HANDLING MULTIPLE OBJECTIVES IN MACROECONOMIC LINEAR PROGRAMMING MODELS (MOLP)

An Advanced Project

Submitted by

Huda KHAMMASH

As a part of M.Soc.Sc. Degree (NEP)

University of Birmingham

October 1972

ACKNOWLEDGEMENT

I owe a debt of thanks to Mr. A. C. McKay, at the National Economic Planning Unit (N.E.P.) at the University of Birmingham, for his great help to prepare this project especially the computer programming part.

It is a pleasure to send my grateful thanks to Mr. D. Liggins (N.E.P.) for supervising most of the parts of my work and for acting as decision maker while carrying out the necessary calculations.

My acknowledgement to both Mr. McKay and Mr. Liggins for pointing out the path I should follow whenever a difficulty arises.

October 1972 M. KHAMMASH

HANDLING MULTIPLE OBJECTIVES

IN

A MACRO-ECONOMIC LINEAR PROGRAMMING MODELS (MOLP)

INTRODUCTION

Planning is always with the aim to attain some goals and/or to retain others. To start with, a planner has to formulate these objectives and goals to determine their form and hence the techniques to be used and applied.

When dealing with national planning, one faces problems on the macro-economic level, where certain global objectives are to be achieved such as decreasing and ultimately eliminating unemployment, improving the standard of living, reducing inequalities in income distribution, the achievement of the best possible co-ordination among the plans of the various planning bodies beside other objectives. In a developing country the central plan objectives extend to the field of specifying the means for generating resources, allocating these resources and establishing policies to determine how these resources are to be used. These and other objectives are to be achieved under specific limitations at the top of which is the availability and the requirements of resources which restrict the attainment of these conflicting objectives, conflicting in the sense that all the objectives compete for the same scarce resources.

These objectives can be joined in a single general objective when attained all the single objectives will be achieved, but when conflicting objectives do exist as the case in reality, each must be considered separately but at the same time taking account of all the others.

Another possibility is to choose to achieve the most important objective considering the others as imposing limitations on that objective or in other words as constraints. But in both alternatives one faces the problem of pre-assigning weights to the various objectives, which is a value judgement problem and represent a difficulty in this concern.

Talking about the techniques to be applied, one finds these days that there are increasing attempts by planners, especially in the developing countries as well asin the already developed countries, to apply borrowed techniques from OR's to National Planning. The most commonly used are the optimization techniques including linear, both integer and dynamic, and non-linear programming, depending on the decided form of the objective function or what can be called the criterion function and the constraints whether they are linear or non-linear. One can mention the return function from most the activities involved in national planning as an example of non-linear objective functions.

These functions if not able to find the best possible solutions due to poor quality of data especially in underdeveloped countries, they can provide a wide range of alternatives and basis for policy makers to choose among these alternatives.

From above one comes out with the conclusion that for a plan to be feasible and successful it must include a set of the most important goals and objectives, the optimum values of which are to be attained for accelerating the rate of growth of the economy under consideration taking into account the expected limitations, including the technical as well as the economic and social limitations.

This means that an optimization technique is required to cope with the multiplicity of objectives to be attained by the economy, without the need to a general objective, or to assign weights to the available objectives to determine the most important one which is a difficult task.

In this project I am intending to deal with the problem of the multiple objective functions in linear programming (MOLP), and the recent techniques, that can be used for solving a linear programming problem (LP) characterized with the multiplicity of objectives.

I will concentrate on the most recent approach which is called the STEP method (or the STEM) in which the aspect of sequential decision making is incorporated, and may be considered as representing a man-model system. The full description of STEM is given in the third chapter together with an evaluation of this approach.

The main aim of this project is to modify an already written computer procedure to take into account the multiplicity of objective functions and to reach at an acceptable solution for all the objectives simultaneously. This procedure will be written according to the computational phase of STEM. A report on the modified computer procedure will be presented at the very end.

The first chapter will give sufficient details of the characteristics of a large scale (macro-economic) model based on accounting identities and other standard relations with multiple objective function of a global nature. This model is an Import substitution model for the Nigerian Economy.

Finally the second chapter will go through all the various attempts to reach at an acceptable approach to solve any MOLP.

CHAPTER ONE

MACROECONOMIC MODELS WITH MULTIPLE OBJECTIVES

The property of the multiplicity of objectives or goals is a realistic property of planning, or in other words it is the essence of planning in reality.

The main purpose of this project was to go through the main features of a macroeconomic model which has this property, to examine the various methods available to solve such a model and then to apply the best to attain a solution to this model. But, unfortunately, in spite of finding a model which gives a good example and agreeing to an algorithm to solve it, a solution could not be found due to the lack of data. Instead the computer procedure written for this algorithm will be tested on a small hypothetical model which was given by Peter Clark in his book, 'Planning Import Substitution' and it was solved for a single objective. My aim is to compare the results obtained for this small model with one objective function and with several ones. I have constructed the extra objectives to resemble those in the large scale model explained below.

This large scale model is an Import Substitution model for the Nigerian economy.

1. The Main Characters of The Import Substitution Model*

This model was constructed mainly to reach the optimal allocation of resources when there is a choice between domestic and foreign supplies. A second aim of establishing this model is to predict the effect of a technological change in some sub-sectors on the activity of the major sectors in the economy.

---

* Peter B. Clark, Planning Import Substitution Vol. 68, Amsterdam 1970, pp. 62-115.

It was established as a linear programming problem with the necessity to keep all the assumptions related with linear programming such as the proportionality assumptions and the additivity assumptions. For all that the model does not include any direct factor substitutions or technological external economies, and incorporates only constant returns to scale. Thus one finds that these assumptions impose unrealistic restrictions on a developing economy which needs to transform the structure of its production by adopting modern techniques, where productivity will increase thus releasing factors of production that can be used in other sectors. But in the case of the Nigerian economy these restrictions of using LP are not very serious, since the relative prices in Nigeria did not change to a large extent, and the price elasticities are almost insignificant. Additionally the elasticity of substitution among factors is close to unity, i.e. no important substitution takes place.* Yet to simulate substitution of inputs, imports and factors in the aggregate technology as the import substitution starts, an explicit choice of technology is incorporated into the model. This was done by allowing new activities and new uses for resources as the import substitution starts.

The remaining properties of the model are concerned with its structure, where it is based mainly on eleven aggregate sectors in spite of the fact that the Nigerian Input/Output tables were published for 20 sectors.

To make the model more detailed, imports are classified into competitive imports which can be (or are) produced domestically, and non-competitive imports which cannot be produced domestically due to technical infeasibilities such as the lack of resources, or due to

---

* P. B. Clark, p. 68.

economic infeasibilities, for example when the size of the market is smaller than the minimum scale of production required for that activity to be profitable.

After this classification of imports, new input/output tables can be constructed to take into account the potential activities not yet existing. These industries or activities will produce the competitive imports. To get these tables, a number of feasibility studies must be used, data on pre-investments can be of a great help and finally use can be made of the input/output tables of other developing countries. The new tables provide the model with the possibility of incorporating new lines of production.

To allow for the competitive imports in the model, 14 sectors related to these imports were disaggregated from the 1959 input/output tables* raising the number of sectors in the model to 25 sectors producing in 1959, since the added 14 sectors has already started at a small scale i.e. the major part is still imported.

Then 61 more potential sectors in the field of competitive imports were added to the new tables, hence the final total is 86 sectors. A production function was added whenever possible for the purpose of completeness.

2. The Relationships Involved in the Model

For a better understanding of the nature of the model a closer look at the type of relationships involved is needed.**

---

* P. B. Clark, p. 71

** ibid., pp. 78-97.

2.1 The Set of Constraints

The model involves 269 constraints. The first main set of these constraints are those concerned with the balance relations which take the following form:

$$X_i + M_i \geq \sum_i a_{ij} x_j + C_i + I_i + R_i + \Delta S_i + \overline{E}_i$$

where there are 86 such constraints.

X_i = domestic output of the ith sector

M_i = competitive imports used by the ith sector

a_{ij} = I/O coefficients

$\Sigma a_{ij} x_j$ = total intermediate use of outputs

C_i = final consumption of government and households

I_i = net fixed capital formation using the ith commodity

R_i = capital replacement using the ith commodity

ΔS_i = changes in inventory of the ith commodity

$\overline{E}_i$ = the amount exported of good i and it is determined exagenously.

The components of these relations were estimated using the available data and information. For example the consumption demand C_i was estimated using Engel's curve and assuming a constant expenditure elasticity for each good during the plan period. A linear approximation of the consumption function was incorporated into the model instead.

The total consumption was calculated taking into account the part of consumption supplied from non-competitive imports and the value added by the government. The non-competitive imports supplied for final consumption were expressed as a function of total consumption using as coefficient the ratio between the non-competitive imports used in year 1959 and total consumption for the same year.

To ensure an acceptable level of total consumption, taking into account the population rate of growth (p) and the growth rate of consumption (g), a minimum requirement was incorporated. This requirement is expressed in the form of a lower bound on C

$$C \geq \overline{C}\ (59)\ (1 + p)^t\ (1 + g)^t$$

where $\overline{C}$ (59) is the base year total consumption and t is the planning period.

Next investment demand was incorporated through a set of capacity constraints to account for idle capital in the final year. The incremental capital/output ratio of each activity was calculated assuming that capital in the base year is used at full capacity.

From the terminal capital stock K_j the net investment is calculated as ΔK_j during the plan period. ΔK_j is translated into a flow of capital goods to the jth activity from the ith activity using the stock flow conversion factors to arrive at the net investment demand. Since the great part of capital was still imported and not considered as competitive imports, due to technological factors, an allowance was made for this part of investment.

Another set of relations was used to account for the demand for capital replacement using the stock flow conversion factors. Similarly the demand for replacement is supplied domestically and from non-competitive imports. The same procedure was adopted when calculating inventory accumulation. The final gross investment consists of the total of the net investment, replacement and the demand for inventory accumulation.

From above one finds that the demand for imports consists of two parts; competitive and non-competitive. The competitive imports are given as a single total, while the non-competitive imports are shown in greater detail, where they are classified into four parts; total imports for intermediate use, imports for final consumption, imports for investment, and finally for capital replacement. Total imports included the import duties as well.

A separate model of projection was used to calculate exports, $\bar{E}$, which were considered as an exagenous variable. In this projection model, the expected international prices in the final year were used to evaluate exports. These international prices have incorporated the projections of F.A.O. and I.B.R.D. assuming a constant rate of growth. Alternatively a linear function of foreign exchange revenue could have been used in this projection model if it was possible to demonstrate that the Nigerian output of the main exported products were affecting world prices.* The value of exports included the export duties as well as the Marketing Board surplus and the Petroleum and Mining Royalties.

Then G.D.P. was expressed as the total of the previous items after adjusting for import duties as a part of government value added.

$$Y \equiv C + I + \bar{E} + (d_M - 1)\, M$$

where d_M is the average rate of import duties.

---

* P. B. Clark, p. 88.

To guarantee the economic feasibilities, two constraints were included to ensure an acceptable balance of payments gap. These constraints impose upper limits on the import surplus and at the same time allow for capital inflow determined endogenously in these constraints. The two constraints took the form.

$$F \geqslant (1-d_M)\ M - \overline{E}$$

$$F \leqslant \overline{F}$$

where F = the foreign exchange requirements to cover the balance of payments gap, and it varies with imports since $\overline{E}$ (the exports) were fixed, and $\overline{F}$ is an upper bound on imports surplus.

To set upper limits on saving, saving and the marginal rate of saving ($\overline{t}$) were calculated using the national income statistics.

$$T = I - (1 - d_M)\ M + \overline{E}$$

$$\overline{t} \geqslant \Delta T/\Delta Y$$

where ΔT and ΔY are the change in saving and G.D.P. during the plan period respectively.

The importance of these constraints emerges from the fact that a larger share of the supply of investment funds should be obtained from Nigerian savings due to declining foreign aid and private foreign supplies of capital.

An additional set of constraints to restrict the output of some sectors was formulated in the form of upper bounds to represent diseconomies of scale, and lower bounds to impose exogenously determined minimum production levels to ensure that those activities appear in the optimal solution.

The final set of constraints is concerned with the limited resources mainly skilled labour. Since there were no estimates of potential supply of labour it was assumed that in the short run, any shortage of skilled labour could be imported. This shortage would be reflected in lower efficiency since new industries were not curtailed. In the long run local supply will be stimulated by the high costs in terms of foreign exchange.

To avoid the effect of the proportionality assumption labour needs to be classified into completely independent categories, but this was a difficult task due to the lack of the data needed. The effect of the proportionality assumption expresses itself by making some constraints, especially labour constraints, binding while the others still have some slack resources leading to zero shadow prices for the latter. This effect is due to the fact that such an assumption allows no direct substitution between the factors. A way out of this problem is that whenever labour constraints become binding, imports of labour are allowed through an importing activity in spite of the high costs. This can be done by combining the value added of particular labour categories in a constraint. Another alternative is to include this value added in an objective function.

2.2 The Objective Functions

The objective function is the criterion which determines the activities that should appear in the final solution. This criterion will be subject to a set of constraints which express the desired economic, social, and technological feasibilities that should be taken into account, in other words this set of constraints will specify the feasible region within which any value of the objective function is acceptable or within which the best value of the criterion lies.

In this model a set of eight such criteria was chosen to represent typical goals for development planning and to give a good example of MOLP in this field. These objective functions were valued for the terminal year and they are as follows:

i) To maximize G.D.P. with the aim of accelerating the rate of growth.

ii) To maximize total consumption which is taken as an approximation of the social welfare, which is to be maximized as a social requirement.

iii) To maximize investment, since the need to encourage import substitution requires an increase in the productive capacity.

iv) To minimize total imports, this being the aim of any developing country which depends to a large extent on imported goods with the need to provide large amount of foreign exchange. Thus minimizing imports will have two purposes; to minimize the foreign exchange requirements and to encourage import substitution at the same time.

v) To maximize total value added to help in achieving the main goal viz: import substitution.

vi) To maximize G.D.P. less total imports mainly to ensure lower foreign exchange requirements and a higher rate of growth simultaneously.

vii) To minimize total value added plus total imports with the purpose of minimizing both domestic and foreign costs, but this will minimize profits as well since value added includes profits along with costs and other components. Besides this target, minimizing costs could be attained by the sixth objective function to a large extent. Finally the presence of this objective function shows the idea of conflicting objectives clearly where it conflicts with the fifth objective function which aims at maximizing value added.

viii) To minimize competitive imports to allow for the increase in import substitution production.

3. <u>Remarks On The Model</u>

This model is a large scale one. It includes 351 activities, broken down into five sets, each of the first four sets consisting of 86 activities. These four sets belong to total output, imports, net investment and demand for replacement for each of the 86 sectors of the model. The fifth set includes 7 variables which are total imports, total consumption, total gross investment, G.D.P., foreign exchange requirements, total demand for inventory and total demand for capital replacement. The first five variables are aggregates while the last two consist of non-competitive imports.

The number of constraints add up to 269 consisting of five sets as follows:

1) 86 balance constraints.
2) 86 capacity constraints.
3) 86 replacement constraints.
4) 6 resources constraints imposing lower bounds on consumption and upper bounds on each of the balance of payments gap, capital inflow, marginal rate of saving, and upper and lower bounds on output levels.

5) 5 identities to express imports, investment, G.D.P., non-competitive imports for stocks and for capital replacement.

Turning to the objective functions, we find that they are typical goals for most of the developing countries and they clearly conflict with each other.

It seems that there was no attempt to solve this model as a MOLP, on the contrary it was suggested that it could be solved by optimizing it for a single objective putting all the remaining ones as constraints with certain acceptable minimum levels. This is similar to the efficient surface approach suggested by Hanssmann with one difference, that Hanssmann's approach is based on integer programming. The advantage of this approach is that it allows for policy analysis where the sensitivity of the model will be increased and will facilitate the testing of the effect of the available alternative policies. It was suggested that the single objective should be to maximize total consumption. This approach has the disadvantage of requiring a set of acceptable levels for the other objectives.

CHAPTER TWO

SIMPLE DESCRIPTION OF THE METHODS SUGGESTED TO SOLVE THE MOLP PROBLEM

1. Some years ago, planners who applied optimization techniques were criticized for ignoring the problem of conflicting objectives and the aspect of sequential decision making. Hence there were many attempts to overcome this drawback in the utilization of the technique, although these attempts could not reach to a method to solve a MOLP problem by solving for all the objectives simultaneously by finding unique values for the activity levels that optimize all the objectives under consideration at the same time. That was due to the fact, which must be taken into account when examining any approach in this respect, that there exists no feasible solution such that all objective functions can simultaneously take their optimal values within the specified feasible region, or such a possibility is rare.

From the above argument one reaches the conclusion that any MOLP is an LP model requiring several simultaneous objective functions and requiring an efficient procedure which will obtain the best possible solution, in the decision maker's (DM) point of view, where this solution can be called the compromise solution.

In the following sections and chapters a brief description of the already available and suggested procedures is produced together with the drawbacks and advantages of each.

2. One of the pioneers in this respect was Fred Hanssmann* who prepared a multiple objective model for the Arab Republic of Egypt in 1960 and proposed an approximation method, where solutions are insensitive to errors in the data and depend on certain ranking ideas.

---

* F. Hanssmann, Operations Research In National Planning of Underdeveloped Countries. The National Cash Register Company, Dayton, Ohio, 1960.

Hanssmann thinks of the resources allocation problem as a project selection problem on a nationwide level, where this approach has the advantages of taking into account technological feasibilities and future opportunities, besides giving the best compromises or alternative programmes among the conflicting objectives. Finally the DM will have a better chance to select a suitable compromise.

This approach can be used by developing countries to achieve faster development under a state controlled economy where the free enterprise system can not be relied on due to the lack of sufficient enterprises to ensure that the price mechanism solves the resources allocation problem efficiently. The application of this approach must be preceded by certain planning decisions concerning the availability of resources, the limits on them and the problem of generating new resources by deciding on the saving ratio and the limits on the indebtness of the country to the rest of the world.

To get better results, the responsibility of resources allocation must be delegated to a number of ministries with the need to achieve a certain degree of coordination which helps in avoiding any idle projects due to lack of raw materials, or any excess capacity or shortage of capacity due to insufficient investment which may lead to inflation. This coordination problem could be eased by using the Leontief model especially as the target setting approach prevails in most of the developing countries where a vector of final demands is set as a target. Solving the I/O model would then give a vector of the required levels of outputs. The target setting approach is useful due to its psychological effect. But this psychological effect will be more accurate if the targets are a result of an optimization approach, which needs more data and more technical effort that may not be available in most of the developing countries.

When we reach the stage of optimal project selection we can automatically determine the optimal interaction between the various sectors of the economy, the profitability of investment in these sectors and hence the optimal allocation of the available resources.

When applying this project selection approach a good appreciation of the interdependencies between the various projects, and the location of these projects is required. For simplification Hanssmann suggests that a group of Key projects should be considered non-decisional once selected and any dependent projects should be taken as one project to achieve determenacy of projects characteristics, otherwise it will be a complicated problem.

Moving to the sequential decision making problem, Hanssmann assumed that each year a list of projects is chosen taking into account the future properties of project lists in the form of assumptions leading to the consideration of alternative possibilities. Obviously this is the essence of long range planning. For simplification "it can be assumed that every year a project list of the same economic characteristics and technological requirements structure is available implying no rapid technological change but an increase in the average quality".* These lists are obtained by duplicating the characteristics of the presently available list for the planning horizon hence the number of the lists will equal the length of the planning horizon.

A tentative plan can be drawn up by selecting projects from these lists. The process is repeated every year after implementing the 1st year's part shifting the planning horizon by one year ahead.

The main feature of this approach is the ability to cope with several alternative outcomes.

---

* F. Hanssmann p. 235.

In this approach one has to consider the restrictions which emerge, firstly from the necessity to take into account the need to achieve a certain level of coordination among the departments to which the planning responsibility is being delegated, and secondly from the limits on the resources. These restrictions can be classified into three groups depending on the aim of incorporating them into the problem in hand. Some restrictions have the effect of ensuring economic feasibilities in the field of foreign credit or domestic savings, others ensure technological feasibilities such as input/output relations, and finally there is a group of restrictions that ensure social feasibilities, for example to achieve an acceptable minimum consumption level.

After satisfying the constraints one will be left with a smaller degree of freedom of selection when considering the objectives and the targets to be achieved. The major objectives on the national level deal with national income, employment, health, education, housing, etc. When we reach this stage a number of problems arise though of different degrees of complexity. First of all there is the problem of deciding on the criteria to be optimized, is it the present value over the planning horizon or the average annual growth rate of the aspect under consideration. Secondly there are the complexities arising from the multiplicity of objectives where these objectives are both incomensurate, i.e. quantifiable but no common measure is available, and conflicting, i.e. competing for the same scarce resources.

When the problem is established and is optimized, the optimum will be a compromise and one may need to repeat the optimization process over and over until the best compromise, which is a matter of value judgement by the D.M, is achieved.

In his article, Hanssmann mentioned two approaches that are suggested to deal with the problem of multiple objective functions. These approaches differ as to when the value judgement is to be introduced in the optimization process. Briefly they are as follows:

i) The Pre-assigned weights approach:

Where the value judgement as the title indicates is right at the beginning of the process. Each objective will be assigned a weight depending on its importance from the D.M point of view, then the problem will be optimized using a single objective which is a weighted sum of all the available objectives. The disadvantages of this approach arise in the difficulty of preassigning weights to the various objectives.

An example of this approach can be found in the Multiplex Method* developed by Professor R. Frisch, where he calls this weighted sum of the objectives or targets, the Preference function which is a linear function of the variables for which the model is solved, including both instruments and targets and related to the instruments being used, hence the preference function can be expressed in terms of the instruments by substitution. It expresses the preference ordering decided on by a responsible group, sometimes after negotiation. This group consists of elected representatives, experts, pressure group leaders and officials.

On the other hand a preference direction need to be defined, such that any movement in the feasible region in this direction will cause the largest change in the preference function per unit of movement which means that the Multiplex Method finds the optimum by moving mainly

---

* C. J. Vaneijk and J. Sandee, Quantitative Determination of An Optimum Economic Policy, Econometrica, Vol. 27, January 59, No. 1, pp. 1-13.

through the feasible region - while the simplex method moves in steps from corner to corner along the boundary - and every time the constraints planes are actually hit by a preference move a combination of these constraints and the ones immediately beyond the hit point must be tested for compatibility and completeness by means of regression analysis* since those might be among the constraints that define either the optimum or a corner near to it. Further preference moves can be made to reach at new such combinations. The larger the number of instruments dealt with the more difficult this approach will become and the simplex method will be preferred. On the other hand applying the Multiplex Method will make it easier to locate the major constraints very quickly.

To obtain a welfare function to work with, one has to estimate the data for a future period for which the optimum economic policy is required,next a forecast can be made. A provisional programme can be reached at assuming no changes in economic policy. Around this programme a preference function can be determined by reaching at the coefficients of the targets first. Each coefficient is the average of the barter terms** corresponding to each target against all other targets including itself. A knowledge of the policy makers marginal rates of substitutions is essential.

To apply linear programming to the welfare function it must be linear, and if it is not it can be represented by a linear approximation within a reasonable interval, and the continuously curved surface is replaced by a collection of contiguous hits of hyperplans which are assumed to form a convex set as is used in linear programming.†

---

* Van Eijk and Sandee, op. cit., p. 11.
** Ibid, Idem, p. 3.
† Ibid, Idem, p. 3.

This can be done for all the neighbouring facets of the welfare function to determine the welfare function in each of these facets hence the optimum policy related to that function in the facet concerned, without the need to investigate more than those facets that border on the first optimum, then the facets bordering the best of the latter programmes obtained. This is a tedious task, but defining a preference direction will cut down the number of facets to be investigated, without the need to identify all effective constraints in each facet, and this can be done by investigating the adjacent facet as soon as an effective constraint has been found to see whether it is still effective in that facet, and if it is not this new facet will offer a better programme and hence worth working on.

Frisch used a continuous but partitioned welfare function, where no products of the target variables appear in it, hence a move between the intervals needs changing only one coefficient.

This method was applied in Netherlands by the Central Planning Bureau in 1956 to determine the optimum economic policy to overcome the balance of payments and government expenditure deficits. Six targets were considered; the maximal real national income, a high stable level of employment, a balance of payment equilibrium, a high level of investment, a stable price level, and a reasonable income distribution. Six instruments were used which meant that the optimum will be characterized by six variables being at their boundaries.

ii) <u>The Efficient Surface Method:</u>

Here the value judgement is introduced at the end of each optimizing iteration, i.e. after obtaining each compromise. The efficient surface is obtained by maximizing one of the objectives for fixed levels of all others.

This approach is more suitable "for objectives with rather general convexity properties, and the efficient surface also has the stronger property of maximizing each objective for fixed levels of all others".* The resulting surface can be considered as a list of all possible efficient compromises, and it is presented to the D.M to choose the final compromise.

One of the advantages of this approach is that the D.M. will be able to get a better appreciation of the problem in hand hence he will decide on the best combination of fixed levels, on the other hand it has the disadvantages of the previous one, where value judgement is also needed at the beginning of the optimization process where the D.M. has to decide on the most important objective to be optimized and then to decide on a set of acceptable levels for the remaining objectives.

Then, Hanssmann gives a final form of the problem to be optimized according to the efficient surface method, assuming a 10 year planning horizon and m projects. This problem will be for example:
maximize national product 10 year hence

s.t. 1. the assumption of identical project lists

2. the economic, technological and social feasibilities

3. fixed levels of all objectives except national product.

The optimization is repeated for different sets of fixed levels to get the efficient surface.

<u>How to Optimize such a Problem?</u>

The use of integer programming is suggested, where an acceptance variable Z_K is used where

---

* F. Hanssmann, p. 237.

$$Z_K = \begin{cases} 0 \text{ if project K is rejected.} \\ 1 \text{ if project K is accepted.} \end{cases} \qquad K = 1, \text{---}, m.$$

The economy can be divided into n categories of commodities and services that are related to final consumption and to the development projects.

Then the set of suggested coefficients to be considered is given in more detail. These coefficients mainly deal with the addition in the capacity of the various categories when a given project is selected, as well as dealing with the total domestic usage from each category caused by the selection of each project.

Finally a set of non-decisional projects is decided upon, and this consists of the projects which affect the whole set of the possible projects, hence they are selected at once and considered non-decisional. For these projects a set of special coefficients are incorporated separately.

Then a mathematical formulation is given in the same paper* with the whole set of constraints. Assuming L objectives to be achieved, the general form of this problem is as follows:

$$\max A_{10} + \sum_{K}^{m} a_{1K} Z_K$$

s.t.

$$1.\quad A_{h0} + \sum_{K}^{m} a_{hK} Z_K = V_h \qquad h = 2, \text{---}, L$$

$$2.\quad C^t_{i0} - U^t_{i0} + \sum_t e'_{iK} \; Z_K \geqslant 0 \qquad t = 1, \text{---}, 10$$

$$3.\quad E^t_0 + \sum_K E^t_K Z_K \geqslant \overline{E}^t \qquad i = 1, \text{---}, \bar{n}$$

---

* F. Hanssmann p. 239.

where $E_K^t = \sum_{r=1}^{t} \sum_{i=1}^{n} (C_{iK}^r - U_{iK}^r) = \sum_{r=1}^{t} e_{iK}^t$

4. $Z_K = 0$ or 1

where A_{10}, A_{ho}, C_{io}^t, U_{io}^t, and E_o^t is the set of special coefficients corresponding to the non-decisional projects. The explanation of the above set of constraints is as follows:

1. is the set of values of the remaining L-1 objectives
2. is the set of technological and social restrictions for the $\bar{n}$ categories for which no imports are possible.

and 3. is the set of economic constraints and deals with the net creditor position.

To build up the efficient surface this problem must be optimized for a number of vectors V_h.

Hanssmann did not solve this problem in the way he suggested but he gives an approximation method which depends mainly on the ranking approach, using the ratios between the project pay off and burden for ranking, which has the advantages of producing solutions which are not sensitive to errors in the data. But this sort of problem will require a highly complicated ranking scheme due to the multiplicity of objectives and scarce resources as well, and due to the fact that the availability of resources is spread out over time. A way out of these complications is to group the projects according to the primary objective thus taking into account the indirect pay off and burden as well.

This approximation approach calls for the necessity of a capacity to build up curves for systems and their breakdown into projects. These curves show the expansion path of these systems.

The efficient surface approach, which seems to be efficient, has not been applied, up to my knowledge, to any such MOLP problem, perhaps because of the numerous requirements they call for and the several complications they raise when building the model according to the specified procedure.

3.1 The Restricted Bargaining Approach*

Meanwhile there were several attempts to deal with this problem, i.e. the MOLP problem, since the solutions obtained by the available different methods were not satisfactory due to the multiplicity of these solutions giving rise to a "Pareto Optimal set". It was found that a random element needed to be introduced in order to get a unique solution.

Since on any organizational level there are inter-related as well as conflicting objectives, the Organization Theorists suggested that decision making in any organisation must be based on a bargaining process among the group members responsible for each of the objectives under consideration, these may be the various planning departments or the interest or pressure groups etc. This process will be affected by the bargaining powers of these members.

Due to the fact that the bargainers vary in their bargaining power, the process should be restricted in order that it produces a solution which is a balanced or a compromise one, otherwise the negotiation may come to a deadlock or an unbalanced solution will be obtained, i.e. the solution will fulfil the interests of certain groups at the expense of the others. In case the dispute was

---

* B. Contini and S. Zionts, Restricted Bargaining for Organizations With Multiple Objectives, Econometrica, Vol. 36, No. 2, April 68.

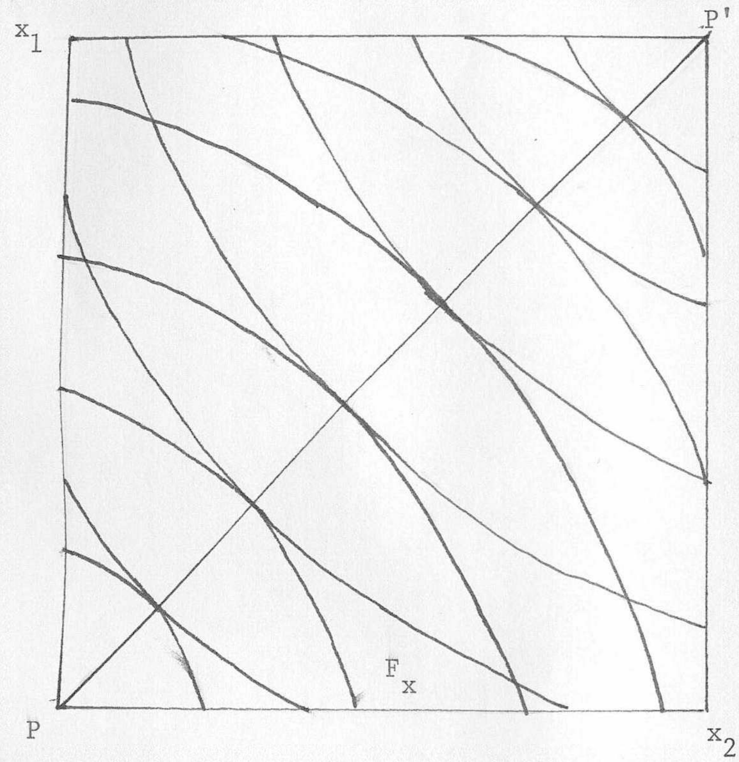
submitted to an arbitrator the chosen solution may not be on the Contract Curve of Pareto Optimals due to the lack of information, available to the arbitrator, about the utility functions of the bargainers.

For this process to be restricted an arbitrator, with the authority to prepare what is called an imposed solution and the authority to enforce it, is needed. This imposed solution, usually is never enforced but is used as a threat or an incentive to the bargainers. Besides a time limit must be specified within which a negotiated solution should be reached, otherwise the imposed solution, which might be an unbalanced solution and not acceptable by the bargainers, will be enforced.

The basic idea in imposing these restrictions is to provide incentives for all the group members to be engaged in a process of concessions until some agreement is reached. The incentive consists of the threat of a preannounced "imposed" solution.

3.2 This process has many advantages; first of all it yields a negotiated solution in most of the cases, this solution has the property of being uniquely defined, Pareto optimal and balanced, where it guarantees an acceptable lower bound for each individual utility. Secondly it is based on a democratic decision making process since the individual autonomy of each D.M. is retained, where each D.M. will have a complete preference ordering over the space of decision variables, represented by a continuous and concave utility function, besides each bargainer can suggest a set of decision vectors that satisfy a certain minimal utility requirement and can make certain concessions depending on his cost-benefit considerations until a feasible decision vector satisfies all the minimal requirements of the involved D.M.s.

The Contract Curve of Pareto Optimal



PP' = The Contract Curve.

F_x = The Feasible Region.

Then at each iteration the set of decision vectors is tested to ensure consistency with the feasible region, which may be stored in a computer for easier test, until a feasible point is reached, using the set of suggested decision vectors, and at each iteration each bargainer is expected to reduce the minimum acceptable level of utility, since due to the concavity of the utility functions no better solution can be achieved without making concessions. Finally, the aforementioned mechanism is but one, and any other suitable mechanism can be designed for the scheme of Restricted Bargaining according to the specific properties of the system under consideration.

On the other hand this process has many disadvantages where the negotiated solution is balanced only relative to the bargaining powers of the various members which affect the solutions obtained to a large extent since some of the bargainers have higher bargaining power than the others. Besides the imposed solution has a role in the process and the choice of this solution affects the minimum level of utility guaranteed to each party. Since the arbitrator who usually suggests this solution does not have enough information, the imposed solution will not be desirable, and to avoid imposing it, the bargainers will do their best to reach a better one. The effect of the imposed solution will increase the nearer it is to Pareto Optimality and the negotiated solution will be more predictable. Hence the imposed solution can be chosen in such a way to direct the negotiated solution into a predetermined direction with some features that are desired but not necessarily related to its desirability in terms of the bargainers preferences. This can be considered as an advantage as well if the chosen solution is directed to achieve social and economical benefits. Yet to select such a solution is a technical problem and needs an accurate technique that will be able to find this solution as near as possible to Pareto Optimal with the need to know the individual preferences.

The approach to be explained in the coming chapter can be considered as a technique to obtain an acceptable solution (X^s) that can be used by the arbitrator to restrict the bargaining game.

3.3 A simplified mathematical model can be constructed*

1. $U_i(X_i^o) = \sigma_i$

2. $U_i(X) \geq \sigma_i - \int_0^t g_i(\tau)\, d\tau = Z_i(t)$

3. $L(t) = \{X \in F_X \mid \bigcap_{i=1}^{q} [U_i(X) \geq Z_i(t)]\}$

where $g_i(t) = \begin{cases} \theta_i(t) > 0 \text{ if } L(t) = 0 \text{ and } Z_i(t) > U_i(X^s) \\ 0 \text{ otherwise} \quad i = 1, ---, K \end{cases}$

X^s = the imposed solution

F_X = the feasible region

$\theta_i(t)$ = a decreasing Convex function $\ni$ $g_i(t)$ will decrease gradually

$g_i(t)$ = the rate of concession which converges to zero, and equals zero when the yielded utility = the utility of X^s, provided that at each stage the ith utility of not conceding = the utility of X^s, and the utility of making a concession is not smaller than that of X^s. To ensure all that, another set of constraints can be imposed. $g_i(t)$ reflects the ith bargainer or D.M. bargaining power, the higher the power /the smaller the rate of concession hence the bargaining power will be reflected in the final agreement.

---

* Contini and Ziouts, pp. 403.

The set of equalities in 1), each shows the utility of the ith bargainer from his optimal vector X_i^o. The set of inequalities in 2) consists of three parts; $U_i(X)$ is the ith bargainer utility, $\int_o^t g_i(\tau)\, d\tau$ is the total utility yielded in the time intermal (0,t), and $Z_i(t)$ is the minimum utility acceptable to the ith bargainer. And finally, 3) shows the set of feasible decision vectors for which all constraints are satisfied or in other words the negotiated solution.

3.4 Although the process seems to be easy to apply, there are many difficulties that one has to face, among which are the difficulty of ensuring the acceptance by the bargainers of an arbitrator's authority, the evaluation of the utility functions of the bargainers, the difficulty in reaching the desired negotiated solution when all utility functions are strictly concave, and finally the need for a technique that will produce a desirable imposed or preannounced solution which is the essence of this approach.

A final point to be mentioned about the features of this approach is that the arbitrator does not have control over all the decision variables but has control over variables which are related to those not under his control, thus the possibility of influencing the final agreement is weakened. This limit becomes clearer in the case of planning for the economy as a whole where the government acts as an arbitrator, and has to choose an optimal set of instrument variables to influence, indirectly, the target variables determined by private decisions.

4. Unrestricted MOLP's

The attempts continued in this reppect to reach a method which might be considered as an improvement in this field. These attempts are described briefly in this section under the category of unrestricted MOLP's.

4.1 The first method is the aggregation of the available objectives, say K objectives, into a single value function.

This approach depends mainly on a good knowledge of the D.M.'s preferences, where an indifference map needs to be available in order to be able to construct the value function. Besides it is important to know whether to apply the additivity rule or any other rules, provided that the obtained function is linear if we want to apply the LP technique for optimization, yet if the obtained value function is not linear it should be linearized as much as possible.

Suppose that additivity is permissible, another problem will emerge, and that is the need to calculate weights for the various available objectives or to decide on a certain preference order of these objectives.

After constructing the value function, two approaches can be utilized; either the non-probablistic approach, or the prohablistic one.* In the non-probablistic approach the value function is optimized such that $V(S_1', ----, S_K') \geqslant V(S_1'', ---, S_K'')$, where $(S_1', ---, S_K')$ is preferred to $(S_1'', ---, S_K'')$, and V stands for the value function defined on K-triples assuming that there are K objective function under consideration.

---

* B. Roy, Introduction To The Session: Multiple Objective Functions, The Mathematical Programming Symposium 1970, The Hague, Sept. 14-18, SEMA (Metra-International) Paris p. 4.

This condition must be fulfilled if the additivity rule is to hold. Besides one will need to have available a set of independent rates of substitution. Independent here means that these rates are independent of any value other than those that are compared. This requirement is not realistic.

In the Probablistic approach the value function is called a utility function and stronger assumptions are needed for additivity to hold, where an acceptable level L_i is defined for the ith objective function (i=1, ---, K) and this is the minimum level that this objective function should reach. Then a comparison of the results is made according to

$$\text{Min } [\gamma_1(a) - L_1, \text{----}, \gamma_K(a) - L_K]$$

or equally

$$\text{Max } Z$$

$$\text{s.t} \quad \gamma_i(a) - L_i - Z = 0 \qquad \forall\, i$$

where $\gamma_i(a)$ stands for the ith objective function.

This approach involves an arbitrary choice of L_i which may be considered as one of the shortcomings, beside the need to use Probability Theory in constructing the value function or the utility function.

4.2 Since it is difficult to construct the utility function as required in the previous approach especially if the D.M. is not able to give enough information on his preferences, another approach is suggested if the completeness axiom is dropped. In this approach we still can construct a utility function which represents the preference order but in a weaker sense although it still keeps many of the useful properties of the utility function. The construction of the utility function will encounter more complications. Hence it will be preferable to build an outranking relation which will reflect only the part of the D.M. preferences that can be fully accounted for by means of the available data. This outranking relation will be used for comparing any couple of feasible actions (a',a") taking into account the quality of the available data and the corresponding vectors of each criterion.*

The main problem here is how to construct such an outranking relation, call it R,. To reach an acceptable R, two feasible levels (a',a") must be defined for each criterion, then consider three subsets for each of these criteria, say $I^{+}(a',a'')$, $I^{-}(a',a'')$, $I^{=}(a',a'')$ which stand for the set of criteria for which a' is preferred to a", a" is preferred to a', and a' is indifferent to a" respectively.** All these are subsets of the set $I = \{1,2,---,K\}$. Each subset can be considered as a new criterion made of the union of its elements. Then the importance of these subsets must be found and let them be $P^{+}(a',a'')$, $P^{-}(a',a'')$, $P^{=}(a',a'')$ respectively, and they can be used as information to assess an outranking relation which is based on a partial order associated with the K objective functions. For a'Ra" to hold, i.e. for a' to be preferred to a", a' and a" must satisfy concordance condition(s) of the form :†

$$C[P^{+}(a',a''), P^{=}(a',a''), P^{-}(a',a'')] \geq C$$

where C stands for an acceptable level of the degree of concordance among the available criteria, when a' is preferred to a". There can

---

* B. Roy, P. 16.
** Ibid., P. 18.
† Ibid., P. 18.

be one or several such conditions. Then a table is constructed with a' for the rows and a" for the columns. Each cell will contain $P^+/P^- > 1$ under the concordance conditions, and it must be noticed that a high degree of concordance is necessary for a'Ra" to hold but not sufficient. Besides it is assumed that it is not possible to aggregate the positive deviations $[\gamma_i(a') - \gamma_i(a'')]$, $i \in I^+$ and the negative deviations $[\gamma_i(a') - \gamma_i(a'')]$, $i \in I^-$ to see whether they affect each other, thus additional conditions must be imposed on the -ve deviations, provided that these conditions are sufficient for a'Ra" to hold when the necessary condition is satisfied (i.e. the concordance condition). The sufficient condition is called the disconcordance condition: a' can not outrank a" if $\gamma_i(a')$ is very bad compared with $\gamma_i(a'')$ giving a -ve deviation that cannot be offset by any +ve deviations.

Yet stronger and more sophisticated conditions can be established to define R depending on the type of the problem.

After defining R according to the mentioned conditions an associated graph can be drawn. In this graph a'Ra" means that there is an arc between a' and a", and some of the included nodes will form the kernel hence the others can be eliminated. For comparison between the nodes in the kernel a number of approaches can be used depending on the D.M.'s answers about his preferences between vectors of the objectives values, and on a set of upper and lower bounds. If the results are not satisfactory another iteration of comparison is needed for more information.

CHAPTER THREE

THE STEP METHOD

1. The latest attempts in the field of MOLP problems have produced a method to which I am devoting a separate chapter, since it is the main core of this project and because it can be considered as the best method so far produced. This method suggests a procedure to reach at a solution of the MOLP problem as a linear programming one, to get the best possible compromise. It can be classified under the category titled Progressive Definition of preferences.

This method was presented at the 7th Mathematical Programming symposium, which was held in the Netherlands, in the period 14th-18th of September, 1970, where a group of Mathematicians; R. Benayon, J. de Montgolfier, J. Tergny: (Direction Scientifigue) at SEMA (METRA International) Paris, presented a method for solving such a problem, i.e. MOLP. This method is given the name the STEP method or STEM.

STEM was established to give the best possible compromise and to allow for the sequential exploration of the solution by the D.M., in the form of stating answers for specific questions forwarded by the analyst. This exploration by the D.M. alternates with a computational process which tries to improve the compromise obtained at every iteration until the best, from the point of view of the D.M. is reached.

The paper explaining the STEP method proposes a number of other possibilities for solving the MOLP problem, but it states that these methods, such as the solution by Hierarchy or by Parametrics, i.e. by considering all the objective functions, save the most important one, as additional constraints with Parameters as right hand sides (RHS), etc., will not work efficiently when the objectives are very different or have complex inter-relations, due to the restrictions imposed by these methods on the D.M. Hence the STEP method is considered as providing the best solutions for all MOLPs.

2. The STEP method consists of two phases, a Computational phase and a Decision making one, and they alternate until the best compromise is found.

2.1 The Computational Phase:

It starts with the construction of a square matrix which is called the Pay-off matrix, and ends with the submission of a compromise solution in the first iteration, or a better compromise in the succeeding iterations, to the D.M. for his consideration.

Before constructing the pay-off table the MOLP must be formulated by constructing the objective functions required, and the constraints in a linear form, or finding the best linear approximations if they cannot be linearly formulated.

The paper gave the notation for the suitable MOLP in a standard form as follows: *

$$\text{Max } Z_1 = \sum_{i=1}^{n} C_i^1 . X_i$$

$$\text{Max } Z_2 = \sum_{i=1}^{n} C_i^2 . X_i$$

$$\vdots$$

$$\text{Max } Z_K = \sum_{i=1}^{n} C_i^K . X_i$$

---

* Benayon et al, op. cit., p. 2.

s.t $AX \leqslant b$ where $X = (X_1, ---, X_n)$

$X_i \geqslant 0$ $i = 1, ---, n$

where K is the number of available objective functions. If any of the objectives is to be minimized it can be transformed to maximization by reversing the signs of its coefficients.

The Pay-Off matrix is constructed by optimizing the first objective under the given set of constraints and evaluating each of the others at the achieved optimum feasible solution and filling in the first row, where rows represent the extreme points obtained, then repeating the same procedure for the remaining objectives, without the need to solve K LP problems, by introducing successive objectives in the optimum simplex (or modified simplex) tableau of the previous objective (or objectives). Filling in the remaining rows in the same way we get the Pay-Off matrix which is of order K x K.

From this matrix, in which each column contains the different values of the corresponding objective function evaluated at the different optimum solutions attained, the ideal solution X^{Ω} can be found from the diagonal elements, which represent the optimum values of all the objectives. This ideal solution is generally not feasible since no feasible simultaneous optima exist for all the objectives. To make X^{Ω} feasible one has to reach at the compromise solution X^{π} by solving what is called LP(π) which can be stated as follows:

$$\text{Min } Z = \lambda - \varepsilon \sum_{j=1}^{K} C^j(X) \qquad \varepsilon \text{ is very small but } \neq 0$$

$$\text{s.t} \qquad \lambda \geqslant \left[\left|M^j - Z_j(X)\right|\right] . \pi^j \quad j = 1, ---, K$$

Beside the original set of constraints

$X \varepsilon D; \quad \lambda \varepsilon R$

where in this LP(π) we try to minimize the distance between the best compromise choosen by the D.M. and the ideal solution given by the analyst from the pay-off table, or in other words to obtain an acceptable compromise which is feasible and the nearest to the ideal solution which, as mentioned before, is not feasible in general.

The size of the LP(π) will increase by adding more constraints after every conversational phase to improve the compromise solution X^{π} until the best is reached at.

The π^js here stand for the weights given to the various objective functions and they determine how far we are from the optimum of each objective. These weights can be calculated regardless of whether there are a priori weights or a given hierarchy or even if nothing is known at all, where in the latter case the pay-off table can be used for calculating these weights (π^js). In all the cases these weights are obtained by normalizing the values assumed by the objectives.

$$\pi^j = \gamma^j / \sum_{j=1}^{K} \gamma^j \qquad j = 1, ---, K.$$

where γ^j can be calculated in two different ways depending onthe available data and prior information.

i) $\gamma^j = \sigma^j / \sqrt{\sum_{i=1}^{n} (C_i^j)^2}$ $\qquad$ n = the no. of activities involved

σ^j = a priori weight assigned to the jth objective or the sum of the elements of the jth row in a K x K matrix constructed from a knowledge of a certain hierarchy of the objectives.

C_i^j = the ith coefficient of the jth objective.

or ii) $\gamma^j = (|M^j - m^j|)/(\max(|M^j|, |m^j|). \sqrt{\sum_{i=1}^{n} (C_i^j)^2})$

Here the weights are calculated using the Pay-Off table.

M^j = the best value of the jth objective and it is the ith diagonal

m^j = the worst value of the jth objective

both M^j & m^j are obtained from the jth column of the table.

2.2 The Conversational Phase

After solving the LP(π) and reaching to X^π, the D.M. compares this compromise solution with X^Ω (i.e. the ideal solution) to decide on whether X^π is satisfactory by examining for each objective both values and obviously he will not accept any X^π giving a value for any objective which is worse than m^j, and if such a situation arises he must accept the need to relax the values of the objectives he considers satisfactory, in order to improve the value of the objective(s) he is not satisfied with, hence to achieve a better compromise X^π in the next iteration. Thus he has to decide on the amount of relaxation he is prepared to allow say ΔC_M^{j*}, where j* is the objective which value can be relaxed. Then new weights have to be calculated in accordance but with

$$\gamma^{j*} = \pi^{j*} = 0$$

To compute the improved compromise another set of constraints must be added. They take the form:

$$Z^{j*}(X) > Z^{j*}(\pi) - \Delta Z_M^{j*}$$

$$Z^j(X) > Z^j(X^\pi) \qquad j \neq J^*$$

These constraints will allow for the improvement of the values of the unacceptable objectives in the following iteration(s).

All this information is obtained during a conversation between the analyst and the D.M. in the form of certain questions and the answers to them. Then the previous steps are repeated including the conversation between the D.M. and the analyst.

The satisfactory compromise can be reached in less than K iterations depending on how many objectives are relaxed at once, bearing in mind that any objective j* cannot be relaxed more than once.

The amount of relaxation ΔC_M^{j*} can be determined easily if sensitivity analysis is carried out using a range of values including the compromise solution X^π, i.e. specifying upper and lower limits of the changes in some of the objectives when some of them are relaxed by one unit. The paper explaining the STEP method gives a detailed explanation of how to determine the objective to be relaxed and the amount of relaxation, where the set of the objective functions is to be added to the LP(π) .

The algorithm explained above was applied by KODAK-PATHE FRANCE with the collaboration of members of the SEMA (Metra International, Paris) Company. The model to which it was applied included only four objectives, two of them were to be maximized and the other two to be minimized and they were conflicting with each other. The best compromise was reached after exactly three iterations which is less than the number of objective functions included in the model.

It is also being applied to large scale models of the macro-economic type of models to deal with planning for trade and development in the developing countries. These models incorporated more than one objective function and depending on input output analysis.

3. Although this can be considered as the best method so far, one of its snags is in calculating the weights and deciding on them. Another major snag emerges from the type of model to which it is applied and this will be discussed later on when discussing the practical part by applying STEP to a hypothetical model.

On the other hand it has many advantages; first of all the D.M. will be able, during the successive iterations, to learn more about the relative importance of the various objectives, and second, that it allows a wide range of choice of solutions, besides and in spite of the weights problem, these weights, as mentioned, can be calculated even if we know nothing at all about the relative importance of the various objectives, and finally although STEM yields a solution which is not a mathematical optimum, the final compromise might coincide with the optimum of one of the objectives of it dominates the others.

A final point is that, as mentioned before, STEM can be regarded as the best possible technique that can be used to obtain the nearest solution to the Pareto Optimal solution, to be used as the imposed solution in the Restricted Bargaining Approach.

As I mentioned before, I have started this project with the hope that I will be able to give some real results by solving a real large scale model built for the Nigerian economy, but after a while I found that it will be a semi-impossible task given the available data and time. Hence I was advised to work with a small and hypothetical model to get myself acquainted with the STEP method algorithm. Being a very small model with four objectives, three variables and two constraints, the first calculation of the ideal solution and the pay-off table was done by hand since the computer programme for this part was still tested. Then the successive compromises were obtained using a Modified Simplex LP computer procedure.

This simple model is:

$$\max Z_1 = -2X_1 + 3X_2 - X_3$$

$$\max Z_2 = -X_1 - X_2 - 3X_3$$

$$\max Z_3 = 2X_1 + 2X_2 + X_3$$

$$\max Z_4 = 2X_1 + X_2 + 3X_3$$

$$\text{s.t.} \quad X_1 + X_2 + 2X_3 \leqslant 6$$

$$3X_1 - X_2 + X_3 \geqslant 4$$

Referring to the D.M. several compromises were obtained but after two iterations the D.M. decided that the solution is not satisfactory even after relaxing all but the second objective. The original MOLP was resolved to discover that a second ideal solution X^{Ω} exists for the same MOLP when it was solved by starting with a different pivot. Re-examining the problem it appeared that this multiplicity of solutions was due to the fact that one of the objectives (second objective) was almost parallel to one of the constraint viz; the first constraint, hence there are two extreme points satisfying this objective function plus all other points that fall in between on the corresponding border line.

This situation lead to the need to give a second thought to the actual meaning of some of the terms defined in the original paper on the STEP method.

First of all, I found that the weights πj's calculated for the above mentioned example do not satisfy the condition mentioned in the STEM algorithm, i.e. the wider the difference between the ideal solution and other values of objective j, all obtained from the pay-off table, the higher the value of πj and vice versa, where in the given example the weights are almost equal for all the objectives, so they did not serve the purpose of directing attention to the most important objective with the larger difference.

From the above discussion a second question arises: what was meant by the worst value of the jth objective when calculating the weights according to

$$\gamma_j = \frac{|Z_j^{opt} - Z_j^{worst}|}{\max\,[|Z_j^{opt}|, |Z_j^{worst}|]} \cdot \frac{1}{\sqrt{\sum C_i^{j\,2}}}$$

is Z_j^{worst} is the worst value as found from the pay-off table or the worst possible value that objective can take in the specified region regardless of the pay-off table?

The ambiguity of this term might be another reason for the failure of the weights to comply with the purpose of their evaluation.

Besides there might be another reason represented in the existence of the normalization factor when calculating γ_j's

$$\frac{1}{\sqrt{\sum C_i^{j\,2}}}$$

All these reasons and mainly the existence of multiple solutions might have lead to these unacceptable results.

This situation was discussed with the D.M. in an attempt to reach at a method to cure this problem whenever it arises, i.e. the problem of the multiplicity of solutions due to the parallelism or simi-parallelism of one or more of the objective functions to one or more of the constraints.

A modification on the STEM algorithm, whemever such a situation arises is suggested as follows:

The above discussion costs dought on the necessity to construct a pay-off table as a first step in the algorithm whenever the multiplicity of solution occurs. It is suggested that first of all one has to solve for all the available objectives as in the STEP method. Then examine the dual variables corresponding to each objective. If some of these variables are at zero level, this will indicate that the corresponding objective will give rise to multiple solutions, hence a modification in the algorithm is needed. This testing is necessary if such an objective cannot be spotted with the bare eye.

If one (or more) of the objectives is found, all what we need at first is to record the ideal solution i.e. the optimum of each objective. Then the worst value of all objectives can be found according to the suggested procedure.

This procedure can be explained in a few points:

(i) After finding the ideal solution X^{Ω} and finding the objective function(s) causing the multiplicity of solutions, a new constraint can be added to the original set of constraints. The new constraint can take the form

$$Z_j = Z_j^{opt}$$

where Z_j is the jth objective causing the trouble, and Z_j^{opt} is its optimum value. In the above example it is obvious that the second objective is almost parallel to the first constraint

$$\min Z_2 = X_1 + X_2 + 3X_3$$

$$\text{s.t. } X_1 + X_2 + 2X_3 \leq 6$$

I say it is almost parallel inspite of the difference in the coefficient of X_3 in these two relations, since from the solution obtained it appears that X_3 does not appear in any of the solutions.

Hence the new constraint to be added is

$$Z_2 = Z_2^{opt}$$

or mathematically as obtained from the solution.

$$X_1 + X_2 + 3X_3 = 1.3333$$

(ii) Minimize each of the other objectives subject to the new set of constraints, and the values of these objectives will be taken as the worst value for each of them, and they can be used to calculate the weight γ_j's and π_j's as before. As for the jth objective causing the multiplicity the worst value can be obtained by minimizing it subject to the original set of constraints.

(iii) If more than one objective has zero dual variables or seen to be parallel to some of the constraints the previous two steps should be repeated for each such objectives, i.e. they should be repeated as many times as there are such objective functions, and each time we add only one new constraint of the form $Z_i = Z_i^{opt}$ to the original set of constraints and minimize all the objectives but the ith subject to this set. In this case an incomplete pay-off table will be constructed from which the worst possible value of each function can be found as before.

After that one can continue according to STEM algorithm to find a satisfactory solution.

I tried to apply the aforementioned procedure to the given example, so I found the worst values fow each objective and they differ from those calculated according to the STEP Method in the pay-off table where some were better than the worst taken from the pay-off table, as happened with the first objective function where the difference was 8 units and some were even worse as for the remaining three objectives. Then using the expression given for γ_j & π_j the weights were calculated, but again they didn't satisfy the condition that they should differ up and down in accordance with the width of the gap between the optimum and the worst value of each objective.

As mentioned before, it might be that the ambiguity of the term "Z_j^{worst}" is not the main cause behind all these difficulties, apart from the possibility of the existence of multiple solution as declared. There is a third possibility and that is the existence of the normalization factor in the expression used in calculating the weights. According to this possibility a new set of weights γ_j's & π_j's were calculated and they seem to be more satisfactory as far as the purpose of their evaluation is concerned.

Calculations of the Pay-Off Table and γ_j's and π_j's

(1)

	z_1	z_2	z_3	z_4
1	$\frac{8}{3}$	$\frac{4}{3}$	$\frac{8}{3}$	$\frac{8}{3}$
2	$\frac{8}{3}$	$\frac{4}{3}$	$\frac{8}{3}$	$\frac{8}{3}$
3	$\frac{31}{2}$	6	12	$\frac{17}{2}$
4	12	6	12	12

The weights from (1)

$\gamma_1 = .221$ $\gamma_2 = .235$ $\gamma_3 = .259$ $\gamma_4 = .221$

$\pi_1 = .236$ $\pi_2 = .251$ $\pi_3 = .277$ $\pi_4 = .236$

Calculation of the worst values and the weights using the normalization factor

$Z_1^{worst} = 4$ $Z_2^{worst} = \frac{44}{5}$ $Z_3^{worst} = \frac{4}{9}$ $Z_4^{worst} = \frac{4}{3}$

$\gamma_1 = .08607$ $\gamma_2 = .25585$ $\gamma_3 = .32099$ $\gamma_4 = .22951$

$\pi_1 = .09645$ $\pi_2 = .28669$ $\pi_3 = .35968$ $\pi_4 = .25619$

Calculation of weights dropping the normalization factor

$\gamma_1 = .3333$ $\gamma_2 = .84849$ $\gamma_3 = .96296$ $\gamma_4 = .88889$

$\pi_1 = .1098$ $\pi_2 = .2797$ $\pi_3 = .3174$ $\pi_4 = .2931$

CHAPTER FOUR

THE COMPUTER PROGRAMME MOLP

1. As mentioned before the main purpose of this project was to write a computer programme which can be used to solve any Multiple Objective Linear-Programming Model (MOLP) on the computer especially if the MOLP in hand is a large scale one.

To write this programme I used as a base an ALGOL programme which is used to solve any linear programming problem but with a single objective function. The algorithm on which this basic programme LPMODSIMPLEX* is based is the Modified Simplex algorithm. LPMODSIMPLEX will solve any problem and print the results if any.

This programme is able to test whether the problem has a solution at all and whether the problem is bounded or not.

To make this programme deals with linear programming problems with more than one objective function according to STEM, as explained in the previous chapter, many modifications on the procedure involved had to be done, then many blocks were added to the main programme.

I will explain these modifications and additions according to the stages I followed. A listing of two versions of the programme MOLP will be given at the end of this chapter.

STAGE 1

First of all I tried to make the original procedure solves for several objectives one after another without the need to solve as many problems as there are objective functions. This was done by

---

* LPMODSIMPLEX, BIT Contribution No. 10, 4 (1964), H. Josefsen.

first changing the size of the coefficient matrices and the value column. For example the coefficient array C was in LPMODSIMPLEX of size (0:M, 1:N) and now it becomes of size (1:M + K + 1, 1:N) and this alteration was due to the need to use FORTRAN Routines in which the indexing of the elements cannot start at 0. The first K rows of C will hold the K objective function available instead of just a single objective that used to be located at the row zero, the row K + 1 is filled with zeros, and it was allowed for to be used where the compromise solution is to be attained, where in this row the objective function of the LP(π) will be stored. Then the remaining M rows contain the set of constraints without requiring any order due to another alteration explained below. The N^{th} column is filled with zeros at the beginning but it will be used when solving the LP(π), where it will locate the values of λ in LP(π).

A second modification was in the way to distinguish between the types of constraints equalities and inequalities. In LPMODSIMPLEX there was a need to arrange the constraints in certain order where inequalities $\leq$ and $\geq$ come first then the equalities. To do this two parameters were used one indicate the total number of artificial variables that must be used and the other is the number of negative slacks or in other words the number of inequalities with a $\geq$ sign.

But this requirement of a certain order was not suitable to what I have in mind of modifications in later stages since there will be an increase in the number of constraints when solving the successive compromises and it will be difficult to get the added constraints to agree with the order required. Thus instead I introduced a new array called CODE of size M, the elements of which will take the values -1, 1, or 0 according to the type of constraint, $\geq$, $\leq$ or = respectively. At the beginning of the first run the elements of CODE will be read in as data.

A third modification was to read in a new parameter to state the number of objective functions we are dealing with (K).

The procedure is now called to solve the MOLP using the provided data. The improvement in this procedure as mentioned above was by making it solves for K objectives one after the other by solving for the first objective function subject to the set of constraints, then after reaching the optimal tableau and instead of jumping out of the procedure, the next objective function is tested at this optimum, if the optimal tableau remains feasible this means that this optimum satisfies this objective, and if it does not remain feasible a few more iterations will be carried out until feasibility is retained. This is repeated for each objective function.

Each time an optimum is reached a row of a K X K array NP is filled, hence NP will contain the Pay-Off table (see Section 2, Chapter 3). When all elements of NP are there a jump from the procedure to print the already available results is required.

These results are used to calculate the weights γ_j's and π_j's and these are stored in the column vectors GAMA and PY each of size K.

The formulae used are those supplied by STEM

$$\gamma_j = \frac{|z_j^{opt} - z_j^{worst}|}{\max\left[|z_j^{opt}|, |z_j^{worst}|\right]} \cdot \frac{1}{\sqrt{\sum c_i^{j\,2}}}$$

$$\pi_j = \frac{\gamma_j}{\sum \gamma_j}$$

Both these stages i.e. calculating and printing NP, GAMA and PY were done successfully.

STAGE 2

At this stage I attempted to construct the first compromise solution in the first run. It was a difficult stage due to the fact that there will be an increase in the size of some of the arrays used and these were the coefficient arrays C, the value column S, the inverse matrix CP and Code. This increase is simply due to the increase in the number of constraints subject to which the LP(π) will be solved or in other words due to a higher value of the parameter M (No. of constraints), where in the first compromise the number of constraints will increase by K (No. of objectives).

This difficulty would not have appeared if I was using a computer language other than ALGOL 60 where this language does not have any facilities to allow for this increase in the size. On the other hand any other language even ALGOL 68 has such a facility.

To allow for this increase in the size a use is made of the magnetic disc, where a number of FORTRAN Routines is used to allow for writing these arrays at their old size to disc, and after increasing M by K the arrays are redeclared and read from disc at the new sizes. This means that they will be stored in the upper parts of the new space allowed, leaving the lower parts for the new information to be added. The empty spaces are filled with values that are calculated to agree with STEM, where the K + 1 row of C is filled according to

$$\min Z = \lambda - \sum_{i=1}^{K} C_i^j \qquad j = 1, ---, N-1$$

Then the coefficients of the new K constraints are stored in the lower space after being calculated from the formula

$$\lambda \geq \pi_j \left[|Z_j^{opt} - Z_j|\right]$$

and the signs and hence the corresponding elements of code are adjusted according to whether Z_j is to be minimized or maximized. At the same time the new elements of the value column are added and the phase one infeasibility element is adjusted correspondingly. Finally the basis column will increase in size too.

Another call of the procedure is made to solve the LP(π). Here a Boolean variable 2 is used to tell whether we are still constructing NP or solve LP(π).

A disadvantage of this attempt is that a space will be allowed for a square matrix NP when solving for LP(π), yet only the first row will be used destroying the old values stored in it. This mow will contain the values of the K objective functions evaluated at the solution of LP(π) i.e. the first compromise.

STAGE 3

A third stage was introduced to allow for using the same programme for the first as well as for further runs to obtain new compromises after consulting the D.M. and acquiring new information about the degree of his satisfaction with the previous compromise shown to him. This information consists of the objective functions which values are or are not satisfactory, the percentage of relaxation of the values of these objectives which are satisfactory. This stage needed a complete change of the indexing system used in the previous one to allow for several runs where the sizes of the mentioned arrays will change in an irregular manner where they will increase in the first run and might increase even more, or decrease in the second run, and will definitely decrease in the succeeding runs, according to the number of objectives that are relaxed. Here again the same FORTRAN routines will be used even more since more arrays and parameters need to be stored on disc for the second and later runs instead of re-reading them as data. Among the arrays that need be stored are the weights GAMA and a new array COMP in which the values of the objectives at the previous compromise are stored, and will be used for recalculating the weights and the right hand sides of some of the new constraints. An integer variable, H, is used and its value will be read more than once at one run, and each time it will take a different value to indicate which objective is satisfactory and can be relaxed hence the corresponding weights will take the value zero. This is done in the second and later runs. Accordingly a change in the value of M will take place keeping the previous value in another parameter T to facilitate the reading for and writing to disc. At each of the second and succeeding runs a call of the procedure will be made, the compromise solution, the primal solution to the LP(π) and the simplex multipliers will be printed in order.

STAGE 4

The final stage was incorporated to allow for the testing of the existence of a multiple solution to the original MOLP. This was done by introducing some modifications on the procedure as well as on the main programme. In this stage two new arrays were declared, one is local to the procedure, MULT (1:M) and it is a real array, and the other was common to both the procedure and the main programme and it is a Boolean array OBJ (1:K), where the elements of the latter are initiated as TRUE. In MULT the simplex multipliers corresponding to each objective function are stored after reaching the optimum and if any of the elements is at zero level the element of OBJ corresponding to the current objective function will take the value FALSE indicating that this objective function is causing the multiplicity of solutions. This test is repeated for each objective keeping the corresponding element TRUE if the objective function does not cause any multiplicity. All this testing is done within the first call of the procedure i.e. together with the construction of the pay-off table. Coming out of the procedure we will go through the elements of OBJ and whenever an element is FALSE another call of the procedure is needed after adding a single constraint using the available and the calculated data according to the procedure suggested in the last section of the previous chapter. The parameter M will be changed temporarily before intering the procedure to solve for K-1 objective functions subject to the new set of constraints. These K-1 objective functions are the old ones multiplied by -1. This is repeated for each objective causing the multiplicity hence a new pay-off table will overwrite the old one wherever necessary.

Of course this stage will precede the previous one, i.e. it will take place within the first run of the programme and before calculating the first compromise.

In this stage I didn't allow for finding the worst value of each of the FALSE objectives by reversing the sign hence the worst values will be those already given in the old pay-off table.

This stage will come before calculating the weights since the information supplied by this stage will be needed to calculate these weights according to the same formulae shown above.

2. The Data Structure

2.1. The First Run of MOLP

The data to be read are as follows:

U   a parameter to indicate at which run we are, it will take the value zero in the first run.

N number of variables, and it will be one variable more than the number of variables in the original MOLP to allow for λ in the compromise solution.

M number of constraints

F 1 if we are to start at phase one when there are negative slacks and/or artificial variables and 2 otherwise.

K number of objective functions

CODE an integer array with M elements with the values 0, -1, or 1 to indicate =, $\geqslant$, or $\leqslant$ respectively without any order.

C The coefficient matrix which is read by rows but indicating the number of the row first then the number of the column then the value of the element.

S The value column of size M+2, the elements 3-M+2 are read in the manner as for C but whenever the column number is zero.

2.2 The Second and Further Runs

The data needed are:

U as before but will take the values 1,2,----

H an integer variable used to indicate the objective which value is satisfactory according to the D.M. It might take more than one value in the same run depending on how many objectives are to be relaxed in one run, each time H is read GAMA(H) = PY(H) = 0 (the weight will be at zero level). When a zero value is read for H it means that no more objectives are to be relaxed.

The percentage of relaxation:

This is not given any name in the programme and is read whenever applicable to construct new LP(π). (See previous chapter.)

3. A Listing of MOLP

A listing of two versions of MOLP are given below.

In the first version stages 1-2 are incorporated while the second version will show all the four stages.

The first version is producing results while the second is not producing any since I did not have sufficient time to carry on with it till the end.

```
'BEGIN' 'INTEGER' I,J,L,M,N,F,K,V,NOS,UNB,
LAY1,LAY2,LAY3£
'REAL' MM,E,TIM,SUM1,SUM2,ZJW£
'BOOLEAN'Z£
'PROCEDURE' MOLP(M,N,F,Z,K,MM,E,C,S,CP,CODE,NP,BD,
NOSOLUTION, UNBOUNDED)£
'VALUE' M,N,F,K,MM£
'INTEGER'M,N,F,K£
'REAL' MM,E£
 'BOOLEAN'Z£
'ARRAY' C,S,CP,NP£
'INTEGERARRAY'BD,CODE£
'LABEL'NOSOLUTION,UNBOUNDED£
'BEGIN' 'INTEGER' Q,P,I,J,L,V,IQN,
U,
FORM,ITR£
'REAL'R,X£
 'ARRAY' TT(F'.'M+2)£
'BOOLEANARRAY' UU(1'.'N)£
'FOR'J=1'STEP'1'UNTIL'N'DO'
UU(J)='TRUE'£
FORM=F£
ITR=1£
V=1£
NEXTITR'.'R=0£
OUTPUT(30,S(FORM,1))£
Q=0£
'FOR'J=1'STEP'1'UNTIL'N'DO IF'UU(J)'THEN'
'BEGIN' X='IF'FORM'EQ'1'THEN'0
'ELSE'('IF'Z'THEN'C(K+1,J)'ELSE'C(V,J))£
'FOR'I=1'STEP'1'UNTIL'M'DO'
X=X+CP(FORM,I)*C(I+K+1,J)£
 'IF'X-R+E'LT'0'THEN BEGIN'R=X£Q=J'END'£
 'END'£
 'FOR'J=1'STEP'1'UNTIL'M'DO'
 'BEGIN'X='IF' CODE(J,1)'EQ'-1'THEN'
```

```
-CP(FORM,J)'ELSE' CP(FORM,J)£
'IF'X-R+E'LT'O'THEN BEGIN'R=X£Q=N+J'END'£
'END'£
'IF'Q'EQ'O'THEN BEGIN'
'IF'FORM'EQ'2'THEN''GOTO' EXIT£
'IF' FORM'EQ'1'AND' ABS(S(1,1))
 'LE' E*((M+ITR)**3)'THEN'
'BEGIN'FORM=2£ 'GOTO'NEXTITR'END'£
'IF' FORM'EQ'1'AND'S(1,1)'LT'O'THEN''GOTO'NOSOLUTION£
'END'£
R=MM£
P=O£
'IF'Q'LE'N'THEN'
'BEGIN'
'FOR'I=F'STEP'1'UNTIL'M+2'DO'
'BEGIN' X=('IF' I'EQ'2'THEN'
('IF'Z'THEN'C(K+1,Q)'ELSE'C(V,Q))
'ELSE'O)£
'FOR'J=1'STEP'1'UNTIL'M'DO'
X=X+CP(I,J)*C(J+K+1,Q)£
TT(I)=X£
'IF'I'GE'3'THEN' 'BEGIN'
'IF'X'GT'E'AND'S(I,1)'LT'R*X'THEN'
'BEGIN' R=S(I,1)/X£  P=I£  'END'£
'END' 'END'£
'END ELSE'
'BEGIN'
'IF'Q'GT'N'THEN' IQN=Q-N£
'FOR'I=F'STEP'1'UNTIL'M+2'DO'
'BEGIN'X='IF'CODE(IQN,1)'EQ'-1'THEN'
-CP(I,IQN)'ELSE'CP(I,IQN)£
TT(I)=X£
'IF' I'GE'3'AND'X'GT'E'AND'S(I,1)'LT'R*X'THEN'
'BEGIN'R=S(I,1)/X£P=I'END'
'END END'£
'IF'P'EQ'O'THEN GOTO'UNBOUNDED£
```

```
TT(P)=1/TT(P)£
'FOR'J=1'STEP'1'UNTIL'M'DO'CP(P,J)=CP(P,J)*TT(P)£
S(P,1)=S(P,1)*TT(P)£
'FOR'I=F'STEP'1'UNTIL'M+2'DO'
'IF'P'NE'I'THEN'
'BEGIN'
S(I,1)=S(I,1)-S(P,1)*TT(I)£
'FOR'J=1'STEP'1'UNTIL'M'DO'CP(I,J)=CP(I,J)-CP(P,J)*TT(I)£
'END'£
'IF' BD(P-2)'LE'N'THEN'UU(BD(P-2))='TRUE'£
'IF'Q'LE'N'THEN'UU(Q)='FALSE'£
BD(P-2)=Q£
ITR=ITR+1£
'GOTO'NEXTITR£
EXIT'.'
'FOR' I=1'STEP'1'UNTIL'M'DO'
'BEGIN' WRITE(30,LAYOUT('('-NDD')'),BD(I))£
WRITE(30,LAYOUT('('-NDDDDD.DDC')'),S(I+2))£ 'END'£
WRITET(30,'('SOLVED'('C')'')')£
'FOR'J=1'STEP'1'UNTIL'K'DO'
'FOR'L=1'STEP'1'UNTIL'N'DO'
'FOR'I=1'STEP'1'UNTIL'M'DO'
'IF'BD(I)'LE'N
'AND'BD(I)'EQ'L'THEN'
NP(V,J)=NP(V,J)+C(J,L)*S(I+2,1)£
'FOR'J=1'STEP'1'UNTIL'K'DO'
WRITE(30,LAYOUT('('SS-D.DDDDDDDD'E'+ND')'),NP(V,J))£
NEWLIN(30,1)£
R=0£
'IF'V'EQ'K'OR'Z'THEN''GOTO' FIN 'ELSE' 'BEGIN'
'FOR'L=1'STEP'1'UNTIL'N'DO'
'IF' 'NOT'UU(L)'AND'C(V+1,L)'NE'0'THEN'
'BEGIN' 'IF'R'EQ'0'THEN'
'BEGIN' R=R+1£
S(FORM,1)=0£
'FOR'J=1'STEP'1'UNTIL'M'DO'
```

```
CP(FORM,J)=0.0£
'END'£
'FOR'U=1'STEP'1'UNTIL'M'DO'
'FOR'I=3'STEP'1'UNTIL'M+2'DO'
'IF'BD(I-2)'LE'N
'AND'BD(I-2)'EQ'L'THEN'
'BEGIN' CP(FORM,U)=CP(FORM,U)+
CP(I,U)*(-C(V+1,L))£
S(FORM,1)=S(FORM,1)+S(I,1)*(-C(V+1,L))£
'END'£ 'END'£
V=V+1£ 'GOTO' NEXTITR£ 'END'£
FIN'.'
'END'£
Z='FALSE'£
MM=2**39£E='E'-6£
LAY1=LAYOUT('('-NDDD')')£
LAY2=LAYOUT('('S-D.DDDDDDDD'E'+ND')')£
LAY3=LAYOUT('('S-D.DDDDDDDD'E'+NDC')')£
D'.'N=READ(40)£'IF'N'LE'0'THEN GOTO'OUT£
M=READ(40)£  F=READ(40)£  K=READ(40)£
WRITET(30,'(''('C')'VARIABLES'*'CONSTRAINTS'*'OBJECTIVES'('C')'')')£
OUTPUT(30,N)£   OUTPUT(30,M)£   OUTPUT(30,K)£
NEWLIN(30,1)£
'BEGINARRAY'
GAMA(1'.'K), PY(1'.'K),
NP(1'.'K,1'.'K)£
Y'.' 'BEGINARRAY' C(1'.'M+K+1,1'.'N),
S,S1(1'.'M+2,1'.'1),
CP(1'.'M+2,1'.'M)£
'INTEGERARRAY' BD(1'.'M),CODE(1'.'M,1'.'1)£
'IF' Z'THEN' 'BEGIN'
RD(0,C,M+1,N,M+K+1)£
RD2(2,CODE,M-K,1,M)£
'FOR'J=1'STEP'1'UNTIL'N'DO'
'FOR' I=1'STEP'1'UNTIL'K'DO'
C(K+1,J)='IF'J'EQ'N'THEN'1'ELSE'C(K+1,J)+C(I,J)£
```

```
'FOR'J=1'STEP'1'UNTIL'N-1'DO'
C(K+1,J)=-E*C(K+1,J)£
'FOR'I=M+2'STEP'1'UNTIL'M+K+1'DO'
'FOR'J=1'STEP'1'UNTIL'N-1'DO'
C(I,J)=
C(I-(M+1),J)*PY(I-(M+1))£
F=1£
J=0£
'FOR'I=M-K+1'STEP'1'UNTIL'M'DO'
'BEGIN'
J=J+1£
'IF'NP(J,J)'GT'0'THEN'
'BEGIN' CODE(I,1)=1£
C(I+K+1,N)=-1£ 'END'
'ELSEBEGIN' 'FOR'L=1'STEP'1'UNTIL'N-1'DO'
C(I+K+1,L)=-C(I+K+1,L)£
CODE(I,1)=-1£
C(I+K+1,N)=1£ 'END'£
'END'£
RD(1,S1,M-K+2,1,M+2)£
'FOR'I=M-1'STEP'1'UNTIL'M+2'DO'
S1(I,1)=PY(I-K)*ABS(NP(I-K,I-K))£
'FOR'I=F'STEP'1'UNTIL'M+2'DO'
S(I,1)=S1(I,1)£
'GOTO' B£
'END'
'ELSE' 'BEGIN'
'FOR' I=1'STEP'1'UNTIL'M+K+1'DO'
'FOR' J=1'STEP'1'UNTIL'N'DO'
'BEGIN'C(I,J)=0£
'IF'I'LE'K'THEN'
'FOR'L=1'STEP'1'UNTIL'K'DO'
NP(I,L)=0.0£
'END'£
'FOR'I=F'STEP'1'UNTIL'M+2'DO'
S(I,1)=S1(I,1)=0.0£
```

```
'FOR'I=1'STEP'1'UNTIL'M'DO'
CODE(I,1)=READ(40)£
A'.' I=READ(40)£
'IF'I'EQ'-1'THENGOTO'B£
'IF'I'LT'-1'OR'I'GT'M+K+1'THEN' 'GOTO'ERROR
£
J=READ(40)£'IF'J'LT'0'OR'J'GT'N'THEN GOTO'ERROR£
'IF'J'EQ'0'THEN'
S(I,1)=S1(I,1)=READ(40)
'ELSE''IF'J'LE'N-1'THEN'
C(I,J)= READ(40)£
'GOTO'A£
'END'£
B'.' 'FOR' I=1'STEP'1'UNTIL'M'DO'
BD(I)=N+I£
'FOR' I=F'STEP'1'UNTIL' M+2'DO'
'FOR'J=1'STEP'1'UNTIL'M'DO'
CP(I,J)='IF'I-2'EQ'J'THEN'1'ELSE'0£
'FOR' I=1'STEP'1'UNTIL'M'DO'
'IF' CODE(I,1)'EQ'-1'THEN'
'BEGIN' CP(1,I)=-1£
'IF''NOT'Z'OR'Z'AND'I'GT'M-K'THEN'
S1(F,1)=S1(F,1)-S1(I+2,1)£
'END'£
S(F,1)=S1(F,1)£
TIM=TIME(E)£
 'FOR'I=1'STEP'1'UNTIL'M+2'DO'
 'FOR'J=1'STEP'1'UNTIL'M'DO'
OUTPUT(30,CP(I,J))£
'FOR'I=1'STEP'1'UNTIL'M+K+1'DO'
'FOR'J=1'STEP'1'UNTIL'N'DO'
OUTPUT(30,C(I,J))£
'FOR'I=1'STEP'1'UNTIL'M+2'DO'
OUTPUT(30,S(I,1))£
'FOR'I=1'STEP'1'UNTIL'M'DO'
OUTPUT(30,CODE(I,1))£
```

```
'FOR'I=1'STEP'1'UNTIL'M'DO'
OUTPUT(30,BD(I))£
MOLP(M,N,F,Z,('IF'Z'THEN'1'ELSE'K),MM,E,C,S,CP,
CODE,NP,BD,NOSOLUTION,UNBOUNDED)£
'IF NOT' Z 'THEN'
'BEGIN'
SUM2=0.0£
'FOR'J=1'STEP'1'UNTIL'K'DO'
'BEGIN'
SUM1=0.0£
'FOR'L=1'STEP'1'UNTIL'N'DO'
SUM1=SUM1+C(J,L)**2£
ZJW=NP(J,J)£
'FOR'I=1'STEP'1'UNTIL'K'DO'
'IF'NP(I,J)'GT'ZJW'THEN' ZJW=NP(I,J)£
GAMA(J)= (ABS(NP(J,J)-ZJW))/
(
('IF'ABS(NP(J,J))'GT'ABS(ZJW)'THEN'
ABS(NP(J,J))'ELSE'ABS(ZJW))*
SQRT(SUM1))£
SUM2=SUM2+GAMA(J)£
'END'£
'FOR'L=1'STEP'1'UNTIL'K'DO'
PY(L)=GAMA(L)/SUM2£
'FOR'L=1'STEP'1'UNTIL'K'DO'
OUTPUT(30,GAMA(L))£
'FOR'J=1'STEP'1'UNTIL'K'DO'
OUTPUT(30,PY(J))£
Z= 'TRUE'£
WD(0,C,M+K+1,N)£
WD(1,S1,M+2,1)£
WD2(2,CODE,M,1)£
M=M+K£
'GOTO' Y£
'END'
'ELSE'
```

```
'BEGIN'
'FOR' I=1'STEP'1'UNTIL'K'DO'
'FOR'J=1'STEP'1'UNTIL'N-1'DO'
'FOR'L=1'STEP'1'UNTIL'M'DO'
'IF' BD(L)'LE'N-1'AND'BD(L)'EQ'J'THEN'
WRITE(30,LAYOUT('('SS-D.DDDDDDDDDD'E'+ND')'),C(I,J)*S(L+2,1))£
WD(0,C,M+K+1,N)£
WD(1,S1,M+2,1)£
WD2(2,CODE,M,1)£
'END'£
TIM=TIME(E)-TIM£
'GOTO'CONT£
NOSOLUTION'.'WRITET(30,'('NO'*'SOLUTION')')£
'GOTO'CONT£
UNBOUNDED'.'WRITET(30,'('UNBOUNDED')')£
CONT'.'
WRITET(30,'(''('C')'PRIMAL'*'SOLUTION'('C')'')')£
'FOR'I=F'STEP'1'UNTIL'M+2'DO'
'BEGIN' 'IF' I'LT' 3'THEN'
SPACE(30,5)'ELSE' WRITE(30,LAY1,BD(I-2))£
WRITE(30,LAY3,S(I,1))£
'END'£
WRITET(30,'(''('C')'SIMPLEX'*'MULTIPLIERS'('C')'')')£
'FOR'I=0'STEP'1'UNTIL'(M-1)'/'10'DO'
'BEGIN FOR'J=1,J+1'WHILE'J'LE'10'AND'J+I*10'LE'M'DO'
WRITE(30,LAY2,'IF'I*10+J'LE'M'THEN'
CP(2,I*10+J)'ELSE'-CP(2,I*10+J))£
NEWLIN(30,1)£
'END'£
WRITET(30,'(''('C')'TIME'*'TAKEN'*'')')£
WRITE(30,LAYOUT('('NDD.DDD')'),TIM)£
WRITET(30,'(''*'SECS'('CC')'')')£
'GOTO'D£
ERROR'.'WRITET(30,'(''('C')'DATA'*'ERROR'('C')'')')£
'END'£
'END'£OUT'.''END'
```

```
      SUBROUTINE WD(IBL,A,M,N)
      DIMENSION A(M,N)
      IF(M*N.GT.640)GOTO 1
      WRITE DISC1,IBL,((A(I,J),J=1,N),I=1,M)
      RETURN
    1 L=640/N
      WRITE DISC 1, IBL,((A(I,J),J=1,N),I=1,L)
      L=L+1
      IBL1=IBL+1
      WRITE DISC 1, IBL1, ((A(I,J),J=1,N),I=L,M)
      RETURN
      END
      SUBROUTINE  RD(IBL, A,M,N,IA)
      DIMENSION A(IA,N)
      IF(M*N.GT.640)GOTO 1
      READ DISC 1, IBL,((A(I,J),J=1,N),I=1,M)
      RETURN
    1 L=640/N
      READ DISC 1, IBL,((A(I,J),J=1,N),I=1,L)
      L=L+1
      IBL1=IBL+1
       READ DISC1,IBL1,((A(I,J),J=1,N),I=L,M)
      RETURN
      END
       SUBROUTINE WD2(IBL,K,M,N)
       DIMENSION K(M,N)
       IF(M*N.GT.640)GOTO1
       WRITE DISC 1,IBL,((K(I,J),J=1,N),I=1,M)
       RETURN
    1   L=640/N
       WRITE DISC1, IBL,((K(I,J),J=1,N),I=1,L)
       L=L+1
       IBL1=IBL+1
       WRITE DISC1, IBL1,((K(I,J),J=1,N),I=L,M)
       RETURN
       END
```

```
      SUBROUTINE RD2(IBL,K,M,N,IA)
      DIMENSION K(IA,N)
      IF(M*N.GT.640)GOTO 1
      READ DISC1,IBL,((K(I,J),J=1,N),I=1,M)
      RETURN
 1     L=640/N
     READ DISC1,IBL,((K(I,J),J=1,N),I=1,L)
      L=L+1
      IBL1=IBL+1
      READ DISC1,IBL1,((K(I,J),J=1,N),I=L,M)
      RETURN
       END
```

```
'BEGIN' 'INTEGER' I,J,L,M,N,F,K,V,NOS,UNB,
LL, H,U,T,AC,TT,G,R2,R,
LAY1,LAY2,LAY3£
'REAL' MM,E,TIM,SUM1,SUM2,ZJW£
'BOOLEAN'Z£
'PROCEDURE' MOLP(M,N,F,R2,Z,K,MM,E,C,S,CP,COMP,OBJ,CODE,NP,BD,
NOSOLUTION, UNBOUNDED)£
'VALUE' M,N,F,K,MM£
'INTEGER'M,N,F,K,R2£
'REAL' MM,E£
 'BOOLEAN'Z£
'ARRAY'C,S,CP,NP,COMP£
'BOOLEANARRAY' OBJ£
'INTEGERARRAY'BD,CODE£
'LABEL'NOSOLUTION,UNBOUNDED£
'BEGIN' 'INTEGER' Q,P,I,J,L,V,IQN,
 U,
 FORM,ITR£
 'REAL'R,X£
 'ARRAY'TT(F'.'M+2),MULT(1'.'M)£
 'BOOLEANARRAY' UU(1'.'N)£
 'FOR'J=1'STEP'1'UNTIL'N'DO'
 UU(J)='TRUE'£
 FORM=F£
 ITR=1£
 V=1£
 NEXTITR'.'R=0£
 OUTPUT(30,S(FORM,1))£
 Q=0£
 'FOR'J=1'STEP'1'UNTIL'N'DO IF'UU(J)'THEN'
 'BEGIN' X='IF'FORM'EQ'1'THEN'0
 'ELSE'('IF'Z'THEN'C(K+1,J)'ELSE'C(V,J))£
 'FOR'I=1'STEP'1'UNTIL'M'DO'
 X=X+CP(FORM,I)*C(I+K+1,J)£
 'IF'X-R+E'LT'0'THEN BEGIN'R=X£Q=J'END'£
 'END'£
```

```
'FOR'J=1'STEP'1'UNTIL'M'DO'
'BEGIN'X='IF'CODE(J,1)'EQ'-1'OR'CODE(J,1)'EQ'0'THEN'
-CP(FORM,J)'ELSE' CP(FORM,J)£
'IF'X-R+E'LT'0'THEN BEGIN'R=X£Q=N+J'END'£
'END'£
'IF'Q'EQ'0'THEN BEGIN'
'IF'FORM'EQ'2'THEN''GOTO' EXIT£
'IF' FORM'EQ'1'AND' ABS(S(1,1))
 'LE' E*((M+ITR)**3)'THEN'
'BEGIN'FORM=2£ 'GOTO'NEXTITR'END'£
'IF' FORM'EQ'1'AND'S(1,1)'LT'0'THEN''GOTO'NOSOLUTION£
'END'£
R=MM£
P=0£
'IF'Q'LE'N'THEN'
'BEGIN'
'FOR'I=F'STEP'1'UNTIL'M+2'DO'
'BEGIN' X=('IF' I'EQ'2'THEN'
('IF'Z'THEN'C(K+1,Q)'ELSE'C(V,Q))
'ELSE'0)£
'FOR'J=1'STEP'1'UNTIL'M'DO'
X=X+CP(I,J)*C(J+K+1,Q)£
TT(I)=X£
'IF'I'GE'3'THEN' 'BEGIN'
'IF'X'GT'E'AND'S(I,1)'LT'R*X'THEN'
'BEGIN' R=S(I,1)/X£  P=I£  'END'£
'END' 'END'£
'END ELSE'
'BEGIN'
'IF'Q'GT'N'THEN' IQN=Q-N£
'FOR'I=F'STEP'1'UNTIL'M+2'DO'
'BEGIN'X='IF'CODE(IQN,1)'EQ'-1'OR'CODE(IQN,1)'EQ'0'THEN'
-CP(I,IQN)'ELSE'CP(I,IQN)£
TT(I)=X£
'IF' I'GE'3'AND'X'GT'E'AND'S(I,1)'LT'R*X'THEN'
'BEGIN'R=S(I,1)/X£P=I'END'
```

```
'END END'£
'IF'P'EQ'O'THEN GOTO'UNBOUNDED£
TT(P)=1/TT(P)£
'FOR'J=1'STEP'1'UNTIL'M'DO'CP(P,J)=CP(P,J)*TT(P)£
S(P,1)=S(P,1)*TT(P)£
'FOR'I=F'STEP'1'UNTIL'M+2'DO'
'IF'P'NE'I'THEN'
'BEGIN'
S(I,1)=S(I,1)-S(P,1)*TT(I)£
'FOR'J=1'STEP'1'UNTIL'M'DO'CP(I,J)=CP(I,J)-CP(P,J)*TT(I)£
'END'£
'IF' BD(P-2)'LE'N'THEN'UU(BD(P-2))='TRUE'£
'IF'Q'LE'N'THEN'UU(Q)='FALSE'£
BD(P-2)=Q£
ITR=ITR+1£
'GOTO'NEXTITR£
EXIT'.'
'FOR' I=1'STEP'1'UNTIL'M'DO'
'BEGIN' WRITE(30,LAYOUT('('-NDD')'),BD(I))£
WRITE(30,LAYOUT('('-NDDDDD.DDC')'),S(I+2))£ 'END'£
WRITET(30,'('SOLVED'('C')'')')£
'IF'R2'EQ'O'ORNOT'Z'THEN'
'BEGINFOR'I=1'STEP'1'UNTIL'M'DO'
'BEGIN' MULT(I)='IF'CODE(I,1)'EQ'O'THEN'CP(FORM,I)'ELSE'
CODE(I,1)*CP(FORM,I)£
'IF' MULT(I)'LE'E'THEN'OBJ(V)='FALSE'£ 'END'£
'FOR'J=1'STEP'1'UNTIL'R2-1,R2+1'STEP'1'UNTIL'K'DO'
'FOR'L=1'STEP'1'UNTIL'N'DO'
'FOR'I=1'STEP'1'UNTIL'M'DO'
'IF'BD(I)'LE'N
'AND'BD(I)'EQ'L'THEN'
NP(V,J)=NP(V,J)+C(J,L)*S(I+2,1)£
'FOR'J=1'STEP'1'UNTIL'K'DO'
WRITE(30,LAYOUT('('SS-D.DDDDDDD'E'+ND')'),NP(V,J))£
NEWLIN(30,1)£
'END'
```

```
'ELSE''IF'Z'THEN'
'FOR'J=1'STEP'1'UNTIL'K'DO'
'FOR'L=1'STEP'1'UNTIL'N'DO'
'FOR'I=1'STEP'1'UNTIL'M'DO'
'IF'BD(I)'LE'N'AND'BD(I)'EQ'L'THEN'
COMP(J)=COMP(J)+C(J,L)*S(I+2,1)£
R=0£
'IF'V'EQ'K'OR'Z'THEN''GOTO' FIN 'ELSE' 'BEGIN'
'FOR'L=1'STEP'1'UNTIL'N'DO'
'IF' 'NOT'UU(L)'AND'C(V+1,L)'NE'0'THEN'
'BEGIN' 'IF'R'EQ'0'THEN'
'BEGIN' R=R+1£
S(FORM,1)=0£
'FOR'J=1'STEP'1'UNTIL'M'DO'
CP(FORM,J)=0.0£
'END'£
'FOR'U=1'STEP'1'UNTIL'M'DO'
'FOR'I=3'STEP'1'UNTIL'M+2'DO'
'IF'BD(I-2)'LE'N
'AND'BD(I-2)'EQ'L'THEN'
'BEGIN' CP(FORM,U)=CP(FORM,U)+
CP(I,U)*(-C(V+1,L))£
S(FORM,1)=S(FORM,1)+S(I,1)*(-C(V+1,L))£
'END'£ 'END'£
V=V+1£ 'GOTO' NEXTITR£ 'END'£
FIN'.'
'END'£
 Z='FALSE'£
 MM=2**39£E='E'-6£
 LAY1=LAYOUT('('-NDDD')')£
 LAY2=LAYOUT('('S-D.DDDDDDDD'E'+ND')')£
 LAY3=LAYOUT('('S-D.DDDDDDDD'E'+NDC')')£
 R2=0£  U=READ(40)£  LL=0£
 'IF'U'GT'0'THENGOTO'W£
 D'.'N=READ(40)£'IF'N'LE'0'THEN GOTO'OUT£
 M=READ(40)£  F=READ(40)£   K=READ(40)£
```

```
T=TT=M£
WRITET(30,'(''('C')'VARIABLES'*'CONSTRAINTS'*'OBJECTIVES'('C')'')')£
OUTPUT(30,N)£    OUTPUT(30,M)£    OUTPUT(30,K)£
NEWLIN(30,1)£
'IF'U'EQ'O'THEN'
'BEGIN' 'BOOLEANARRAY' OBJ(1'.'K)£
'ARRAY' COMP,GAMA,PY(1'.'K),NP(1'.'K,1'.'K)£
'FOR'I=1'STEP'1'UNTIL'K'DO' OBJ(I)='TRUE'£
LABEL'.'
'FOR'I=R2+1'STEP'1'UNTIL'K'DO'
'IF'R2'NE'O'ANDNOT'OBJ(I)'THEN'
'BEGIN'
LL=LL+1£
M=M+1£   R2=I£
'GOTO'Y£ 'END'£
'END'£
Y'.' 'BEGINARRAY' C(1'.'M+K+1,1'.'N),
S,S1(1'.'M+2,1'.'1),
CP(1'.'M+2,1'.'M)£
'INTEGERARRAY' BD(1'.'M),CODE(1'.'M,1'.'1)£
'IF' Z'THEN' 'BEGIN'
RD(0,C,T+K+1,N,M+K+1)£
RD2(2,CODE,T,1,M)£
G=TT+K+2£
'IF'U'EQ'O'THENBEGIN'
'FOR'J=1'STEP'1'UNTIL'N'DO'
'FOR' I=1'STEP'1'UNTIL'K'DO'
C(K+1,J)='IF'J'EQ'N'THEN'1'ELSE'C(K+1,J)+C(I,J)£
'FOR'J=1'STEP'1'UNTIL'N-1'DO'
C(K+1,J)=-E*C(K+1,J)£
'END'£
'FOR'I=G'STEP'1'UNTIL'M+K+1'DO'
'FOR'J=1'STEP'1'UNTIL'N'DO'
C(I,J)='IF'J'EQ'N'THEN'1'ELSE'
'IF'I'LE'G+K-1'AND'PY(I-(G-1))'NE'O'THEN'
C(I-(G-1),J)*PY(I-(G-1))'ELSE'
```

```
'IF'U'GT'O'AND'I'GE'G+K-R'THEN'C(I-(G-1),J)£
J=O£
'FOR'I=TT+1'STEP'1'UNTIL'M'DO'
'BEGIN' CODE(I,1)=-1£
J=J+1£
'IF'NP(J,J)'GT'O'THEN'
'BEGIN'CODE(I,1)=1£
C(I+K+1,N)='IF'PY(J)'NE'O'AND'I'LE'TT+K-R'THEN'-1'ELSE'O£'END'
'ELSE' 'IF'PY(J)'NE'O'AND'I'LE'TT+K-R'THEN'
'FOR'L=1'STEP'1'UNTIL'N-1'DO'
C(I+K+1,L)=-C(I+K+1,L)£ 'END'£
F=1£
RD(1,S1,T+2,1,M+2)£
T=M£  L=TT+2+(K-R)£
'FOR'I=TT+3'STEP'1'UNTIL'M+2'DO'
'FOR'J=1'STEP'1'UNTIL'K'DO'
'BEGIN'
'IF'PY(J)'NE'O'THEN'
S1(I,1)=('IF'I'LE'L'THEN'PY(J)*ABS(NP(J,J))
'ELSE' 'IF'I'GT'L'AND'I'LE'L+(K-R)'THEN'COMP(J)
)
'ELSEIF'PY(J)'EQ'O'THEN'
S1(I,1)=COMP(J)-COMP(J)*READ(40)£ 'END'£
'FOR'I=F'STEP'1'UNTIL'M+2'DO'
S(I,1)=S1(I,1)£
'GOTO' B£
W'.' 'IF'Z'AND'U'GT'O'THEN'
'BEGIN' 'ARRAY' COMP,GAMA,PY(1'.'K)£
RD1(3,K,T,N,M,GAMA,COMP)£
AC=O£
SUM1=O.O£
W1'.' H=READ(40)£
'IF'H'EQ'O'THEN''GOTO'W2'ELSE'
'BEGIN'
GAMA(H)=PY(H)=O.O£
AC=AC+1£   OUTPUT(30,H)£
```

```
'GOTO'W1£
'END'£
W2'.'  'FOR'I=1'STEP'1'UNTIL'K'DO'
 SUM1=SUM1+GAMA(I)£
'FOR'I=1'STEP'1'UNTIL'K'DO'
'IF'GAMA(I)'GT'O'THEN'
PY(I)=GAMA(I)/SUM1£
'IF'U'EQ'1'THEN''BEGIN'
M=M+K-AC£  R=AC£ 'END ELSE'
'BEGIN' M=M-AC£   R=R+AC£'END'£
'GOTO'Y£
'END'£
'END'
 'ELSEIF''NOT'Z'THENBEGIN'
'IF'R2'EQ'O'THENBEGIN'
'FOR' I=1'STEP'1'UNTIL'M+K+1'DO'
'FOR' J=1'STEP'1'UNTIL'N'DO'
'BEGIN'C(I,J)=O£
'IF'I'LE'K'THEN'
'FOR'L=1'STEP'1'UNTIL'K'DO'
NP(I,L)=0.0£
'END'£
'FOR'L=1'STEP'1'UNTIL'K'DO'
COMP(L)=0.0£
'FOR'I=F'STEP'1'UNTIL'M+2'DO'
S(I,1)=S1(I,1)=0.0£
'FOR'I=1'STEP'1'UNTIL'M'DO'
CODE(I,1)=READ(40)£
A'.' I=READ(40)£
'IF'I'EQ'-1'THENGOTO'B£
'IF'I'LT'-1'OR'I'GT'M+K+1'THEN' 'GOTO'ERROR
£
J=READ(40)£'IF'J'LT'O'OR'J'GT'N'THEN GOTO'ERROR£
'IF'J'EQ'O'THEN'
S(I,1)=S1(I,1)=READ(40)
'ELSE''IF'J'LE'N-1'THEN'
```

```
C(I,J)= READ(40)£
'GOTO'A£
'END'
'IF'R2'NE'0'THEN'
'BEGIN' 'FOR'I=1'STEP'1'UNTIL'N-1'DO'
'BEGIN'
C(K+M+1,I)='IF'LL'EQ'1'THEN'C(R2,1)'ELSE'-C(R2,I)£
'IF'LL'EQ'1'THEN' 'FOR'J=1'STEP'1'UNTIL'K'DO'
C(J,I)=-C(J,I)£
'END'£
S1(M+2,1)=S(M+2,1)=ABS(NP(R2,R2))£
CODE(M,1)=0£
'END'£
'GOTO'B£
'END'£
B'.' 'FOR' I=1'STEP'1'UNTIL'M'DO'
BD(I)=N+I£
'FOR' I=F'STEP'1'UNTIL' M+2'DO'
'FOR'J=1'STEP'1'UNTIL'M'DO'
CP(I,J)='IF'I-2'EQ'J'THEN'1'ELSE'0£
'FOR' I=1'STEP'1'UNTIL'M'DO'
'BEGIN'
'IF'Z
'AND'CODE(I,1)'EQ'-1'OR'CODE(I,1)'EQ'0'THEN'
'BEGIN'
CP(1,I)=-1£
'IF'I'LE'M-2'THEN'
S1(F,1)=S1(F,1)-S1(I+(M-K+2),1)£
'END'
'ELSEIFNOT'Z'AND'CODE(I,1)'EQ'-1'OR'CODE(I,1)'EQ'0'THEN'
'BEGIN' CP(1,I)=-1£
S1(F,1)=S1(F,1)-S1(I+2,1)£
'END'£
S(F,1)=S1(F,1)£
'END'£
TIM=TIME(E)£
```

```
 'FOR'I=1'STEP'1'UNTIL'M+2'DO'
 'FOR'J=1'STEP'1'UNTIL'M'DO'
OUTPUT(30,CP(I,J))£
'FOR'I=1'STEP'1'UNTIL'M+K+1'DO'
'FOR'J=1'STEP'1'UNTIL'N'DO'
OUTPUT(30,C(I,J))£
'FOR'I=1'STEP'1'UNTIL'M+2'DO'
OUTPUT(30,S(I,1))£
'FOR'I=1'STEP'1'UNTIL'M'DO'
OUTPUT(30,CODE(I,1))£
'FOR'I=1'STEP'1'UNTIL'M'DO'
OUTPUT(30,BD(I))£
MOLP(M,N,F,R2,Z,('IF'Z'THEN'1'ELSE'K),MM,E,C,S,CP,COMP,OBJ,
CODE,NP,BD,NOSOLUTION,UNBOUNDED)£
'IF NOT' Z 'THEN'
'BEGIN'
'IF'R2'EQ'0'THENGOTO'LABEL'ELSEBEGIN'
'FOR'I=R2+1'STEP'1'UNTIL'K'DO'
'IFNOT'OBJ(I)'THEN'
'BEGIN' M=TT£  'GOTO'LABEL£  'END'£
'FOR'J=1'STEP'1'UNTIL'K'DO'
'FOR'I=1'STEP'1'UNTIL'N-1'DO'
C(J,I)=-C(J,I)£
R2=0£
'END'£
SUM2=0.0£
'FOR'J=1'STEP'1'UNTIL'K'DO'
'BEGIN'
SUM1=0.0£
'FOR'L=1'STEP'1'UNTIL'N'DO'
SUM1=SUM1+C(J,L)**2£
ZJW=NP(J,J)£
'FOR'I=1'STEP'1'UNTIL'K'DO'
'IF'NP(I,J)'GT'ZJW'THEN' ZJW=NP(I,J)£
GAMA(J)= (ABS(NP(J,J)-ZJW))/
(
```

```
('IF'ABS(NP(J,J))'GT'ABS(ZJW)'THEN'
ABS(NP(J,J))'ELSE'ABS(ZJW))*
SQRT(SUM1))£
SUM2=SUM2+GAMA(J)£
'END'£
'FOR'L=1'STEP'1'UNTIL'K'DO'
PY(L)=GAMA(L)/SUM2£
'FOR'L=1'STEP'1'UNTIL'K'DO'
OUTPUT(30,GAMA(L))£
'FOR'J=1'STEP'1'UNTIL'K'DO'
OUTPUT(30,PY(J))£
Z= 'TRUE'£
M=TT£
WD(0,C,M+K+1,N)£
WD(1,S1,M+2,1)£
WD2(2,CODE,M,1)£
M=M+K£
'GOTO' Y£
'END'
'ELSE'
'BEGIN'
'FOR' I=1'STEP'1'UNTIL'K'DO'
'FOR'J=1'STEP'1'UNTIL'N-1'DO'
'FOR'L=1'STEP'1'UNTIL'M'DO'
'IF' BD(L)'LE'N-1'AND'BD(L)'EQ'J'THEN'
WRITE(30,LAYOUT('('SS-D.DDDDDDDDDD'E'+ND')'),C(I,J)*S(L+2,1))£
WD(0,C,M+K+1,N)£
WD(1,S1,M+2,1)£
WD2(2,CODE,M,1)£
WD1(3,K,T,N,M,GAMA,COMP)£
'END'£
TIM=TIME(E)-TIM£
'GOTO'CONT£
NOSOLUTION'.'WRITET(30,'('NO'*'SOLUTION')')£
'GOTO'CONT£
UNBOUNDED'.'WRITET(30,'('UNBOUNDED')')£
```

```
CONT'.'
WRITET(30,'(''('C')'PRIMAL'*'SOLUTION'('C')'')')£
'FOR'I=F'STEP'1'UNTIL'M+2'DO'
'BEGIN' 'IF' I'LT' 3'THEN'
SPACE(30,5)'ELSE' WRITE(30,LAY1,BD(I-2))£
WRITE(30,LAY3,S(I,1))£
'END'£
WRITET(30,'(''('C')'SIMPLEX'*'MULTIPLIERS'('C')'')')£
'FOR'I=0'STEP'1'UNTIL'(M-1)'/'10'DO'
'BEGIN FOR'J=1,J+1'WHILE'J'LE'10'AND'J+I*10'LE'M'DO'
WRITE(30,LAY2,'IF'I*10+J'LE'M'THEN'
CP(2,I*10+J)'ELSE'-CP(2,I*10+J))£
NEWLIN(30,1)£
'END'£
WRITET(30,'(''('C')'TIME'*'TAKEN'*'')')£
WRITE(30,LAYOUT('('NDD.DDD')'),TIM)£
WRITET(30,'(''*'SECS'('CC')'')')£
'GOTO'D£
ERROR'.'WRITET(30,'(''('C')'DATA'*'ERROR'('C')'')')£
'END'£OUT'.''END'
      SUBROUTINE WD(IBL,A,M,N)
      DIMENSION A(M,N)
      IF(M*N.GT.640)GOTO 1
      WRITE DISC1,IBL,((A(I,J),J=1,N),I=1,M)
      RETURN
    1 L=640/N
      WRITE DISC 1, IBL,((A(I,J),J=1,N),I=1,L)
      L=L+1
      IBL1=IBL+1
      WRITE DISC 1, IBL1, ((A(I,J),J=1,N),I=L,M)
      RETURN
      END
      SUBROUTINE  RD(IBL, A,M,N,IA)
      DIMENSION A(IA,N)
      IF(M*N.GT.640)GOTO 1
      READ DISC 1, IBL,((A(I,J),J=1,N),I=1,M)
```

```
      RETURN
    1 L=640/N
      READ DISC 1, IBL,((A(I,J),J=1,N),I=1,L)
      L=L+1
      IBL1=IBL+1
       READ DISC1,IBL1,((A(I,J),J=1,N),I=L,M)
      RETURN
      END
       SUBROUTINE WD1(IBL,M,K,N,L,Z,X)
       DIMENSION Z(M),X(M)
       WRITE DISC1,IBL,(M,K,N,L),(Z(I),I=1,M),(X(I),I=1,M)
        RETURN
       END
       SUBROUTINE RD1(IBL,M,K,N,L,Z,X)
       DIMENSIONZ(M),X(M)
       READ DISC1,IBL,(M,K,N,L),(Z(I),I=1,M),(X(I),I=1,M)
       RETURN
       END
       SUBROUTINE WD2(IBL,K,M,N)
       DIMENSION K(M,N)
       IF(M*N.GT.640)GOTO1
       WRITE DISC 1,IBL,((K(I,J),J=1,N),I=1,M)
       RETURN
1      L=640/N
       WRITE DISC1, IBL,((K(I,J),J=1,N),I=1,L)
       L=L+1
       IBL1=IBL+1
       WRITE DISC1, IBL1,((K(I,J),J=1,N),I=L,M)
       RETURN
       END
       SUBROUTINE RD2(IBL,K,M,N,IA)
       DIMENSION K(IA,N)
       IF(M*N.GT.640)GOTO 1
       READ DISC1,IBL,((K(I,J),J=1,N),I=1,M)
       RETURN
1      L=640/N
```

```
READ DISC1,IBL,((K(I,J),J=1,N),I=1,L)
 L=L+1
 IBL1=IBL+1
 READ DISC1,IBL1,((K(I,J),J=1,N),I=L,M)
 RETURN
  END
```

APPENDIX

Here I will give the simple model constructed mainly by P. B. Clark in an Appendix p. 116 in his book "Planning Import Substitution", where he gives 10 constraints and a single objective function (max total consumption). But the model listed below will contain five constraints including the one given by Clark. The new four objectives are constructed according to the large scale model explained in Chapter One of this project.

I could not obtain any solutions to this small model since its size is relatively too large to be handled by hand, on the other hand I could not use the computer programme MOLP which produces only the first part of the required solution without solving the first compromise.

$$\max Z_1 = C + I_1 + I_2 + I_3 + I_4 - M_1 - M_2 - M_3 - M_4 \quad \text{(approx G.D.P.)}$$

$$\max Z_2 = C$$

$$\max Z_3 = .1x_1 + .7x_2 + .5x_3 + .9x_4 \quad \text{(v.a.)}$$

$$\max Z_4 = M_1 + M_2 + M_3 + M_4 \quad \text{(competitive imports)}$$

$$\max Z_6 = I_1 + I_2 + I_3 + I_4 \quad \text{investment}$$

s.t

$$-.56C - .23I_1 - .2I_2 - .28I_3 - .25I_4 + .3M_1 + .7x_1 \geq 15$$

$$-.17C + 1.5M_2 - .3x_1 + .9x_2 \geq 50$$

$$-.11C + 2.2M_3 - .1x_1 - .1x_2 + .6x_3 \geq 0$$

$$-.16C + M_4 - .2x_1 - .1x_2 - .1x_3 + .9x_4 \geq 0$$

$$-I_1 + x_1 \leq 400$$

$$-I_2 + 0.5x_2 \leq 125$$

$$-I_3 + 2x_3 \leq 300$$

$$-I_4 + 4x_4 \leq 800$$

$$M_1 + M_2 + M_3 + M_4 \leq 90$$

$$.23I_1 + .2I_2 + .28I_3 + .25I_4 - M_1 - M_2 - M_3 - M_4 \leq 110$$

Bibliography

1. R. Benayoun et al., opt., Multiple objectives in linear programming, 7th Mathematical Symposium, Netherlands, Direction Scientifique, SEMA (METRA International), Paris, 1970.

2. P.B. Clark: Planning Import Substitution, Vol. 68, Amsterdam, 1970.

3. B. Contini and S. Zionts: Restricted Bargaining for Organisations with Multiple Objectives, Econometrica, Vol. 36, No. 2, April 1968.

4. F. Hanssmann: Operations Research in National Planning of Underdeveloped countries, The National Cash Register Co., Dayton, Ohio, 1960.

5. S. Josefsen: LPMODSIMPLEX, BIT Contribution No. 10, 4 (1964)

6. C.J. Vaneijk and J. Sandee: Quantitative Determination of an Optimum Economic Policy, Econometrica, Vol.27, January 1959.

7. B. Roy: Introduction to the Sessioni Multiple Objective Functions, 7th Mathematical Programming Symposium, 1970, The Hague, September 14th-18th, SEMA (METRA International), Paris.