of those indices. In this problem the number M of variables classes is the same as the degree D , $M = D = 3$. It will be used frequently below for illustrative purposes.

The dual to a multilinear programming problem can be defined in the way which is customary in nonlinear programming theory [1]. To do so, one first introduces the Lagrangian associated with it. If the problem is a minimization of an objective function, such as s in (3.1.1a), it is convenient to write all constraints in the form

$$(3.1.2) \quad \varphi_m(x, y, z, \dots) \leq 0$$

(which would mean that, in the example, all inequalities (3.11b) and (3.11c) would have to be multiplied by (-1)). The Lagrangian is then

$$(3.13) \quad L(x, y, z, \dots; \omega) = s(x, y, z, \dots) + \sum \omega_m \phi_m(x, y, z, \dots)$$

in which the ω_m are nonnegative Lagrange multipliers. The index m ranges over all constraints, and $x, y, z, \dots$ are vectors with the variables x_i, y_j, z_k , etc., in each of the classes s components. The dots after x, y, z indicate the possible presence of (a finite number of) additional variable classes. In the example (3.11) the Lagrangian is

$$(3.14) \quad \begin{aligned} L(x, y, z; \omega) = & \sum c_{jk} x_i y_j z_k - \omega_1 (\sum a_{ijk} x_i y_j z_k - a) \\ & - \omega_2 (\sum b_{ijk} x_i y_j z_k - b) - \omega_3 (\sum a_i x_i - a_0) \\ & - \omega_4 (\sum b_j y_j - b_0) - \omega_5 (\sum c_k z_k - c_0) \end{aligned}$$

The conventional dual to a multilinear, as to any other nonlinear, programming problem would then be the maximization of the objective function

$$(3.15) \quad \sigma_0(\omega) = \inf_{x, y, z, \dots} L(x, y, z, \dots; \omega)$$

in which the Lagrange multipliers ω_m are the variables. They are subject to no constraints other than their nonnegativity.

One can conclude immediately that

$$(3.16) \quad \sigma_0(\omega) \leq s(x, y, z, \dots)$$

because the sum in (3.13) is made up of nonnegative terms. One can in fact conclude further that if a set of values $\omega^*, x^*, y^*, y^*, \dots$ has somehow been discovered which achieves equality in (3.16), that set constitutes a solution to both the primal and dual problems. The maximum of $\sigma_0(\omega)$ evidently cannot exceed the minimum of $s(x, y, z, \dots)$, or vice versa.

The point of this discussion is that, for a multilinear programming problem, the dual can be defined in a way other than the conventional one. In that way, the objective function is simpler than $\sigma_0(\omega)$ in (3.15). It, too, is to be maximized with respect to the nonnegative Lagrange multipliers and it also satisfies an inequality such as (3.16). The maximization, however, is subject to constraints in addition to the non-negativity requirements of the variables ω .

This alternative dual is defined as follows. Let the degree of the objective function of the given problem be S and the highest degree among its constraints C . (The degree D of the problem is thus $D = \max(S, C)$.) The objective function of the new dual is then

$$(3.17a) \quad \sigma(\omega) = \frac{C}{S} \sum_{[C]} \omega_m \varphi_m(o) + \frac{C-1}{S} \sum_{[C-1]} \omega_m \varphi_m(o) + \dots$$

$$+ \frac{1}{S} \sum_{[1]} \omega_m \varphi_m(o)$$

in which $\varphi_m(o)$ are the constant terms in the constraints (3.12), and the summation $\sum_{[K]}$ ranges over all constraints of degree K . The maximization of this objective function is with respect to the nonnegative variables ω_m and subject to the constraints

$$(3.17b) \quad \frac{\partial L}{\partial x_i} \geq 0, \quad \frac{\partial L}{\partial y_j} \geq 0, \quad \frac{\partial L}{\partial z_k} \geq 0, \quad \dots$$

which must hold for all nonnegative $x_i, y_j, z_k, \dots$ obeying (2.2).

It may be useful to exhibit this dual for the special trilinear program defined by $C=3, S=2$ in that example, the dual objective function is

$$(3.18) \quad \sigma(\omega) = \frac{3}{2}(a\omega_1 + b\omega_2) + \frac{1}{2}(a_0\omega_3 + b_0\omega_4 + c_0\omega_5).$$

It is to be maximized with respect to the nonnegative variables $\omega_m, m=1,2,\dots,5$, subject to the constraints

$$(3.19) \quad \begin{aligned} \frac{\partial L}{\partial x_i} &= \sum_j c'_{ij} y_j + \sum_k c'''_{ik} z_k - \omega_1 \sum_j a_{ijk} y_j z_k - \omega_2 \sum_j b_{ijk} y_j z_k \\ &\quad - \omega_3 a_i \geq 0 \\ \frac{\partial L}{\partial y_j} &= \sum_i c'_{ij} x_i + \sum_k c''_{jk} z_k - \omega_1 \sum_i a_{ijk} x_i z_k - \omega_2 \sum_i b_{ijk} x_i z_k \\ &\quad - \omega_4 b_j \geq 0 \\ \frac{\partial L}{\partial z_k} &= \sum_j c''_{jk} y_j + \sum_i c'''_{ik} x_i - \omega_1 \sum_j a_{ijk} x_i y_j - \omega_2 \sum_j b_{ijk} x_i y_j \\ &\quad - \omega_5 c_k \geq 0 \end{aligned}$$

for all nonnegative x_i, y_j , and z_k obeying (3.11b) and (3.11c),

The relationship between a primal multilinear programming problem, and the dual defined above, is described in the

following theorem.

Theorem 3.11. Consider a multilinear programming problem of degree D in which an objective function $s(x, y, z, \dots)$ of degree S is to be minimized with respect to the nonnegative variables $x_i, y_j, z_i, \dots$ and subject to the constraints $\varphi_m(x, y, z, \dots) < 0$ of degrees C , or lower. A dual to the problem is the maximization of the objective function $\sigma(\omega)$ in (3.17a) with respect to the nonnegative variables $\omega_m, m = 1, 2, \dots$ and subject to the constraints (3.17b) which must hold for all nonnegative $x_i, y_j, z_k, \dots$ obeying (3.12). The feasible values of these objective functions, if any, are governed by the inequality

$$(3.110) \quad \sigma(\omega) \leq s(x, y, z, \dots).$$

If a set variables $x^*, y^*, z^*, \dots; \omega^*$ is known which is feasible for both problems and which achieves equality in (3.110) it is optimal for both.

Proof. Consider the expression $\sum x_i \partial L / \partial x_i$. It contains all terms of L which involved x to begin with. By analogy $\sum y_j \partial L / \partial y_j$ contains all terms of L which involved y , among those, however, some that contained x and y . The sum $(\sum x_i \partial L / \partial x_i + \sum y_j \partial L / \partial y_j)$ therefore will contain all terms from the objective function and the constraints which involve x or y , but those which involve both x and y will appear twice in this expression. The latter will, of course, be traceable either to constraints of degree 2 or higher, or to the objective function (unless it is linear). Continuing this argument leads to the following. The sum

$$\sum x_i \frac{\partial L}{\partial x_i} + \sum y_j \frac{\partial L}{\partial y_j} + \sum z_k \frac{\partial L}{\partial z_k} + \dots$$

contains all terms that appeared in the Lagrangian L , except the constant terms of the constraints, but it contains each term twice that was bilinear in the primal variables, each term three times that was trilinear in them, and so on. Thus,

$$\begin{aligned}
 & \sum_i x_i \frac{\partial L}{\partial x_i} + \sum_i y_i \frac{\partial L}{\partial y_i} + \sum_k z_k \frac{\partial L}{\partial z_k} + \dots \\
 (3.111) \quad &= S s(x, y, z, \dots) + C \sum [C] \omega_m [\varphi_m(x, y, z, \dots) - \varphi_m(o)] + \dots \\
 &+ 2 \sum [2] \omega_m [\varphi_m(x, y, z, \dots) - \varphi_m(o)] + \sum [1] \omega_m [\varphi_m(x, y, z, \dots) \\
 &- \varphi_m(o)].
 \end{aligned}$$

But the sum on the left-hand side of this equation is nonnegative, by (3.17b), and the φ_m are non-positive by (3.12).

Thus

$$\begin{aligned}
 & C \sum [C] \omega_m \varphi_m(o) + \dots + 2 \sum [2] \omega_m \varphi_m(o) + \sum [1] \omega_m \varphi_m(o) \\
 & \leq S s(s, y, z, \dots)
 \end{aligned}$$

which is (3.110) and hence proves the first statement of the theorem.

The second statement follows from the observation that since the maximum of σ cannot exceed the minimum of s , a set of variables $x^*, y^*, z^*, \dots; \omega^*$ which achieves equality in (3.110) is surely optimal. ¶

The formulation of a dual to a multilinear programming problem via eq.'s (3.17) may seem unusual and perhaps even somewhat awkward since it involves the primal variables, along with the dual ones, in many of its constraints. It may therefore be appropriate to point out that it is a direct and more natural

generalization of its linear counterpart than the conventional one of nonlinear programming. To see this, consider an example of a linear programming problem, namely one requiring the minimization of

$$s(x) = \sum c_i x_i$$

subject to, say, two constraints

$$\sum a_i x_i - a \geq 0, \quad \sum b_i x_i - b \geq 0.$$

The Lagrangian of this problem is

$$L(x; \omega) = \sum c_i x_i + \omega_1 (a - \sum a_i x_i) + \omega_2 (b - \sum b_i x_i).$$

The dual problem, formulated according to (3.17), is the maximization of

$$\sigma(\omega) = a\omega_1 + b\omega_2$$

with respect to non-negative ω_1, ω_2 and subject to

$$\frac{\partial L}{\partial x_i} = c_i - a_i \omega_1 - b_i \omega_2 \geq 0.$$

This is the dual to which is one led by the theory of linear programming (see, e.g., [2]), as opposed to the maximization of

$$\sigma_0(\omega) = \inf_x L(x; \omega)$$

which is the dual derived from nonlinear programming theory.

The theory of linear programming provides also certain results concerning the existence and feasibility of primal and dual solutions. This derivation, however, relies on the convexity of the feasible regions and on the existence of a

separating hyperplane. No such results are known for multilinear programs (see further comments in Sect. 3.14).

3.12 Solutions "Classwise"

A multilinear programming problem, in the sense in which the term is used here, has a property which was pointed out in the preceding section. It becomes a linear programming problem in the variables of one class if those in all other classes are fixed. This suggests the following question. Suppose that a set of variables $\hat{x}, \hat{y}, \hat{z}, \dots$ has been found which has the property that the solution to the linear programming problem for x that results from fixing $y=\hat{y}, z=\hat{z}, \dots$ actually is $x=\hat{x}$, and that the analogous is true of the others. Is it perhaps true that such a set is a solution to the multilinear programming problem? This question is to be the next subject of discussion. It develops that the answer to it is "no" in general but that it can be turned into a qualified yes if the solutions to the duals of those linear programs can be coordinated in a certain way.

These duals are formed according to the well-known rules of the theory of linear programming (see, e.g. [2]). If the original multilinear programming problem involves M variable classes, there will be M primal and M dual linear programs. Each of the latter will require the solution for a different set of variables $\lambda, \mu, \nu, \dots$. Suppose that λ is the set of variables in the problem that is dual to a primal one in which x is variable and $y=\hat{y}, z=\hat{z}, \dots$ are fixed. The solution $\hat{\lambda}$ to the dual problem will then be indexed by $\hat{y}, \hat{z}, \dots$, as a matter of course. It will comprise as many components $\hat{\lambda}_m$ as there are constraints in the primal linear program, ℓ_x say. Some of these components may be associated with constraints that were linear even in the original

multilinear problem. Others may be associated with bilinear constraints. Let λ_1 be one of these and let it be associated with a constraint $\varphi_1(x,y) \leq 0$ in the original multilinear program or, equivalently, with $\varphi_1(x,\hat{y}) \leq 0$ in the linear program for x . This constraint will then re-appear as $\varphi_1(\hat{x},y) \leq 0$ in the linear program for y and the dual to that will involve a variable, μ_1 say, which is associated with it. The desired coordination among the duals then is that the solutions $\hat{\lambda}_1$ and $\hat{\mu}_1$ be equal for such constraints.

More generally, if the original multilinear program involves a constraint of degree K , this constraint will enter into K of the linear programs derived from it and hence will be associated with K dual variables, one in each of the duals to those linear programs. The requirement is that the solution values of those variables be equal.

In order to shorten the subsequent discussion, some terminology will be helpful.

Definition 3.11. Let a multilinear program involve M variables classes, $x,y,z,\dots$. The linear program that results if e.g., the variables $y,z,\dots$ are fixed will be called a primal problem "of class x ," and its solution a "solution classwise" of the original multilinear problem. A set of values $\hat{x},\hat{y},\hat{z},\dots$ with the property that each is a solution classwise for one class of variables when the others are fixed at those values will be a "simultaneous solution (classwise)" of the original multilinear problem. An optimal solution of the problem, whenever a distinction is needed, will be called a "solution proper." The

designation "simultaneous (classwise)" will be applied also to the duals connected with the linear programs. There, however, it will refer to equality among the solution values of dual variables associated with the same multilinear constraint in the original problem.

These notions are readily illustrated with the example (3.1.2). The problem in that example involves $M=3$ variables classes. There are accordingly three primal linear problems P_x , P_y , P_z that can be formulated for it classwise, and three dual ones D_x , D_y , D_z . They are the following.

P_x : Minimize

$$(3.112a) \quad s_x(x, \hat{y}, \hat{z}) = \sum c'_{ij} x_i \hat{y}_j + \sum c''_{ik} x_i \hat{z}_k$$

subject to

$$(3.112b) \quad \sum a_{ijk} x_i \hat{y}_j \hat{z}_k - a \geq 0, \quad \sum b_{ijk} x_i \hat{y}_j \hat{z}_k - b \geq 0,$$

$$\sum a_i x_i - a_0 \geq 0$$

D_x : Maximize

$$(3.113a) \quad \sigma_x(\lambda) = \lambda_1 a + \lambda_2 b + \lambda_4 a_0$$

subject to

$$(3.113b) \quad \lambda_1 \sum_{jk} a_{ijk} \hat{y}_j \hat{z}_k + \lambda_2 \sum_{jk} b_{ijk} \hat{y}_j \hat{z}_k + \lambda_3 a_i \leq \sum_j c'_{ij} \hat{y}_j + \sum_k c''_{ik} \hat{z}_k$$

P_y : Minimize

$$(3.114a) \quad s_y(\hat{x}, y, \hat{z}) = \sum c'_{ij} \hat{x}_i y_j + \sum c''_{jk} y_j \hat{z}_k$$

subject to

$$(3.114b) \quad \sum a_{ijk} \hat{x}_i \hat{y}_j \hat{z}_k - a \geq 0, \quad \sum b_{ijk} \hat{x}_i \hat{y}_j \hat{z}_k - b \geq 0, \\ \sum b_j \hat{y}_j - b_0 \geq 0$$

D_Y : Maximize

$$(3.115a) \quad \sigma_Y(\mu) = \mu_1 a + \mu_2 b + \mu_4 b_0$$

(the reason for the subscript on μ_4 will become apparent presently) subject to

$$(3.115b) \quad \mu_1 \sum_{ik} a_{ijk} \hat{x}_i \hat{z}_k + \mu_2 \sum_{ik} b_{ijk} \hat{x}_i \hat{z}_k + \mu_4 b_j \leq \sum_i c'_{ij} \hat{x}_i \\ + \sum_k c''_{jk} \hat{z}_k$$

P_Z : Minimize

$$(3.116a) \quad s_Z(\hat{x}, \hat{y}, z) = \sum c''_{jk} \hat{y}_j \hat{z}_k + \sum c'''_{ik} \hat{x}_i \hat{z}_k$$

subject to

$$(3.116b) \quad \sum a_{ijk} \hat{x}_i \hat{y}_j \hat{z}_k - a \geq 0, \quad \sum b_{ijk} \hat{x}_i \hat{y}_j \hat{z}_k - b \geq 0, \quad \sum c_k \hat{z}_k - c_0 \geq 0$$

D_Z : Minimize

$$(3.117a) \quad \sigma_Z(v) = av_1 + bv_2 + c_0 v_5$$

subject to

$$(3.117b) \quad v_1 \sum_{ij} a_{ijk} \hat{x}_i \hat{y}_j + v_2 \sum_{ij} b_{ijk} \hat{x}_i \hat{y}_j + v_5 c_k \leq \sum_j c'_{jk} \hat{y}_j \\ + \sum_i c'''_{ik} \hat{x}_i$$

In the terminology of Definition 3.1, a solution of e.g., P_X is a solution classwise of the multilinear problem and a triplet $\hat{x}, \hat{y}, z$ that solves P_X, P_Y , and P_Z is a simultaneous solution classwise. It may or may not be a solution proper, however. A simultaneous solution for the duals D_X, D_Y and D_Z would require

that also two pairs of equations hold among their variables namely

$$\lambda_1 = \mu_1 = v_1, \quad \lambda_2 = \mu_2 = v_2,$$

the first pair equating the three dual variables associated with the first of the two trilinear constraints (3.11c) of the trilinear problem, and the second equating those of the other.

The objective of the remaining discussion in this section will be to develop relationships between the solutions classwise and the simultaneous solutions of a multilinear programming problem. It will be shown ultimately that a solution classwise that is simultaneous in both the primal and dual linear programs obtained from the original problem, is always a Kuhn-Tucker point [1].

Proposition 3.11. A necessary condition for a set of values for the variables to be a solution proper of a multilinear programming problem is that it be a simultaneous solution classwise.

Proof. Suppose that the solution proper were not optimal classwise. An improvement would then be possible in the value of the objective function by a change in the variables of at least one class, a fact which would contradict the proper optimality of the solution. ¶

Proposition 3.12. Among the solutions proper of a multilinear programming problem, if there are any, there is at least one that lies on the boundary of the feasible region.

Proof. Consider a solution $\hat{x}, \hat{y}, \hat{z}, \dots$ classwise first. The

solutions x for the variables of class x , when $y=\hat{y}$, $z=\hat{z}, \dots$, lie on the intersection between the feasible region of the multilinear problem and the hyperplane $y=\hat{y}$, $z=\hat{z}, \dots$. In fact those solutions that do not lie on the boundary of that intersection attain the same value of the objective function $s(x, \hat{y}, \hat{z}, \dots)$ as those that do. The boundary, however, is a subset of the boundary of the feasible set as a whole. By Proposition 3.1.1 the solution proper is among solutions classwise. Suppose it is not among those that lie on the boundary of the feasible set. In that case it would attain the same value of s as one that does lie there. This argument bears out Proposition 3.1.2.

The main result of this section is the following.

Theorem 3.1.2. A set of values for the variables of a multilinear programming problem which are simultaneous solutions classwise for both the primal and dual problems satisfy the Kuhn-Tucker conditions.

Proof. Let $x^*, y^*, z^*, \dots$ be the set of values produced by the primal simultaneous solution. Note first that it satisfies all of the primal constraints of the multilinear problem (in the way in which e.g., the triplet x, y, z satisfies (3.1.1b) and (3.1.1c)). It will now be shown that it satisfies the constraints (3.1.7b) of the original dual as well, at least when $x=x^*$, $y=y^*, \dots$. To this end observe first that the constraints of the dual problem of class x (such as (3.1.13b) in the example) are of precisely the same form as the constraints $\partial L / \partial x_i \geq 0$ in (3.1.7b), except that they involve the dual variables λ_m associated with the class x primal

problem instead of the variables ω_m . To avoid this mismatch, replace λ_m with ω_m . An analogous parallel exists between the constraints of the dual of class y (such as (3.115b) and $\partial L / \partial y_i \geq 0$ in (3.1.7b) except for the involvement of the duals variables μ_m . One can, however, set also $\mu_m = \omega_m$, as is readily seen from the following argument. Suppose μ_m is associated with a constraint $\varphi_m \leq 0$ of the original problem. It is then possible that φ_m did not involve any variables of class x to begin with. In this case, the constraint did not enter into the optimization classwise with respect to x , and no λ -variable was associated with it. Thus setting $\mu_m = \omega_m$ can introduce no conflict with earlier notational arrangements. On the other hand if φ_m does involve variables of class x , a dual variable λ_m was associated with it. The simultaneity of the dual solution, however, requires $\lambda_m = \mu_m$. The equation $\mu_m = \omega_m$ is thus again free of conflict. In this way, identity is readily established among all dual constraints, those of the multilinear problem itself as well as those in its treatments classwise.

The solution $x^*, y^*, z^*, \dots$, together with its dual $\lambda^*, \mu^*, \nu^*, \dots$, has thus been shown to satisfy all primal constraints of the original problem, and those of dual ones as well but possibly only when $x=x^*, y=y^*, z=z^*, \dots$ (i.e., not necessarily for all nonnegative $x, y, z, \dots$, as required). It will now be shown that it also satisfies

$$(3.118) \quad \sigma(\omega^*) = s(x^*, y^*, z^*, \dots).$$

This part of the proof is based on the fact that the problems solved classwise are linear programs and hence that the optimal values of the primal and dual objective functions for each such problem coincide. Thus if $s_x, s_y, s_z, \dots$ and $\sigma_x, \sigma_y, \sigma_z, \dots$ are these objective functions

$$(3.119) \quad \begin{aligned} s_x(s^*, y^*, z^*, \dots) &= \sigma_x(\omega^*), \quad s_y(x^*, y^*, z^*, \dots) \\ &= \sigma_y(\omega^*), \quad \dots \end{aligned}$$

Let the objective function $s(x, y, z, \dots)$ of the multilinear program be of degree S , and let $\sigma(\omega)$ be its dual counterpart. It will then be shown more specifically that

$$(3.120a) \quad \begin{aligned} s_x(x^*, y^*, z^*, \dots) + s_y(x^*, y^*, z^*, \dots) + \dots \\ = Ss(x^*, y^*, z^*, \dots) \end{aligned}$$

$$(3.120b) \quad \sigma_x(\omega^*) + \sigma_y(\omega^*) + \dots = S\sigma(\omega^*)$$

These relations, together with (3.1.19), then imply (3.1.18).

Since s is of degree S , every term in it is a product of S variables and hence appears in S of the objective functions $s_x, s_y, s_z, \dots$. Each such term is thus replicated S times in the sum of the left of (3.120a), a fact which confirms the right-hand side. Similarly, σ_x in (3.120b) contains the term $\omega_m \varphi_m(o)$ for every constraint $\varphi_m \leq 0$ in the original problem which involves x in the first place. If that constraint is of degree K , the same term will recur in $(K-1)$ other of the dual objective functions $\sigma_y, \sigma_z, \dots$. The sum on the left of (3.120b) will thus be

$$\begin{aligned} \sigma_x(\omega^*) + \sigma_y(\omega^*) + \dots = \\ = \sum [C] \omega_m \varphi_m(o) + (C-1) \sum [C-1] \omega_m \varphi_m(o) + \dots + \sum [1] \omega_m \varphi_m(o) \end{aligned}$$

which confirms the right-hand side as well.

It will now be shown that the Kuhn-Tucker conditions are satisfied at $x^*, y^*, z^*, \dots; w^*$. There are two of these. For one, the inequalities (3.18b) must be satisfied at that point, and this has just been shown to be so. For another,

$$\omega_m^* \varphi_m(x^*, y^*, z^*, \dots) = 0$$

must hold for all m . To see that this is so, note first that the left-hand side of (3.111) is non-negative at $x^*, y^*, \dots; \omega^*$. The same is thus true of the right-hand side, i.e.,

$$\begin{aligned} 0 \leq Ss(x^*, y^*, z^*, \dots) - S\sigma(\omega^*) + \\ + (\sum [C] \omega_m^* \varphi_m^* + \dots + 2 \sum [2] \omega_m^* \varphi_m^* + \sum [1] \omega_m^* \varphi_m^* \end{aligned}$$

where φ_m^* has been written for $\varphi_m(x^*, y^*, z^*, \dots)$. But the first two terms cancel, by (3.118). The remaining terms are non-positive because $\omega^* \geq 0, \varphi_m^* \leq 0$. It follows that their sum vanishes and, since they are all of the same sign, that each term vanishes individually. Thus, it has been proven that both Kuhn-Tucker conditions hold at $x^*, y^*, z^*, \dots; \omega^*$. ¶

The theorem, as formulated here, actually asserts a little less than it could. The assumptions assure not only that the Kuhn-Tucker conditions hold. They assure also that the points at which they hold are either local minima or local saddle points. Local maxima are incompatible with the minimization classwise.

The converse to the theorem, on the other hand, does not appear to be true. The fact, in other words, that the Kuhn-Tucker conditions hold at a point does not, by all indications, imply that it is a simultaneous solution classwise for a multilinear problem and its dual. The set of such solutions is thus apparently smaller in general for a given problem than the set of Kuhn-Tucker points. Under the circumstances, one can say the following.

Corollary 3.121. Let $x^*, y^*, z^*, \dots; \omega^*$ be a solution proper for a multilinear program and its dual, with objective functions $s(x, y, z, \dots)$ and $\sigma(\omega)$, respectively. It then also solves the primal and dual problems classwise simultaneously, and holds there. In fact the values of s and σ are the smallest among those attained by the solutions classwise.

Proof. According to Proposition 3.11, the values $x^*, y^*, z^*, \dots$ are a simultaneous solution classwise of the primal problems. Associating the solution values $\lambda^*, \mu^*, \nu^*, \dots$ of the duals classwise with the components of ω^* in the same way as in the proof of Theorem 3.12, establishes that ω^* is also a simultaneous solution of those duals. The first statement above is thus borne out. As in that proof, it now follows that (3.118), and hence the second statement, holds. The third finally is a consequence of the fact that a solution proper is among the Kuhn-Tucker points. It is, by its definition, the point with the smallest value of s , and hence also the smallest value of σ . ¶

The discussion in this section has assumed that a simultaneous solution of a multilinear programming problem is

known but it is not suggestive of any procedure by which it might be found. Some comments on the general characteristics of such a solution are made in Sect. 3.14.

3.13 A "CURVILINEAR" SIMPLEX ALGORITHM

Multilinear programming problems are, as has been mentioned above, rather natural generalizations of the linear ones. It is therefore of interest to inquire whether similarly natural generalizations exist to the well-known Simplex algorithm. As will be shown in this section, such algorithms do exist. However, while the simplex algorithm proceeds from one feasible solution to another and better one along certain straight-line search paths, the one to be developed here advances along certain curved paths which are generated as the solutions to differential equations. It will accordingly be referred to as a "curvilinear" Simplex algorithm. Its computational efficiency is not known but it is considered unlikely to be competitive with the general nonlinear programming algorithms which have profited from many years of thoughtful and painstaking development.

In order to avoid unhelpful notational complications, the idea of the curvilinear simplex algorithm will be explained using a variant of the example defined by (3.1.1). The extensions to multilinear programming problems of other kinds will no doubt be quite plain thereafter. In that example, all constraints were assumed in the form of inequalities. For the present purpose, it is more convenient for them to have been converted into equalities. It is readily recognized that such conversions are always possible by introducing appropriate slack variables and without destroying the multilinearity of the problem. It will be assumed here that these conversions have been carried out.

The problem to be solved is thus the minimization of

$$(3.1.19a) \quad s(x, y, z) = \sum c_{ij} x_i y_j + \sum c_{jk} y_j z_k + \sum c_{ik} x_i z_k$$

subject to the five equality constraints

$$(3.1.19b) \quad \sum a_{ijk} x_i y_j z_k = a, \quad \sum b_{ijk} x_j z_k = b$$

$$(3.1.19c) \quad \sum a_i x_i = a_0, \quad \sum b_j y_j = b_0, \quad \sum c_k z_k = c_0.$$

Let it be assumed to start with that a feasible solution is known for this problem, i.e., a set of values $x_i = x'_i$, $y_j = y'_j$, $z_k = z'_k$, which satisfy the constraints (3.1.19b) and 3.1.19c). In fact, let it be assumed that, in this solution, exactly five components (as many as there are constraints in the problem) are positive while all others are zero. In linear programming, such a solution is called a "basis," and the same term will be used here. Let the basis in this example consist of x'_1, x'_2, y'_1, y'_2 , and z'_1 . Denote the values of the objective function for this solution with s' . The question to be asked here then is whether there exist small departures $\delta x, \delta y, \delta z$ in the variables from x', y', z' which satisfy the constraints and which induce a reduction $\delta s < 0$ in the value of the objective function. If they do, the departures must satisfy five linear homogeneous equations which are the variations of the five constraints, and δs must satisfy a sixth one which is the variation of (3.1.19a). The coefficients of these equations depend on x', y', z' , of course. In fact they are the coefficients of the linear programs P_x, P_y, P_z classwise, defined by (3.1.12), (3.1.14) and (3.1.16). Let them be arranged into a table similar to a Simplex tableau, namely,

	δx	δy	δz	δs
(3.1.20)	A	B	C	o
	α	β	γ	1

In it, A is the matrix of the coefficients of δx in the variations of the constraints and α is the vector of coefficients of δx in the variation of (3.119a). The last column in the array are the right-hand sides of these variational equations, and the coefficient of δs .

The compound matrix $[A \ B \ C]$ in (3.120) has five rows in this problem, hence is of rank 5 at most. Assume that it is of rank 5 exactly and that the submatrix consisting of the columns corresponding to $\delta x_1, \delta x_2, \delta y_1, \delta y_2$, and δz_1 is among those that are invertible. One can then proceed as in the Simplex algorithm. That is, one can invert that submatrix and incidentally reduce to zero the elements in the last rows of those columns. The result will be a new array

$$(3.121) \quad \begin{array}{cccc} \delta x & \delta y & \delta z & \delta s \\ \hline A' & B' & C' & 0 \\ \alpha' & \beta' & \gamma' & 1 \end{array}$$

in which the matrices and vectors are of the form

$$A' = \begin{bmatrix} 1 & 0 & a'_{13} & \dots \\ 0 & 1 & a'_{23} & \dots \\ 0 & 0 & a'_{33} & \dots \\ 0 & 0 & a'_{43} & \dots \\ 0 & 0 & a'_{53} & \dots \end{bmatrix} \quad B' = \begin{bmatrix} 0 & 0 & b'_{13} & \dots \\ 0 & 0 & b'_{23} & \dots \\ 1 & 0 & b'_{33} & \dots \\ 0 & 1 & b'_{43} & \dots \\ 0 & 0 & b'_{53} & \dots \end{bmatrix} \quad C' = \begin{bmatrix} 0 & c'_{12} & c'_{13} & \dots \\ 0 & c'_{22} & c'_{23} & \dots \\ 0 & c'_{32} & c'_{33} & \dots \\ 0 & c'_{42} & c'_{43} & \dots \\ 1 & c'_{52} & c'_{53} & \dots \end{bmatrix}$$

$$\alpha' = [0 \ 0 \ \alpha'_3 \ \dots] \quad \beta' = [0 \ 0 \ \beta'_3 \ \dots] \quad \gamma' = [0 \ \gamma'_2 \ \gamma'_3 \ \dots]$$

Suppose now that, among the elements $\alpha'_3, \dots, \beta'_3, \dots, \gamma'_2, \gamma'_3, \dots$ there is one that is negative, γ'_2 for instance. In that case, one can increase z_2 from $z'_2 = 0$ by a small positive δz_2 , yet keep $\delta x_3 = \dots = \delta y_3 = \dots = \delta z_3 = \dots = 0$, and achieve

$$\delta s = \gamma_2 \delta z_2 < 0,$$

i.e., a reduction in the magnitude of the objective function.

There are many ways of realizing this. For instance, the equation

$$\delta z_2 = -\gamma_2' \delta \theta$$

where θ is a parameter, will do so. Such a parameter then allows a transition to a differential equation, namely

$$(3.122a) \quad dz_2/d\theta = -\gamma_2', \quad z_2(0) = 0.$$

The remaining variables and the value s of the objective function then obey

$$(3.122b) \quad \begin{aligned} dx_1/d\theta &= -c_{12}'\gamma_2', & dx_2/d\theta &= -c_{22}'\gamma_2', & dy_1/d\theta &= -c_{32}'\gamma_2' \\ dy_2/d\theta &= -c_{42}'\gamma_2', & dz_1/d\theta &= -c_{52}'\gamma_2', & ds/d\theta &= -(\gamma_2')^2 \\ x_3 &= \dots = y_3 = \dots = z_3 = \dots = 0, \end{aligned}$$

with the initial conditions

$$x_1(0)=x_1', \quad x_2(0)=x_2', \quad y_1(0)=y_1', \quad y_2(0)=y_2', \quad z_1(0)=z_1', \quad s(0)=s'.$$

Evidently, z_2 functions very much like a pivot variable in linear programming and the same piece of terminology will be used here.

In fact, if the programming problem is linear to begin with, hence the right-hand sides of the differential equations (3.122) constants, the integrations can be carried out explicitly and yield the straight-line searches of the conventional Simplex method.

In the multilinear programming problems under consideration the integration of (3.122) continues until either

- (i) one of the basis variables x_1, x_2, y_1, y_2, z_1 goes to zero, in which case the solution curve has reached the boundary of the feasible region, or
- (ii) γ_2' goes to zero in which case the reduction of the value of the objective function ceases.

The first of these two stop rules is analogous to linear programming. The pivot variable z_2 is now introduced into the basis, in place of the newly vanished variable. This is done by a Gauss reduction of the submatrix that includes the z_2 -column of $[A \ B \ C]$ but which omits the one corresponding to the newly vanished variable.

The second stop rule, however, has no analog in linear programming. When it halts the integration, say when $\theta = \theta_2$, it does so because the reduction in the value of the objective function by (3.122) has ceased. The variable z_2 is positive and can be thought of as having been added to the basis as a "redundant" basis variable.

It is possible that all non-vanishing coefficients in the last row of the array (3.121) are positive at this point. In such a case no further reduction is possible in the value of s by small variations. In other words, a local optimum has been reached, and the algorithm is stopped altogether.

Assume accordingly that one of the elements in the last row is negative, for instance the one in the z_3 -column, namely $\gamma_3(\theta_2)$. One can then change z_3 in the same way as z_2 in (3.122a),

i.e., according to

$$dz_3/d\theta = -\gamma'_3(\theta), \quad z_3(\theta_2) = 0. \quad (3.122b)$$

The remaining differential equations that are analogous to (3.122a) at this stage of the algorithm are, however, of a different form in general. The first of these, for example, is now

$$dx_1/d\theta + c'_{12} dz_2/d\theta = -c'_{13} \gamma'_3$$

The next four are analogous, and the last is

$$ds/d\theta = \gamma'_2 dz_2/d\theta - (\gamma'_3)^2$$

in which an allowance is made for the fact that $\gamma'_2(\theta) = 0$ for $\theta = \theta_2$. It will be noticed that these equations do not determine one of the variables of the problem. One of these can accordingly be chosen in any way which is consistent with its initial condition at $\theta = \theta_2$. The most convenient choice seems to be that of setting the variable z_2 equal to a constant, i.e.

$$z_2 = z_2(\theta_2), \quad dz_2/d\theta = 0.$$

For, under this arrangement, the differential equations above take the same form precisely as (3.122) and their integration can proceed in the same way.

The stop rules for this integration are the same as for (3.122), also, namely (i) and (ii) above. However, while the algorithm could be resumed at $\theta = \theta_2$ in only one of two ways, there is now a third as well. Let this latest stop take place at $\theta = \theta_3$. If it is due to rule (i) above, z_3 will now be used as a pivot variable and a new basis generated in this way. If the stop is due to rule (ii), in the present case more particularly due to the event $\gamma_3 = 0$, the variable z_3 will have been added to the basis as a redundant basis variable. Both of these cases are

thus analogs to those encountered previously at $\theta = \theta_2$.

The third alternative arises from the fact that, in the process of integrating the equations from θ_2 to θ_3 , $\gamma_2'(\theta)$ may have changed from $\gamma_2'(\theta_2) = 0$ to some other positive or negative value while z_2 , which was held constant, will still be positive at $\theta = \theta_3$. Suppose that $\gamma_2'(\theta_3)$ is negative. If it is, one can re-appoint z_2 as pivot variable, and decrease s by a further increase in z_2 according to (3.122). On the other hand, $\gamma_2'(\theta_3)$ may be positive. In that case it is a decrease in z_2 that will entail one in s , and since $z_2(\theta_2) > 0$ such a reduction is possible. It is then reasonable to make it again in accordance with the differential equation (3.122a), and to adjust the basis variables proper as in (3.122b). Thus, z_2 is used as pivot in either case. The variable z_3 on the other hand, being a redundant basis variable, is held constant and equal to $z_3(\theta_3)$ till the next stopping point.

This discussion implies that, in general, the basis of a curvilinear Simplex step with ℓ constraints will be made up of more than ℓ non-vanishing variables, ℓ among them basis variables "proper" and the rest redundant. The latter are held constant between stopping points and the former are varied according to differential equations of the form (3.122b). The assumption made earlier, therefore, according to which exactly ℓ non-vanishing basis variables must be known initially can be weakened. Any number greater or equal to ℓ is manageable by the algorithm, either at the start or later on.

The algorithm continues in the way described here until

- (a) the coefficients of the last line in the array or its analogous successors, are nonnegative, or
- (b) the right-hand side of one or more of the differential equations go to infinity in magnitude.

In the first case, as mentioned before, a local optimum has been attained and the algorithm comes to a stop. In the second case, if the divergence of a right-hand side is that of the differential equation of the pivot-variable, such as z_2 in (3.122a), the objective function is unbounded from below. This is evident from the last of eq.'s (3.122b). One can show that divergences in other equations are always accompanied by one in the pivot equation. In any such case, there is no natural stopping point for the algorithm, as described here.

The absence of a natural stopping point can arise in other ways as well. One is analogous to degeneracy in the conventional Simplex method, i.e., the vanishing of a (non-redundant) basis variable and subsequent cycling. Another is the approach of the variables to a local optimum which is, however, attained only in a limit as $\theta \rightarrow \infty$. This, of course, is a phenomenon that cannot arise in the linear Simplex method.

There is yet another phenomenon that may occur and that is peculiar to the curvilinear method. Suppose that at some point $\theta = \theta_r$ of its application to a problem with ℓ constraints, $m \geq \ell$ basis variables have been established. Any ℓ among these are then eligible as basis variables proper provided they meet one condition. The columns that correspond to these variables in a table such as (3.120) must form an invertible matrix. It is,

however, conceivable that no set of ℓ columns exists among the m corresponding to the basis variables which form such a matrix.

If that happens, the derivatives of the basis variables are linearly dependent. In the example above, if such a dependence held at $\theta=\theta_1$, one could set

$$k_1 dx_1/d\theta + k_2 dx_2/d\theta + k_3 dy_1/d\theta + k_4 dy_2/d\theta + k_5 dz_1/d\theta = 0.$$

there and be sure that not all of the k_i vanish. In such a case, however, one of the differential equations (3.1.22) would be inconsistent with the others in general and could not be enforced at $\theta=\theta_1$. In linear programming, such an inconsistency is not resolvable unless the constraints are linearly dependent to begin with. When they are not, it implies infeasibility. In multilinear programming on the other hand, it will typically be merely a local phenomenon, prevailing at $\theta=\theta_1$ and evaporating outside the immediate neighborhood of that point. The phenomenon would thus be roughly in the nature of numerical ill-conditioning and could presumably be treated in one of the several possible ways which can be resorted to in such cases.

A final comment concerns the derivation of an initial basic feasible solution. By present indications, the device used in linear programming can be relied on also in the multilinear counterpart. A problem with ℓ constraints is expanded by introducing ℓ "artificial" variables and then subjected to a series of Simplex steps in which these variables are removed from the basis, if at all possible.

3.14 COMMENTS

The results in the preceding sections were derived under a number of assumptions, some of which appear to be unnecessary, while others are suspected of being that. This section presents a discussion of these assumptions and their waivers.

(a) In homogeneous constraints and objective functions

At the very outset, a multilinear programming problem was defined as one whose objective function is homogeneous of a certain degree and whose constraints are similarly homogeneous, except for their constant terms. This restriction to homogeneity can be avoided by a simple artifice and it may therefore be superfluous altogether. Suppose that one of the constraints were inhomogeneous, namely

$$xy + x + y \geq 1 .$$

It could then be promptly "homogenized" by introducing a new variable z , and two inequality constraints, namely

$$xy + xz + yz \geq 1, \quad z \geq 1, \quad -z \geq -1 .$$

In this form it would satisfy the assumptions made above.

(b) Polynomial programming

Another assumption made in Sect. 3.11 ruled out powers other than zero and 1 for the programming variables. A second artifice readily extends this to other integer powers. Consider, for example, the constraint

$$x^2 + xy + y^2 \geq 1 .$$

It can be converted to

$$x_1 x_2 + x_1 y_1 + y_1 y_2 \geq 1 ,$$

$$x_1 - x_2 \geq 0, \quad -x_1 + x_2 \geq 0, \quad y_1 - y_2 \geq 0 ,$$

$$-y_1 + y_2 \geq 0$$

and thus conform with the assumption of Sect. 3.11. Some further extensions are fairly obvious. How far they can be carried, however, is not known.

(c) Duality theory

The duality theory developed in Sect. 3.11 seems a natural counterpart in many respects to that of linear programming. One might object to it on the argument that the dual problem, as defined there, contains the primal variables while neither the one in linear theory nor that in the conventional nonlinear theory does. There are counter-arguments, however. The fact that the primal variables do not enter the linear problem is due not so much to the nature of the dual but to its linearity, as the example at the end of Sect. 3.11 shows. The fact that they do not enter into formulation the conventional nonlinear dual problem on the other hand is actually a notational arrangement. The primal variables do not appear in the problem statement (3.115) explicitly because of the infimum operation there. A similar arrangement can be made in the dual formulated in Sect. 3.11 writing, instead of (3.117b),

$$\inf \frac{\partial L}{\partial x_i} \geq 0 , \quad \inf \frac{\partial L}{\partial y_3} \geq 0, \dots$$

with infimum taken over all primal variables obeying the primal

constraints.

The dual of Sect. 3.11, however, falls short of its linear counterpart in one important respect. There is no theorem which states that if the dual has no solution, then neither has the primal, and vice versa. In linear programming that statement relies on another theorem, namely one regarding the separating hyperplane between two convex sets (e.g., [31], p. 334). In multilinear programming, however, the feasible sets need not be convex. The line of proof used in linear programming therefore fails there and no substitute for it is known to the writer.

(d) Primal-Dual Algorithms

The notion developed in Sect. 3.12 of simultaneous solution classwise suggests the idea of an algorithm which solves the primal and dual of a multilinear programming problem. In fact it suggests reliance on the concept of a fixed point which has been used in nonlinear programming theory already [1]. An algorithm that lives up to these expectations appears to exist. It utilizes an equivalence between the programming problem on one hand and a certain many-person zero-sum game on the other. For linear programming problems, the equivalence reduces to the well-known one between such problems and two-person, zero-sum, symmetric games (e.g., [2], Chap. 13). The development of the algorithm is, however, quite lengthy. It is presented in Sect.'s 3.23 and 3.24 of this report.

(e) Computational Aspects

The computational efficiency of the curvilinear Simplex algorithm described in Sect. 3.13 is unknown. It is considered

unlikely, however, to be very high, in the form in which it is given there. The main source of inefficiency may be the computations of the expressions on the right-hand sides of the differential equations, such as (3.122), that need to be executed at every integration step. These expressions are ratios of determinants which are of high orders if the problem under study is of large scale. It is possible that simpler alternatives exist to the differential equations suggested in Sect. 3.13 or, if not, that efficient procedures can be developed for the computations of the determinants. Such lines of inquiry have not been pursued, however. Considering the advanced state of development of the algorithms for the solution of nonlinear programming problems in general, one can question whether a pursuit would be worthwhile. The answer is not necessarily negative, however. The solutions to differential equations such as (3.122) are after all characteristic of the geometry of the feasible region of a multilinear programming problem, in roughly the same way in which the straight line searches of the conventional Simplex method are characteristic of the feasible polytope of a linear problem. There is some circumstantial evidence that the geometry of the solution curves of equations like (3.122) can be highly complicated. This evidence may militate for solution algorithms that are more subtle, if less efficient, than those now in use for nonlinear programming problems in general.

3.2 MANY-PERSON SYMMETRIC GAMES

3.2.1 Problem and Solution Concepts

As mentioned in the Overview to this Section, a potential user of the solution algorithms for nonlinear programming problems is the control branch of an organization. The type of algorithm described in Sect. 3.13, however, is unlikely to have much of an appeal for such purposes. Control processes typically rely on solution algorithms that are realizable as feedback schemes, for various reasons. The reasons that are pertinent in organizational control will become apparent in Sect. 3.31. The Simplex-like algorithm developed in Sect. 3.13 does not seem to be amenable to execution by feedback. The purpose in Sect. 3.2 is to develop another algorithm that has the desired properties. It relies on an equivalence between multilinear programming problems with M variable classes and certain games with $(M + 1)$ players. In particular a solution classwise of such a programming problem is equivalent in many cases to the Nash equilibrium of the game. This section is devoted to the definition of the game and the derivation of some of its properties.

For notational convenience much of the discussion will be based on a four-person game. Extensions to more than four players are so simple, however, that the results can be stated for games with any number of players.

Let the four players be called F, G, H and K. Their payoffs, if they use the pure strategies t, u, v, w , resp., will be denoted with $f(t,u,v,w)$, $g(t,u,v,w)$, $h(t,u,v,w)$, and $k(t,u,v,w)$, and the components of their mixed strategy vectors p, q, r, s with p_t, q_u, r_v, s_w . The games to be considered here are of a kind to be called "symmetric, zero-sum" games.

The concept of a zero-sum game is well established (e.g., [3], p. 5). It is a game in which the gains of any subset of its players are borne by the losses of the others or, to be more specific, in which the payoffs add up to zero,

$$(3.21) \quad f(t,u,v,w) + g(t,u,v,w) + h(t,u,v,w) + k(t,u,v,w) = 0.$$

The concept of a symmetric game is well established only for two-person games. In such a game an interchange of the strategies of the two players entails the interchange of their payoffs. I.e.,

$$f(t,u) = g(u,t).$$

It is natural to adopt the same definition for a game with more than two players. For the four-person game considered here, it would imply

$$(3.22) \quad f(t,u,v,w) = g(u,t,v,w), \quad f(t,u,v,w) = h(v,u,t,w)$$

$$f(t,u,v,w) = k(w,u,v,t),$$

and this will in fact be the definition adopted here.

Admittedly there is a remote possibility of confusion since, in the theory of many-person games, symmetry is sometimes defined differently ([4], p. 163). The risk of misunderstanding however seems small and worth taking, for sake of terminological simplicity.

Eq. (3.22) leads to a useful observation, namely this.

Lemma 3.211. The payoff of one player in a symmetric game determines the payoffs of all others. It is invariant under the interchanges pairwise of the pure strategy indices of the other players. Thus, in a four person-game

$$(3.23a) \quad f(t, u, v, w) = f(t, v, u, w) = f(t, w, v, u) = f(t, u, w, v),$$

and analogously for g, h , and k . In the more general case of $(M+1)$ players in which the payoff of one of the players is $f(t, u, v, \dots w)$ this symmetry condition can be written

$$(3.23b) \quad f(t, u, v, \dots w) = f(t, v, u, \dots w) = \dots = f(t, w, v, \dots u)$$

Proof. The first statement of this lemma is merely a restatement in words of (3.22) (though without the restriction to four players). The second may be best argued by induction on the number of players. It is patently true for two players.

Assume it to be true for three, in which case the invariance for the payoffs is expressed by

$$f(t,u,v) = f(t,v,u), \quad g(t,u,v) = g(v,u,t), \quad h(t,u,v) = h(u,t,v).$$

It is then true also for a four-person symmetric game. To see this, assume w fixed for the moment. The pay-offs f, g, h then define a three-person game for which

$$f(t,u,v,w) = f(t,v,u,w),$$

bearing out the first of eq.'s (3.23a). Fixing similarly first u and then v bears out the other two. The argument used here for the transition from three players to four is clearly good also for the transition from M to $(M+1)$ and hence proves the lemma. ¶

One can now use (3.22) and (3.23a) to eliminate $g, h,$ and k from (3.21) which leads immediately to another conclusion.

Lemma 3.2.12. The payoffs of a player in a zero-sum symmetric game, which differ from each other only by interchanges pairwise of his strategies with those of the other players, add up to zero.

In the four-person game considered here this sum for e.g., the player F is

$$(3.24a) \quad f(t,u,v,w) + f(u,t,v,w) + f(v,u,t,w) + f(w,u,v,t) = 0.$$

For the other three players it follows from this via the symmetry relations (3.22). The analog to (3.24a) for such games with an arbitrary number of players can be written

$$(3.24b) \quad f(t,u,v,\dots w) + f(u,t,v,\dots w) + \dots + f(u,v,\dots t) = 0.$$

As is well-known, there are many solution concepts for games with more than two players. The one which matters here is the so-called Nash equilibrium. For a four-person game, such an equilibrium is defined by a quadruplet $\hat{p}, \hat{q}, \hat{r}, \hat{s}$ of mixed strategy vectors, with the property that no player can improve his payoff by a unilateral departure from his equilibrium strategy. For player F this means that

$$(3.25) \quad \sum f(t,u,v,w) p_t \hat{q}_u \hat{r}_v \hat{s}_w \leq \sum f(t,u,v,w) \hat{p}_t \hat{q}_u \hat{r}_v \hat{s}_w$$

using the same convention as before regarding summation signs without range indication.

It is well known (e.g. [3], p. 92-97) that every many-person game has at least one Nash equilibrium and that the inequalities (3.25) imply, and are implied by, the equivalent

$$(3.26) \quad \sum_{uvw} f(t, u, v, w) \hat{q}_u \hat{r}_v \hat{s}_w \leq \sum f(t, u, v, w) \hat{p}_t \hat{q}_u \hat{r}_v \hat{s}_w$$

These remarks about Nash equilibria apply to many-person games in general. For the zero-sum symmetric games treated here some additional ones can be made.

Theorem 3.21. A zero-sum symmetric game has equilibria at which the strategies are the same for all players and at which the mean payoffs are the same, namely zero, as well.

Proof. The proof utilizes a transformation employed by Nash towards establishing the existence of equilibria in general ([3], *ibid.*). Under it, the coordinates of F 's strategy vector in a four-person game are transformed according to

$$(3.27a) \quad p'_t = (p_t + \phi_t) / (1 + \sum \phi_t)$$

where ϕ_t is

$$(3.27b) \quad \phi_t = \sum_{uvw} f(t, u, v, w) q_u r_v s_w - \sum f(t, u, v, w) p_t q_u r_v s_w$$

whenever the quantity on the right is positive. Otherwise,

$$(3.27c) \quad \phi_t = 0.$$

The transformations of the components of q_u , r_v , and s_w are analogously constructed. Taken together, they are continuous and carry into itself a bounded set, namely the product of the four simplexes over which the mixed strategy vectors p, q, r, s are defined. They therefore have a fixed point $\hat{p}, \hat{q}, \hat{r}, \hat{s}$. Nash's proof shows further that (3.27c) holds there for all t and hence that all of the inequalities (3.26) are satisfied. In other words, $\hat{p}, \hat{q}, \hat{r}, \hat{s}$ is a quadruple of equilibrium strategies.

In the more special case of a zero-sum symmetric game, one notes first that the strategy vectors of the four players have the same number of components, for otherwise the interchangeability between strategies, required by the symmetry relations (3.22), would not be possible. One can thus choose a point where $p = q = r = s$ as pre-image of the transformation (3.22). In such a case, the gradients of the mean payoffs in (3.27b) are the same for all players and hence, so are the images p', q', r' and s' under the transformation (3.27). This transformation however evidently has a fixed point also and more specifically one at which $\hat{p} = \hat{q} = \hat{r} = \hat{s}$. The first statement of the theorem is accordingly proven for a four-person game.

To verify the second, assume that the four players employ identical strategies, not necessarily the equilibrium strategy $\hat{p}$. The zero-sum property (3.21) of the game then implies that the sum of the mean pay-offs of the four players vanishes because

$$\begin{aligned}
& \Sigma f(t,u,v,w) p_t p_u p_v p_w + \Sigma g(t,u,v,w) p_t p_u p_v p_w \\
& \quad + \Sigma h(t,u,v,w) p_t p_u p_v p_w + \Sigma k(t,u,v,w) p_t p_u p_v p_w \\
& = \Sigma [f(t,u,v,w) + g(t,u,v,w) + h(t,u,v,w) + k(t,u,v,w)] p_t p_u p_v p_w \\
& = 0.
\end{aligned}$$

According to the symmetry properties (3.22) however the four sums on the left are equal because, e.g.,

$$\begin{aligned}
\Sigma g(t,u,v,w) p_t p_u p_v p_w & = \Sigma f(u,t,v,w) p_t p_u p_v p_w \\
& = \Sigma f(t,u,v,w) p_t p_u p_v p_w
\end{aligned}$$

where the last equation is a consequence of a mere re-indexing of the summation. Thus,

$$4 \Sigma f(t,u,v,w) p_t p_u p_v p_w = 0$$

which proves the second statement of the theorem for a four-person game. Moreover, since the line of reasoning does not rely on how many players are involved in the game, the proofs of both statements are good for any number of players. ¶

It is not known whether all Nash equilibria of a zero-sum symmetric game have the properties described in this theorem but for the purposes of this paper it is only necessary to know that at least some have them. The conditions which characterize it can then be simplified over the more general (3.26).

Corollary 3.2.11. At least one of the Nash equilibria of a symmetric zero-sum four-person game is characterized by

$$(3.28) \quad \sum_{uvw} f(t, u, v, w) \hat{p}_u \hat{p}_v \hat{p}_w \leq 0.$$

3.22 Solution Algorithm

The games required in the sequel are somewhat more specialized yet than those discussed so far. It is accordingly of interest that a computational procedure exists for such games by which their equilibria can be determined (at least in principle). It utilizes a minor modification of Nash's transformation (3.27) which was, in a way, anticipated by Brown and von Neumann [5] in their proposal for a solution algorithm for two-person zero-sum symmetric games. In the present context this modification merely replaces the quantity ϕ_t in the transformation (3.27) with $\rho\phi_t$ where $0 < \rho < 1$. In other words (3.27a) is replaced with

$$(3.29a) \quad p_t' = (p_t + \rho\phi_t)/(1 + \rho\Sigma\phi_t)$$

where $0 < \rho < 1$. The expression for ϕ_t , for a four-person game, can now be written (utilizing (3.28))

$$(3.29b) \quad \phi_t = \Sigma_{u,v,w} f(t,u,v,w) p_u p_v p_w$$

whenever the right-hand side is positive, and

$$(3.29c) \quad \phi_t = 0$$

otherwise. If the game has an arbitrary number of players,

(3.29b) reads

$$(3.2.9d) \quad \phi_t = \sum f(t, u, v, \dots w) p_u p_v \dots p_w$$

in which the sum ranges over all indices of f except t .

The types of game which can be solved, using this transformation, are symmetric and zero-sum but they have the additional property that in the zero-sum conditions only two payoffs, at most, differ from zero. In the condition (3.2.4a) for a four-person game, these payoffs for player F might be the first and the last, in which case (3.2.4a) would be specialized to

$$f(t, u, v, w) + f(w, u, v, t) = 0,$$

(3.2.10a)

$$f(u, t, v, w) = f(v, u, t, w) = 0$$

For a game with an arbitrary number of players the conditions would be

$$f(t, u, v, \dots w) = -f(w, u, v, \dots t)$$

(3.2.10b)

$$f(u, t, v, \dots w) = f(v, u, t, \dots w) = \dots = 0.$$

Under such an additional restriction one can say the following.

Theorem 3.2.2. Consider an $(M+1)$ -person, zero-sum, symmetric game with the additional property that, in the zero-sum conditions for the pay-offs of any player two terms at most differ from zero. These payoffs are indexed by the same pair of indices for every $(M+1)$ -tuple of pure strategies. The transformation

(3.2.9), applied iteratively to any strategy p that is the same for all players, converges on an equilibrium strategy, provided only ρ is chosen small enough. The equilibrium strategy is the same for all players, and so is the mean payoff which is zero for all.

Proof. The proof is again carried out for a four-person game. Moreover the special assumption of this theorem, regarding the player's payoffs, will be used in the form (3.2.10) for player F. Those for the others then follow from it via the symmetry relations (3.22).

The proof begins with the demonstration that the quantity

$$(3.2.11) \quad V(\phi) = \sum \phi_t^2$$

is a Lyapunov function. It then follows from the properties of such functions that the repeated application of (3.2.9) will induce the convergence of all ϕ_t to zero. Moreover, since

$$\sum (p'_t - p_t)^2 = [\rho^2 \sum \phi_t^2 (1-p_t)^2] / (1+\rho \sum \phi_t)^2$$

approaches zero, as the ϕ_t do, such an application generates a sequence of vectors p which converges to a limit $\hat{p}$ at which

$$\hat{p}' = \hat{p},$$

i.e., which is a fixed point of the transformation (3.2.9).

The demonstration that V in (3.2.11) is indeed a Lyapunov function requires the proof of three statements [6], namely

(i) $V(0) = 0$ and $V(\phi) > 0$ for $\phi \neq 0$

(ii) $V(\phi) - V(\phi) < 0$ for $\phi \neq 0$

(iii) $V(\phi)$ is continuous in ϕ and $V(\phi) \rightarrow \infty$ as $\|\phi\| \rightarrow \infty$

Statement (i) is clearly correct and so is statement (iii) since $\|\phi\|$ can be taken the same as $V(\phi)$. Thus only (ii) needs proving. To that end, note that

$$\begin{aligned}
 (3.2.12) \quad \Sigma \phi_t'^2 - \Sigma \phi_t^2 &= 2\rho \Sigma \phi_t \left. \frac{\partial \phi_t}{\partial \rho} \right|_{\rho=0} + o(\rho) \\
 &= 2\rho \Sigma_t \phi_t \Sigma_{uvw} f(t, u, v, w) \left[\frac{\partial p_u'}{\partial \rho} p_u p_w + \frac{\partial p_v'}{\partial \rho} p_u p_w \right. \\
 &\quad \left. + \frac{\partial p_w}{\partial \rho} p_u p_v \right]_{\rho=0} + o(\rho).
 \end{aligned}$$

The terms corresponding to values of t for which $\phi_t = 0$ vanish in the sum over t . Those in which $\phi_t > 0$ involve three sums, among them

$$\begin{aligned}
 (3.2.13) \quad &2 \Sigma_t \phi_t \Sigma_{uvw} f(t, u, v, w) \left. \frac{\partial p_u}{\partial \rho} p_v p_w \right|_{\rho=0} \\
 &= 2 \Sigma f(t, u, u, w) \phi_t \phi_u p_v p_w - 2 \phi \Sigma_t \phi_t \Sigma_{uvw} f(t, u, v, w) p_u p_v p_w \\
 &= 2 \Sigma f(t, u, v, w) \phi_t \phi_w p_u p_v - 2 \rho \Sigma \phi_t^2
 \end{aligned}$$

where the last equation holds because of the symmetry of f under an exchange of t and w . The other two sums on the right

of (3.2.12) can be treated similarly and lead to the same expression. Thus,

$$\begin{aligned}\Sigma \phi_t'^2 - \Sigma \phi_t^2 &= 6\rho \Sigma f(t,u,v,w) \phi_t \phi_w p_u p_v - 6\rho \phi \Sigma \phi_t^2 + o(\rho) \\ &= 6\rho \Sigma_{uv} p_u p_v [\Sigma_{tw} f(t,u,v,w) \phi_t \phi_w] - 6\rho \phi \Sigma \phi_t^2 + o(\rho)\end{aligned}$$

Under assumption (3.2.10) of this theorem however, $f(t,u,v,w)$ is a skew-symmetric matrix in the pair of indices (t,w) , for every pair (u,v) . The sum over t and w in the bracket therefore vanishes and

$$(3.2.14) \quad \Sigma \phi_t'^2 - \Sigma \phi_t^2 = -6\rho \phi \Sigma \phi_t^2 + o(\rho)$$

which is negative if ρ is small enough unless all ϕ_t vanish. Thus, $V(\phi)$ is indeed a Lyapunov function for the transformation (3.2.9), and the strategy vector p converges on a $\hat{p}$ at which $\phi_t = 0$ for all t . Nash's existence proof, as summarized in the proof of Theorem 3.2.1, now assures that $\hat{p}$ is an equilibrium strategy, and Theorem 3.2.1 assures further than this strategy, as well as the mean payoff, is the same for all players. The latter is in fact zero.

The assertions of the theorem have thus been verified but under two assumptions. The first of these is that the two non-vanishing payoffs in each zero-sum condition are those shown in (3.2.10a). It is however easy to convince oneself that any other pair could have been chosen without invalidating the line of reasoning used in the proof so far.

A similar comment applies to the second assumption which is that the game under consideration is a four-person game. If there are $(M+1)$ players, instead of four, the bracket on the right-hand side of (3.212) contains M terms, rather than three. As a consequence, the factor of 6 in (3.214) is replaced with $2M$. This however does not change the final conclusion.¶

The upshot of this theorem is that, under certain assumptions, the transformation (3.2.9) represents a solution algorithm for many-person games. In the next section it will be shown that it is often also an algorithm for the solution classwise of multilinear programming problems.

3.23 GAMES AND MULTILINEAR PROGRAMS

The point of introducing the concept of a zero-sum symmetric game, as just mentioned, is that there exists a parallel between it and that of a multilinear programming problem. A solution procedure for one, such as (3.29) for a game, can thus be expected to be good also for the other. This section is to show this to be frequently true.

It will more particularly be shown that, given a multilinear program with M variable classes, a game with $(M+1)$ players can be constructed which satisfies all of the assumptions of Theorem 3.22 and whose Nash equilibrium strategy is frequently equivalent to a solution of the programming problem. The equivalence is most direct with the simultaneous solution classwise primal as well as dual. With certain modifications to be discussed in Sect. 3.24 it extends to the solution proper.

The main idea of this equivalence is due to Gale, Kuhn, and Tucker [7]. In the form in which it applies here it is the following. One lists (in a certain order to be described below) all constraints of the M linear programs classwise, primal as well as dual, that are associated with the given multilinear problem. Let the result be a list of N inequalities. One continues it by entering also the inequalities between the primal and dual objective functions of those linear programs (though, for a reason to be explained below, with the directions of

of these inequalities reversed). If the multilinear problem involves M variable classes, there will be M of these inequalities, and hence a total of $(N+M)$ on the complete list. They will involve all of the variables of the original problem, as well as those of the dual problems classwise.

These variables are associated with the components of the mixed strategy of one of the players in a zero-sum symmetric game. The game solution that is of interest is the Nash equilibrium. There is thus no loss of generality in assuming that all players use the same strategy and to consider only that. Let this be the strategy p of player F . The association between strategy components and programming variables is then the following. If the number of primal variables in classes $x, y, z, \dots$ is $n_x, n_y, n_z, \dots$, resp., with $n_x + n_y + n_z + \dots = n$, one sets

$$(3.215a) \quad x_i = \frac{p_u}{p_{N+1}}, \quad y_j = \frac{p_u}{p_{N+2}}, \quad z_k = \frac{p_u}{p_{N+3}}, \quad \dots$$

where u ranges lexicographically from 1 to n_x in the first equation, from (n_x+1) on to (n_x+n_y) in the second, and so on for all M classes. The components in the denominators are correspondingly $p_{N+1}, p_{N+2}, \dots, p_{N+M}$. Similarly, if the number of dual variables classwise $\lambda, \mu, \nu, \dots$ is $\ell_x, \ell_y, \ell_z, \dots$, resp., with $\ell_x + \ell_y + \ell_z + \dots = \ell$, one sets

$$(3.215b) \quad \lambda_i = \frac{p_u}{p_{N+1}}, \quad \mu_j = \frac{p_u}{p_{N+2}}, \quad \nu_k = \frac{p_u}{p_{N+3}}, \quad \dots$$

where u in the first equation ranges from $(n+1)$ to $(n+\ell_x)$, and

so on. Since the number of variables and the number of constraints is the same in any linear program, primal as well as dual, the same is true of the linear programs classwise discussed here. Thus

$$n + \ell = N$$

which implies further that the number of components p_u of p is $(N+M)$, i.e., the same as the number of inequalities entered into the list described above.

One can now replace the programming variables in all of these inequalities with the components p_u of the game strategy p , in accordance with (3.215). Multiplying both sides by the product $(p_{N+1} p_{N+2} \dots p_{N+M})$ will then remove all fractions. In fact it will convert the inequalities into others which are again multilinear, though now in the components of p , and homogeneous (i.e., without constant terms). For the case of $M = 3$ for instance they will be of the form

$$(3.216) \quad \sum_{uvw} f(t, u, v, w) p_u p_v p_w \leq 0$$

where t is an index denoting the place of the inequality on the list and f is the coefficient in it of the product $(p_u p_v p_w)$. A problem with M classes would of course lead to expressions with M factors $(p_u p_v p_w \dots)$ and to coefficients f depending on $(M+1)$ indices.

The inequalities (3.216), it will be noticed, are of precisely the same form as those in (3.28) which define the Nash equilibrium

of a symmetric zero-sum four-person game. Under certain conditions therefore, if such an equilibrium $\hat{p}$ has been found a simultaneous solution classwise can be derived from the original programming problem simply by calculating the corresponding values $\hat{x}, \hat{y}, \hat{z}, \dots$ of the problem variables from $\hat{p}$ by (3.2.15a). The conditions under which such a calculation is possible are the following:

- a) The components $\hat{p}_{N+1}, \hat{p}_{N+2}, \dots, \hat{p}_{N+M}$ of the equilibrium strategy must not vanish. This will be made an assumption in what follows.
- b) The list of inequalities of the original programming problem must permit an arrangement in which the coefficients of these inequalities form an $(N+M)$ -dimensional array of payoffs, with the properties required by a symmetric zero-sum game. In fact if at all possible, the array should have the properties required by Theorem 3.22 for, in that case, the solution procedure described there is applicable. The demonstration that such an arrangement is always possible is the main remaining subject of this section.

However before proceeding to that subject, it may be useful to exhibit the list of inequalities, after their conversion to the components of the strategy p , for the example (3.11). The list will in fact be developed in the arrangement needed for an equivalence with a symmetric zero-sum game.

The example (3.11), it will be recalled, involves $M = 3$ variables classes x, y, z . The equivalent game therefore is a four-person game. The list contains all of the constraints of the primal and dual problems classwise, as well as the inequalities between the three objective functions of these programs, but with the directions reversed. It starts with the constraints of the dual programs which are (3.113b) (3.115b) and (3.117b). Replacing the programming variables x_i, y_j, z_k with the components p_u of a game strategy vector, as specified by (3.215), gives

$$\begin{aligned}
 & \sum_{vw} a_{ijk} p_v p_w p_{n+1} + \sum_{vw} b_{ijk} p_v p_w p_{n+2} + a_i p_{n+3} p_{N+2} p_{N+3} \\
 & - \sum_v c_{ij} p_v p_{N+1} p_{N+3} - \sum_w c''_{ik} p_w p_{N+1} p_{N+2} \leq 0 \\
 (3.217) \quad & \sum_{uw} a_{ijk} p_u p_w p_{n+4} + \sum_{uw} b_{ijk} p_u p_w p_{n+5} + b_j p_{n+6} p_{N+1} p_{N+3} \\
 & - \sum_u c'_{ij} p_u p_{N+2} p_{N+3} - \sum_w c''_{ik} p_w p_{N+1} p_{N+2} \leq 0 \\
 & \sum_{uv} a_{ijk} p_u p_v p_{n+7} + \sum_{uv} b_{ijk} p_u p_v p_{n+8} + c_k p_{n+9} p_{N+1} p_{N+2} \\
 & - \sum_v c''_{jk} p_v p_{N+1} p_{N+3} - \sum_u c'''_{ik} p_u p_{N+2} p_{N+3}
 \end{aligned}$$

where $u = i$, $v = n_x + j$, $w = n_x + n_y + k$. The first of these inequalities will be entered in the list at location $t = i$, the second at $t = n_x + j$, and the third at $t = n_x + n_y + k$. The total number of entries, at this point is $n_x + n_y + n_z = n$.

The entries that follow are those derived from the constraints of the primal problems classwise, i.e., from (3.112b),

(3.114b), (3.116b). They are listed in that same order, i.e., the three inequalities obtained from

$$\begin{aligned}
 & \sum_{uvw} a_{ijk} p_u p_v p_w - a_{N+1} p_{N+2} p_{N+3} \leq 0 \\
 (3.218) \quad & \sum_{uvw} b_{ijk} p_u p_v p_w - b_{N+1} p_{N+2} p_{N+3} \leq 0 \\
 & \sum_u a_i p_u p_{N+2} p_{N+3} - a_o p_{N+1} p_{N+2} p_{N+3} \leq 0
 \end{aligned}$$

The ranges of the indices u , v , and w are the same as those in (3.217). The next entries are the inequalities derived from those primal constraints of class y , namely (3.114b). Two of these, namely those obtained from the trilinear constraints are identical to the first two inequalities (3.218). This is true generally: in the list of the inequalities for a multilinear problem, an inequality of degree K occurs K times.) The linear constraint in (3.114b) however, leads to an entry without a precedent, namely

$$\sum_v b_{vj} p_v p_{N+1} p_{N+3} - b_o p_{N+1} p_{N+2} p_{N+3} \leq 0.$$

The analogous comments apply to the entries derived from (3.116b). The first two are duplicates of (3.218a) and (3.218b) above, and the third is

$$\sum_w c_{wk} p_w p_{N+1} p_{N+2} - c_o p_{N+1} p_{N+2} p_{N+3} \leq 0.$$

At this point the list contains

$$N = n_x + n_y + n_z + \ell_x + \ell_y + \ell_z = n + 9$$

entries. Three more must be made, namely those derived from the inequalities between the objective functions of the linear problems classwise. These inequalities are

$$(3.219) \quad s_x \geq \sigma_x, \quad s_y \geq \sigma_y, \quad s_z \geq \sigma_z.$$

To obtain the inequalities for the list, however, these are first reversed. The programming variables are then replaced with the strategy components, leading to

$$\begin{aligned} & \sum_{uv} c'_{ij} p_u p_v p_{N+3} + \sum_{uw} c'''_{ik} p_u p_w p_{N+2} - \\ & - a p_{n+1} p_{N+2} p_{N+3} - b p_{n+2} p_{N+2} p_{N+3} - a_o p_{n+3} p_{N+2} p_{N+3} \leq 0 \\ (3.220) \quad & \sum_{uv} c'_{ij} p_u p_v p_{N+3} + \sum_{uw} c''_{jk} p_v p_w p_{N+1} - \\ & - a p_{n+4} p_{N+1} p_{N+3} - b p_{n+5} p_{N+1} p_{N+3} - b_o p_{n+6} p_{N+1} p_{N+3} \leq 0 \\ & \sum_{vw} c''_{jk} p_v p_w p_{N+1} + \sum_{uw} c'''_{ik} p_u p_w p_{N+2} - \\ & - a p_{n+7} p_{N+1} p_{N+2} - b p_{n+8} p_{N+1} p_{N+2} - c_o p_{n+9} p_{N+1} p_{N+2} \leq 0 \end{aligned}$$

in which the ranges of u , v , and w are the same as in (3.217).

The list now contains $(N+3)$ entries, and it is complete for this example.

The reason for enforcing inequalities between the objective functions classwise which are the opposite of those in (3.219), and which are valid always, is the following. If a set of variables is found which obeys, e.g., $s_x \geq \sigma_x$, along with all the other constraints, one can conclude from this and (3.219a) that $s_x = \sigma_x$ and hence that a solution of class x has been

found. If a set of variables has been discovered which obeys $s_x \geq \sigma_x, s_y \geq \sigma_y, \dots$, along with all constraints, one can conclude further that it is a simultaneous solution classwise of the primal problem.

The procedure for setting up the list of inequalities for multilinear programs in general proceeds is conveniently defined as a sequence of ten steps. Their description is made somewhat simpler by the following piece of nomenclature.

Definition 3.21. A component p_u of the strategy vector p associated by (3.215) with a variable of class x , primal or dual, will be called a "p-component of class x , primal or dual." The component p_{N+1} will be called the "last" of class x . Components associated with other variables classes are analogously designated.

In (3.215a), the components p_u for u from 1 to n_x are primal components of class x , and (3.215b) those from $u = n + 1$ to $n + l_x$ are dual of class x .

The setting up of the list then proceeds as follows:

Step 1. Assign the locations $t = 1$ to $t = n_x$ on the list to the dual constraints of the linear programming problems of class x , cast in the form

$$\varphi(\lambda; y, z, \dots) \leq 0.$$

Order these inequalities in the same order as the variables x_i . I.e., set $t = i$. Replace all programming variables by strategy components according to (3.2.15), and multiply both sides of the resulting inequalities by $(p_{N+1} p_{N+2} \dots p_{N+M})$.

The result is a set of n_x homogeneous inequalities in the components p_u . Proceed analogously with the dual constraints of the linear programming problems of classes $x, y, \dots$, in that order.

At the conclusion of Step 1, the first $n_x + n_y + \dots = n$ locations on the list will have been filled.

Step 2. Continue by entering in the same way as under Step 1, the primal constraints of the linear programming problems of classes $x, y, z, \dots$, in that order, all first cast in the form

$$\varphi(x, y, z, \dots) \leq 0.$$

Assuming that there are $\ell_x, \ell_y, \dots$ primal constraints in these linear programs, locations $t = n + 1, n + 2, \dots, n + \ell = N$, $\ell = \ell_x + \ell_y + n$ will be filled at the conclusion of Step 2.

Step 3. Continue by entering in the same way as under Step 1, the inequalities between the objective functions of the primal and dual linear programs classwise, but with the directions of the inequalities reversed. I.e., enter

$$s_x - \sigma_x \leq 0, \quad s_y - \sigma_y \leq 0, \quad \dots$$

Complete the step in the same way as Steps 1 and 2.

The list at this point contains $(N+M)$ inequalities, all homogeneous and of the form

$$(3.221) \quad \sum_{uvw\dots} f(t, u, v, w, \dots) p_u p_v p_w \dots \leq 0$$

The next three steps define those coefficients f in (3.221) which can be identified with coefficients in the constraints and

objective function inequalities of the linear programs class-wise.

Step 5. Identify the values of the first index t of f , for t from 1 to n_x , with the locations on the list that are occupied by inequalities derived from the dual constraints of class x . For each such t , the remaining indices are used to identify the terms in the inequality in the t -th location of the list. The last index of f , w say, in particular is identified with the subscript of the p -component of class x in the term. The indices u, v , etc. between t and w are identified with the subscripts of the other p -components in the term, for instance in such a way that $u < v < \dots$. The coefficient f with indices defined in this fashion is then set equal to the coefficient of the term. The values of t from $n_x + 1$ to n are then lexicographically allocated to the inequalities derived from the dual constraints of classes $y, z, \dots$; for each term in one of those inequalities, the identification of $w, u, v, \dots$ is then made in a manner analogous to the one described above.

In the example that has been referred to repeatedly the inequalities derived from the dual constraints of class x are (3.217a). The one shown in particular is in location $t = i$. Suppose the term of interest is $b_{i23} p_v p_w p_{n+2}$. As it appears in that inequality, the indices u, v , and w are $v = n_x + 2$, $w = n_x + n_y + 3$, and $u = n + 2$. Step 5 however requires that

this indexing be changed: w should denote the index of the p -component of class x which is $n + 2$ in this case, and u should be the smaller of the other two indices. Thus, the coefficient b_{i23} in (3.217a) is

$$b_{i23} = f(i, n_x+2, n_x+n_y+3, n+2),$$

in the notation of (3.221). The same coefficient in (3.217b) which is derived from a dual constraint of class y has the factors $p_u p_w p_{n+5}$ there, with $u = i$, $w = n_x + n_y + 3$, and is listed in place $t = n_x + 2$ on the list. Thus, in this case

$$b_{i23} = f(n_x+2, i, n_x+n_y+3, n+5)$$

Step 5 is specific in how the last index w of f is to be chosen. It is useful to establish that this choice is always feasible.

Lemma 3.231. After the conversion to the form (3.221) of a dual constraint in a linear program, formulated classwise from a multilinear one, each term of the resulting inequality contains as factor a p -component of every class.

Proof. Consider a term of degree K in such a constraint. It will contain as factor one dual variable, say from class y , and, $(K-1)$ primal ones of classes other than y . Suppose first that x_i is among the latter. Replacement of all of these variables with p -components will then introduce one such component of class x . Its presence will remain unchanged by the multiplication of the term with $(p_{N+1} p_{N+2} \dots p_{N+M})$. Thus, Step 5 will be feasible in this case. Suppose accordingly that

the term contains no variable of class x at the outset. In this case however the multiplication introduces the factor p_{N+1} which is a p -component of class x , according to

Definition 3.21. Step 5 is thus feasible also in this case. ¶

Step 6. The values of the first index t of f , from $t = n + 1$ to $n + \ell_x$ are also identified with the positions on the list of the inequalities derived from the primal constraints of class x . The identification of the other indices is done as in Step 5. The values of t from $t = n + \ell_x + 1$ to $t = n + \ell = N$, and the other indices of f are analogously defined.

A lemma similar to 3.231 might be interpolated at this point, showing that the identifications required by Step 6 are always feasible. The similarities however are great enough to make it superfluous. The same applies to the following:

Step 7. The values from $t = N + 1$ to $t = N + M$ of the first index of f are allocated to the reversed inequalities derived from those between the objective functions of the linear programs, primal and dual, of classes x, y, z , etc., in that order.

At this point, coefficients $f(t, u, v, \dots w)$ have been defined for every t and for certain combinations of the remaining indices. These definitions must be extended to other combinations of indices but in such a way that they satisfy the symmetry property and the zero-sum property of an $(M+1)$ -person game.

These properties were defined in the preceding section. The symmetry property in particular was stated as eq. (3.23b). It required that for every t , f be invariant under interchanges pairwise of the remaining indices.

$$(3.222) \quad f(b, u, v, \dots) = f(t, v, u, \dots w) = \dots = f(t, w, v, \dots u)$$

The zero-sum property will be enforced here in the special form (3.210b) required by Theorem 3.22. It specifies that f change its sign under the interchange of t with the last index w of f but that it be zero for all interchanges of t with other indices. Thus

$$(3.223) \quad f(t, u, v, \dots w) = -f(w, u, v, \dots t)$$

$$f(u, t, v, \dots w) = f(v, u, t, \dots w) = \dots = 0$$

The next two steps impose these specifications and the last assigns values to all those f which were left unspecified by the first nine.

Step 8. For each t , supplement the coefficient values f defined by Steps 5, 6 and 7 with those mandated by the symmetry rule (3.222).

Step 9. For each t , supplement the coefficient values f defined in the preceding steps with those mandated by the zero-sum rule in the form (3.223).

Step 10. All f values that remain undefined by the preceding steps are now set equal to zero.

This ten-step procedure, as Lemma 3.2.31 indicates, is never infeasible. I.e., each of the steps is made up of instructions which can always be executed. The end product is an $(M+1)$ -dimensional array with at least one value of f assigned to every location $(t,u,v,\dots w)$ in the array. If it can be shown further that never is there more than one value assigned to the same location, the array is a payoff function for an $(M+1)$ -person game satisfying the assumptions of Theorem 3.22. It will now be shown that those assignments are indeed unique.

Theorem 3.23. The ten-step procedure developed above assigns one and only one value of f to every $(M+N)$ -tuple of indices $(t,u,v,\dots w)$.

Proof. The proof consists of three parts. It is first pointed out that non-uniqueness, if any, can be introduced only in Steps 8 and 9. Secondly, it is shown that Step 8 makes assignments only to $(M+N)$ -tuples that have not had any from earlier steps. Finally, it is shown that the assignments of Step 9 always duplicate earlier ones.

The first three steps of the procedure provide merely for an orderly listing of all inequalities of the linear programs classwise that are derived from a given multilinear problem, and for their conversion by some simple algebraic manipulations. No non-uniqueness can occur. None can occur in the next three steps which in effect provide for a similarly orderly listing of the individual terms in each inequality. Finally, non-uniqueness cannot be introduced in Step 10. It thus remains to review Steps 8 and 9.

In Step 8, a coefficient value f that has been previously assigned to one $(N+M)$ -tuple $(t, u, v, \dots w)$ is assigned also to others. These new assignments are however always made to $(N+M)$ -tuplets with the same t , as (3.222) shows, and only to those differing from the original by interchanges of indices pairwise. Consider first an interchange which does not involve the last index w of f , for instance one between u and v . Since initially $u < v$, always, according to Steps 5, 6, 7, the interchange establishes a situation for which no value assignment has been made prior to Step 8. That step therefore makes only new assignments in its enforcement of the symmetry rule (3.222), whenever the last index w of f is not involved. Assume accordingly next that it is. But since w is always last its interchange with, say u , also establishes an unprecedented $(M+N)$ -tuple. The same conclusion therefore applies, and exonerates Step 8 from all non-uniqueness problems.

It remains to do the same for Step 9. Fix t again but assume also that $t \leq n$. Considerations are thus limited for the moment to coefficients f in inequalities that are derived from the dual constraints classwise. To be more specific, assume first that $t = i$, $1 \leq i \leq n_x$. The original constraint was then dual of class x , by Step 1. The dual variables in it must be of class x also and the p -components that corresponded to them according to (3.215b) had indices running from $(n+1)$ to $(n+l_x)$. These indices were relegated to the last place by Step 5, as arguments of f , and denoted with w . Consider $w = n + r$,

$1 \leq r \leq n + \ell_x$. Step 9 then requires that, first of all, (3.223a) hold under the exchange of t and w , i.e., that

$$f(i, u, v, \dots, n+r) = -f(n+r, u, v, \dots, i)$$

Now, the coefficient on the right figures in an inequality that is in the $(n+r)$ -th place on the list. By Step 1, it is an inequality derived from a primal constraint of class x , and the term that involves that coefficient has (by Step 6) p_i as the p -component of class x as factor. The other factors $p_u, p_v, \dots$, and the order $u < v < \dots$ of their subscripts, are unaffected by the interchange. The coefficient of this term was defined by Step 6 but, since it appeared in an inequality derived from a primal constraint of class x , had a sign opposite to that of its dual counterpart. Unlike (3.222) therefore, (3.223a) does not assign a coefficient value to an $(N+M)$ -tuple to which no value had been assigned previously; it merely duplicates such an assignment. Eq. (3.223b) on the other hand does define new assignments. It is concerned with interchanges pairwise between the first index t of f and any other, except the last one. Such interchanges generate $(N+M)$ -tuples that could not have been generated by any of the earlier steps. To see this assume again that t designates an inequality of class x . According to Step 5, w then always designates a p -component that is also of class x . An interchange between t and any index other than w , u for instance, will however destroy the symmetry classwise between the first and last indices. It will thus

generate an $(N+M)$ -tuple which could not have been assigned a coefficient value by any of the earlier steps.

This line of argument is clearly valid also for variable classes other than x . It is valid, as one readily convinces oneself, if inequalities derived from those between the objective functions of the linear programs classwise are involved, rather than the primal constraints. The theorem may accordingly be considered proven. ¶

The upshot of the portion of this proof dealing with Step 9 is that that step is really superfluous. The rule embodied in (3.223a) merely duplicates assignments made earlier in the procedure, and (3.223b) makes a provision which merely anticipates that of Step 10. Hence, one has

Corollary 3.231. Step 9 can be omitted from the ten-step procedure.

The main purpose of the discussion in this section so far is the following implication of Theorem 3.23.

Corollary 3.232. A solution classwise to a multilinear programming problem of degree D , and with M classes of variables, can be sought by way of the equilibrium strategies of an appropriately constructed $(M+1)$ -person game. The iterative application of the transformation (3.29), supplemented with the (3.215), will converge on such a solution, provided only that the last M components of the equilibrium strategy differ from zero. The expression for ϕ_t in this transformation is the

left-hand side of the inequality (3.221) in the t - th place on the list of such inequalities.

Theorem 3.23 and Corollary 3.232 provide a computational process which often leads to solutions classwise of multilinear programming problems. These solutions are also simultaneous for the primal problem but they lack the simultaneity (as defined in Definition 3.11) for the dual problems. This is likely to make the solution uninteresting in many cases because it need not be a Kuhn-Tucker point, let alone properly optimal. The question of how to insure simultaneity classwise in both the primal and dual solutions, and hence how to satisfy the Kuhn-Tucker conditions, is taken up in the next section.

3.24 Simultaneous Solution

The procedure described in the preceding section leads to a simultaneous solution classwise of a multilinear programming problem. It also leads to solutions of the dual problems classwise but not to simultaneous solutions, in the sense of Definition 3.11. This would require that certain equations be enforced among the optimal values for the dual variables. It would be good however if such an enforcement could be achieved for in that case the solutions would not be merely optimal classwise. They would be Kuhn-Tucker points, as Theorem 3.11 shows. This section is devoted to proving that a relatively minor modification in the solution algorithm (3.29) can accomplish the desired end.

The discussion begins with a lemma in which this modification is introduced and the convergence of the resulting algorithm is demonstrated. The theorem that follows then shows that proper optimality is indeed achieved in this way.

Lemma 3.241. Suppose that an equilibrium solution of a symmetric zero-sum game were to be found which is subject to the requirement that the components $\hat{p}_t$ of the common equilibrium strategy of the players be equal for $t \in T$ where T is some subset of the range of t . In this case, the iterative application of the transformation

$$(3.224a) \quad p'_t = (p_t + \rho\psi_t)/(1 + \rho\psi) \quad , \quad \psi = \sum \psi_t$$

with

$$(3.224b) \quad \psi_t = \begin{cases} \phi_t & \text{for } t \notin T \\ \sum \phi_t / |T| & \text{for } t \in T \end{cases}$$

will approach such an equilibrium strategy. (Here, $|T|$ denotes the number of elements in T .)

Proof. This lemma is evidently a generalization of Theorem 3.22 and its proof proceeds along the same lines as that of the theorem. For notational convenience assume here, as there, that there are four players. Moreover, let the subset T consist of the indices of the last three p -components. I.e., $T = \{N+1, N+2, N+3\}$. Suppose that in the course of applying (3.224), with

$$\psi_t = \begin{cases} \phi_t & t = 1, 2, \dots, N \\ \frac{1}{3} (\phi_{N+1} + \phi_{N+2} + \phi_{N+3}) & t = N+1, N+2, N+3 \end{cases}$$

the equality $p_{N+1} = p_{N+2} = p_{N+3}$ has been realized. The next application of (3.224) will then clearly preserve this equality, and so will therefore all subsequent ones. It will now be shown that these transformations converge on $\psi_t = 0$ (hence also $\phi_t = 0$). The proof relies on the fact that $\sum \psi_t^2$ is a Lyapunov function, i.e., that unless it vanishes

$$\Sigma \psi_t'^2 - \Sigma \psi_t^2 = 2 \Sigma \psi_t \frac{\partial \psi_t'}{\partial \rho} \Big|_{\rho=0} + O(\rho^2)$$

is negative if ρ is small enough.

Consider first the derivative on the right for $t \leq N$. It differs from its counterpart in (3.2.12) only in that the transformation from p to p' is now given by (3.2.24) instead of by (3.2.9). The same manipulations as in the proof of Theorem 3.22 now lead to an analogous expression, namely

$$\frac{\partial \psi_t'}{\partial \rho} = 3 \Sigma_{uvw} f(t, u, v, w) p_u p_v \psi_w - 3 \psi \psi_t \quad (\rho=0).$$

The derivatives of ψ_t' for $t = N+1, N+2, N+3$ are treated differently. Consider

$$\frac{\partial \psi_{N+1}'}{\partial \rho} = \frac{1}{3} \Sigma_k \frac{\partial \phi_{N+k}'}{\partial \rho} \quad (k=1, 2, 3; \rho=0)$$

where, using the same analogy as above

$$\frac{\partial \phi_{N+k}'}{\partial \rho} = 3 \Sigma_{uvw} f(N+k, u, v, w) p_u p_v \psi_w - 3 \psi \psi_{N+k} \quad (\rho=0)$$

so that

$$\frac{\partial \psi_{N+1}'}{\partial \rho} = \Sigma_{uvw} p_u p_v \psi_w \Sigma_k f(N+k, u, v, w) - \psi \Sigma_k \psi_{N+k} \quad (\rho=0).$$

One now concludes that, for $j = 1, 2, 3$ and $\rho = 0$,

$$\begin{aligned} \Sigma \psi_t \frac{\partial \psi_t'}{\partial \rho} &= 3 \Sigma_t^N \psi_t \Sigma_{uvw} f(t, u, v, w) p_u p_v \psi_w - 3 \psi \Sigma_t^N \psi_t^2 \\ &+ \Sigma_j \psi_{N+j} \Sigma_{uvw} \Sigma_k f(N+k, u, v, w) p_u p_v \psi_w - \psi \Sigma_j \psi_{N+j} \Sigma_k \psi_{N+k} \end{aligned}$$

Keeping in mind that $\psi_{N+1} = \psi_{N+2} = \psi_{N+3}$, the last term in this expression is readily converted to

$$\psi_j \psi_{N+j} \sum_k \psi_{N+k}^2 = 3 \psi_j \sum_k \psi_{N+1} \psi_{N+k} = 3 \psi_j \sum_k \psi_{N+k}^2.$$

In the term next to the last, the sums of interest are

$$\begin{aligned} \sum_j \psi_{N+j} \sum_k f(N+k, u, v, w) &= 3 \sum_k f(N+k, u, v, w) \psi_{N+1} \\ &= 3 \sum_k \psi_{N+k} f(N+k, u, v, w). \end{aligned}$$

These two conversions show that, for $\rho = 0$,

$$\sum_t \psi_t \frac{\partial \psi_t}{\partial \rho} = 3 \sum_{uv} p_u p_v \sum_{tw} f(t, u, v, w) \psi_t \psi_w - 3 \psi \sum_t \psi_t^2$$

which is of the same form exactly as (3.213), except for the deployment of ψ_t in place of ϕ_t . The same argument as in the proof of Theorem 3.22 can now be used to complete the one of this lemma. ¶

There are obvious ways in which this lemma can be generalized but, since its present form is adequate for what follows, these extensions will be ignored. The main result to be derived here is the following:

Theorem 3.24. Consider a multilinear programming problem involving M classes of variables. The solution algorithm (3.224), coupled with (3.215), then converges on the problem solution that is simultaneous classwise for the primal as well as dual problems, provided the initial values of the components p_t of the strategy p are suitably chosen and that the equilibrium values $\hat{p}_{N+1}, \hat{p}_{N+2}, \dots, \hat{p}_{N+M}$ of the last M components differ from zero.

Proof. According to Theorem 3.12, a solution classwise to a primal multilinear programming problem is also such a solution to the dual one if equality prevails among the dual variables classwise which are associated with every nonlinear constraint. Thus if λ_1, μ_1 are the dual variables of classes x and y that are associated with the bilinear constraint $\phi_1(x, y) \leq 0$ their solution values $\hat{\lambda}_1, \hat{\mu}_1$ must satisfy

$$(3.226) \quad \hat{\lambda}_1 = \hat{\mu}_1.$$

It will now be shown that the solution algorithm (3.224), under the assumptions of this theorem, converges to an equilibrium strategy $\hat{p}$ which implies (3.226) for the bilinear constraint ϕ_1 .

Suppose that relation (3.215) between programming variables and strategy components, for λ_1 and μ_1 , is

$$(3.227) \quad \lambda_1 = p_u / p_{N+1}, \quad \mu_1 = p_v / p_{N+2}$$

$$u = n+1, \quad v = n + \ell_x + 1.$$

Suppose further that during the execution of the algorithm (3.224), a situation arises in which, for the indices u, v in (3.227b)

$$(3.228) \quad p_u = p_v, \quad p_{N+1} = p_{N+2} = \dots = p_{N+M}.$$

It will be shown first that in such case the same equations hold also among the transformed p - components, i.e., that

$$p'_u = p'_v, \quad p'_{N+1} = p'_{N+2} = \dots = p'_{N+M}.$$

For the components p_{N+k} ($k=1,2, \dots M$) this follows from Lemma 3.241. To show that it is valid also for p_u, p_v , note that ψ_u, ψ_v in their transformations (3.224) to p'_u, p'_v are (see Corollary 3.232) derived from the left-hand side of the constraint, i.e., from $\phi_1(x,y)$. This derivation is the same for both ψ_u and ψ_v . First, x and y are replaced, as required by Step 2 of the procedure in the preceding section, with the p -components of classes x and y . Denote the vectors of those components with p_x and p_y . The resulting expressions are then multiplied by all of the last p -components. This leads to

$$\psi_u = \psi_v = \phi_1(p_x, p_y) p_{N+3} \dots p_{N+M}$$

if the right-hand side is positive, and

$$\psi_u = \psi_v = 0$$

otherwise. The indices u, v are as in (3.227b). Since $p_u = p_v$ by assumption, and $\phi_u = \phi_v$ as well, (3.224) now implies that also $p'_u = p'_v$, as was to be shown. It follows that the equalities (3.228) are preserved under subsequent applications of the transformation (3.224) and hence that, in the limit (3.226) holds, as desired. It holds in particular therefore if the values of p_u, p_v , and p_{N+k} ($k=1,2, \dots M$) are so chosen at the start of the solution algorithm that (3.228) holds among them.

The validity of this argument is clearly not restricted to single bilinear constraints. Thus if the p -components that are associated (via the dual variables $\lambda_m, \mu_m, \nu_m, \dots$ classwise)

with a constraint $\phi_m(x,y,z,\dots) \leq 0$ are chosen equal at the start of the solution algorithm, and if the same choice is made for the last p - components $p_{N+1}, \dots p_{N+M}$, the algorithm will ensure that the equalities needed for simultaneity are enforced in the equilibrium strategy. ¶

3.25 Comments

The objective of this section was to derive an iterative solution algorithm for the solution of multilinear programming problems. Such algorithms can be implemented as feedback control procedures of the kind that may be used in organization. Some of the ways in which they could actually be used are described in the next section.

The derivation on this section required the development of a good deal of new theory. It may be worth commenting on some of the results we obtained, especially on those which indicate a need for further development.

An area that seems in particular in urgent need of further research is the following. The solution algorithm we derive in Sect. 3.22, when applied to a multilinear programming problem involves several more variables (among those denoted with p_t in (3.2.15)) than the problem itself. If the latter is of Degree D , the algorithm requires D additional variables p_t . The convergence of the algorithm is proven on the assumption that none of these go to zero as $t \rightarrow \infty$. The question is what happens if they do? At this writing we have no answer but we conjecture that, if one of the p_t goes to zero, they all do, and that in such a case the programming problem is either infeasible or unbounded. There is some basis for this conjecture, mostly in the theory of linear programming (see Appendix 4.2B for an account of it) but it is well short of a proof.

Yet, a proof would be highly desirable. The algorithm

we derive is a control scheme for systems whose components have limited capacities of various kinds. In our work, organization members are limited in their processing capacities by their overload constraints. Computers and communication networks have similar limitations. Our algorithms are therefore control schemes for the considerable variety of systems that involve such components. It would be extremely useful to be able to say that the algorithms have, so to speak, built-in warning signals of infeasibility or of near-infeasibility. We conjecture that they do, and that the vanishing of those D extra solution variables p_t constitutes such a signal. The extra components are thus not merely useless baggage, to be discarded if at all possible, but essential elements in any control process for systems which are made, partly or wholly, of components with limitations on their capacities.

Our solution algorithm on the other hand does contain one piece of baggage that should be and that, by all present indications, can be discarded. This is its requirement that the sum of the solution variables p_t be equal to 1. It seems both undesirable and unnecessary. The main incentive for our wish to discard it is that, in our application, it is not the variables p_t which should add up to 1 but certain sets of variables in the underlying programming problem. many of the latter variables represent probabilities. They are accordingly the ones that should add up to 1. Under our solution algorithm, however, they need not do so, and in

general they will not do so, except in the limit as the solution is approached. However, it develops that one can enforce the requirement that certain sets of programming variables sum to unity but only if one abandons the requirement that the p_t do the same. Convergence of the algorithm is unaffected. The proof of this statement is, however, omitted in this report, partly because it is quite laborious and partly because it is too much in the nature of a technicality.

The convergence of the algorithm, it might be pointed out, is not one of the usual type. We do not prove in Sect. 3.2.2 that the variables p_t of (3.2.9) approach a limit. We only show that the quantities ϕ_t in (3.2.9) approach zero. This is unavoidable. It is the equations $\phi_t = 0$ which characterize a solution of the problem but there will in general be several vectors of variables p_t that are compatible with them. This fact will enter into our conclusions later on (see Corollary 3.2.1).

Multilinear programming problems can be solved by algorithms other than those we derived in Sect. 3.2. None of these are iterative, hence unattractive for the purpose we consider here. Some have also other features which we found unappealing. Among these are first of all the solution techniques for nonlinear programs in general (see, e.g., [1] for an introduction). Some of these are quite efficient numerically but they do not have the specific properties appropriate to multilinear problems. Geometric

programming [32][33] however, may seem tailored to such problems. This impression is deceiving. In our applications, the feasible region of the solution process of a geometric program is always unbounded and the solution itself almost always lies at infinity. Another precedent are the "generalized" linear programs of Dantzig and Wolfe [2] (Chap. 22). The solution algorithm for these, however, does not in general reach a solution that is a Kuhn-Tucker point, let alone optimal, for the problem.