Available online at https://journal.rescollacomm.com/index.php/ijqrm/index

International Journal of Quantitative Research and Modeling

e-ISSN 2721-477X
p-ISSN 2722-5046

Vol. 6, No. 3, pp. 316-322, 2025

Implementation of Dynamic Programming Algorithm on The Integer Knapsack Problem (0/1) (Case Study: J&T Cargo Agent Purwokerto)

Leni Puspitasari[1*], Agus Sugandha[2], Siti Rahmah Nurshiami[3]

[1,2,3]*Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Jenderal Soedirman, Indonesia*

**Corresponding author email: leni.puspitasari@mhs.unsoed.ac.id*

Abstract

The aim of this research is to solve the integer knapsack 0/1 problem, which is a problem of selecting goods from a large number of available goods where each item has a different weight and profit. Shipping goods in the J&T Cargo Purwokerto shipping service is one of the many problems in selecting goods. Shipping goods in J&T Cargo Purwokerto is carried out in stages with a higher profit value first because the shipping load capacity can only accommodate 700 kg. In order for J&T Cargo Purwokerto agents to obtain maximum profits, the selection of goods to be shipped must be carried out first. The selection of goods in J&T Cargo Purwokerto agents can be solved using the integer knapsack problem 0/1 method using the forward recursive dynamic programming algorithm with the help of Matlab R2021A software. The results showed that on July 1, 2025, the maximum profit was obtained of IDR 3,038,850 with a weight of 700 kg. On July 2, the maximum profit was obtained of IDR 4,884,985 with a weight of 700 kg. On July 3, the maximum profit was obtained of IDR 7,732,155 with a weight of 699 kg.

Keywords: Integer Knapsack Problem 0/1, Algorithm, Dynamic Programming Algorithms

1. Introduction

Optimization is a common problem in everyday life. Solving optimization problems is a way to determine optimal results within existing constraints. Some optimization problems encountered in everyday life include determining public transportation routes, determining workforce size, and scheduling. Another interesting optimization problem to discuss is the challenges faced by freight forwarding services.

Freight forwarding services are now commonplace and highly sought after to facilitate the delivery of goods, both long and short distances. Shipping costs depend on the distance and weight of the goods. To maximize profits with minimal costs and limited container capacity, the goods to be distributed must be carefully selected.

*J&T Cargo*Purwokerto is one of the freight forwarding service providers that has ten type A outlets, namely outlets that receive and send packages from all over Indonesia and eleven type B outlets, namely outlets that only accept package pick-ups. There are several services available at J&T Cargo Purwokerto, namely fast track with a minimum weight of 10 kg, mass track with a minimum weight of 100 kg, and FTL (full truck load) which is a fleet rental service such as cars, trucks, and others. Shipping goods at J&T Cargo Purwokerto prioritizes longer distance locations, because they have large profits, for closer locations they will be sent the next day. This is done because the available container capacity to accommodate goods is limited and not comparable to the number of goods to be shipped, so goods must be shipped in installments based on greater profits first to obtain maximum profits. The problem of selecting shipping goods by J&T Cargo Purwokerto can be solved by solving the integer knapsack problem.

The knapsack problem or rucksack problem is a combinatorial optimization problem to find the best solution from many possible solutions (Martello & Toth, 1990; Messac, 2015). The knapsack problem occurs when there are n items that cannot all be put into a storage area. The available items have different weights and values. The goal of solving the knapsack problem is to obtain maximum profit from the selection of available items without exceeding the storage capacity (Cormen et al., 2022; Siang, 2014). There are four types of knapsack problems, namely the bounded knapsack problem which means that each item available is limited in number or there is only one unit of the item, the unbounded knapsack problem which means that each item available is unlimited in number or there is more than one unit available, the fractional knapsack problem which means that the number of items for each item can be fractional, and the integer knapsack problem (0/1) which means that each item is only available in one unit. This research will

use the integer knapsack problem (0/1) because a value of 0 will be given to items that are not selected and sent the next day and a value of 1 will be given to items that are selected to be sent that day.

2. Literature Review

Research on the knapsack problem has been conducted by Rois et al. (2019) on solving integer knapsack problems using greedy, dynamic programming, brute force and genetic algorithms, from this study it can be concluded that the dynamic programming algorithm is an effective and efficient algorithm for solving integer knapsack problems for small and large scale data. Research by Pitaloka (2017) on the 0-1 knapsack optimization problem using the dynamic programming algorithm concluded that dynamic programming solves problems by decomposing the solution into a set of steps or stages so that there is a series of related decisions.

Extensive research has been conducted on the knapsack problem. A comparative study by Ezugwu et al. (2016) analyzed heuristic algorithms for the 0/1 knapsack problem and compared their performance with the dynamic programming approach, showing the relative effectiveness of both methods. Similarly, Sharma & Kumar (2017) conducted a comparative study of dynamic programming and genetic algorithms for solving the knapsack problem, providing insights into which algorithm is more suitable under different conditions.

Furthermore, Agrawal & Jain (2018) specifically discussed the implementation of a dynamic programming approach for the 0/1 knapsack problem. Their work concluded that dynamic programming solves problems by decomposing the solution into a set of stages or a series of related decisions. A survey by Dutta & Barman (2020) also reviewed various algorithms used for the 0/1 knapsack problem, including dynamic programming, and concluded that it is a robust choice for optimization problems where an optimal solution is required. Based on these studies, it is clear that dynamic programming is an effective and efficient algorithm for solving the integer knapsack problem (0/1) for both small and large-scale data. Therefore, this research is interested in discussing and continuing this line of inquiry with different data, using the dynamic programming algorithm with a case study at J&T Cargo Purwokerto, to enable the company to select the goods for shipment to obtain maximum profit.

Another study conducted by Research by Salsabila (2022) on the application of the greedy algorithm and the dynamic programming algorithm to the integer knapsack problem, concluded that the dynamic programming algorithm is more optimal than the greedy algorithm based on the maximum profit obtained. Therefore, researchers are interested in discussing and continuing the study by Salsabila (2022) on integer knapsack (0/1) with different data and using the dynamic programming algorithm. With the case study of J&T Cargo Purwokerto using the forward recursive dynamic programming algorithm so that J&T Cargo Purwokerto can select the goods to be sent first so as to obtain maximum profit.

3. Methods

The methods used in this research are literature review and case study. Literature review involves reading, studying, understanding, and analyzing materials from literature or other references such as books, theses, journals, and e-books. The case study was conducted at J&T Cargo Purwokerto. The data used in this research is secondary data in the form of data on shipping destinations, weight, and profit data obtained from J&T Cargo Purwokerto. The steps taken in this research are as follows:

1) Data collection,
2) determine the problem structure by defining all the variables obtained,
3) solve problems using dynamic programming algorithms,
4) compile a table of completion of each stage and capacityiy,
5) make decisions by choosing the maximum benefit at each stage,

4. Results and Discussion

4.1. Goods Delivery Data

Data obtained from the J&T Cargo Purwokerto agent in the form of data on the weight of goods, goods profit, shipping destination, and the maximum capacity of the shipping load in the form of a Grand Max car with a maximum capacity of 700 kg to accommodate goods that will be sent to the Sokaraja branch on July 1, 2025 to July 3, 2025. The data is then processed using Matlab R2021A to find out which goods will be sent first, by giving a value of 1 to the goods to be sent and giving a value of 0 to the goods that will be sent the next day.

Table 1: Data on goods received on July 1, 2025

Item number	Shipping Destination	Weight of Goods	*Profit* Goods
1	Gunungkidul Regency	11	IDR 41,655
2	Bogor Regency	11	IDR 47,320
3	Tangerang Regency	11	IDR 47,320
4	Bandung	11	IDR 58,500
5	Cilacap Regency	19	IDR 73,500
6	Bekasi City	11	IDR 47,320
7	Serang City	37	IDR 153,000
8	Lamongan Regency	12	IDR 50,000
9	South Jakarta	11	IDR 40,500
10	Sleman Regency	19	IDR 63,000
11	Surakarta City	11	IDR 35,000
12	East Jakarta	16	IDR 62,000
13	Kuningan Regency	11	IDR 40,500
14	Semarang City	11	IDR 35,000
15	Denpasar City	11	IDR 71,000
16	Gresik Regency	100	IDR 250,000
17	Surabaya City	54	IDR 197,000
18	Central Jakarta	20	IDR 75,000
19	Tangerang Regency	11	IDR 46,000
20	Karanganyar Regency	11	IDR 55,000
21	Surakarta City	20	IDR 65,000
22	Blitar Regency	11	IDR 46,000
23	Batang Regency	49	IDR 176,500
24	Bontang City	26	IDR 317,000
25	Semarang City	11	IDR 35,000
26	Bogor Regency	40	IDR 165,000
27	Sleman Regency	12	IDR 39,080
28	Bintan Regency	34	IDR 379,000
29	Kebumen Regency	11	IDR 41,655
30	West Jakarta	11	IDR 40,500
31	West Bandung Regency	25	IDR 92,500
32	Pekalongan Regency	23	IDR 85,500
33	Central Jakarta	11	IDR 40,500
34	North Jakarta	34	IDR 124,000

Based on the data in Table 1, there are 34 items with a total weight of 727 kg and a total profit of IDR 3,135,850, with a maximum shipping load capacity of$(M)$to the Sokaraja branch office, weighing 700 kg. Because the total weight of the goods exceeds the maximum shipping capacity, it is necessary to select which goods will be shipped to the Sokaraja branch office on July 1, 2025.

4.2. Integer Model Knapsack Problem (0/1)

The steps in forming an integer knapsack problem model include:

1) Identify the data variables in Table 4.1 by notating the goods as$k$, the weight of the item as , and the profit of the item as $.kw_k kp_k$
2) Forming an integer model of the knapsack problem (0/1)
   a. Determine the objective function

$$Z = \sum_{k=1}^{n} x_k p_k \tag{1}$$

$$Z = 41.655x_1 + 47.320x_2 + 47.320x_3 + \cdots + 40.500x_{33} + 124.000x_{34}$$

   b. Determine the constraint function

$$z = \sum_{k=1}^{n} x_k w_k \leq M \tag{2}$$

$$z = 11x_1 + 11x_2 + 11x_3 + \cdots + 11x_{33} + 34x_{34} \leq 700$$

c. Determining decision variables

$$x_k = \{x_1, x_2, \ldots, x_{33}, x_{34}\}$$

x_k: Decision whether the item is taken (1) or not (0),$\forall k \in \{1, 2, \ldots, n\}$

3) Solve the integer knapsack problem (0/1) with the dynamic programming algorithm.

4.3. Methods

Based on Table 1, the stages for solving integer knapsack (0/1) using the dynamic programming algorithm are as follows:

1) Determining the problem structure
2) Using the forward recursive equation, namely:

$$f_i(y) = \begin{cases} f_{i-1}(y) & if\ y < w_k \\ max\{f_{i-1}(y), p_k + f_{i-1}(y - w_k)\} & if\ y \geq w_k \end{cases} \quad (3)$$

with,

$f_i(y)$:Maximum profit at the stage with capacityiy
y :Knapsack capacity at stage-i
p_k :Profit object to-k
w_k :Weight of the object k

3) Prepare a settlement table at the 6th stage$i$with capacity , for and .$y_i = 1, 2, 3 \ldots, 34;\ y = 0, 1, 2, \ldots, 700$
4) Determine decisions by choosing optimal profits at each stage that has been carried out.

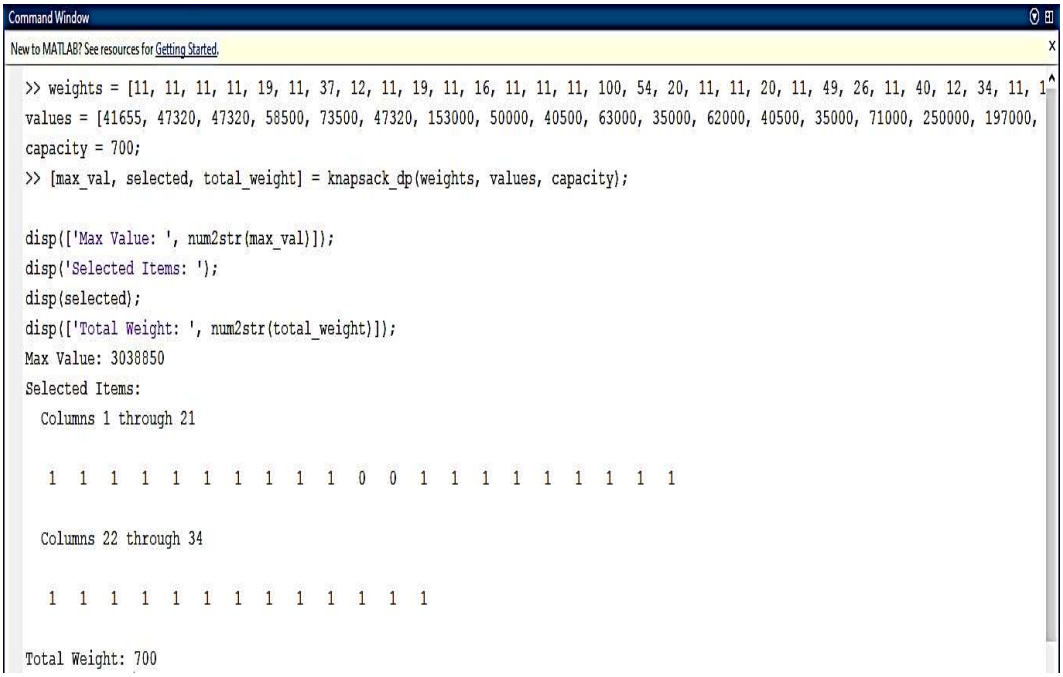

Figure 1: Matlab R2021A output of calculation results using dynamic programming

Based on Figure 1, it is obtained the solution of the integer knapsack problem (0/1) with the dynamic programming algorithm using Matlab R2021A software resulted in a total weight of goods included in the shipping load of 700 kg with a profit of IDR 3,038,850. The goods that will be included in the shipping load on July 1, 2025 are items 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34. The 11th and 12th items will be included in the shipping load the next day.

4.4. Next Day Item Selection

1) Selection of items to be shipped on July 2, 2025

Items to be shipped on July 2, 2025 consist of items that were not shipped on July 1, 2025 and items that were newly received on July 2, 2025 that will be selected for shipping.On July 2, 2025, there were 44 items with a total weight of 723 kg and a total profit of IDR 4,931,900, with a maximum shipping load capacity of (M) to the

Sokaraja branch office amounting to 700 kg and the selection of incoming goods delivery on July 2, 2025 will be carried out by reducing the maximum capacity of the delivery load with the total weight of the goods on July 1, 2025 that cannot be sent, namely 27 kg, so that the maximum delivery load capacity is 673 kg.

With the objective function:

$$Z = \sum_{k=1}^{n} x_k p_k$$
$$Z = 45.260x_1 + 66.500x_2 + \cdots + 205.400x_{43} + 105.000x_{44} \quad (4)$$

and the constraint function:

$$z = \sum_{k=1}^{n} x_k w_k \leq M$$
$$z = 18x_1 + 19x_2 + 19x_3 + \cdots + 52x_{43} + 30x_{44} \leq 673 \quad (5)$$

```
>> weights = [18, 19, 19, 14, 16, 11, 11, 11, 13, 19, 11, 17, 11, 11, 16, 11, 11, 11, 11, 13, 19, 11, 11, 11, 60, 15, 11, 21, 11, 11,
values = [45260, 66500, 57000, 41700, 56000, 46000, 149500, 40500, 61000, 61000, 35000, 68000, 59000, 109000, 334000, 40500, 47500, 4
capacity = 673;
[max_val, selected, total_weight] = knapsack_dp(weights, values, capacity);
disp(['Max Value: ', num2str(max_val)]);
disp('Selected Items: ');
disp(selected);
disp(['Total Weight: ', num2str(total_weight)]);
Max Value: 4787985
Selected Items:
  Columns 1 through 27

   0   1   0   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1   1

  Columns 28 through 44

   1   1   1   1   1   1   1   0   1   1   1   1   1   1   1   1   1

Total Weight: 673
```

Figure 2: Matlab R2021A output of selection of items shipped on July 2, 2025

Based on Figure 2, the items that will be included in the shipping load are items 2, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44 with a profit of IDR 4,787,985. The total weight of the goods to be shipped on July 2, 2025 was 700 kg and the total profit obtained was IDR 4,884,985, where 27 kg was the weight of the goods that could not be shipped on July 1, 2025 with a profit of IDR 97,000 and 673 kg was the weight of the goods selected to be shipped on July 2, 2025.

2) Selection of items to be shipped on July 3, 2025

Items to be shipped on July 3, 2025 consist of items that were not shipped on July 2, 2025 and items that were newly received on July 3, 2025 that will be selected for shipping.There are 73 new items that have just arrived with a total weight of 2041 kg and a total profit of IDR 13,017,575, with a maximum shipping load capacity of$(M)$to the Sokaraja branch office amounting to 700 kg and the selection of incoming goods delivery on July 3, 2025 will be carried out by reducing the maximum capacity of the delivery load with the total weight of the goods on July 2, 2025 that cannot be sent, namely 50 kg, so that the maximum delivery load capacity is 650 kg.

```
>> weights = [50, 16, 11, 13, 14, 136, 11, 11, 11, 15, 11, 11, 11, 11, 37, 66, 19, 12, 11, 11, 11, 30, 11, 11, 11, 26, 26, 11, 11, 22,
values = [159500, 70920, 152290, 57000, 173000, 1001200, 40500, 45000, 40500, 65000, 153500, 46000, 137000, 49000, 930000, 249500, 81
capacity = 650;
[max_val, selected, total_weight] = knapsack_dp(weights, values, capacity);
disp(['Max Value: ', num2str(max_val)]);
disp('Selected Items: ');
disp(selected);
disp(['Total Weight: ', num2str(total_weight)]);
Max Value: 7588240
Selected Items:
  Columns 1 through 33

   0   0   1   0   1   1   0   0   0   0   1   0   1   0   1   0   0   0   0   0   0   0   1   1   0   0   0   0   0   0   0   0   0

  Columns 34 through 66

   0   0   0   1   0   0   0   0   0   0   0   1   1   0   0   0   1   1   1   0   0   0   0   0   1   0   1   1   0   0   0   0   0

  Columns 67 through 73

   0   1   1   0   0   0   0

Total Weight: 649
```

Figure 3: Matlab R2021A output of selection of items shipped on July 3, 2025

Based on Figure 3, the items that will be included in the shipping loadnamely items 3, 5, 6, 11, 13, 15, 23, 24, 37, 45, 46, 50, 51, 52, 58, 60, 61, 68, 69 with a maximum profit of IDR 7,588,240 and a maximum weight of 649 kg. The total weight of the goods transported on July 3, 2025, was 699 kg.and the total profit obtained was IDR 7,732,155 where 50 kg was the weight of the goods that could not be sent on July 2, 2025 with a profit of IDR 143,915 and 649 kg was the weight of the goods selected to be sent on July 3, 2025.

5. Conclusion

5.1. Conclusion

Based on the research that has been conducted, the following conclusions were obtained.

1) The optimization model obtained for the integer knapsack problem 0/1 using this dynamic programming algorithm is as follows:

$$f_i(y) = \begin{cases} f_{i-1}(y) & if\ y < w_k \\ max\{f_{i-1}(y), p_k + f_{i-1}(y - w_k)\} & if\ y \geq w_k \end{cases} \quad (6)$$

By k stating the number of items and y stating the load capacity, it can be concluded:

a. On July 1, 2025, 34 items were imported with a loading capacity of 700 kg.
b. On July 2, 2025, 44 items were shipped, and the loading capacity was 673 kg.
c. On July 3, 2025, there were 73 items entered, and the loading capacity was 650 kg.

2) In the J&T Cargo Purwokerto case study data from July 1-3, the results of solving the integer knapsack problem 0/1 using dynamic programming were obtained as follows:

a. On July 1, 2025, after calculations were carried out using the dynamic programming algorithm, 32 items were obtained to be transported on that day with a maximum weight of 700 kg and a maximum profit of IDR 3,038,850.
b. On July 2, 2025, after calculations were carried out using the dynamic programming algorithm, a total of 43 items were obtained that would be transported on that day with a maximum weight of 700 kg and a maximum profit of IDR 4,884,985.

c. On July 3, 2025, after calculations were carried out using the dynamic programming algorithm, a total of 22 items were obtained that would be transported on that day with a maximum weight of 699 kg and a maximum profit of IDR 7,732,155.

5.2. Suggestion

For further research, it can be developed by using other variables such as the volume variable of goods so that it does not only use the weight and profit variables of goods. It can also use the backward recursive dynamic programming algorithm or use other algorithms that can solve the integer knapsack problem. Furthermore, it can also be developed by using different knapsack problems such as bounded knapsack or fractional knapsack to adjust the problem to the existing reality.

References

Agrawal, S., & Jain, R. (2018). "Implementation of 0/1 Knapsack Problem using Dynamic Programming Approach." International Journal of Advanced Research in Computer Science, 9(2), 22-26.

Alp, O., Ergun, O., Luss, H., & Weintraub, A. (2006). "Solving the 0-1 Knapsack Problem Using a Branch-and-Bound Algorithm." European Journal of Operational Research, 171(2), 527-540. (Walaupun terbit sebelum 2015, jurnal ini sering dikutip dalam penelitian-penelitian terbaru dan relevan.)

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). Introduction to Algorithms (4th ed.). MIT Press.

Devita, RN, & Wibawa, AP (2020). Knapsack Problem Optimization Techniques. Science, Applications, Computing and Information Technology, Vol. 2 No. 1, 35–40.

Dutta, R., & Barman, G. (2020). "A Survey on Algorithms for the 0/1 Knapsack Problem." International Journal of Emerging Trends in Engineering Research, 8(4), 1152-1157.

Ezugwu, A. E., Akwu, F. I., & Agajo, J. (2016). "A Comparative Analysis of Heuristic Algorithms for the 0/1 Knapsack Problem." Journal of Computer Science and Engineering, 8(2), 1-10.

Kellerer, H., Pferschy, U. & Pisinger, D. (2004). Knapsack Problems. New York: Springer-Verlag.

Kleinberg, J., & Tardos, É. (2006). Algorithm Design. Addison-Wesley.

Martello, S., & Toth, P. (1990). Knapsack Problems: Algorithms and Computer Implementations. John Wiley & Sons.

Messac, A. (2015). Optimization in Practice with MATLAB. Cambridge University Press.

Pitaloka, ADA, (2017). 0-1 Knapsack Optimization Problem Using Dynamic Programming Algorithm. (Thesis). Jakarta State University, Jakarta.

Rois, MA, Maslihah, S. & Cahyono, B. (2019). Solving Integer Knapsack Problem Using Greedy, Dynamic Programming, Brute Force, and Genetic Algorithms. Telematika, Vol. 12 No. 2, 87-97.

Salsabila, K. (2022). Application of Greedy Algorithms and Dynamic Programming Algorithms to Integer Knapsack Problems. (Thesis). Jenderal Soedirman University, Purwokerto.

Sampurno, GI, Sugiharti, E. & Alamsyah. (2018). 'Comparison of Dynamic programming Algorithm and Greedy Algorithm on Integer knapsack Problem in Freight Transportation', Scientific Journal of Informatics, Vol. 5 No. 1, 40-49.

Sedgewick, R., & Wayne, K. (2011). Algorithms (4th ed.). Addison-Wesley Professional.

Sharma, S., & Kumar, R. (2017). "Solving the Knapsack Problem Using Genetic Algorithm and Dynamic Programming: A Comparative Study." International Journal of Engineering and Technology, 9(4), 2826-2831.