of this example with different methods can be made. However, the computational procedure of the hybrid algorithm is an efficient implicit enumeration technique in which no optimality criterion is violated, and hence the solution by this method based on the principle of optimality must be an optimum. Currently the only available information is the execution time of this single example. This example is solved using a WATFIV coded program on the AMDHAL 470V/6 computer. The execution time for this problem was 244.25 seconds. Considering the size and difficulty of the problem, this appears quite reasonable for obtaining an optimal solution.

Table 1
The optimal Allocation of The Budget levels for Industrial Branches

Branch No.	Budget Levels($) Min	Budget Levels($) Max	Optimal Budget($)	Optimal Return
1	290.000	400.000	310.000	762.84
2	100.000	600.000	600.000	1,190.85
3	340.000	540.000	520.000	822.77
4	260.000	420.000	420.000	809.24
5	350.000	600.000	600.000	1,640.67
		TOTAL	2.450.000	5.226.37

Table 2
The Optimal Investment alternatives

	Firm No.									
Branch No.	1	2	3	4	5	6	7	8	9	10
1	1	1	1	5	2	1	2	5	1	1
2	2	2	2	1	4	1	5	2	5	2
3	1	1	1	1	3	2	5	5	5	1
4	3	1	1	4	1	1	1	5	4	1
5	4	1	1	5	1	1	1	5	4	1

REFERENCES

1) Beightler, C.S., Pillips, D.T., and Wild, D.J., Foundations of Optimization, Prentice Hall, Englewood, N.J., 1979.
2) Ghasemi-Tari, F., An Algorithm for Large-Scale NKP and its Extensions, Ph.D. Dissertation, 1980.
3) Greenberg, H.J., and pierskalla, W.P., Surrogate Mathematical Programming, Operations Res., 18, No.5, 1970.
4) Morin, T.L., and Marsten, R.E., Branch and Bound Strategies for Dynamic Programming, Operations Res., 24, No.4, 1976.
5) Nauss, R.M., The 0-1 Knapsack Problem with Multiple Choice Constraints, University of Missouri, 1976.
6) Sinha, p., and Zoltners, A.A., The Multiple Choice Knapsack Problem, Operations Res., 27, No.3, 1979.
7) Toyoda, Y., and Senju, S., An Approach to Linear Programming with 0-1 Variables, Management Science, 15, No.4, 1968.

S.T.

$$\sum_{l=1}^{L} z_l = B$$

$$z_l \in S_l$$

where

S_l=The vector of the budget levels in the final stage of branch 1
R_l=The return vector obtained in the final stage of branch 1.

Problem 2.2 is indeed a one-dimensional (single linking constraint) nonlinear knapsack model which can be easily solved with dynamic programming techniques. The optimal solution resulting from solving problem 2.2 will define the optimal budget level, z_l, and after obtaining this value, the optimal set of investment policies for every firm of each branch can be recovered.

To compare the size of 0-1 linear programming model with the dynamic programming model, consider a problem with 5 branches, each including 10 firms, 5 investment alternatives for each firm, 3 resource constraints and ten budget levels for each branch. The dynamic programming model for this example will have 50 variables each having five different values, and three resource constraints. The 0-1 linear model will have 2800 zero-one vairables including 2500 variables for Z'S, 50 variables for Y'S and 250 variables for X'S, 15 resource constraints. 6 budget constraints and 5055 bounding constraints including 50 multiple-choice constraints for X'S, 5 multiple choice constraints for Y's and 5000 constraints for the variable transformation.

The above discussion indicates that the 0-1 linear programming model will be of incredible size in a realistic model. Perhaps one of the only alternative to the solution of the problem is the use of a diverging branch dynamic programming model.

A Case Problem

A case problem with 5 industrial branches, each including ten industrial firms, five investment alternatives, and three resource constraints will now be considered. The required data for this problem was randomly generated using Monte Carlo concepts of process generations; i.e., generating pseudo-random numbers using subroutine RANF and applying inverse transform to obtain a uniform distribution function for the required data.

The problem of finding the best set of investment alternatives for each firms in order to maximize the overall return is solved by the hybrid algorithm.

The algorithm is called a hybrid algorithm, and it is essentially a dynamic programming approach in the sense that the problem is divided into smaller subproblems. However, the idea of fathoming the partial solution by branch and bound is incorporated within the algorithm. The main feature of hybrid algorithm is its capability of reducing the state-space which otherwise would present an obstacle in solving multiple-constraint dynamic programming problems. part of this reduction is due to the use of the imbedded-state approach, which reduces an M-dimensional dynamic program to a one-dimensional problem. Other reductions are made through fathoming the state-space and subseqent elimination of state-space regions, which tend to eliminate inferior solutions compared to the predetermined lower or updated lower bound.

The use of a surrogate constraint methodology is implemented in the algorithm to obtain initial lower and upper bounds for the objective function. At each stage, the lower and upper bounds are also updated by use of a surrogated problem, and the updated upper bound will be used for termination criteria. The procedure for updating lower and upper bounds in the surrogated problem is very efficient. In addition, the primary advantages of using the surrogate problem to estimate these bounds, are(1) it provides a narrow range between the lower and upper bound, and(2) it may provide the optimal solution to the problem at the first step.

The generated data are used as input to the problem and optimal decision variables for each firm in each industrial branch are obtained along with the optimal allocation of the budget levels for each district. Table 1 illustrates the optimal allocation of budget levels in each branch. The information concerning the minimum and maximum budget levels, The return obtained by each allocation, and the total budget and total return are also presented in this table. Table 2 illustrates the optimal investment alternative which resulted from the allocation of the optimal budget to each branch. Each branch number and the corresponding optimal decision variables are also shown in this table. This example was discussed previously, wherein it was noted that the 0-1 linear programming model for this problem has 2800 0-1 decision variables, and 5055 constraints. The current state of the art in 0-1 integer programming techniques indicates that an optimal solution to a problem of this magnitude can not be achieved in a reasonable amount of computation time. Therefore, no comparison to the solution within a branch, and the node S from which each branch diverges represents the allocation of total budget to each individual branch.

In each branch there will be J stages representing the number of firms in that branch, and L branches diverging from node S.each branch may be solved as an initial-value problem in terms of z_{jl}. This is accomplished using forward recursion carrying z_{jl} as an extera state variable. At the final stage the return vector, a function of the state variables, will be obtained for each branch. The state variables represent the consumption of the resources such as types of equipment, materials, personnel, and the total budget level. Among these state variables, only consumption of the budget is the subject of further optimization and all other state variable inputs are fixed. As a result, the returns of each branch, as a function of budget level, are obtained. Considering each branch as a single stage in the dynamic programming model, a decision must be made with regard to the allocation of budget levels to each branch in order to obtain the maximum return.

Referring to Fig.2, it can be seen that each branch involves a multiple-Constraint dynamic programming problem. These constraints are divided into two groups. The first group is represented by a state vector y_{jl}. The second group, projects cost, is represented by a single-state variable, z_{jl}. This separation has just been justified; i.e., the cost constraint interrelates the decision-making process between the different branches, while the group of constraints represents by y_{jl} can be considered independently in each branch.

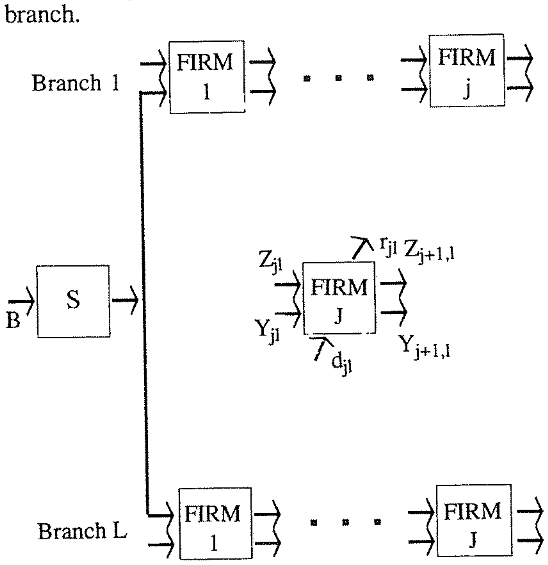

Figure 2. Schematic presentation for the allocation of funds within and between branches.

Consider branch1; allocation of resources to this branch using a dynamic programming technique results in the following recursive equations:

$$R_{1l}(z_{1l}, y_{1l}) = \text{Max. } r_{1l}(x_j)$$

over

$$0 \leqslant A_{1l}(x_j) \leqslant y_{1l}$$

$$0 \leqslant c_{1l}(x_j) \leqslant z_{1l}$$

$$R_{lj}(z_{jl}, y_{jl}) = \text{Max. }\{r_{jl}(z_j) + \text{Max. }\{R_{j-1,l}(z_{j-1,l}, y_{j-1,l})\}\}$$

for j=2,3,...,J, and over;

$$0 \leqslant A_{jl}(x_j) \leqslant y_{1l}$$

The state recursion equations are:

$$z_{1l} = B$$

$$y_{1l} = TR$$

$$z_{j-1,l} = z_{jl} - c_{jl}(x_j) \quad \text{for } j=2,3,\dots,J$$

$$y_{j-1,l} = y_{jl} - A_{jl}(x_j) \quad \text{for } j=2,3,\dots,J$$

where the state variables are defined as;

z_{jl} =The amount of budget available for stages j, j+1, ..., J

y_{jl} =The vector whose components represents the amount of each type of resource available for stages j, j+1, ..., J

and

$$TR=(b_{1l}, b_{2l}, \dots, b_{I-1,l}).$$

The recursive equations developed for branch 1 can be applied to all branches, i.e., 1=1, 2, ..., L. After dynamic programming is applied to all the branches the return

$$R_{Jl}(z_{Jl}, y_{Jl})$$

will be obtained. Since the first group of constraints is not involved in the allocation of budget to branches, let

$$R_l(z_l) = R_{Jl}(z_{Jl}, y_{Jl}).$$

The distribution of budget levels to each branch is then obtained by solving the following problem:

Problem 2.2

$$\text{Max. } \sum_{l=1}^{L} R_l(z_l)$$

$$x_{ijl} = \begin{cases} 1 \text{ if project j is selected for firm i in branch 1} \\ 0 \text{ otherwise} \end{cases}$$

$$y_{kl} = \begin{cases} 1 \text{ if budget level k is used in branch 1} \\ 0 \text{ otherwise} \end{cases}$$

$$z_{ijkl} = x_{ijl} y_{kl}$$

I=The number of firms in each branch
J=The number of alternatives projects for each firm
K=The number of budget level in each branch
L=The number of branches
r_{ijk} =The return obtained by selecting project j for firm i in branch1
d_{ijklm} =The amount of resource type m used by selecting project j for firm i in branch 1 (M is the total number of resources)
D_{ml} =The amount of the available resource of type m in branch1
b_{ml} =The budget level k considered for branch 1
c_{ijl} =The cost of project j for firm i in branch 1
B=The total available budget for entire organization

A more desirable approach to the problem of allocating budget levels to branches is to develop a model capable of handling both the within and between branches allocation process optimally. The mathematical representation of such a model in the form of a NKP is presented below.

Problem 2

$$\text{Max.} \sum_{l=1}^{L} \sum_{j=1}^{J} r_{jl}(x_j)$$

S.T.

$$\sum_{j=1}^{J} a_{ijl}(x_j) \leqslant b_{il} \quad \text{for } i=1,2,...,I\text{-}l \ \& \ l=1,2,...,L$$

$$\sum_{l=1}^{L} \sum_{j=1}^{J} c_{jl}(x_j) < B$$

$$x_j \in S_{jl}$$

$$S_{jl} = \{1,2,...,K_{jl}\}$$

where

a_{ijl} =The amount of resource type i (excluding projects cost) consumed as a function of alternative X, for firm j at branch 1
b_{il}=Total amount of type i available resource (excluding budget level) at branch 1
c_{jl} =The amount of consumption of project cost, a function of alternative X, for firm j at branch 1
L=The number of branches in the organization
K_{jl} =The number of investment alternatives that can be selected for firm j at branch 1
M=The number of resource constraints excluding cost
r_{jl} =The return function of alternative X, for firm j at branch 1
B=Total amount of available budget for entire organization
x_j =The decision variable indicating the types of alternative to be selected.

problem 2 can be decomposed into two levels. The first level is to decompose the problem according to the branches. Each branch can then be considered as a single stage of a total dynamic programming problem. The second level is a decomposition of the branch. This process can be more clearly illustrated by expanding
problem 2.

Problem 2.1

$$\text{Max.} \sum_{j=1}^{J} r_{j1}(x_j) + \sum_{j=1}^{J} r_{j2}(x_j) + \ldots \sum_{j=1}^{J} r_{jL}(x_j)$$

S.T.

$$\sum_{j=1}^{J} a_{ij1}(x_j) \leqslant b_{i1}$$

$$\sum_{j=1}^{JJ} aij2\,(xj) \leqslant bi2$$

$$\sum_{j=1}^{J} a_{ijL}(x_j) \leqslant b_{iL}$$

for i=1,2,..., I-1

$$\sum_{j=1}^{J} c_{j1}(x_j) + \sum_{j=1}^{J} c_{j2}(x_j) + \ldots + \sum_{j=1}^{J} c_{jL}(x_j) \leqslant B.$$

Referring to problem 2.1, the limitations on all the resources are considered independently for each branch with the exception of the limitation on the budget level which interrelates the decisions in all branches. However, the allocation process within each branch could be developed independently if it were developed as a function of budget level in that branch. That is, a vector presenting the optimal return as a function of budget level in each branch could be obtained. These branches benefits and their associated cost levels could be used for the allocation of total budget to individual branch. This two-level allocation process can be suitably performed using a non-serial dynamic programming model. This model is illustrated schematically in Figure 2. In this Figure,each branch represents the allocation of resources levels through the organization branches is indeed a subsystem optimization. Generally, a subsystem optimization may not reveal all possible decisions and hence, may limit the study's range of possible solutions. The fact that subsystem optimization cannot cover all possible solutions can happen in our problem where one or more of the budget levels lies out of the predicted range. In addition, the current approach is a very inefficient way of handling this problem, because an optimization process in the form of a large 0-1 model must be performed at each budget level for each branch, in order to obtain the required information for further distribution of funds throughout all the branches. Thus, a better model must be developed to resolve the above problems, in order to develop a total system optimization model.

The overal optimization process is considered as a sequence of interrelated decisions in each branch. These decisions are interrelated by the amount of total available budget to be distributed in each branch. Considering each branch as a stage of a dynamic programming model, in each stage a multiple-resource constrained nonlinear knapsack problem must be solved, and the resulting returns with a single-state transition equation (amount of funds spent in each branch) is transferred to the next stage. This process is shown schematically in figure 1. In this figure, S_0 is the total amount of available funds; S_i is the amount of funds available for the next i+1, i+2, ..., N stages; d_i is the decision variable presenting the amount of funds allocated to branch i; and r_i is the retun from branch i when d_i unit of funds are used, for i+1, i+2, ..., N, where N is the number of branches.

The return r_i is the expected value of total benefits from branch i, which is calculated as the summation of the expected benefits gained in each firm.

The value of the return r_i is actually obtained by solving an NKP as a branch of a diverging branch dynamic programming model. Hence each branch of this model consist of a serial dynamic programming model in which each stage represents the selection of different alternatives in each firm in any specific industrial branch.

Development of a Mathematical Model for the problem

A 0-1 linear programming approach to the problem is interesting to examine, even though the NKP can be shown to be more efficient optimization technique. The propose is to provide an insight to a better comparison between two models; 1)dynamic programming model, and 2) the 0-1 linear model.

The 0-1 linear model for allocation of resources within each branch and the distribution of organization-wide budget between branches can be presented in the following form:

Problem 1

$$\text{Max. } R=\sum_{i=1}^{I}\sum_{j=1}^{J}\sum_{k=1}^{K}\sum_{l=1}^{L} r_{ijk}\, z_{ijkl}$$

S.T.

$$\sum_{i=1}^{I}\sum_{j=1}^{J} d_{ijlm}x_{ijl} \leq D_{ml} \quad \text{for } m=1,2,\ldots,M \;\&\; l=1,2,\ldots,L$$

$$\sum_{i=1}^{I}\sum_{j=1}^{J} C_{ijl}\,x_{ijl} \leq \sum_{k=1}^{K} y_{kl}\, b_{kl} \quad \text{for } l=1,2,\ldots,L$$

$$\sum_{l=1}^{L}\sum_{k=1}^{K} b_{kl}\,y_{kl} \leq B$$

$$\sum_{j=1}^{J} x_{ijl} = 1 \quad \text{for } i=1,2,\ldots,I \;\&\; l=1,2,\ldots,L$$

$$\sum_{k=1}^{K} y_{kl} = 1 \quad \text{for } l=1,2,\ldots,L$$

$$x_{ijl} + y_{kl} - z_{ijkl} < 1 \quad \text{for all } i,j,k,l\text{ 's}$$

$$-x_{ijl} - y_{kl} - 2\,z_{ijkl} < 0 \quad \text{for all } i,j,k,l\text{ 's}$$

$$x_{ijl}\,,\; y_{kl}\,,\;\&\; z_{ijkl} = 0 \text{ or } 1 \quad \text{for all } i,j,k,l\text{ 's}$$

where;

R=The total return obtained from the scheduled investment activities

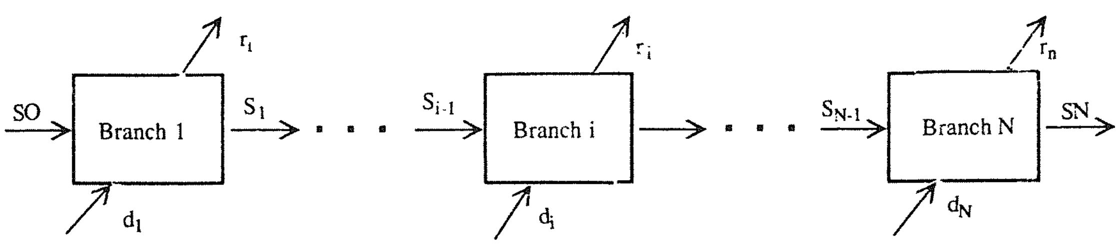

Figure 1. *The optimization process for the allocation of funds among organization branches*

resource constrained problems.

An alternative approach to the single industrial branch optimization is to formulate the problem as a "Nonlinear knapsack problem" (NKP) which significantly reduces the number of variables and eliminates all the constraints except the resource constraints. A promising solution approach to handle NKP is discrete dynamic programming. Although this approach reduces the dimensionality of the decision variables, it suffers from the fact that the existence of more than three resource constraints renders this approach computationally intractable. This is the well-Known problem of dimensionality of state variables in the dynamic programming technique. One way to reduce the M-state variable dynamic programming problem to a single-state variable problem is through the use of lagrangian multipliers which incorporate some of constraints in the objective function, and solve a series of lower dimensional state problems sequentially. Another attempt in this area was based on the concept of surrogate constraints in which some or all of the constraints were replaced by a single constraint defined by a linear combination of the original constraints. As the result of this transformation, a series of single constrained problems are solved sequentially. However the problem of duality gaps, which is likely in case of discrete variables, makes these approaches somewhat dubious. The other alternative method to reduce the dimensionality of the state variables is by employing the "imbedded state" technique. Although the comparative efficiency of both the lagrangian and imbedded state approach is a questionable matter, and probably depends on the structure of the problem, the latter approach is reported to be relatively more efficient for NKP.

The use of dynamic programming techniques for single branch optimization provides a bookkeeping record of returns for different funding levels which can be used for the overall distribution of funds, in an optimal manner, throughout different industrial branches. Conversely, in the use of 0-1 programming for single branch optimization, the overall distribution of funding levels would be either impossible or a very difficult and time consuming task. The fact that the 0-1 integer programming model necessitated by the large number of decision variables, dose not even facilitate a model to obtain a near-optimal solution to the single-branch problem is the best reason for using one of the alternative dynamic programming based techniques.

Through this research a model will be developed which will be capable of distributing foreign currency funds over a single branch throughout organization branches and allocating other available resources within each branch.

Problem Discription

consider an organization which is faced with the problem of allocating budget throughout its different industrial branches. In particular the scarcement of foreign currency imposes special consideration to be paid to this allocation. this organization has catagorized its owned firms into the five distinct branches, namely:

1) Casting and rolling branch.
2) motorized vehicles branch.
3) production machinery branch.
4) pressure vessels and equipments branch.

Each branch may owns more than 30 industrial firms. Every year a number of firms may propose a set of capacity expansion and new projects to the executive committee of their branches. This committee reviewes the proposal and based on the attractiveness of the proposed investment project and its requirements, will decide which projects to be selected. However the decision made by each committee branches may be adjusted due to the limitation of budget. Hence the final decision on the selection of projects is made by a top committee in the organization based on the availability of budget. Which indeed allocates the budget levels throughout the entire branches.

Until recently the process of allocating resources in each branch and allocation of budget through the entire branches is performed subjectively without use of a systematic approach. Therefore it was a need to employ a systematic way of allocating available resources.

As an early attempt to resolve the allocation problem, we considered each branch individually and developed a knapsack model in the form of 0-1 problem. This model was able to allocate the available resources in each branch assumming a range for the amount of fund is predicted (based on previous years) and the lowest value of this range was used as a budget constraint in the optimization model of each branch.

The amount of funds in each branch then varied over this range and the optimization routine was applied to each funding level. As a result, a vector of funding levels with its associated vector of optimal returns was obtained. These two vectors were then used for the allocation of the total budget among the branches for the entire organization.

The current approach for distribution of funding

A Strategic Planning Model For Capacity Expansion And New Investment Projects

Farhad Ghasemi-Tari (Ph. D.)

Associate Prof. Indust. Eng. Dept.

Sharif University of Technology

Abstract

A diverging-branch dynamic programming model is developed to solve the problem of selecting investment alternatives for each firm in an organization with multi-industrial branch. A case problem is formulated and solved via a developed hybrid algorithm.The computational result is also presented.

Introduction

The allocation of funds for the problem of strategic planning concerning capacity expansion and the selection of new investment projects requires the use of a systematic approach to maximize the return of the investment and to minimize the waste of available resources. The strategic objective for the problem is the selection of optimal investment policy for a given project in order to maximize the total return of all investment activities scheduled for the entire organization in each year.

This is an optimization process, requiring a sequence of interrelated decisions in each "industrial branch" of the organization. Each industrial branch includes several industrial firms and every year a number of this firms propose several investment alternatives for their capacity expansion and their new projects. The problem of allocating resources in an individual branch can be considered as an optimization process in which the objective is to find the best set of alternative investments policy for each firm under supervision of this branch, subject to the existing menpower, equipments, materials, and other overhead cost limitations in this branch.

The single branch optimization process is a knapsack type problem with multiple choice constraints. An attempt to find a solution to this single branch optimization problem is to formulate the problem as a 0-1 integer linear programming model. As a result a large scale 0-1 problem can be solved by the use of a heuristic algorithm and a near optimal solution can be obtained.

The simplest version of the knapsack problem is called the "one dimensional knapsack problem" which includes only one resource constraint. A general version of the knapsack problem is the case where there are several resource constraints, several sets of mutually exclusive alternatives, and the objective is to select the optimal alternative from each set. The restriction of having only one alternative to be selected from each set adds a set of constraints, called multiple choice constraints, to the original problem.

One of the earliest potentionally successful research efforts in the area of large scale binary knapsack problems is the work of Toyoda and Senju. They have developed an approximation algorithm which is capable of generating near optimal solution to large scale binary knapsack problems in a relatively short computational time.

Nauss developed an algorithmic procedure based on the idea of lagrangian relaxation to solve a binary one-dimensional knapsack problem. The idea of branch and bound (B &B) is implemented in the algorithm in order to find the optimal values of the Lagrangian multipliers. Although the computational experience with this method has shown that algorithm is very efficient for large problems, the fact that it can only solve one-constraint problems limits its applicability.

Sinha and Zoltners have developed an algorithm for multiple choice knapsack problems which is reported to be much faster than Nauss method. The applicability of this method is also limited to single-