ON THE SOLUTION OF A CLASS OF CAPITAL INVESTMENT PROBLEMS

by

ARCADIO MANUEL GAMBOA-MEDINA

Submitted in partial fulfilment

of the requirements for the degree

of

DOCTOR OF PHILOSOPHY

The University of Aston in Birmingham
Department of Mathematics

OCTOBER, 1978

ON THE SOLUTION OF A CLASS OF CAPITAL INVESTMENT PROBLEMS

by

ARCADIO MANUEL GAMBOA-MEDINA

Submitted for the degree of

DOCTOR OF PHILOSOPHY

October, 1978

SUMMARY

In this work the solution of a class of capital investment problems is considered within the framework of mathematical programming. Upon the basis of the net present value criterion, the problems in question are mainly characterized by the fact that the cost of capital is defined as a non-decreasing function of the investment requirements. Capital rationing and some cases of technological dependence are also included, this approach leading to zero-one non-linear programming problems, for which specifically designed solution procedures supported by a general branch and bound development are presented. In the context of both this development and the relevant mathematical properties of the previously mentioned zero-one programs, a generalized zero-one model is also discussed. Finally, a variant of the scheme, connected with the search sequencing of optimal solutions, is presented as an alternative in which reduced storage limitations are encountered.

CAPITAL INVESTMENT,
OPERATIONAL RESEARCH
OPTIMIZATION

ACKNOWLEDGEMENTS.

The author wishes to acknowledge the always present readiness to help, of all the members of the Department of Mathematics of the University of Aston in Birmingham. He is particularly indebted to Dr.A.M.Rajput, his research supervisor, as well as to Mr.N.Kerruish, Reader and Senior Tutor of the Department, and to Professor D.Lawden, Head of Department. His gratitude to Mrs.M.Hasleton, Senior Secretary of the Department, for her excellent typing of this thesis.

The financing of this work was supported by the Bank of Mexico (BANXICO) and the National Council of Science and Technology of Mexico (CONACYT).

CONTENTS.

CONTENTS (contd)

CONTENTS (contd)

CONTENTS (contd)

CHAPTER I

INTRODUCTION

1.1) THE CAPITAL INVESTMENT PROBLEM.

Investment, in the sense with which this work is concerned, is usually defined in the dictionaries as "any placing of money to secure income or profit". Narrow and incomplete as it may seem, this definition contains the very characteristic nature of what the attitude of an investor would be. Most certainly, it only accurately reflects the traditional behaviour of capitalistic entities, but it is also true that, regardless of means, ways and philosophical differences, it features in general a very important practice pertaining to achievements of goals, commonly, although not necessarily, measured in terms of money. To analyse the concept of investment to the level of formally structuring a precise and flexible definition is not the task of the present discussion. For its purposes, an investment should simply be interpreted as any placing of money with which the achievement of a goal, with subsequent flows of money as defining elements, is associated. While accepting that this concept is closely related to that of the dictionaries, it should be noted that the basis to generalize the definition is now included. Scarce resources could be placed instead of money, and no specification has been imposed upon the character of the goal. In the context of this work, however, the maximization of the present worth will be referred to as the goal, under the assumption that its degree of achievement is measurable by the resulting subsequent flows of money. This position might not be applicable in general, but it is doubtlessly valid for most purposes within Western type mixed economies.

The general investment process[1] includes the following four phases:

1. Identification of the need to invest.
2. Identification of ways (investment proposals or projects) to satisfy the need to invest.

3. Appraisal of the identified projects.

4. Selection of projects to invest in.

Individuals, private firms and national bodies normally know that an appropriate course of action to increase their present worth is to invest. On the other hand, they also know that a wrong decision may lead to catastrophic results, and this is certainly the main reason upon which many decision makers support their tendency to look for alternative ways of improvement. These alternatives, however, are usually rather conservative in relative terms and, in many cases, insufficient. Arising from either subjective or objective considerations, the fact of qualifying this set of alternatives as unsatisfactory, marks the beginning of the investment process. It is precisely at this point that the need to invest becomes meaningful, giving place to the necessity of counting with a rational guide to approach the problem. In theory, the problem is to determine, subject to the prevailing circumstances, a feasible set of investment decisions maximizing the contribution to the present worth. Such a concept is of course purely academic, since, at the present, it imposes unrealistic requirements regarding the knowledge of the existing environment. Nonetheless, it is "the" point of reference towards which the analysis is to be directed. Accordingly, three questions have to be answered: Which ways (or projects) will be considered for investment purposes? How will each project contribute to the present worth? Under feasibility considerations, which of these projects will define the set of investment decisions maximizing the contribution to the present worth? In the sense of logically answering these questions, the problem posed by the need to invest is solved when this is accomplished.

The first two phases of the general investment process have to be developed as a result of the experience and the initiative of the analysts, as, for obvious reasons, no realistic mathematical model in this connection is available. Along these lines, an assumption of this work will be that both the need to invest and a universe of a class of capital investment projects to satisfy this need have been identified. Consequently, the aim will be to analyse an aspect of what in the present study will be referred to as the Capital Investment Problem: Amongst the elements of a set of capital investment projects, which feasible combination should be selected in order to maximize the corresponding contribution to the present worth? As can be easily observed, this is nothing but a re-statement of the last question in the preceding paragraph, and the first step to attack the problem is to determine the way by which the projects are to be appriased. Under specific assumptions, in this work the net present value (NPV) - the basis of the most accepted criterion to handle this problem[2] - will be used as the appraisal measure.

1.2) THE CAPITAL GROWTH MODEL.

The supportung theory of the NPV is what in most texts on mathematics for finance is referred to as Compound Interest Theory[3]. In this section a formal development leading to the fundamental results of this theory is presented. Concepts related to capital growth processes will be mentioned, and, for illustrative purposes, they can be interpreted as concepts related to deposits of money in financial institutions.

DEFINITION 1.1 : Let it be considered that capital grows in a process at a unitary rate i in a fixed period of time. Referred to that period of time, i is said to be the EFFECTIVE RATE OF INTEREST.

LEMMA 1.1 : Let n be any natural number, and let it be assumed that capital grows in a process during n successive periods of time at an effective rate of interest i per period. Let C_o be the initial capital and C_j the capital at the end of the j-th period $(j = 1,2,\dots,n)$. Then $C_n = C_o(1+i)^n$.

Proof (by induction): Let n = 1. It has to be shown that

$$C_1 = C_o(1+i)^1.$$

From Definition 1.1, it follows that, after one time period, i additional units of capital will correspond to each unit of C_o. Accordingly, the total capital after one time period will be $C_o + iC_o$, or $C_o(1+i)^1$.

Let it now be supposed that Lemma 1.1 is valid for n = k. It has to be shown that $C_{k+1} = C_o(1+i)^{k+1}$. Following the same reasoning as above, given that at the beginning of the (k+1)-th time period the total capital is $C_o(1+i)^k$, at the end of that same period the total capital will be $C_o(1+i)^k + i\,C_o(1+i)^k$, or $C_o(1+i)^{k+1}$.

DEFINITION 1.2: Let m successive time subperiods define a time period. Referred to that time period, $i^{(m)}$ is said to be the NOMINAL RATE OF INTEREST, if $\frac{i^{(m)}}{m}$ is the effective rate in each of the m subperiods.

COROLLARY 1.1: Let i and $i^{(m)}$ be the effective and the nominal rates of interest in any time period, respectively. Then $1+i = \left(1+\frac{i^{(m)}}{m}\right)^m$.

Proof: This result follows directly from Lemma 1.1.

DEFINITION 1.3: For any natural number m, let $i^{(m)}$ be the nominal rate of interest in a time period. Referred to that time period, δ is said to be the FORCE OF INTEREST, if $\delta = \lim_{m \to \infty} i^{(m)}$.

LEMMA 1.2: Let i and δ be the effective rate of interest and the force of interest in any time period, respectively. Then $1+i = e^{\delta}$.

Proof: From Corollary 1.1, it follows that:

$$i^{(m)} = m[(1+i)^{\frac{1}{m}} - 1]$$

Hence,

$$\delta = \lim_{m \to \infty} m[(1+i)^{\frac{1}{m}} - 1] = \ln(1+i)$$

$$\therefore\ 1+i = e^{\delta}$$

LEMMA 1.3: In a capital growth process starting at time point zero, let C_t be the capital at time point t. For any non-negative real number t, if:

i) $\frac{dC_t}{dt}$ exists;

ii) $C_t \neq 0$; and

iii) the time subperiod $[t, t+\frac{1}{m}]$ has an associated effective rate of interest $\frac{i^{(m)}}{m}$, where m is any natural number;

then $C_t = C_0(1+i)^t$, the basic time period being defined by m successive time subperiods of length $\frac{1}{m}$.

Proof: Let $\Delta t = \frac{1}{m}$. Then, since by ii) $C_t \neq 0$, by iii) it may be stated that:

$$\frac{C_{t+\Delta t} - C_t}{C_t} = \frac{i^{(m)}}{m} = \Delta t\ i^{(m)}$$

$$\therefore\ i^{(m)} = \frac{C_{t+\Delta t} - C_t}{\Delta t\ C_t}$$

Therefore, taking i) into account, it may also be stated that δ

can be expressed in terms of C_t. Namely:

$$\delta = \lim_{m \to \infty} i^{(m)} = \frac{1}{C_t} \lim_{\Delta t \to 0} \frac{C_{t+\Delta t} - C_t}{\Delta t} = \frac{1}{C_t} \frac{dC_t}{dt}$$

$$\therefore \int_0^t \frac{1}{C_t} \frac{dC_t}{dt} dt = \ln C_t - \ln C_0 = \ln \frac{C_t}{C_0} = \delta t,$$

or $$C_t = C_0 e^{\delta t}$$

Finally, from Lemma 1.2, it follows that:

$$C_t = C_0(1+i)^t$$

DEFINITION 1.4: A time point marking the end of a capital growth process is said to be a REDEMPTION POINT.

DEFINITION 1.5: In a capital growth process starting at time point zero, if all possible redemption points t lead to a final capital $C_0(1+i)^t$, then the process is said to be a COMPOUND INTEREST PROCESS at an effective rate of interest i, the basic time period being defined by $t = 1$.

DEFINITION 1.6: In a compound interest process, C_t is said to be EQUIVALENT to C_T $(t,T \geq 0)$, if a real number τ exists, such that $C_T = C_t(1+i)^\tau$.

COROLLARY 1.2: In a compound interest process starting at time point zero, C_t is equivalent to $C_0 = C_t(1+i)^{-t}$, for any non-negative real number t.

Proof: This result follows as a direct consequence of Definition 1.6.

1.3) THE NET PRESENT VALUE

1.3.1) AN EXAMPLE

Let P be a capital investment project, defined in terms of:

i) an initial capital outlay (cost) C_0,

ii) a series of cash in-flows (benefits) $B_1, B_2, \ldots, B_m$, and

iii) a series of cash out-flows (costs) $C_1, C_2, \ldots, C_m$,

where $B_k (k \geq 1)$ and $C_k (k \geq 0)$ take place at the beginning of the (k+1)-th time period and m, called the PLANNING HORIZON, is any natural number.

Under normal circumstances, capital deposited in financial institutions can always be increased by the interest paid for its use. Therefore, taking into account the conditions regarding the acceptance of deposits (interests determined by means of a percentage referred to a time period), the capital growth in this kind of institution tends to define a compound interest process. Let it be assumed that $B_k \geq C_k$ $(k = 1,2,\ldots,m)$. Then, if P were to be considered as an alternative of depositing an owned capital of C_0 monetary units at an effective rate of interest c during m time periods, one way of handling the problem would be to think of the net cash in-flows $B_1-C_1, B_2-C_2, \ldots, B_m-C_m$ as the result of a compound interest process, referred to an initial capital $X = \sum_{k=1}^{m} (B_k-C_k)(1+c)^{-k}$. In this context, X would be the maximal initial capital outlay to secure an effective rate of interest at least as high as c, and hence P should be selected if:

$$C_0 < X ,$$

or, equivalently, if:

$$Y = \sum_{k=0}^{m} \frac{B_k - C_k}{(1+c)^k} > 0,$$

where $B_0 = 0$.

It can be easily seen that in this example Y is simply the difference between the equivalent present value of the future net cash in-flows at an effective rate of interest c and the actual present cash out-flow C_0. Consequently, Y can be interpreted as an equivalent present net cash flow of project P, if a minimal yield c is required. In this sense, Y is said to be the NPV of P.

1.3.2) CERTAINTY CONDITIONS.

In general, the NPV of a project is defined as the difference between the present value of its benefits and the present value of its costs, and, accordingly, the NPV CRITERION establishes that a contribution towards the maximization of the present worth is encountered, if the NPV turns out to be a benefit. Upon the basis of an appropriate measure of the actual costs and benefits, these concepts are logical and consistent as such, but the task of measuring the NPV in practice has not been a simple one. As in the previous example, due to the fact that it accurately represents the functioning of financial transactions, the model of the compound interest processes has been accepted as a correct way to obtain the present value of a stream of future cash flows. The associated rate of interest c (usually referred to as the DISCOUNT RATE or COST OF CAPITAL), however, has been a very controversial issue, ever since the publication of an article due to Modigliani and Miller[4] in which the conclusion that in perfect capital markets the cost of capital to a firm is independent of the financing used to raise capital funds for investment was reached. Durand: "Modigliani and Miller have cut out for themselves the extremely difficult, if not impossible, task of being pure and practical at the same time. Starting with a perfect market in a perfect world, they have taken a few steps in the direction of realism; but they have not made significant progress ..."[5]. Modigliani and Miller, replying: "... he has focussed on the apparent limitations of the perfect market model instead of trying to surmount these limitations by extending our basic approach."[6] Boness: "Perhaps the most exciting event in economic controversy was the publication of work by M.H.Miller and F.Modigliani[4,6,7,8] on the cost of capital and related problems. The controversy continues, judging from a session of the professional meetings at Pittsburgh in December, 1962, with more passion than reason."[9]

"The term cost of capital may be defined as the price paid by a firm for funds acquired from its capital suppliers."[10]... "While there is fairly general agreement concerning the usefulness of the concept and how it should be applied, there has been a fundamental lack of agreement on exactly what it is or how it should be measured."[11] ... "The definition of c as the cost of capital is only one way of expressing its nature and function and perhaps it is not the most useful way. Other descriptions of its role exist. Thus c has been referred to as (a) the minimum required rate of return on proposals using capital funds, (b) the cutoff rate for capital expenditure, (c) the "hurdle" rate or "target" rate of return which must be surpassed if capital-use is to be justified, (d) the financial standard"[12] ... "There has been considerable controversy over the rate at which public undertakings should discount future receipts and costs when appraising investment ... It was commonly assumed that investment needed to earn only enough to cover amortization and interest at a rate resembling the current yield of government securities, or the rate at which the industry borrowed from the Exchequer ... If public investment is financed by withdrawing through taxation money which might have been used - and which, having been withdrawn from consumption could in principle be used - to finance additional private investment, the real marginal cost of the capital invested in the public sector will be its opportunity cost, defined as the return that might have been earned by a marginal addition to private investment."[13]

In the present work no attempt to discuss or narrow the range of disagreement about the cost of capital will be made. Instead, it will be considered in terms of a number of imperfect capital markets assumptions likely to be encountered, particular in this relation, but general in regard to the context within which the problem could arise. The class of problems under consideration will be formulated on the basis of variants of the usual definition of the NPV and the NPV criterion for perfect capital markets which are next presented:

DEFINITION 1.7: Let P be an independent capital investment project, in the sense that its acceptance or rejection in no way affects other existing projects and vice versa. Then, if the cost of capital c is a constant,

i) The present value of the cash in-flows $B_1, B_2, \ldots, B_m$ (costs and benefits of P to be denoted as in Section 1.3.1) is given by

$$B = \sum_{k=1}^{m} \frac{B_k}{(1+c)^k}$$

ii) The present value of the cash out-flows $C_0, C_1, \ldots, C_m$ is given by:

$$C = \sum_{k=0}^{m} \frac{C_k}{(1+c)^k}$$

iii) The NPV of project P is given by:

$$NPV_p = B - C$$

THE NPV CRITERION: Accept P, if $NPV_p > 0$; reject it, otherwise.

Associated with perfect capital markets assumptions is an important implication. Capital funds supply at the market rate of interest (constant and equal to the cost of capital) is unbounded. Consequently, decision makers can accept as many profitable independent projects as they wish. This means that this kind of projects are automatically dealt with after the third phase of the general investment process, and that the fourth phase is irrelevant in this context.

Finally, it is pointed out that Definition 1.7 can be extended for non-independent projects in a straightforward fashion, but in reference to particular selections. Therefore, in this case the NPV criterion can only go as far as asserting whether an individual contribution towards the maximization of the present worth is being made or not.

1.3.3) UNCERTAINTY CONDITIONS.

The importance of the NPV as a quantitative measure may be summarized as follows:

1. It tells whether or not an investment proposal contributes towards the maximization of the present worth.
2. In terms of present monetary units, it determines the size of such a contribution.

Under deterministic assumptions and on the basis of an appropriate discount rate, these two points are meaningful, because the required information (and hence the NPV) is known with certainty. However, if a certain degree of uncertainty were associated with the information, then it would no longer be possible to determine whether a contribution would be achieved. Nevertheless, probabilistic statements and related parameters can serve as useful tools to deal with the problem. In this framework, it would be desirable to know the NPV, now a random variable, to such an extent that the following assertions could be made:

1. It leads to the probability of an investment proposal contributing towards the maximization of the present worth ($P\{NPV_p > 0\}$).
2. In terms of present monetary units, it determines the expected size of the contribution ($E\{NPV_p\}$).

In this regard, the distribution of the NPV was derived by Hillier[14] for a class of normally distributed net cash flows, and handled by Hertz[15] via computer simulation under specific subjective probability assumptions. As examples of two general approaches, that of Hertz has the advantages inherent to computer simulation, but both require complete probabilistic information in connection with the net cash flows. More recently, Díaz-Padilla[16,17] developed successfully a practical method using first order approximations[18] to obtain the mean and variance of the NPV. His approach is also based on subjective probability assumptions, but only the means, variances and covariances of the net cash flows are required. In any case, it should be observed that in uncertainty conditions the assertions involved are strictly probabilistic, and that, as a result, the NPV criterion as such cannot be established. Therefore, any criterion approaching the NPV criterion for deterministic conditions can be accepted as relatively valid, its level of validity being determined by the accuracy of the estimates used to measure subjective preferences. A simple and very logical alternative is to base the criterion on the sign of the mean restricting the size of the variance[19], but other related possibilities exist (Markowitz[20], for example, penalizes the mean with the variance in a linear fashion).

1.4) THE KNAPSACK PROBLEM.

In Section 1.3.2 it was noted that independent projects are particularly simple to handle in perfect capital markets. The associated conditions, however, are restrictive in a number of aspects. One of these aspects is referred to the supply of capital which, either internally or externally, is very often found to be bounded. In this case the funds for investment are limited, and hence, in general, not all the profitable projects can be accepted. In other words, if capital rationing is assumed, the role of the fourth phase of the general investment process becomes active.

Let $U = \{ P_1, P_2, \ldots, P_n \}$, a set of capital investment projects, be considered under the following assumptions:

i) Each project is indivisible, in the sense that it is either totally accepted or rejected.

ii) Each project P_j is defined by a sequence of cash in-flows $B_{j1}, B_{j2}, \ldots, B_{jm}$ and a sequence of cash out-flows $C_{jo}, C_{j1}, \ldots, C_{jm}$, where B_{jk} $(k \geq 1)$ and C_{jk} $(k \geq 0)$ take place at the beginning of the (k+1)-th time period.

iii) For any $j \in \{1,2,\ldots,n\}$ and any $k \in \{1,2,\ldots,m\}$, $B_{jk} > C_{jk}$

iv) All the projects are TECHNOLOGICALLY INDEPENDENT, in the sense that, apart from capital rationing and desirability limitations, the acceptance or rejection of any one of them in no way affects the possibility of accepting or rejecting any one of the others.

v) Acceptances and rejections do not affect the size of any of the defining cash flows.

vi) The cost of capital is given by the constant c.

vii) The NPV of each project P_j (NPV_j) is positive.

If perfect capital markets conditions were further assumed, then the projects would be independent and, as was previously mentioned, all the projects could be accepted. However, if such conditions were restricted by assuming that an active limit b on the capital expenditures exists, then a non-trivial capital investment problem would have to be solved. Under the assumption of this kind of capital rationing, let the variable x_j $(j = 1,2,\ldots,n)$ be defined as follows:

$$x_j = \begin{cases} 1, & \text{if } P_j \text{ is accepted} \\ 0, & \text{if } P_j \text{ is rejected} \end{cases}$$

Then, if U is the inverse of investment proposals, the problem can be stated as:

$$\left.\begin{aligned} \text{Max } z &= \sum_{j=1}^{n} NPV_j x_j && (1.1)\\ \text{s.t.} \quad & \sum_{j=1}^{n} C_{jo} x_j \leq b && (1.2)\\ & x_j = 0 \text{ or } 1,\ j = 1,2,\ldots,n && (1.3)\end{aligned}\right\}(1.4)$$

where $NPV_j = \sum_{k=o}^{m} \frac{B_{jk}-C_{jk}}{(1+c)^k}$, $B_{jo} = 0$ and c is the cost of capital.

Referred to a hiker facing the decision of carrying a number of items within a limited weight capacity, problem (1.4) was named by Dantzig[21] as the KNAPSACK PROBLEM. He described a straightforward inspection rule to obtain the optimal solution to the continuous problem (the x_j's taking any value in the interval [0,1]), indicating that the rounded-off solution - a feasible solution to the original problem - should be probably satisfactory for most practical problems. Strictly speaking, however, this inspection rule was not new, as two years before the publication of Dantzig's article, Lorie and Savage[22] had already proposed the procedure, despite the fact that they were not explicitly dealing with a mathematical programming problem. In their section "Given a fixed sum for capital investment, what group of investment proposals should be undertaken?", Lorie and Savage approached problem (1.4) stating that acceptances should be made in decreasing order of the unitary net present values until exhausting the capital funds for investment. In a different framework, this is nothing but the rule suggested by Dantzig. In any case, Weingartner[23] was the first to identify capital rationing and indivisibilities as a mathematical programming problem, and, as such, the knapsack problem can be solved by any integer linear programming method[24]. Before Gomory's pioneer systematic cutting planes development in this field[25], Dantzig noted that the problem could be solved by means of dynamic programming[21], this idea being later exploited by Gilmore and Gormory[26,27,28,29]. Of course, cutting plane methods can also be used to solve the problem, as well as the more specialized and efficient Balas-type[30] algorithms . Nonetheless, it was the branch and bound approach of Land and Doig[31] the one which led to more efficient solution methods[32,33,34]. Specialized approximate procedures of fast convergence can be found in the works of Senju and Toyoda[35] and Toyoda[36].

1.5) <u>SUMMARY AND SCOPE OF THE STUDY</u>.

Within the context of the general investment process, the capital investment problem was defined in this chapter upon the basis of the NPV criterion. Preceded by a formal description of its mathematical model, the usual definition of this criterion for perfect capital markets was presented, pointing out its relevance in connection with the capital investment problem. The NPV and the NPV criterion were also discussed under conditions of uncertainty.

Under deterministic assumptions, perfect capital markets conditions were restricted in a nowadays classical example on the subject (the knapsack problem) by allowing capital funds to be rationed. In this work the solution to a number of variants of a generalized version of the knapsack problem, to be referred to as the MULTI-DIMENSIONAL KNAPSACK PROBLEM, is considered within the framework of mathematical programming. This problem was first studied by Weingartner[23] and can be stated as follows:

$$\left.\begin{aligned} &\text{Max } z = \sum_{j=1}^{n} NPV_j x_j \\ &\text{s.t.} \quad \sum_{j=1}^{n} a_{jk} x_j \leq b_k,\ k = 0,1,\ldots,m \\ &\qquad x_j = 0 \text{ or } 1,\ j = 1,2,\ldots,n, \end{aligned}\right\} \quad (1.5)$$

where all the parameters are non-negative constants.

The capital market conditions will be mainly characterized by the fact that the cost of capital will be defined as a non-decreasing function of the level of expenditure for investment. For practical purposes, this can be interpreted as a consequence of capital attraction being a non-decreasing function of capital productivity, or, equivalently, as a result of capital suppliers evaluating the investor's "intention and ability to repay"[37]. Under this assumption, the problems under consideration lead to zero-one non-linear programs, for which specifically designed solution procedures are presented. These solution methods are supported by a general branch and bound development, with which, together with the relevant mathematical properties of the previously mentioned zero-one programs, a generalized zero-one programming model is discussed. Finally, a variant of the scheme dealing with reduced storage limitations along the solution processes, is also presented.

REFERENCES.

[1] F.OCHOA-ROSSO (1973) Estructura del problema de selección óptima de inversiones. Felipe Ochoa and Associates (Mexico), Inc.Paper presented at the Seminar on Operations Research Applications to the Financial Sector, Mexico City, December, 1973.

[2] H.BIERMAN,JR. and S.SMIDT (1975) The Capital Budgeting Decision. Macmillan, New York.

[3] D.W.A.DONALD (1975) Compound Interest and Annuities-Certain. Heinemann, London.

[4] F.MODIGLIANI and M.H.MILLER (1958) The cost of capital, corporation finance, and the theory of investment. Am.Econ. Rev. 48, 261-297.

[5] D.DURAND (1959) The cost of capital, corporation finance, and the theory of investment: Comment. Am.Econ.Rev. 49, 639-655.

[6] F.MODIGLIANI and M.H.MILLER (1959) The cost of capital,corporation finance, and the theory of investment: Reply. Am.Econ.Rev. 49, 655-669.

[7] F.MODIGLIANI and M.H.MILLER (1961) Dividend policy, growth, and the valuation of shares. J.Bus.34, 411-432.

[8] F.MODIGLIANI and M.H.MILLER (1963) Corporate income taxes and the cost of capital: A correction. Am.Econ.Rev.53, 433-442.

[9] A.J.BONESS (1964) A pedagogic note on the cost of capital. The Journal of Finance 19, 99-106.

[10] J.C.T.MAO (1969) Quantitative Analysis of Financial Decisions. Macmillan, Toronto.

[11] G.D.QUIRIN (1967) The Capital Expenditure Decision. Richard D. Irwin, Inc.,Homewood, Illinois.

[12] E.SOLOMON (1963) The Theory of Financial Management. Columbia University Press, New York.

[13] J.L.CARR (1969) Investment Economics. Routledge and Kegan Paul Ltd., London.

[14] F.S.HILLIER (1963) The derivation of probabilistic information for the evaluation of risky investments. Mgmt.Sci.9, 443-457.

REFERENCES (contd)

[15] D.B.HERTZ (1964) Risk analysis in capital investment. Harvard Business Review 42, 95-106.

[16] J.DIAZ-PADILLA (1974) Análisis de inversiones bajo condiciones de incertidumbre. Felipe Ochoa and Associates (Mexico), Inc.Paper presented at the National Convention of the Mexican Institute of Chemical Engineers, Guadalajara, Mexico, November, 1974.

[17] J.DIAZ-PADILLA (1976)Evaluación bajo incertidumbre de projectos hidráulicos. Felipe Ochoa & Associates (Mexico), Inc. Paper presented at the Fourth National Congress of Hydraulics, A.M.H.,Acapulco, Mexico, October, 1976.

[18] J.R.BENJAMIN & C.A.CORNELL (1970) Probability, Statistics, and Decisions for Civil Engineers. McGraw-Hill, New York.

[19] J.CORD (1964) A method for allocating funds to investment projects when returns are subject to uncertainty. Mgmt.Sci.10, 335-341.

[20] H.M.MARKOWITZ (1959) Portfolio Selection: Efficient Diversification of Investments. John Wiley & Sons, Inc., New York.

[21] G.B.DANTZIG (1957) Discrete variable extremum problems. Ops.Res.5, 266-277.

[22] J.H.LORIE and L.J.SAVAGE (1955) Three problems in rationing capital. J.Bus. 28, 229-239.

[23] H.M.WEINGARTNER (1963) Mathematical Programming and the Analysis of Capital Budgeting Problems. Prentice-Hall, Englewood Cliffs, New Jersey.

[24] H.A.TAHA (1975) Integer Programming: Theory, Applications, and Computations. Academic Press, New York.

[25] R.E.GOMORY (1958) Outline of an algorithm for integer solutions to linear programs. Bull.Amer.Math.Soc.64, 275-278.

[26] P.C.GILMORE and R.E.GOMORY (1961) A linear programming approach to the cutting stock problem. Ops.Res.9, 849-859.

[27] P.C.GILMORE and R.E.GOMORY (1963) A linear programming approach to the cutting stock problem - Part II. Ops.Res.11,863-888.

REFERENCES (contd)

[28] P.C.GILMORE and R.E.GOMORY (1965) Many stage cutting stock problems of two or more dimensions. Ops.Res.13,94-120.

[29] P.C.GILMORE and R.E.GOMORY (1966) The theory of computation of knapsack functions. Ops.Res.14, 1045-1074.

[30] E.BALAS (1965) An additive algorithm for solving linear programs with zero-one variables. Ops.Res.13, 517-546.

[31] A.H.LAND and A.DOIG (1960) An automatic method for solving discrete programming problems. Econometrica 28, 497-520.

[32] P.KOLESAR (1967) A branch and bound algorithm for the knapsack problem. Mgmt.Sci.13, 723-735.

[33] F.OCHOA-ROSSO (1968) Applications of Discrete Optimization Techniques to Capital Investment and Network Synthesis Problems. Sc.D.Dissertation, Department of Civil Engineering, Massachusetts Institute of Technology, Boston,Massachusetts.

[34] H.GREENBERG and R.L.HEGERICH (1970) A branch search algorithm for the knapsack problem. Mgmt.Sci.16, 327-332.

[35] S.SENJU and Y.TOYODA (1968) An approach to linear programming with 0-1 variables. Mgmt.Sci.15, B196-B207.

[36] Y.TOYODA (1975) A simplified algorithm for obtaining approximate solutions to zero-one programming problems. Mgmt.Sci.21, 1417-1427.

[37] A.G.HART (1940) Anticipations, uncertainty and dynamic planning. Studies in Business Administration 11, 1. University of Chicago Press, Chicago.

CHAPTER II

A BRANCH AND BOUND SOLUTION SCHEME FOR FINITE PROGRAMMING PROBLEMS.

2.1) INTRODUCTION

Ever since the publication of Land and Doig's article[1] to solve integer linear programming problems, branch and bound algorithms have constituted an important part of integer programming theory. As in the cutting planes method of Gomory[2], this approach starts relaxing the integer restrictions on the variables. The original feasibility region is of course a subset of the continuous space, and so the continuous optimal solution is also optimal to the integer program, if it satisfies the integer restrictions. In this case the problem is solved. Otherwise, the continuous space is partitioned into a collection of subspaces on the basis of necessary conditions for integer solutions, disregarding those parts of the space for which the original problem is not feasible. As a result, this collection of subspaces will contain all the feasible solutions to the integer program. The associated continuous subproblems may then be treated in the same way as the original continuous problem. After repeated application of the procedure, an optimal solution to the integer program is given by an optimal solution to the subproblem with the best optimal objective value (amongst all the generated subproblems) satisfying the integer restrictions. Each of these optimal objective values defines a bound to the objective value corresponding to the integer solutions of the associated subspace, and, together with other criteria, they allow implicit consideration of many subproblems involving feasible integer solutions. The name BRANCH AND BOUND, due to Little et al.[3], is referred to the role of partitioning and bounding, respectively.

After the work of Land and Doig, pioneers in the development of branch and bound algorithms, extensive study of applications and improvements of this approach followed[4,5], including the generalization of its underlying principle[4,6,7,8,9].

In this chapter, the mathematical structure of a branch and bound solution scheme is presented. The scheme, subsequently to be implemented for the particular problems under consideration, is also discussed in the context of the branch and bound principle, as stated by Balas[8].

2.2) THE DIRECTED TREE.

Let the following optimization problem be considered:

$$\left.\begin{array}{l} \text{Max } z = f(s) \\ \text{s.t } s \in S \end{array}\right\} \quad (2.1)$$

where:

i) S is a subset of T with more than one element,

ii) T is a finite subset of Y,

iii) Y is an arbitrary set, and

iv) $f: Y \rightarrow \mathbb{R}$, $\mathbb{R}$ being the set of real numbers

DEFINITION 2.1: S and f are said to be the FEASIBILITY REGION and the OBJECTIVE FUNCTION of problem (2.1), respectively. The elements of Y are called SOLUTIONS, those of S FEASIBLE SOLUTIONS, and those of S for which f is maximal (in S) OPTIMAL SOLUTIONS.

An optimal solution s* to problem (2.1) clearly exists, because S is finite and non-empty. As usual in the framework of branch and bound algorithms, the proposed method to find such a solution is a search procedure which can be interpreted as the generation of a directed tree. The concepts and properties linking this approach with both problem (2.1) and the proposed solution method are next presented.

DEFINITION 2.2: Let C be the collection of subsets of T with more than one element, and let P be a function, such that $P:C \rightarrow D$, where D is the collection of sets with non-empty subsets of T as elements. P will be said to be a PARTITIONING FUNCTION, if for any $c \epsilon C$, $P(c)$ is a partition of C (the notation used in the definitions will hereafter be referred to the corresponding concepts).

DEFINITION 2.3: Let E be the collection of non-empty subsets of T. If N is an injective function, such that $N:E \rightarrow \mathbb{N}$ ($\mathbb{N}$ being the set of the natural numbers), then N(e) will be said to be a NODE, for any $e \epsilon E$.

DEFINITION 2.4: Let $e_1 \epsilon C$ and $e_2 \epsilon E$. The ordered pair $[N(e_1),N(e_2)]$ will be said to be a DIRECTED ARC, if, and only if, $e_2 \epsilon P(e_1)$.

LEMMA 2.1: Let $\mathcal{I}m(N)$ be the image of N, SN a non-empty subset of $\mathcal{I}m(N)$, and SA a set of directed arcs. If:

i) $N(T) \epsilon SN$,

ii) $n_2 \epsilon SN \Longrightarrow (n_1,n_2) \epsilon SA$, where $n_2 \neq N(T)$ and $n_1 \epsilon SN$, and

iii) $(n_1,n_2) \epsilon SA \Longrightarrow n_1,n_2 \epsilon SN$,

then $TR = \{SN,SA\}$ is a finite directed rooted tree, N(T) being the root of the tree.

Proof: See Appendix CH II

COROLLARY 2.1: For any $n \epsilon SN$, if $n \neq N(T)$, then one and only one directed path from N(T) to n exists in TR.

Proof: This result follows from Lemma AII.1, Definition AII.15 (see Appendix CHII) and Lemma 2.1.

COROLLARY 2.2: Every $n \epsilon SN$ belongs to one and only one level of TR.

Proof: This result follows from Corollary 2.1 and Definition AII.17 (see Appendix CHII).

COROLLARY 2.3: Let N^{-1}, with $\mathcal{I}m(N)$ as domain, be the inverse function of N. Then, if $n_1, n_2 \in \ell(j)$ and $n_1 \neq n_2$, then $N^{-1}(n_1) \cap N^{-1}(n_2) = \phi$, for any level $\ell(j)$ of TR.

Proof: Let $\underline{p}_1$ and $\underline{p}_2$ be the directed paths from N(T) to n_1 and n_2, respectively (by Corollary 2.1, they exist and are unique). If $j = 1$, $N^{-1}(n_1) \cap N^{-1}(n_2) = \phi$, because both $N^{-1}(n_1)$ and $N^{-1}(n_2)$ are elements of P(T). Otherwise, let m_i be the second component of $\underline{p}_i$ $(i = 1,2)$. Again, $N^{-1}(m_1)$ and $N^{-1}(m_2)$ are elements of P(T), and hence $N^{-1}(m_1) \cap N^{-1}(m_2) = \phi$. Consequently, given that $N^{-1}(n_i)$ is a proper subset of $N^{-1}(m_i)$ $(i = 1,2)$, the required result follows.

DEFINITION 2.5: TR will be said to be COMPLETE, if the following conditions hold:

i) For any $n \in SN$, if $n \notin TN$ and $e \in P[N^{-1}(n)]$, then $N(e) \in SN$

ii) For any $n \in TN$, $N^{-1}(n) \cap C = \phi$,

where TN is the set of terminal nodes (see Appendix CH II)

COROLLARY 2.4: Let $V = \{x/x = N^{-1}(n) \text{ and } n \in TN\}$, and let TR be complete. Then T=V and $\#(V) = \#(TN)$

($\#$: cardinality).

Proof: This result follows from Corollary 2.1 and Definition 2.5.

DEFINITION 2.6: TR will be said to be FEASIBLE, if $s \in S \Rightarrow s \in N^{-1}(n)$, for some $n \in TN$. It will be said to be COMPLETELY FEASIBLE, if, in addition, $N^{-1}(n) \cap S \neq \phi$, for any $n \in TN$.

LEMMA 2.2: TR is feasible, if, and only if, $e \in P[N^{-1}(n)]$ and

$e \cap S \neq \phi \Rightarrow N(e) \in SN$, for any $n \in SN$, such that $n \notin \tilde{T}N$.

Proof: $\Rightarrow$) H : TR is feasible.

Let it be assumed that some non-terminal element n of SN exists for which $N(e) \notin \tilde{S}N$, where $e \in P[N^{-1}(n)]$ and $e \cap S \neq \phi$. This means that all the directed paths of the form $(n_1, \ldots, n_k)$, where $n_k \in TN$, are such that $N^{-1}(n_k) \cap e \cap S = \phi$. If all the terminal nodes were involved with these paths, then, since $e \cap S \neq \phi$, at least one element of S would not belong to $N^{-1}(n_k)$, for any $n_k \in TN$ (contradiction to H). Otherwise, let $\tilde{T}N_1$ be the set of the remaining terminal nodes, and let the path $(N(T), r_1, \ldots, r_\ell, n)$ be considered. The sets $N^{-1}(r_i)$ $(i = 1, 2, \ldots, \ell)$ contain e properly, and, by corollaries 2.2 and 2.3, if $n_k \in TN_1$, then $N^{-1}(n_k) \cap e = \phi$. Hence, again, $s \in S \not\Rightarrow s \in N^{-1}(n_k)$ for some $n_k \in \tilde{T}N_1$, and a contradiction to H is encountered.

$\Leftarrow$) H: $e \in P[N^{-1}(n)]$ and $e \cap S \neq \phi \Rightarrow N(e) \in SN$, for any $n \in SN$, such that $n \notin \tilde{T}N$.

Let it be assumed that TR is not feasible. Then an element s of S exists, such that $s \notin N^{-1}(n)$, for any $n \in \tilde{T}N$. Under this assumption, some level $\ell(j)$ in which $s \notin N^{-1}(m)$ has to exist, where m is a non-terminal element of $\ell(j)$ (otherwise, by H, TR would have to be feasible). However, since obviously $s \in T$, considering in succession the levels of the tree, it follows (also by H) that no such level $\ell(j)$ can exist. Therefore, T has to be feasible.

LEMMA 2.3: Let TR be feasible. TR is completely feasible if, and only if, $e \in P[N^{-1}(n)]$ and $e \cap S = \phi \Rightarrow$ $N(e) \notin SN$, for any $n \in \tilde{S}N$, such that $n \notin TN$.

Proof: $\Rightarrow$) H : TR is completely feasible

If a non-terminal element n of SN existed for which $N(e) \in SN$, where $e \in P[N^{-1}(n)]$ and $e \cap S = \phi$, then a path of the form $(n, n_1, \ldots, n_k)$ would exist, n_k being a terminal node and $N^{-1}(n_k)$ containing no elements of S. Hence, H would be contradicted.

($\Leftarrow$) H : $e \in P[N^{-1}(n)]$ and $e \cap S = \phi \Rightarrow N(e) \notin SN$, for any $n \in SN$, such that $n \notin TN$.

Let it be assumed that $m \in TN$ and that $N^{-1}(m) \cap S = \phi$ (or, equivalently, that TR is not completely feasible). Because $S \subset T$, in the path from N(T) to m has to be a component n, such that $N^{-1}(n) \cap S \neq \phi$ and $N^{-1}(n_1) \cap S = \phi$, where n_1 is also a component of the path and $(n, n_1) \in SA$. Hence, $N^{-1}(n_1) \in P[N^{-1}(n)]$, contradicting hypothesis H. Therefore, TR has to be completely feasible.

DEFINITION 2.7: TR will be said to be PARTIALLY COMPLETE, if the following conditions hold:

i) For any $n \in SN$, if $n \notin TN$, $e \in P[N^{-1}(n)]$ and $e \cap S \neq \phi$, then $N(e) \in SN$.

ii) For any $n \in TN$, $N^{-1}(n) \cap C = \phi$.

LEMMA 2.4: If TR is partially complete, then it is also feasible.

Proof: By condition i) of Definition 2.7, for every non-terminal node n, all the elements of $N^{-1}(n)$ belonging to S have to belong to the union of the elements $e_1, e_2, \ldots, e_k$ of $P[N^{-1}(n)]$, for which $N(e_i) \in SN (i = 1, 2, \ldots, k)$. Consequently, every $s \in S$ has to be an element of $N^{-1}(m)$, for some $m \in TN$.

DEFINITION 2.8: TR will be said to be SUFFICIENTLY COMPLETE, if it is completely feasible and partially complete.

COROLLARY 2.5: Let $V = \{x/x = N^{-1}(n) \text{ and } n \in \tilde{T}N\}$, and let TR be sufficiently complete. Then S=V and $\#(V) = \#(\tilde{T}N)$.

Proof: This result follows from Corollary 2.1 and Definition 2.8.

DEFINITION 2.9: TR will be said to be SUFFICIENT, if:

i) $s^* \in N^{-1}(n)$, where $n \in TN$ and f is maximal in S at s^*.

ii) The functions $g:\tilde{T}N \rightarrow \{n\}$ and $h:N^{-1}(n) \rightarrow \{s^*\}$ are known.

Clearly, a sufficient directed tree leads to an optimal solution to problem (2.1), this being the basis of the proposed procedure to find such a solution. In a most rudimentary approach, the construction of a complete directed tree would provide the required information, and, although with some improvement involved, the same can be stated in connection with partially and sufficiently complete directed trees. In these cases, all the subsets of S containing one single element would be associated with terminal nodes of the tree, and hence this would be equivalent to an explicit enumeration of the elements of S. The problem would be then to choose that or those elements of S for which f is maximal. Obviously, however, this is not the task of the solution method. The purpose will indeed be to find a sufficient directed tree, but as far apart from the different concepts of completeness as possible. In terms of enumeration, the goal can be described as finding an optimal solution by means of inspecting only a few elements of S. Relevant concepts in this regard and their properties are now introduced.

DEFINITION 2.10: Any element n of S_n for which $S_n = S \cap \tilde{N}^{-1}(n) \neq \phi$ will be said to be a FEASIBLE NODE.

DEFINITION 2.11: Let n be any feasible node and let Y_n be a subset of Y, such that $S_n \subseteq Y_n$. A function

DEFINITION 2.11: (contd) $Z_n : Y_n \rightarrow \mathbb{R}$ will be said to be an UPPER BOUNDING FUNCTION of n, if $Z_n(y) \geq f(y)$ for all $y \in S_n$, and $Z_n^* = \max_{y \in Y_n} \{Z_n(y)\}$ exists.

DEFINITION 2.12: TR will be said to be UPPER-BOUNDING, if $s^* \in N^{-1}(n)$, for some $n \in TN$, and an upper bounding function is associated with all the feasible elements of TN.

LEMMA 2.5: Let TR be upper-bounding, and let F be any subset of the set of feasible terminal nodes. If $s^* \in N^{-1}(m)$, for some $m \in F$, then $U_F = \max_{n \in F} \{Z_n^*\}$ is an upper bound to problem (2.1).

Proof: By definition, $Z_m(s^*) \geq f(s^*)$. Hence, $U_F \geq f(s^*)$.

COROLLARY 2.6: If TR is upper-bounding, then U_{FTN} is an upper bound to problem (2.1), FTN being the set of feasible terminal nodes.

Proof: $s^* \in N^{-1}(n)$, for some $n \in FTN$, because TR is upper-bounding. Consequently, the required result follows from Lemma 2.5.

DEFINITION 2.13: Let TR be upper-bounding. If, for any two feasible nodes n and m with which an upper-bounding function is associated, $(n,m) \in SA$ and $Z_n^* \geq Z_m^*$, then TR will be said to be CONSISTENTLY UPPER-BOUNDING.

DEFINITION 2.14: Let Z_n be an upper bounding function of any feasible node n, and let f_L be a known lower bound to problem (2.1). If $Z_n^* < f_L$, n will be said to be a REJECTED NODE (after Ochoa-Rosso[10]).

LEMMA 2.6: If n is a rejected node, then $s^* \notin N^{-1}(n)$.

Proof: Let s^*_n be such that f is maximal in S_n at s^*_n. By definition, $Z_n(s^*_n) \geq f(s^*_n)$ and $Z^*_n \geq Z_n(s^*_n)$. As a result, $Z^*_n \geq f(s^*_n)$ and $f_L > f(s^*_n)$. Hence, $s^* \notin \tilde{N}^{-1}(n)$.

DEFINITION 2.15: Let W be a collection of non-empty subsets of T, and let d be a function, such that $d: W \to S$. If $N^{-1}(n) \in W$ and $t_n = d[N^{-1}(n)] \in S_n$, then $\tilde{t}_n$ will be said to be an AUXILIARY SOLUTION of n, for any feasible node n.

DEFINITION 2.16: Let n be a feasible node. If $N^{-1}(n) \in W$ and $S_n = \{t_n\}$, then n will be said to be CONCLUDING.

DEFINITION 2.17: TR will be said to be LOWER-BOUNDING, if an auxiliary solution is associated with all the feasible terminal nodes. It will be said to be CONSISTENTLY LOWER-BOUNDING, if, in addition, $s^* \in \tilde{N}^{-1}(n)$, for some $n \in TN$.

A direct consequence of the previous definition is that, if TR is lower-bounding, lower bounds to problem (2.1) will be available (those corresponding to the values of f at the existing auxiliary solutions). Denoting by AN the set of nodes with which an auxiliary solution is associated, $L = \max_{n \in AN} \{f(t_n)\}$ will obviously be the BEST of these bounds.

DEFINITION 2.18: TR will be said to be BOUNDING, if it is upper- and lower-bounding. It will be said to be CONSISTENTLY BOUNDING, if, in addition, it is consistently upper-bounding.

LEMMA 2.7: If TR is bounding, then it is also consistently lower-bounding.

Proof: TR is lower-bounding and $s^* \in \tilde{N}^{-1}(n)$, for some $n \in TN$, because it is also upper-bounding. Hence, TR is consistently lower-bounding.

DEFINITION 2.19: Any feasible terminal node will be said to be ACTIVE (after Lawler and Wood[4]), if it is not known to be concluding or rejected.

LEMMA 2.8: Let TR be upper-bounding, and let A be the set of active nodes. If $A = \phi$, then TR is sufficient.

Proof: $s^* \in \tilde{N}^{-1}(n)$, for some $n \in TN$, because TR is upper-bounding. Let m be such a node. By Lemma 2.6, m cannot be a rejected node. Hence, it has to be a concluding node ($A = \phi$). This means that the auxiliary solution t_m is, in fact, s^*, or at least such that $f(t_m) = f(s^*)$. The definition of L would lead to node m, or to some other node for which the same reasoning holds.

LEMMA 2.9: Let TR be bounding. If $A \neq \phi$, then $L \leqslant U_A$ and U_A is an upper bound to problem (2.1).

Proof: Again, TR being upper-bounding, a terminal node m exists, such that $s^* \in \tilde{N}^{-1}(m)$. If $m \in A$, then, by Lemma 2.5, U_A is an upper bound to problem (2.1), and, since TR is also lower-bounding, L is a well defined lower bound to the problem. Hence, $L \leqslant U_A$. On the other hand, if $m \notin A$, by Lemma 2.6, m has to be a concluding node. In this case, as happens when $A = \phi$, $L = f(t_m) = f(s^*)$. Clearly, $L \leqslant U_A$ (otherwise, A could only have rejected nodes as elements) and the required result follows.

COROLLARY 2.7: If TR is bounding and:

i) $A = \phi$, or

ii) $A \neq \phi$ and $L = U_A$,

then it is sufficient.

Proof: This follows from lemmas 2.8 and 2.9.

The importance of bounding trees is summarized in Lemma 2.9 and Corollary 2.7. According to the corresponding statements, a bounding tree leads either to an optimal or to a feasible solution to problem (2.1), in which case an indication $(L-U_A)$ of how far it is from an optimal solution is also given. Clearly, the associated auxiliary solutions and upper bounding functions play a relevant role in this connection, and, due to the fact that the proposed algorithms will be based on iterative constructions of sufficient trees, it would be desirable to deal with consistently bounding trees. This being the case, the possibilities to find a sufficient tree, neither complete, nor partially or sufficiently complete, would certainly increase.

DEFINITION 2.20: Given problem (2.1), P,d and $\{Z_n | n \text{ is a mode}\}$, let ρ be a collection of consistently bounding trees with non-empty sets of active nodes, and let τ be the set of all directed trees. A function $B_r : \rho \to \tau$, such that, for any $TR \epsilon \rho$:

i) TR is a sub-tree (see Appendix CHII) of $B_r(TR) = \hat{TR}$,

ii) $n \epsilon \hat{TN} \Rightarrow$ either $n \epsilon TN$ or $N^{-1}(n) \epsilon P[N^{-1}(\ell)]$ and $S_n \neq \phi$, where $\ell \epsilon A$, $U_A = Z^*_\ell$ and $\hat{TN}$ is the set of terminal nodes of $\hat{TR}$, and

iii) $\hat{TR}$ is consistently bounding,

will be said to be a BRANCHING RULE.

DEFINITION 2.21: A function $B_o : \rho \to R \times R$, such that $B_o(TR) = (L, U_A)$, will be said to be a BOUNDING RULE.

The mathematical structure associated with the proposed solution methods for the class of problems under consideration has now been completed. These problems will be formulated in the form of problem (2.1), and, accordingly, followed by its connection with the branch and bound principle, a general statement of the algorithms will be presented in the next section.

General relevant comments regarding the proposed solution methods are the following (both "directed tree" and "tree" will be terms used in the sense of the particular concept introduced in Lemma 2.1, rather than in the general sense [see Appendix CH II]):

1. The procedure to find an optimal solution to problem (2.1) will be based on a search defined in terms of successive partitions of T and subsets of T. This will be achieved by means of successive applications of branching and bounding rules, by which a sufficient directed tree (to be called the final directed tree) will be constructed. Therefore, at any stage of the construction, a consistently bounding tree (referred to as intermediate, if the stage is not the last), and hence also a feasible solution to the problem, will be available. Additionally, intermediate trees will also be seen to be completely feasible.
2. Since any intermediate tree is a sub-tree both of intermediate trees corresponding to subsequent stages and of the final tree, improving upper and lower bounds will be associated with the development of the procedure (given that the defining elements of the lower bounds are the available auxiliary solutions, the current best feasible solution will also be improving). The result is a systematic possibility of implicitly inspecting subsets of T containing several feasible solutions, and it will be directly related with the current number of rejected nodes.
3. In the absence of implicit inspections along the construction process, the final tree would be sufficiently complete. As was mentioned before, in this extreme case every feasible solution would be associated with one and only one terminal

3. contd.

node of the tree, and vice versa (see Corollary 2.5), due to the fact that partitions are used to divide T and subsets of T. Also because of this feature, the search procedure will be sharp, in the sense that any subset of T associated with a node n (either of an intermediate or of the final tree) will only be contained in those subsets of T associated with the nodes defining the path from the root to node n (see corollaries 2.2 and 2.3).

2.3) THE ALGORITHM.

Given problem (2.1), and under the assumption that:

i) A branching rule (and hence also a bounding rule) is available,

ii) $A \neq \phi \Longrightarrow TR \epsilon \rho$, TR being a consistently bounding tree, and

iii) An initial consistently bounding tree TR_o is available,

the algorithm can be stated as follows (with an added sub-index referred to the iterations of the procedure, the same notation as that of the preceding section will be used).

START (ITERATION 0)

1. Obtain L_o and A_o.
2. If $A_o = \phi$, stop; the auxiliary solution associated with L_o is optimal. Otherwise, complete the bounding rule obtaining U_o (for convenience, U_{A_i} will be written as U_i).
3. If $L_o = U_o$, stop; the auxiliary solution associated with L_o is optimal. Otherwise, set i = 1 and proceed.

ITERATION i

1. Apply the branching rule B_r to TR_{i-1}, obtaining $TR_i = B_r(TR_{i-1})$
2. Obtain L_i and A_i.
3. If $A_i = \phi$, stop; the auxiliary solution associated with L_i is optimal. Otherwise, complete the bounding rule obtaining U_i.
4. If $L_i = U_i$, stop; the auxiliary solution associated with L_i is optimal. Otherwise, set $i \leftarrow i+1$ and start iteration i.

The algorithm clearly leads to an optimal solution to problem (2.1) in a finite number of iterations, because the elements of ρ are finite and consistently bounding. As usual, it should be observed that such a procedure could prove to be useful, only if relatively easy work is involved at each iteration (in terms of the difficulty associated with the original problem). This will be, therefore, a further assumption.

2.4) THE ALGORITHM AND THE BRANCH AND BOUND PRINCIPLE.

In this section, the connection between the proposed algorithm and the branch and bound principle is discussed. Referred to problem (2.1), this principle, as stated by Balas[8] (in turn based on Bertier and Roy[7]), consists in the fulfilment of the following three conditions:

i) There exists a finite superset T' of S and a function $w:T' \rightarrow \mathbb{R}$, such that $s \in S \Rightarrow w(s) = f(s)$.

ii) A function $B:C' \rightarrow D'$ (referred to as a BRANCHING RULE), such that $c \in C'$ and $B(c) = \{d_1, d_2, \ldots, d_q\} \Longrightarrow \bigcup_{i=1}^{q} d_i = c - \{t_k\}$, where C' is the collection of subsets

ii) contd.

of T' with more than one element, D' is the collection of sets with non-empty subsets of T' as elements, and $w(t_k) = \max_{t \in c} \{w(t)\}$, can be defined.

iii) For any $c \in C'$ and any $t \in c$, the UPPER BOUND $w(t_k)$ on $f(t)$ can be (easily) determined.

When these three conditions are satisfied, s* can be found by means of successive applications of B. At the beginning, T' is considered. If the corresponding upper bound $w(t_0)$ is such that $t_0 \in S$, then $f(t_0) = f(s^*)$ and the problem is solved. Otherwise, an element d_i of B(T') with maximal upper bound $w(t_1)$ is next considered, and the same reasoning is repeated. If t_1 is not an optimal solution, B is applied to d_i, and, from then on, in order to proceed in the same fashion, a current maximal upper bound $w(t_k)$ is obtained on the basis of all the subsets of T' generated by the applications of B, but not yet used as arguments of the branching rule. Hence, at any stage of the procedure, if $t_k \in S$, then $f(t_k) = f(s^*)$. Finally, because T' is finite and one of its elements is eliminated each time the branching rule is applied, an optimal solution has to be found in a finite number of steps.

The relations linking the conditions of the branch and bound principle and the structure associated with the proposed algorithm are next presented.

CONDITION i: This can be seen to be a particular case of the concept of upper bounding function (see Definition 2.11). Specifically, if $Z_i = Z_j$, for any $i \neq j$, and $Z_m(y) \not< f(y)$, for any n and any $y \in S$, then, since $T \subset Y_1$, this condition is satisfied ($T' = T$ and $w = Z_n$ restricted to T, for any n).

CONDITION ii: The function B corresponds to the role of the concept of partitioning function (see Definition 2.2), if C' = C and D' = D. This being the case, for any $c \in C$, the elements of B(c), although preferably, do not necessarily have to be mutually exclusive, as opposed to those of P(c). The main difference, however, is that B imposes the exclusion of t_k, whereas P does not. Now, if condition i) is satisfied, as was explained before, this exclusion is completely justified. Nevertheless, if this condition were relaxed to the extent that $s \in S \Rightarrow w(s) \geq f(s)$, then t_k could be feasible and non-optimal. Therefore, by Definition 2.11, t_k cannot be excluded in general under P. This function, on the other hand, only imposes the inclusion of all $e_i \in P(c)$, such that $e_i \cap S \neq \phi$, as a result of Definition 2.20. Obviously, this last feature could be incorporated to B without changing its fundamental role, as it merely states that, if a subset of T' is excluded, then none of its elements belongs to S.

CONDITION iii: In the proposed algorithm it has been assumed that Z_n^* can be (easily) determined for any feasible node n. Hence, if only completely feasible trees are involved (as will be seen to be the case) and condition i) is satisfied, then this condition is also satisfied. Otherwise, if condition i) is not satisfied, then Z_n^* would correspond to the role of $w(t_k)$ in a looser sense ($Z_n^* \geq w(t_k)$).

From the preceding discussion follows the fact that the proposed algorithm is defined on the basis of a more general context than that associated with the conditions of Balas' branch and bound principle, for these are only fulfilled under specific circumstances. In particular, when the first condition is satisfied (as indicated previously), the whole principle and its corresponding procedure to find s* are directly applicable, if t_k is excluded each time P is evaluated. Otherwise, however, t_k could not be excluded and the procedure would no longer be valid. In this case, successive applications of P would eventually lead to a complete enumeration of the elements of S, but improving upper bounds would still be available along the process. Consequently, if at any of its stages a feasible solution s for which $f(s) = U$ (U being the best current upper bound) were also available, the process would then come to an end, without necessarily having to complete the enumeration explicitly. The knowledge of feasible solutions is obviously equivalent to the knowledge of lower bounds to the original problem, and it is by all means a very useful knowledge. It not only provides a tool to solve the problem approximately when time and storage requirements are scarce, but it also can be used to estimate (together with the upper bounds) how large the incurred error might be. When Land and Doig first introduced their approach to solve integer linear programs[1], the idea of counting with lower bounds was not systematized, because by means of their algorithm feasible solutions, in general, are neither (easily) obtainable nor intended to be obtained. This problem has been dealt with in a variety of ways[10,11,12,13], and the concept itself was incorporated by Mitten[9] in his formulation of branch and bound methods. This formulation is a generalized version of both Balas' principle (although in a way more restrictive, precisely due to the requirement of a lower bounding rule) and Ochoa-Rosso's general branch and bound algorithm[10], in which, in the context of directed trees, lower bounds are supposed to be available.

Mitten also assumed the existence of the function w satisfying condition i), but the essence of his approach is mainly based on well defined improving upper and lower bounds and a BRANCH AND BOUND RECURSIVE OPERATION which divides elements of C', excluding subsets that are known either not to contain an optimal solution or to contain only non-feasible solutions. In this sense, the proposed algorithm coincides with the requirements of Mitten, the difference being a more flexible background to produce upper bounds. Again, this difference would not be present, if condition i) were satisfied. This condition is of course desirable to obtain smaller upper bounds, but it should be noticed that it could prove to be too restrictive, not allowing the use of, although looser, less costly and still effective upper bounds.

2.5) SUMMARY.

In this chapter a solution method for discrete programming problems was considered. This method belongs to a class of techniques which can be interpreted as the construction of a directed tree, and a precise mathematical formulation linking this concept and the proposed solution method was presented. The relation of the method with the branch and bound approach originally developed by Land and Doig to solve integer linear programming problems was analysed, on the basis of Balas' generalization of the principle[8]. It was shown that the defining conditions of this principle are only fulfilled for particular cases, and reference to the similarity between the method and a more general formulation (that of Mitten[6]) was also indicated. In both cases, the basic difference was seen to be the more general context within which the proposed solution method deals with the calculation of upper bounds. In the following chapters the application of the method to the previously mentioned capital investment problems will be discussed.

APPENDIX CH II

PROOF OF LEMMA 2.1

AII. 1) BASIC CONCEPTS OF GRAPHS.

The basic concepts of graph theory involved with the proof of Lemma 2.1 (see Section 2.2) are presented in this section. These concepts, with minor essential differences, may be found in any standard text on the subject[14].

DEFINITION AII.1: Let SN be a non-empty set and SA an arbitrary set. If a function $h: SA \rightarrow SN^2$ exists, SN^2 being the set of unordered pairs of elements of SN, then $G_h = \{SN, SA, h\}$ is said to be a GRAPH. The elements of SN are called the NODES, and those of SA the ARCS of G_h. If $SA = \phi$, then $\{SN, SA\}$ is said to be a NULL GRAPH.

DEFINITION AII.2: If SN and SA are finite, then G_h is said to be a FINITE GRAPH. Otherwise, it is said to be an INFINITE GRAPH.

DEFINITION AII.3: A graph $G'_g = \{SN', SA', g\}$ is said to be a SUB-GRAPH of G_h, if $SN' \subseteq SN$, $SA' \subseteq SA$ and $g(e) = h(e) \forall e \in SA'$.

DEFINITION AII.4: For any $e \in SA$, the nodes n_1 and n_2 are said to be the END NODES of e, if $h(e) = n_1, n_2$. If $n_1 = n_2$, e is said to be a SELF-LOOP.

DEFINITION AII.5: For any node n and any arc e, if n is an end node of e, then n and e are said to be INCIDENT on each other. For any node n, its DEGREE is defined as the number of arcs incident on n, with self-loops counted twice. Any node of degree one is referred to as a TERMINAL NODE.

DEFINITION AII.6: A WALK is defined as a finite sequence $\langle a_i \rangle_{i=1}^{N}$, where :

DEFINITION AII.6: (contd)
i) $a_i \in SN$, if i is odd,

ii) $a_i \in SA$, if i is even,

iii) a_i is incident on a_{i+1}, for $i = 1,2,\ldots,N-1$,

iv) $a_i, a_j \in SA \Longrightarrow i = j$ or $a_i \neq a_j$, and

v) N is an odd natural number.

DEFINITION AII.7: A walk is said to be CLOSED, if $a_1 = a_N$. Otherwise, it is said to be OPEN.

DEFINITION AII.8: A PATH is defined as an open walk in which $a_i, a_j \in SN \Longrightarrow i = j$ or $a_i \neq a_j$. If, with the exception of a_1 and a_N, this last property is satisfied and the walk is closed, it is then refered to as a CYCLE.

DEFINITION AII.9: A graph G_h is said to be CONNECTED, if for any pair n_1, n_2 of its nodes, a path with $a_1 = n_1$ and $a_N = a_2$ exists. It is said to be ACYCLIC, if no cycles can be defined in G_h.

DEFINITION AII.10: A graph is said to be a TREE, if it is connected and acyclic.

LEMMA AII.1: In a tree there is one and only one path between every pair n_1, n_2 of its nodes ($a_1 = n_1, a_N = n_2$).

Proof: This is a direct result of Definition AII.10.

DEFINITION AII.11: A tree is said to be ROOTED, if one of its nodes, called the ROOT, is distinguished from all the others. Given a path in a rooted tree, a_1 being the root of the tree, a_N is said to be at LEVEL $\ell\left(\frac{N-1}{2}\right)$ of the tree. A level $\ell(j)$ is constituted by all the nodes at that level.

DEFINITION AII.12: Given a graph G_h, let $g: \mathcal{I}m(h) \rightarrow SN \times SN$ be such that $g(n_1, n_2) = \overline{(n_1, n_2)}$, where $\mathcal{I}m(h)$ is the image of h and $SN \times SN$ is the Cartesian

DEFINITION AII.12: (contd) product of SN (i.e., with ordered pairs of SN as elements). Then $G_{goh}=\{SN,SA,goh\}$ is said to be a DIRECTED GRAPH, G_h being its associated UNDIRECTED GRAPH. The elements of SN are called the NODES, and those of SA the DIRECTED ARCS of G_{goh} (directed subgraphs, finite and infinite directed graphs are defined in the same way as that of graphs).

DEFINITION AII.13: For any directed arc e, n_1 is said to be the INITIAL NODE and n_2 the FINAL NODE of e, if $goh(e) = (n_1,n_2)$. In this case, e is said to be INCIDENT OUT OF n_1 and INCIDENT INTO n_2.

DEFINITION AII.14: For any node n of G_{goh}, the number of directed arcs incident out of (into) n is called the OUT-DEGREE (IN-DEGREE) of n. Any node of out-degree zero and in-degree one is referred to as a TERMINAL NODE.

DEFINITION AII.15: Let $<a_i>_{i=1}^{N}$ be a path in G_h. The path is said to be a DIRECTED PATH in G_{goh}, if $goh(a_i) = (a_{i-1},a_{i+1})$, for $i = 2,4,\ldots,N-1$.

DEFINITION AII.16: G_{goh} is said to be CONNECTED, if G_h is connected. Similarly, G_{goh} is said to be ACYCLIC, if G_h is acyclic.

DEFINITION AII.17: A directed graph is said to be a DIRECTED TREE, if its associated undirected graph is a tree. If the latter is rooted, then the directed tree is also said to be ROOTED. The ROOT and the LEVELS of the directed tree correspond to the root and the levels of the undirected tree.

AII.2 THE PROOF.

In this section, SN and SA will be used to denote the specific concepts of definitions 2.3 and 2.4 in Section 2.2.

LEMMA AII.2: SN and SA are finite.

Proof: Clearly, the number of elements in SN is bounded by the number of elements in E, which is a collection of subsets of a finite set (T). Hence, SN has to be finite. On the other hand, the elements of SA are ordered pairs of the elements of SN. Therefore, SA has to be finite too.

LEMMA AII.3: If $h:SA \rightarrow SN^2$ is such that $(n_1,n_2) \in SA \Longrightarrow h[(n_1,n_2)] = n_1,n_2$, then:

i) The inverse function g of h exists,

ii) $TR_h = \{SN,SA,h\}$ is a finite rooted tree,

iii) $TR_{goh} = \{SN,SA,goh\}$ is a finite directed rooted tree.

Proof: i) If $(n_1,n_2) \in SA$, then, by Definition 2.4, e_2 is a proper subset of e_1, where $n_1 = N(e_1)$ and $n_2 = N(e_2)$. This being the case, (n_2,n_1) cannot belong to SA, and hence h is an injective function. Therefore, its inverse function g exists.

ii) By Definition AII.1 and property iii) of Lemma 2.1, TR_h is a graph. Let n_1 and n_2 be any two elements of SN. Either if $n_1 = N(T)$ of if $n_2 = N(T)$, a path with $a_1 = n_1$ and $a_M = n_2$ exists, as a result of Lemma AII.2 and property ii) of Lemma 2.1. Otherwise, due to the same reason, a path with $a_1' = n_1$ and $a_{M_1}' = N(T)$ and a path with $a_1'' = n_2$ and $a_{M_2}'' = N(T)$ have to exist. For simplicity, under the consideration that the elements of SA are defined in terms of their end nodes, let these two paths be denoted by $\underline{p}_1 = (a_1',a_3',\ldots,a_{M_1}')$ and

ii) contd.

$\underline{p}_2 = (a_1'', a_3'', \ldots, a_{M_2}'')$, respectively. Again, either if $a_1'' = a_i'$, for some $i = 1,3,\ldots,M_1$, or if $a_1' = a_i''$, for some $i = 1,3,\ldots,M_2$, a path with $a_1 = n_1$ and $a_M = n_2$ exists. Otherwise, $a_i' = a_j''$ for some $i = 3,5,\ldots,M_1$ and some $j = 3,5,\ldots,M_2$, where $a_k' \neq a_\ell''$ for any $k < i$ and any $\ell < j$, because at least $a_{M_1}' = a_{M_2}'' = N(T)$. Hence, once again, a path with $a_1 = n_1$ and $a_M = n_2$ would have to exist. Therefore, TR_h is connected. On the other hand, TR_h is also acyclic, given that, for any walk $(a_1, a_3, \ldots, a_M)$ (following the same notation as that of $\underline{p}_1$ and $\underline{p}_2$), e_i is a proper subset of e_j if $i > j$, where $N(e_i) = a_i$ for $i = 1,3,\ldots,M$. Finally, $N(T)$ is distinguished from all the other elements of SN, taking into account that $(n_1, n_2) \in SA \Longrightarrow n_2 \neq N(T)$. Thus, TR is a rooted tree, which, by Lemma AII.2, is also finite.

iii) By Definition AII.12 and property i) of this lemma, TR_h is the associated undirected graph of TR_{goh}. Therefore, by Definition AII.17, TR_{goh} is a finite directed rooted tree, with $N(T)$ as its root. Again, as in the case of directed paths in which a simplified notation was introduced (in the preceding part of this proof), due to the fact that the elements of SA are ordered pairs of SN and that goh is the identity function, TR_{goh} will be simply denoted as $TR = \{SN, SA\}$. Moreover, also for simplicity, TR will be indistinctively be referred to as a directed tree or as a tree.

REFERENCES.

1 A.H.LAND and A.DOIG (1960) An automatic method for solving discrete programming problems. Econometrica 28, 497-520.

2 R.E.GOMORY (1958) Outline of an algorithm for integer solutions to linear programs. Bull.Amer.Math.Soc. 64, 275-278.

3 J.D.C.LITTLE, K.G.MURTY, D.W.SWEENEY and C.KAREL (1963) An algorithm for the traveling salesman problem. Ops.Res. 11, 972-989.

4 E.L.LAWLER and D.E.WOOD (1966) Branch-and-bound methods: A survey. Ops.Res.14, 699-719.

5 H.A.TAHA (1975) Integer Programming: Theory Applications, and Computations. Academic Press, New York.

6 J.AGAIN (1966) Optimum seeking with branch-and-bound. Mgmt.Sci.13, B176-B185.

7 P.BERTIER and B.ROY (1964) Procédure de résolution pour une classe de problèmes pouvant avoir un caractère combinatoire. Cahiers du Centre d'Études de RechercheOperationelle 6,202-208.

8 E.BALAS (1968) A note on the branch-and-bound principle. Ops.Res.16, 442-445 (Errata p.886).

9 L.G.MITTEN (1970) Branch-and-bound methods: General formulation and properties. Ops.Res.18,24-34.

10 F.OCHOA-ROSSO (1968) Applications of Discrete Optimization Techniques to Capital Investment and Network Synthesis Problems. Sc.D.Dissertation, Department of Civil Engineering, Massachusetts Institute of Technology, Boston, Massachusetts.

11 H.A.TAHA (1971) Solution of integer linear programs using cuts and imbedded zero-one problems. Technical Report No.71-5, Department of Industrial Engineering, University of Arkansas, Fayetteville.

12 J.E.MOORE (1974) An Improved Branch and Bound Algorithm for Integer Linear Proglems. Ph.D.Disertation, University of Arkansas, Fayetteville.

[13] J.J.H.FORREST, J.P.H.HIRST and J.A.TOMLIN (1974)
Practical solution of large mixed integer programming problems with UMPIRE. Mgmt.Sci.20,736-773.

[14] N.DEO (1974) Graph Theory with Applications to Engineering and Computer Science. Prentice-Hall, Englewood Cliffs, New Jersey.

CHAPTER III.

CAPITAL INVESTMENT ON A CLASS OF ECONOMICALLY DEPENDENT PROJECTS: THE BASIC ALGORITHM.

3.1) INTRODUCTION.

In this chapter the multi-dimensional knapsack problem presented in Section 1.5 will be discussed under more general conditions. Specifically, based on the principle that capital attraction is in direct relation with capital productivity, the cost of capital will not be assumed to be constant. Instead, it will be defined as a non-decreasing function of capital expenditure for investment. Under this extended assumption, the individual net present values of the projects vary according to the different values of the cost of capital, and hence economic dependence between projects is automatically introduced. In terms of the mathematical programming model, this means that a non-linear problem has to be solved. It will be seen that the form of this problem corresponds to that of problem (2.1) (see Section 2.2), and a procedure satisfying the requirements of the solution method described in Section 2.3 will be developed.

3.2) FORMULATION OF THE PROBLEM.

Let $U = \{P_1, P_2, \ldots, P_n\}$, a set of capital investment projects, be considered under the following assumptions:

i) Each project is indivisible.

ii) Each project is defined by a sequence of cash in-flows and a sequence of cash out-flows, as indicated in Section 1.4.

iii) For any project P_j, a natural number M_j ($< m$) exists, such that $B_{jk} < C_{jk}$, if $k \geq M_j$ ($B_{jo} = 0$), and $B_{jk} > C_{jk}$, if $k > M_j$. It will be assumed that, for any $k = 1, 2, \ldots, M_j$, the OPERATING REQUIREMENT $C_{jk} - B_{jk}$ is not large in terms of the INVESTMENT REQUIREMENT C_{jo}.

iv) For any project P_j, $C_{jo} - \sum_{k=1}^{M_j} (B_{jk}-C_{jk}) < \sum_{k=M_j+1}^{m} (B_{jk}-C_{jk})$. This, together with the preceding assumption, means that the internal rate of return IRR_j associated with each project P_j (see Appendix CHIII) exists. It will be assumed that the internal rates of return are relatively small in terms of the cost of capital (to be defined).

v) All the projects are technologically independent, and acceptances and rejections do not affect the size of the defining cash flows.

vi) Any final selection should not result in an overall investment requirement surpassing the limit b_o, nor in an overall operating requirement at the end of the k-th time period surpassing the limit $b_k (k=1,2,\ldots,M = \max_j\{M_j\})$. Overall operating requirements are simply defined as the sum of the corresponding individual operating requirements (in other words, overall operating requirements are not supposed to be compensated by positive net cash flows taking place at the time point under consideration).

vii) Capital funds to invest in the projects (and, if necessary, to keep them operating) are available at an associated cost of capital c(t), where t is the overall investment requirement, and c is a positive non-decreasing function of t, only if projects with positive NPV are accepted.

For any $k \leq M$ and any j, let the parameter a_{jk} be defined as follows:

$$a_{jk} = \begin{cases} C_{jk}-B_{jk}, & \text{if } k \leq M_j \\ 0, & \text{otherwise} \end{cases}$$

Then, again, if U is the decision maker's inverse of investment proposals, the problem can be stated as:

$$\left.\begin{array}{lll} \text{Max } Z = \sum_{j=1}^{n} NPV_j(t)x_j & & (3.1) \\ \text{s.t. } \sum_{j=1}^{n} a_{jk}x_j \leq b_k, \; k = 0,1,\ldots,M & & (3.2) \\ NPV_j(t) > 0, \text{ if } x_j=1, \; j= 1,2,\ldots,n & & (3.3) \\ x_j = 0 \text{ or } 1, \; j = 1,2,\ldots,n & & (3.4) \end{array}\right\} (3.5)$$

where $NPV_j(t) = \sum_{k=o}^{m} \frac{B_{jk} - C_{jk}}{[1+c(t)]^k}$ and $t = \sum_{j=1}^{n} C_{jo}x_j$

Clearly, if $c(t)$ were a constant, problem (3.5) would correspond to problem (1.5), which is Weingartner's generalization[1] of the Lorie and Savage problem[2] discussed in Section 1.4. In this case, all the $NPV_j(t)$'s would be constant, and direct elimination of non-promising projects (those with non-positive NPV) would be possible. Constraint (3.3) would therefore be unnecessary, and problem (3.5) would precisely take the form of problem (1.5). Otherwise, however, a number of additional considerations have to be taken into account. In the first place, simply because the $NPV_j(t)$'s are non-linear functions of $\underline{x} = (x_1,x_2,\ldots,x_n)$, the projects are now ECONOMICALLY DEPENDENT (the acceptance of any one of the projects can affect the contributions of individual net present values of the others), and the problem is obviously no longer linear. In qualitative terms, this does not change the form of the objective function, but direct elimination becomes restricted, as a result of the possibility of the $NPV_j(t)$'s taking negative or positive values, depending on the value of t. Under these circumstances, only projects with non-positive NPV for any feasible value of t can be regarded as non-promising. On the other hand, constraint (3.3) now has to be incorporated, in order to guarantee that an optimal solution to problem (3.5) will only include projects with positive NPV. The problem being a maximization problem, no optimal solution could include a project with negative

NPV. Nevertheless, without constraint (3.3), at least in theory, such a solution could include a project with NPV equal to zero. Finally, it is pointed out that, as was mentioned in Section 2.1, the original basis of branch and bound methods when solving integer linear programs[3] was the (relative) simplicity with which the associated continuous problem could be handled. For the special case of problem (1.5) this feature is very powerful[4], and, consequently, in the context of branch and bound methods, it is particularly useful[5,6,7]. In the case of problem (3.5), however, an easy-to-handle problem is not defined by allowing the x_j's to take any value in the interval [0,1],as c(t) is an arbitrary non-decreasing positive function. To overcome this problem, an alternative based on the interal rate of return (IRR) criterion (see Appendix CHIII) will be presented in this chapter.

3.3) DEVELOPMENT OF THE SOLUTION METHOD.

3.3.1) IDENTIFICATION OF THE PROBLEM.

Let S,T,Y and f be defined as follows:

$$S = \{\underline{x} \in E^n / (3.2)\text{-}(3.4) \text{ are satisfied}\}$$

$$T = \{\underline{x} \in E^n / (3.4) \text{ is satisfied}\},$$

$$Y = T, \text{ and}$$

$$f: Y \to \mathbb{R}, \text{ where } \underline{x} \in Y \Rightarrow f(\underline{x}) = \sum_{j=1}^{n} NPV_j(\underline{x}) x_j$$

The set T is clearly finite because there are only 2^n elements in E^n satisfying constraint (3.4). On the other hand, the objective function (3.1) is equal to $f(\underline{x})$ and S is a non-empty subset of T ($\underline{0} = (0,0,\dots,0) \in S$). Hence, if $\#(S) > 1$, then the form of problem (3.5) corresponds to that of problem (3.1) (see Section 2.2); otherwise, $\underline{0}$ is the optimal solution to problem (3.5). In the following section the concepts presented in Section 2.2 will be implemented for problem (3.5). Accordingly, they will be established for the cases in which $\#(S) > 1$.

3.3.2) THE DIRECTED TREE.

Using the same notation as that of Section 2.2, let $\mathbb{P}_j: C \to D$ be such that $e \in C \Rightarrow \mathbb{P}_j(e) = \{e_1, e_2\}$, where $e_1 = \{\underline{x} \in e / x_j = 1\}$ and $e_2 = \{\underline{x} \in e / x_j = 0\}$, for some given $j \in \{1, 2, \ldots, n\}$ (at least two elements $\underline{x}' = (x_1', x_2', \ldots, x_n')$ and $\underline{x}'' = (x_1'', x_2'', \ldots, x_n'')$ of e with $x_j' \neq x_j''$ have to exist for j to be suitable), and let $N : E \to \mathbb{N}$ be an injective function. $\mathbb{P}_j$ is obviously a partitioning function, and $\mathcal{I}m(N)$ is a set of nodes. Associated with each node ℓ will be the following concepts (to be referred to as the ASSOCIATED CONCEPTS of the node):

DEFINITION 3.1: If, for all $\underline{x} \in N^{-1}(\ell)$, either $x_j = 0$ or $x_j = 1$, x_j will be said to be a NON-FREE VARIABLE at node ℓ, for any $j \in \{1, 2, \ldots, n\}$. Otherwise, it will be said to be a FREE VARIABLE at node ℓ.

DEFINITION 3.2: $\underline{x}(\ell) = [x_1(\ell), x_2(\ell), \ldots, x_n(\ell)]$ will be said to be the ℓ-th SPECIFICATION of $\underline{x}$, if:

$$x_j(\ell) = \begin{cases} x_j, \text{ if } x_j \text{ is a non-free variable at node } \ell \\ 0, \text{ otherwise} \end{cases}$$

The objective value of $\underline{x}(\ell)$ will be denoted by $\hat{Z}_\ell$ and the associated cost of capital by $r(\ell)$. In particular, if $\underline{x}(\ell) = \underline{0}$, $r(\ell)$ can be conventionally fixed at $\min\{c(t)/\underline{x} \in T \text{ and } \underline{x} \neq \underline{0}\}$. This is a fictitious cost of capital representing a lower bound to the incurred cost of capital, if investment is to take place. Therefore, it can be associated with $\underline{x} = \underline{0}$ without introducing a conceptual or quantitative contradiction, so long as it is referred to solutions distinct from $\underline{0}$.

LEMMA 3.1: Let $FV(\ell)$ be a set of free variables at node ℓ (the elements of $FV(\ell)$ to be denoted by sub-indices of

LEMMA 3.1: (contd) the variables), such that $j \notin FV(\ell)$, only if x_j is known to be non-promising (i.e.,if $x_j = 1$ is known to lead to non-feasible solutions). Further, let Y_ℓ be a subset of $\bar{N}^{-1}(\ell)$, such that $\underline{x} \in \bar{N}^{-1}(\ell) - Y_\ell \Longrightarrow x_j$ is free at ℓ and $j \notin FV(\ell)$. If $Y_\ell \neq \phi$, $Z_\ell : Y_\ell \to \bar{\mathbb{R}}$

and $\underline{x} \in Y_\ell \Longrightarrow Z_\ell(\underline{x}) = \sum_{j=1}^{n} x_j \sum_{k=o}^{m} \frac{B_{jk} - C_{jk}}{[1+r(\ell)]^k}$, then

$Z_\ell(\underline{x}) \geq f(\underline{x})$ for all $\underline{x} \in Y_\ell$.

Proof: By Definition 3.2, associated with any element of $\bar{N}^{-1}(\ell)$ is a cost of capital at least as high as $r(\ell)$. Hence,

since $\sum_{k=o}^{m} \frac{B_{jk} - C_{jk}}{[1+ c(t)]^k}$ is a decreasing function of $c(t)$, for any $j \in \{1,2,\ldots,n\}$ (see Appendix CHIII), and $c(t)$ is a non-decreasing function of t, then

$\sum_{k=o}^{m} \frac{B_{jk} - C_{jk}}{[1+r(\ell)]^k} \geq \sum_{k=o}^{m} \frac{B_{jk} - C_{jk}}{[1+c(t)]^k}$, for any $\underline{x} \in \bar{N}^{-1}(\ell)$ and any

$j \in \{1,2,\ldots,n\}$. Therefore, $\sum_{j=1}^{n} x_j \sum_{k=o}^{m} \frac{B_{jk}-C_{jk}}{[1+r(\ell)]^k} \geq \sum_{j=1}^{n} x_j \sum_{k=o} \frac{B_{jk}-C_{jk}}{[1+c(t)]^k}$,

for any $\underline{x} \in \bar{N}^{-1}(\ell)$. In particular, this result holds for any $\underline{x} \in Y_\ell$. Consequently, the required result follows.

LEMMA 3.2: Let $Y(\ell)$ be a subset of $\{1,2,\ldots,n\}$, such that $j \in Y(\ell)$, if, and only if, either $x_j(\ell) = 1$, or $j \in FV(\ell)$ and $IRR_j > r(\ell)$. Then, for any node ℓ:

$$Z_\ell^* = \max_{\underline{x} \in Y_\ell} \{Z_\ell(\underline{x})\} = \begin{cases} \sum_{j \in Y(\ell)} \sum_{k=o}^{m} \frac{B_{jk} - C_{jk}}{[1+r(\ell)]^k}, & \text{if } Y(\ell) \neq \phi \\ \hat{Z}_\ell, & \text{otherwise} \end{cases}$$

Proof: $Z_\ell(\underline{x}) = \sum_{j=1}^{n} a_j x_j(\ell) + \sum_{j \in FV(\ell)} a_j x_j$, where $a_j = \sum_{k=o}^{m} \frac{B_{jk} - C_{jk}}{[1+r(\ell)]^k}$

$(j = 1,2,\ldots,n)$. If $Y(\ell) = \phi$ and $FV(\ell) \neq \phi$, then

Proof: (contd)

$IRR_j \leq r(\ell)$, for all $j \epsilon FV(\ell)$, which means that $a_j \leq 0$, for all $j\epsilon FV(\ell)$ (see Appendix CH III). Therefore, as happens when $FV(\ell) = \phi$, in this case $Z_\ell^* = \sum_{j=1}^{n} a_j x_j(\ell) = \hat{Z}_\ell$. Let it now be assumed that $Y(\ell) \neq \phi$, and that $Z_\ell(\underline{x}') = Z_\ell^*$, where $\underline{x}' \epsilon Y_\ell$, $x_j' = 0$ and $j\epsilon FV(\ell) \cap Y(\ell)$. Since $j\epsilon FV(\ell)$, $\underline{x}'' = (x_1', x_2', \ldots, x_{j-1}', 1, x_{j+1}', \ldots, x_n') \epsilon Y_\ell$ exists. Clearly, $a_j > 0$, because, j belonging to $Y(\ell)$, $IRR_j > r(\ell)$. Hence, $Z_\ell(\underline{x}'') > Z_\ell(\underline{x}')$, which contradicts the assumption that $Z_\ell(\underline{x}') = Z_\ell^*$. In other words, if $Z_\ell(\underline{x}') = Z_\ell^*$ and $j\epsilon FV(\ell) \cap Y(\ell)$, then $x_j' = 1$. Therefore, in this case

$$Z_\ell^* = \sum_{j=1}^{n} a_j x_j(\ell) + \sum_{j\epsilon FV(\ell)\ Y(\ell)} a_j x_j = \sum_{j\epsilon Y(\ell)} a_j =$$

$$\sum_{j\epsilon Y(\ell)} \sum_{k=0} \frac{B_{jk} - C_{jk}}{[1+r(\ell)]^k}.$$

COROLLARY 3.1: If ℓ is a feasible node, then Z_ℓ is an upper bounding function of ℓ.

Proof: $S_\ell \subset Y_\ell$. Hence, by lemmas 3.1 and 3.2, the required result follows.

It is now possible to start dealing with the assumptions of the algorithm proposed in Section 2.3. To begin with, the procedure to find the initial tree TR_0 will be presented. This procedure will always lead to TR_0, unless $\underline{0}$ is the only element of S.

CONSTRUCTION OF THE INITIAL TREE (PROCEDURE P1)

Step 1: Define $N(T) = 1$ and obtain the associated concepts of node 1.

Step 2: Select an element j of FV(1) ($FV(1) = \{1,2, \ldots, n\}$ at the beginning) with maximal internal rate of return IRR_j (suggested tie breaking rule: maximal investment).

Step 3: If $IRR_j \leq r(1)$, stop; $\underline{x}^* = \underline{0}$ is the optimal solution to problem (3.5).

Otherwise continue.

Step 4: If $IRR_j \leq c(C_{jo})$, go to step 6.

Otherwise, continue.

Step 5: If $a_{jk} \leq b_k$, for $k = 0,1,\ldots,M$, define $N(t_1) = 2$ and $N(t_2) = 3$, where $\mathbb{P}_j(T) = \{t_1,t_2\}$. The procedure is complete: $SN_o = \{1,2,3\}$, $SA_o = \{(1,2),(1,3)\}$ and $TR_o = \{SN_o,SA_o\}$.

Otherwise, continue.

Step 6: Subtract $\{ j \}$ from FV(1).

If $FV(1) \neq \phi$, go to step 2.

Otherwise, stop; $\underline{x}^* = \underline{0}$ is the optimal solution to problem (3.5).

The underlying justifications regarding this procedure are the following:

Case 1: $IRR_j \leq c(C_{jo})$

Bearing in mind that the NPV of a project can only be positive if its IRR is greater than the cost of capital, and that c is a non-decreasing function of t, it is clear that constraint (3.3) would not be satisfied, if $x_j = 1$. Therefore, $\{j\}$ is subtracted from FV(1), which, as will be seen further on, is equivalent to reject P_j.

Case 2: $a_{jk} > b_{jk}$, for some $k = 0,1,\ldots,M$

Because all the a_{jk}'s are non-negative, it would be impossible to avoid the violation of constraint (3.2), if P_j were accepted. Again, $\{j\}$ has to be subtracted from FV(1), rejecting P_j.

Case 3: The procedure stops when $FV(1) = \phi$.

This is merely an extension of the preceding cases, leading to the rejection of all projects.

Case 4: The procedure stops at step 3.

The reasoning justifying Case 1 is also applicable, since $r(1) \leqslant c(t)$, for any $t \geqslant 0$. In addition, in this case there is no other non-rejected project with a greater IRR, and so all the projects have to be rejected.

LEMMA 3.3: TR_0 is a completely feasible directed tree, and $\underline{x}(\ell)$ is an auxiliary solution of ℓ, for any $\ell \epsilon SN_0$.

Proof: TR_0 is obviously a directed tree. On the other hand, steps 3,4 and 5 of P1 gurantee that $\underline{x}(2) \epsilon S$, and $\underline{x}(3)$ equals $\underline{x}(1)$, which is also an element of S. Hence, $S_2 = S \cap N^{-1}(2)$ and $S_3 = S \cap N^{-1}(3)$ are both non-empty. Finally, the two elements of $\tilde{\mathbb{P}}_j(T)$ correspond to the two elements of TN_0, which means that TR_0 is completely feasible. That $\underline{x}(\ell)$ is an auxiliary solution of ℓ ($\ell = 1,2,3$) follows then from the fact that $\underline{x}(\ell) \epsilon S_\ell$.

LEMMA 3.4: Together with its associated concepts, TR_0 is consistently bounding.

Proof: By Corollary 3.1 and Lemma 3.3, TR_0 is upper-bounding and consistently lower-bounding. Additionally, both domains of Z_2 and Z_3 are subsets of the domain of Z_1, and $r(2), r(3) \geqslant r(1)$. Therefore, $Z_2^*, Z_3^* \leqslant Z_1^*$. This means that TR_0 is consistently upper-bounding, and hence also consistently bounding.

In summary, it has been seen that, after performing procedure P1, either the optimal solution ($\underline{x}^* = \underline{0}$) to problem (3.5) is found, or assumption iii) of the proposed solution method (see Section 2.3) is satisfied. This being the case, $L_0 = \hat{Z}_2$, and A_0 has to be determined. The following is a procedure to obtain the set of active nodes A_i to be considered, for any iteration i (observe that A_i is not necessarily uniquely determined, due to the way active nodes are defined).

DETERMINATION OF ACTIVE NODES (PROCEDURE P2).

Step 1: Set $A_i = TN_i - C_i - R_i$, where C_i is the set of terminal nodes which are known to be concluding ($C_0 = \phi$ at the beginning), and R_i is the set of terminal rejected nodes.

Step 2: If $A_i = \phi$, stop; A_i is the current set of active nodes. Otherwise, continue.

Step 3: Select an element ℓ of A_i with maximal Z^* (suggested the breaking rule: maximal $\hat{Z}$).

Step 4: If $FV(\ell) = \phi$, go to step 9.
Otherwise, select an element j of $FV(\ell)$ with maximal internal rate of return IRR_j (suggested tie breaking rule: maximal C_{jo}).

Step 5: If $IRR_j \leqslant r(\ell)$, set $FV(\ell) = \phi$ and go to step 9.
Otherwise, continue.

Step 6: If $IRR_j \leqslant c\left(\sum_{\nu=1}^{n} C_{\nu o} x_\nu(\ell) + C_{jo}\right)$, go to step 8.
Otherwise, continue.

Step 7: If $\sum_{\nu=1}^{n} a_{\nu k} x_\nu(\ell) + a_{jk} \leqslant b_k$, for $k = 0,1,\ldots,M$, stop; A_i is the considered current set of active nodes, with ℓ and j as parameters.
Otherwise, continue.

Step 8: Subtract $\{j\}$ from $FV(\ell)$ and go to step 4.

Step 9: Subtract $\{\ell\}$ from A_i, add $\{\ell\}$ to C_i and go to step 2.

LEMMA 3.5: Let A_i, with ℓ and j as parameters, be the result of procedure P2. If TR_i is completely feasible and bounding, and $IRR_\mu \geqslant IRR_\nu$, for any μ such that $x_\mu(\ell) = 1$ and any $\nu \in FV(\ell)$, then:

i) A_i is a set of active nodes, and

ii) $A_i \neq \phi \implies \ell$ is not a concluding node.

Proof:

i) Because TR_i is completely feasible and bounding, R_i is well defined. Hence, at step 1, A_i is a set of active nodes. This part of the proof may be completed justifying the possible subsequent changes in A_i. These changes can only take place at step 9, which is performed either after step 4 or after step 5. In both cases, step 9 comes as a result of $FV(\ell)$ being empty, and this means that ℓ is a concluding node. Thus, at this stage $\{\ell\}$ has to be subtracted from A_i and added to C_i. Finally, steps 4-8 (a simple extension of steps 2-6 of procedure P1) guarantee that $FV(\ell)$ is appropriately handled along the procedure.

ii) Steps 4-7, together with the two assumptions of the lemma, guarantee that at least $\underline{x}(\ell)$ and $\underline{x}^+(\ell)$ are elements of S_ℓ, where $\underline{x}^+(\ell) = [x_1{}^+(\ell), x_2{}^+(\ell), \ldots, x_n{}^+(\ell)]$, $x_j{}^+(\ell) = 1$ and $x_\nu{}^+(\ell) = x_\nu(\ell)$, for any $\nu \neq j$. Therefore, ℓ cannot be a concluding node.

After determining A_0 (lemmas 3.3 and 3.4 and step 2 of P1 show that the assumptions of Lemma 3.5 hold for TR_0), with which U_0 is also obtained, the start (iteration 0) of the algorithm will lead either to an optimal solution to problem (3.5) or to the necessary information to proceed with the next iteration. In this case, a branching rule has to be available. The following is a procedure which will be seen to fulfil the requirements of a branching rule. Assumption ii) of Section 2.3 will also be seen to be satisfied (again, the procedure will be referred to any iteration i).

THE BRANCHING RULE (PROCEDURE P3).

Step 1: Define $N(t_1) = r+1$ and $N(t_2) = r+2$, where r is the total number of nodes of TR_{i-1}, $\mathbb{P}_j[\tilde{N}^{-1}(\ell)] = \{t_1, t_2\}$ and ℓ and j are the parameters of A_{i-1}.

Step 2: Set $SN_i = SN_{i-1} \cup \{r+1, r+2\}$ and

$SA_i = SA_{i-1} \cup \{(\ell, r+1), (\ell, r+2)\}$.

Step 3: Define $TR_i = \{SN_i, SA_i\}$

LEMMA 3.6: If TR_{i-1} is a directed tree, then TR_i is also a directed tree.

Proof: The two directed arcs by which SA_i differs from SA_{i-1} are incident out of one terminal node of TR_{i-1} and incident into the two nodes by which SN_i differs from SN_{i-1}. Therefore, if SN_{i-1} and SA_{i-1} satisfy the assumptions of Lemma 2.1 (see Section 2.2), so do SN_i and SA_i.

LEMMA 3.7: If TR_{i-1} is a directed tree, then TR_{i-1} is a sub-tree of TR_i.

Proof: Clearly, $SN_{i-1} \subset SN_i$, $SA_{i-1} \subset SA_i$, and no changes of association between arcs and nodes are involved.

LEMMA 3.8: Together with their associated concepts, if TR_{i-1} is completely feasible and consistently bounding, then TR_i is also completely feasible and consistently bounding.

Proof: P2 and Step 1 of P3 guarantee that nodes r+1 and r+2 are feasible nodes, because at least $\underline{x}^+(\ell) = \underline{x}(r+1) \epsilon S_{r+1}$ and $\underline{x}(\ell) = \underline{x}(r+2) \epsilon S_{r+2}$. All the other terminal nodes of TR_i are feasible terminal nodes of TR_{i-1}, and, since $\tilde{N}^{-1}(r+1) \cup \tilde{N}^{-1}(r+2) = \tilde{N}^{-1}(\ell)$, TR_i has to be completely feasible. Hence, given that $\underline{x}(r+1)$ and $\underline{x}(r+2)$ are auxiliary solutions of r+1 and r+2, respectively, and that, by assumption, an auxiliary solution is associated with all the other terminal nodes of TR_i, by Corollary 3.1,

TR_i is upper-bounding and consistently lower-bounding. Finally, TR_i is consistently bounding, because Z^*_{r+1}, $Z^*_{r+2} \leq Z^*_{\ell}$ (see proof of Lemma 3.4).

COROLLARY 3.2: If TR_{i-1} is completely feasible and consistently bounding, then the requirements of a branching rule (see Definition 2.20 in Section 2.2) are satisfied by procedure P3.

Proof: Because assumption ii) is clearly satisfied by procedure P3, this result follows from lemmas 3.7 and 3.8.

Having proved that TR_0 is completely feasible and consistently bounding, the general validity of P3 is implied by induction, under the consideration that assumption ii) of Section 2.3 is always satisfied after P2 has been performed. The algorithm can now be stated.

3.3.3) STATEMENT OF THE ALGORITHM.

In the following statement, the steps previous to the construction of the initial tree will be included as part of the start .

START (ITERATION 0)

1. Follow the steps of procedure P1.
2. If $\underline{x} = \underline{0}$ is optimal, stop.

   Otherwise, continue.
3. Follow the steps of procedure P2 for i = 0.
4. If $A_0 = \phi$, stop; $L_0 = \hat{Z}_2$ corresponds to the optimal solution to problem (3.5).

   Otherwise, continue.
5. If $L_0 = U_0$, stop; $L_0 = \hat{Z}_2$ corresponds to an optimal solution to problem (3.5).

   Otherwise, set i=1 and continue.

ITERATION i.

1. Follow the steps of procedure P3.
2. Follow the steps of procedure P2.
3. If either $A_i = \phi$ or $L_i = U_i$, stop; L_i corresponds to an optimal solution to problem (3.5).

   Otherwise, set $i \leftarrow i+1$ and start iteration i.

It should be noted that, when following procedure P2, L_i has to be available in order to determine R_i. In this connection, there is no need to perform a search of any kind. The auxiliary solutions associated with TR_i differ from those associated with TR_{i-1} only by $\underline{x}(r+1)$, since $\underline{x}(r+2) = \underline{x}(\ell)$. Therefore, L_i will simply be given by $\max\{L_{i-1}, \hat{Z}_{r+2}\}$. On the other hand, R_{i-1} is clearly a subset of R_i, because $L_{i-1} \leq L_i$. Hence, apart from r+1 and r+2, only non-rejected nodes of TN_{i-1} have to be inspected to determine R_i. C_{i-1}, on the other hand, can be used as input for C_i. Finally, it is important to bear in mind that, as usual, only information corresponding to current active nodes has to be stored for computational purposes.

3.4) A NUMERICAL EXAMPLE.

In Tables 3.1 and 3.2 the data corresponding to an example of 6 projects over 9 time periods are presented ($b_0 = 15$ and $b_1 = 5$). The summarized results of the complete procedure and the final directed tree can be found in Table 3.3 and Figure 3.1, respectively. As can be observed, projects P_1, P_4 and P_6 could be directly eliminated. However, for reasons which will become evident in subsequent chapters, they have been included in the calculations. The algorithm will be followed iteration by iteration.

START (IERATION 0)

Node 1 is created

$FV(1) = \{1,2,3,4,5,6\}$

$\underline{x}(1) = (0,0,0,0,0,0)$, $r(1) = 0.126$ and $\hat{Z}_1 = 0$

$Y(1) = \{1,2,3,4,5,6\}$ and $Z_1^* = 6.280$

Nodes 2 and 3 and arcs (1,2) and (1,3) are created (j=2)

$FV(2) = FV(3) = \{1,3,4,5,6\}$

$\underline{x}(2) = (0,1,0,0,0,0)$, $r(2) = 0.126$ and $\hat{Z}_2 = 3.593$

$Y(2) = \{1,2,3,4,5,6\}$ and $Z_2^* = 6.280$

$\underline{x}(3) = \underline{x}(1)$, $r(3) = r(1)$ and $\hat{Z}_3 = 0$

$Y(3) = \{1,3,4,5,6\}$ and $Z_3^* = 2.687 < \hat{Z}_2$ Node 3 REJECTED

$R_0 = \{3\}$, $C_0 = \phi$ and $A_0 = \{2\}$ ($\ell = 2$, $j = 5$)

$L_0 = 3.593$ and $U_0 = 6.280$

j	Project P_j	B_{jo} C_{jo}	B_{j1} C_{j1}	B_{j2} C_{j2}	B_{j3} C_{j3}	... ...	B_{j9} C_{j9}	IRR_j
1	P_1	0 5	2 5	3 1.2	3 1.2	... ...	3 1.2	0.130
2	P_2	0 2	1 2	2 0.5	2 0.5	... ...	2 0.5	0.369
3	P_3	0 4	1 2	2.3 1	2.3 1	... ...	2.3 1	0.160
4	P_4	0 10	4 9	7.5 4.1	7.5 4.1	... ...	7.5 4.1	0.130
5	P_5	0 4	1 3	2.5 0.8	2.5 0.8	... ...	2.5 0.8	0.188
6	P_6	0 6	2 3	3 1.4	3 1.4	... ...	3 1.4	0.127

TABLE 3.1 Cash Flows and Internal Rates of Return of the Projects.

Level of Investment	Cost of Capital $c(x)$
$0 \leq x \leq 4$	0.126
$4 < x \leq 8$	0.132
$8 < x \leq 12$	0.144
$x > 12$	0.150

TABLE 3.2 Levels of Investment and Cost of Capital

Iteration i	ℓ	j	A_i	Concluding Nodes	Rejected Nodes	$Z_1^* = 6.280$	
						L_i	U_i
0	2	5	{2}	–	3	3.593	6.280
1	4	3	{4}	–	5	4.823	5.413
2	–	–	ϕ	7	6	4.823	–

TABLE 3.3: Summarized Results of the Numerical Example.

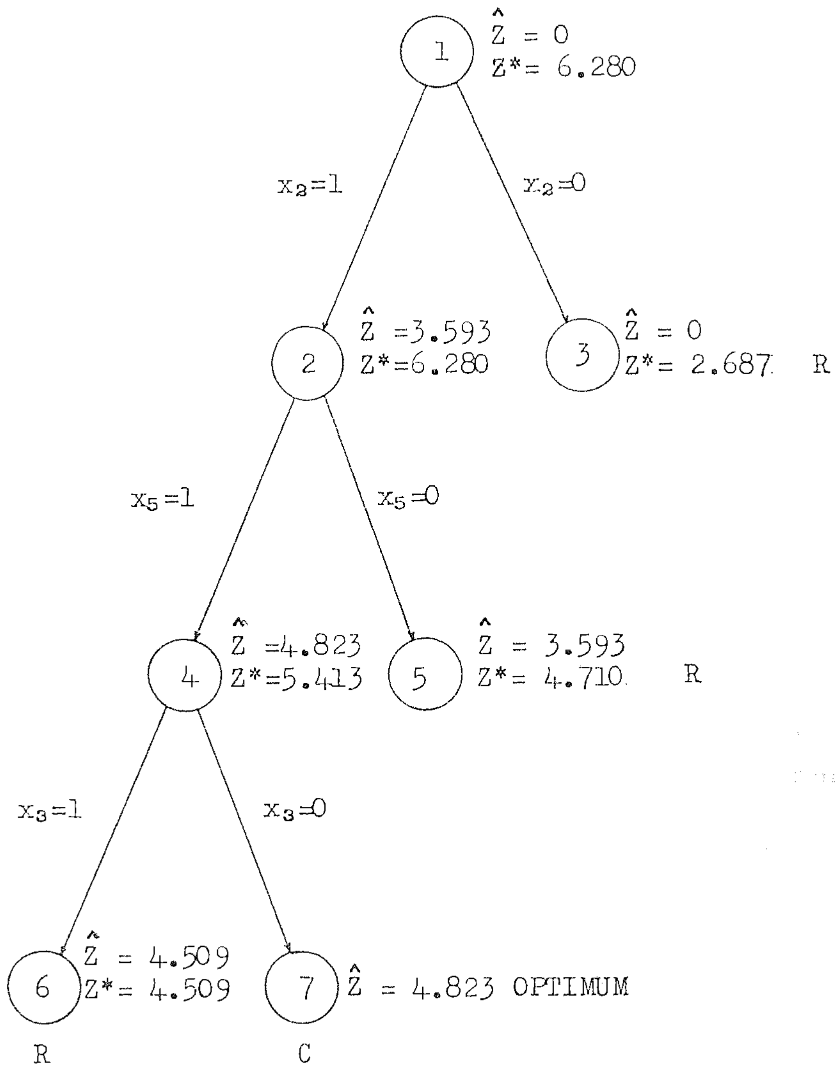

FIGURE 3.1: The Final Directed Tree.

ITERATION 1.

Nodes 4 and 5 and arcs (2,4) and (2,5) are created ($j=5$)

$FV(4) = FV(5) = \{1,3,4,6\}$

$\underline{x}(4) = (0,1,0,0,1,0), r(4) = 0.132$ and $\hat{Z}_4 = 4.823$

$Y(4) = \{2,3,5\}$ and $\hat{Z}_4 = 5.413$

$\underline{x}(5) = \underline{x}(2)$, $r(5) = r(2)$ and $\hat{Z}_5 = 3.593$

$Y(5) = \{1,2,3,4,6\}$ and $Z_5{}^* = 4.710 < \hat{Z}_4$ Node 3 is REJECTED

$R_1 = \{3,5\}$, $C_1 = \phi$ and $A_1 = \{4\}$ ($\ell=4$, $j=3$)

$L_1 = 4.823$ and $U_1 = 5.413$

ITERATION 2.

Nodes 6 and 7 and arcs (4,6) and (4,7) are created ($j=3$)

$FV(6) = FV(7) = \{1,4,6\}$

$\underline{x}(6) = (0,1,1,0,1,0)$, $r(6) = 0.144$ and $\hat{Z}_6 = 4.509$

$Y(6) = \{2,3,5\}$ and $Z_6{}^* = 4.509 < L_1$ Node 6 is REJECTED

$\underline{x}(7) = \underline{x}(4)$, $r(7) = r(4)$ and $\hat{Z}_7 = 4.823$

$Y(7) = \{2,5\}$ and $Z_7{}^* = 4.823$

$R_2 = \{3,5,6\}$, $C_2 = \{7\}$ ($FV(7)$ changes from $\{1,4,6\}$ to ϕ)and

$A_2 = \phi$

$L_2 = 4.823$

After iteration 2,it is found that the current set of active nodes is empty. Therefore, $\underline{x}^* = (0,1,0,0,1,0)$, associated with $L_2 = 4.823$, is optimal. This example clearly shows that, when the cost of capital is dependent on the level of investment, it is not sufficient to know that the NPV of a project is positive (given a level of investment) in order to be in a position to accept it, even without active expenditure limits. The auxiliary solution $\underline{x}(6) = (0,1,1,0,1,0)$ is obviously feasible, but not optimal.

3.5) AN UNCONSTRAINED VERSION OF THE PROBLEM.

Problem (3.5) was formulated in Section 3.2 under the assumption that both the investment and the operating overall requirements could not surpass their corresponding pre-determined limits $b_0, b_1, \ldots, b_M$. This can be interpreted either as a self-imposed financial policy, or as an external restriction associated with the availability of funds for investment. Within a deterministic context, however, the possibility of obtaining as much as necessary to invest in profitable projects cannot be disregarded, if the capital supply is sufficiently large. In this case, the expenditure limits at each time period, rather than being previously fixed, are determined by the maximal requirements corresponding to a selection of projects with positive NPV. The limits are, therefore, a function of the universe of investment proposals and of the cost of capital under consideration, and they are uniquely established once these two concepts are given. Nonetheless, there is no need to exhibit them explicitly in the model, because of the assumption that, in any case, funds to meet the expenditure requirements are available, only if projects with positive NPV are selected. Hence, the problem may be formulated as:

$$\left.\begin{aligned} &\text{Max} \quad Z = \sum_{j=1}^{n} NPV_j(t)x_j && (3.6) \\ &\text{s.t. } NPV_j(t) > 0, \text{ if } x_j = 1, j = 1,2,\ldots,n && (3.7) \\ &x_j = 0 \text{ or } 1, \; j = 1,2,\ldots,n && (3.8) \end{aligned}\right\} (3.9)$$

Problem (3.9) differs from problem (3.5) only by constraint (3.2), and, as a result of (3.7), strictly speaking, it is a constrained zero-one programming problem. Let the problem defined by (3.6) and (3.8) be considered; namely:

$$\left.\begin{aligned} &\text{Max} \quad Z = \sum_{j=1}^{n} NPV_j(t)x_j \\ &\quad \text{s.t. } x_j = 0 \text{ or } 1, \; j = 1,2,\ldots,n \end{aligned}\right\} \qquad (3.10)$$

Problem (3.10) is the zero-one unconstrained version of problem (3.9), and, as was mentioned before, although the feasbility region of the first can properly contain that of the second, their optimal objective values have to be the same, because they are maximization problems. Any optimal solution to problem (3.9) is an optimal solution to problem (3.10), and, for most practical purposes, vice versa. In short, the difference between the two problems is to a large extent negligible, and, since constraint (3.7) will be dealt with anyway, problem (3.9) will be referred to as the (zero-one) unconstrained version of problem (3.5). To handle this problem in the framework of the solution method developed for problem (3.5), it is sufficient to think of the b_k's as very large numbers ($b_k = \infty$, $k = 0,1,\ldots,M$), as a consequence of which S reduces to:

$$S_U = \{\underline{x} \in E^n / (3.7) \text{ and } (3.8) \text{ are satisfied}\}$$

Similarly, the steps of the precedure associated with constraint (3.2) are no longer relevant and may therefore be eliminated, as shown below:

PROCEDURE $P1_u$

Step 1: Define N(T) = 1 and obtain the associated concepts of node 1.

Step 2: Select an element j of FV(1) with maximal internal rate of return IRR_j (suggested tie breaking rule: maximal investment).

Step 3: If $IRR_j \leq r(1)$, stop; $\underline{x}^* = \underline{0}$ is the optimal solution to problem (3.5).

Otherwise, continue.

Step 4: If $IRR_j > c(C_{jo})$, define $N(t_1) = 2$ and $N(t_2) = 3$, where $\mathbb{P}_j(T) = \{t_1, t_2\}$. The procedure is complete: $SN_o = \{1,2,3\}$, $SA_o = \{(1,2),(1,3)\}$ and $TR_o = \{SN_o, SA_o\}$.

Otherwise, continue.

Step 5: Subtract $\{j\}$ from $FV(1)$

If $FV(1) \neq \phi$, go to step 2.

Otherwise, stop; $\underline{x}^* = \underline{0}$ is the optimal solution to problem (3.5).

PROCEDURE $P2_u$

Step 1: Set $A_i = TN_i - C_i - R_i$

Step 2: If $A_i = \phi$, stop; A_i is the current set of active nodes.

Otherwise, continue.

Step 3: Select an element ℓ of A_i with maximal Z^*.

Step 4: If $FV(\ell) = \phi$, go to step 8.

Otherwise, select an element j of $FV(\ell)$ with maximal internal rate of return IRR_j.

Step 5: If $IRR_j \leq r(\ell)$, set $FV(\ell) = \phi$ and go to step 8.

Step 6: If $IRR_j > \left(\sum_{\nu=1}^{n} C_{\nu o} x_\nu(\ell) + C_{jo} \right)$, A_i is the considered current set of active nodes, with ℓ and j as parameters.

Otherwise, continue.

Step 7: Subtract $\{j\}$ from $FV(\ell)$ and go to step 4

Step 8: Subtract $\{\ell\}$ from A_i, add $\{\ell\}$ to C_i and go to step 2.

Problem (3.9) can then be solved by means of the algorithm presented in Section 3.3.3.

3.6) CHANGING THE EXPENDITURE LIMITS.

In this section some postoptimality aspects associated with problem (3.5) and the change of the expenditure limits $b_1, b_0, \ldots, b_M$ will be considered. Problem (3.5) will be referred to as the CONSTRAINED PROBLEM (P_c), and problem (3.9), as indicated in the preceding section, as the UNCONSTRAINED PROBLEM (P_u).

Let $\underline{x}^*$ be an optimal solution to P_c, let $\underline{b} = (b_0, b_1,,,, b_M)$, $\hat{\underline{b}} = \underline{b} + \underline{\Delta b} = \underline{b} + (\Delta b_0, \Delta b_1, \ldots, \Delta b_M)$, and let $\hat{P}_c$ denote the problem defined by replacing $\underline{b}$ by $\hat{\underline{b}}$ in P_c.

LEMMA 3.9: If $\underline{x}^*$ is an optimal solution to P_u and a feasible solution to $\hat{P}_c$, then $\underline{x}^*$ is an optimal solution to $\hat{P}_c$.

Proof: Let $\hat{S}$ be the feasbility region of $\hat{P}_c$. Obviously, $\hat{S} \subset S_u$. Hence, $f(\underline{x}^*)$ is maximal in $\hat{S}$.

COROLLARY 3.3: If $\underline{x}^*$ is an optimal solution to P_u and $\sum_{j=1}^{n} a_{jk} x_j^* \leq b_k + \Delta b_k$ $(k = 0, 1, \ldots, M)$, then $\underline{x}^*$ is an optimal solution to $\hat{P}_c$.

Proof: If $\sum_{j=1}^{n} a_{jk} x_j^* \leq b_k + \Delta b_k$ $(k = 0, 1, \ldots, M)$, then $\underline{x}^*$ is a feasible solution to $\hat{P}_c$. Thus, by Lemma 3.9, the required result follows.

These straightforward results indicate that, if $\underline{x}^*$ is an optimal solution to the unconstrained problem, then the expenditure limits can arbitrarily be increased without altering the optimality of $\underline{x}^*$. Alternatively, the same assertion can be made if negative changes Δb_k's are introduced, but only when $\sum_{j=1}^{n} a_{jk} x_j^* < b_k$ and $-\Delta b_k \leq b_k - \sum_{j=1}^{n} a_{jk} x_j^*$.

LEMMA 3.10: A sufficient condition for $\underline{x}^*$ to be an optimal solution to P_u is encountered, if, along the solution process of P_c, step 5 of P1 and Step 7 of P2 are never followed by Step 6 of P1 and Step 3 of P2, respectively.

Proof: Indeed, in this case $P1 \equiv P1_u$ and $P2 \equiv P2_u$. Therefore, $\underline{x}^*$ is an optimal solution to P_u.

To use this result, it is only necessary to record whether or not the sequences of steps were followed. For example, in the problem presented in Section 3.4, $\underline{x}^* = (0,1,0,0,1,0)$ is an optimal solution to both the constrained and the unconstrained problem. Accordingly, so long as $b_0 \geq 6$ and $b_1 \geq 3$, it can be guaranteeed that $\underline{x}^*$ is optimal.

LEMMA 3.11: If $\underline{\Delta b} \leq 0$ and $\underline{x}^* \in \hat{S}$, then $\underline{x}^*$ is an optimal solution to $\hat{P}_c$.

Proof: In this case $\hat{S} \subset S$. Thus, since $\underline{x}^* \in \hat{S}$, $f(\underline{x}^*)$ is maximal in $\hat{S}$.

COROLLARY 3.4: If $\underline{\Delta b} \leq \underline{0}$ and $\sum_{j=1}^{n} a_{jk} x_j^* \leq b_k + \Delta b_k$ $(k = 0,1,\ldots,M)$, then $\underline{x}^*$ is an optimal solution to $\hat{P}_c$.

Proof: Again, since $\sum_{j=1}^{n} a_{jk} x_j^* \leq b_k + \Delta b_k$ $(k = 0,1,\ldots,M)$, $\underline{x}^*$ is a feasible solution to $\hat{P}_c$. Therefore, by Lemma 3.11, the required result follows.

In other words, $\underline{x}^*$ being an optimal solution to P_u or otherwise, if $\sum_{j=1}^{n} a_{jk} x_j^* < b_k$, then b_k can be decreased down to $\sum_{j=1}^{n} a_{jk} x_j^*$ $(k = 0,1,\ldots,M)$, without having to recompute $\underline{x}^*$.

LEMMA 3.12: If $\underline{x}^*$ is not an optimal solution to P_u and $S \subset \hat{S}$, then $\underline{x}^*$ is not necessarily an optimal solution to $\hat{P}_c$.

Proof: If, for example, $\hat{S} = S_u$, then $\underline{x}^*$ is not an optimal solution to $\hat{P}_c$.

COROLLARY 3.5: If $\underline{\Delta b} \geq \underline{0}$ and $\underline{x}^*$ is not an optimal solution to P_u, then $\underline{x}^*$ is a feasible, but not necessarily an optimal solution to $\hat{P}_c$.

Proof: If $\underline{\Delta b} \geq \underline{0}$, then $S \subset \hat{S}$, from which it follows that $\underline{x}^*$ is a feasible solution to $\hat{P}_c$. However, by Lemma 3.12, it is not necessarily an optimal solution to this problem.

These results are relevant for the cases in which increments for the expenditure limits are considered, not having met the condition given by Lemma 3.10. Of course, this does not mean that $\underline{x}^*$ is not an optimal solution to the unconstrained problem, but the contrary is not known either. In terms of the final directed tree from which $\underline{x}^*$ was obtained, this means that the search was not necessarily complete, but the calculations prior to the step at which the condition of Lemma 3.10 was violated can obviously be used to obtain the new optimal solution or to check that $\underline{x}^*$ is optimal. For example, if the problem of Section 3.4 had been solved for $b_0 = 8$ and $b_1 = 3$ (see Table 3.4 and Figure 3.2), the condition would have been violated at Step 8 of P2 when performing iteration 1, and $\underline{x}^* = (0,1,0,0,1,0)$. If then the increments $\Delta b_0 = 2$ and $\Delta b_1 = 2$ had been considered, it would only have been necessary to repeat Step 7 at that point for $b_0 = 10$ and $b_1 = 5$, and go on with the procedure. After one iteration it would have been found that $\underline{x}^*$ is an optimal solution to both $\hat{P}_c$ and P_u.

LEMMA 3.13: If $\underline{x}^* \epsilon \hat{S}$, then $\underline{x}^*$ is not an optimal solution to $\hat{P}_c$.

Proof: This trivial result is a direct consequence of Definition 2.1 (see Section 2.2).

COROLLARY 3.6: If $\underline{\Delta b} \leq \underline{0}$ and $\sum_{j=1}^{n} a_{jk}x_j^* > b_k + \Delta b_k$ for some $k = 0,1,\ldots,M$, then $\underline{x}^*$ is not an optimal solution to $\hat{P}_c$.

Proof: Since $\sum_{k=1}^{n} a_{jk}x_j^* > b_k + \Delta b_k$ for some $k = 0,1,\ldots,M$, $\underline{x}^* \notin \hat{S}$. Hence, it cannot be an optimal solution to $\hat{P}_c$.

As illustrated in the preceding example, the work pior to the violation of the condition of Lemma 3.10 could also be used in this case, if no expenditure surpassing $b_k + \Delta b_k$ $(k = 0,1,\ldots,M)$ had been considered at the time. Otherwise, it would be necessary to go further back, until this last requirement could be met. If in the problem of Section 3.4 b_0 and b_1 were to be decreased to 6 and 2, respectively,

Iteration i	ℓ	j	A_i	Concluding Nodes	Rejected Nodes	$Z_1^*=6.280$	
						L_i	U_i
0	2	5	{2}	–	3	3.593	6.280
1	–	–	ϕ	4	5	4.823	–

TABLE 3.4 Summarized Results of the Numerical Example (b_0 = 8 and b_1 = 3).

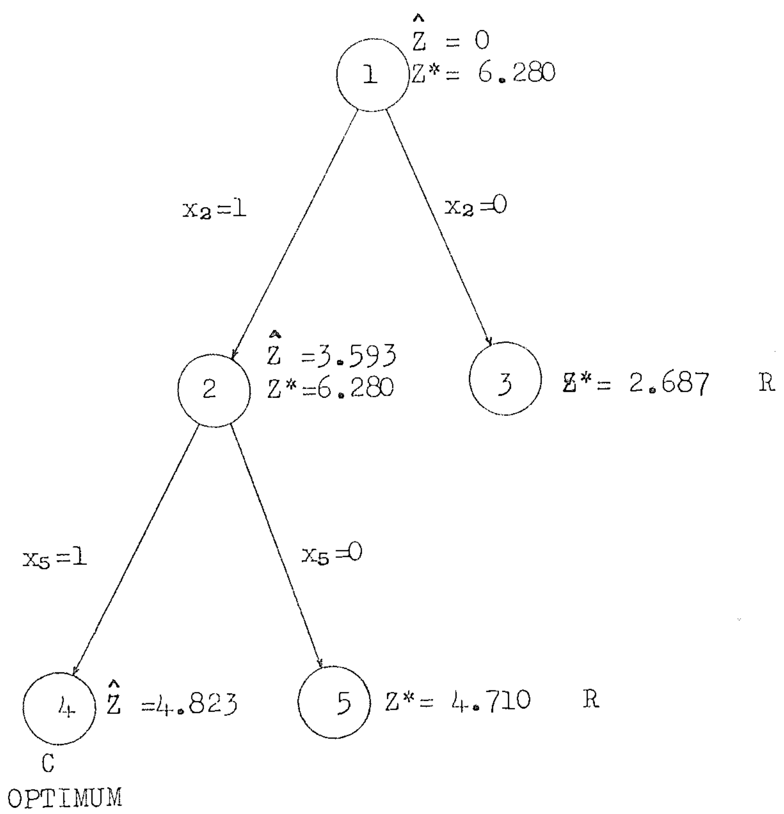

FIGURE 3.2 The Final Directed Tree (b_0 = 8 and b_1 = 3)

(note that condition of Lemma 3.10 was not violated in this example), then iteration 0, as opposed to iteration 1 in which $a_{22} + a_{52} = 3 < 2$, could be used to obtain the new optimal solution (see Table 3.5 and Figure 3.3).

3.5) SUMMARY AND FINAL REMARKS.

In this chapter a class of capital investment problems involving economically dependent projects was considered, under the assumption that the cost of capital, rather than being constant, is a positive non-decreasing function of the level of investment. This approach leads to a zero-one non-linear programming problem, for which a branch and bound solution method was fully developed, on the basis of the scheme presented in Section 2.3. Within this context, the solution to an unconstrained version of the problem was discussed, as well as some post-optimality aspects connected with changes in the expenditure limits. Illustrative numerical examples were also included.

Two final relevant remarks are the following:

1. It was assumed that the cost of capital is a function of the overall investment requirement only, and, accordingly, that one, and only one, investment requirement is associated with each project. However, by means of what was referred to as operating requirements, subsequent capital net outflows were also considered. For intuitive reasons linked with the way the cost of capital was defined, they were assumed to be not large in terms of the corresponding investment requirements, but with no implications at all, as far as the solution method is concerned. Therefore, if more than one investment requirement were involved with the projects, or, equivalently, if the operating requirements were large enough to have some effect in the

Iteration i	ℓ	j	A_i	Concluding Nodes	Rejected Nodes	$Z_1^* = 6.280$	
						L_i	U_i
0	2	3	{2}	-	3	3.593	6.280
1	-	-	ϕ	4	5	4.022	-

TABLE 3.5 Summarized Results of the Numerical Example (b_0 = 6 and b_1 = 2).

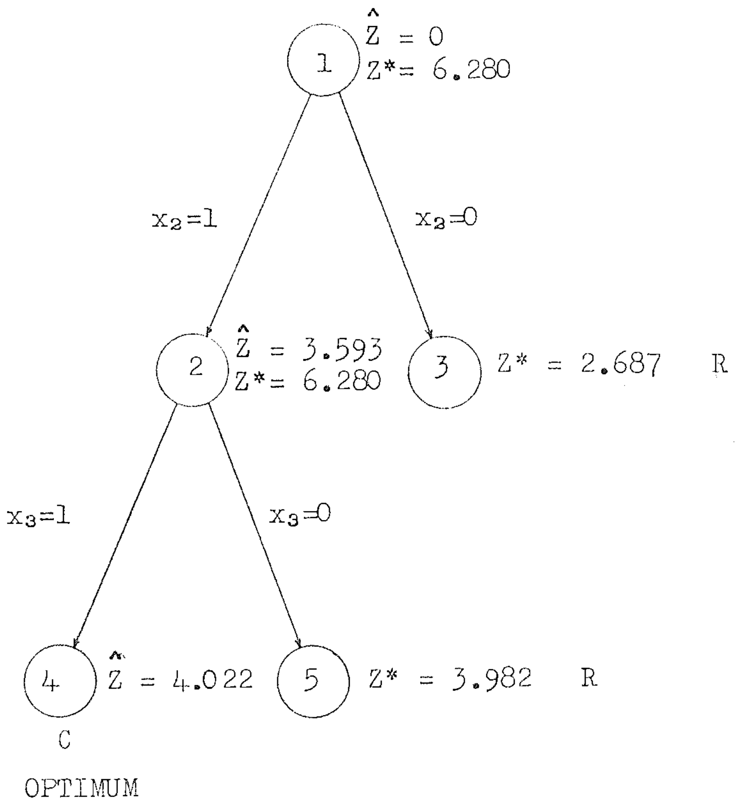

FIGURE 3.3 The Final Directed Tree (b_0=6 and b_1=2).

1. contd.

cost of capital, the solution method would still be applicable, so long as this cost were a positive non-decreasing function of each of its variables (either operating or investment requirements). In this case, steps 4 and 5 of P1 and steps 5 and 6 of P2 would have to be extended according to the number of variables of c.

2. The IRR criterion plays an important role both in the search procedure (reducing the sets of free variables) and in the calculation of the upper bounds. In both cases, this role is likely to become active (or more active), depending on the number of the IRR_j's lying near or within the range of c. Disregarding those which are under this range (the corresponding projects can directly be eliminated), the higher this number, the more active will the role be likely to become. For this reason, although, again, with no theoretical implications, it was assumed that the IRR_j's are relatively small. In the next chapter an alternative to deal with large IRR_j's will be discussed.

APPENDIX CH III.

THE INTERNAL RATE OF RETURN.

AIII.1) INTRODUCTION.

In this section the definition and some properties of the internal rate of return (IRR), an alternative appraisal measure to the NPV, will be presented. The properties under consideration are basically those connected with the way the measure was used in this chapter, and, therefore, they are by no means comprehensive. More complete discussions on the subject may be found, for example, in the works of Solomon[8], Mao[9] and Bierman and Smidt[10].

AIII.2) DEFINITION AND PROPERTIES.

Let P be a capital investment project defined as in Section 1.3 ($B_o = 0$).

DEFINITION AIII.1: If a unique non-negative constant IRR_p exists, such that:

$$\sum_{k=o}^{m} \frac{B_k - C_k}{(1+IRR_p)^k} = 0 \ , \qquad \text{(AIII.1)}$$

then IRR_p is said to be the INTERNAL RATE OF RETURN of project P.

In general, more than one non-negative constant satisfying equation (AIII.1) may exist, in which case P would normally be referred to as a project with multiple internal rates of return. However, as shown in the next lemma, this case is not relevant for the purposes of this work.

LEMMA AIII.1: If a natural number M ($< m$) exists, such that:

i) $B_k < C_k$, if $k \leq M$,

ii) $B_k > C_k$, if $k > M$, and

iii) $\sum_{k=o}^{M} (C_k - B_k) < \sum_{k=M+1}^{m} (B_k - C_k)$

then the IRR of project P exists, and $g(y) = \sum_{k=o}^{m} \frac{B_k - C_k}{(1+y)^k}$ is decreasing for non-negative values of y.

Proof: $g(0) = \sum_{k=o}^{m} (B_k - C_k)$. Therefore, by assumption iii),

$g(0) > 0$. On the other hand, $\lim_{y \to \infty} g(y) = - C_o < 0$.

Hence, since g is continuous for non-negative values of y, it has at least one positive real root. In general, to find the roots of g, the solutions to the following equation would have to be obtained.

$$\sum_{k=o}^{m} \frac{B_k - C_k}{(1+y)^k} = 0 \qquad \text{or}$$

$$d_o x^m + d_1 x^{m-1} + \ldots + d_m = 0, \qquad \text{(AIII.2)}$$

where $d_k = B_k - C_k$ and $x = 1+y$.

Consequently, taking into account assumptions i) and ii) and the Law of Signs of Descartes, equation (AIII.2) has only one positive real solution. In terms of g, this means that it only has one positive real root (the IRR of project P), and that $g(y) > 0$ for any $y \epsilon [0, IRR_p)$. Let the first derivative of g be considered in this interval.

$$g'(y) = - \sum_{k=o}^{m} k \frac{B_k - C_k}{(1+y)^{k+1}}$$

$$= - \frac{1}{1+y} \sum_{k=1}^{m} k \frac{B_k - C_k}{(1+y)^k}$$

$$= - \frac{1}{1+y} \sum_{i=1}^{m} \sum_{k=i}^{m} \frac{B_k - C_k}{(1+y)^k}$$

It is known that $g(y) > 0$, if $y \epsilon [0, IRR_p)$. Hence, as a consequence of assumptions i) and ii), in this interval $\sum_{k=i}^{m} \frac{B_k - C_k}{(1+y)^k} > 0$ for any $i \epsilon \{1, 2, \ldots, m\}$, and thus $g'(y) < 0$. Therefore, g is decreasing in $[0, IRR_p)$. The same reasoning leads to the conclusion that g is decreasing for non-negative values of y, if the y-axis is translated downwards arbitrarily above $-C_0$.

Having shown that, according to Definition AIII.1, for each project of problem (3.5) (see Section 3.2) the IRR exists, the well known connection between the NPV and the IRR is next presented. Again, this result will be referred to the class of projects under consideration.

COROLLARY AIII.1: Let NPV_p be the NPV of project P (see Definition 1.7 in Section 1.3). Under the assumptions of Lemma AIII.1, $NPV_p > 0$ if, and only if, $IRR_p > c$, where c is the cost of capital.

Proof: As a consequence of Lemma AIII.1,

$$NPV_p = g(c) > g(IRR_p) = 0, \text{ if, and only if, } IRR_p > c.$$

AIII.3) FINAL COMMENTS.

Several arguments against the use of the IRR as an appraisal measure have been established, starting with the possible analytical difficulties of its definition. Not only more than one, but also no non-negative (or even real) constant satisfying equation (AIII.1) could be found to exist. Otherwise, the IRR can be used directly in the sense of Corollary AIII.1 to know whether or not the NPV is positive. This implies that in perfect capital markets (see Section 1.3) the IRR can be used instead of the NPV to accept or reject independent projects. However, either because of dependence constraints or because of imperfect capital markets conditions, if the problem is to choose amongst competitive projects, the IRR is no longer consistent with the NPV. Its use as a pay-off would certainly lead to a selection of projects with positive NPV, but not necessarily to an optimal selection.

REFERENCES.

1 H.M.WEINGARTENER (1963) Mathematical Programming and the Analysis of Capital Budgeting Problems. Prentice-Hall, Englewood Cliffs, New Jersey.

2 J.H.LORIE and L.J.SAVAGE (1955) Three problems in rationing capital.J.Bus.28, 229-239.

3 A.H.LAND and A.DOIG (1960) An automatic method for solving discrete programming problems. Econometrica 28, 497-520.

4 J.B.DANTZIG (1957) Discrete variable extremum problems. Ops.Res.5, 266-277.

5 P.KOLESAR (1967) A branch and bound algorithm for the knapsack problem. Mgmt.Sci.13, 723-735.

6 F.OCHOA-ROSSO (1968) Applications of Discrete Optimzation Techniques to Capital Investment and Network Synthesis Problems.Sc.D.Dissertation, Department of Civil Engineering, Massachusetts Institute of Technology, Boston, Massachusetts.

7 H.GREENBERG and R.L.HEGERICH (1970) A branch search algorithm for the knapsack problem. Mgmt.Sci.16, 327-332.

8 E.SOLOMON (1963) The Theory of Financial Management. Columbia University Press, New York.

9 J.C.T.MAO (1969) Quantitative Analysis of Financial Decisions. Macmillan, Toronto.

10 H.BIERMAN, Jr. and S SMIDT (1975) The Capital Budgeting Decision, Macmillan, New York.

CHAPTER IV

ON SOME BOUNDING ASPECTS OF THE ALGORITHM.

4.1) INTRODUCTION.

The calculation of improved upper and lower bounds in connection with the algorithm presented in Section 3.3.3 will be considered in this chapter. The proposed procedures will be shown to be applicable in general for both lower and upper bounds, but that associated with the latter will be seen to be particularly useful for the cases of large internal rates of return and samll expenditure limits.

4.2) IMPROVED UPPER BOUNDS.

4.2.1) PRELIMINARY REMARKS.

The role of upper bounds in the framework of the proposed solution method is clearly of great importance. Essentially, the usefulness of these bounds could be summarized in that, together with the lower bounds, (1) they provide a means to estimate how far the optimal solution might be, at any stage of the algorithm; and (2) their defining elements are the basis to reject feasible terminal nodes, and hence to reduce the number of steps in the search procedure. It is obvious too that the effectiveness of these two aspects are in inverse relation to the size of the upper bounds, and, in any case, upper bounds can only be interesting, if they are reasonably small. Now, as indicated in Section 2.2 (see Lemma 2.9), upper bounds in the proposed solution method correspond to optimal objective values of the upper bounding functions of active nodes. For each node ℓ, these functions were defined in Section 3.3.2 (see Lemma 3.1) with Y_ℓ as domain, which means that constraint (3.2) of problem (3.5) (see Section 3.2) was not explicitly considered (simply because $Y_\ell - S_\ell$ is not necessarily empty). It was assumed, however, that the internal rates of return of the projects were relatively small in terms of the cost of capital, so that the number of elements in $Y(\ell)$ (see Lemma 3.2) could also be expected to be relatively small, in the sense of representing an element of Y_ℓ near or within the region defined by constraint (3.2). In this sense, this constraint was implicitly considered. Relaxing this assumption, the theoretical support of the algorithm would not be affected, but then, especially if small expenditure limits were involved, many relatively large internal rates of return would be likely to define relatively large numbers of elements in the $Y(\ell)$'s. This would obviously lead to less effective upper bounds, and, therefore, to a less efficient solution method. To illustrate the impact of the relative size of the internal rates of return, the problem of Section 3.4 will be considered under different ranges for the cost of capital.

4.2.2) SOME NUMERICAL COMPARISONS.

THE ORIGINAL EXAMPLE: In this case IRR_1, IRR_4 and IRR_6 are within the range $R_c = [0.126, 0.150]$ of the cost of capital; IRR_3, IRR_5 and IRR_2 are above R_c by 1%, 3.8% and 21.9%, respectively. The procedure ended after 3 iterations (taking the start into account), and only 4 terminal nodes were considered. Additionally, the current set of active nodes never had more than one element along the procedure.

FIRST CHANGE: The original example solved with $c(x)$ replaced by $c_1(x)$ given in Table 4.1 (see Table 4.2 and Figure 4.1). In this case the internal rates of return are comparatively larger: IRR_3, IRR_5 and IRR_2 are now above $R_{c_1} = [0.166, 0.141]$ by 1.9%, 4.7% and 22.8%, respectively, but IRR_1, IRR_4 and IRR_c are still well within R_{c_1}. Nevertheless, both the number of iterations and the number of terminal nodes, increased by 1, as well as the number of elements in A_i.

Level of Investment x	Cost of Capital	
	$c_1(x)$	$c_2(x)$
$0 \leq x \leq 4$	0.116	0.106
$4 < x \leq 8$	0.123	0.113
$8 < x \leq 12$	0.135	0.125
$x > 12$	0.141	0.131

TABLE 4.1 Levels of Investment and Costs of Capital.

SECOND CHANGE: The preceding example solved with $c_1(x)$ replaced by $c_2(x)$ given in Table 4.1 (see Table 4.3 and Figure 4.2). Although the difference between $c_2(x)$ and $c_1(x)$ is practically the same as that between $c_1(x)$ and $c(x)$, in this case IRR_1, IRR_4 and IRR_6, again within $R_{c_2} = [0.106, 0.131]$, are very close to the upper limit of R_{c_2}. As a consequence, in reference to the first change, the number of iterations increased by 5 to 9, the number of terminal nodes by 5 to 10, and the maximal number of current active nodes by 3 to 5.

Finally, it is interesting to observe that, after having found the optimal solution, the number of iterations to complete the procedure was 1 in the first case, 2 in the second case, and 6 in the third case, even though the same sequencing was followed to obtain the optimal solutions. It can therefore be inferred that the extra computations and requirements were mainly due to comparatively large increments in the upper bounds.

4.2.3) THE IMPROVED UPPER BOUNDS.

Relaxing the assumption of comparatively small internal rates of return, an alternative definition for the upper bounding functions leading to improved upper bounds will be proposed in this section. The corresponding domains will now be allowed to

Iteration i	ℓ	j	A_i	Concluding Nodes	Rejected Nodes	$Z_1^* = 8.362$	
						L_i	U_i
0	2	5	{2,3}	-	-	3.875	8.362
1	4	3	{4,5}	-	3	5.337	6.890
2	5	3	{5,7}	-	6	5.337	6.480
3	-	-	ϕ	7	8,9	5.337	-

TABLE 4.2 Summarized Results of the Numerical Example with $c_1(x)$.

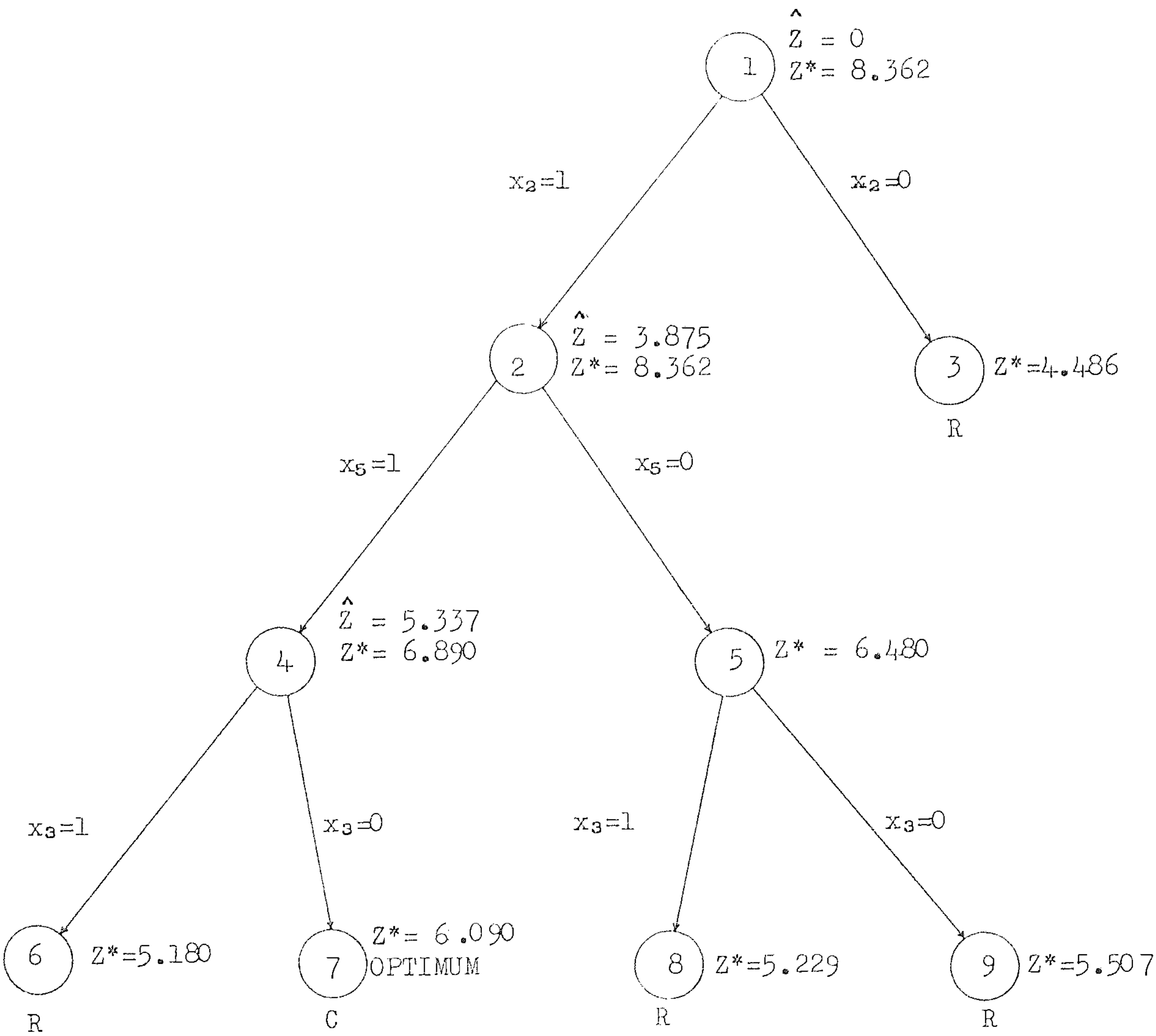

FIGURE 4.1 The Final Directed Tree for $c_1(x)$.

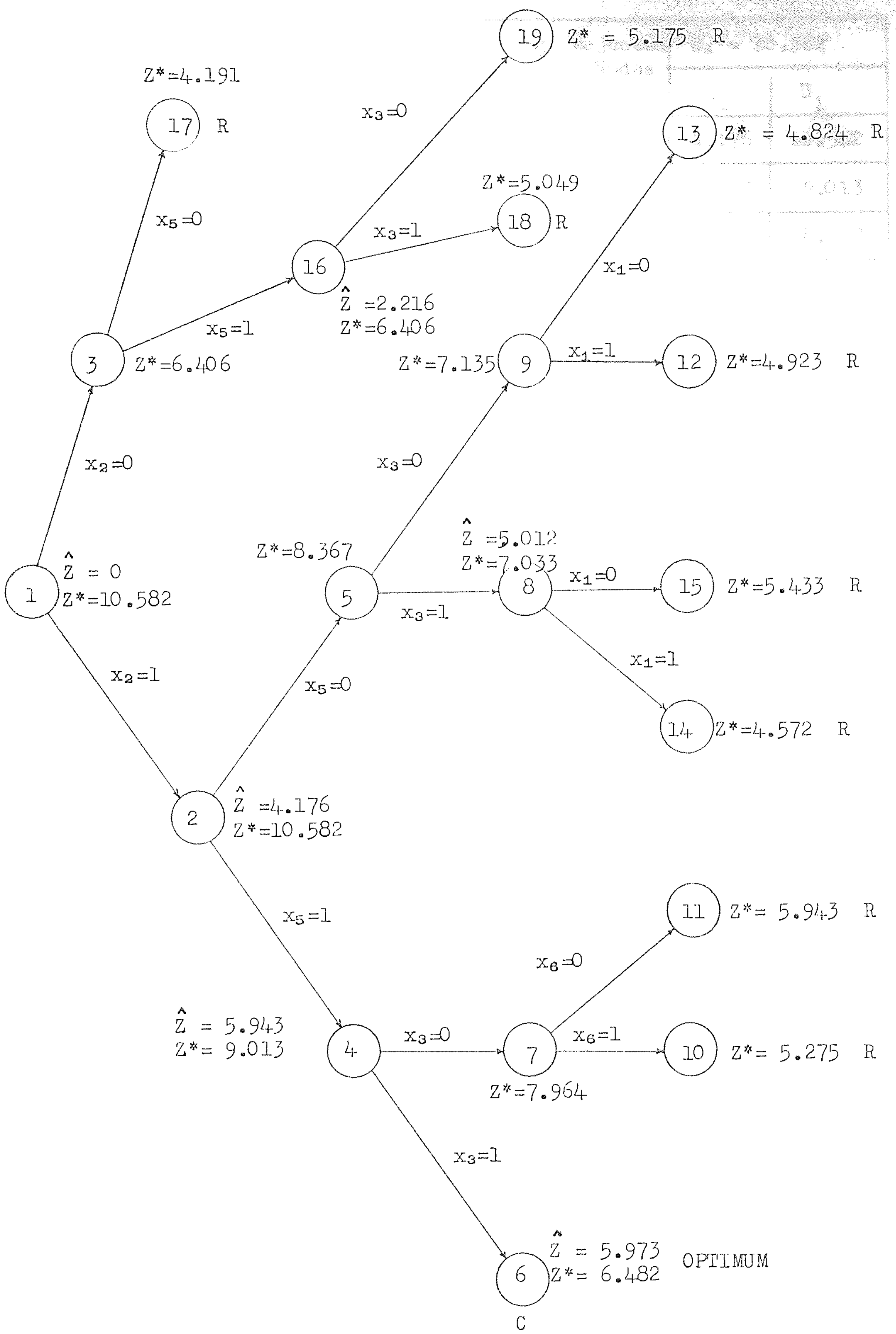

FIGURE 4.2 The Final Directed Tree for $c_2(x)$.

Iteration i	ℓ	j	A_i	Concluding Nodes	Rejected Nodes	$Z_1^* = 10.582$	
						L_i	U_i
0	2	5	{2,3}	–	–	4.176	10.582
1	4	3	{3,4,5}	–	–	5.943	9.013
2	5	3	{3,5,6,7}	–	–	5.973	8.367
3	7	6	{3,6,7,8,9}	–	–	5.973	7.964
4	9	1	{3,6,8,9}	–	10,11	5.973	7.135
5	8	1	{3,6,8}	–	12,13	5.973	7.033
6	3	5	{3}	6	14,15	5.973	6.406
7	16	3	{16}	–	17	5.973	6.406
8	–	–	ϕ	–	18,19	5.973	–

TABLE 4.3 Summarized Results of the Numerical Example with $c_2(x)$.

be infinite, and, in consequence, the set Y will have to be redefined in order to fulfil the requirements of Section 2.2.

Let $Y = \{\underline{x} \in E^n / 0 \leq x_j \leq 1,\ j = 1,2,\ldots,n\}$. Obviously, T, as defined in Section 3.3.1, is a subset of Y. Hence, with the only exception of the definition of Y, the complete development of Section 3.3 remains unaffected with this change.

LEMMA 4.1: Let $W = \{\underline{x} \in E^n / (3.2) \text{ is satisfied}\}$ (see section 3.2), and let $\Omega_\ell : Y_\ell' \to \mathbb{R}$ be such that

$$\underline{x} \in Y_\ell' \Longrightarrow \Omega_\ell(x) = \sum_{j=1}^{n} x_j \sum_{k=0}^{m} \frac{B_{jk} - C_{jk}}{[1+r(\ell)]^k}, \text{ where}$$

$Y_\ell' = W \cap V_\ell$, $V_\ell = \{\underline{x} \in E^n / \text{either } j \in FV(\ell) \text{ and } 0 \leq x_j \leq 1,$ or $x_j = x_j(\ell)\}$, and ℓ is any node. If $Y_\ell' \neq \phi$, then $\Omega_\ell(\underline{x}) \geq f(\underline{x})$, for all $\underline{x} \in Y_\ell'$.

Proof: See proof of Lemma 3.1 in Section 3.3.2.

COROLLARY 4.1: If ℓ is a feasible node, Ω_ℓ is an upper bounding function of ℓ.

Proof: By definition, any $\underline{x} \epsilon S$ has to satisfy constraints (3.2) and (3.4). Consequently, if $\underline{x} \epsilon S_\ell$, then it also has to belong both to W and to V_ℓ. This means that $S_\ell \subset Y_\ell$, and, evidently, $Y_\ell \subset Y$. On the other hand, Y_ℓ is bounded and Ω_ℓ is linear. Thus, $\Omega_\ell{}^*$ has to exist. Finally, by Lemma 4.1, it can be concluded that Ω_ℓ is an upper bounding function of any feasible node ℓ.

LEMMA 4.2: If ℓ is any feasible node, then $\Omega_\ell{}^* \leq Z_\ell{}^*$.

Proof: Let $\underline{x}' = (x_1', x_2', \ldots, x_n')$ and $\underline{x}'' = (x_1'', x_2'', \ldots, x_n'')$ be any elements of Y for which $\Omega_\ell(\underline{x}') = \Omega_\ell{}^*$ and $Z_\ell(\underline{x}'') = Z_\ell{}^*$. If $j \notin FV(\ell)$, then $x_j' = x_j'' = x_j(\ell)$. On the other hand, if $j \epsilon FV(\ell)$ and $x_j' > 0$, then $IRR_j \geq r(\ell)$. When $IRR_j = r(\ell)$, $j \notin Y(\ell)$ and $x_j'' = 0$. However, in this case $\sum_{k=o}^{m} \frac{B_{jk}-C_{jk}}{[1+r(\ell)]^k} = 0$, and so

$$x_j' \sum_{k=o}^{m} \frac{B_{jk}-C_{jk}}{[1+r(\ell)]^k} = x_j'' \sum_{k=o}^{m} \frac{B_{jk}-C_{jk}}{[1+r(\ell)]^k}.$$ Otherwise,

if $IRR_j > r(\ell)$, then $j \epsilon Y(\ell)$ and $x_j'' = 1 \geq x_j'$. Hence,

$$x_j' \sum_{k=o}^{m} \frac{B_{jk}-C_{jk}}{[1+r(\ell)]^k} \leq x_j'' \sum_{k=o}^{m} \frac{B_{jk}-C_{jk}}{[1+r(\ell)]^k}.$$

Finally, if $j \epsilon FV(\ell)$ and $x_j' = 0$, then $x_j'' = 0$ or 1, depending on whether or not $j \epsilon Y(\ell)$, but in any case

$$x_j'' \sum_{k=o}^{m} \frac{B_{jk}-C_{jk}}{[1+r(\ell)]^k} \geq 0.$$ Therefore, $\Omega_\ell{}^* \leq Z_\ell{}^*$.

The relevance of the new upper bounding functions in connection with what was previously discussed, can now be appreciated. In defining Y_ℓ' by means of W, constraint (3.2) was explicitly considered, and, as expected, the new optimal objective values to determine upper bounds were found to be better. This last statement is not fully substantiated by the assertion of Lemma 4.2, because the possibility of Ω_ℓ^* being equal to Z_ℓ^* is included. Nonetheless, under the assumption that the internal rates of return are comparatively large, and that the expenditure limits are sufficiently small, the statement becomes meaningful. In this case, if $j \in FV(\ell)$ and $x_j' = 0$, it can logically be inferred that, in many instances, x_j'' will be equal to 1, and that Ω_ℓ^* will be significantly smaller than Z_ℓ^*. The next step is then to consider the problem of how to obtain Ω_ℓ^*.

Let ℓ be any feasible node. Following the definition of Y_ℓ', the problem of calculating Ω_ℓ^* may be expressed as:

$$\left.\begin{aligned}
&\text{Max } \Omega = \sum_{j=1}^{n} x_j \sum_{k=0}^{m} \frac{B_{jk}-C_{jk}}{[1+r(\ell)]^k} && (4.1)\\
&\text{s.t. } \sum_{j=1}^{n} a_{jk}x_j \leq b_k,\ k = 0,1,\ldots,M && (4.2)\\
&0 \leq x_j \leq 1,\quad j \in FV(\ell) && (4.3)\\
&x_j = x_j(\ell),\ j \notin FV(\ell) && (4.4)
\end{aligned}\right\} \quad (4.5)$$

Substituting (4.4) in (4.1) and (4.2), problem (4.5) may be reformulated as follows:

$$\left.\begin{aligned}
&\text{Max } \hat{\Omega} = \sum_{j \in FV(\ell)} c_j x_j && (4.6)\\
&\text{s.t. } \sum_{j \in FV(\ell)} a_{jk}x_j \leq \hat{b}_k, k=0,1,\ldots,M && (4.7)\\
&0 \leq x_j \leq 1,\ j \in FV(\ell) && (4.8)
\end{aligned}\right\} \quad (4.9)$$

where

$$\hat{\Omega} = \Omega - \sum_{j \notin FV(\ell)} c_j x_j(\ell),$$

$$c_j = \sum_{k=0}^{m} \frac{B_{jk}-C_{jk}}{[1+r(\ell)]^k} \quad (j = 1,2,\ldots,n), \text{ and}$$

$$\hat{b}_k = b_k - \sum_{j \notin FV(\ell)} a_{jk}x_j(\ell)$$

Problem (4.9) is a linear programming problem which can obviously be solved by the simplex method. However, it can also be solved by repeated application of Dantizig's inspection rule[1], expressing the problem in the form of Ochoa-Rosso's formulation of the continous multi-dimensional knapsack problem[2]. To do this, it is only necessary to express (4.6)-(4.8) in terms of new variables y_{jk}.

LEMMA 4.3: Let y_{jk} be such that:

$$0 \leqslant y_{jk} \leqslant a_{jk}, \quad j \in FV(\ell), k = 0,1,\ldots,M \qquad (4.10)$$

and

$$\sum_{k=o}^{m} y_{jk} = x_j \sum_{k=o}^{m} a_{jk}, \quad j \in FV(\ell) \qquad (4.11)$$

Then problem (4.9) is equivalent to:

$$\left.\begin{aligned} &\text{Max } \bar{\Omega} = \sum_{j\in FV(\ell)} \sum_{k=o}^{m} f_j y_{jk} && (4.12) \\ &\text{s.t.} \sum_{j\in FV(\ell)} y_{jk} \leqslant \hat{b}_k, \quad k = 0,1,\ldots,M, && (4.13) \end{aligned}\right\} (4.14)$$

where :

$$f_j = \frac{c_j}{\sum_{k=o}^{m} a_{jk}}, \quad j \in FV(\ell) \qquad (4.15)$$

Proof: From (4.10) and (4.11), it follows that (4.8) is always satisfied. On the other hand, summing in (4.11) over j:

$$\sum_{k=o}^{m} \left(\sum_{j=1}^{n} y_{jk} \right) = \sum_{k=o} \left(\sum_{j=1}^{n} a_{jk} x_j \right), \quad j \in FV(\ell)$$

Hence, (4.7) holds, if, and only if, (4.13) holds. Finally, substituting (4.11) and (4.15) in (4.6), it can be seen that $\bar{\Omega} = \hat{\Omega}$.

Observing that $\bar{\Omega}$ is separable, and that each variable y_{jk} appears in one, and only one, inequality of (4.13) (with a coefficient of 1), an optimal solution to (4.14) can be obtained by successively setting each variable at its upper bound (or at its highest possible value so that (4.13) is not violated) in decreasing order of the f_j's. For each inequality in (4.13), if a non-negative f_j is found following the prescribed order, the remaining variables are then set at its lower bound. In this way, the optimal objective value of (4.14), which equals that of (4.9) and, ℓ being a feasible node, exists, can be obtained. Adding $\sum_{j \notin FV(\ell)} c_j x_j(\ell)$ to this value, Ω_ℓ^* is determined.

It only remains to show that, using the Ω_ℓ's instead of the Z_ℓ's, the scheme of the solution method leads to consistently bounding trees. To see that this is the case, let (ℓ,k) be any directed arc of TR_i $(i \geq 0)$. Y_ℓ' is a subset of Y_k' and $r(\ell) \leq r(k)$. Hence, $\Omega_k^* \leq \Omega_\ell^*$.

4.2.4) A NUMERICAL EXAMPLE.

In Table 4.4 and Figure 4.2 the summarized results of the procedure and the final directed tree for the third case of Section 4.2.2 are presented. As can be observed from these results, by the use of the new upper bounds the number of iterations decreased by 4 to 5, the number of terminal nodes by 4 to 6, and the maximal number of current active nodes by 2 to 3.

4.2.5) COMMENTS.

In sections 3.5 and 3.6 a number of aspects regarding the variability of the expenditure limits were discussed. Under the assumption of comparatively small internal rates of return, these were feasibility considerations based on a likely low level of involvement of the expenditure limits. Nevertheless, their real support was the fact that the upper bounds were independent of these limits. If the proposed improved upper bounds are used

Iteration i	ℓ	j	A_i	Concluding Nodes	Rejected Nodes	$\Omega_1^* = 8.235$	
						L_i	U_i
0	2	5	{2,3}	-	-	4.176	8.235
1	4	3	{4,5}	-	3	5.943	7.417
2	7	6	{5,6,7}	-	-	5.973	6.722
3	5	3	{5,6}	-	8,9	5.973	6.630
4	-	-	ϕ	6	10,11	5.973	-

TABLE 4.4 Summarized Results of the Numerical Example with $c_2(x)$ and Improved Upper Bounds.

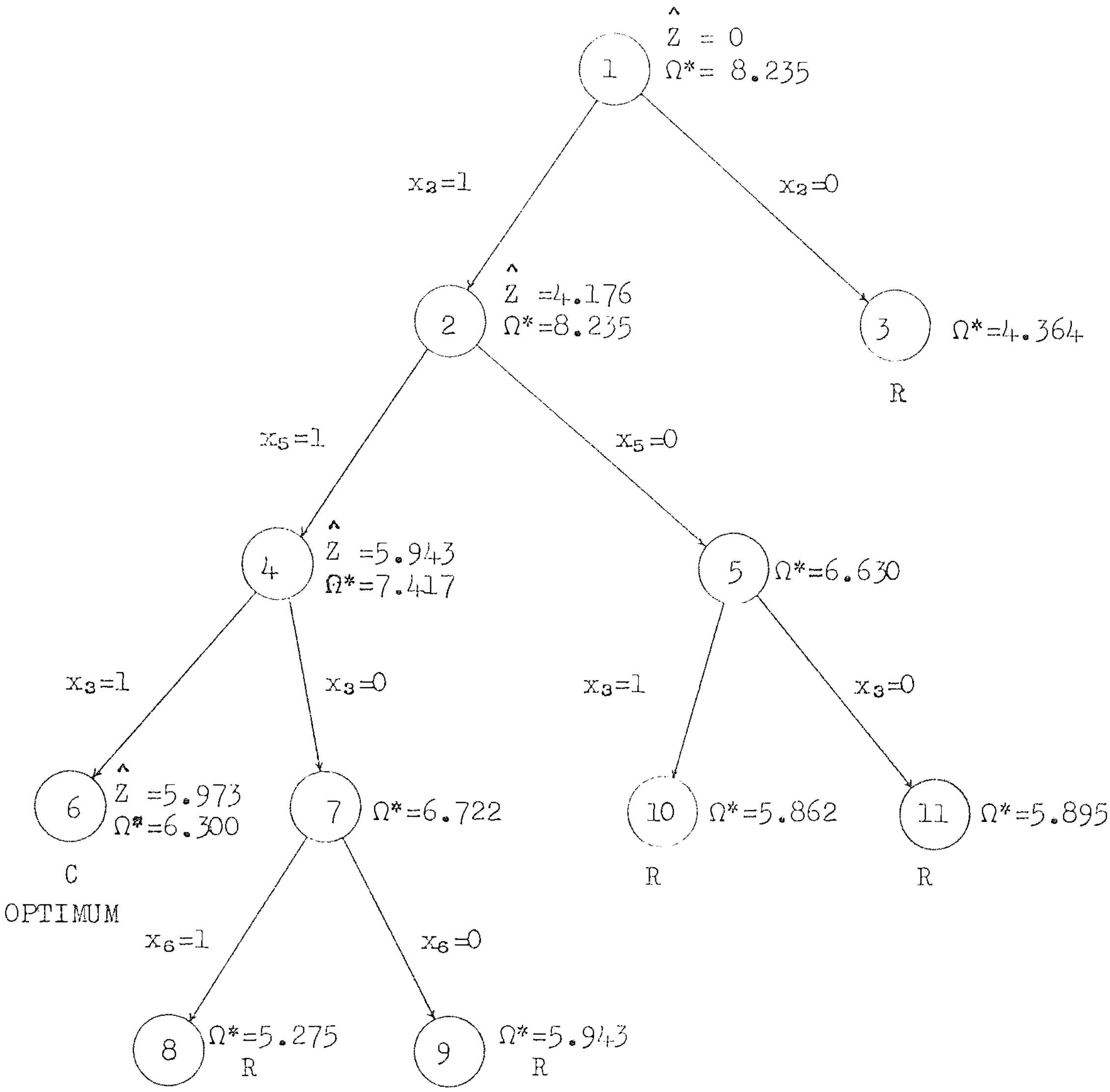

FIGURE 4.3 The Final Directed Tree with Improved Upper Bounds for $c_2(x)$.

to solve the original problem, this is obviously no longer the case. Hence, in order to know how much of the original calculations can be utilized when changes in the expenditure limits are introduced, not only feasibility, but also upper bounding considerations have to be taken into account.

4.3) IMPROVED LOWER BOUNDS.

4.3.1) PRELIMINARY REMARKS.

In the sense that, amongst the elements of S_ℓ, $\underline{x}(\ell)$ is the one with which the minimal number of accepted projects is associated, and that, at each iteration, only one project is considered for acceptance, the proposed solution method is similar to the so-called implicit enumeration methods[3,4]. In terms of the search procedure and the lower bounds, this means that only one free variable is fixed at one at a time, and that the current lower bound can only be increased accordingly. Therefore, particularly at the first stages of the algorithm, if the optimal solution (or solutions) to the problem includes the acceptance of a relatively large number of projects, the current lower bounds will be likely to correspond to poor feasible solutions. The availability of auxiliary solutions with a reasonable large number of acceptances, regardless of those associated with the $\underline{x}(\ell)$'s, is hence very desirable. These solutions, together with the $\underline{x}(\ell)$'s (as sillustrated in Section 3.4, if $\hat{\underline{x}} \in S_\ell$ and $\hat{\underline{x}} \neq \underline{x}(\ell)$, it cannot be guaranteed that $f(\hat{\underline{x}}) \geq \hat{Z}_\ell$), would certainly provide a much stronger basis to determine lower bounds.

4.3.2) ALTERNATIVE AUXILIARY SOLUTIONS.

Given $\underline{x}(\ell)$, $\tilde{\underline{x}}(\ell) = [\tilde{x}_1(\ell), \tilde{x}_2(\ell), \ldots, \tilde{x}_n(\ell)]$ will be defined by the following procedure:

CALCULATION OF THE ALTERNATIVE AUXILIARY SOLUTION (PROCEDURE P4)

Step 1: Set $\tilde{\underline{x}}(\ell) = \underline{x}(\ell)$

Step 2: If $FV(\ell) = \phi$, stop; $\tilde{\underline{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 3: Select an element j of $FV(\ell)$ with maximal IRR_j (suggested tie breaking rule: maximal investment).

Step 4: If $IRR_j \leq r(\ell)$, set $FV(\ell) = \phi$ and stop; $\tilde{\underline{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 5: If $IRR_j \leq c\left(\sum_{\nu=1}^{n} C_{\nu o}\tilde{x}_\nu(\ell) + C_{jo}\right)$, go to step 7.

Otherwise, continue.

Step 6: If $\sum_{\nu=1}^{n} a_{\nu k}\tilde{x}_\nu(\ell) + a_{jk} \leq b_k$, for $k = 0,1,\ldots,M$, set $\tilde{x}_j(\ell) = 1$ and $\widetilde{FV}(\ell) = FV(\ell) - \{j\}$, and go to step 8.

Otherwise, continue.

Step 7: Subtract $\{j\}$ from $FV(\ell)$ and go to step 2.

Step 8: If $\widetilde{FV}(\ell) = \phi$, stop; $\tilde{\underline{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 9: Select an element j of $\widetilde{FV}(\ell)$ with maximal IRR_j (suggested tie breaking rule: maximal investment).

Step 10: If $IRR_j \leq c\left[\sum_{\nu=1}^{n} C_{\nu o}\tilde{x}_\nu(\ell)\right]$, stop; $\tilde{\underline{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 11: If $IRR_j \leqslant c(\sum_{\nu=1}^{n} C_{\nu o}\tilde{x}_\nu(\ell) + C_{jo})$, go to step 13.

Otherwise, continue.

Step 12: If $\sum_{\nu=1}^{n} a_{\nu k}\tilde{x}_\nu(\ell) + a_{jk} \leqslant b_k$, for $k = 0,1,\ldots,M$, set

$\tilde{x}_j(\ell) = 1$ and go to step 13.

Otherwise, continue.

Step 13: Subtract $\{j\}$ from $\tilde{FV}(\ell)$ and go to step 8.

A number of relevant properties associated with $\underline{\tilde{x}}(\ell)$ are next presented.

LEMMA 4.4: For any feasible node ℓ, $\underline{\tilde{x}}(\ell)$ is an auxiliary solution of ℓ.

Proof: $\underline{x}(\ell) \epsilon S_\ell$, because ℓ is a feasible node. If $\underline{\tilde{x}}(\ell) = \underline{x}(\ell)$, then it is obviously an auxiliary solution of node ℓ. Otherwise, by steps 5,6,11 and 12 of P4, $\underline{\tilde{x}}(\ell) \epsilon S_\ell$ and the required result follows.

LEMMA 4.5: For any feasible node ℓ, if $\underline{x}(\ell) = \underline{\tilde{x}}(\ell)$, then ℓ is a concluding node.

Proof: Let it be assumed that $\underline{\tilde{x}} \neq \underline{x}(\ell)$, and that $\underline{\hat{x}} \epsilon S_\ell$. This means that, if $\hat{x}_j = 1$, then $IRR_j > c(\sum_{\nu=1}^{n} C_{\nu o}\hat{x}_\nu)$, and $\sum_{\nu=1}^{n} a_{\nu k}\hat{x}_\nu \leqslant b_k$, for $k = 0,1,\ldots,M$. On the other hand, for any $\underline{x} \epsilon S_\ell$, if $x_j(\ell) = 1$, then $x_j = 1$. Therefore, if $\hat{x}_j = 1$, then $\sum_{\nu=1}^{n} C_{\nu o}\hat{x}_\nu \geqslant \sum_{\nu=1}^{n} C_{\nu o}x_\nu(\ell) + C_{jo}$ and

$\sum_{\nu=1}^{n} a_{\nu k}\hat{x}_\nu \geqslant \sum_{\nu=1}^{n} a_{\nu k}x_\nu(\ell) + a_{jk}$, for $k = 0,1,\ldots,M$. In turn, this implies that $IRR_j > c(\sum_{\nu=1}^{n} C_{\nu o}x_\nu(\ell) + C_{jo})$ and that $\sum_{\nu=1}^{n} a_{\nu k}x_\nu(\ell) + a_{jk} \leqslant b_k$, for $k = 0,1,\ldots,M$. Hence, by steps 5 and 6 of P4, $\underline{\tilde{x}}(\ell) \neq \underline{x}(\ell)$, contradicting the original assumption of the lemma. Consequently, if $\underline{x}(\ell) = \underline{\tilde{x}}(\ell)$, Proof: (contd)

then S_ℓ has only one element and ℓ is a concluding node.

A direct implication of this result is that, with $\underline{\tilde{x}}(\ell)$, a means to know whether or not each current terminal node is concluding is introduced. The determination of active nodes as such is hence no longer necessary, and only the associated parameters have to be determined. The corresponding procedure would then be the following:

DETERMINATION OF PARAMETERS OF ACTIVE NODES (PROCEDURE P5).

Step 1: Obtain C_i and R_i, where C_i is the set of terminal concluding nodes and R_i is the set of terminal rejected nodes. Set $A_i = TN_i - C_i - R_i$.

Step 2: If $A_i = \phi$, stop; no parameters are necessary. Otherwise, continue.

Step 3: Select an element ℓ of A_i with maximal Z^* (or Ω^*) (suggested tie breaking rule: maximal $\hat{d}$, where $\hat{d} = \max\{\hat{Z}_\ell, \tilde{Z}_\ell = f[\tilde{x}(\ell)]\}$). Stop; the parameters of A_i are ℓ and j, where $IRR_j = \max_{\nu \epsilon FV(\ell)} \{IRR_\nu\}$

It can be seen that all the possible changes of $FV(\ell)$ in P2 are covered and justified along the same lines in P4, taking into account that for each created note the latter will have to be followed. This, of course, will also be done for the three nodes of TR_o, and so P1 can also be reduced accordingly:

CONSTRUCTION OF THE INITIAL TREE (PROCEDURE P6).

Step 1: Define N(T) = 1 and obtain the associated conepts of node 1.

If $\underline{x}(1) = \underline{\tilde{x}}$, stop; $\underline{x}^* = \underline{0}$ is the optimal solution to problem (3.5).

Otherwise, continue.

Step 2: Define $N(t_1) = 2$ and $N(t_2) = 3$, where $\mathbb{P}_j(T) = \{t_1, t_2\}$ and j is such that $IRR_j = \max_{\nu \epsilon FV(1)} \{IRR_\nu\}$. The procedure is complete: $SN_0 = \{1,2,3\}$, $SA_0 = \{(1,2),(1,3)\}$ and $TR_0 = \{SN_0, SA_0\}$.

As regards the branching rule, no changes are introduced with this approach.

It is finally noted that, after completing procedure P4, it will be known whether or not $\underline{x}(\ell)$ is the only auxiliary solution of ℓ, and, if not, an alternative auxiliary solution $\tilde{\underline{x}}(\ell)$ will be available. Obviously, the auxiliary solution to be used will be the one with greater objective value (again, it is pointed out that the possibility of $\underline{x}(\ell)$ being a better feasible solution than $\tilde{\underline{x}}(\ell)$ cannot be disregarded). This value will be denoted by $\hat{\Omega}_\ell$.

4.3.3) RE-STATEMENT OF THE ALGORITHM.

Taking into account the considerations of the preceding section, the algorithm presented in Section 3.3.3 can be re-stated as follows:

START (ITERATION 0)

1. Follow the steps of procedure P6.
2. If $\underline{x} = \underline{0}$ is optimal, stop.
   Otherwise, continue.
3. Follow the steps of procedure P5 for $i = 0$.
4. If $A_0 = \phi$, stop; $L_0 = \max \{\hat{\Omega}_2, \hat{\Omega}_3\} = \hat{\Omega}_2 = \hat{Z}_2$ corresponds to the optimal solution to problem (3.5).
   Otherwise continue.
5. If $L_0 = U_0$, stop; $L_0 = \max \{\hat{\Omega}_2, \hat{\Omega}_3\}$ corresponds to an optimal solution to problem (3.5).
   Otherwise, set $i = 1$ and continue.

ITERATION i.

1. Follow the steps of procedure P3

2. Follow the steps of procedure P5

3. If either $A_i = \phi$ or $L_i = U_i$, stop; L_i corresponds to an optimal solution to problem (3.5)

   Otherwise, set $i \leftarrow i+1$ and start ieration i.

The following points in connection with the algorithm should be observed:

1. If $\underline{x} = \underline{0}$ is not optimal and $A_0 = \phi$, then $S_2 = \{\underline{x}(2)\}$ and $S_3 = \{\underline{x}(3) = \underline{0}\}$. Clearly, since $\hat{Z}_2 > \hat{Z}_3 = 0$, $L_0 = \hat{Z}_2$ corresponds to the optimal solution to problem (3.5). However, if $A_0 \neq \phi$, in general nothing precise about the elements of S_2 and S_3, nor about $\hat{\Omega}_2$ and $\hat{\Omega}_3$, can be stated. Therefore, L_0 is simply defined as $\max\{\hat{\Omega}_2, \hat{\Omega}_3\}$.

2. When procedure P5 is followed, C_i and R_i have to be obtained (note that $R_i \cap C_i$ is not necessarily empty). For this reason, procedure P4 has to befollowed at this stage for nodes r+1 and r+2. In this case, the corresponding auxiliary solutions could differ from all the auxiliary solutions associated with TR_{i-1}, and so L_i will be given by $\max\{L_{i-1}, \hat{\Omega}_{r+1}, \hat{\Omega}_{r+2}\}$. After completing P4, it will be known whether or not nodes r+1 and r+2 are concluding, and R_i can then be obtained as indicated previously (see Section 3.3).

4.3.4) <u>A NUMERICAL EXAMPLE</u>.

In Table 4.5 and Figure 4.4 the summarized results of the procedure and the final directed tree for the example of Section 4.2.4 with improved lower bounds are presented. As can be observed, the configuration of the final directed tree did not change. However, the optimal solution was found after the first iteration, as a consequence of which node 3 was rejected at the same iteration. This, together with the fact that node 6 was found to be concluding after the third iteration, rather than after the fifth, made the maximal number of current active nodes decrease by 1 to 2.

4.3.5) COMMENTS.

Although at different stages of the algorith, both the original and the improved version of the lower bounds are linked with the expenditure limits in exactly the same way. Of course, with the improved lower bounds arising from auxiliary solutions involving a relatively large number of acceptances, the restrictions imposed by these limits will be likely to be considered at earlier stages of the procedure. However, the considerations themselves do not differ. Therefore, this and further restrictions associated with the upper bounds being taken into account, the same kind of criteria as those of the original version can be used to deal with changes of the expenditure limits.

4.4) SUMMARY.

In this chapter improved upper and lower bounds for the original version of the algorithm were proposed. As opposed to how the upper bounds were defined in that version, the constraints associated with the expenditure limits were now fully considered. The implication of this approach is the need to know the solution of the linear programming problem each time a node is created, but the problem is simple enough to solve it by inspection. In general, the resulting upper bounds were seen to be at least as good as those of the original version; in particular, if comparatively large internal rates of return and small expenditure limits are involved,

Iteration i	ℓ	j	A_i	Concluding Nodes	Rejected Nodes	$\underline{x}(1) \neq \tilde{\underline{x}}(1)$	
						L_i	U_i
0	2	5	{2}	-	3	5.973	8.235
1	4	3	{4,5}	-	-	5.973	7.417
2	7	6	{5,7}	6	-	5.973	6.722
3	5	3	{5}	-	8,9	5.973	6.630
4	-	-	ϕ	-	10,11	5.973	-

TABLE 4.5 Summarized Results of the Numerical Example with $c_2(x)$ and Improved Upper and Lower Bounds.

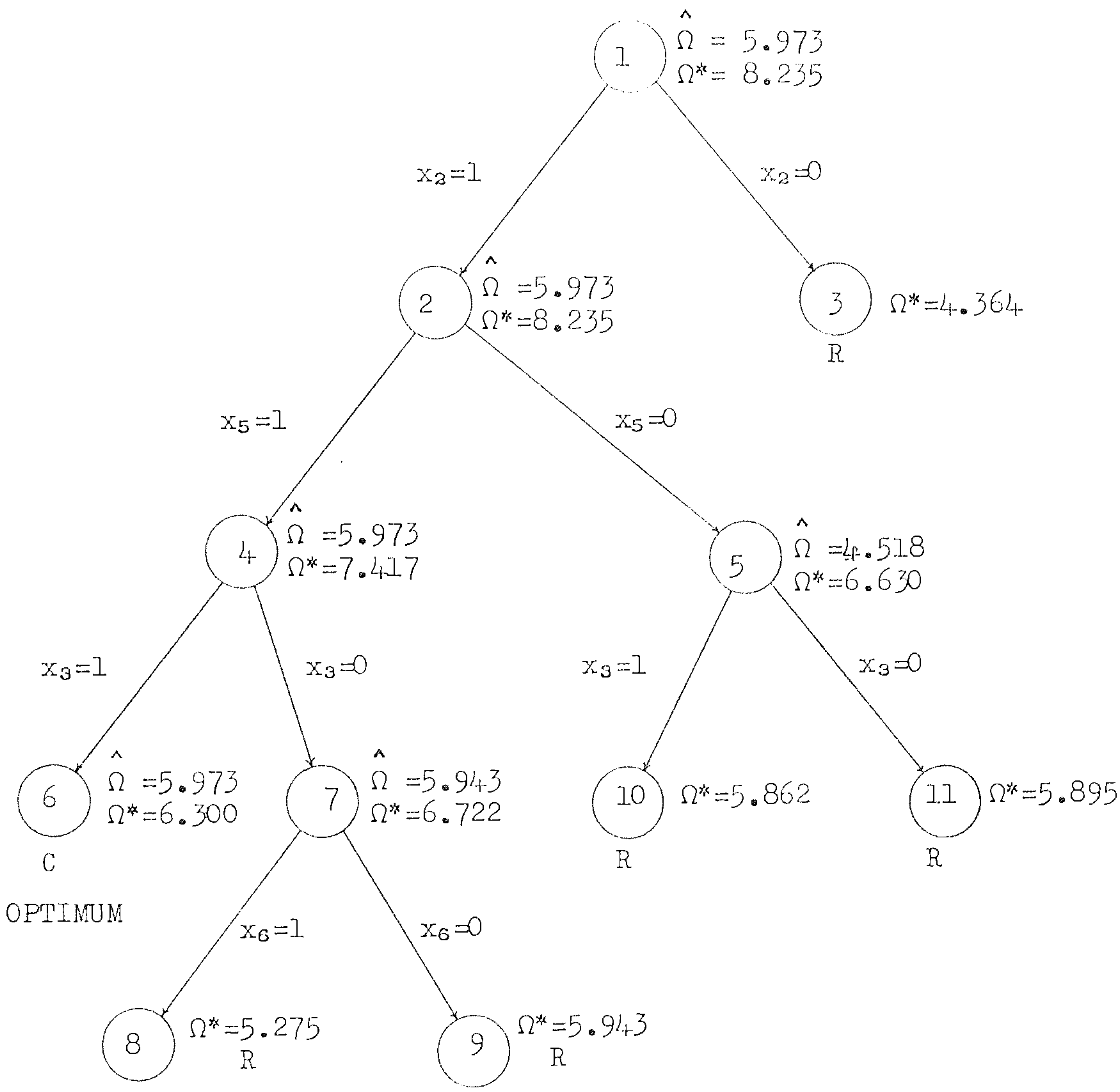

FIGURE 4.4 The Final Directed Tree with Improved Upper and Lower Bounds for $e_2(x)$.

significant improvements can be expected at the cost of the corresponding extra computations. As for the lower bounds, it was pointed out that, while the original auxiliary solutions are a result of feasibility conditions, they are only linked with possible increments of the objective function in terms of the acceptances imposed by the applications of the branching rule. The number of these acceptances is equal to 1 at each iteration, and so good early feasible solutions cannot be expected in general, particularly if many acceptances are involved with the optimal solution. Accordingly, a look-ahead procedure following the same criterion as that of the branching rule (the maximal IRR criterion) was proposed to anticipate future feasible acceptances which could eventually be reached by the original auxiliary solutions. Alternative auxiliary solutions with relatively many acceptances were thus defined and incorporated, leading to overall improved lower bounds.

REFERENCES.

[1] J.B.DANTZIG (1957) Discrete variable extremum problems Ops.Res.5, 266-277.

[2] F.OCHOA-ROSSO (1968) Applications of Discrete Optimization Techniques to Capital Investment and Network Synthesis Problems. Sc.D.Dissertation, Department of Civil Engineering, Massachusetts Institute of Technology, Boston, Massachusetts.

[3] E.BALAS (1965) An additive algorithm for solving linear programs with zero-one variables. Ops.Res.13,517-546.

[4] F.GLOVER (1965) A multiphase-dual algorithm for the zero-one integer programming problem. Ops.Res.13,879-919.

CHAPTER V

APPLICATIONS UNDER EXTENDED ASSUMPTIONS.

5.1) INTRODUCTION.

In this chapter a number of variants arising from the class of capital investment problems under consideration will be discussed. Specifically, aspects in connection with financing decisions and technological dependence will now be contemplated, in addition to those regarding the economical structure associated with the variability of the cost of capital (as a function of the level of investment). In each case, the problem will be formulated and identified within the context of the solution technique developed in the preceding chapters.

5.2) FINANCING DECISIONS.

5.2.1) FORMULATION OF THE PROBLEM.

In Section 3.2 it was assumed that capital funds for investment, available only for combinations of projects with positive individual net present values, could be obtained at a fixed unitary cost $c(t)$, given the corresponding overall investment requirement t. One way to interpret this assumption is to think of a market in which capital suppliers impose identical conditions on the availability of money, the total level of investment t always being taken into account. Under these circumstances, assuming further that the supply of money is sufficiently large, and that both the funds for investment and the subsequent cash requirements have to be borrowed from the capital suppliers at the cost $c(t)$, the supposition in question can appropriately be used. In this framework, it will now be considered that the capital suppliers work under different competitive terms. In particular, together with suppositions i)- vii) of Section 3.2 (without imposing a restriction on the size of the IRR_j's), it will be assumed that:

viii) $S_1, S_2, \ldots, S_S$ are the existing capital suppliers from which the requirements have to be borrowed

ix) For any final selection including only projects with positive NPV, each capital suppliers S_i is in position to provide funds as indicated below:

ix)' At the beginning of the first time period, any amount Q_{io} up to b_{io} units at the rate of interest $r_{io}(Q_{io})$. The r_{io}'s are assumed to be positive non-decreasing functions of the Q_{io}'s.

ix)" At the end of the k-th time period, any amount Q_{ik} up to b_{ik} units at the rate of interest $r_{ik}(Q_{ik})$, where:

$$r_{ik}(Q_{ik}) = \frac{\sum_{i=1}^{s} Q_{io} r_{io}(Q_{io})}{\sum_{i=1}^{s} Q_{io}}$$

This means that the rates of interest of the operating requirements are determined by the investment requirements. It is also assumed that, for each selection, the total supply does not surpass the corresponding requirements (in other words, that only that what is necessary can be borrowed).

x) The cost of capital associated with the funds borrowed from the capital suppliers is given by the following weighted average:

$$c(\underline{Q}) = \frac{\sum_{k=o}^{M} \sum_{i=1}^{s} Q_{ik} c_k(\underline{Q}_k)}{\sum_{k=o}^{M} \sum_{i=1}^{s} Q_i k},$$

where:

$$c_k(\underline{Q}_{ik}) = \frac{\sum_{i=1}^{s} Q_{ik}\, r_{ik}(Q_{ik})}{\sum_{i=1}^{s} Q_{ik}}$$

Accordingly, the problem may be formulated as:

$$\text{Max } z = \sum_{j=1}^{n} NPV_j(\underline{Q})x_j \qquad (5.1)$$

$$\text{s.t. } \sum_{i=1}^{s} Q_{ik} = \sum_{j=1}^{n} a_{jk}x_j, \; k = 0,1, \ldots, M \qquad (5.2)$$

$$\sum_{j=1}^{n} a_{jk}x_j \leq b_k, \; k = 0,1, \ldots, M \qquad (5.3)$$

$$NPV_j(\underline{Q}) > 0, \text{ if } x_j = 1, \; j = 1,2,\ldots,n \qquad (5.4)$$

$$0 \leq Q_{ik} \leq b_{ik}, \; i = 1,2,\ldots,s, \; k = 0,1,\ldots,M \qquad (5.5)$$

$$x_j = 0 \text{ or } 1, \; j = 1,2, \ldots, n \qquad (5.6)$$

$$\Bigg\} \; (5.7)$$

where

$$NPV_j(\underline{Q}) = \begin{cases} \sum_{k=o}^{n} \dfrac{B_{jk}-C_{jk}}{[1+c(\underline{Q})]^k}, & \text{if } \underline{Q} \neq (\underline{0},\underline{0},\ldots,\underline{0}) \\ 0, \text{ otherwise} & \end{cases}$$

Expressions (5.1), (5.3), (5.4) and (5.6) correspond to expressions (3.1)-(3.4) of Section 3.2, respectively, and constraints (5.2) and (5.5) arise from assumption ix). In the next section it will be seen how the form of problem (5.7) can be simplified to that of problem (3.5), and hence that the proposed solution method for the latter may also be applied in this case.

5.2.2) IDENTIFICATION OF A SOLUTION METHOD.

While the aim of problem (5.7) continues to be the maximization of the overall NPV, the final decision now involves not only projects, but also capital suppliers and corresponding capital funds to be borrowed. Each project selection defines fixed levels of requirements, which, in turn, depending upon the amounts borrowed from each capital supplier, define a fixed cost of capital determining the individual contribution of each project towards the maximization in question. In this sense, the financing problem and the project selection problem are clearly linked. Nevertheless, for many practical purposes, the optimal strategy can be expected to minimize the cost of capital. This being the case, the financing problem could be imbedded into the project selection problem, as fixed levels of requirements would always be known to incur the associated minimal cost of capital. It will be seen that this argument is valid for problem (5.7).

LEMMA 5.1: Let $\underline{Q}^*(\hat{\underline{t}}) = \underline{Q}^*(\hat{t}_0, \hat{t}_1, \ldots, \hat{t}_k)$ be an optimal solution to the problem:

$$\left.\begin{aligned} &\text{Min } c(\underline{Q}) \\ &\text{s.t.} \quad \sum_{i=1}^{s} Q_{ik} = \hat{t}_k, \; k = 0,1,\ldots,M \\ &\qquad 0 \leq Q_{ik} \leq b_k, \; i = 1,2,\ldots,s, k = 0,1,\ldots,M, \end{aligned}\right\} \quad (5.8)$$

where:

$$\hat{t}_k = \sum_{j=1}^{n} a_{jk} \hat{x}_j > 0$$

$$k = 0,1,\ldots,M,$$

and $\hat{\underline{x}} = (\hat{x}_1, \hat{x}_2, \ldots, \hat{x}_m)$ is a binary specification of $\underline{x}$.

Then $\underline{Q}^*(\hat{\underline{t}})$ is also an optimal solution to the problem:

$$\left.\begin{aligned} &\text{Max } \sum_{j=1}^{n} NPV_j(\underline{Q})\hat{x}_j \\ &\text{s.t.} \sum_{i=1}^{s} Q_{ik} = \hat{t}_k, \; k = 0,1,\ldots,M \\ &\qquad 0 \leq Q_{ik} \leq b_{ik}, \; i = 1,2,\ldots,s, k = 0,1,\ldots,M \end{aligned}\right\} \quad (5.9)$$

Proof: The feasibility regions of both problems are exactly the same. On the other hand, there is no feasible solution $\underline{Q}$ such that $c(\underline{Q}) < c[\underline{Q}^*(\hat{\underline{t}})]$. Therefore, since $\sum_{k=0}^{m} \frac{B_{jk} - C_{jk}}{(1+y)^k}$ is decreasing for non-negative values of y (see Appendix CHIII), no feasible solution $\underline{Q}$ exists, such that:

$$\sum_{k=0}^{m} \frac{B_{jk}-C_{jk}}{[1+c(\underline{Q})]^k} > \sum_{k=0}^{m} \frac{B_{jk}-C_{jk}}{\{1+c[\underline{Q}^*(\hat{\underline{t}})]\}^k},$$

or

$$\sum_{j=1}^{n} \hat{x}_j \sum_{k=0}^{m} \frac{B_{jk}-C_{jk}}{[1+c(\underline{Q})]^k} > \sum_{j=1}^{n} \hat{x}_j \sum_{k=0}^{m} \frac{B_{jk}-C_{jk}}{\{1+c[\underline{Q}^*(\hat{\underline{t}})]\}^k}$$

Consequently, $\underline{Q}^*(\hat{\underline{t}})$ is an optimal solution to problem (5.9).

In view of constraint (5.6), one rudimentary way to solve problem (5.7) would be to consider all the binary specifications of $\underline{x}$ satisfying constraint (5.3). For each such specification $\hat{\underline{x}}$, the procedure could be defined as follows:

1. If $\sum_{i=1}^{s} b_{ik} > b_k$ and $\sum_{j=1}^{n} a_{jk} \hat{x}_j > \sum_{i=1}^{s} b_{ik}$, for some $k = 0,1,\ldots,M$, $\hat{\underline{x}}$ is not feasible (because of (5.5), (5.2) could not be satisfied). Otherwise, continue.
2. Solve problem (5.8) (hence dealing with constraints (5.2) and (5.5)).
3. If $NPV_j[\underline{Q}^*(\hat{\underline{t}})] \leq 0$ and $\hat{x}_j = 1$, $\hat{\underline{x}}$ is not feasible (constraint (5.4) would not be satisfied). Otherwise, $[\hat{\underline{x}}, \underline{Q}^*(\hat{\underline{t}})]$ is an optimal solution to (5.7) when $\underline{x} = \hat{\underline{x}}$ (by Lemma 5.1).

Either by explicit or by implicit enumeration of all the binary specifications of $\underline{x}$ satisfying constraint (5.3), this procedure would lead to an optimal solution to problem (5.7), or to an indication that no such solution exists (apart from the trivial solution $\underline{x} = \underline{0}$). Along these lines, an equivalent formulation of the problem would then be:

$$\left.\begin{aligned} &\text{Max } Z = \sum_{j=1}^{n} NPV_j[\underline{Q}^*(\underline{t})]x_j \\ &\text{s.t.} \quad \sum_{j=1}^{n} a_{jk}x_j \leqslant \min\{b_k \sum_{i=1}^{s} b_{ik}\}, k=0,1,\ldots,M \\ &\quad NPV_j[\underline{Q}^*(\underline{t})] > 0, \text{ if } x_j=1, j=1,2,\ldots,n \\ &\quad x_j = 0 \text{ or } 1, \ j = 1,2,\ldots,n, \end{aligned}\right\} \qquad (5.10)$$

where $\underline{t} = (t_0,t_1,\ldots,t_M)$, $t_k = \sum_{j=1}^{n} a_{jk}x_j (k = 0,1,\ldots,M)$,

$\underline{Q}^*(\underline{d})$ is an optimal solution to the problem:

$$\left.\begin{aligned} &\text{Min } c(\underline{Q}) \\ &\text{s.t.} \quad \sum_{i=1}^{s} Q_{ik} = d_k \geqslant 0, \ k = 0,1,\ldots,M \\ &\qquad 0 \leqslant Q_{ik} \leqslant b_{ik}, \ i = 1,2,\ldots,s, \ k = 0,1,\ldots,M \end{aligned}\right\} \qquad (5.11)$$

and $\underline{Q}^*(\underline{0}) = (\underline{0},\underline{0},\ldots,\underline{0})$.

Obviously, if $c[\underline{Q}^*(\underline{d})]$ were found to be a positive non-decreasing function of d_0 (independent of $d_1,d_2,\ldots,d_M$), then the form of problem (5.10) would exactly correspond to that of problem (3.5), and the proposed solution method would therefore be also applicable (of course, as will be discussed further on, this would also mean that problem (5.11) would have to be dealt with). Now, by assumption x):

$$c(\underline{Q}) = \frac{\sum_{k=o}^{m} \sum_{i=1}^{s} Q_{ik}c_k(\underline{Q}_k)}{\sum_{k=o}^{M} \sum_{i=1}^{s} Q_{ik}}$$

$$= \frac{\sum_{k=o}^{M} \sum_{i=1}^{s} Q_{ik}r_{ik}(Q_{ik})}{\sum_{k=o}^{M} \sum_{i=1}^{s} Q_{ik}}$$

Hence, by assumption ix)":

$$c(\underline{Q}) = \frac{\sum_{k=o}^{M} \sum_{i=1}^{s} Q_{ik} c_o(\underline{Q}_o)}{\sum_{k=o}^{M} \sum_{i=1}^{s} Q_{ik}} = c_o(\underline{Q}_o) \tag{5.12}$$

By assumptions ix)' and x), $c_o(\underline{Q}_o)$ is an average of positive non-decreasing rates of interest. By (5.12), on the other hand, $c_o[\underline{Q}_o^*(d_o)]$ is minimal in the feasibility region of (5.11), and so $c[\underline{Q}^*(\underline{d})]$ is a positive non-decreasing function of d_o (independent of $d_1, d_2, \ldots, d_M$). The proposed solution method is, therefore, applicable, and it only remains to discuss how problem (5.11) could be handled.

Expression (5.12), in accordance with assumption ix)", indicates that the cost of capital is not affected by the operating requirements. Thus, so long as:

$$\sum_{i=1}^{s} Q_{ik} = d_k, \ k = 1,2,\ldots,M, \text{ and}$$

$$0 \leq Q_{ik} \leq b_{ij}, \ i = 1,2,\ldots,s, \ k = 1,2,\ldots,M,$$

problem (5.11) can be replaced by:

$$\left.\begin{aligned} &\text{Min} \quad c_o(\underline{Q}_o) \\ &\text{s.t.} \ \sum_{i=1}^{s} Q_{io} = d_o > 0 \\ &\qquad 0 \leq Q_{io} \leq b_{io}, \ i = 1,2,3\ldots,s \end{aligned}\right\} \tag{5.13}$$

Multiplying $c_o(\underline{Q}_o)$ by d_o, which is a positive parameter of (5.13), this problem can again be replaced by:

$$\left.\begin{aligned} &\text{Min} \sum_{i=1}^{s} Q_{io} r_{io}(Q_{io}) \\ &\text{s.t.} \ \sum_{i=1}^{s} Q_{io} = d_o \\ &\qquad 0 \leq Q_{io} \leq b_{io}, \ i = 1,2,\ldots,s \end{aligned}\right\} \tag{5.14}$$

If the r_{io}'s were constant (observe that in this case the cost of capital would not necessarily have to be constant), then (5.14) would be a linear programming problem which could be solved by inspection (setting each variable at its upper bound, or at its highest possible value so that feasibility is maintained, in decreasing order of the r_{io}'s). Otherwise, depending upon the properties of the r_{io}'s, a number of alternative methods could be used to solve problem (5.14). For example, it could be solved by separable programming[1,2], and, under certain differentiability conditions, also by other methods using linear approximations[3] or penalities[4]. In any case, if only solutions corresponding to one value of d_o were provided, then the problem would have to be solved several times along or before the execution of the algorithm (although, clearly, existing solutions could be used as inputs to obtain other solutions). If the problem were suitable to be approximated by restricting the variables to integer values, one way to overcome this difficulty would be to use dynamic programming[5,6]. In general, a reasonably small number of capital suppliers can be expected, which means that, in terms of the number of variables, a problem of reasonably small size would have to be solved. However, in order to avoid severe storage requirements, d_o should not be very large. This being the case, the problem could be solved for $d_o = \min\{ \sum_{j=1}^{n} a_{jo}, \sum_{i=1}^{s} b_{io}, b_o \}$ and all the solutions corresponding to smaller values of d_o would be generated during the process.

5.2.3) COMMENTS.

As in Section 3.2, in the preceding sections the discussion was addressed to projects involving only singe-stage investment requirements , under assumptions in which the cost of capital is only affected by these requirements. Again, however, the possibility of subsequence negative net cash flows was included in the formulation of the problem. If these cash flows were to affect the cost of capital and the weighted average approach[7] were used to define this cost (see Section 5.2.1), then the applicability of the proposed solution method, as mentioned in Section 3.5, would simply depend on whether or not the cost of capital is a positive non-decreasing function in each of its variables, at least in the range under consideration. Otherwise, the method could still be adapted to solve the problem, if after reasonably low levels of requirements the cost of capital, once again, were positive and non-decreasing. Finally, it is pointed out that extending assumption ix)' for several periods (see section 5.2.1), the conditions regarding the direct applicability of the method and the cost of capital would not necessarily be met.

5.3) TECHNOLOGICALLY DEPENDENT PROJECTS.

5.3.1) DEFINITION.

DEFINITION 5.1: Let $U = \{P_1, P_2, \ldots, P_n\}$ be a set of projects. For any $j = 1,2,\ldots,n$, if, apart from capital rationing and disirability limitations, the acceptance or rejection of P_j affects the possibility of accepting or rejecting at least one of the others, or vice versa, then P_j will be said to be TECHNOLIGICALLY DEPENDENT.

Three kinds of technologically dependent projects will be discussed in this section: mutually exclusive, complementary and supplementary projects (hereafter, no imposition on the size of the IRR_j's will be assumed).

5.3.2) MUTUALLY EXCLUSIVE PROJECTS.

DEFINITION 5.2: Given a set of projects, if the acceptance of any one of them implies the rejection of all the others, they are referred to as MUTUALLY EXCLUSIVE projects.

Analytically, it is very easy to deal with mutually exlcusive projects. In addition to suppositions i)-iv), vi) and vii) of Section 3.2, let it be assumed that projects $P_1, P_2, \ldots, P_{n_1}$ are technologically independent, that projects $P_{n_1+1}, P_{n_1+2}, \ldots, P_n$ are mutually exclusive, and, again, that acceptances and rejections do not affect the size of the defining cash flows. The following constraint would clearly guarantee that the zero-one values of the x_j's $(j = n_1+1, n_1+2, \ldots, n)$ comply with the requirements of Definition 5.2:

$$\sum_{j=n_1+1}^{n} x_j \leq 1$$

Hence, the problem can be stated as follows:

$$\left.\begin{array}{ll} \text{Max } Z = \sum_{j=1}^{n} NPV_j(t)x_j & \\ \text{s.t.} \quad \sum_{j=1}^{n} a_{jk}x_j \leq b_k, k = 0,1,\ldots,M+1 & \\ \quad NPV_j(t) > 0, \text{ if } x_j=1, j=1,2,\ldots,n & \\ \quad x_j = 0 \text{ or } 1, \; j = 1,2,\ldots,n, & \end{array}\right\} \quad (5.15)$$

where:

$$a_{j(M+1)} = \begin{cases} 0, & \text{if } j \in \{1,2,\ldots,n_1\} \\ 1, & \text{otherwise} \end{cases}$$

and

$$b_{M+1} = 1$$

Problem (5.15) has exactly the same form as that of problem (3.5) (see Section 3.2), and, therefore, the application of the proposed solution method is direct (a more straightforward way to deal with mutually exclusive projects is illustrated in Section 5.3.5).

5.3.3) COMPLEMENTARY PROJECTS.

DEFINITION 5.3: Let P_j and P_k be any two different projects, such that, if P_j is accepted (rejected), then P_k has to be accepted (rejected), and vice versa. In this case, P_j and P_k will be said to be COMPLEMENTARY projects.

This definition can be taken into account by means of the following constraint:

$$x_j - x_k = 0 \tag{5.16}$$

By equation (5.16), $x_j = 1(0) \Leftrightarrow x_k = 1(0)$, which means that P_j and P_k are complementary. Let supposition v) of Section 3.2 be modified as follows:

v)' Projects $P_1, P_2, \ldots, P_{n_1}$ are technologically independent. The remaining projects are such that, given P_j for some $j \in \{n_1+1, n_1+2, \ldots, n\}$, at least one element k of $\{n_1+1, n_1+2, \ldots, n\}$ exists, for which P_j and P_k are complementary. Acceptances and rejections do not affect the size of the defining cash flows.

Obviously, as a consequence of this assumption, different pairs of complementary projects with a common component would be permitted. However, not all of these pairs have to be exhibited. For example, if (n_1, n_2) and (n_2, n_3) corresponded to pairs of complementary projects, then (n_1, n_3) would clearly correspond to a pair of complementary projects too. Nonetheless, only two constraints would have to be included; say:

$$x_{n_1} - x_{n_2} = 0, \tag{5.17}$$

and

$$x_{n_2} - x_{n_3} = 0 \qquad (5.18)$$

If (5.17) and (5.18) hold, then $x_{n_1} - x_{n_3} = 0$ also has to hold. Therefore, this constraint would be redundant. The minimal set of pairs representing all the complementary relationships among the projects (pairs leading to redundant constraints would not be included) will be denoted by A.

Under all the other assumptions of Section 3.2, the problem would now be to:

$$\left.\begin{aligned} &\text{Max } Z = \sum_{j=1}^{n} NPV_j(t)x_j \\ &\text{s.t. } \sum_{j=1}^{n} a_{jk}x_j \leq b_k,\ k = 0,1,2...,M \\ &x_j - x_\ell = 0,\ (j,\ell)\epsilon A \\ &NPV_j(t) > 0,\ \text{if } x_j=1, j=1,2,...,n \\ &x_j=0 \text{ or } 1,\ j = 1,2,...,n \end{aligned}\right\} \qquad (5.19)$$

Problem (5.19) does not have the same structure as that of problem (3.5), but simple modifications can be introduced to adapt the proposed solution method. The discussion will be referred to the improved lower bounds approach (see Section 4.3).

In the first place, since the partitioning function is defined in terms of one variable only , Definition 3.2, in view of assumption v)', would not necessarily lead to an auxiliary solution in the context of the proposed algorithm (see Section 3.3.2). In this connection, $\underline{x}(\ell)$ can be re-definied as follows:

DEFINITION 5.4: $\underline{x}(\ell) = [x_1(\ell), x_2(\ell),...,x_n(\ell)]$ will be said to be the ℓ-th SPECIFICATION of $\underline{x}$, where:

$$x_j(\ell) = \begin{cases} x_j, \text{if } x_j \text{ is non-free at } \ell \\ x_t, \text{if } x_t \text{ is non-free at } \ell \text{ and } j\epsilon Q_t \\ 0, \text{otherwise} \end{cases}$$

and

$$Q_t = \{\nu > n_1 / \nu \nmid t \text{ and } (\nu,t) \in A\},$$

for any j and any t.

The set Q_t merely represents the projects with which P_t is complementary, and a number of conditions regarding its role have to be observed in order to be consistent with the method. It can be easily seen that the following modified versions of procedures P4, P6 and P3 satisfy the corresponding additional requirements.

CALCULATION OF THE ALTERNATIVE AUXILIARY SOLUTION (PROCEDURE P7)

Step 1: Set $\underline{\tilde{x}}(\ell) = \underline{x}(\ell)$.

Step 2: If $FV(\ell) = \phi$, stop; $\underline{\tilde{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 3: Select an element j of $FV(\ell)$ with maximal IRR_j (suggested the breaking rule: maximal investment).

Step 4: If $IRR_j \leqslant r(\ell)$, set $FV(\ell) = \phi$ and stop; $\underline{\tilde{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 5: If $IRR_t \leqslant c(\sum_{\nu=1}^{n} C_{\nu o}\tilde{x}_\nu(\ell) + \sum_{\nu \in Q_j \cup \{j\}} C_{\nu o})$ for any $t \in Q_j \cup \{j\}$, go to step 7.

Otherwise, continue.

Step 6: If $\sum_{\nu=1}^{n} a_{\nu k}\tilde{x}_\nu(\ell) + \sum_{\nu \in Q_j \cup \{j\}} a_{\nu k} \leqslant b_k$, for $k = 0,1,\ldots,M$, set $\tilde{x}_\nu(\ell) = 1 \; \forall \, \nu \in Q_j \cup \{j\}$ and $\tilde{FV}(\ell) = FV(\ell) - (Q_j \cup \{j\})$, and go to step 8.

Otherwise, continue.

Step 7: Subtract $Q_j \cup \{j\}$ from $FV(\ell)$ and go to step 2.

Step 8: If $\tilde{FV}(\ell) = \phi$, stop; $\underline{\tilde{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 9: Select an element j of $\widetilde{FV}(\ell)$ with maximal IRR_j (suggested tie breaking rule: maximal investment)

Step 10: If $IRR_j \leq c[\sum_{\nu=1}^{n} C_{\nu o}\tilde{x}_{\nu}(\ell)]$, stop; $\underline{\tilde{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 11: If $IRR_t \leq c(\sum_{\nu=1}^{n} C_{\nu o}\tilde{x}_{\nu}(\ell) + \sum_{\nu\epsilon Q_j \cup \{j\}} C_{\nu o})$ for any $t\epsilon Q_j \cup \{j\}$, go to step 13.

Otherwise, continue.

Step 12: If $\sum_{\nu=1}^{n} a_{\nu k}\tilde{x}_{\nu}(\ell) + \sum_{\nu\epsilon Q_j \cup \{j\}} a_{\nu k} \leq b_k$, for $k = 0,1,\ldots,M$, set $\tilde{x}_{\nu}(\ell) = 1 \;\forall \nu\epsilon Q_j \cup \{j\}$, and go to step 13.

Otherwise, continue.

Step 13: Subtract $Q_j \cup \{j\}$ from $\widetilde{FV}(\ell)$ and go to step 8.

CONSTRUCTION OF THE INITIAL TREE (PROCEDURE P8).

Step 1: Define $N(T) = 1$ and obtain the associated concepts of node 1.

If $\underline{x}(1) = \underline{\tilde{x}}(1)$, stop; $\underline{x}^* = \underline{0}$ is the optimal solution to problem (5.19).

Otherwise, continue.

Step 2: Define $N(t_1) = 2$ and $N(t_2) = 3$, where $\mathbb{P}_j(T) = \{t_1,t_2\}$ and j is such that $IRR_j = \max_{\nu\epsilon FV(1)} \{IRR_{\nu}\}$.

Step 3: Subtract Q_j from both FV(2) and FV(3).

Step 4: Define $TR_0 = \{SN_0, SA_0\}$, where $SN_0 = \{1,2,3\}$ and $SA_0 = \{(1,2),(1,3)\}$.

THE BRANCHING RULE (PROCEDURE P9).

Step 1: Define $N(t_1) = r+1$ and $N(t_2) = r+2$, where r is the total number of nodes of TR_{i-1}, $\mathbb{P}_j[N^{-1}(\ell)] = \{t_1,t_2\}$ and ℓ and j are the parameters of A_{i-1}.

Step 2: Subtract Q_j from both FV(r+1) and FV(r+2).

Step 3: Set $SN_i = SN_{i-1} \cup \{r+1, r+2\}$ and $SA_i = SA_{i-1} \cup \{(\ell, r+1), (\ell, r+2)\}$.

Step 4: Define $TR_i = \{SN_i, SA_i\}$

Clearly, Definition 5.4 and procedures P7, P8 and P9 differ from Definition 3.2 and procedures P4, P6 and P3, only in that the complementary constraints are contemplated by the modifications. Since no other essential difference is involved, the algorithm of Section 4.3.3 can be used to solve problem (5.19) with these modifications, provided associated upper bounding functions are available. Naturally, both the Z_ℓ's and the Ω_ℓ's continue to be upper bounding functions in this case, because the feasibility region of problem (5.19) is contained by that of problem (3.5). They can, however, be improved, by intersecting their domains with the complementary constraints. In the first case, it would only be necessary to subtract from $Y(\ell)$ every j, such that $Y(\ell) \cap Q_j \neq \phi$, to obtain the new Z_ℓ*. In the second case, the complementary constraints would have to be added to those of the linear programming problem which solution leads to Ω_ℓ*. Of course, in this case the inspection procedure would not necessarily be valid to solve this problem, because of the possibility of having to deal with negative coefficients in the constraints. In Table 5.1 and Figure 5.1, the results corresponding to the example of Section 3.4 are presented, under the assumption that projects 1 and 2 are complementary. As can be observed, the optimal solution in this case is $\underline{x}^* = (0,0,1,0,1,0)$. Project 2, although by itself very promising, now had to be rejected as a consequence of both its connection with project 1 and constraint (3.3). In fact, it can be seen that without this constraint the optimal selection would include these two projects, and that the overall NPV, despite the presence of an individual negative NPV (that of P_1), would be higher (see Table 5.2 and Figure 5.2). In terms of the underlying assumptions, this could only be possible if the availability of funds were not conditional to selections of projects with positive NPV. The previous example shows that this is an alternative which should not be disregarded a priori, when complementary projects are involved. In such cases, funds to invest in an undesirable project could be expected to be available, but only if it were appropriately compensated by a promising complementary project. One way to deal with this problem would be to consider complementary projects as one. If assumptions iii) and iv) of Section 3.2 were satisifed, the form of the problem would coincide with that of problem (3.5). Otherwise, modifying procedure P7 so as to ensure that, if $x_j = 1$, then

$\sum_{\nu \in Q_j \cup \{j\}} NPV_\nu(t) > 0$, would also lead to the required solution.

5.3.4) SUPPLEMENTARY PROJECTS.

DEFINITION 5.5: Let P_j and P_k be any two different projects, such that P_k can only be accepted if P_j is accepted, but not vice versa. In this case, P_k will be said to be a SUPPLEMENTARY project pf P_j.

This kind of technological dependence can be dealt with by means of the following inequality:

$$x_j - x_k \geqslant 0 \qquad (5.20)$$

By (5.20), x_k can only be equal to 1, if x_j is equal to 1, but not vice versa. Instead of supposition v) of Section 3.2, let it now be assumed that:

v)" Projects $P_1, P_2, \ldots, P_{n_1}$ are technologically independent. The remaining projects are such that, for any $j \epsilon \{n_1+1, n_2+2, \ldots, n\}$, P_j is a supplementary project of P_k, or P_k is a supplementary project of P_j, for at least one element k of $\{n_1+1, n_1+2, \ldots, n\}$. Acceptances and rejections do not affect the size of the defining cash flows.

Iteration i	ℓ	j	A_i	Concluding Nodes	Rejected Nods	$Z_1^* = 2.573$	
						L_i	U_i
0	2	3	{2}	-	3	1.981	2.573
1	-	-	ϕ	4,5	-	1.981	-

TABLE 5.1 Summarized Results of the Original Example ($x_1 = x_2$ and $NPV_j(t) > 0$ if $x_j = 1$).

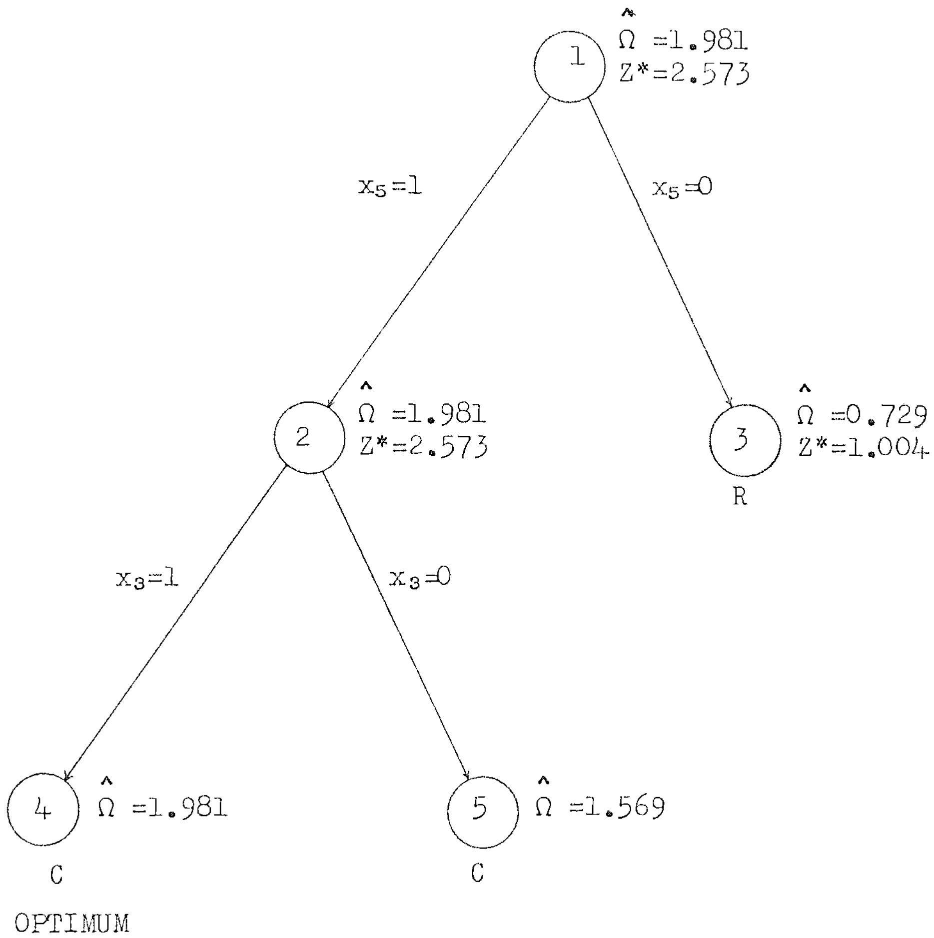

FIGURE 5.1 The Final Directed Tree of the Original Example ($x_1=x_2$ and $NPV_j(t) > 0$ if $x_j = 1$).

Iteration i	ℓ	j	A_i	Concluding Nodes	Rejected Nodes	$Z_1^* = 6.280$	
						L_i	U_i
0	2	3	{2}	–	3	3.361	5.341
1	–	–	ϕ	4,5	–	3.361	–

TABLE 5.2 Summarized Results of the Original Example ($x_1 = x_2$ and $\sum_{\nu \in Q_j \cup \{j\}} NPV_\nu(t) > 0$ if $x_j = 1$).

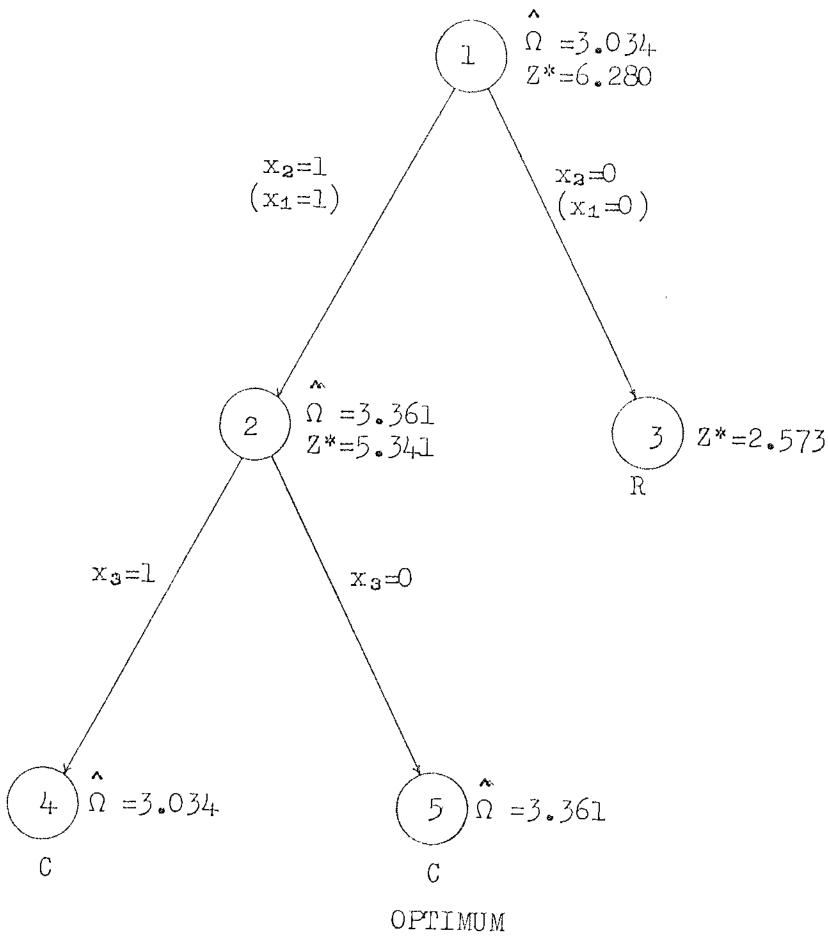

FIGURE 5.2 The Final Directed Tree of the Original Example ($x_1 = x_2$ and $\sum_{\nu \in Q_j \cup \{j\}} NPV_\nu(t) > 0$ if $x_j = 1$).

Again, under all the other assumptions of Section 3.2, the problem may now be formulated as:

$$\left.\begin{aligned} & \text{Max } Z = \sum_{j=1}^{n} NPV_j(t)x_j \\ & \text{s.t.} \quad \sum_{j=1}^{n} a_{jk}x_j \leq b_k, \; k = 0,1,\ldots,M \\ & \qquad x_j - x_\ell \geq 0, \; (j,\ell) \epsilon A_1 \\ & \qquad NPV_j(t) > 0, \text{ if } x_j = 1, \; j = 1,2,\ldots,n \\ & \qquad x_j = 0 \text{ or } 1, \; j = 1,2,\ldots,n, \end{aligned}\right\} \quad (5.21)$$

A_1 being the minimal set of pairs representing all the supplementary relationships amongst the projects.

As in the case of problem (5.19), the form of problem (5.21) does not correspond to that of problem (3.5), but the proposed solution method can be easily adapted. In fact, the required modifications would be very similar to those associated with problem (5.19), the difference being that in this case some elements of $\{P_{n_1+1}, P_{n_1+2}, \ldots, P_n\}$ can be accepted, without having to accept any other element of this set of projects. Taking this point into consideration, the modifications can be introduced along the same lines. On the other hand, if, under appropriate conditions, projects with negative NPV were allowed, the alternative of grouping the projects would have to be complemented. For example, if P_k were a supplementary project of P_j and these two projects were to be considered as one project $P_{j,k}$, then P_k would have to be eliminated; P_j, however, would still have to be considered in order to cover all the feasible combinations, but under a mutually exclusive constraint with respect to $P_{j,k}$.

5.3.5) RE-INVESTMENT ALTERNATIVES.

Technological dependence can be found in a wide variety of ways and combinations. In this connection, three different cases have been considered separately in previous sections, but no specific problem leading to the conditions of dependence was discussed. In this section, this aspect will be illustrated with an example of technologically independent projects admitting alternatives of re-investment.

Let $U = \{P^{(o)}_1, P^{(o)}_2, \ldots, P^{(o)}_n\}$, a set of capital investment projects, be considered under the following assumptions:

i) Each project is indivisible.

ii) Each project $P^{(o)}_j$ is defined by a sequence of cash in-flows $B^{(o)}_{j1}, B^{(o)}_{j2}, \ldots, B^{(o)}_{jm}$ and a sequence of cash out-flows $C^{(o)}_{jo}, C^{(o)}_{j1}, \ldots, C^{(o)}_{jm}$, where $B^{(o)}_{jk}$ $(k \geq 1)$ and $C^{(o)}_{jk}$ $(k \geq 0)$ take place at the beginning of the (k+1)-th time period.

iii) Associated with each project $P^{(o)}_j$ is an alternative of re-investment, only available if $P^{(o)}_j$ is accepted. If this alternative is considered at the end of the i-th time period, a sequence of cash in-flows $B^{(i)}_{j(i+1)}, B^{(i)}_{j(i+2)}, \ldots, B^{(i)}_{jm}$ and a sequence of cash out-flows $C^{(i)}_{ji}$, $C^{(i)}_{j(i+1)}, \ldots, C^{(i)}_{jm}$ are generated independently of $P^{(o)}_j$, where $B^{(i)}_{jk}$ $(k \geq i+1)$ and $C^{(i)}_{jk}$ $(k \geq i)$ take place at the beginning of the (k+1)-th time period. These cash flows will be referred to as project $P^{(i)}_j$.

iv) For any project $P^{(i)}_j$, a natural number $M^{(i)}_j$ $(< m)$ exsits, such that $B^{(i)}_{ik} < C^{(i)}_{jk}$, if $k \leq M^{(i)}_j$ $(B^{(i)}_{ji} = 0)$, and $B^{(i)}_{jk} > C^{(i)}_{jk}$, if $k > M^{(i)}_j$. For any $k = i+1, i+2, \ldots, M^{(i)}_j$, the operating requirement $C^{(i)}_{jk} - B^{(i)}_{jk}$ is relatively small in terms of the investment (or re-investment) requirement $C^{(i)}_{ji}$.

v) For any project $P_j^{(o)}$, the re-investment alternative may be considered (only once) at the end of any time period i, for which $M_j^{(o)} < i \leq m_j < m$, where m_j is a natural number. Clearly, this is a convention which could be changed according to the circumstances, without altering the basic structure of the problem.

vi) For any project $P_j^{(i)}$, $C_{ji}^{(i)} - \sum_{k=i}^{M_j^{(i)}} (B_{jk}^{(i)} - C_{jk}^{(i)}) < \sum_{k=M_j^{(i)}+1}^{m} (B_{jk}^{(i)} - C_{jk}^{(i)})$. Again, this assumption, together with iv), means that the internal rate of return $IRR_j^{(i)}$ is well defined for each project $P_j^{(i)}$.

vii) Any set of the form $\{P_{j_1}^{(i_1)}, P_{j_2}^{(i_2)}, \ldots, P_{j_\ell}^{(i_\ell)}\}$, where $j_1 \neq j_2 \neq \ldots \neq j_\ell$, is a set of technologically independent projects. Additionally, acceptances and rejections do not affect the size of the defining cash flows.

viii) Any final section should not result in an overall requirement surpassing the limit b_k at the end of the k-th time period $(k = 0,1,\ldots,M = \max_{i,j} \{M_j^{(i)}\})$, re-investment alternatives being included. Once again, overall requirements are not supposed to be compensated by positive net cash flows taking place at the time point under consideration.

ix) Capital funds for an overall investment or re-investment requirement of t_i at the beginning of the (i+1)-th time period $(i \geq 0)$ are available at the cost $c_i(t_i)$, only if projects with positive NPV are accepted. The c_i's are assumed to be positive non-decreasing functions of the t_i's, and operating requirements are obtainable at the cost of the corresponding investment requirements.

Let

$$a_{ijk} = \begin{cases} C^{(i)}_{jk} - B^{(i)}_{jk}, & \text{if } k \leq M^{(i)}_j \\ 0, & \text{otherwise} \end{cases}$$

$$k = 0,1,\ldots,M$$

and

$$x_{ij} = \begin{cases} 1, & \text{if } P^{(i)}_j \text{ is accepted} \\ 0, & \text{otherwise} \end{cases}$$

Taking both U and the re-investment alternatives into account, the problem can now be stated as:

$$\left.\begin{aligned}
&\text{Max } Z = \sum_{j=1}^{n} \sum_{i=M^{(o)}_j+1}^{m_j} NPV_{ij}(\underline{t}_i)x_{ij} && (5.22)\\
&\text{s.t.} \quad \sum_{j=1}^{n} \sum_{i=M^{(o)}_j+1}^{m_j} a_{ijk}x_{ij} \leq b_k, k=0,1,\ldots,M && (5.23)\\
&x_{oj}-x_{ij} \geq 0, \quad \begin{cases} j=1,2,\ldots,n, \\ i=M^{(o)}_j+1, M^{(o)}_j+2,\ldots,m_j \end{cases} && (5.24)\\
&\sum_{i=M^{(o)}_j+1}^{m_j} x_{ij} \leq 1, \quad j = 1,2,\ldots,n && (5.25)\\
&NPV_{ij}(\underline{t}_i) > 0, \text{if } x_{ij}= 1, \begin{cases} j=1,2,\ldots,n, \\ i=M^{(o)}_j+1, M^{(o)}_j+2,\ldots,m_j \end{cases} && (5.26)\\
&x_{ij}=0 \text{ or } 1, \begin{cases} j=1,2,\ldots,n \\ i=M^{(o)}_j+1, M^{(o)}_j+2,\ldots,m_j, \end{cases} && (5.27)
\end{aligned}\right\} (5.28)$$

where $NPV_{ij}(\underline{t}_i) = \dfrac{1}{[1+c_0(t_0)]^i} \sum_{k=i}^{m} \dfrac{B^{(i)}_{jk} - C^{(i)}_{jk}}{[1+c_i(t_i)]^{k-i}}$, $\underline{t}_i = (t_0, t_i)$,

$t_0 = \sum_{j=1}^{n} a_{ojo}x_{oj}$ and $t_i = \sum_{j=1}^{n} a_{iji}x_{ij}$.

The following explanatory relations regarding both the structure of problem (5.28) and the proposed solution method can be observed:

1. The elements of U, together with the associated projects of re-

1. contd.

investments, constitute the universe of investment proposals. In this sense, (5.22), (5.23), (5.26) and (5.27) are the extended versions of (3.1), (3.2), (3.3) and (3.4), respectively. In particular, the form of the objective function in (5.22) can be explained as follows. If $i = 0$, then $NPV_{ij}(\underline{t}_i)$ is obviously a simple extension of $NPV_j(t)$. Otherwise, $NPV_{ij}(\underline{t}_i)$ corresponds to a project of re-investment. By assumptions iii) and ix), the NPV of $P_j^{(i)}$ at the end of the i-th time period is given by

$$IV_{ij} = \sum_{k=i}^{m} \frac{B_{jk}^{(i)} - C_{jk}^{(i)}}{[1+c_i(t_i)]^{k-i}} .$$ In order to know the NPV of IV_{ij} at the beginning of the first time period, IV_{ij} can be thought of as a cash flow which can only be obtained, if project $P_j^{(o)}$ is accepted. Since the incurred cost of capital for $P_j^{(o)}$ is $c_o(t_o)$, the NPV in question is $\frac{IV_{ij}}{[1+c_o(t_o)]^i}$.

2. By assumption iii), $P_j^{(i)}$ is a supplementary project of $P_j^{(o)}$, for any j and any $i > 0$. Consequently, in accordance with (5.20), constraint (5.24) has to be included. On the other hand, by assumption v), the re-investment alternative for any project $P_j^{(o)}$ can only be considered once for implementation. Hence, projects of re-investment corresponding to $P_j^{(o)}$ have to be treated as mutually exclusive projects. This means that constraint (5.25) also has to be included.

3. Having indicated how the solution method can be adapted to deal with constraints (5.24) and (5.25) (see sections 5.3.2-5.3.4), further modifications arising from the fact that projects with different starting time points are involved have to be introduced. In the first place, a unique cost of capital will no longer be associated with the specifications of the variables. However, the relationship between projects starting at a given

3. contd.

time point i and the cost of capital $c_i(t_i)$ is exactly the same as that of projects $P_1, P_2, \ldots, P_n$ and $c(t)$ in Section 3.2. Indeed, IV_{ij} will only be positive if $IRR_j^{(i)} > c_i(t_i)$ (see Appendix CHIII),

and, since $\frac{1}{[1+c_0(t_0)]^i} > 0$ regardless of the value of $c_0(t_0)$,

the same holds for $NPV_{ij}(\underline{t}_i) = \frac{IV_{ij}}{[1+c_0(t_0)]^i}$. Therefore, bearing in mind that internal rates of return have to be compared with the cost of capital incurred at the starting point of the projects, tests of exactly the same character as before can be performed in this case. This follows both from the fact that the c_i's are positive non-decreasing functions of the t_i's, and from the fact that $\frac{1}{(1+y_1)^i} \sum_{k=i}^{m} \frac{B_{jk}^{(i)} - C_{jk}^{(i)}}{(1+y_2)^{k-i}}$ is decreasing with respect to positive values of y_1 and y_2, as can easily be verified (see Appendix CHIII). Finally, $Z_\ell(\underline{x})$ and $\Omega_\ell(\underline{x})$ would now be given by the expression $\sum_{j=1}^{n} \sum_{i=M_j^{(o)}+1}^{m_j} \frac{1}{[1+r_0(\ell)]^i} \sum_{k=i}^{m} \frac{B_{jk}^{(i)} - C_{jk}^{(i)}}{[1+r_i(\ell)]^{k-i}} x_{ij}$,

where $r_i(\ell)$ is the cost of capital at the beginning of the (i+1)-th time period associated with the ℓ-th specification of $\underline{x}$ ($r_i(\ell) = \min_{t_i > o} \{c_i(t_i)\}$, if the investment or re-investment requirement is zero), this time defined as:

$$x_{ij}(\ell) = \begin{cases} x_{ij}, \text{ if } x_{ij} \text{ is non-free at } \ell \\ x_{pj}, \text{ if } x_{pj} \text{ is non-free at } \ell, \\ 0, \text{ otherwise} \end{cases}$$

The modified versions of procedures P4, P6 and P3 for this example are next presented.

CALCULATION OF THE ALTERNATIVE AUXILIARY SOLUTION (PROCEDURE P10).

Step 1: Set $\underline{\tilde{x}}(\ell) = \underline{x}(\ell)$

Step 2: If $FV(\ell) = \phi$, stop; $\underline{\tilde{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 3: Select an element ij of $FV(\ell)$ with maximal $IRR_j^{(i)}$ (suggested tie breaking rule: maximal investment)

Step 4: If $IRR_j^{(i)} \leq r_i(\ell)$, subtract from $FV(\ell)$ all the elements of the form iq (including ij). If, in addition, $i = 0$, subtract from $FV(\ell)$ all the elements of the form pj.

Go to step 2.

If $i \neq 0$, $oj \in FV(\ell)$ and $IRR_j^{(o)} \leq r_o(\ell)$, subtract from $FV(\ell)$ oj and ij. Go to step 2.

Otherwise, continue.

Step 5: If $IRR_j^{(i)} \leq c(\sum_{\nu=1}^{n} a_{i\nu i}\, \tilde{x}_{i\nu}(\ell) + a_{iji})$, set $I = 1$ and go to step 7.

If $i \neq 0$, $oj \in FV(\ell)$ and $IRR_j^{(o)} \leq c(\sum_{\nu=1}^{n} a_{o\nu o}\tilde{x}_{o\nu}(\ell) + a_{ojo})$, set $I = 0$ and go to step 7.

Otherwise, continue.

Step 6: If $\sum_{\nu=1}^{n} \sum_{\mu=M_\nu^{(o)}+1}^{m_\nu} a_{\mu\nu k}\tilde{x}_{\mu\nu}(\ell) + a_{ijk} > b_k$, for some $k = 0,1,\ldots,M$, set $I = 1$ and go to step 7.

If $i \neq 0$, $oj \in FV(\ell)$ and $\sum_{\nu=1}^{n} \sum_{\mu=M_\nu^{(o)}+1}^{m_\nu} a_{\mu\nu k}\tilde{x}_{\mu\nu}(\ell) + a_{ojk} + a_{ijk} > b_k$, for some $k = 0,1,\ldots,M$, set $I = 1$ and go to step 7.

Otherwise, set $\widetilde{FV}(\ell) = FV(\ell)$ and $\tilde{x}_{ij}(\ell) = 1$. Subtract ij from $\widetilde{FV}(\ell)$. If $i \neq 0$ and $oj \in FV(\ell)$, set $\tilde{x}_{oj}(\ell) = 1$ and subtract oj from $\widetilde{FV}(\ell)$. If $i \neq 0$, subtract from $\widetilde{FV}(\ell)$ all the elements of the form pj. Go to step 8.

Step 7: Subtract ij from $\widetilde{FV}(\ell)$.

If i = o, subtract from $\widetilde{FV}(\ell)$ all the elements of the form pj.

If I = 0, subtract oj from $\widetilde{FV}(\ell)$.

Go to step 2.

Step 8: If $\widetilde{FV}(\ell) = \phi$, stop; $\underline{\tilde{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 9: Select an element ij of $\widetilde{FV}(\ell)$ with maximal $IRR_j^{(i)}$ (suggested the breaking rule: maximal investment).

Step 10: If $IRR_j^{(i)} \leqslant c[\sum_{\nu=1}^{n} a_{i\nu i}\tilde{x}_{i\nu}(\ell)]$, subtract from $\widetilde{FV}(\ell)$ all the elements of the form iq (including ij). If, in addition, i=o, subtract from $\widetilde{FV}(\ell)$ all the elements of the form pj. Go to step 8.

If $i \neq o, oj \epsilon FV(\ell)$ and $IRR_j^{(o)} \leqslant c[\sum_{\nu=1}^{n} a_{o\nu o}\tilde{x}_{o\nu}(\ell)]$, subtract from $\widetilde{FV}(\ell)$ oj and ij. Go to step 8.

Otherwise, continue.

Step 11: If $IRR_j^{(i)} \leqslant c(\sum_{\nu=1}^{n} a_{i\nu i}\tilde{x}_{i\nu}(\ell)+a_{iji})$, set I = 1 and go to step 13.

If $i \neq o$, $oj \epsilon FV(\ell)$ and $IRR_j^{(o)} \leqslant c(\sum_{\nu=1}^{n} a_{o\nu o}\tilde{x}_{o\nu}(\ell)+a_{ojo})$, set I = 0 and go to step 13.

Otherwise, continue.

Step 12: If $\sum_{\nu=1}^{n} \sum_{\mu=M_\nu^{(o)}+1}^{m_\nu} a_{\mu\nu k}\tilde{x}_{\mu\nu}(\ell)+a_{ijk} > b_k$, for some k = 0,1,...,M, set I = 1 and go to step 13.

If $i \neq o$, $oj \epsilon FV(\ell)$ and $\sum_{\nu=1}^{n} \sum_{\mu=M_\nu^{(o)}+1}^{m_\nu} a_{\mu\nu k}\tilde{x}_{\mu\nu}(\ell)+a_{ojk}+a_{ijk} > b_k$, for some k = 0,1,...,M, set I = 1 and go to step 13.

Otherwise, set $\tilde{x}_{ij}(\ell) = 1$. Subtract ij from $\widetilde{FV}(\ell)$. If $i \neq o$ and $oj \epsilon FV(\ell)$, set $\tilde{x}_{oj}(\ell) = 1$ and subtract oj from

Step 12: contd.

$\widetilde{FV}(\ell)$. If $i \neq o$, subtract from $\widetilde{FV}(\ell)$ all the elements of the form pj. Go to step 8.

Step 13: Subtract ij from $\widetilde{FV}(\ell)$.

If $i = o$, subtract from $\widetilde{FV}(\ell)$ all the elements of the form pj.

If $I = 0$, subtract oj from $\widetilde{FV}(\ell)$.

Go to step 8.

CONSTRUCTION OF THE INITIAL TREE (PROCEDURE P11).

Step 1: Define $N(T) = 1$ and obtain the associated concepts of node 1.

If $\underline{x}(1) = \tilde{\underline{x}}(1)$, stop; $\underline{x}^* = \underline{o}$ is the optimal solution to problem (5.28).

Otherwise, continue.

Step 2: Define $N(t_1) = 2$ and $N(t_2) = 3$, where $\mathbb{P}_{ij}\{T\} = \{t_1, t_2\}$ and ij is such that $IRR_j^{(i)} = \max\limits_{\nu\mu = FV(1)} \{IRR_\nu^{(\mu)}\}$.

Step 3: If $i \neq o$, subtract from FV(2) all the elements of the form pj.

If $i = o$, subtract from FV(3) all the elements of the form pj.

Step 4: Define $TR_o = \{SN_o, SA_o\}$, where $SN_o = \{1,2,3\}$ and $SA_o = \{(1,2),(1,3)\}$.

THE BRANCHING RULE (PROCEDURE P12).

Step 1: Define $N(t_1) = r+1$ and $N(t_2) = r+2$, where r is the total number of nodes of TR_{i-1}, $\mathbb{P}_{pj}\{N^{-1}(\ell)\} = \{t_1, t_2\}$ and ℓ and pj are the parameters of $\bar{A}_{i-1}$.

Step 2: If $p \neq o$ subtract from FV(r+1) all the elements of the form sj.

If $p = o$, subtract from FV(r+2) all the elements of the form sj.

Step 3: Set $SN_i = SN_{i-1} \cup \{r+1, r+2\}$ and $SA_i = SA_{i-1} \cup \{(\ell, r+1), (\ell, r+2)\}$.

Step 4: Define $TR_i = \{SN_i, SA_i\}$.

It can be easily checked that the modifications introduced in procedures P10-P12 and based on the relation between the internal rates of return and the individual net present values, taking into account that: a) If a re-investment project $P_j^{(i)}$ is accepted, then $P_j^{(o)}$ has to be accepted and all the other re-investment projects associated with $P_j^{(o)}$ have to be rejected; and b) If a project $P_j^{(o)}$ is rejected, then all the re-investment projects associated with $P_j^{(o)}$ have to be rejected. Clearly, this is equivalent to deal with constraints (5.24) and (5.25). Replacing P4, P6 and P3 by P10,P11 and P12,respectively, the algorithm of Section 4.3.3 can be used to solve problem (5.28). Procedure P5, with the only difference of the numeration of variables (ij instead of j), is essentially the same, and, as in the case of complementary projects, the upper bounding functions as originally defined (constraints (5.24) and (5.25) not being considered) can be improved by intersecting their domains with (5.24) and (5.25). In Table 5.3 and Figure 5.3 the summarized results and the final directed tree for the example of Section 3.4 is presented, under the assumption that, either at the end of period 2 or at the end of period 3, re-investment can take place. Due to constraint (5.26), projects $P_1^{(o)}, P_4^{(o)}$ and $P_6^{(o)}$ (P_1, P_4 and P_6 in the notation of Section 3.4) were directly eliminated, and the net cash flows of re-investment were considered as follows: -1 at the beginning and 0.4 thereafter for $P_2^{(o)}$ ($IRR_2^{(2)} = 0.351$ and $IRR_2^{(3)} = 0.327$), -3 at the beginning and 0.7 thereafter for $P_3^{(o)}$ ($IRR_3^{(2)} = 0.140$ and $IRR_3^{(3)} = 0.106$), and -4 at the beginning and 1 thereafter for

$P_5^{(0)}$ ($IRR_5^{(2)} = 0.163$ and $IRR_5^{(3)} = 0.130$). It was further assumed that the cost of capital for re-investment is 0.142 for requirements above 4, and 0.136 otherwise ($x_{33} = 0$), and that $b_2 = b_3 = 5$.

5.4) SUMMARY.

In reference to the solution scheme under study, problems involving financing decisions and technological dependence were considered in this chapter. In the first case, the problem was formulated in terms of capital suppliers offering funds at different competitive costs, thus defining two types of decisions: those associated with the financing of the projects, and those associated with the selection of the projects. It was shown that, although linked, these decisions can be dealt with separately under the considered assumptions, the minimal cost of capital being the criterion to handle the financing decisions. This was seen to be a separable programming problem, which solution allows the direct applicability of the optimal selection procedures presented in previous chapters. As for the problems including technologically dependent projects, these procedures were extended to solve them, on the basis of both the constraints of dependence and the general supporting framework of the procedures. Mutually exclusive, complementary and supplementary projects, and an example of re-investment alternatives were discussed.

Iteration i	ℓ	ij	A_i	Concluding Nodes	Rejected Nodes	$\underline{x}(1) \neq \tilde{\underline{x}}(1)$	
						L_i	U_i
0	2	22	{2}	-	3	5.247	7.173
1	4	05	{4,5}	-	-	5.247	6.600
2	5	32	{5,6}	-	7	5.398	6.292
3	8	05	{6,8,9}	-	-	5.398	6.292
4	6	25	{6,9,10}	-	11	5.398	6.284
5	12	03	{9,10,12,13}	-	-	5.578	6.155
6	10	25	{9,10,13}	14,15	14	5.578	6.104
7	16	03	{9,13,16,17}	-	-	5.578	6.104
8	13	03	{9,13,17}	18,19	18,19	5.578	6.018
9	9	05	{9,17}	21	20,21	5.578	5.891
10	17	03	{17,22}	-	23	5.578	5.838
11	22	25	{22}	25	24,25	5.578	5.710
12	26	03	{26}	-	27	5.578	5.710
13	-	-	ϕ	28,29	28,29	5.578	-

TABLE 5.3 Summarized Results of the Original Example with Re-Investment Alternatives.

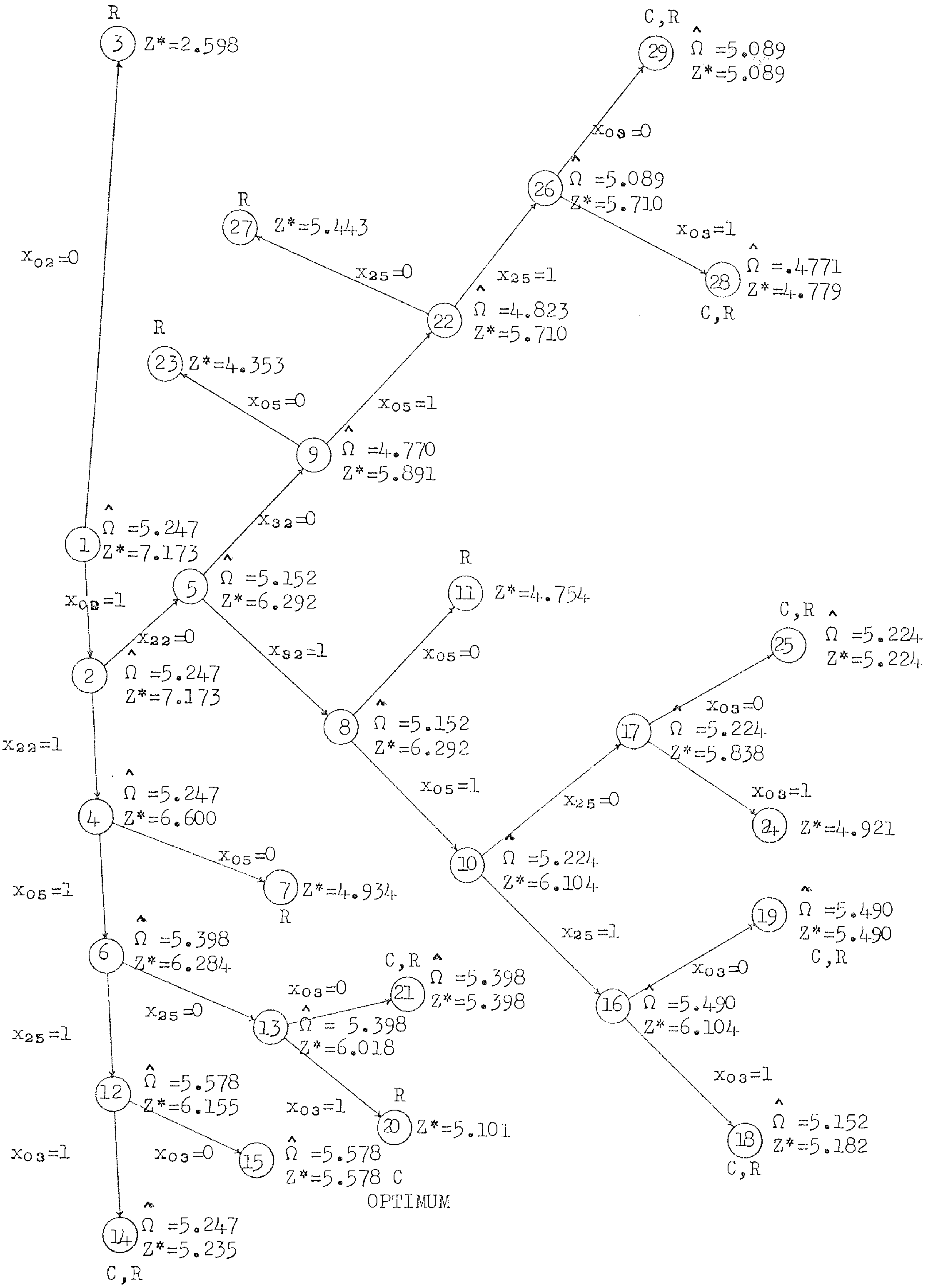

FIGURE 5.3 The Final Directed Tree for the Original Example with Re-Investment Alternatives.

REFERENCES.

1 A.CHARNES and C.LEMKE (1954) Minimization of nonlinear separable convex functionals. Naval Research Logistics Quarterly 1, 301-312.

2 C.MILLER (1963) The simplex method for local separable programming. Recent Advances in Mathematical Programming, R.GRAVES and P.WOLFE, editors. MacGraw-Hill, New York.

3 R.E.GRIFFITH and R.A.STEWART (1961) A nonlinear programming technique for the optimization of continuous processing systems. Mgmt.Sci.7, 379-392.

4 A.V.FIACCO and C.P.McCORMICK (1968) Nonlinear Programming. SUMT. John Wiley & Sons, 1968.

5 R.BELLMAN (1954) Some applications of the theory of Dynamic Programming. Ops.Res.2, 275-288.

6 G.HADLEY (1964) Nonlinear and Dynamic Programming. Addison-Wesley, Reading, Massachusetts.

7 E.SOLOMON (1963) The Theory of Financial Management. Columbia University Press, New York.

CHAPTER VI

A GENERALIZED ZERO-ONE PROGRAMMING MODEL.

6.1) INTRODUCTION.

The general solution scheme developed in sections 2.2 and 2.3 was applied in the preceding chapters to a class of problems arising in the field of capital investment, taking advantage of some of their structural properties. In this chapter, the application of the scheme is extended to problems which, sharing these properties, are suitable to be dealt with in a similar fashion.

6.2) THE PROBLEM.

Let the following zero-one programming problem be considered:

$$\left.\begin{aligned} &\text{Max } Z = \sum_{j=1}^{n} f_j[h(\underline{t})]x_j && (6.1) \\ &\text{s.t. } g_i(\underline{x}) \leq b_i,\ i = 1,2,\ldots,m && (6.2) \\ &\qquad x_j = 0 \text{ or } 1,\ j = 1,2,\ldots,n && (6.3) \end{aligned}\right\} \quad (6.4)$$

where $\underline{t} = (t_1,t_2,\ldots,t_m)$, $t_i = g_i(\underline{x})$, $\underline{x} = (x_1,x_2,\ldots,x_n)$ and:

i) $f_j: \mathbb{R} \to \mathbb{R}$ is decreasing and $f_j(s_j) = 0$ for some $s_j \in \mathbb{R}$ $(j = 1,2,\ldots,n)$; and

ii) $h: E^m \to \mathbb{R}$ and $g_i: E^n \to \mathbb{R}$ $(i = 1,2,\ldots,m)$ are non-decreasing with respect to each of its variables.

LEMMA 6.1: For any $j = 1,2,\ldots,n$, if $f_j(y) = 0$, then $y = s_j$.

Proof: Since f_j is decreasing, either $f_j(y) < f_j(s_j)$ or $f_j(y) > f_j(s_j)$, if $y \neq s_j$. Therefore, $f_j(y) = 0$ only if $y = s_j$.

COROLLARY 6.1: For any $j = 1,2,\ldots,n$:

$$f_j(y) \begin{cases} > 0 \ , \text{ if } y < s_j \\ = 0 \ , \text{ if } y = s_j \\ < 0 \ , \text{ if } y > s_j \end{cases}$$

Proof: This result is a direct consequence of assumption i) and Lemma 6.1.

LEMMA 6.2: If $g_i(\underline{0}) > b_i$, for some $i = 1,2,\ldots,m$, then no feasible solution to problem (6.4) exists.

Proof: By assumption ii), for any $\underline{x}$ satisfying constraint (6.3), $g_i(\underline{x}) \geq g_i(\underline{0})$ $(i = 1,2,\ldots,m)$. Hence, if $g_i(\underline{0}) > b_i$, then $g_i(\underline{x}) > b_i$. This means that no solution satisfying constraint (6.3) can satisfy constraint (6.2).

In view of Lemma 6.2, it will further be assumed that $g_i(\underline{0}) \leq b_i$ $(i = 1,2,\ldots,m)$, which is the case of interest.

Let S, T, Y and f be defined as follows:

$$S = \{\underline{x} \in E^n / (6.2) \text{ and } (6.3) \text{ are satisfied}\},$$

$$T = \{\underline{x} \in E^n / (6.3) \text{ is satisfied}\},$$

$$Y = \{\underline{x} \in E^n / 0 \leq x_j \leq 1,\ j = 1,2,\ldots,n\}, \text{ and}$$

$$f: Y \to \mathbb{R}, \text{ where } \underline{x} \in Y \Longrightarrow f(\underline{x}) = \sum_{j=1}^{n} [h(\underline{t})] x_j$$

As can easily be checked, the form of problem (6.4) corresponds to that of problem (2.1) (see Section 2.2).

6.3) THE DIRECTED TREE.

It will be assumed that $\mathbb{P}_j$, N, $\underline{x}(\ell)$, $\hat{Z}_\ell$, $FV(\ell)$ and Y_ℓ are defined as in Section 3.3.2, and that:

$$r(\ell) = \begin{cases} h[\underline{t}(\ell)], & \text{if } \underline{x}(\ell) \neq \underline{0} \\ \min\limits_{\substack{\underline{x} \in T \\ \underline{x} \neq \underline{0}}} \{h(\underline{t})\}, & \text{otherwise,} \end{cases}$$

for any node ℓ,

where:

$$\underline{t}(\ell) = [t_1(\ell), t_2(\ell), \ldots, t_m(\ell)], \text{ and}$$

$$t_i(\ell) = g_i[\underline{x}(\ell)],\ i = 1,2,\ldots,m.$$

LEMMA 6.3: Let $\overline{Z}_\ell : N^{-1}(\ell) \to \mathbb{R}$ be such that $\underline{x} \in N^{-1}(\ell) \Longrightarrow$ $\overline{Z}_\ell(\underline{x}) = \sum_{j=1}^{n} f_j[r(\ell)] x_j$, where ℓ is any node.

Then $\overline{Z}_\ell(\underline{x}) \geq f(\underline{x})$, for all $\underline{x} \in N^{-1}(\ell)$.

Proof: If $x_j(\ell) = 1$ and $\underline{x}\epsilon N^{-1}(\ell)$, then $x_j = 1$, for any $j = 1,2,\ldots,n$. Thus, since the g_i's and h are non-decreasing, $h[g(\underline{x})] = h(\underline{t}) \geq r(\ell)$, for any $\underline{x}\epsilon N^{-1}(\ell)$. The f_j's, on the other hand, are decreasing, and so:

$$f_j[h(\underline{t})] \leq f_j[r(\ell)]\ ,\ j = 1,2,\ldots,n,$$

$$\therefore \quad f(\underline{x}) = \sum_{j=1}^{n} f_j[h(\underline{t})]x_j \leq \sum_{j=1}^{n} f_j[r(\ell)] = \overline{Z}_\ell(\underline{x}),$$

for any $\underline{x}\epsilon N^{-1}(\ell)$.

COROLLARY 6.2: If ℓ is a feasible node, then $\overline{Z}_\ell$ is an upper bounding function of ℓ.

Proof: Since, for any node ℓ, $N^{-1}(\ell)$ is finite, $\overline{Z}_\ell$ is bounded and $\overline{Z}_\ell^* = \max\limits_{\underline{x}\epsilon N^{-1}(\ell)} \{\overline{Z}_\ell(\underline{x})\}$ exists. Therefore, by Lemma 6.3, the required result follows.

COROLLARY 6.3: If ℓ is a feasible node and $w: D \rightarrow \mathbb{R}$ is such that $S_\ell \subset D \subset N^{-1}(\ell)$ and $w(\underline{x}) = \overline{Z}_\ell(\underline{x})$, for all $\underline{x}\epsilon D$, then w is an upper bounding function of ℓ.

Proof: Again, D is finite. Hence, since $w(\underline{x}) = \overline{Z}_\ell(\underline{x})\ \forall\ \underline{x}\epsilon D$, the required result follows from Lemma 6.3.

COROLLARY 6.4: If ℓ is a feasible node and $Z_\ell : Y_\ell \rightarrow \mathbb{R}$ is such that $Z_\ell(\underline{x}) = \overline{Z}_\ell(\underline{x})$, for all $\underline{x}\epsilon Y_\ell$, then Z_ℓ is an upper bounding function of ℓ.

Proof: Because $S_\ell \subset Y_\ell \subset N^{-1}(\ell)$, this result follows from Corollary 6.3.

LEMMA 6.4: Let $Y(\ell)$ be a subset of $\{1,2,\ldots,n\}$, such that $j\epsilon Y(\ell)$, if, and only if, either $x_j(\ell) = 1$, or $j\epsilon FV(\ell)$ and $s_j > r(\ell)$. Then, for any node ℓ:

$$Z_\ell^* = \max_{\underline{x}\epsilon Y_\ell} \{Z_\ell(\underline{x})\} = \begin{cases} \sum\limits_{j\epsilon Y(\ell)} f_j[r(\ell)], & \text{if } Y(\ell) \neq \phi \\ \hat{Z}_\ell, & \text{otherwise} \end{cases}$$

Proof: See proof of Lemma 3.2 in Section 3.3.2 ($a_j = f_j[r(\ell)]$ and s_j replacing IRR_j).

LEMMA 6.5: Let $\Omega_\ell : Y_\ell' \to \mathbb{R}$ be such that $\underline{x} \in Y_\ell' \implies$

$$\Omega_\ell(\underline{x}) = \sum_{j=1}^{n} f_j[r(\ell)]x_j, \text{ where } Y_\ell' = W \cap V_\ell, \text{ and}$$

$W = \{\underline{x} \in E^n / (6.2) \text{ is satisfied}\}$ (V_ℓ defined as in Section 4.2.3). If $Y_\ell \neq \phi$, then $\Omega_\ell(\underline{x}) \geq f(\underline{x})$, for all $\underline{x} \in Y_\ell'$.

Proof: Again, if $x_j(\ell) = 1$ and $\underline{x} \in Y_\ell'$, then $x_j = 1$, for any $j = 1,2,\ldots,n$. Thus (see proof of Lemma 6.3),

$$f_j[h(\underline{t})] \leq f_j[r(\ell)], \quad j = 1,2,\ldots,n,$$

$$\therefore \quad f(\underline{x}) = \sum_{j=1}^{n} f_j[h(\underline{t})]x_j \leq \sum_{j=1}^{n} f_j[r(\ell)]x_j = \Omega_\ell(\underline{x}).$$

COROLLARY 6.5: If ℓ is a feasible node, then Ω_ℓ is an upper bounding function of ℓ.

Proof: Y_ℓ' is a bounded region, and so $\Omega_\ell^* = \max_{\underline{x} \in Y_\ell'} \{\Omega_\ell(\underline{x})\}$ exists. Therefore, by Lemma 6.5, Ω_ℓ is an upper bounding function of ℓ.

LEMMA 6.6: If ℓ is any feasible node, then $\Omega_\ell^* \leq Z_\ell^*$.

Proof: See proof of Lemma 4.2 in Section 4.2.3 ($f_j[r(\ell)]$ and s_j replacing $\sum_{k=o}^{m} \frac{B_{jk} - C_{jk}}{[1+r(\ell)]^k}$ and IRR_j, respectively).

LEMMA 6.7: $c_1 < c_2 \implies \sum_{j=1}^{n} f_j(c_1)x_j \geq \sum_{j=1}^{n} f_j(c_2)x_j$, for any $x_j \geq 0$ $(j = 1,2,\ldots,n)$.

Proof: Since f_j is decreasing, $f_j(c_1) > f_j(c_2)$ for any $j = 1,2,\ldots,n$. Hence, the required result follows.

COROLLARY 6.6: If (ℓ,k) is a directed arc, then $Z_\ell^* \geq Z_k^*$ and $\Omega_\ell^* \geq \Omega_k^*$.

Proof: Since $r(\ell) \leq r(k)$ for any directed arc (ℓ,k),

$$\sum_{j=1}^{n} f_j[r(\ell)]x_j \geq \sum_{j=1}^{n} f_j[r(k)]x_j, \text{ where } x_j \geq 0$$

(by Lemma 6.7). On the other hand, $Y_k \subset Y_\ell$ and $Y_k' \subset Y_\ell'$. Therefore, $\Omega_\ell^* \geq \Omega_k^*$.

The re-formulation of fundamental concepts and properties associated with the proposed solution method and problem (3.5) (see Section 3.2) is now complete in connection with problem (6.4). The remaining structure is essentially the same in this case, and it can be justified on exactly the same grounds as before. In relation to the improved lower bounds approach (see Section 4.3.2) the version of procedure P4 would hence be the following:

CALCULATION OF THE ALTERNATIVE AUXILIARY SOLUTION (PROCEDURE P13).

Step 1: Set $\underset{\sim}{\tilde{x}}(\ell) = \underline{x}(\ell)$

Step 2: If $FV(\ell) = \phi$, stop; $\underline{\tilde{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 3: Select an element j of $FV(\ell)$ with maximal s_j.

Step 4: If $s_j \leq r(\ell)$, set $FV(\ell) = \phi$ and stop; $\underline{\tilde{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 5: If $s_j \leq h(\underline{t})$, where

$t_i = g_i[\tilde{x}_1(\ell),\tilde{x}_2(\ell),\ldots,\tilde{x}_{j-1}(\ell),1,\tilde{x}_{j+1}(\ell),\ldots,\tilde{x}_n(\ell)]$,

for $i = 1,2,\ldots,m$, go to step 7.

Otherwise, continue.

Step 6: If $g_i[\tilde{x}_1(\ell),\tilde{x}_2(\ell),\ldots,\tilde{x}_{j-1}(\ell),1,\tilde{x}_{j+1}(\ell),\ldots,\tilde{x}_n(\ell)] \leq b_i$, for $i = 1,2,\ldots,m$, set $\tilde{x}_j(\ell) = 1$ and $\widetilde{FV}(\ell) = FV(\ell) - \{j\}$, and go to step 8.

Otherwise, continue.

Step 7: Subtract $\{j\}$ from $FV(\ell)$ and go to step 2.

Step 8: If $\widetilde{FV}(\ell) = \phi$, stop; $\underline{\tilde{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 9: Select an element j of $\widetilde{FV}(\ell)$ with maximal s_j.

Step 10: If $s_j \leqslant h(\underline{t})$, where $t_i = g_i[\underline{\tilde{x}}(\ell)]$, for $i = 1,2,\ldots,m$, stop; $\underline{\tilde{x}}(\ell)$ is the alternative auxiliary solution.

Otherwise, continue.

Step 11: If $s_j \leqslant h(\underline{t})$, where

$t_i = g_i[\tilde{x}_1(\ell), \tilde{x}_2(\ell),\ldots,\tilde{x}_{j-1}(\ell),1,\tilde{x}_{j+1}(\ell),\ldots,\tilde{x}_n(\ell)]$,

for $i = 1,2,\ldots,m$, go to step 13.

Otherwise, continue.

Step 12: If $g_i[\tilde{x}_1(\ell),\tilde{x}_2(\ell),\ldots,\tilde{x}_{j-1}(\ell),1,\tilde{x}_{j+1}(\ell),\ldots,\tilde{x}_n(\ell)] \leqslant b_i$ for $i = 1,2,\ldots,m$, set $\tilde{x}_j(\ell) = 1$ and go to step 13.

Otherwise, continue.

Step 13: Subtract $\{j\}$ from $FV(\ell)$ and go to step 8.

Replacing P4 by P13 and bearing in mind that j in P5 and P6, instead of $IRR_j = \max_{\nu\epsilon FV(\ell)} \{IRR_\nu\}$, should now satisfy the equality $s_j = \max_{\nu\epsilon FV(\ell)} \{s_\nu\}$, problem (6.4) can be solved by means of the algorithm of Section 4.3.3. As regards the upper bounding functions, the Z_ℓ*'s will always be (easily) available (see Lemma 6.4), and for the Ω_ℓ*'s the solution of the following non-linear programming problem will be required:

$$\left.\begin{aligned} \text{Max } \Omega \;&=\; \sum_{j=1}^{n} f_j[r(\ell)]x_j \\ \text{s.t.} \quad & g_i(\underline{x}) \leqslant b_i,\; i = 1,2,\ldots,m \\ & x_j = x_j(\ell)\;,\; j \notin FV(\ell) \\ & 0 \leqslant x_j \leqslant 1,\; j = 1,2,\ldots,n \end{aligned}\right\} \qquad (6.5)$$

Clearly, if the g_i's were linear, then problem (6.5) would be a linear programming problem which, as in the case of problem (4.5) (see Section 4.2.3), could be solved by inspection.

6.4) A NUMERICAL EXAMPLE.

Let the following zero-one programming problem be considered:

$$\left.\begin{aligned} \text{Max } Z &= \sum_{j=1}^{6} [a_j - b_j (\tfrac{3t}{t+1})^{\frac{1}{2}}] x_j \\ \text{s.t.} \quad t &= 8x_1+2x_2+x_3+ 2x_4+x_5+5x_6 \leq 15 \\ x_j &= 0 \text{ or } 1, \; j = 1,2,\ldots,6, \end{aligned}\right\} \quad (6.6)$$

where:

j	1	2	3	4	5	6
a_j	2	5	10	3	4	9
b_j	1	3	7	2	3	5

Writing $g(\underline{x}) = 8x_1+2x_2+x_3+2x_4+x_5+5x_6$, $h(t) = \frac{3t}{t+1}$ and $f_j(y) = a_j-b_j\sqrt{y}$, it can be seen that the form of problem (6.6) corresponds to that of problem (6.4): f_j is decreasing and $s_j = (a_j/b_j)^2$, for $j = 1,2,\ldots,6$, and h and g are increasing. Hence, the problem can be solved by the proposed solution method. The summarized results of the procedure and the final directed tree are presented in Table 6.1 and Figure 6.1, respectively; the optimal solution is $\underline{x}^* = (0,0,1,0,0,0)$.

6.5) SUMMARY.

In this chapter the solution of a class of zero-one programming problems was considered. These problems constitute a generalization based upon properties which were used to develop a solution method for particular applications in capital investment. Accordingly, the solution method was extended, after re-formulating and dealing with the corresponding requirements. Further extensions like those involved with technological dependence between projects (see Section 5.3) could also be adapted in this case.

Iteration i	ℓ	j	A_i	Concluding Nodes	Rejected Nodes	$\underline{x}(1) \neq \underline{\tilde{x}}(1)$	
						L_i	U_i
0	3	6	{2,3}	–	–	1.039	6.505
1	5	2	{2,4,5}	–	–	1.094	3.629
2	7	4	{2,4,7}	6	–	1.094	2.303
3	9	3	{9}	8	2,4	1.427	1.753
4	–	–	ϕ	10	11	1.427	–

TABLE 6.1 Summarized Results of the Numerical Example.

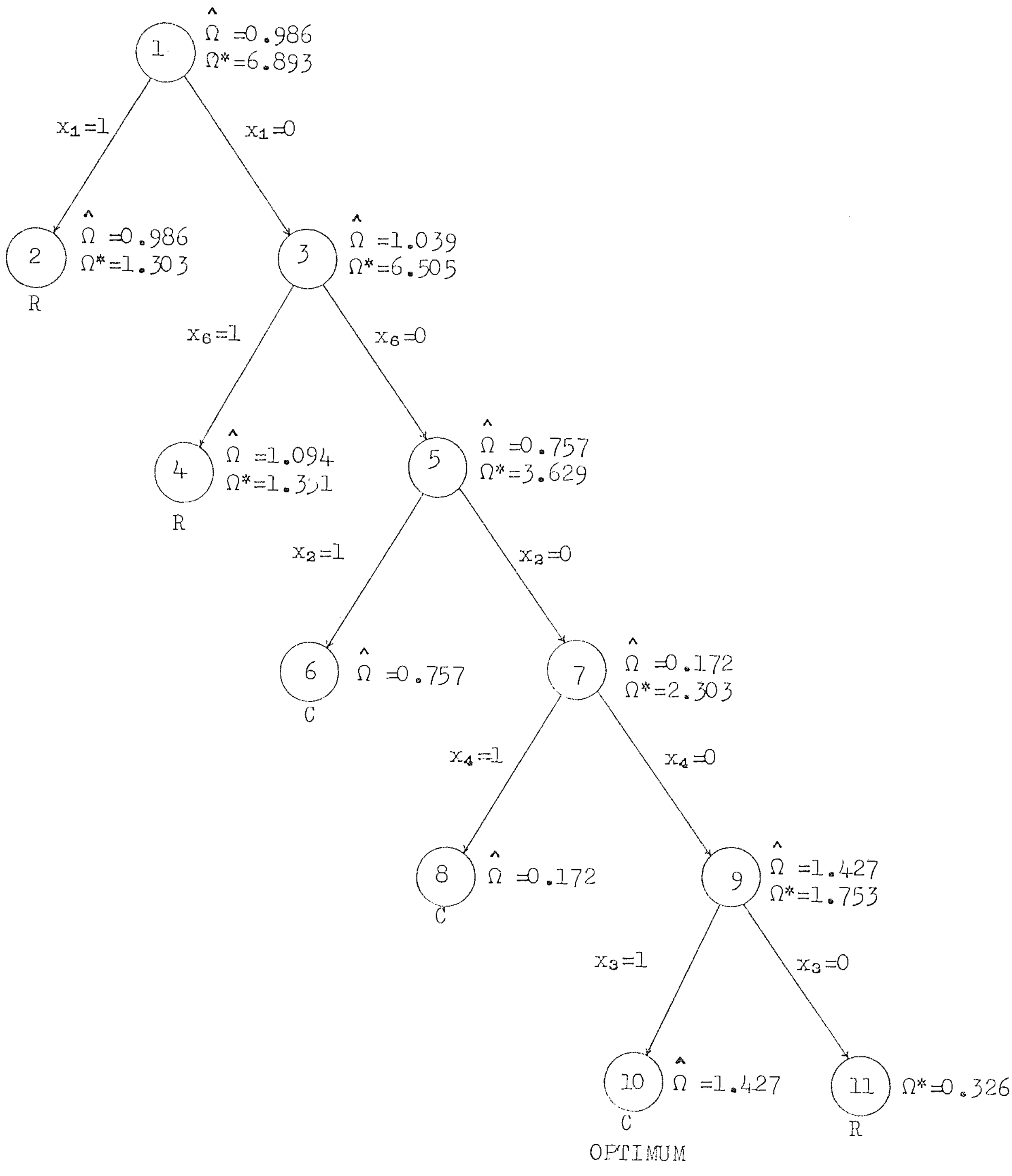

FIGURE 6.1 The Final Directed Tree.

CHAPTER VII

A ZERO-ONE BRANCH SEARCH APPROACH.

7.1) INTRODUCTION.

In the context of directed trees, as characterized in Section 2.2, the memory requirements of branch and bound algorithms are determined by the maximal number (M_A) of active nodes. These requirements are, therefore, bounded by the number (M_T) of terminal nodes in a complete final directed tree. Of course, M_A is to be expected to be much smaller than M_T, but as the number of variables (n) increases, it can be large enough to impose severe storage requirements. One way to overcome this problem is to alter the sequencing of the tree construction in such a way as to ensure that information corresponding to no more than n nodes has to be stored, at the cost of affecting the improving character of the upper bounds. Based on the search sequencing of implicit enumeration methods[1,2], Greenberg and Hegerich[3] applied this approach to the knapsack problem in what they called a "branch search" algorithm, using ideas of both Kolesar[4] and Land and Doig[5]. Upon the basis of the same search sequencing (often referred to as the "last created node" branching option[6]), problem (6.4) of Section 6.2 will be discussed in this chapter, making use of the properties by which the proposed branch and bound solution scheme was implemented.

7.2) THE SEARCH SEQUENCING.

In reference to problem (2.1) (see Section 2.2), let $T = \{x \in E^n / x_j = 0 \text{ or } 1,\ j = 1,2,\ldots,n\}$, and let $\mathbb{P}_j$ be defined as in Section 3.3.2.

DEFINITION 7.1: A non-terminal node ℓ of TR will be said to be NON-BOUNDED, if $\mathbb{P}_j\{N^{-1}(\ell)\} = \{t_1, t_2\}$ and either $N(t_1) \notin SN$ or $N(t_2) \notin SN$. Otherwise, ℓ will be said to be BOUNDED.

This definition is based in the way that f is bounded over S_ℓ when ℓ is non-terminal and TR is consistently upper-bounding. Such an upper bound is of course given by Z_ℓ^*, but then an upper bound at least as good as Z_ℓ^* and hopefully better than Z_ℓ^* is given by $\max\{Z^*_{r+1}, Z^*_{r+2}\}$, where $(\ell, r+1)$, $(\ell, r+2) \in SA$. However, if $(\ell, r+1)$ or $(\ell, r+2)$ are not elements of SA, then Z_ℓ^* cannot be improved as previously indicated. In this sense, it is said that ℓ is non-bounded.

DEFINITION 7.2: A terminal node ℓ will be said to be FINAL, if $\#\{\tilde{N}^{-1}(\ell)\} = 1$.

In order to see that the sequencing of the tree construction is well defined, it will be first introduced in terms of the construction of the complete directed tree. After showing that the sequencing does lead to the complete tree, implicit inspection considerations will be incorporated. It will be assumed that $TR_0 = \{SN_0, SA_0\}$ is given, where $SN_0 = \{1,2\}$, $SA_0 = \{(1,2)\}$, $N(T) = 1$, $N(t_1) = 2$ and $\mathbb{P}_j(T) = \{t_1, t_2\}$, for some $j \in \{1,2,\ldots,n\}$, chosen according to some criterion C.

CONSTRUCTION OF THE COMPLETE DIRECTED TREE (PROCEDURE P14).

Step 1: Set $i = 0$ and $\ell = 2$.

Step 2: If ℓ is final and the path $\underline{p}$ from the root of TR_i to k has only bounded nodes as elements, where $(k,\ell) \in SA_i$, stop; $TR(n) = TR_i$ is complete.

Otherwise, continue.

Step 3: If ℓ is final, define $N(t_2) = r+1$, where r is the total number of nodes of TR_i, $\mathbb{P}_j\{\tilde{N}^{-1}(s)\} = \{t_1, t_2\}$, and s is a non-bounded node, either equal to k, or such that $\underline{p} = (N(T), \ldots s, n_1, n_2, \ldots, n_q = k)$, n_ν being bounded, for $\nu = 1,2,\ldots,q$. Define $SN_{i+1} = SN_i \cup \{r+1\}$, $SA_{i+1} = SA_i \cup \{(s, r+1)\}$ and $TR_{i+1} = \{SN_{i+1}, SA_{i+1}\}$.

Set $i \leftarrow i+1$ and $\ell = r+1$. Go to step 2.

Otherwise, continue.

Step 4: Define $N(t_1) = r+1$, $SN_{i+1} = SN_i \cup \{r+1\}$,

$SA_{i+1} = SA_i \cup \{(\ell, r+1)\}$ and $TR_{i+1} = \{SN_{i+1}, SA_{i+1}\}$,

where $\mathbb{P}_j\{N^{-1}(\ell)\} = \{t_1, t_2\}$, and j is chosen according

to criterion C. Set $i \leftarrow i+1$ and $\ell = r+1$. Go to step 2.

LEMMA 7.1: For any $n \geq 2$, TR(n) is complete.

Proof (by induction):

If $n = 2$, procedure P14 leads to $TR_5 = \{SN_5, SA_5\}$, where $SN_5 = \{1,2,\ldots,7\}$ and $SA_5 = \{(1,2),(2,3),(2,4),(1,5),(5,6),(5,7)\}$. By Definition 2.5 (see Section 2.2), TR_5 is complete.

Let it now be assumed that TR(k) is complete, and that the variable x_{k+1} is added to the problem. Without loss of generality, it can also be assumed that x_{k+1} is such that x_j is always chosen before x_{k+1} by criterion C, for any $j < k+1$. Incorporating x_{k+1} to TR(k) (simply adding to T the corresponding 2^k new elements), a directed tree $\overline{TR}(k+1)$ of exactly the same configuration of nodes and arcs is obtained. Let $\overline{TN}(k+1)$ be the set of terminal nodes of $\overline{TR}(k+1)$. Clearly, since TR(k) is complete, $\#\{N^{-1}(\ell)\} = 2$ (for any $\ell \in \overline{TN}[k+1]$), $\bigcup_{\ell \in \overline{TN}(k+1)} N^{-1}(\ell) = T$ and $N^{-1}(\ell) \cap N^{-1}(s) = \phi$, for any two different terminal nodes ℓ and s of $\overline{TR}(k+1)$. This means that if, for each $\ell \in \overline{TN}(k+1)$, nodes $N(t_1)$ and $N(t_2)$ and arcs $[\ell, N(t_1)]$ and $[\ell, N(t_2)]$ were added to $\overline{TR}(k+1)$, where $\mathbb{P}_{k+1}\{N^{-1}(\ell)\} = \{t_1, t_2\}$, then the resulting tree would be complete for the problem with k+1 variables. Now, because x_{k+1} is the last variable to be chosen by criterion C, P14 would guarantee that $\overline{TR}(k+1)$ in terms of TR(k+1) is constructed, if for each terminal node ℓ of $\overline{TR}(k+1)$ the corresponding node ℓ' in TR(k+1) were generated and, after intermediate steps, used to carry on with the procedure, without altering the sequence of the construction. Bearing in mind that ℓ is final for TR(k), the next step after ℓ is created is to look for a node s, as defined in Step 3. On the other hand, after creating ℓ', first

$N(t_1)$ and $[\ell', N(t_1)]$, and then $N(t_2)$ and $[\ell', N(t_2)]$ are created, where $\mathbb{P}_{k+1}\{\tilde{N}^{-1}(\ell')\} = \{t_1, t_2\}$. As a result, $N(t_1)$ and $N(t_2)$ are final, and ℓ' is bounded. So, after creating $N(t_2)$, the next step is again to look for a node s', as defined in Step 3. Obviously, $s' \neq \ell'$, because ℓ' is now bounded. This shows not only that $\overline{TR}(k+1)$ in terms of TR(k+1) is constructed, but, as mentioned before, also that TR(k+1) is complete.

That procedure P14 finishes after a finite number of steps is clear from the fact that T is finite; that it leads to the optimal solution (or solutions) to problem (2.1) is a direct consequence of Lemma 7.1. Finally, it can be easily seen that, storing the best current feasible solution, only information corresponding to the path $\underline{p}$ (see Step 2) has to be available for computational purposes.

7.3) THE IMPLICIT ENUMERATION SCHEME.

The exhaustive search scheme provided by procedure P14 can be appropriately shortened for problem (6.4) (see Section 6.2) by means of the same implicit inspection criteria of the branch and bound approach. All that has to be taken into account is that no further considerations are necessary for concluding or rejected nodes. Therefore, these nodes can be treated as if they were final nodes. The same concepts associated with the nodes will hence be assumed. The changes due to the new search sequencing are next presented.

CONSTRUCTION OF THE INITIAL TREE (PROCEDURE P15).

Step 1: Define N(T) = 1 and obtain the associated concepts of node 1.

If $\underline{x}(1) = \tilde{\underline{x}}(1)$, stop; $\underline{x}^* = \underline{0}$ is the optimal solution to problem (6.4).

Otherwise, continue.

Step 2: Define $N(t_1) = 2$, where $\mathbb{P}_j\{T\} = \{t_1, t_2\}$ and j is such that $s_j = \max_{\nu \epsilon FV(1)} \{s_\nu\}$. The procedure is complete:

$SN_0 = \{1,2\}$ and $SA_0 = \{(1,2)\}$.

From procedure P 14, it can be observed that at each iteration only one new node ℓ will be created. If $\ell \epsilon A_i$, at the next iteration it will no longer be terminal, as a consequence of the creation of another node k such that $(\ell,k) \epsilon SA_{i+1}$. This means that, at the most, the number of elements of A_i will be one, for any $i \geqslant 0$. Only one parameter - that corresponding to the free variable j determining $\mathbb{P}_j$ - will therefore be associated with A_i. The determination of parameters of active nodes is hence reduced to the selection of the free variable j.

FREE VARIABLE SELECTION FOR ACTIVE NODES (PROCEDURE P16).

Step 1: Obtain A_i

Step 2: If $A_i = \phi$, stop; no parameter is necessary. Otherwise, continue.

Step 3: The parameter of A_i is j, where $s_j = \max_{j \epsilon FV(\ell)} \{s_\nu\}$ and $A_i = \{\ell\}$.

Of course, to obtain A_i procedure P13 (see Section 6.3) has to be followed for the current possible candidate for A_i, which is the node that was last created.

Only the procedure to obtain TR_i from TR_{i-1} remains to be described.

CONSTRUCTION OF INTERMEDIATE TREES (PROCEDURE P17).

Step 1: If $A_{i-1} \neq \phi$, define $N(t_1) = r+1$, where r is the total number of nodes of TR_{i-1}, $\mathbb{P}_j\{N^{-1}(\ell)\} = \{t_1, t_2\}$, $A_{i-1} = \{\ell\}$ and j is the parameter of A_{i-1}. Set $k = \ell$ and go to Step 3. Otherwise, continue.

Step 2: Define $N(t_2) = r+1$, where $\mathbb{P}_j\{N^{-1}(s)\} = \{t_1,t_2\}$ and s is defined as in Step 3 of P14, ℓ being the last created node. Set $k = s$.

Step 3: Set $SN_i = SN_{i-1} \cup \{r+1\}$ and $SA_i = SA_{i-1} \cup \{(k,\ell+1)\}$. Define $TR_i = \{SN_i, SA_i\}$.

Intermediate trees obtained following the sequencing under consideration are not necessarily upper-bounding. Therefore, upper bounds as defined in the branch and bound approach will not be available. However, a weaker definition can be introduced replacing A_i by $A_i \cup NBN_i$, NBN_i being the set of non-bounded nodes of TR_i. The resulting upper bounds are also improving along the procedure, but no priority is given to this effect. Thus, they can be expected to remain unchanged for relatively many iterations.

7.4) STATEMENT OF THE ALGORITHM.

Combining the concepts of the preceding sections, the algorithm to solve problem (6.4) can be stated as follows:

START (ITERATION 0)

1. Follow the steps of procedure P15
2. If $\underline{x} = \underline{o}$ is optimal, stop.
   Otherwise, continue.
3. Follow the steps of procedure P16 for $i = o$.
4. Set $i = 1$.

ITERATION i

1. Follow the steps of procedure P17.
2. Follow the steps of procedure P16.
3. If $A_i \neq \phi$, set $i \leftarrow i+1$ and re-start iteration i.
   Otherwise, continue.

4. If $A_i = \phi$ and the path $\underline{p}$ from the root of TR_i to k has only bounded nodes as elements, where $(k,r+1) \epsilon SA_i$, stop; L_i corresponds to an optimal solution to problem (6.4). Otherwise, set $i \leftarrow i+1$ and re-start iteration i.

7.5) A NUMERICAL EXAMPLE.

The application of the branch search approach is illustrated in this section iteration by iteration for the example of Section 4.3.4. The optimal solution to problem (4.5) (see Section 4.2.3) will be denoted by $\underline{x}^*(\ell)$, and the objective value of $\tilde{\underline{x}}(\ell)$ by $\tilde{Z}_\ell$.

START (ITERATION 0).

Node 1 is created

$\underline{x}(1) = (0,0,0,0,0,0)$, $r(1) = 0.106$ and $\hat{Z}_1 = 0$

$\tilde{\underline{x}}(1) = (0,1,1,0,1,0)$ and $\tilde{Z}_1 = 5.973$

$\therefore \hat{\Omega}_1 = 5.973$

$\underline{x}^*(1) = (0,1,1,\frac{2}{5},1,0) \therefore \Omega_1^* = 8.235$

Node 2 and arc (1,2) are created (j=2)

$\underline{x}(2) = (0,1,0,0,0,0)$, $r(2) = 0.106$ and $\hat{Z}_2 = 4.176$

$\tilde{\underline{x}}(2) = \tilde{\underline{x}}(1)$

$\therefore \hat{\Omega}_2 = 5.973$

$\underline{x}^*(2) = \underline{x}^*(1) \therefore \Omega_2^* = 8.235$

$A_0 = \{2\}$ and j=5

$L_0 = 5.973$

ITERATION 1

Node 3 and arc (2,3) are created (j=5)

$\underline{x}(3) = (0,1,0,0,1,0)$, $r(3) = 0.113$ and $\hat{Z}_3 = 5.943$

$\tilde{\underline{x}}(3) = \tilde{\underline{x}}(2)$

$\therefore \hat{\Omega}_3 = 5.973$

$\underline{x}^*(3) = \underline{x}^*(2) \therefore \Omega_3^* = 7.417$

$A_1 = \{3\}$ and j = 3

$L_1 = 5.973$

ITERATION 2

Node 4 and arc (3,4) are created (j=3)

$\underline{x}(4) = \tilde{\underline{x}}(4)$ $\therefore$ $\hat{\Omega}_4 = 5.973$

Node 4 is CONCLUDING

$A_2 = \phi$ and $\underline{p} = (1,2,3)$

$L_2 = 5.973$

ITERATION 3

Node 5 and arc (3,5) are created (s = 3 and j = 3)

$\underline{x}(5) = \underline{x}(3)$, $r(5) = r(3)$ and $\hat{Z}_5 = 5.943$

$\tilde{\underline{x}}(5) = (0,1,0,0,1,1)$ and $\tilde{Z}_5 = 5.273$

$\therefore$ $\hat{\Omega}_5 = 5.943$

$\underline{x}^*(5) = (0,1,0,\frac{11}{15},1,0)$ $\therefore$ $\Omega_5{}^* = 6.722$

$A_3 = \{5\}$ and $j = 6$

$L_3 = 5.973$

ITERATION 4

Node 6 and arc (5,6) are created (j = 6)

$\underline{x}(6) = (0,1,0,0,1,1) = \tilde{\underline{x}}(6) = \underline{x}^*(6)$ $\therefore$ $\hat{\Omega}_6 = \Omega_6{}^* = 5.275$

Node 6 is CONCLUDING and REJECTED

$A_4 = \phi$ and $\underline{p} = (1,2,3,5)$

$L_4 = 5.973$

ITERATION 5

Node 7 and arc (5,7) are created (s = 5 and j = 6)

$\underline{x}(7) = \underline{x}(5) = \tilde{\underline{x}}(7) = \underline{x}^*(7)$ $\therefore$ $\hat{\Omega}_7 = \Omega_7{}^* = 5.943$

Node 7 is CONCLUDING and REJECTED

$A_5 = \phi$ and $\underline{p} = (1,2,3,5)$

$L_5 = 5.973$

ITERATION 6

Node 8 and arc (2,8) are created (s = 2 and j = 5)

$\underline{x}(8) = \underline{x}(2)$, $r(8) = r(2)$ and $\hat{Z}_8 = 4.176$

$\underline{\tilde{x}}(8) = (1,1,1,0,0,0)$ and $\tilde{Z}_8 = 4.518$

$\therefore\ \hat{\Omega}_8 = 4.518$

$\underline{x}^*(8) = (0,1,1,\frac{4}{5},0,0)\ \therefore\ \Omega_8{}^* = 6.630$

$A_6 = \{8\}$ and $j = 3$

$L_6 = 5.973$

ITERATION 7

Node 9 and arc (8,9) are created (j = 5)

$\underline{x}(9) = (0,1,1,0,0,0) \neq \underline{\tilde{x}}(9) = \underline{\tilde{x}}(8)$

$\underline{x}^*(9) = (0,1,1,\frac{4}{5},0,0)\ \therefore\ \Omega_9{}^* = 5.862 < L_6$

Node 9 is REJECTED

$A_7 = \phi$ and $\underline{p} = (1,2,8)$

$L_7 = 5.973$

ITERATION 8.

Node 10 and arc (8,10) are created (s = 8 and j = 3)

$\underline{x}(10) = \underline{x}(8) \neq \underline{\tilde{x}}(10)$

$\underline{x}^*(10) = (\frac{3}{8},1,0,\frac{14}{15},0,0)\ \therefore\ \Omega_{10}{}^* = 5.895 < L_7$

Node 10 is REJECTED

$A_8 = \phi$ and $\underline{p} = (1,2,8)$

$L_8 = 5.973$

ITERATION 9.

Node 11 and arc (1,11) are created (s=1 and j=2)

$\underline{x}(11) = \underline{x}(1) \neq \underline{\tilde{x}}(11)$

$\underline{x}^*(11) = (0,0,1,\frac{3}{5},1,0),\ \therefore\ \Omega^*_{11} = 4.364 < L_8$

Node 11 is REJECTED

$A_9 = \phi$, $\underline{p} = (1)$ and node 1 is bounded

$L_9 = 5.973$ corresponds to the optimal solution

The final directed tree is presented in Figure 7.1. As can be observed, in this case the corresponding configuration in connection with that of Section 4.3.4 did not change. In general, however, due to the absence of effective upper bounds, larger final configurations can be expected to be obtained by means of this approach.

7.6) SUMMARY.

In this chapter, an alternative approach to deal with the zero-one programming problems under study was presented, the implementation of the implicit enumeration search sequencing being the main feature of the procedure. This sequencing, also considered as a branching option in the context of branch and bound methods, does not give priority to the calculation of upper bounds. Therefore, larger final directed trees are likely to be involved. The storage requirements, however, are bounded by the number of variables, as opposed to those associated with the original search sequencing.

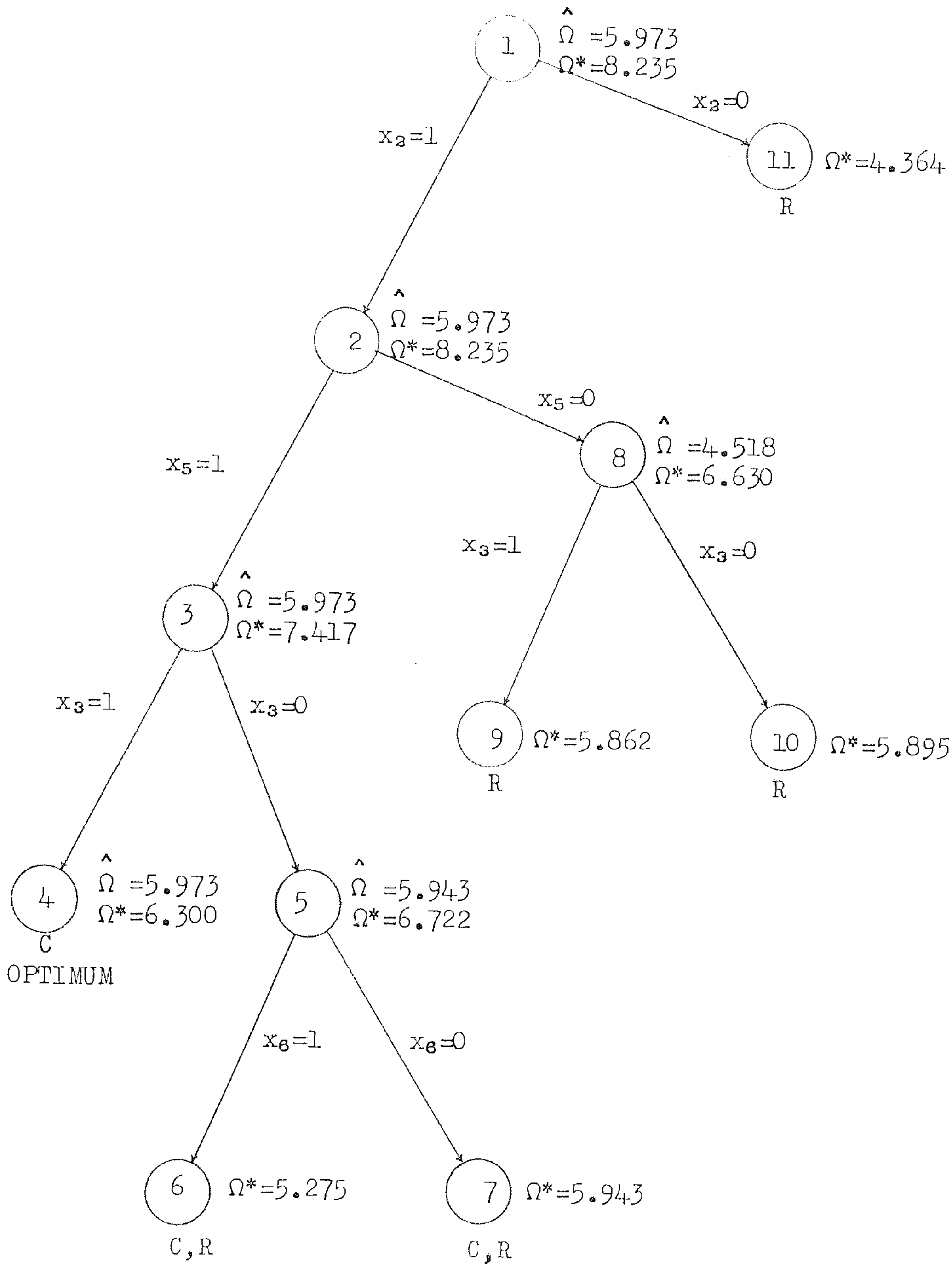

FIGURE 7.1 The Final Directed Tree.

REFERENCES.

1 F.GLOVER (1965) A multiphase-dual algorithm for the zero-one integer programming problem. Ops.Res.13, 879-919.

2 A.M.GEOFFRION (1967) Integer programming by implicit enumeration and Balas' method. SIAM Rev.7, 178-190.

3 H.GREENBERG and R.L.HEGERICH (1970) A branch search algorithm for the knapsack problem.Mgmt.Sci.16, 327-332.

4 P.KOLESAR (1967) A branch and bound algorithm for the knapsack problem. Mgmt.Sci.13, 723-735.

5 A.H.LAND and A.DOIG (1960) An automatic method for solving discrete programming problems. Econometrica 28, 497-520.

6 E.L.LAWLER and D.E.WOOD (1966) Branch-and-bound methods: A survey. Ops.Res.14, 699-719.