



Received: 20 October 2015
Accepted: 27 February 2016
Published: 31 March 2016

*Corresponding author: Santosh Kumar, Department of Mathematics and Statistics, University of Melbourne, Parkville, Australia
E-mails: skumar@ms.unimelb.edu.au, Santosh.Kumarau@gmail.com.

Reviewing editor:
Akiko Yoshise, University of Tsukuba, Japan

Additional information is available at the end of the article

APPLIED & INTERDISCIPLINARY MATHEMATICS | RESEARCH ARTICLE

Knapsack constraint reformulation: A new approach that significantly reduces the number of sub-problems in the branch and bound algorithm

Elias Munapo¹ and Santosh Kumar^{2*}

Abstract: The paper presents a new approach to significantly reduce the number of sub-problems required to verify optimality in the branch and bound algorithm. The branch and bound algorithm is used to solve linear integer models and these models have application in areas such as scheduling, resource allocation, transportation, facility allocation and capital budgeting. The single constraint of the knapsack linear integer problem (KLIP) is reformulated in such a way that the number of standard branch and bound sub-problems required to verify optimality is significantly reduced. Computational results of the proposed approach on randomly generated KLIPs are also presented.

Subjects: Advanced Mathematics; Discrete Mathematics; Finite Mathematics; Mathematics & Statistics; Operational Mathematics; Science

Keywords: knapsack integer problem; reformulation; branch and bound; computational complexity

Subject classification codes: 90C10; 90C11; 90C27



Elias Munapo

ABOUT THE AUTHORS

Elias Munapo and Santosh Kumar have interest in discrete optimization methods and they have put forward several ideas that may provide further insight into discrete optimization and linear programming methodology. Their approaches need an appropriate software support, which is a challenge not only for the authors but for the field of optimization. They have made direct contributions to the field of integer programming and special models like quadratic assignment, travelling salesman, network routing by link weight modification. The current paper on knapsack constraint reformulation opens up many interesting questions that can only be answered provided this paper reaches researchers in discrete optimization. We believe, our work challenges software developers to raise funds, support university research for higher degrees. We can provide supervision, if required and hope universities will welcome such a move from the software companies.

PUBLIC INTEREST STATEMENT

Resource constraints often arise in all studies. Mathematical classification is generally based on intrinsic property of the situation. One such problem in optimization has been described by a generic name, “Knapsack Problem”, meaning that the knapsack has a finite limited capacity and our interest is to fit in that knapsack as much as possible from the available opportunities. Its applications arise in various areas. A typical knapsack constraint, which deals with only positive numbers, can be written as:

$$a_1x_1 + a_2x_2 + \dots + a_nx_n \leq b$$

Here, $a_1, a_2 + \dots + a_n \geq 0$ are the known amounts of resource that will be used per unit and $x_1, x_2 + \dots + x_n \geq 0$ represent the unknown integers to be determined with some objective to be achieved. This paper reformulates the above constraint and reduces the required computational effort. This reformulation may have applications in other areas, which requires further attention.

1. Introduction

The general linear integer programming (LIP) problem has many applications in real life, for example, they arise in set covering, travelling salesman, assignment, transportation, knapsack, capital budgeting, facility location, timetabling and airline scheduling. Many of these applications have been presented in Taha (2004) and Winston (2004), where these problems have been formulated as a linear integer programming (LIP) model. Because of real-life applications, see Chinneck (2004), the LIP model has attracted so much attention from researchers, yet a consistent and efficient general purpose method has not been developed. We are not aware of any polynomial time algorithm for the general LIP model. In fact, the LIP model is NP complete and a polynomial algorithm for the LIP is believed not to exist. This paper presents a new approach, which takes advantage of the presence of a single constraint to solve a knapsack linear integer problem (KLIP). The problem is reformulated in such a way that the number of standard branch and bound sub-problems required to verify optimality is significantly reduced. Computational results of the proposed approach on randomly generated KLIPs are also presented.

A mathematical statement of the knapsack problem is given in Section 2 and in Section 3, a few problems are discussed where the branch and bound (B & B) performance is poor. We have categorized these problems according to the possible source leading to poor performance. These problems have been subdivided in seven classes. Knapsack constraint reformulation is presented in Section . A summary of the computational experiments on randomly generated problems is presented in Section 5 and finally, the paper has been concluded in Section 6.

2. The knapsack linear integer problem

Maximize or Minimize $Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$,

such that:

$$a_1x_1 + a_2x_2 + \dots + a_nx_n \leq \text{ or } \geq b, \quad (1)$$

where a_j , b and c_j are given nonnegative constants, and $x_j \geq 0$ are integer valued quantities and $j = 1, 2, \dots, n$.

3. A collection of LIP models where the B & B method performs poorly

3.1. Standard branch and bound algorithm

The B & B algorithm was first proposed by Land and Doig (1960) for solving integer programs. The algorithm was further modified by Dakin (1965) to solve both pure and mixed integer programs. The B & B algorithm in general relies on the usual strategy of first relaxing the integer problem into a linear programming (LP) model. If the linear programming optimal solution is an integer then, the optimal solution to the integer problem has been obtained and the search concludes. If the LP optimal solution is not an integer, then a variable with a fractional value is selected to create two sub-problems such that part of the feasible region is discarded without eliminating any of the feasible integer solutions. The process is repeated on all variables with fractional values until an integer solution is found, see Bealie (1979), Beasley (1996), Mitchell and Lee (2001), Taha (2004) and Winston (2004). In this paper, the standard B & B refers to that version of the B & B algorithm proposed by Dakin (1965) and we assume that there are no state-of-art branching rules, cuts or pricing.

In the following, several problems have been presented where B & B performance is poor.

3.2. Complexity of the standard B & B method with numerical illustrations

The worst case complexity of the B & B algorithm is discussed for the LIP, which is NP Complete. The number of sub-problems can easily reach unmanageable levels even for very small problems. In this

section, we present some classes of the LIP models that cause serious challenges for the standard B & B algorithm.

Class 1: Knapsack binary linear problem, taken from Kumar, Munapo, and Jones (2007) .

$$\text{Maximize } Z = \sum_{i=1}^{n-1} x_i, \text{ or Minimize } Z = x_n$$

such that:

$$2 \sum_{i=1}^{n-1} x_i \pm x_n = n - 1$$

where $x_j = 0$ or $1 \forall j$ and n is even.

The behaviour of the standard branch and bound method for $n = 4, 6, 8, 16$ and 40 is given in Table 1.

Class 2: Step pattern formed by negative signs.

The second class is the integer problems that have constraints with negative signs forming a step pattern. Also, the coefficients in the objective function as well as the constants on the right-hand side of the constraints are significantly different. The standard B & B algorithm on this class of problems can behave in a very bizarre way if the branching is not properly managed. The following three numerical illustrations 3.2.1–3.2.2 were taken from Kumar et al. (2007). Numerical illustration 3.2.3 was slightly modified.

Numerical illustration 3.2.1

$$\text{Maximize } Z = 3x_1 + 5x_2 + 7x_3$$

such that:

$$4x_1 + 9x_2 - 8x_3 \leq 81,$$

$$5x_1 - 7x_2 + x_3 \leq 42,$$

$$-2x_1 + x_2 + 7x_3 \leq 10000,$$

where $x_1, x_2, x_3 \geq 0$ and integer.

Table 1. Complexity of the problem as n value increases

Value of n in the model	Number of sub-problems created by the standard branch and bound approach to reach the optimum solution
4	11
6	39
8	139
16	25,739
40	Computer was stopped after the sub-problems exceeded 30,000

Note the step formed by the minus signs and the fluctuating values of constants in the right-hand side. Using the standard B & B algorithm, it generated 209 sub-problems to verify optimality. The optimal solution is: $x_1 = 590, x_2 = 544, x_3 = 897$ and $Z = 10,769$.

Numerical illustration 3.2.2

$$\text{Maximize } Z = 11x_1 + 21x_2 + 17x_3 + 25x_4 + 15x_5$$

such that:

$$\begin{aligned} 10x_1 + 20x_2 + 15x_3 + 12x_4 - 3x_5 &\leq 789, \\ 5x_1 + 18x_2 + 21x_3 - 7x_4 + 25x_5 &\leq 678, \\ 12x_1 + 24x_2 - 10x_3 + 19x_4 + 13x_5 &\leq 290, \\ 24x_1 - 8x_2 + 18x_3 + 19x_4 + 13x_5 &\leq 1568, \\ -15x_1 + 22x_2 + 28x_3 + 16x_4 + 17x_5 &\leq 230, \end{aligned}$$

where $x_1, x_2, x_3, x_4, x_5 \geq 0$ and integer.

Using the standard B & B algorithm, it generated 605 sub-problems to verify optimality. The optimal solution is: $x_1 = 36, x_2 = 0, x_3 = 24, x_4 = 5, x_5 = 0$ and $Z = 929$.

Numerical illustration 3.2.3

$$\text{Maximize } Z = 5x_1 + 90x_2 + 12x_3 + 27x_4 + 56x_5 + 56x_6 + 23x_7 + 36x_8 + 8x_9 + 178x_{10},$$

such that:

$$\begin{aligned} -x_1 + x_2 + x_3 + x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} &\leq 95, \\ x_1 - 6x_2 + x_3 + x_4 + 56x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} &\leq 5679, \\ x_1 + x_2 - 5x_3 + x_4 + x_5 + x_6 + 16x_7 + 20x_8 + x_9 + x_{10} &\leq 1990, \\ 5x_1 + x_2 + x_3 - 8x_4 + x_5 + x_6 + x_7 + x_8 + x_9 + x_{10} &\leq 450, \\ x_1 + 19x_2 + x_3 + x_4 - 166x_5 + 3x_6 + x_7 + x_8 + x_9 + x_{10} &\leq 670, \\ x_1 + x_2 + x_3 + x_4 + x_5 - 12x_6 + x_7 + x_8 + x_9 + x_{10} &\leq 80, \\ x_1 + x_2 + x_3 + x_4 + 90x_5 + 8x_6 - x_7 + x_8 + 7x_9 + x_{10} &\leq 8887, \\ x_1 + x_2 + 34x_3 + 5x_4 + x_5 + x_6 + x_7 - 9x_8 + x_9 + x_{10} &\leq 68, \\ x_1 + x_2 + x_3 + 0x_4 + x_5 + 81x_6 + x_7 + x_8 - x_9 + 25x_{10} &\leq 350, \\ 23x_1 + x_2 + x_3 + x_4 + x_5 + 5x_6 + x_7 + x_8 + x_9 - 10x_{10} &\leq 523, \end{aligned}$$

where $x_1, x_2, \dots, x_{10} \geq 0$ and integer.

The standard B & B requires 2101 sub-problems to verify optimality, which is given by: $x_1 = 6, x_2 = 87, x_5 = 7, x_6 = 2, x_8 = 1, x_{10} = 3, x_3 = x_4 = x_7 = x_9 = 0$ and $Z = 8934$.

The other classes of linear integer problems that make the B & B an unreliable method are presented in Classes 3 to 7.

Class 3: Complementary constraints.

$$\text{Minimize } Z = x_1$$

such that:

$$\begin{aligned} a_1x_1 + a_2x_2 + \dots + a_jx_j + \dots + a_nx_n &\geq b \text{ and} \\ -(a_1x_1 + a_2x_2 + \dots + a_jx_j + \dots + a_nx_n) &\geq b, \end{aligned}$$

where $x_j \geq 0$ and integer, a_j and b are integers and constants, $j = 1, 2, \dots, n$ and $-\bar{a}_j < a_j \forall j \neq 1$. Also note that the first constraint in this class of problem contains both positive and negative coefficients. The two constraints are complementary in the sense that when added give a zero on the left-hand side. More examples are given as Class 3.1 and Class 3.2.

Class 3.1

Minimize $Z = x_1$,

such that:

$$\begin{aligned} 3x_1 + 6x_2 - 8x_3 &\geq 100, \\ -3x_1 - 6x_2 + 8x_3 &\geq 100 \end{aligned}$$

where $x_j \geq 0 \forall j$ and integer.

Class 3.2

Minimize $Z = x_1$

such that:

$$\begin{aligned} 4x_1 + 2x_2 - 9x_3 + 7x_4 - 5x_5 + 12x_6 &\geq 210 \\ -4x_1 - 2x_2 + 9x_3 - 7x_4 + 5x_5 - 12x_6 &\geq 210 \end{aligned}$$

where $x_j \geq 0 \forall j$ and integer.

Class 4: Binary knapsack problem.

This is an extension of class 1. The problem becomes more difficult for the standard B & B algorithm if it is slightly modified as given in below.

$$\text{Maximize } Z = \sum_{j=1}^{n-1} x_j$$

such that:

$$2 \sum_{j=1}^{n-1} x_j \pm \kappa x_n = n - 1$$

where $x_j = 0$ or $1 \forall j$, $\kappa \leq n - 1$, κ and n are even.

Class 4 has an alternate form also as was the case in Class 1. This alternate form will have the objective function as given below and constraints remain unchanged. This alternative objective is given by: Minimize $Z = x_n$.

For the Class 4 problem, the behaviour of the B & B method for $n = 4, 6, 8$ and 16 becomes worse as given in Table 2.

Class 5: Knapsack problem: A pure integer case.

Mere changing of variables from binary to pure integer makes any LIP worse for the standard B & B algorithm. For example, consider:

Table 2. Complexity of the problem as n value increases

Value of n in the model	Number of sub-problems created by the branch and bound approach to reach the optimum
4	$\kappa = 3$, Sub-problems =17
6	$\kappa = 3$, Sub-problems =59
	$\kappa = 5$, Sub-problems =59
8	$\kappa = 3$, Sub-problems =209
	$\kappa = 5$, Sub-problems =209
	$\kappa = 7$, Sub-problems =209
16	Computer was stopped after the number of sub-problems exceeded 30,000

$$\text{Maximize } Z = \sum_{j=1}^{n-1} x_j$$

such that:

$$2 \sum_{j=1}^{n-1} x_j + \kappa x_n = n - 1$$

where $x_j \geq 0$ and integer $\forall j, 1 \leq \kappa \leq n - 1, \kappa$ is odd and n is even.

Once again, an alternate problem for class 5 is when objective function is changed but constraints remain unchanged. This alternative objective function is given by:

$$\text{Minimize } Z = x_n.$$

The computational behaviour of the standard branch and bound method for $n = 4, 6, 8$ and 16 is given in Table 3.

Class 6: Hard Knapsack Problems.

$$\text{Maximize } Z = \sum_{j=1}^{n-1} x_j$$

Table 3. Complexity of the problem as n value increases

Value of n in the model	Number of sub-problems created by the branch and bound approach to reach the optimum solution
4	$\kappa = 3$, Sub-problems =23
6	$\kappa = 3$, Sub-problems =129
	$\kappa = 5$, Sub-problems =129
8	$\kappa = 3$, Sub-problems =755
	$\kappa = 5$, Sub-problems =755
	$\kappa = 7$, Sub-problems =755
16	Computer was stopped when the number of sub-problems exceeded 30,000

such that:

$$2 \sum_{j=1}^{n-1} x_j + \kappa x_n = \lambda,$$

where $x_j \geq 0$ and integer $\forall j$, $2(n-1) + \kappa = \lambda$, and $\kappa, \lambda \geq 0$ are odd and n even. Once again, an alternate problem can be for a minimizing objective function given by: Minimize: $Z = x_n$.

The standard B & B method cannot solve most of these problems for large values of κ . For example, a knapsack problem with the parameters: $n = 4$, $\kappa = 91$, and $\lambda = 97$ becomes

$$\text{Minimize } Z = x_4,$$

such that:

$$2x_1 + 2x_2 + 2x_3 + 91x_4 = 97$$

where $x_j \geq 0$ and integer $\forall j$.

The standard B & B method requires 7,449 sub-problems to verify the optimal solution. For large values of λ , the knapsack problems cannot be solved by standard B & B algorithm on its own.

Class 7: Hard pure integer models.

The following general integer model is also very difficult to solve by the standard B & B algorithm on its own.

$$\text{Minimize } Z = \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n$$

such that :

$$\begin{aligned} \alpha_{11}x_1 + \alpha_{12}x_2 + \dots + \alpha_{1n}x_n &\geq \beta_1 \\ \alpha_{21}x_1 + \alpha_{22}x_2 + \dots + \alpha_{2n}x_n &\geq \beta_2 \\ &\vdots \\ \alpha_{m1}x_1 + \alpha_{m2}x_2 + \dots + \alpha_{mn}x_n &\geq \beta_m \end{aligned}$$

where $\forall j$, $\omega_j = 1$ or some large number. There must be one large number in the objective row and $\forall j$, $\alpha_{ij} = 2$ or some large odd number. There must be one large odd number in every row and in every column. Further, it must satisfy $\alpha_{i1} + \alpha_{i2} + \dots + \alpha_{in} < \beta_i \forall i$.

A numerical illustration of the above model

$$\text{Minimize } Z = x_1 + 141x_2 + x_3$$

such that:

$$\begin{aligned} 55x_1 + 2x_2 + 2x_3 &\geq 73 \\ 2x_1 + 91x_2 + 2x_3 &\geq 97 \\ 2x_1 + 2x_2 + 85x_3 &\geq 99 \end{aligned}$$

where $x_1, x_2, x_3 \geq 0$ and integer.

A total of 653 standard B & B sub-problems are required to verify optimality. The optimal solution is: $x_1 = 45$, $x_2 = 0$, $x_3 = 4$ and $Z = 49$.

There are so many other classes of difficult integer models that occur in real life. These include the travelling salesman problem (TSP) and the generalized assignment problem (GAP). At the moment, it may be very difficult to come up with an efficient general purpose algorithm. Instead, it makes sense to study one class and then propose an efficient way of solving it. In this paper, the KLIP is targeted because of its special features.

4. Reformulation of the knapsack problem

The knapsack problem has special features that we can take advantage of for reformulation. This problem has only one constraint and all coefficients are nonnegative. The coefficients in the constraint of any KLIP can be arranged in ascending order i.e.

$$a_1x_1 + a_2x_2 + \dots + a_nx_n \text{ where } a_1 \leq a_2 \leq \dots \leq a_n. \quad (2)$$

This is possible for all KLIP models since all coefficients are nonnegative, and since a_1 is the smallest one can rewrite the expression (2) as follows:

$$a_1x_1 + a_1x_2 + \dots + a_1x_n + (a_2 - a_1)x_2 + (a_3 - a_1)x_3 + \dots + (a_n - a_1)x_n$$

Let $a_j^1 = a_j - a_1$ for $j = 2, 3, \dots, n$ then (2) can be expressed as given by (3).

$$(a_1x_1 + a_1x_2 + \dots + a_1x_n) + a_2^1x_2 + a_2^1x_3 \dots + a_2^1x_n, \text{ where } a_2^1 \leq a_3^1 \dots \leq a_n^1 \quad (3)$$

Repeating the process given in (3), where smallest coefficient is a_2^1 , we obtain

$$\begin{aligned} &(a_1x_1 + a_1x_2 + \dots + a_1x_n) + a_2^1x_2 + a_2^1x_3 \dots + a_2^1x_n \\ &\quad + (a_3^1 - a_2^1)x_3 + (a_4^1 - a_2^1)x_4 + \dots + (a_n^1 - a_2^1)x_n, \\ &\quad \text{where } a_2^1 \leq a_3^1 \dots \leq a_n^1 \end{aligned} \quad (4)$$

If the process repeated until there is no term left, the transformed constraint will become as shown in (5).

$$\begin{aligned} &(a_1x_1 + a_1x_2 + \dots + a_1x_n) + a_2^1x_2 + a_2^1x_3 \dots + a_2^1x_n \\ &\quad + a_3^2x_3 + a_3^2x_4 + \dots + a_3^2x_n + \dots + a_n^kx_n, \end{aligned} \quad (5)$$

$$a_1x_1 + a_1x_2 + \dots + a_1x_n = a_1y_1$$

$$a_2^1x_2 + a_2^1x_3 \dots + a_2^1x_n = a_2^1y_2$$

$$a_3^2x_3 + a_3^2x_4 + \dots + a_3^2x_n = a_3^2y_3$$

...

$$a_n^kx_n = a_n^ky_k \quad (6)$$

Note, y_j are integer valued quantities and $j = 1, 2, \dots, k$

The KLIP becomes

$$\text{Maximize or Minimize } Z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

such that:

$$a_1x_1 + a_1x_2 + \dots + a_1x_n = a_1y_1$$

$$a_2^1x_2 + a_2^1x_3 \dots + a_2^1x_n = a_2^1y_2$$

$$a_3^2x_3 + a_3^2x_4 + \dots + a_3^2x_n = a_3^2y_3$$

...

$$a_n^kx_n = a_n^ky_k$$

$$a_1y_1 + a_2^1y_2 + \dots + a_n^ky_n \leq \text{or } \geq b \quad (7)$$

Once again, y_j are integer value quantities and $j = 1, 2, \dots, k$.

Phase 1: Solve reformulated problem (7) using the standard B & B method with the integral restriction on y_j only. If solution is an integer, then it is also optimal to the original KLIP, else go to Phase 2.

Phase 2: Continue with the standard B & B method used in Phase 1 but this time with the integral restriction extended to x_j also.

This reformulated model is easier to solve than the original KLIP. The standard B & B method takes a significantly smaller number of sub-problems to verify optimality on the reformulated model than the original KLIP. The proposed approach works better if $a_1, a_2^1, a_3^2, \dots, a_n^k$ are different.

Numerical Illustrations 4.1

$$\text{Minimize } Z = 20x_1 + 8x_2 + 3x_3 + 5x_4 + 33x_5,$$

such that:

$$29x_1 + 20x_2 + 18x_3 + 24x_4 + 12x_5 \geq 679 \tag{8}$$

where $x_j \geq 0$ and integer $\forall j$.

The standard B & B method takes 231 sub-problems to verify the optimal solution of (8), which is given by: $x_1 = x_2 = 0, x_3 = 38, x_4 = x_5 = 0$ and $Z = 114$.

Note the coefficients, when arranged in an increasing order are given by 12, 18, 20, 24 and 29. Thus, we have $a_1 = 12, a_2^1 = 18 - 12 = 6, a_3^2 = 20 - 12 - 6 = 2, a_4^3 = 24 - 12 - 6 - 2 = 4$ and $a_5^4 = 29 - 12 - 6 - 2 - 4 = 5$. After the reformulation, the KILP (8) becomes (9).

$$\text{Minimize } Z = 20x_1 + 8x_2 + 3x_3 + 5x_4 + 33x_5,$$

Such that :

$$\begin{aligned} 12y_1 + 6y_2 + 2y_3 + 4y_4 + 5y_5 &\geq 679 \\ 12(x_1 + x_2 + x_3 + x_4 + x_5) &= 12y_1 \\ 6(x_1 + x_2 + x_3 + x_4) &= 6y_2 \\ 2(x_1 + x_2 + x_4) &= 2y_3 \\ 4(x_1 + x_4) &= 4y_4 \\ 5x_1 &= 5y_5 \end{aligned} \tag{9}$$

Note that the coefficients 12, 6, 2, 4 and 5 are different.

Phase 1: Where $y_j \geq 0$ and integer $\forall j$. In phase 1, there is no integral restriction on the variables x_j . The number of sub-problems necessary to verify the optimal solution reduces to 37, resulting once again in the solution given by: $x_3 = y_1 = y_2 = 38, x_1 = x_2 = x_4 = x_5 = y_3 = y_4 = y_5 = 0$ and $Z = 114$.

Note that Phase 2 was not necessary since Phase 1 solution was optimal and the number of sub-problems required to reach the optimal solution dropped from 231 to 37, which is a significant reduction. This shows that the reformulation is effective for the KLIP. Here are two more illustrations before we present a summary of computational experiments.

Numerical illustration 4.2

Table 4. Computational experiments on randomly generated Knapsack problems

KLIP S No.	No. of Variables	No. of sub-problems before	No. of sub problems after	% Reduction in sub-problems
1	5	25	4	84.0
2	10	41	7	82.9
3	20	158	29	81.6
4	30	132	12	90.9
5	40	89	9	80.0
6	50	*	32	-
7	60	289	18	93.7
8	70	6590	47	99.3
9	80	2198	29	98.7
10	90	1789	15	99.2
11	100	691	106	84.7
12	150	11209	89	99.2
13	200	721	19	97.4
14	250	*	51	-
15	300	2187	71	96.8
16	350	6842	29	99.6
17	400	895	67	92.5
18	450	17866	27	99.8
19	500	*	105	-
20	550	96	11	88.5
21	600	14	3	78.6
22	650	871	69	92.1
23	700	956	17	98.2
24	750	4008	189	95.3
25	800	38	2	94.7
26	850	2983	63	97.9
27	900	345	22	93.6
28	950	27689	58	99.8
29	1000	116	26	77.6
30	1500	19	4	78.9
31	2000	4578	23	99.5
32	2500	89	13	85.4
33	3000	8104	269	96.7
34	3500	978	47	95.2
35	4000	67	15	77.6
36	4500	*	44	-
37	5000	24672	56	99.8
38	6000	*	391	-
39	6500	897	35	96.1
40	7000	1946	219	88.7

*Number of sub-problems exceeded 30,000.

Table 5. More computational experiments on randomly generated Knapsack problems

KLIP S No.	No. of variables	No. of sub-problems before	No. of sub problems after	% Reduction in sub-problems
41	7500	*	456	-
42	8000	15008	375	97.5
43	8500	6895	132	98.1
44	9000	*	249	-
45	9500	1193	83	93.1
46	10000	5829	683	88.3
47	10500	*	43	-
48	12000	*	102	-
49	13000	28972	219	99.2
50	15000	*	147	-

*Number of sub-problems exceeded 30,000.

Minimize $Z = x_4$,

Such that :

$$2x_1 + 2x_2 + 2x_3 + 91x_4 = 97 \tag{10}$$

where $x_j \geq 0$ and integer $\forall j$.

This KLIP is a special case from Class 6. The number of sub-problems reduces from 7449 to only 5 after reformulation. Optimal solution was obtained in Phase 1.

Numerical illustration 4.3

Minimize $Z = x_n$,

such that:

$$2 \sum_{i=1}^{n-1} x_i \pm x_n = n - 1, \tag{11}$$

where $x_j \geq 0$ and integer $\forall j$ and $n = 16$.

This KLIP comes from Class 1 but in this case, variables are integer and not necessarily binary. The number of sub-problems reduces from over 30, 000 to only 11 after reformulation and the optimal solution was obtained in Phase 1.

5. Computational experiments

Fifty randomly generated knapsack problems of different sizes were used in the analysis. The objective of the computational experiments was to determine whether the number of sub-problems decrease after the reformulation. The computational results were tabulated as given in Tables 4 and 5. MATLAB R2013 (version 8.2) running on an Intel Pentium Dual desktop (Dual core G2020 2.9 GHz CPU, 2GB DDR3 1333 RAM) was used for the computational experiments. In all the fifty cases, it was observed that the number of standard B & B sub-problems decreased significantly after reformulation, as can be seen from the results in Tables 4 and 5.

6. Conclusions

What has emerged from the computational experiments is that the number of standard B & B sub-problems required to verify optimality can be significantly reduced by reformulation. In all the fifty cases analysed, the number of sub-problems were significantly reduced after reformulation. So many improvements have been done on the branch and bound algorithm in terms of addition of cuts to get the branch and cut algorithm (Brunetta, Conforti, & Rinaldi, 1997; Mitchell, 2001; Padberg & Rinaldi, 1991), pricing to get the branch and price algorithm (Barnhart, Johnson, Nemhauser, Savelsbergh, & Vance, 1998; Salvetsbergh, 1997) and also combining the two improved versions to get the branch cut and price hybrid algorithm (Barnhart, Hane, & Vance, 2010; Fukasawa et al., 2006; Ladanyi, Ralphs, & Trotter, 2001). Here, in this paper, we have we have achieved improvement in the standard B & B algorithm on its own before combining it with other approaches. In addition to using cuts and pricing within the context of a branch and bound algorithm, preprocessing given in Savelsbergh (1994) can reduce the number of sub-problems needed to verify optimality. Reformulation proposed in this paper significantly reduces the number of B & B sub-problems required to verify optimality.

In subsequent publications, attempt will be made to:

- (1) Extend the proposed constraint reformulation to the general LIP model, when applicable.
- (2) Search for the mathematical reasons that give rise to the efficiency observed.
- (3) Extend the reformulation concept to other situations.

Acknowledgements

Authors are grateful to the referees for their constructive suggestions for improving this paper.

Funding

The authors received no direct funding for this research.

Author details

Elias Munapo¹

E-mail: emunapo@gmail.com

Santosh Kumar²

E-mails: skumar@ms.unimelb.edu.au, Santosh.Kumarau@gmail.com

ORCID ID: <http://orcid.org/0000-0001-9321-5622>

¹ Graduate School of Business and Leadership, University of KwaZulu-Natal, Westville Campus, Durban, South Africa.

² Department of Mathematics and Statistics, University of Melbourne, Parkville, Australia.

Citation information

Cite this article as: Knapsack constraint reformulation: A new approach that significantly reduces the number of sub-problems in the branch and bound algorithm, Elias Munapo & Santosh Kumar, *Cogent Mathematics* (2016), 3: 1162372.

References

- Barnhart, C., Johnson, E. L., Nemhauser, G. L., Savelsbergh, M. W. P., & Vance, P. H. (1998). Branch and price column generation for solving huge integer programs. *Operations Research*, 46, 316–329.
- Barnhart, C., Hane, C. A., & Vance, P. H. (2010). Using branch-and-price-and-cut to solve origin–destination integer multicommodity flow problems. *Operations Research*, 48, 318–326.
- Bealie, E. M. L. (1979). Branch and bound methods for mathematical programming systems. *Annals of Discrete Mathematics*, 5, 201–219.
- Beasley, J. E. (Ed.). (1996). *Advances in linear and integer programming*. New York, NY: Oxford University Press.
- Benders, J. F. (1962). Partitioning procedures for solving mixed variables programming problems. *Numerische Mathematik*, 4, 238–252.
- Brunetta, L., Conforti, M., & Rinaldi, G. (1997). A branch and cut algorithm for the equicut problem. *Mathematical Programming*, 78, 243–263.
- Chinneck, J. W. (2004). *Practical optimization: A gentle introduction lecture notes*. Retrieved from <http://www.sce.carleton.ca/faculty/chinneck/po/Chapter13.pdf>
- Dakin, R. J. (1965). A tree search algorithm for mixed integer programming problems. *The Computer Journal*, 8, 250–255.
- Fukasawa, R., Longo, H., Lysgaard, J., Poggi de Aragao, M., Uchoa, E., & Werneck, R. F. (2006). Robust branch-and-cut-price for the capacitated vehicle routing problem. *Mathematical Programming Series A*, 106, 491–511.
- Kumar, S., Munapo, E., & Jones, B. C. (2007). An integer equation controlled descending path to a protean pure integer program. *Indian Journal of Mathematics*, 49, 211–237.
- Land, A. H., & Doig, A. G. (1960). An automatic method for solving discrete programming problems. *Econometrica*, 28, 497–520.
- Ladanyi, L., Ralphs, T. K., & Trotter, L. E. (2001). Branch, cut and price: Sequential and parallel. In N. Naddef & M. Jenger (Eds.), *Computational combinatorial optimization* (pp. 223–260). Berlin: Springer.
- Mitchell, J. E., & Lee, E. K. (2001). Branch and bound methods for integer programming. In C. A. Floudas & P. M. Pardalos (Eds.), *Encyclopedia of optimization* (pp. 329–347). Dordrecht: Kluwer Academic.
- Mitchell, J. E. (2001). Branch and cut algorithms for integer programming. In C. A. Floudas & P. M. Pardalos (Eds.), *Encyclopedia of optimization* (pp. 295–309). Dordrecht: Kluwer Academic.
- Padberg, M., & Rinaldi, G. (1991). A branch and cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33, 60–100.
- Savelsbergh, M. W. P. (1994). Preprocessing and probing techniques for mixed integer programming problems. *ORSA Journal on Computing*, 6, 445–454.
- Salvetsbergh, M. W. P. (1997). A branch and price algorithm to solve the generalized assignment problem. *Operations Research*, 45, 381–841.
- Taha, H. A. (2004). *Operations research: An introduction*. (7th ed.). Upper Saddle River, NJ: Pearson Educators.
- Winston, W. L. (2004). *Operations research applications and algorithms*. (4th ed.). Belmont, CA: Duxbury Press.



© 2016 The Author(s). This open access article is distributed under a Creative Commons Attribution (CC-BY) 4.0 license.

You are free to:

Share — copy and redistribute the material in any medium or format

Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made.

You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.

No additional restrictions

You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.



Cogent Mathematics (ISSN: 2331-1835) is published by Cogent OA, part of Taylor & Francis Group.

Publishing with Cogent OA ensures:

- Immediate, universal access to your article on publication
- High visibility and discoverability via the Cogent OA website as well as Taylor & Francis Online
- Download and citation statistics for your article
- Rapid online publication
- Input from, and dialog with, expert editors and editorial boards
- Retention of full copyright of your article
- Guaranteed legacy preservation of your article
- Discounts and waivers for authors in developing regions

Submit your manuscript to a Cogent OA journal at www.CogentOA.com

