Solution of Multiple Constraints Knapsack Problem (MCKP) by Using Branch and Bound Method and Greedy Algorithm

Irmeilyana, Putra Bahtera Jaya Bangun, Hazrati Izzah
Department of Mathematics, Faculty of Mathematics and Natural Science, University of Sriwijaya, Indralaya, Indonesia
Email address: imel_unsri@yahoo.co.id

Abstract: In this research, we analyze the steps to solve MCKP by combining the constraints into a 0-1 knapsack problem. The purpose of this research was also comparing the solution of MCKP model by using the data research of Prasetyowati & Wicaksana (2013) in selection of promotion media at UMN. The constraints to get the audiences interest consist of limited cost, limited time, and limited workers. Furthermore, in this research, the chosen promotion media is done by using Branch and Bound (BB) method and Greedy algorithm. By using BB method, the optimal audience for newspaper media is gained through Kompas. While for online media, the optimal audience is gained by Facebook and Youtube. The optimal audience for newspaper media is gained by Greedy by Profit, Greedy by Weight, and Greedy by Density algorithms consecutively Kompas, Jawa Pos, and Suara Merdeka. While for online media, the optimal audience is gained by Greedy by Profit, Greedy by Weight, and Greedy by Density algorithms consecutively through Facebook and Youtube, Youtube, Google and Facebook. The final result of BB method and Greedy by Profit algorithm for newspaper media are the same as dynamic programming. While the result of BB method and Greedy by Density algorithm for online media are the same as dynamic programming.
Keywords: knapsack, Multiple Constraints Knapsack Problem (MCKP), Branch and Bound Method, Greedy Algorithm

1. Introduction

Knapsack is a combinatorial optimization problem to obtain the best solution of the many possibilities generated. Because the container has limited capacity, certainly not all items can be accommodated in the container. Each item will be included in a knapsack based on its weight and benefit. Total weight of the items to be put must not exceed the capacity of knapsack.

The 0-1 knapsack problem is a single knapsack problem, because n items will only be put into a knapsack. The 0-1 knapsack problem is the basis for the development of several other knapsack problems. Knapsack problems can be solved by dynamic programming algorithms, Branch and Bound methods, and Greedy algorithms. Multiple Constraints Knapsack Problem (MCKP) is a knapsack problem which is often called as the "Multidimensional Knapsack Problem (MKP)". MCKP is included in the bounded knapsack problem that limits the amount of duplication of each option into a maximum integer. Each selective item has more than one constraint. The goal of MCKP is to obtain optimal solutions by selecting a combination of items such that all restrictions are also not beyond the capacity.

Prasetyowati, M & Wicaksana have researched how to determine the optimal solution of MCKP by using dynamic programming algorithm in the election case of promotion media at University of Multimedia Nusantara (UMN). The results of the research generate a Java-based application to optimize the resources of MCKP in the selection of promotion media.

This study is limited to the implementation of MCKP in data of [1] for the case of election promotion media, but we use Branch and Bound method and Greedy algorithm. Constraints on these data include cost, time, and worker. There are 2 promotion media i.e. the newspaper media and online media.

The purposes of this study are:

1) Discuss the steps to solve MCKP by combining the constraints into a 0-1 knapsack problem.

2) Solve MCKP by using Branch and Bound method and Greedy algorithm in the election case of promotion media.

3) Compare the optimal solution by using Branch and Bound method, Greedy algorithm, and the result of dynamic programming algorithm in data of [1].

2. Research method

Based on the literature of [2], [3], [4], [5], it can be arranged steps in this research, which are:
1) Formulate MCKP for each promotion media.

Maximize:

$$Z = \sum_{j=1}^{n} p_j \, x_j \tag{1}$$

with constraints:

$$\sum_{j=1}^{n} w_{ij} \, x_j \le c_i, \quad i = 1, 2, \dots, m. \tag{2}$$

$$x_j \in \{0, 1\} \,; j = 1, 2, \dots, n.$$

Description: Z = total profit; Decision variable $x_j = 1$ if *item j* is put into *knapsack*, and $x_j = 0$ for the others; profit $p_j \ge 0$; each resource $w_{ij} \ge 0$; $i \in M = \{1, 2, \dots, m\}$; and capacity c_i .

2) Change MCKP into a 0-1 knapsack problem, i.e.

Maximize:

$$Z = \sum_{i=1}^{n} v_i \, x_i \tag{3}$$

with constrains:

$$\sum_{i=1}^{n} w_i \, x_i \le c \tag{4}$$

$$x_i \in \{0,1\}, \quad i = 1, 2, \dots, n.$$

Description: c = maximum capacity of knapsack; w_i = the weight of each item; and v_i = the profits of each item.

a) Combine constraints on MCKP between right-hand and left-hand side of the constraints by using the upper bound x_j , so it is obtained the upper bound $g(x)$ as

$$c_1 - \sum_{j=1}^{n} w_1{}^{+}{}_j \, x_j \le g(x) \le c_1 - \sum_{j=1}^{n} w_1{}^{-}{}_j \, x_j \tag{5}$$

Where $w_i{}^{+}{}_j = \max\{w_{ij};\ 0\}$ and $w_i{}^{-}{}_j = \min\{w_{ij};\ 0\}$ by choosing a positive integer and λ must meet:

$$\lambda > \max\left\{ c_1 - \sum_{j=1}^{n} w_1{}^{-}{}_j \, x_j \,;\, -c_1 + \sum_{j=1}^{n} w_1{}^{+}{}_j \, x_j \right\} \tag{6}$$

and $|g(x)| < \lambda$.

The second constraint is multiplied by λ and summed to the first constraint, so it is obtained new constraint as follow:

Maximize

$$Z = \sum_{j=1}^{n} p_j x_j \tag{7}$$

with constraints:

$$\sum_{j=1}^{n} (w_{1j} + \lambda w_{2j}) x_j = c_1 + \lambda c_2, \tag{8}$$

$$0 \le x_j \le 1;\ x_j \ integer;\ j = 1, 2, \dots, n.$$

Combining result of the constraints in Equation (8) is inserted into Equation (9):

$$g(x) = c_{1^*} - \sum_{j=1}^{n} w_{1j^*} \, x_j \tag{9}$$

In the same way, it is done for each c_i ; $i > 2$.

Description:

c_1^* = capacity of a merger result of one and two constraints.

w_{1j^*} = coefficient x_j from the merging results of first and second constraints.

c_i = capacity for each source *i* ($c_i > 0$)

u_j = value of decision variables are worth 1 ($0 \leq x_j \leq 1$)
i = source
j = item
M = set of sources, namely $M = \{1, 2, ..., m\}$
N = set of items, namely $N = \{1, 2, ..., n\}$

b) Change MCKP form from the result of Step 2.1 becomes standard Linear Programming.

3). Resolve MCKP by using Branch and Bound method with the following steps:

a) Determine the optimal solution of relaxation linear equations based on the objective function and constraints formed by using Dantzig bound. The process is resolved with LINDO 6.1 Program.
b) Search the upper bound, lower bound, and fixing the upper bound.
c) Branch in the fraction variables.
d) Get the optimal solution from selected sub problem.
e) Conduct cuts if:
   (i) Sub problem is not feasible.
   (ii) the sub problem provides the optimal solution where all the variables are integer.
   (iii) The optimal solution of the sub problem was no better than optimal solution of the other sub problem.
   In maximum problem, an optimal solution of the sub problem is not larger than the lower bound that has been obtained).
f) If the chosen sub problem is not cut, then the branching has been done by creating a new sub problem.
g) Iteration completed if all variable in the optimal solution of feasible sub problem is integer and the solution is most optimal.

4) Solve MCKP by using three categories of Greedy algorithms as follows:

a) Greedy by Profit
b) Greedy by Weight
c) Greedy by Density

5) Obtain the most optimal solution by comparing the results from Branch and Bound method and Algorithm Greedy.

6) Compare the optimal solution obtained from Step 5 with the results that were obtained by [1].

3. Results and discussion

3.1 Research of Prasetyowati, M & Wicaksana

Prasetyowati, M & Wicaksana discussed the selection of the best promotion media in the UMN to attract prospective of students to be studying at the university. This test was done to find the optimal solution of Multiple Constraints Knapsack Problem (MCKP) and its implementation by using a dynamic programming algorithm. There are 6 promotion media in the printed newspaper media, i.e. Kompas, Media Indonesia, Jawa Pos, Koran Tempo, Seputar Indonesia and Suara Merdeka. Similarly, there are also 6 online media, i.e. Google.com, Facebook.com, Yahoo.com, Kompas.com, Detik.com, and Youtube.com. Both of these media types have the same constraints that are the cost, time, and labor. There are three sources in this study with 6 items that become consideration in order to obtain optimal solution.

Optimization results in choosing the newspaper media based on number of most audience is through Kompas. While the optimal solution in the online media based on number of most audience is through Google.com and Facebook.com.

3.2 MCKP model in selection of promotion media

MCKP for printed newspaper media become 0-1 Knapsack Model

MCKP to maximize profits for the selection of printed newspaper media is:

Maximize

$$Z_1 = 1.7\, x_1 + 0.75\, x_2 + 1.3\, x_3 + 0.90\, x_4 + 1\, x_5 + 0.33\, x_6 \quad (10)$$

with constraints:

a. Costs (in the millions hundreds rupiah)

$$1.7442\, x_1 + 0.8424\, x_2 + 1.4256\, x_3 + 1.1016\, x_4 + 1.1556\, x_5 + 0.648\, x_6 \leq 2 \quad (11)$$

b. The time (in days)

$$3x_1 + 4x_2 + 5x_3 + 4x_4 + 3x_5 + 3x_6 \leq 5 \quad (12)$$

c. Workers (in person)

$$2x_1 + 2x_2 + 2x_3 + 2x_4 + 2x_5 + 2x_6 \leq 2 \quad (13)$$

$x_j \geq 0$ for $j = 1, 2, 3, \ldots, 6$

Value $w_{i^+j} = \max\{w_{ij};\ 0\}$ and $w_{i^-j} = \min\{w_{ij};\ 0\}$ from first constraint; i.e. costs (in the millions hundreds rupiah)

$w_{i^+j} = \max\{1.7442\ ; 0.8424\ ; 1.4256\ ; 1.1016\ ; 1.1556\ ; 0.648\ ; 0\}$

$= 1.7442$

$w_{i^-j} = \min\{1.7442\ ; 0.8424\ ; 1.4256\ ; 1.1016\ ; 1.1556\ ; 0.648\ ; 0\}$

$= 0$

So the λ value as follows:

$$c_1 - \sum_{j=1}^{6} w_{1^+j}\, x_j \leq g(x) \leq c_1 - \sum_{j=1}^{6} w_{1^-j}\, x_j \Leftrightarrow 2 - 1{,}7442(1) \leq g(x) \leq 2 - 0(1)$$

$$\Leftrightarrow 0.2558 \leq g(x) \leq 2$$

$$\lambda > \max\left\{c_1 - \sum_{j=1}^{n} w_{1^-j}\, x_j\ ;\ -c_1 + \sum_{j=1}^{n} w_{1^+j}\, x_j\right\}$$

$\lambda > \max\{2 - 0\ ;\ -2 + 1.7442\}$

$\lambda > \max\{2\ ; -0.2558\}$

$\lambda > 2$

In this matter, suppose $\lambda = 3$;

$$\sum_{j=1}^{6}(w_{1j} + \lambda w_{2j})x_j = c_1 + \lambda c_2$$

$$\Leftrightarrow (1.7442 + 3(3))x_1 + (0.8424 + 3(4))x_2 + (1.4256 + 3(5))x_3 + (1.1016 + 3(4))x_4 + (1.1556 + 3(3))x_5 + (0.648 + 3(3))x_6 = 2 + 3(5)$$

$$\Leftrightarrow 10.7442\ x_1 + 12.8424\ x_2 + 16.4256\ x_3 + 13.1016\ x_4 + 10.1556\ x_5 + 9.648\ x_6 = 17$$

Furthermore,

$$g(x) = c_{1^*} - \sum_{j=1}^{6} w_{1j^*}\, x_j$$

$$= 17 - (10.7442\ x_1 + 12.8424\ x_2 + 16.4256\ x_3 + 13.1016\ x_4 + 10.1556\ x_5 + 9.648\ x_6)$$

with

c_1^* = capacity in the merger of Equation (11) and Equation 12.

Constraint on Equation (12) is $c_{1^*} = c_1 + \lambda c_2 = 17$.

w_{1j^*} = coefficient x_j from the results of a merger of constraints; with

$w_{11^*} = 10.442\ ;\ w_{12^*} = 12.8424\ ;\ w_{13^*} = 16.4256\ ;\ \ w_{14^*} = 13.1016;$
$w_{15^*} = 10.1556\ ;\ w_{16^*} = 9.648.$

Because the result of merging capacity and value of each source of cost and time have been obtained, the next step repeat the steps to combine results from the merger costs and time constraints with worker constraint as follows:

$w_{i^*+j} = \max\{10.7442\ ; 12.8424\ ; 16.4256\ ; 13.1016\ ; 10.1556\ ; 9.648\ ; 0\}$

$= 16{,}4256$

$w_{i^*-j} = \min\{10.7442\ ; 12.8424\ ; 16.4256\ ; 13.1016\ ; 10.1556\ ; 9.648\ ; 0\}$

$= 0$

So the λ^* value as follows :

$$c_{1^*} - \sum_{j=1}^{6} w_{i^*+j}\, x_j \leq g_1(x) \leq c_{1^*} - \sum_{j=1}^{6} w_{i^*-j}\, x_j$$

$$\Leftrightarrow 17 - 16.4256(1) \leq g_1(x) \leq 17 - 0(1)$$

$$\Leftrightarrow 0.5774 \leq g_1(x) \leq 17$$

Based on Equation (6), then λ^* can be obtained as follows :

$$\lambda^* > \max\left\{c_{1^*} - \sum_{j=1}^{6} w_{1^*-j}\, x_j\ ;\ -c_{1^*} + \sum_{j=1}^{6} w_{1^*+j}\, x_j\right\}$$

$\lambda^* > \max\{17 - 0\ ;\ -17 + 16.4256\}$

$\lambda^* > 17$

In this matter, suppose $\lambda^* = 18$.
The result of merging constraints:

$$\sum_{j=1}^{6} (w_{1j^*} + \lambda^* w_{3j}) x_j = c_{1^*} + \lambda^* c_3$$

$$\Leftrightarrow (10.7442 + 18(2))x_1 + \big(12.8424 + 18(2)\big)x_2 + \big(16.4256 + 18(2)\big)x_3$$
$$+\big(13.1016 + 18(2)\big)x_4 + \big(10.1556 + 18(2)\big)x_5 + \big(9.648 + 18(2)\big)x_6$$
$$= 17 + 18(2)$$
$$\Leftrightarrow 46.7442\ x_1 + 48.8424\ x_2 + 52.4256 x_3 + 49.1016\ x_4 + 46.1556\ x_5$$
$$+ 45.648\ x_6 = 53$$

Based on the result of merging the three constraints, such as cost, time and worker it is acquired new constraints that is used to solve MCKP for the newspaper media. Constraints are determined from the weight of each item to be selected and limited by the capacity of the knapsack. So the objective function and constraints for the 0-1 knapsack problem in the printed newspaper media used is
Maximize

$$Z_1 = \ 1.7\ x_1 + \ 0.75\ x_2 + \ 1.3\ x_3 + \ 0.90\ x_4 + \ 1\ x_5 + \ 0.33\ x_6 \qquad (14)$$

with constraints:

$$46.7442\ x_1 + 48.8424\ x_2 + 52.4256 x_3 + 49.1016\ x_4 + 46.1556\ x_5$$
$$+ 45.648\ x_6 \le 53 \qquad (15)$$
$$0 \le x_j \le 1;\ x_j \textit{ integer} \text{ for } j = 1, 2, \dots, 6.$$

MCKP for online media is:
Maximize:

$$Z_2 \ = 2.99125\ x_1 + \ 13.589908\ x_2 + \ 1.348\ x_3 + \ 1.209156\ x_4 + \ \ 2.799810\ x_5 + 4.470320\ x_6 \qquad (16)$$

with constraints:

a. Costs (in the millions hundreds rupiah)
$$0.75x_1 + \ 1.2x_2 + \ 0.5x_3 + \ 1.5x_4 + \ 1.5x_5 + \ 1.5x_6 \le 3 \qquad (17)$$
b. The time (in days)
$$1x_1 + \ 1x_2 + \ 2x_3 + \ 2x_4 + \ 2x_5 + \ 2x_6 \ \le 5 \qquad (18)$$
c. Workers (in person)
$$2x_1 + \ 2x_2 + \ 3x_3 + \ 2x_4 + \ 3x_5 + \ 3x_6 \ \le 5 \qquad (19)$$

$x_j \ \ge 0$ for $j = 1, 2, 3, \dots, 6$

In the same way as MCKP in the printed media, it is gained 0-1 knapsack problem on the online media:
Maximize:

$$Z_2 \ = 2.99125\ x_1 + \ 13.589908\ x_2 + \ 1.348\ x_3 + \ 1.209156\ x_4 + 2.799810\ x_5$$
$$+ 4.470320\ x_6 \qquad (20)$$

with constraints:

$$52.75\ x_1 + 52.8\ x_2 + 76\ x_3 + 57.5\ x_4 + 81.5\ x_5 + 81.5\ x_6 \le 143 \qquad (21)$$
$$0 \le x_j \le 1;\ x_j \textit{ integer},\ j = 1, 2, \dots, n.$$

3.2.1 The solution of MCKP by using Branch and Bound Method

MCKP model in Equation (14) and Equation (15) is converted to standard form by adding slack variables as follows:
Maximize:

$$Z_1 = 1.7\ x_1 + 0.75\ x_2 + 1.3\ x_3 + 0.90\ x_4 + 1\ x_5 + 0.33\ x_6 + 0\ s_1 + 0\ s_2$$
$$+ 0\ s_3 + 0\ s_4 + 0\ s_5 + 0\ s_6 \qquad (22)$$

with constraints:

$$46.7442\ x_1 + 48.8424\ x_2 + 52.4256 x_3 + 49.1016\ x_4 + 46.1556\ x_5$$
$$+ 45.648\ x_6 + s_1 \ = 53$$
$$x_j + s_{j+1} \ = 1 \text{ for } j = 1, 2, \dots, 6.$$
$$x_j + s_{j+7} \ = 0 \text{ for } j = 1, 2, \dots, 6.$$
$$x_1, x_2, x_3, x_4, x_5, x_6 \ge 0$$
$$s_1, s_2, s_3, s_4, s_5, s_6, s_7 \ge 0 \qquad (23)$$

The optimal solution of MCKP on the test data of newspaper media is $X = \{1,0,0,0,0,0\}$ with *Z*–optimal 1.7. So the newspapers media items are used as promotional to obtain the optimal number of audience is Kompas.

In the same way for the solution of MCKP on online media is obtained X = {0, 1, 0, 0, 0, 1} with Z-optimal 18.06023. So the online promotional media items are used as promotional to obtain the optimal number of audience are Facebook and Youtube.

3.2.2 The Solution of MCKP by using *Greedy*

Value constraints w_j is a coeficient x_j on the combined constraints in Equation (15) for the newspaper media and Equation (21) for online media.

The weight of item *j* i. e. w_j; is a combination of w_{ij} ; source *i* on item *j* call as "*constraints item*". w_{ij} combined is weight of item *j* on resource constraints of cost, time, and number of workers. In this case, *"constraints item"* has no unit.

Greedy by Profit Algorithm in testing data of newspaper media

The first step, the calculation of Greedy by Profit is sorting the items based on the acquisition of the largest number of audience. Then the items are taken one by one until there is no more items can be selected.

Table 1. Testing data on newspapers media by using greedy by profit

No.	Item - j (x_j)	Promotion Media (N)	Constraints (w_j)	Audience (p_j)	Value (x_j)	Total Constraints $\sum_{j=m}^{k} w_j \cdot x_j$
1	1	Kompas	4,674,42,.000	1,700,000	1	4,674,420,000
2	3	Jawa Pos	5,242,560,000	1,300,000	0	-
3	5	Seputar Indonesia	4,615,560,000	1,000,000	0	-
4	4	Koran Tempo	4,919,160,000	900,000	0	-
5	2	Media Indonesia	4,884,240,000	750,000 (estimation)	0	-
6	6	Suara Merdeka	4,564,800,000	330,000 (estimation)	0	-

Description: *m = item j* that has greatest audience number (item that is firstly put into knapsack)
k = The accumulation of many items are put into the knapsack.

Based on Table 1, the selected item to the sources that is x_1. Item x_1 is worth 1, while other items worth 0, because *constraints item* has exceeded the capacity of knapsack. So the optimal solution based on the Greedy by Profit algorithm is $X = \{1, 0, 0, 0, 0, 0\}$.

Based on the objective function in Equation (14) then the benefits by Greedy by Profit algorithm calculations with the acquisition of audience is 1.7 million and a total constraint is 4.67442 billion. In this case, the capacity of knapsack filled by printed newspaper media is Kompas, i.e. 88.19%.

Greedy by Weight Algorithm in testing data of newspaper media

Table 2. Testing data on newspapers media by using greedy by weight

No.	*Item -j* (x_j)	Promotion Media (N)	*Constraints* (w_j)	*Audience* (p_j)	Value (x_j)	Total *Constraints* $\sum_{j=p}^{k} w_j \cdot x_j$
1	3	Jawa Pos	5.242.560.000	1,300,000	1	5,242,560,000
2	4	Koran Tempo	4.910.160.000	900,000	0	-
3	2	Media Indonesia	4.884.240.000	750,000 (estimation)	0	-
4	1	Kompas	4.674.420.000	1,700,000	0	-
5	5	Seputar Indonesia	4.615.560.000	1,000,000	0	-
6	6	Suara Merdeka	4.564.800.000	330,000 (estimation)	0	-

Description: m = *item j* that has greatest audience number (item that is firstly put into knapsack)
k = The accumulation of many items are put into the knapsack.

Based on Table 2, the optimal solution based on Greedy by Weight algorithm is X ={0, 0, 1, 0, 0, 0}, so that the audience number as many as 1.3 million and total constraints is 5.24256 billion. In this case, the capacity of knapsack filled by the newspaper media is Jawa Pos amounted to 98.92%.

Greedy by Density Algorithm in testing data of newspaper media

Greedy by Density Algorithm sorts items based on the density $\left(\frac{p_j}{w_j}\right)$. The largest density of item is taken one by one that can be accommodated by knapsack until there is no more items that could be included.

Table 3. Testing data on newspapers media by using greedy by weight

No.	*Item -j* (x_j)	Promotion Media (N)	*Constraints* (w_j)	*Audiences* (p_j)	*Density* $\left(\frac{p_j}{w_j}\right)$	Value (x_j)	Total *Constraints* $\sum_{i=r}^{k} w_j \cdot x_j$
1	6	Suara Merdeka	4,564,800,000	330,000 (estimation)	0.00072	1	4.564.800.000
2	1	Kompas	4,674,420,000	1,700,000	0.00036	0	-
3	3	Jawa Pos	5,242,560,000	1,300,000	0.00025	0	-
4	5	Seputar Indonesia	4,615,560,000	1,000,000	0.00022	0	-
5	4	Koran Tempo	4,919,160,000	900,000	0.00018	0	-
6	2	Media Indonesia	4,884,240,000	750,000 (estimation)	0.00015	0	-

Based on Table 3, the selected items into a knapsack that x_6. The optimal solution based on Greedy by Density algorithms is X = {0, 0, 0, 0, 0, 1}, so that the audience number as many as 330,000 and total constraints is 4.5648 billion. In this case, the capacity of knapsack filled by newspaper media is Suara Merdeka that is 86.13%.

Greedy Algorithms in testing data of online media

In the same way as greedy by Profit, Greedy by Weight, and Greedy by Density in printed newspaper media, the obtained solution on online media as follows:

(i) The optimal solution based on Greedy by Profit algorithms is X = {0, 1, 0, 0, 0, 1}, so that the audience number as many as 8,060,228 dan total *contraints* is 134.300.000. In this case, the capacity of knapsack filled by online media is Facebook dan Youtube; that is 93.92%.

(ii) The optimal solution based on Greedy by Weight algorithms is X = {0, 0, 0, 0, 0, 1}, so that the audience number as many as 4,470,320 and total *constraints* is 81,500,000. In this case, the capacity of knapsack filled by online media is Youtube; that is 56.99%.

(iii) The optimal solution based on Greedy by Density algorithms is X = {1, 1, 0, 0, 0, 0}, so that the audience number as many as 16,581,158 and total *constraints* is 105,550,000. In this case, the capacity of knapsack filled by online media is Google dan Facebook; that is 73.81%.

3.3 Comparison of Results of Branch and Bound Methods and Greedy Algorithms with The Results of Dynamic Programming Algorithm

The optimal solution of MCKP in election case of newspaper media by using BB (Branch and Bound) method and Greedy algorithm is Kompas. The number of audience is 1,700,000. As for the online media is Facebook and Youtube with the number of audience of 18,060,228.

The result of MCKP by using BB method and Greedy algorithm for newspaper media is the same with the results obtained by [1] by using dynamic programming algorithm; i.e. Kompas. As for the result of MCKP for online media by using BB method and algorithm Greedy is Facebook and Youtube, in contrast to result obtained by dynamic programming algorithm that is Google and Facebook.

4. Conclution

The conclusions of this research are:

1) Completion of MCKP by using Branch and Bound method and Greedy algorithm is through the incorporation of constraints into a single constraint as the 0-1 knapsack problem.

2) The optimal solution of MCKP for selection of newspaper media based on BB method is gained through Kompas. For newspapers media, the results of the Greedy by Profit, Greedy by Weight, and Greedy by Density Algorithms are consecutively gained through Kompas, Jawa Pos, and Suara Merdeka. While online promotion media based on BB method is through Facebook and Youtube. While for online promotion media, the results of the Greedy by Profit, Greedy by Weight, and Greedy by Density Algorithms are consecutively gained through Facebook and Youtube, Youtube, Google and Facebook.

3) The final results of BB method and Greedy by Profit algorithm for newspaper media are the same as dynamic programming. While the result of BB method and Greedy by Density algorithm for online media are the same as dynamic programming.

References

[1] Prasetyowati, M & Wicaksana, 2013. Implementasi Algoritma *Dynamic Programming* untuk *Multiple Constraints Knapsack Problem.Jurnal UMN (Online).*http://download.portalgaruda.org/article.php?article=95063&val=576 (accessed in February 15, 2015).

[2] Martello, S & P. Toth, 1990. *Knapsack Problem.* John Wiley & Sons, Singapore.

[3] Munir, R. 2004. Algoritma *Greedy*. http://www.kur2003.if.itb.ac.id. (accessed in March 20, 2015).

[4] Pekbey, D. 2003. *Computational Analysis of The Search Based Cuts on The Multidimentional 0-1 Knapsack Problem*. Thesis. Bilkent University.

[5] Pisinger, D. 1995. *Algorithm for Knapsack Problem (Chapter 1*). Thesis. University of Copenhagen, Universitetsparken.