International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 5, 1140–1153, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.5-090

Computational Enhancement in the Application of the Branch and Bound Method for Linear Integer Programs and Related Models

Masar Al-Rabeeah
School of Mathematical and Geospatial Sciences
RMIT University, Melbourne, Australia
Department of Mathematics, College of Sciences
University of Basrah, Al- Basrah, Iraq

Elias Munapo
School of Economic and Decision Sciences
North West University, Mafikeng Campus, South Africa

Ali Al-Hasani
School of Mathematical and Geospatial Sciences
RMIT University, Melbourne, Australia.
Department of Mathematics, College of Sciences
University of Basrah, Al- Basrah, Iraq

Santosh Kumar
School of Mathematical and Geospatial Sciences
RMIT University, Melbourne, Australia
Department of Mathematics and Statistics
University of Melbourne, Melbourne, Australia
Corresponding author: santosh.kumarau@gmail.com

Andrew Eberhard
School of Mathematical and Geospatial Sciences
RMIT University, Melbourne, Australia

(Received April 3, 2019; Accepted June 10, 2019)

Abstract
In this paper, a reformulation that was proposed for a knapsack problem has been extended to single and bi-objective linear integer programs. A further reformulation by adding an upper bound constraint for a knapsack problem is also proposed and extended to the bi-objective case. These reformulations significantly reduce the number of branch and bound iterations with respect to these models. Numerical illustrations have been presented and computational experiments have been carried out to compare the behaviour before and after the reformulation. For this purpose, Tora software was used.

Keywords- Reformulation, Branch and bound, General linear integer programs, Knapsack problem, Bi-objective models, Non-dominated point set.

1. Introduction

Many applications in real-life problems require integer restricted variables in a linear program. Many approaches have been developed to reduce computational effort and time required to get to the optimal solution. Some of the well-known approaches for integer programming problems are

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 5, 1140–1153, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.5-090

by Kelley (1960), Mitten (1970), Taha (2003), Kumar et al. (2010), Kumar and Munapo (2012). For larger integer programs, see the column generation procedure by Barnhart et al. (2018).

One of the well-known integer linear programming problem in discrete optimization is a knapsack problem, which has also been attempted by several researchers in Kellerer et al. (2004), Taha (2003), Della Croce et al. (2017) and real-life applications have been discussed in Winston and Goldberg (2004). A few well-known methods that have been used to solve the knapsack problem are based on dynamic algorithm (Chebil and Khemakhem, 2015), greedy algorithms (Chu and Beasley, 1998) and the branch and bound concept by Davis et al. (1971). For other approaches to a knapsack model, see Dudziński and Walukiewicz (1987), and Haristakeva and Shrestha (2005).

The standard branch and bound algorithm have been improved by using the concept branch and cut algorithm (Padberg and Rinaldi, 1991), which is a combination of the ideas from branch and bound and Gomory cutting plane (Kelley, 1960). Branch and Price (Barnhart et al., 2000) is another improvement in the branch and bound approach for solving the knapsack problem.

Instead of modifying these existing methods to solve a knapsack problem, Munapo and Kumar (2016) suggested a reformulation of the knapsack model itself. This reformulation resulted in a significant reduction in the number of sub-problems used by the standard branch and bound approach to reach the optimal solution.

In this paper, the knapsack reformulation by Munapo and Kumar (2016) has been applied to single and bi-objective general linear integer programs (LIP). Furthermore, the knapsack model is slightly modified resulting in further reduction in branch and bound iterations. These reformulations resulted in:

- Additional constraints and variables yet significantly reduce the number of branch and bound iterations.
- The modification with respect to the knapsack model adds only one extra constraint that is identified by a few simple steps discussed in the paper.
- The computational experiments indicate significant savings in the number of sub-problems required by branch and bound algorithm to verify the optimal solution.
- The efficiency of these new ideas have been investigated by implementing them to solve a few bi-objective models. The ϵ - constraint method (Chankong and Haimes, 2008) has been used to solve the bi-objective models before and after the reformulation. Numerical examples have been used to demonstrate the enhancement.

This paper attempts to:

(i) Extend the reformulation for the knapsack model by Munapo and Kumar (2016) to the general linear integer programming model.
(ii) Propose a new modification by adding an upper-bound constraint for the knapsack model.
(iii) Extend the above two ideas for a few bi-objective models.

The paper has been organized as follows: In Section 2, the preliminaries have been presented. In Section 3, the original knapsack reformulation by Munapo and Kumar (2016) is briefly explained. In Section 4, the knapsack reformulation by Munapo and Kumar (2016) has been extended for a general linear integer program. The proposed new modification for the knapsack problem has been

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 5, 1140–1153, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.5-090

discussed in Section 5. The above two modifications have been applied to bi-objective models in Section 6. Finally, the paper has been concluded in Section 7.

2. Formulation of General Integer and Knapsack Problem

A mathematical model of a general linear integer problem is given by (1):

$$z = \max\left\{\sum_{j=1}^{n} c_j x_j\right\} \tag{1}$$

Subject to

$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i \qquad (i = 1,2,\dots,m)$$

$$a_{ij}\ b_i, c_j\ \geq 0 \text{ and integer.}$$

Here $a_{ij}, b_i \text{ and } c_j$ are positive constants and x_j is an integer restricted variable, $j = 1,2,\dots,n$. The knapsack problem is a special case of (1) with a single constraint as given in (2)

$$z = \min\left\{\sum_{j=1}^{n} c_j x_j\right\} \tag{2}$$

Subject to

$$\sum_{j=1}^{n} a_j x_j \geq b$$

Again $a_j, b \text{ and } c_j$ are positive constants and $x_j \geq 0$ is an integer restricted variable, $j = 1,2,\dots,n$.

In a given LP model, since variables can be easily renamed, and constraints can be rearranged, for convenience of presentation and without any loss of generality, it is assumed that the 1st constraint in (1) satisfies the condition:

$$a_{11} < a_{12} < \cdots < a_{1n} \tag{3}$$

A more general case of (3) can be as given by (4):

$$a_{11} \leq a_{12} \leq \cdots \leq a_{1n} \tag{4}$$

which is also considered in this paper.

Similarly, for convenience of presentation, it has been assumed that the knapsack constraint in (2) also satisfies the condition

$$a_1 < a_2 < \cdots < a_n \tag{5}$$

3. The Knapsack Reformulation by Munapo and Kumar (2016)

3.1 The Reformulation

Munapo and Kumar (2016) converted $a\ (1Xn)$ Knapsack model into $a\ ((n+1)X(2n))$ model with the following substitutions:

$a_1^* = a_1, a_2^* = a_2 - a_1, a_3^* = a_3 - a_2, \dots, a_n^* = a_n - a_{n-1}$.

The original knapsack constraint in (2) can be expressed as:

$a_1^* y_1 + a_2^* y_2 + \dots + a_n^* y_n \geq b$.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 5, 1140–1153, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.5-090

Here, $y_1, y_2, \ldots, y_n$ are new integer restricted variables given by:

$$a_1^*(x_1 + x_2 + \ldots + x_n) = a_1^* y_1$$
$$a_2^*(x_2 + \ldots + x_n) = a_2^* y_2$$
$$\vdots$$
$$a_n^*(x_n) = a_n^* y_n.$$

Thus, the reformulation model of (2) will become:

$$z = \min\left\{\sum_{j=1}^{n} c_j x_j\right\} \tag{6}$$

Subject to

$$a_1^* y_1 + a_2^* y_2 + \ldots + a_n^* y_n \geq b$$
$$a_1^*(x_1 + x_2 + \ldots + x_n) = a_1^* y_1$$
$$a_2^*(x_2 + \ldots + x_n) = a_2^* y_2$$
$$\vdots$$
$$a_n^*(x_n) = a_n^* y_n.$$

The quantities a_j^*, c_j, b are positive integer constants, y_j are integer restricted variables, and $x_j \geq 0, j = 1,2,\ldots,n.$

3.2 Characteristic of the Model (6)

From the model (6), following observations are immediate:

(1) Dimension of the given knapsack problem after the reformulation changes from $(1 \times n)$ to $((n + 1) \times 2n)$.

(2) Since the relationship between the given variables x_j and the new variables y_j is in a canonical form, therefore if y_j are integer restricted values, x_j must also take only integer values without imposing any integer restriction on them.

(3) The above property, in spite of increase in the dimension of the knapsack problem, is reducing the number of the branch and bound iterations. It is this relationship, we wish to explore for the general linear integer program (1) and other related models.

4. The Proposed Reformulation for a General Linear Integer Program

4.1 Consideration of a LIP Model

The main idea of the knapsack reformulation by Munapo and Kumar (2016) is extended to a general linear integer programming model. Let us reconsider the general LIP model (1), where without any loss of generality, we assumed that the first constraint satisfies the condition (3). After implementing the reformulation in the 1st constraint, the model (1) becomes:

$$z = \max\left\{\sum_{j=1}^{n} c_j x_j\right\} \tag{7}$$

Subject to

$$a_{11}^* y_1 + a_{12}^* y_2 + \ldots + a_{1n}^* y_n \leq b_1$$
$$a_{11}^*(x_1 + x_2 + \ldots + x_n) = a_{11}^* y_1$$
$$a_{12}^*(x_2 + \ldots + x_n) = a_{12}^* y_2$$
$$\vdots$$
$$a_{1n}^*(x_n) = a_{1n}^* y_n$$
$$\sum_{j=1}^{n} a_{ij} x_j \leq b_i, \quad i = 2,\ldots,m.$$

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 5, 1140–1153, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.5-090

Here a_{ij}^{*}, c_j, b_i are positive integer constants with x_j real and y_j integer restricted variables, $j = 1,2,\ldots,n$.

Later, the case when $a_{11} \leq a_{12} \leq \ldots \leq a_{1n}$ has also been considered. A numerical example is presented for each case.

4.2 Numerical Illustrations

***Numerical Example* 1**

Model (8) is a trivial example with 5 variables where the condition (3) is satisfied by the first constraint.

$$\max\{19x_1 + 9x_2 + 4x_3 + 11x_4 + 13x_5\} \quad (8)$$

Subject to

$$7x_1 + 9x_2 + 10x_3 + 15x_4 + 18x_5 \leq 99$$
$$17x_1 + 10x_2 + 14x_3 + 8x_4 + 11x_5 \leq 125$$
$$x_j \geq 0 \text{ and integers, } j = 1,2,3,4,5$$

The first constraint in (8) is replaced by:

$$7y_1 + 2y_2 + y_3 + 5y_4 + 3y_5 \leq 99$$
$$7(x_1 + x_2 + x_3 + x_4 + x_5) = 7y_1$$
$$2(x_2 + x_3 + x_4 + x_5) = 2y_2$$
$$1(x_3 + x_4 + x_5) = 1y_3$$
$$5(x_4 + x_5) = 5y_4$$
$$3(x_5) = 3y_5$$

where $x_j \geq 0$ and $y_j \geq 0$ and integer for j = 1,2,3,4,5.

The mathematical model (8) will become as (9):

$$\max\{19x_1 + 9x_2 + 4x_3 + 11x_4 + 13x_5\} \quad (9)$$

Subject to

$$7y_1 + 2y_2 + y_3 + 5y_4 + 3y_5 \leq 99$$
$$17x_1 + 10x_2 + 14x_3 + 8x_4 + 11x_5 \leq 125$$
$$7(x_1 + x_2 + x_3 + x_4 + x_5) = 7y_1$$
$$2(x_2 + x_3 + x_4 + x_5) = 2y_2$$
$$1(x_3 + x_4 + x_5) = 1y_3$$
$$5(x_4 + x_5) = 5y_4$$
$$3(x_5) = 3y_5$$

$$x_j \geq 0 \text{ , } y_j \geq 0 \text{ and integers for j = 1,2,3,4,5.}$$

• The models (8) and (9) were solved using the Tora software and it was noted that (8) required 81 sup-problems to get to the optimal solution while the model (9) required only 15 sup-problems.

• The second constraint also satisfies the condition (3) with respect to variables x_4, x_2, x_5, x_3 and x_1 since $a_{24} (= 8) \leq a_{22} (= 10) \leq a_{25} (= 11) \leq a_{23} (= 14) \leq a_{21} (= 17)$.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 5, 1140–1153, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.5-090

It was noted that when the second constraint in model (8) was subject to similar transformation instead of constraint (1) and solved by Tora software, it required 27 sup-problems to reach to the optimal solution.

• Although, the integer restriction on y_j variables automatically result in integer values of x_j variables, however, if integer restrictions are imposed on both x_j and y_j then the number of sub problems required by Tora software increases to 56 sup-problems.

***Numerical Example* 2**
This is a slight modification of (8) to illustrate the condition (4).

$$\max\{19x_1 + 9x_2 + 4x_3 + 11x_4 + 13x_5\ \} \quad (10)$$

Subject to

$$9x_1 + 10x_2 + 10x_3 + 15x_4 + 18x_5 \leq 99$$
$$17x_1 + 10x_2 + 14x_3 + 8x_4 + 11x_5 \leq 125$$
$$x_j \geq 0 \text{ and integers, } j = 1,2,3,4,5$$

It may be noted that in the first constraint, coefficient of variables x_2 and x_3 are equal to 10. This will make a few minor changes in the reformulated model as given by (11):

$$\max\{19x_1 + 9x_2 + 4x_3 + 11x_4 + 13x_5\ \} \quad (11)$$

Subject to

$$9y_1 + y_2 + 5y_3 + 3y_4 \leq 99$$
$$17x_1 + 10x_2 + 14x_3 + 8x_4 + 11x_5 \leq 125$$
$$9(x_1 + x_2 + x_3 + x_4 + x_5) = 9y_1$$
$$(x_2 + x_3 + x_4 + x_5) = y_2$$
$$5(x_4 + x_5) = 5y_3$$
$$3(x_5) = 3y_4$$

$$x_j \geq 0\ , j = 1,4,5 \text{ and } 2 \text{ or } j = 1,4,5 \text{ and } 3,$$
$$y_j \geq 0 \text{ and integers for j = 1,2,3,4.}$$

Since the coefficients of x_2, x_3 were equal, it has resulted in reduction in one of the y_j variable and instead increased integer requirements on one of the x_j variables that is either x_2 or x_3.

Once again, model (10) required 56 sub-problem while model (11) required 12 sub-problems only.

4.3 The Computational Experiments

The efficiency of the proposed reformulation using Tora software (Taha, 2003) was investigated on different sized randomly generated instances up to 20 variables. Larger than 20 variables became time consuming as only manual input is possible in Tora software, see Table 1.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 5, 1140–1153, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.5-090

Table 1. Comparison with respect to the number of sub-problems before and after the reformulation using the branch and bound on Tora software

Problem	Optimal solution	Before Ref.	After Ref.	% Reduction
Instance1(5)	141	81	15	81.5
Instance2(5)	116	29	3	89.6
Instance3(5)	187	142	22	84.5
Instance4(10)	483	39	9	76.9
Instance5(10)	132	71	2	97.2
Instance6(10)	185	237	19	91.9
Instance7(15)	292	73	16	78.08
Instance8(15)	378	143	27	81.1
Instance9(20)	1288	230	38	83.5
Instance10(20)	1637	436	45	89.7

5. A New Modification by Adding an Upper Bound Constraint for the Knapsack Model

5.1 The Proposed Modification of the Knapsack Model

The main idea behind the modification is to identify a variable j that gives the minimum value of the objective function z denoted by z_j and add the corresponding condition as an upper bound constraint to the given knapsack model. This variable can be identified using the following steps:

(1) $*$ From the given problem (2) the number of integer variables is a known integer value let it be denoted by n.
$*$ Assign $j = 1$
$*$ Find $\frac{b}{a_j} = k_j$ where k_j is the next integer value and find $z_j = c_j k_j$.

(2) Put $j = j + 1$

(3) If $j < n$ find z_j otherwise go to step 4.

(4) Find the min $\{z_1, z_2, \ldots, z_n\}$ and let the minimum be denoted by z_r

(5) Substitute $x_r = k_r - \bar{x}_r$ in (2) then we get:

$$z = min\{c_1x_1 + c_2x_2 \ldots + c_{r-1}x_{r-1} - c_r\,\bar{x}_r + c_{r+1}x_{r+1} + \ldots + c_nx_n + c_rk_r\} \qquad (12)$$

Subject to

$$a_1x_1 + a_2x_2 \ldots + a_{r-1}x_{r-1} - a_r\,\bar{x}_r + a_{r+1}x_{r+1} + \ldots + a_nx_n + a_rx_r \geq b - a_rc_r$$

$x_j \geq 0$ and integer, $j \neq r$, $j = 1,2,\ldots m$, $k_r \geq \bar{x}_r \geq 0$ and integer.

Numerical Example 3

Once again, a trivial example with 5 variables, which is the same that was used by Munapo and Kumar (2016) is considered since it satisfies condition (5).

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 5, 1140–1153, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.5-090

$$z = \min\{20x_1 + 8x_2 + 3x_3 + 5x_4 + 33x_5\} \quad (13)$$

Subject to

$$29x_1 + 20x_2 + 18x_3 + 24x_4 + 12x_5 \geq 679$$
$$x_j \geq 0 \text{ and integer, } j = 1,2,3,4,5$$

Using the simple steps as explained in Section (5.1)

(1) $n = 5$
(2) The range of each variables is given by x_1: $[0 - 24]$, $x_2 = [0 - 34]$, $x_3 = [0 - 38]$, $x_4 = [0 - 29]$, $x_5 = [0 - 57]$.
(3) This gives: Z_1 = 480, Z_2= 272, Z_3= 114, Z_4 = 145, Z_5= 684.
(4) The minimum value Z_3 = 114, which means x_3 is replaced by this substitution

$$x_3 = 38 - \bar{x}_3 \quad (14)$$

(5) Substitute (14) in problem (13) we get

$$\min\{20x_1 + 8x_2 - 3\bar{x}_3 + 5x_4 + 33x_5 + 114\} \quad (15)$$

Subject to

$$29x_1 + 20x_2 - 18\bar{x}_3 + 24x_4 + 12x_5 \geq -5$$
$$x_j \geq 0 \text{ , and integers for j = 1,2,4,5.}$$
$$0 \leq \bar{x}_3 \leq 38, \quad \text{and integers.}$$

Solving problem (13) using the branch and bound on Tora requires 291 sup-problems to get the optimal solution while problem (15) requires only 5 sup-problems. It is a significant reduction in the number of sup-problems.

According to the reformulation by Munapo and Kumar (2016), problem (13) will become as follow:

$$\min\{20x_1 + 8x_2 + 3x_3 + 5x_4 + 33x_5\} \quad (16)$$

Subject to

$$12x_1 + 6x_2 + 2x_3 + 4x_4 + 5x_5 \geq 679$$
$$12(x_1 + x_2 + x_3 + x_4 + x_5) = 12y_1$$
$$6(x_1 + x_2 + x_3 + x_4) = 6y_2$$
$$2(x_1 + x_2 + x_4) = 2y_3$$
$$4(x_1 + x_4) = 4y_4$$
$$3(x_5) = 3y_5$$

Problem (16) requires 37 sub-problem to get the optimal solution while Problem (15) requires only 5. As a result, the proposed modification in the knapsack model outperformed the previous reformation by Munapo and Kumar (2016).

5.2 The Computational Experiments

The efficiency of the proposed approach was investigated using Tora software (Taha, 2003) on different sized randomly generated instances up to 50 variables. Table 2 shows comparison with

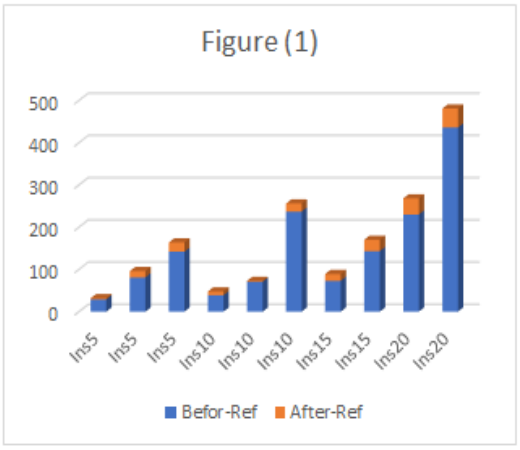
International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 5, 1140–1153, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.5-090

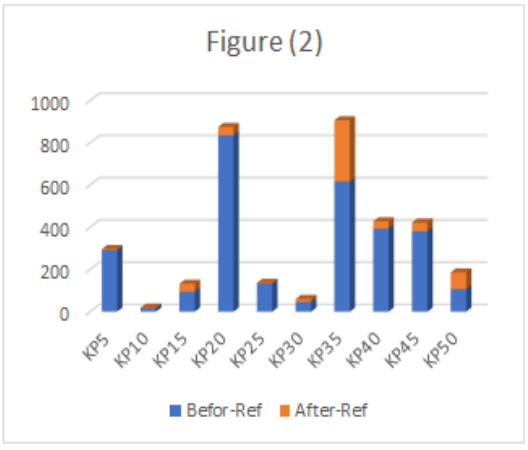
respect to the number of sup-problems that were required by the branch and bound before and after the reformulation.

Table 2. Comparison with respect to the number of sub-problems before and after the reformulation using the branch and bound on Tora software.

Problem	Optimal solution	Before Ref.	After Ref.	% Reduction
KP5	114	291	5	98.2
KP10	122	17	2	88.2
KP15	538	95	37	61.05
KP20	1328	835	39	95.3
KP25	1155	133	3	97.7
KP30	3450	47	13	72.3
KP35	10824	617	289	53.2
KP40	15172	393	35	91.09
KP45	15225	381	41	89.2
KP50	29387	107	79	26.2

Result in Tables 1 and 2 are shown in Figures 1 and 2, respectively





Figures 1 and Figure 2. Comparison with respect to before and after the reformulation

The above reformulations can also be applied to bi-objective and multi-objective models as many methods scalarize and convert the problem into single-objective. In Section 6 we illustrate the use of above reformulations on a bi-objective model.

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 5, 1140–1153, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.5-090

6. Consideration of Bi-Objective Models

6.1 Reformation Models With Respect to Bi-Objective General Linear Integer Program

A mathematical model of a bi-objective linear integer programming problem is given by:

$$z = max\{z_1(x), z_2(x)\} \quad (17)$$
Subject to $x \in X$

where $z_1(x)$ and $z_2(x)$ represent the two linear objective functions. Here *X* represents the feasible set in the decision space and *Y* represents the feasible set in the criterion space such that $x_j \in Z$ for all $j = 1,2,\ldots,n$, where n is the number of variables.

Definition 6.1. A feasible solution $x^* \in X$ in the decision space is called an efficient solution if there is no $x \in X$ such that $z_i(x) \leq z_i(x^*)$ for each i and $z_i(x) < z_i(x^*)$ for at least one i. The image $z(x^*)$ of efficient point x^*, in the criterion space, is called non-dominated point. Let X_E, Y_N denote the set of all efficient solution and non-dominated points, respectively.

The effectiveness of the proposed reformulation was tested for the bi-objective general linear integer program given by example 4.

***Numerical Example* 4**

Consider the following bi-objective general linear integer problem in (18). For comparison problem (18) is solved before and after reformulation using the ϵ - constraint method.

Part 1 (Before the Reformulation)

$$\begin{aligned} &\max\ z_1 = x_1 - x_2 + x_3 + x_4 + 5x_5 \\ &\max\ z_2 = x_1 + 2x_2 + 3x_3 + 2x_4 + 2x_5 \end{aligned} \quad (18)$$
Subject to
$$\begin{aligned} x_1 + 2x_2 + 3x_3 + 3x_4 + 6x_5 &\leq 25 \\ 14x_1 + 6x_2 + x_3 + 6x_4 + 2x_5 &\leq 60 \\ x_1 + 5x_2 + 10x_3 + 3x_4 + 6x_5 &\leq 30 \\ 10x_1 + 3x_2 + x_3 + 5x_4 + 7x_5 &\leq 50 \\ x_j \geq 0 \text{ and integer } j &= 1,2,3,4,5. \end{aligned}$$

Scalarise the bi-objective problem (18) using $w1 = w2 = 1$, so it can be written as $z_3 = z_1 + z_2$ as shown in (19)

$$\max\ z_3 = 2x_1 + x_2 + 4x_3 + 3x_4 + 7x_5 \quad (19)$$
Subject to
$$\begin{aligned} x_1 + 2x_2 + 3x_3 + 3x_4 + 6x_5 &\leq 25 \\ 14x_1 + 6x_2 + x_3 + 6x_4 + 2x_5 &\leq 60 \\ x_1 + 5x_2 + 10x_3 + 3x_4 + 6x_5 &\leq 30 \\ 10x_1 + 3x_2 + x_3 + 5x_4 + 7x_5 &\leq 50 \\ x_j \geq 0 \text{ and integer } j &= 1,2,3,4,5. \end{aligned}$$

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 5, 1140–1153, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.5-090

The lower and upper bound with respect to $z_2 = 10$ and 20, respectively. Then the problem (19) was solved with an additional constraint that is $z_2 \geq 10$, as shown in (20):

$$\max z_3 = 2x_1 + x_2 + 4x_3 + 3x_4 + 7x_5 \quad (20)$$

Subject to

$$\begin{gathered} x_1 + 2x_2 + 3x_3 + 3x_4 + 6x_5 \leq 25 \\ 14x_1 + 6x_2 + x_3 + 6x_4 + 2x_5 \leq 60 \\ x_1 + 5x_2 + 10x_3 + 3x_4 + 6x_5 \leq 30 \\ 10x_1 + 3x_2 + x_3 + 5x_4 + 7x_5 \leq 50 \\ x_1 + 2x_2 + 3x_3 + 2x_4 + 2x_5 \geq 10 \\ x_j \geq 0 \text{ and integer } j = 1,2,3,4,5. \end{gathered}$$

Solving (20) resulted in the first non-dominated point (25, 10) at iteration 2.

Increase the right-hand side of the additional constraint in (20) to 11 as in problem (21). It resulted in the second non-dominated point (22, 12) at iteration 7.

$$\max z_3 = 2x_1 + x_2 + 4x_3 + 3x_4 + 7x_5 \quad (21)$$

Subject to

$$\begin{gathered} x_1 + 2x_2 + 3x_3 + 3x_4 + 6x_5 \leq 25 \\ 14x_1 + 6x_2 + x_3 + 6x_4 + 2x_5 \leq 60 \\ x_1 + 5x_2 + 10x_3 + 3x_4 + 6x_5 \leq 30 \\ 10x_1 + 3x_2 + x_3 + 5x_4 + 7x_5 \leq 50 \\ x_1 + 2x_2 + 3x_3 + 2x_4 + 2x_5 \geq 11 \\ x_j \geq 0 \text{ and integer } j = 1,2,3,4,5. \end{gathered}$$

Continue the process by increasing the right-hand side of the additional constraint until one reaches the upper bound 20. The whole set of 6 non-dominated points were obtained and 37 sub-problems were solved.

Part 2 (After the Reformulation)

The same problem (19) was reformulated and the two objectives were scalarised as in (21). It gave (22).

$$\max z_3 = 2x_1 + x_2 + 4x_3 + 3x_4 + 7x_5 \quad (22)$$

Subject to

$$\begin{gathered} 1y_1 + 1y_2 + y_3 + 3y_4 \leq 25 \\ 14x_1 + 6x_2 + x_3 + 6x_4 + 2x_5 \leq 60 \\ x_1 + 5x_2 + 10x_3 + 3x_4 + 6x_5 \leq 30 \\ 10x_1 + 3x_2 + x_3 + 5x_4 + 7x_5 \leq 50 \\ 1(x_1 + x_2 + x_3 + x_4 + x_5) = 1\,y_1 \\ 1\,(x_2 + x_3 + x_4 + x_5) = 1\,y_2 \\ 1(\,x_3 + x_4 + x_5) = 1\,y_3 \\ 3(\,x_5) = 3\,y_4 \end{gathered}$$

$$\begin{gathered} x_j \geq 0\,,\, j = 1,2,5 \text{ and } 3 \text{ or } j = 1,2,5 \text{ and } 4, \\ y_j \geq 0 \text{ and integers for j = 1,2,3,4.} \end{gathered}$$

International Journal of Mathematical, Engineering and Management Sciences
Vol. 4, No. 5, 1140–1153, 2019
https://dx.doi.org/10.33889/IJMEMS.2019.4.5-090

Following the same steps as were used in the solution before the reformulation, the whole set of non-dominated points were obtained by solving 30 sub-problems.

6.2 Consideration of the Bi-Objective Knapsack Model

***Numerical Example* 5**

This numerical example explains the upper bound constraint idea to solve the knapsack problem.

$$\begin{aligned} &\max\ z_1 = 20x_1 + 18x_2 + 10x_3 + 9x_4 \\ &\max\ z_2 = 11x_1 + 8x_2 + 12x_3 + 20x_4 \end{aligned} \tag{23}$$

Subject to

$$\begin{aligned} 19x_1 + 8x_2 + 3x_3 + 10x_4 &\leq 168 \\ x_j \geq 0 \text{ and integer}, j &= 1,2,3,4. \end{aligned}$$

Part 1 (Before the Reformulation)

The given two objectives can be scalarised into single-objective by using $w_1 = w_2 = 1$. Problem (23) can be written as $z_3 = z_1 + z_2$ as in (24)

$$\max\ z_3 = 31x_1 + 26x_2 + 22x_3 + 29x_4 \tag{24}$$

Subject to

$$\begin{aligned} 19x_1 + 8x_2 + 3x_3 + 10x_4 &\leq 168 \\ x_j \geq 0 \text{ and integer}, j &= 1,2,3,4. \end{aligned}$$

The lower bound for $z_2 = 99$. The problem (24) with additional constraint that is $z_2 \geq 99$ is solved, the optimal solution is 180 giving the first non-dominated point that is (180, 99).

The process is continued by increasing the right-hand side of the additional constraint until the upper bound is reached. The whole set of non-dominated points required 493 sub-problems.

Part 2 (After the Reformulation)

$$\min\ z_3 = -3\bar{x}_1 + 26x_2 + 22x_3 + 29x_4 \tag{25}$$

Subject to

$$\begin{aligned} -19\bar{x}_1 + 8x_2 + 3x_3 + 10x_4 &\geq -3 \\ 11\bar{x}_1 + 8x_2 + 3x_3 + 10x_4 &\geq -1 \\ \bar{x}_1 &\leq 9 \\ x_j \geq 0 \text{ and integer}, j &= 1,2,3,4. \end{aligned}$$

After scalarize problem (23) and adding the upper bound constraint, as was discussed in Section (5.1), the whole set of non-dominated points for problem (25) were obtained by solving 378 sub-problem as against 493. For a recent approach to a bi-objective knapsack model, see Al-Rabeeah et al. (2019).

7. Conclusion

A knapsack reformulation has been applied to a general linear integer programming models with significant reduction in the number of sub-problems required by the branch and bound when using the Tora software. A numerical illustration has been given and tested on several randomly generated problems of different sizes. An upper bound constraint has been added to the knapsack model that resulted in reduction of sub-problems requirement. These two ideas have been illustrated on a bi-

objective general linear integer program and a knapsack model. Future studies will investigate concepts discussed in this paper and applied to bi-objective mix-integer models, see Al-Hasani et al. (2018).

Conflict of Interest
The authors confirm that there is no conflict of interest to declare for this publication.

Acknowledgement
The authors would like to express their appreciation to the Editor and reviewers for their suggestions. This research did not receive funding from any public, commercial or not-for-profit sector.

References

Al-Hasani, A., Al-Rabeeah, M., Kumar, S., & Ebehard, A. (2018). An improved CPU time and the triangle splitting method for solving a bi-objective mixed integer program. *International Journal of Mathematical, Engineering and Management Sciences*, *3*(4), 351-364.

Al-Rabeeah, M., Kumar, S., Al-Hasani, A., Eberhard, A., & Munapo, E.A. (2019). Bi-objective integer programming analysis based on the characteristic equation. *International Journal of Systems Assurance, Engineering and Management* (Under review).

Barnhart, C., Hane, C.A., & Vance, P.H. (2000). Using branch-and-price-and-cut to solve origin-destination integer multicommodity flow problems. *Operations Research*, *48*(2), 318-326.

Barnhart, C., Johnson, E.L., Nemhauser, G.L., Savelsbergh, M.W., & Vance, P.H. (1998). Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, *46*(3), 316-329.

Chankong, V., & Haimes, Y.Y. (2008). *Multiobjective decision making: theory and methodology*. Courier Dover Publications.

Chebil, K., & Khemakhem, M. (2015). A dynamic programming algorithm for the knapsack problem with setup. *Computers & Operations Research*, *64*, 40-50.

Chu, P.C., & Beasley, J.E. (1998). A genetic algorithm for the multidimensional knapsack problem. *Journal of Heuristics*, *4*(1), 63-86.

Davis, R.E., Kendrick, D.A., & Weitzman, M. (1971). A branch-and-bound algorithm for zero-one mixed integer programming problems. *Operations Research*, *19*(4), 1036-1044.

Della Croce, F., Salassa, F., & Scatamacchia, R. (2017). An exact approach for the 0–1 knapsack problem with setups. *Computers & Operations Research*, *80*, 61-67.

Dudziński, K., & Walukiewicz, S. (1987). Exact methods for the knapsack problem and its generalizations. *European Journal of Operational Research*, *28*(1), 3-21.

Hristakeva, M., & Shrestha, D. (2005, April). Different approaches to solve the 0/1 knapsack problem. In *The Midwest Instruction and Computing Symposium*.

Kellerer, H., Pferschy, U., & Pisinger D. (2004). Introduction to NP-completeness of knapsack problems. In: *Knapsack Problems*. Springer, Berlin, Heidelberg, pp. 483–493.

Kelley, Jr, J.E. (1960). The cutting-plane method for solving convex programs. *Journal of the Society for Industrial and Applied Mathematics*, *8*(4), 703-712.

Kumar, S, & Munapo, E. (2012). Some Lateral Ideas and their applications for developing new solution procedures for a pure integer programming model, Keynote address". In: *Proceeding of Herbal International Conference on Applications of Mathematics and Statistics for Intelligent solutions through Mathematics and Statistics*, Edited by Marriappan, Srinivasan and Amritraj, Excel India Publisher, pp. 13–21.

Kumar, S., Luhandjula, M.K., Munapo, E., & Jones, B.C. (2010). Fifty years of integer programming: a review of the solution approaches. *Asia Pacific Business Review*, *6*(3), 5-15.

Mitten, L.G. (1970). Branch-and-bound methods: general formulation and properties. *Operations Research*, *18*(1), 24-34.

Munapo, E., & Kumar, S. (2016). Knapsack constraint reformulation: A new approach that significantly reduces the number of sub-problems in the branch and bound algorithm. *Cogent Mathematics*, *3*(1), p. 1-13. , 1162372, http://dx.doi.org/10.1080/23311835.2016.1162372.

Padberg, M., & Rinaldi, G. (1991). A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, *33*(1), 60-100.

Taha, H.A. (2003). *Operations research: an introduction*". Printice-Hall of India Private Limited.

Winston, W.L., & Goldberg, J.B. (2004). *Operations research: applications and algorithms* (Vol. 3). Belmont^ eCalif Calif: Thomson/Brooks/Cole.