FINANCIAL PLANNING WITH 0-1 KNAPSACK PROBLEMS, PART I: DOMINATION RESULTS

Daniel E. O'Leary

ABSTRACT

The 0/1 knapsack problem provides an important model for financial planning. It is used to determine what subset of items provides the greatest return. Typically, it is used in situations such as budgeting, where there are only enough funds to sponsor a subset of projects. This paper provides results that allow us to determine when one project "dominates" another, that is, when some project is always preferable to another. Those results are useful since they allow us the ability to reduce the number of variables and the overall budget constraint. This leads to a smaller, more tightly constrained problem. In some cases, establishing domination results leads to a complete solution of knapsack problems.

I. INTRODUCTION

The 0-1 knapsack problem has been used extensively in the structuring of financial planning problems, such as budgeting (e.g., Weingartner,

Advances in Mathematical Programming and Financial Planning,
Volume 4, 1995, pages 139-150

ISBN: 1-55938-724-6

1962). Typically, the firm faces the problem of choosing among a subset of a portfolio of projects from a set $X = (x_1, \ldots, x_n)$. For example, a firm must choose a set of capital investments from the set proposed. Typically, the firm is assumed to have a budget constraint limiting the number of projects, of \$b. Each project x_i has a return c_i and a cost a_i. The problem then becomes one of choosing the projects that maximize return, subject to the budget constraint. This problem is an integer programming problem, known as the 0-1 knapsack problem.

A. 0-1 Knapsack Problem

It is assumed that the variables, x_i, can take only values of 0 or 1. In addition, it is assumed that the concern is with maximizing return subject to a single constraint. In particular, the mathematical programming version of the problem can be written as follows:

$$\max \sum_{i=1}^{n} c_i x_i$$

$$\sum_{i=1}^{n} a_i x_i \leq b, \; x_i = 0 \text{ or } 1, \tag{1}$$

where,

a. $\dfrac{c_i}{a_i} \geq \dfrac{c_j}{a_j}$ for all $i < j$,

b. If $\dfrac{c_i}{a_i} = \dfrac{c_j}{a_j}$ then $a_i \geq a_j$ for $i < j$.

c. c_i, a_i and b are positive integers,

d. $\sum_{i=1}^{n} a_i \geq b$.

There are a number of versions of the problem. One alternative is to permit the variables to take on a larger set of values (e.g., all integers). Another alternative formulation is to choose a portfolio of minimum cost activities that provide a cumulative benefit that exceeds a particular

benefit parameter. A variety of formulations have found use in financial planning problems (e.g., Weingartner, 1962, and others).

Since Problem (1) is an integer programming problem it is difficult to solve. As a result, there is interest in finding ways to improve solution processes in order to facilitate financial planning processes. This paper presents a basic approach referred to as "domination." In some cases conditions can be developed where it can be determined it is always preferable to use one variable as opposed to another. This is referred to as a domination relationship. Those domination relationships can be used to assist in the solution of problems formulated as (1). In particular, using the results of this paper, the dimension n, of the problem and the value of b can be reduced. Further, in some cases, those reductions lead directly to a solution to Problem (1).

B. This Paper

This paper has two parts. Part I is concerned with the reduction of n and b in (1), illustrating the approach. Part II of the paper provides an analysis of the use of the domination results in the solution of knapsack problems with a variety of algorithms.

This paper proceeds as follows. Section II develops the domination results. Section III uses the results of Section II to provide additional results on the reduction of the problem size of (1). Section IV illustrates the approach using an example. Section V discusses some empirical results of the impact of reduction. Section VI briefly discusses some extensions and summarizes the paper.

Part II of this paper uses the domination results from Part I in a number of different ways. Section VII extends the use of domination results beyond problem reduction to branch and bound algorithms to solve Problem (1). Similarly, Section VIII uses the domination results to reduce the computations in dynamic programming solutions. Section IX investigates the use of domination in ϵ approximate algorithms. Finally, Section X briefly discusses some extensions and summarizes Part II of the paper.

II. DOMINATION RESULTS

This section presents the principle results of the paper. These results are based on the notion that in a given problem one variable is always at least as desirable as some other variable, that is, one variable *dominates*

another. This variable domination results in a partial ordering of the variables that can aid in the search for optimal solutions to Problem (1).

Definition

If there exists an optimal solution to Problem (1), where, for $i < j$,

$x_i = 0$ implies $x_j = 0$ and

$x_j = 1$ implies $x_i = 1$, then

x_i *dominates* $x_j (x_i \rightarrow x_j)$.

Note that if $x_i \rightarrow x_j$ then $x_i \geq x_j$. Using this observation, transitivity is an immediate consequence.

Theorem 1 (Transitivity). If $x_i \rightarrow x_j$, $x_j \rightarrow x_k$ then $x_i \rightarrow x_k$.

There are three different approaches to establishing domination, based on the relationship between parameters associated with each pair x_i and x_j. In particular, either: $c_i \geq c_j$ and $a_i \leq a_j$; or $c_i \geq c_j$ and $a_i \geq a_j$; or $c_i \leq c_j$ and $a_i \leq a_j$. The following theorems relate to these cases, respectively.

Theorem 2. If $i < j$, $c_i \geq c_j$ and $a_i \leq a_j$ then $x_i \rightarrow x_j$.

Proof

Suppose in some optimal solution $x_i = 0$. If $c_i > c_j$, then $x_j = 0$ or the solution is not optimal. If $c_i = c_j$ and $x_j = 1$, then x_i can replace x_j to form at least as good a solution.

Let $\gamma_j^* = \{i \mid c_i \geq c_j \text{ and } a_i \leq a_j\}$ and $\Delta_i^* = \{j \mid c_i \leq c_j \text{ and } a_i \geq a_j\}$. γ_j^* is the set of variables dominated by x_i, through Theorem 2. Δ_i^* is the set of variables that dominate x_i through Theorem 2.

Theorem 3. If $i < j$ and

a. $c_i \geq c_j$ and $a_i \geq a_j$

b. $\gamma_i^* \subseteq \gamma_j^*$

c. $c_i + \sum_{k \in \gamma_i^*} c_k \geq c_j + \sum_{k \in \gamma_j^*} c_k$

d. $a_i + \sum_{k \in \gamma_i^*} a_k \leq a_j + \sum_{k \in \gamma_j^*} a_k$

Then $x_i \rightarrow x_j$.

Proof

Since assumptions (b), (c) and (d) hold, this means

$$c_i \geq c_j + \sum_{\substack{k \in \gamma_j^* \\ k \notin \gamma_i^*}} c_k$$

$$a_i \leq a_j + \sum_{\substack{k \in \gamma_j^* \\ k \notin \gamma_i^*}} a_k$$

If $x_j = 1$, then $x_k = 1$ for $k \in \gamma_j^*$, in some optimal solution or else $x_j \neq 1$. Either $x_i = 1$ or $x_i = 0$. In the later case x_i can replace x_j and x_k, $k \in \gamma_j^*$, $k \notin \gamma_i^*$, to form at least as good a solution. Thus if $x_j = 1$ in some optimal solution then $x_i = 1$.

If $x_i = 0$ then either $x_j = 0$ or $x_j = 1$ and $x_k = 1$ for $k \in \gamma_j$. In the later case x_i can replace x_j and $x_k, k \in \gamma_j^*$, $k \notin \gamma_i^*$ to form at least as good a solution. As a result if $x_i = 0$ then $x_j = 0$ in some optimal solution. Thus, $x_i \rightarrow x_j$.

Theorem 4. Let Δ_i^{**} be a subset of Δ_i^*. If $i < j$ and there exist k such that

a. $c_i \leq c_j$ and $a_i \leq a_j$

b. $\Delta_j^* \subseteq \Delta_i^{**}$

c. $c_i + \sum_{k \in \Delta_i^{**}} c_k \geq c_j + \sum_{k \in \Delta_j^*} c_k$

d. $a_i + \sum_{k \in \Delta_i^{**}} a_k \leq a_j + \sum_{k \in \Delta_j^*} a_k$

Proof

Since assumptions (c) and (d) hold this means

$$c_i + \sum_{\substack{k \in \Delta_i^{**} \\ k \notin \Delta_j^{*}}} c_k \geq c_j$$

and

$$a_i + \sum_{\substack{k \in \Delta_i^{**} \\ k \notin \Delta_i^{*}}} a_k \leq a_j$$

If $x_j = 1$ and $x_i = 0$, then for all k in Δ_i^{**}, $x_k = 0$. But x_i and x_k for $k \in \Delta_i^{**}$ could be set equal to 1 and x_j set to 0 to produce at least as good a solution. Thus, $x_i = 1$. If $x_i = 0$, then either $x_j = 0$ or $x_j = 1$. If $x_j = 1$, then as above x_i and some x_k for $k \in \Delta_i^{**}$ could be set to 1 in some optimal solution. Thus, if $x_i = 0$, then $x_j = 0$ in some optimal solution. As a result, $x_i \rightarrow x_j$.

This section has provided a means for determining a domination relationship in all pairs of variables x_i and x_j. The results of Theorems 3 and 4 mirror Theorem 2 with the assumptions (c) and (d) in each case. In addition, Theorems 3 and 4 are "dual-like" results since Theorem 3 uses γ_i^* and γ_j^*, while Theorem 4 uses Δ_i^* and Δ_j^*.

III. IMPLEMENTATION OF THE DOMINATION RESULTS

The domination results in Theorems 2, 3 and 4 can be implemented to reduce the problem size of (1).

A. Reduced Domination Network

A reduced domination network can be constructed to facilitate summarizing each of the domination relationships. A "reduced domination network" can be constructed as follows.

a. For x_i a variable, let i be a node.
b. If $x_i \rightarrow x_j$ let there be an arc, a_{ij}, from node i to node j.
c. If a_{ij}, a_{ik}, and a_{jk} are arcs, remove a_{ik} from the network.

The resulting network preserves the domination results in the following sense. If $x_i \rightarrow x_j$, then there is a path from i to j. In addition, the network maintains the property that it is acyclic.

At each node in the network, the sum of all the c_i and a_i parameters associated with each variable x_i (or node i in the network) can be computed and used to assess whether or not a variable should be in some optimal solution, given a specific b value. In particular, the theorems of the next subsection can be used to generate decisions as to whether one variable should be a 0 or a 1.

B. Determining Variable Values

If some variable dominates "enough" other variables, then for a given value of b, that variable can be valued at 1 in some optimal solution. On the other hand, if some variable is dominated by enough variables, then that variable can be valued at 0 in some optimal solution. This section provides results to establish both decisions.

Let $\gamma_i = \{j \mid x_i \rightarrow x_j\}$ and $\Delta_i = \{j \mid x_j \rightarrow x_i\}$

Let $I^0 = \{i \mid x_i = 0\}$, $I^1 = \{i \mid x_i = 1\}$ and $I^2 = \{i \mid i \notin (I^0 \cup I^1)\}$.

Theorem 5. If $a_i + \Sigma_{j \in \gamma_i} a_j > b$ then $x_i = 0$ in some optimal solution.

Theorem 6. If $\Sigma_{j \notin (\Delta_i \cup I^0)} a_j \leq b$ then $x_i = 1$ in some optimal solution.

Theorem 7 consolidates the information in Theorems 5 and 6 to present a reduced form of Problem (1).

Theorem 7. Problem (1) can be reduced to

$$\max \sum_{i \in I^2} c_i x_i + \sum_{i \in I^1} c_i$$

$$b - (\sum_{i \in I^1} a_i) \geq \sum_{i \in I^2} a_i x_i$$

$$x_i = 0 \text{ or } 1, i \in I^2.$$

C. Implementation of Theorems 2, 3, and 4

Theorem 2 is easy to implement, requiring $2((n-1) + \ldots + 1)$ comparisons. Theorem 3 would only be done for those variables for which $c_i \geq c_j$ and $a_i \geq a_j$. Theorem 3 would use the domination results computed under Theorem 2 for the entire set of variables Δ_j that dominate x_j. However, Theorem 4 applies to subsets of variables that x_i dominates. The enumeration of the set of subsets potentially could be computationally expensive. As a result, for implementation purposes, only sets of size one are used, where the element chosen is that with the smallest index t. Alternatively, the smallest a_t could be chosen.

D. Example

Consider the example discussed by Kolesar (1967) and others.

$$\max 60x_1 + 60x_2 + 40x_3 + 10x_4 + 20x_5 + 10x_6 + 3x_7$$

$$100 \geq 30x_1 + 50x_2 + 40x_3 + 10x_4 + 40x_5 + 30x_6 + 10x_7$$

$$x_i = 0 \text{ or } 1$$

a. By Theorem 2,

$$x_1 \rightarrow x_2, x_3, x_5, x_6$$

$$x_3 \rightarrow x_5$$

$$x_4 \rightarrow x_6, x_7.$$

b. By Theorem 3,

$$x_2 \rightarrow x_5$$

$$x_3 \rightarrow x_6$$

c. By Theorem 4,

$$x_4 \rightarrow x_5 \text{ (using } x_4 \text{ and } x_6)$$

The set of relationships from Theorem 2 are summarized in Figure 1. The entire set of relationships are summarized in Figure 2. Finally, the reduced domination networks, with cumulative c_i and a_i parameters for each i, are summarized in Figure 3.

Given the domination results, a smaller problem can be constructed.

By Theorem 5, $x_5 = x_6 = 0$. By Theorem 6, $x_1 = 1$.

The example problem has thus been reduced to,

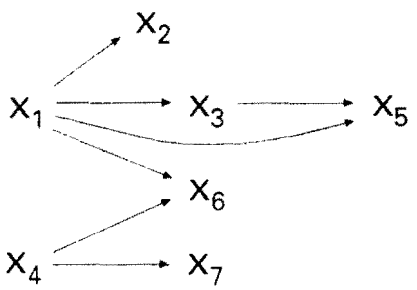

Figure 1. Example Application of Theorem 2.

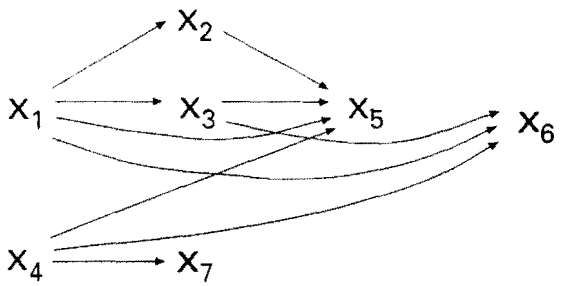

Figure 2. Example Application of Theorems 2, 3 and 4.

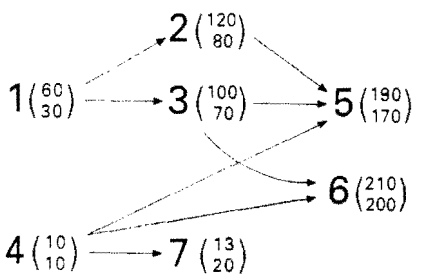

Figure 3. Reduced Domination Network for Example.

$$\max 60 + 60x_2 + 40x_3 + 10x_4 + 3x_7$$

$$70 \geq 50x_2 + 40x_3 + 10x_4 + 10x_7$$

$$x_i = 0 \text{ or } 1.$$

Thus, the dimensionality was reduced from 7 to 4 and the state space from 100 to 70.

IV. EMPIRICAL RESULTS

The effectiveness of the ability of domination to reduce problem size was investigated empirically and found to have a substantial impact on reducing the problem size. An algorithm version of the results of this paper was generated. The algorithm integrated the results of Sections II and III of this paper. Test problems were generated using a random number generator and the algorithm was used to determine the extent to which the problem size could be reduced.

A. Test Problems

A total of 80 knapsack problems were generated using the system, (0, 1) uniform random number generator. The problems were developed so that four sets of 20 problems were investigated. Problem size was either 10 or 20 variables. In each set of problems c_i was allowed to take on integer values from 1–100. Then, for the two problem sizes, a_i was allowed to take on integer values from 1–20 and 1–100. The b value was varied as $\text{INT}(\alpha \cdot \Sigma_{i=1}^{n} a_i)$ with α taking the values .1, .3, .5, .7, and .9, where $\text{INT}(x) = \lambda$ where $\lambda \leq x < \lambda + 1$ and λ is integer. The results are contained in Table 1. The average reduction in the number of variables varied from a low of 37.50% to a high of 82.50%.

B. Reduction Algorithm

An algorithm was programmed and used on the test problems to determine the number of variables that could be reduced from the problems. The algorithm consisted of sequentially generating the problem and then using Theorems 2, 3, 4, 5 and 6. Then another problem was generated, and so forth.

C. Findings

The findings are summarized in Table 1. Up to 82.5% of the variables were reduced, with the smallest reduction being 37.5%. Thus, reduction can be substantial.

The results were found to be a function of at least two different parameters. First, the relative value of b had a significant impact on the number of variables reduced. It was found that if b was either small or large compared to the total sum of the a_i values, a larger number of variables could be reduced. For example, the largest reduction in problem size occurred when α was .1 and .9. The least reduction occurred if b was close to the sum of the a_i values, divided by two.

Second, the range of the values c_i and a_i influenced the number of variables reduced. For example, only two of the ten categories of problems (e.g., $n = 10$ and $\alpha = .1$) resulted in greater reduction when the range was large (1–100) compared to when it was small (1–20).

Table 1. Reduction Results

n	α	a_i Range	*Problems completely solved*	*Average number of variables reduced*	*Average number of variables remaining*	*Average percent reduction*
10	.1	1–20	8	8.2	1.8	82.00
10	.3	1–20	4	6.3	3.7	63.00
10	.5	1–20	0	5.05	4.95	50.50
10	.7	1–20	3	6.1	3.9	61.00
10	.9	1–20	4	7.2	2.8	72.00
20	.1	1–20	1	13.75	6.25	68.75
20	.3	1–20	0	9.7	10.3	48.50
20	.5	1–20	0	8.65	11.35	43.25
20	.7	1–20	0	10.5	9.5	52.50
20	.9	1–20	1	14.85	5.15	74.25
10	.1	1–100	7	8.25	1.75	82.50
10	.3	1–100	1	4.7	5.3	47.00
10	.5	1–100	0	3.75	6.25	37.50
10	.7	1–100	0	4.75	5.25	47.50
10	.9	1–100	5	7.45	2.55	74.50
20	.1	1–100	0	12.2	7.8	61.10
20	.3	1–100	0	7.65	12.35	38.25
20	.5	1–100	0	7.55	12.45	37.75
20	.7	1–100	0	10.75	9.25	53.75
20	.9	1–100	1	14.55	5.45	72.75

In some cases, the reduction process was able to completely solve the problem. In 35 out of 400 problems (80 * 5 parameter variations on the b value) complete solutions were found.

V. SUMMARY AND EXTENSION

This section briefly summarizes and discusses an extension to the results in Part I.

A. Summary

This paper has provided results on how to determine domination in 0-1 knapsack problems. Results are provided for each of the three cases where $c_i \geq c_j$ and $a_i \leq a_j$; $c_i \geq c_j$ and $a_i \geq a_j$; and $a_i \leq a_j$ and $c_i \leq c_j$. The domination results are used to determine which variables must be 1 and which must be 0. Those result in reduced versions of the Problem (1) where b and n can be smaller.

B. Extension

Other domination results can be developed. The primary extension is for the case where the variables are not constrained to 0 and 1, but can take on any integer value. In those situations, an alternative approach to domination can account for the larger variable solutions.

Suppose that the requirement that $x_i = 0$ or 1 is dropped from the formulation (1). The following Theorem would replace Theorem 2.

Theorem 2′

$$\text{If } c_i \geq c_j \text{ and } a_i \leq a_j \text{ then } x_j = 0.$$

REFERENCES

Kolesar, P., "A Branch and Bound Algorithm for the Knapsack Problem," *Management Science*, Vol. 13, No. 9 (1967).

Weingartner, H., *Mathematical Programming and the Analysis of Capital Budgeting Problems*. Englewood Cliffs, NJ, Prentice-Hall, 1962.