Mixed Linear and Quadratic Programming by the Multiplex Method[1]

In a paper presented at the Tokyo-meeting of the International Statistical Institute May–June 1960 and in a memorandum from the University Institute of Economics, Oslo, of 27 August 1960, I give detailed computing instructions as well as a detailed numerical example of how to use the multiplex method for finding the solution of mixed linear and quadratic programming problems. Special considerations in this memorandum on coding for automatic computations were worked out in cooperation with Mr. Ole-Johan Dahl, research mathematician at the Norwegian Defence Research Institute.

The purpose of the present paper is to give a non-technical presentation, exhibiting the main lines of thought underlying the multiplex method and its use for mixed linear and quadratic programming.

Vector and matrix notation will not be used. This notation is entirely unnecessary both for technical and non-technical discussions on linear or quadratic programming.

Originally the multiplex method was developed for completely linear programming problems, but it has turned out that the method may be used in more or less the same way for rather general forms of the preference function. In this contribution only the mixed linear and quadratic case will be considered (including the strictly linear and the strictly quadratic).

The completely linear case has been coded for automatic computation on the Norwegian Defence Organization's electronic Mercury computer (Ferranti) by Mr. Ole-Johan Dahl, and for the English Electric Deuce, it has been coded by the Indian mathematician Mr. C. P.

[1] This contribution to the volume in honour of my old friend Johan Akerman is also issued as a mimeographed memorandum of 7 November 1960 from the University Institute of Economics, Oslo.

Saksena, formerly a research associate in the University Institute of Economics, Oslo.

1. *The problem*

Consider N variables x_j ($j=1, 2 \dots N$) connected by the m ($\leqq N$) independent linear equations

$$(1.1) \qquad a_{io}+\sum_{j=1}^{N} a_{ij}x_j=0 \qquad (i=1, 2 \dots m)$$

where a_{io} and a_{ij} are given constants.

Since the equations (1.1) are assumed linearly independent, it is possible at least in one way to select $n=N-m$ ($\geqq 0$) *basisvariables* x_k ($k=u, v \dots w$ or shorter $k=$bas) and solve the equations (1.1) in terms of the remaining m variables, the *dependent* variables. Let the expressions thus obtained be

$$(1.2) \qquad x_j=b_{jo}+\sum_{k=u \dots w} b_{jk}x_k \qquad (j=1, 2 \dots N)$$

where the b_{jo} and b_{jk} are coefficients that are uniquely determined when the coefficients of (1.1) are given.[1]

If by convention we put

$$(1.3) \qquad b_{jo}=0 \qquad b_{jk}=e_{jk}=\begin{bmatrix} 1 \text{ if } j=k \\ 0 \text{ otherwise} \end{bmatrix} \quad \begin{array}{l}\text{when } j \text{ is one of} \\ \text{the basis affixes} \\ u, v \dots w\end{array}$$

the equations (1.2) hold for all $j=1, 2 \dots N$ (or shorter $j=$all), and not only for the non-basis values of j.

We assume that the variables are bounded by

$$(1.4) \qquad \underline{x}_j \leqq x_j \leqq \bar{x}_j \qquad (j=\text{all})$$

where $\underline{x}_j$ and $\bar{x}_j$ are given constants, satisfying

$$(1.5) \qquad \underline{x}_j \leqq \bar{x}_j \qquad (j=\text{all})$$

As special cases we may for one or more of the j have $\underline{x}_j=-\infty$ and|or $\bar{x}_j=+\infty$.

[1] The following terminological remark may be helpful: In the simplex method the "basis" is the $m \cdot m$ matrix whose inverse permits to solve the equations (1.1) in the form (1.2). In this method a series of different selections of the set x_k ($k=u \dots w$) – and hence of the "basis" – is considered. In the multiplex terminology the set x_k ($k=u \dots w$) defines the "basis variables". And in this method one will as a rule only consider one such set.

We consider a preference function of the form

$$f = p_o + \sum_{\varkappa = u\,..\,)U\,..\,W(\,..\,w} p_\varkappa x_\varkappa + \sum_{K=U\,..\,W} p_K x_K - \frac{1}{2} \sum_{K=U\,..\,W} \sum_{H=U\,..\,W} P_{KH} x_K x_H \tag{1.6}$$

where p_o, $p_\varkappa$ and p_K may be any arbitrarily given real numbers, and P_{KH} a real (symmetric) positive definite and *non-singular* matrix. The variables $x_\varkappa$ ($\varkappa = u, v \dots$) $U \dots W$ ($\dots w$ or shorter $\varkappa=$lin) are *lineo* variables, i.e. variables that occur *only* linearly in the preference function, while x_K ($K = U \dots W$ or shorter $K=$quad) are *quadrato* variables, i.e. variables that actually occur in the non-singular quadratic form (and may or may not enter also linearly in the preference function). The two sets: lineo and quadrato are mutually exclusive. Together they make up the complete set of n basis variables x_k ($k = u \dots w$). The inverted parenthesis in (1.6) means "exclusion of". That is, $\varkappa = u \dots)U \dots W(\dots w$ means the basis affixes with the exclusion of the quadrato affixes $K = U \dots W$. The constant p_o in (1.6) is immaterial, but is included for formal completeness. The matrix element P_{KH} is interpreted as zero whenever K and|or H is the affix of a lineo variable.

Any mixed linear and quadratic programming problem — with a singular or non-singular quadratic part - can be reduced to the form (1.6) with a non-singular P_{KH}. The essence of the problem is that if this is done, there may - and virtually always will - appear certain variables that enter only linearly in the preference function. In the desingularized form the bounds on the transformed variables will still be linear.

The partial derivatives of (1.6) in the point x are

$$f^x_k = \frac{\delta f}{\delta x_k} = p_k - \Sigma_H P_{kH} x_H \qquad (k=\text{bas}) \tag{1.7}$$

If x^o is any given point, the expression for the preference function can also - by Taylor's formula - be written

$$f = f^o + \Sigma_k f^o_k (x_k - x^o_k) - \frac{1}{2} \Sigma_{KH} P_{KH} (x_K - x^o_K)(x_H - x^o_H) \tag{1.8}$$

where the superscript o refers to the point x^o.

The problem is to maximize (1.6) - or, if we like, (1.8) - subject to the constraints (1.4) where the x_i are given by (1.2).

In many practical problems the quadratic part of the preference function will *have a much lower rank* than the number of degrees of freedom in the problem, (e.g. the case where the preference function is simply a quadratic function of one of the dependent variables or of a linear form in a few of them). This means that a number of lineo variables will emerge when the quadratic form in the preference function is reduced to a non-singular form. Hence it is essential to consider the mixed linear and quadratic case. The completely quadratic case (with a non-singular matrix) is so special that it has little practical interest.

2. *Graphical illustration of the multiplex method*

Fig. (2.1) illustrates the situation in the completely linear case. The interior of the pentagon PQRST and its boundary represent the set of points where all the constraints (1.4) are satisfied. This is the admissible region. Each of the sides of the pentagon represent points where one of the variables is on one of its bounds. For instance the straight line TS may represent the points where x_1 is at one of its bounds, while SR may represent the points where x_2 is at one of its bounds, and so on. It is a classical fact that the admissible region bounded by linear constraints of the form (1.4), where the x_j are given by (1.2), must always be *convex*, i.e. such that the segment of straight line joining any two points in this region must in its entirety lie in the admissible region. The convexity of the admissible region is a feature that considerably facilitates the programming problem, whether attacked by the multiplex method or by some other method.

The dotted lines in fig. (2.1) indicate contour-lines of the (strictly linear) preference function, i.e. lines along which the preference is constant. The value of the preference function increases towards north-east in the diagram. The gradient on these contour lines is exemplified by the arrows starting from the point A and the point B. These arrows are orthogonal to the contour lines.

The essence of the multiplex method is that we may start in *any* point in the admissible region (its boundary included) and from this starting point we move as if the point were pulled by a gravitational force represented by the preference gradient.

What will happen in the example of fig. (2.1) is obviously that the point will move from A to B, then follow the boundary from B to S, and next follow the boundary from S to R, this last point being the *optimum*, i.e. the point in the admissible region where the preference function assumes the highest value which it can reach in the admissible region. We can visualize this movement from A to R by turning the paper in such a way that the preference gradient points directly downwards, and then think that the moving point seeks to find the lowest possible position.

The essence of this approach is that the point may move either completely unconstrained, as from A to B, or constrained by a part of the boundary. In the two dimensional diagram in fig. (2.1) it is not possible to illustrate all the cases of a boundary movement that may occur in several dimensions. In the general case of n degrees of freedom the point may move either constrained by one boundary condition (i.e. move in an $(n-1)$ dimensional manifold), or move constrained by two boundary conditions (i.e. move in an $(n-2)$ dimensional manifold) . . . etc. or possibly move constrained by $(n-1)$ boundary conditions, (i.e. move along a one dimensional manifold, i.e. along an *edge* on the boundary). Hence the term *multiplex*: We may move in *any* such manifold – or even move completely unconstrained – until we reach an optimum point.[3]

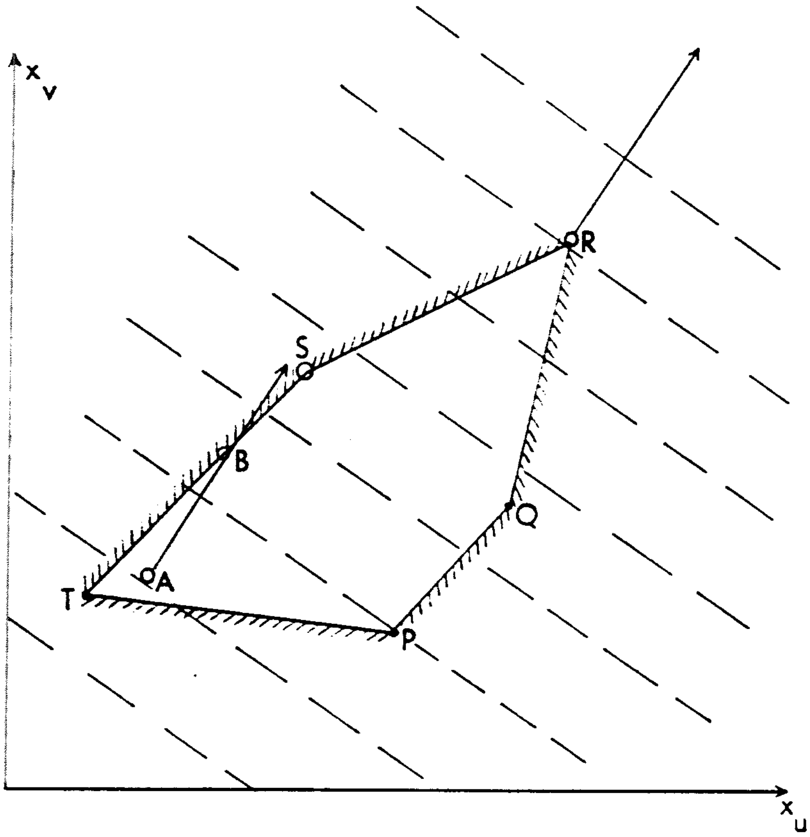

Fig. (2.1)

[3] The simplex method represents the special case where we only move along a one dimensional manifold, i.e. always move along an edge, going from one vertex to another, finally to reach an optimal vertex.

Fig. (2.2) illustrates the case where the preference function is of the form

$$(2.3) \qquad f = x_{\varkappa} - x_{K}^{2}$$

i.e., it contains one lineo variabel $x_{\varkappa}$ and one quadrato variabel x_{K}.

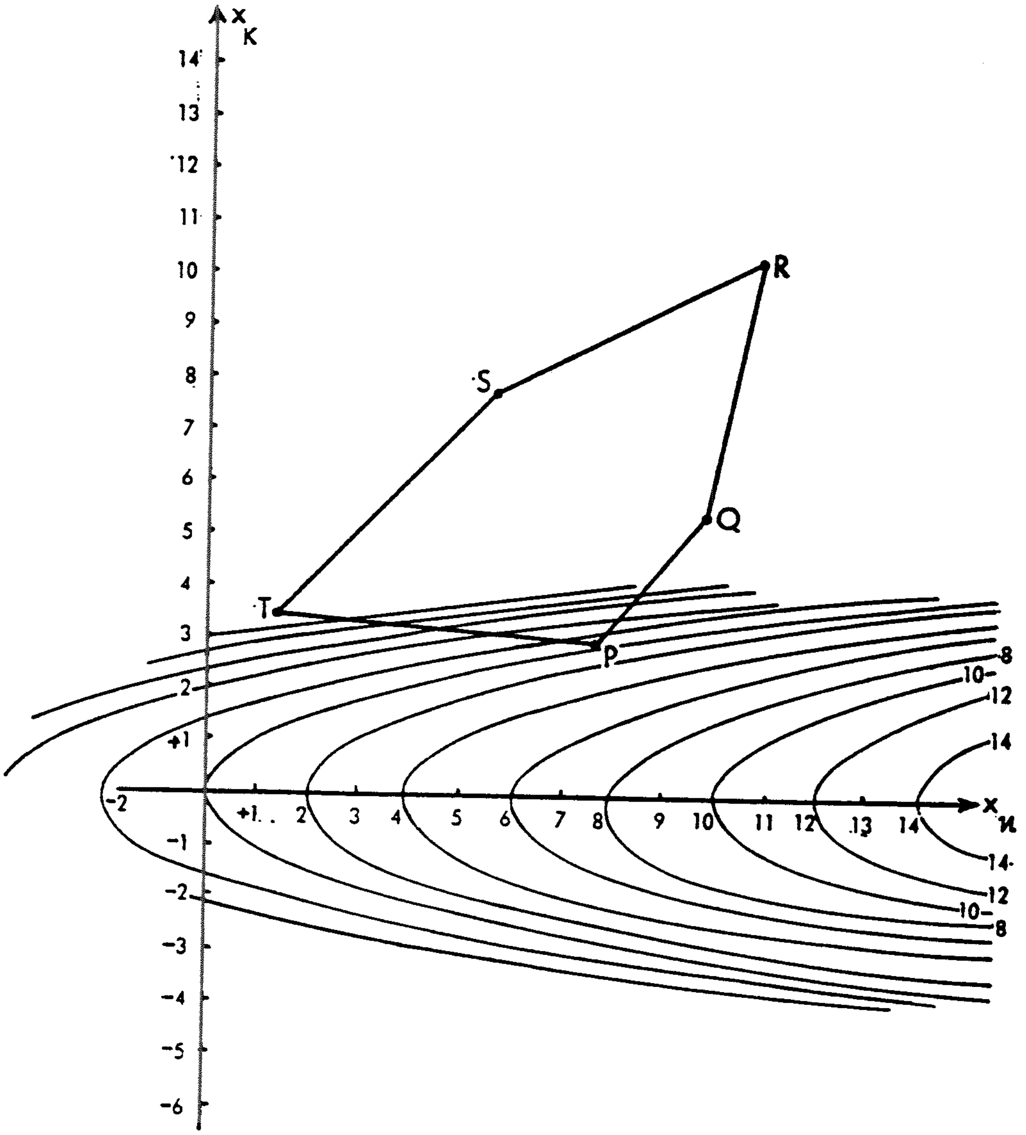

Fig. (2.2)

The contour lines of the preference function are in this case the parabolas indicated in the figure. The value of f along any given contour line is indicated by the number written to the right (and may also be read off as the value of $x_{\varkappa}$ in the point where the parabola in question intersects the $x_{\varkappa}$-axis).

If the variables are not bounded at all, there does not exist in fig. (2.2) any finite point where the preference function reaches its maximum. Indeed, if the quadrato variable x_{K} is kept constant, we may render f arbitrarily great by increasing $x_\varkappa$. On the other hand if the lineo variable $x_\varkappa$ is kept constant, a finite maximum is always reached by putting $x_K=0$. In this specialized maximum we have $f=x_\varkappa$.

If the admissible region from fig. (2.1) is superimposed on fig. (2.2), we see at a glance that the optimum is in the point P. Reading off from the diagram by the eye, we see that the optimum value of f is approximately equal to 0.

In fig. (2.2) the optimum occurred in one of the *vertices* of the admissible region. This needs not always be so. We can exemplify this if we replace the admissible region of fig. (2.2) by an admissible region with an eastern border that has *a point of tangency* with one of the contour lines of the preference function. This point of tangency will then be the optimum.

In the general case of n degrees of freedom we must be prepared for the following situation: If there are *no* lineo variables, the optimum may occur either in the interior of the admissible region (i.e. where no variables are bound-attained) or in some point on the boundary. And in the latter case there may be any number ν of variables ($\nu=1, 2 \ldots$) that are bound-attained in the optimum. If one or more lineo variables are involved, the optimum must occur in some point on the boundary. And this may be a point where any number ν of variables ($\nu=1, 2 \ldots$) are bound-attained.

It will be readily recognized that the time-honored method of Lagrange multipliers – which frequently is so effective for locating the maximum of a continuous function under constraints expressed by continuous functions – cannot produce the solution in the general mixed linear and quadratic programming problem here considered.

But the multiplex method will produce the solution by a well defined algorithm. Although the algorithm must necessarily contain a rule that covers a number of different cases which may occur in the course of the computation, and thus prima facie does not appear very simple, it can be worked out in a mechanical way which is fairly simply to apply both for desk machine computation and for automatic computation.

One advantage of the multiplex method is that when we use it, we are fairly well protected against the computational difficulties that may arise when a number of the variables are (exactly or nearly) *linearly dependent*. In many practical problems with a great number of variables, it will often happen that the admissible region is surrounded by *thick layers* of variables that are (exactly or nearly) linearly dependent, and this may cause considerable difficulties unless we have a method that can handle such situations. Further explanations on what is meant by linear dependencies are given below.

3. *Decisions to make in each round of the algorithm*

For the sake of simplicity we will first consider the case of a completely linear preference function, i.e., a preference function containing only lineo variables. The subsequent generalization to the case of a mixed linear and quadratic function will then be easy.

The algorithm in the completely linear case proceeds in *rounds* which in fig. (2.1) are illustrated as follows:

Round 0: The move from the initial point A to the first *breaking-out* point B. Round 1: The move from B to the second breaking-out point S. Round 2: The move from S to the optimum point R.

Round 0 is characterized by the fact that we make a *free preference move,* i.e. we move along a *beam* defined by the basis direction numbers

$$d_k = p_k \qquad (k=\text{bas}) \tag{3.1}$$

The basis direction numbers d_k are simply the direction numbers of the beam, i.e. numbers such that the values of the basis variables along the beam are given by

$$x_k = x_k^\circ + \lambda d_k \qquad (k=\text{bas}) \tag{3.2}$$

where x_k° is the initial point and λ is a parameter whose increase from zero through positive values generates the beam.

When we reach the point B – the first breaking-out point – we cannot proceed further in the direction (3.1). In the further move we must impose the condition that we shall move along one of the *boundary constraints,* namely the one defined by the *constancy* of the variable that is *bound-attained* in the point B. In the beginning of Section 2 this was exemplified as x_1=constant. We will express this by saying that in the move from B, x_1 is included in *the operation set.*

This exemplifies the following general problem: In any given point on the boundary we will have to *decide* which ones of the bound-attained variables that are to be included in the operation set. We cannot always include in the operation set *all* the variables that are bound-attained in a given point. Indeed, for effective computation we must impose the condition that all the variables that are to be included in the operation set, must be *linearly independent.* What is meant by this can again be exemplified by the situation in the point B.

Suppose for instance that there were two variables, say x_1 and x_3, which are such that we always have

(3.3) $x_3 = a + bx_1$

where a and b are given constants. The meaning of (3.3) is that this equation shall hold good – with the *same* constants a and b – regardless of what values are chosen for the basis variables. In this case we will say that x_3 and x_1 are linearly dependent.[4]

Further suppose that the two constants a and b are such that in the point B $x_3 = x_3^*$ and $x_1 = x_1^*$ where x_3^* is the bound which x_3 has reached in B, and x_1^* is the bound which x_1 has reached in B. This means that

(3.4) $x_3^* = a + bx_1^*$

Hence

(3.5) $(x_3 - x_3^*) = b(x_1 - x_1^*)$ (for all values of the basis variables).

In other words, to impose the condition that x_1 shall remain equal to the value it has attained in B is in fact the *same* condition as to impose that x_3 shall remain equal to the value it has attained in B (provided $b \neq 0$). These two conditions are in fact one and the *same* condition. Or, geometrically expressed, the line TS is a *double* line, depicting at the same time the condition $x_3 = x_3^*$ and $x_1 = x_1^*$. In order to avoid computational indeterminacies we must only include one of these two bound-attained variables in the operation set.

In the general case where several variables are bound-attained in a given point, we must make a pick between the bound-attained variables and only include in the operation set such bound-attained variables that are linearly independent. That is, variables such that there does not exist between them any linear relation that holds good regardless of what values we choose for the basis variables. This illustrates the first problem we have to solve in a point on the boundary: *The inclusion problem,* i.e. what variables to include in the operation set.

In point S in fig. (2.1) a second type of decision is illustrated. Here two variables are bound-attained, x_1 and x_2, x_1 being bound-attained along TS and x_2 being bound-attained along SR. And these two variables cannot be linearly dependent (since the slopes of the two lines are different). Therefore, the requirement that the two variables shall be linearly independent, does *not* prevent us from including both of them in the operation set.

[4] Or, more precisely, that x_3 is linearly dependent on x_1.

But there is now another consideration that prevents us from retaining both of them in the operation set. If we did, we would by this fact *petrify* the point S. We would not be able to move away from S. Indeed, the two conditions that both x_1 and x_2 shall remain constantly equal to the values they assume in S, are sufficient to determine the point exactly. But a mere inspection of the diagram tells us that if we remain in the point S, we will not be able to reach the maximum in R. Hence in S we are facing a new problem: The ***exclusion problem.*** In S we have to drop one of the variables from the operation set. In the simple case exhibited in fig. (2.1) it is easy to see from the diagram which one of the two variables we have to drop. We must drop the condition on x_1 (which compels us to move along TS) and only retain the condition on x_2 (which compels us to move along SR). This decision is simple in fig. (2.1), but in the general case we need a criterion for which one (or which ones) of the variables to drop from the operation set.

We now have to consider how the inclusion criterion and the exclusion criterion can be formulated in the general case of n basis variables, i.e., n degrees of freedom.

4. *Directional admissibility. The inclusion criterion*

Suppose that from any given point x_k^o we attempt to make a move along a beam defined by any set of given direction numbers d_k (k=bas). The meaning of these direction numbers is given by (3.2).

If the basis direction numbers are d_k, the increments of any of the *dependent* variables x_j are given by

$$x_j = x_j^o + \lambda d_j \qquad (j=\text{dep}) \tag{4.1}$$

where

$$d_j = \sum_{k=bas} b_{jk} d_k \qquad (j=\text{dep}) \tag{4.2}$$

This simply follows by inserting (4.1) into (1.2). The definition (4.2) can be applied also to the basis variables x_k if we adopt the convention (1.3).

If x_j is bound-attained in the point $x_k{}^o$, the move will be possible only if

$$d_j \text{ is } \left[\begin{array}{l}\text{non-negative when } x_j^o \text{ is at its lower bound } \underline{x}_j \\ \text{non-positive when } x_j^o \text{ is at its upper bound } \bar{x}_j \\ \text{arbitrary if } x_j^o \text{ is not at any of its bounds}\end{array}\right. \tag{4.3}$$

If (4.3) is fulfilled for all j, basis-affix or dependent affix, the move considered is possible. In this case we will say that the beam defined by the basis direction members d_k *is directionally admissible.* (This condition is not to be confounded with the *point-admissibility* expressing that any given point, such as x_k^o satisfies (1.4)). If there is at least one j for which (4.3) is not fulfilled, we will say that the beam defined byd_k is directionally non-admissible. It is seen that (4.3) only imposes a condition for the x_j that are bound-attained. In other words, any x_j that is not bound-attained, has by definition always an admissible direction number.

Now suppose that we are in any given point x_k^o on the boundary, i.e., any point x_k^o such that a certain number of the variables – basis variables or dependent variables – are bound-attained. And suppose that we attempt to move in a *free* preference direction, i.e., by putting $d_k = p_k$. If by so doing, we find that one or more of the variables get an inadmissible direction number, it seems indicated to include in the operation set at least one of the bound-attained variables with inadmissible direction number. In order to avoid difficulties with linearly dependent variables we will always include only one variable at a time. We may conventionally choose the one that has the *most inadmissible* direction number, i.e., the one which has the largest absolute value of the (inadmissible) d_j. If there are several j which give this same largest absolute value of the (inadmissible) d_j, we may decide by a random drawing. If we want to simplify, we could even decide by random drawing amongst all the variables with inadmissible direction numbers, regardless of the absolute size of these direction numbers. In a regular run of the algorithm we will most frequently only get one inadmissible direction number – if any – so that the manner of choosing amongst the variables with inadmissible direction numbers is not of great practical importance.

The above rule for deciding which variable to include in the operation set concerned the case where we previously had no variable in the operation set. It is, however, easy to generalize the rule to the case where we had already one or more variables in the operation set. To explain this generalization we must now consider in more detail what is meant by an operation set.

5. *A preference move on the boundary*

The fact that the bound-attained variables x_i ($i = \alpha, \beta \ldots \gamma$) are included in the operation set, means that for any further move we impose the conditions $x_i = x_i^*$, where x_i^* is the bound (either $\underline{x}_i$ or $\bar{x}_i$) which the variable x_i has reached in the point considered. This can be expressed by the equational constraints

$$(5.1) \qquad b_{io} + \sum_{k=bas} b_{ik} x_k = x_i^* \qquad (i = \alpha, \beta \ldots \gamma)$$

This represents ν (linearly independent) equational constraints in the movement of the point, ν being the number of variables in the operation set.

If $\nu < n-1$, the conditions (5.1) do not determine the direction of the move uniquely and we may add a supplementary condition. This supplementary condition can be defined in a number of ways that all lead to the same computational result. We may for instance say that we want to proceed in a direction d_k which is the *projection* of the preference direction p_k on to the manifold expressed by (5.1). Or we may say that we want to *maximize the increase* in the preference function subject to the conditions (5.1). Or again we may say that we want to determine a direction d_k which satisfies (5.1) and has *maximum correlation* with p_k (over the field of variation k). In all these cases we are led to the same solution, namely

$$(5.2) \qquad d_k = p_k + \sum_{i=\alpha..\gamma} B_i b_{ik} \qquad (k = \text{bas})$$

where B_i $(i = \alpha, \beta \ldots \gamma)$ are constant coefficients satisfying the ν linear equations

$$(5.3) \qquad M_{jo} + \sum_{i=\alpha..\gamma} B_i M_{ij} = 0 \qquad (j = \alpha, \beta \ldots \gamma)$$

where

$$(5.4) \qquad M_{ij} = \sum_{k=bas} b_{ik} b_{jk} \qquad \begin{pmatrix} i = \alpha, \beta \ldots \gamma \\ j = \alpha, \beta \ldots \gamma \end{pmatrix}$$

The coefficients B_i are called the regression coefficients.

The matrix M_{ij} of the equations (5.3) is positive definite (since it is a moment matrix), and it is non-singular (since the variables in the operation set are linearly independent). The solution of the equations (5.3) is therefore a straightforward computational process.

The *preference direction number* d_o which expresses the increment of the preference function along the beam d_k defined by (3.2), i.e., the constant d_o such that

$$(5.5) \qquad f = f^o + \lambda d_o$$

is equal to

$$(5.6) \qquad d_o = \sum_{k=bas} p_k d_k$$

We can prove that d_o is always *non-negative*. If the basis direction numbers are determined by (5.2), the direction numbers of all the dependent variables follow from (4.2). It is, therefore, now possible to say whether a certain operation set gives rise to a directionally admissible move or not, according to the criterion (4.3). For precision we may now speak of directional admissibility *with respect to* the given operation set.

If the move *is* directionally admissible, we take it as indicating that there is no need to include any further variable in the operation set. But if one or more of the d_j are inadmissible, we include *one* (and only one) of these x_j with inadmissible direction number, according to the same rule as we formulated at the end of Section 4.

We now have a complete and unambiguous rule for inclusion: If there is no need to include any more variables, we can make a preference move with the operation set considered. During such a move the direction numbers of all the variables in the operation set are zero, i.e., $d_i=o$ $(i=\alpha, \beta \dots \gamma)$. And all variables that are linearly dependent on the variables in the operation set, will also have their direction numbers equal to zero.

We can proceed along the beam now defined until at least one variable breaks out of the admissible region. Such a breaking-out variable must necessarily be one whose direction number is different from zero. The value λ_{stop} of λ which defines the breaking out point, is given by

$$(5.7) \qquad \lambda_{\text{stop}} = \mathop{Min}_{j} \lambda_j$$

where

$$(5.8) \qquad \lambda_j = \left[\begin{array}{c|c} \dfrac{\overline{x}_i - x_j^o}{d_j} & \quad \text{if } d_j > o \\[2em] \dfrac{x_j^o - \underline{x}_j}{-d_j} & \quad \text{if } d_j < o \end{array} \right]$$

By (5.8) λ_j (necessarily non-negative if the point x_j^o is admissible) is defined only for those j for which $d_j \neq o$. And by (5.7) λ_{stop} is defined as the smallest of these λ_j. There may be several j which produce this smallest λ_j, and if so, all of these variables x_j will simultaneously be breaking out variables.

If the preference direction number $d_o > o$ (it cannot be negative), the preference function will actually increase during the move considered. The new point is determined by putting $\lambda=\lambda_{\text{stop}}$ in (3.2) and (4.1).

In the case of a strictly linear preference function we will always find that the breaking out variable has an inadmissible direction number with respect to the operation set through which we arrived in the breaking-out point in question. If there are several variables which are simultaneously breaking out variables, they will all have inadmissible direction numbers with respect to the operation set through which we arrived in the point considered. Therefore without making any directional admissibility analysis in the new point, we can just as well immediately include one of the breaking out variables in the operation set.

In the mixed linear and quadratic case (as well as in the strictly quadratic case) it is no longer true that a breaking out variable will get an inadmissible direction number with respect to the operation set through which we arrived in the point considered. Nevertheless, it will not do any great harm if it is included in the operation set, since we will by the exclusion criterion get rid of it again if it does not belong in the operation set.

We must now consider the exclusion criterion in more detail.

6. *A sufficient optimality criterion. The exclusion criterion*

Our exclusion criterion will be worked out by considering the criterion for optimality.

Consider a point x_k which is admissible, i.e. satisfying (1.4) and such that in this point the (linearly independent) variables x_i ($i=\alpha, \beta \ldots \gamma$) are bound-attained, and further such that all the corresponding regression coefficients defined by (5.3) *are sign* correct in the sense that

$$(6.1) \qquad B_i \text{ is } \begin{cases} \text{non-negative when } x_i \text{ has hit its lower bound } \underline{x}_i \\ \text{non-positive when } x_i \text{ has hit its upper bound } \bar{x}_i \end{cases}$$

It can be proved that this point is an optimum point in the sense that the value which the preference function assumes in such a point can *not be surpassed* in any other admissible point if the computed d_o is 0.

This suggests a rule for excluding variables from the operation set: We exclude all variables which do not have a sign correct regression coefficient.

We now also have a complete and unambiguous rule for excluding variables from the operation set.

We can further prove that in the optimum the preference direction number d_o is exactly zero, while if d_o is not zero, we can not be in an optimum point, and can make a move which will increase the preference function.[5]

7. *The algorithm in the completely linear case*

The above considerations lead to the following algorithm which tells us what to do when we are in a certain point x_k^o and are in possession of a certain operation set x_i $(i=\alpha, \beta \ldots \gamma)$:

I. If not all B_i are sign correct: Drop from the operation set all x_i with sign incorrect B_i.

II. If all B_i are sign correct: Compute the corresponding basis direction numbers d_k and the direction numbers d_j for the dependent variables. If at least one of these direction numbers are inadmissible, include in the operation set one (and only one) of the variables with inadmissible direction number, according to the rule at the end of Section 4.

III. If all the B_i are sign correct, and all direction numbers are admissible, and the preference direction number d_o is strictly positive, make a preference move and determine the breaking-out point by (5.7)–(5.8). We have now actually changed the point, while in the cases I–II we have only made preparatory operations without changing the point.

In the new point we can start by an admissibility analysis with respect to the operation set through which we arrived in the breaking-out point considered, and decide by rule II if any new variable is to be added to this operation set. There is, however, – neither in the strictly linear nor in the mixed linear and quadratic or strictly quadratic case – no harm in proceeding in a simpler way, namely simply adding the breaking out variable (one of them, if there are several) to the operation set. If this variable does not belong in the operation set, it will subsequently be thrown out again by the exclusion criterion I.

IV. If all B_i are sign correct and all direction numbers admissible and $d_o=0$, we are in an optimum point. (In this case it will turn out that all d_k and hence all d_j are zero.)

[5] Compare Section 10 of the Tokyo-paper.

This algorithm covers all cases, and does so in an unambiguous way. We can therefore proceed according to this algorithm round by round until an optimum point is reached.

8. *The gradient phase in the mixed linear and quadratic case*

We will now consider what modifications in the algorithm are needed in the case where the preference function contains one or more quadrato variables.

In this case we simply start by a phase of the algorithm where we proceed round by round *in exactly the same way* as in the strictly linear case, with the following two modifications:

I. The preference gradient f_k^r in any point x_k^r that is the starting point for a preference move (r="round No.") is not put equal to the *constant* gradient p_k, but is determined by (1.7). This means that the direction numbers d_k – or, more precisely, d_k^r – will also depend on the starting point x_k^r.

II. In addition to the parameters λ_j defined by (5.8) we now compute the parameter

$$\lambda_o = \frac{\sum\limits_{k=bas} f_k^r d_k^r}{\sum\limits_{K=quad} \sum\limits_{H=quad} P_{KH} d_K^r d_H^r} \tag{8.1}$$

where f_k^r is the preference gradient in the starting point x_k^r for the preference move. And we let this λ_o *compete* with the λ_j – defined by (5.8) – when we look for the smallest of the parameters λ. This is done in order to cover the case where the highest value of the preference function along the beam may occur somewhere *before* we reach the breaking out point. In the strictly linear case, this possibility did not exist.

The phase of the algorithm where we proceed round by round in this way, may be called the gradient phase, or the gradient method. It is possible to *continue* all the time by the gradient method. If we do, the algorithm will *converge* towards the solution. But it may – and in general will – do so only through an *infinite* number of rounds. To illustrate that this is so, consider the case where the actual optimum point is situated somewhere in the two-dimensional plane generated by the two basis variables x_u and x_v.

Fig. (8.2) illustrates the contour lines of the preference function as well as the admissible region in this plane. We see at a glance that there is now an optimum point (and a uniquely determined one) in the *interior* of this admissible region, namely the point R. The two-dimensional plane thus considered is defined by keeping all the time a *constant* composition of the operation set, say the set consisting of the variables $x_\alpha, x_\beta \dots x_\gamma$, where $\alpha, \beta \dots \gamma$ are given affixes.

If we start in any admissible point, say A in fig. (8.2), and move along the preference gradient in this point (the preference gradient now being defined conditionally through the operation set $x_\alpha, x_\beta \dots x_\gamma$, which means that it should more correctly be denoted d_K^A), i.e., if we

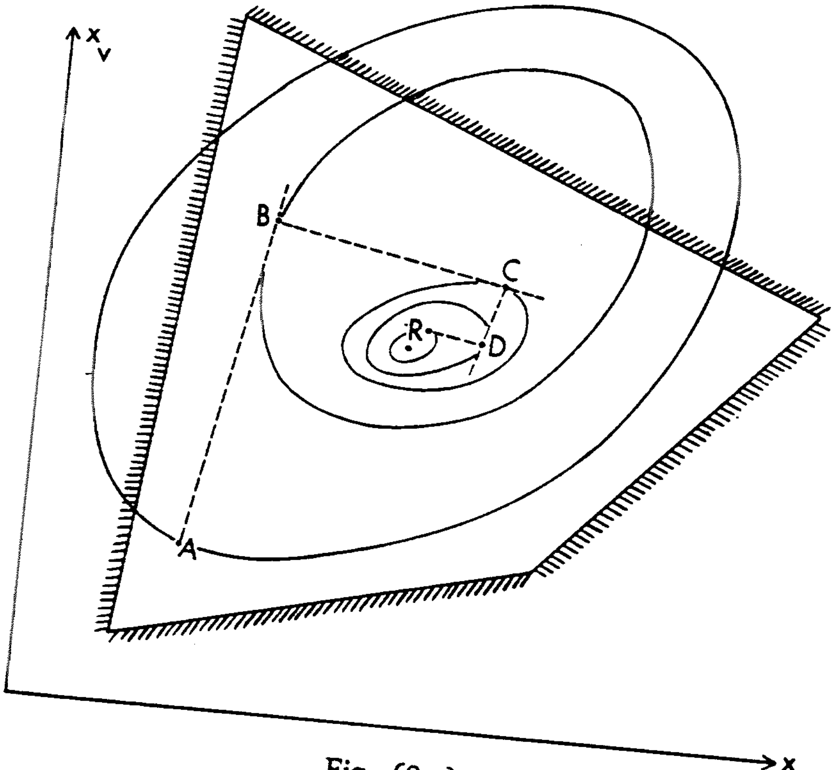

Fig. (8.2)

start to move orthogonally to the contour line that passes through A in fig. (8.2), we will reach the point of tangency B. This point of tangency will be localized by the fact that the parameter λ_o defined by (8.1) is now the smallest of the λ's.

From the point B we will – according to the gradient method defined by I and II above – again move along the preference gradient, and thus in fig. (8.2) reach the new point of tangency C. From this point we will again move orthogonally to the contour line and thus reach the point D. And so on.

It is seen geometrically in fig. (8.2) that this process *converges* towards the optimum point R. But we will not – except for an improbable coincidence – reach the optimum point R exactly in a finite number of steps.

It may happen that the admissible region has such a shape that during this process we will in one of the rounds not be stopped by a point of tangency (i.e. by the fact that λ_o is the smallest of the λ's), but by a *breaking-out* (i.e. by the fact that one of the λ_j is the smallest of the λ's). Such a breaking-out will, however, cause no trouble in the convergency. If it happens, we only have to continue according to the same gradient method, that was defined by I and II above. As long as the optimum point R is situated in the interior of the admissible region, the process will always converge towards this optimum point.

If we follow round after round the gradient method and we discover that the operation set remains *stationary* (i.e. with exactly the same variables in it) at least for a few rounds, we can suspect that we have reached a boundary manifold which is such that the optimum point is situated in it. And if not only the operation set remains stationary, but it also happens round after round that λ_o is the smallest of the λ's, we may suspect that the optimum point is located in the *interior* of the admissible region in the manifold defined by the stationary operation set, i.e., that we have a situation similar to the one exhibited in fig. (8.2). In this case we may say that we have a *tangentially* stationary operation set.

If the suspicion of a tangentially stationary operation set is correct, it is possible to switch from the gradient method to another method which will bring us directly, i.e., in a single round, to the exact optimum.

9. *The maxidirection phase in the mixed linear and quadratic case*

If the suspicion mentioned at the end of Section 8 is correct, we can obviously reach the optimum point in one stroke simply by *disregarding* the conditions expressed by the admissible region, and determining the point x_k^{opt} that will maximize the preference function under the *linear constraints*

$$(9.1) \qquad b_{io} + \sum_{\varkappa-lin} b_{i\varkappa} x_\varkappa + \sum_{K-quad} b_{iK} x_K = x_i^* \qquad (i=\alpha, \beta \ldots \gamma)$$

These constraints express the conditions that each of the variables x_i $(i=\alpha, \beta \ldots \gamma)$ in the (stationary) operation set shall remain constantly equal to the bound x_i^* on which this variable remained during the rounds of the gradient method that brought out the tangential stationariness of the operation set.

Proceeding in this way means that instead of continuing, say from the point C in fig. (8.2), along the gradient in C, we proceed now along a beam that leads *directly* towards R.

The way in which the x_k^{opt} are determined, is discussed in detail in Section 8 of the memorandum of 27 August 1960, and will not be further considered here.

When the x_k^{opt} are computed, the basis direction numbers d_k^r to be used when moving from the point x_k^r (the last point known in the sequence of rounds where the operation set remained tangentially stationary), are defined by

$$(9.2) \qquad d_k^r = x_k^{opt} - x_k^r$$

Apart from the special definition (9.2) of the beam along which the move is made, we proceed exactly in the same way as described under I and II in Section 8. The move is now called a ***maxidirection*** move.

If during this move λ_o turns out to be the smallest of the λ's, we will actually reach the optimum point. In this case this smallest λ will - apart from rounding errors - be exactly equal to 1 (if rounding errors occur, we modify the value so that it becomes exactly 1).

The problem will now be solved, and a new round, No. $r+1$ - the optimum test round - will only confirm that the optimality criterion is now fulfilled, i.e., that $d_o^{r+1} = 0$ and all $d_k^{r+1} = 0$ and hence all $d_j^{r+1} = 0$, within rounding errors.

If, on the contrary, we are during the maxidirection move stopped by a breaking-out (i.e. if one of the λ_j turns out to be smaller than λ_o), no harm is done. We now simply *continue by the gradient method* from the breaking-out point reached, and proceed round by round by the gradient method until we reach a new tangentially stationary operation set, and again can make a maxidirection move.

The occurrence of a breaking-out during a maxidirection move is in practice very seldom if the operation set from which we started was tangentially stationary during several rounds.

10. *The prospective form and the stage form*

Under favorable circumstances a considerable saving in computational cost can be achieved by using a refinement which may be called the *prospective form* of the multiplex method. It consists in using certain criteria which permit us to conclude that during a given preference move there are certain variables that *cannot* become breaking-out variables. This being so, we may save the computation of the direction numbers d_j for these variables. In big problems, particularly in problems with many dependent variables, this may mean a considerable saving.

The details of the prospective form of the method are discussed in a memorandum (in Norwegian) of 20 October 1960 from the University Institute of Economics, Oslo, and will not be further discussed here.

There is also another refinement which may be mentioned. It is possible to impose the condition that one is never allowed to solve any regression equation system – nor any other linear system – of higher order than a given number μ. This number μ may be fixed by consideration of the capacity of the automatic computer, or by the desire not to work with desk computers on linear systems of too high order. It is possible to proceed in this way if we *change* the composition of basis set each time we reach a situation where we would in the straightforward method have needed to consider a linear system of higher order than μ. This procedure may be called the *stage form* of the multiplex method, the consecutive rounds where the content of the basis set is not changed, being called a stage of the algorithm.

The details of the stage form are discussed in a memorandum (in English) of 19 October 1960 from the University Institute of Economics, Oslo, and will not be further discussed here.

11. *Non-linear bounds*

In all the preceeding parts of this paper we have assumed that the *bounds* are linear. Indeed, inserting (1.2) into (1.4) we see that these bounds appear as inequalities between certain linear functions of the basis variables.

If we drop the assumption of linear bounds, the problem becomes considerably more complicated. In the most difficult case the admissible region will now appear as *non-convex*. But even if the admissible region remains convex, a considerable complication is introduced.

The essence of the problem will now consist in defining in a plausible way a one dimensional *path* which is situated in a given ν-dimensional part of the (non linear) boundary, and which is such that by following this path, we will increase the preference function. I have given some thought to this problem, but not yet reached conclusions which warrant publication.