BAREKENG: Journal of Mathematics and Its Applications
December 2024 Volume 18 Issue 4 Page 2449–2458
P-ISSN: 1978-7227 E-ISSN: 2615-3017
https://doi.org/10.30598/barekengvol18iss4pp2449-2458

THE BRANCH AND BOUND APPROACH TO A BOUNDED KNAPSACK PROBLEM (CASE STUDY: OPTIMIZING OF PENCAK SILAT MATCH SESSIONS)

Aditya Ambarwati[1], Sobri Abusini[2*], Vira Hari Krisnawati[3]

[1,2,3] *Department of Mathematics, Faculty of Mathematics and Natural Sciences, Universitas Brawijaya*
Jln. Veteran Malang, Ketawanggede, Malang, Jawa Timur, 65145, Indonesia

*Corresponding author's e-mail: * **sobri@ub.ac.id***

Article History:

Received: 5th, April 2024
Revised: 2nd, June 2024
Accepted: 29th, July 2024
Published:14th, October 2024

Keywords:

Bounded Knapsack Problem; Branch and Bound Approach; Optimization.

ABSTRACT

A method commonly employed to solve integer programming problems is the Branch and Bound. In this article, maximizing the number of matches held on the first day of pencak silat tournaments is essential because it can impact the overall dynamics and results of the competition. The model used to maximize the number of match sessions in pencak silat competitions is a variant of the Bounded Knapsack Problem (BKP), belonging to the category of integer programming models. The result obtained using the Branch and Bound method ensures that the maximum number of match sessions can be conducted. The objective value obtained using the Branch and Bound method decreases as it descends, indicating a decreasing maximum value.

How to cite this article:

A. Ambarwati, S. Abusini and V. H. Krisnawati., "THE BRANCH AND BOUND APPROACH TO A BOUNDED KNAPSACK PROBLEM (CASE STUDY: OPTIMIZING OF PENCAK SILAT MATCH SESSIONS)," *BAREKENG: J. Math. & App.*, vol. 18, iss. 4, pp. 2449-2458, December, 2024.

Journal homepage: *https://ojs3.unpatti.ac.id/index.php/barekeng/*
Journal e-mail: *barekeng.math@yahoo.com; barekeng.journal@mail.unpatti.ac.id*

Research Article · Open Access

1. INTRODUCTION

The Knapsack Problem (KP) is commonly encountered in two scenarios: firstly, when maximizing the utilization of space by selecting items based on their values; and secondly, when dividing space into sections with varying values to achieve the highest possible value cuts [1]. The goal of KP is to maximize overall satisfaction by choosing a subset of items from a given set with specific weights and values without exceeding certain capacity constraints. The KP can be formulated as either an integer or binary programming, depending on the context [2]. The variations of the Knapsack Problem include the Binary Knapsack Problem (KP01), Bounded Knapsack Problem (BKP), and Unbounded Knapsack Problem (UKP) [3].

The Binary Knapsack Problem (KP01) is characterized by solutions represented as either 0 or 1, where 0 indicates not specifying an item and 1 indicates including an item in the knapsack [4], [5]. KP01 revolves around selecting a combination of items that produces the highest possible profit value while remaining within the knapsack's weight capacity [6]. The Bounded Knapsack Problem (BKP) generalizes the concept of KP01 by allowing for a bounded quantity of each type of item to be available [7], [8] The Unbounded Knapsack Problem (UKP) aims to determine the quantity of each type of item to be chosen to maximize the total profit while ensuring that the total weight does not exceed the capacity [9]. The Unbounded Knapsack Problem (UKP) is a problem where a set of item types is given without any limit on their quantity. Each item type has a weight (w_i) and value (v_i), which are the same for all items of that type [10]. The KP is widely acknowledged as a significant challenge in combinatorial optimization. It has numerous real-world applications, including budget allocation [11, 12], location determination [13, 14], resource distribution [15, 16, 17], and management [18, 19]. The KP, along with its various variations and practical applications, continues to be extensively researched. Based on the application of the KP in several problems, the KP will be applied to maximize of pencak silat match session on the first day.

There are several methods for solving the Knapsack Problem such as Greedy Algorithm [20], Branch and Bound method [21, 22], and Dynamic Programming [23, 24]. Due to the Knapsack problem being a type of Integer Programming, it can be resolved by utilizing the Branch and Bound method [25]. Branch and Bound (B&B) is a fundamental methodology for solving exact optimization problems. This method computes solutions by storing subproblems in a tree structure and recursively dividing the solution space into smaller regions (branching). Rules are used to trim regions that are proven to be suboptimal (bounding). Once the tree is explored, the best solution is returned. It's a family of algorithms that share the same core solution procedure. [26]. In the initial stage of the Branch and Bound method, Linear Programming (LP) Relaxation is performed, where eliminating the integer constraints [27]. The solution from this LP is then obtained to ensure that the results are in fractional or integer form. If the solution is fractional, the next step is to perform branching. The branching process involves splitting the fractional solution value into higher and lower integer values. The results are then inserted into the initial problem constraints and tested to obtain the solution [28]. This branching process continues until an optimal solution with integer values is obtained [29, 30].

Pencak Silat is a martial art inherited from ancestors and is part of Indonesia's cultural heritage, thus it requires preservation, cultivation, and development [31]. In Pencak Silat competitions, the duration of match sessions differs between Tanding and Artistic categories [32]. For each tournament, many bouts need to be organized, considering factors such as the number of participants, competition categories, arena availability, as well as the needs of athletes and coaches. Effective scheduling is crucial to ensure the smooth running of tournaments and to avoid potential conflicts or unfairness during matches. In creating the match schedule, the total number of match sessions will be allocated across the days of the competition. In the creation of pencak silat match schedules, the main objective is to conduct match sessions with optimal timing and maximize the number of match sessions that can be held. This requires balancing between time efficiency and the number of match sessions conducted. Therefore, schedule planning must consider arrangements that minimize the time between matches while ensuring that as many match sessions as possible can be held within the available timeframe.

In the District-level Pencak Silat Championship in Sleman Regency in 2023, the optimization of match sessions is necessary because the number of match sessions exceeds the initial planning. The first day of this competition is optimized using the KP because this approach allows for the selection of a combination of matches that will provide maximum value within the available time constraint. Therefore, the use of this approach aids in creating an efficient and optimal schedule for the first day of the competition. The KP used to maximize the number of match sessions in pencak silat competitions is the Bounded Knapsack Problem. This problem will be solved using the Branch and Bound method

2. RESEARCH METHODS

The main aim of this research is to solve the model formulation to maximize the number of pencak silat match sessions on the first day using the Branch and Bound method. The model formulation for maximizing the number of pencak silat match sessions takes the form of a Bounded Knapsack Problem. The software LINDO 6.1 is employed in implementing the Branch and Bound method.

2.1 Bounded Knapsack Problem

Given a collection of item types $\boldsymbol{N} = \{\mathbf{1}, \dots, \boldsymbol{n}\}$, where each item of type $\boldsymbol{j}$ $(\boldsymbol{x_j})$ has a value $(\boldsymbol{v_j})$ and weight $(\boldsymbol{w_j})$. There are $\boldsymbol{b_j}$ items of type $\boldsymbol{j}$. The formulation of the Bounded Knapsack Problem (BKP) is described as follows [27]:

$$\max \sum_{j=1}^{n} v_j x_j \tag{1}$$

subject to:

$$\sum_{j=1}^{n} w_j x_j \leq W \tag{2}$$

$$x_j \leq b_j; \ \ j = 1, 2, \dots, n \tag{3}$$

$$x_j \in N_0; \ \ j = 1, 2, \dots, n \tag{4}$$

2.2 Branch and Bound Method

The branch and Bound method is typically represented using enumeration trees [28]. For example, suppose the solution in $\boldsymbol{LP}$ with the value of $\boldsymbol{x_2 = 3.4}$ is a fraction, then branching is conducted by creating new constraints, namely $\boldsymbol{x_2 \leq 3}$ and $\boldsymbol{x_2 \geq 4}$. After obtaining solutions for $\boldsymbol{A}$ and $\boldsymbol{B}$, they are re-evaluated. If the value of $\boldsymbol{x_1}$ is a fractional number, then new constraints are created with integers below and above it. The process ends when integer solutions are obtained for each branch. Figure 1 depicted below showcases the settlement procedure employing the Branch and Bound.

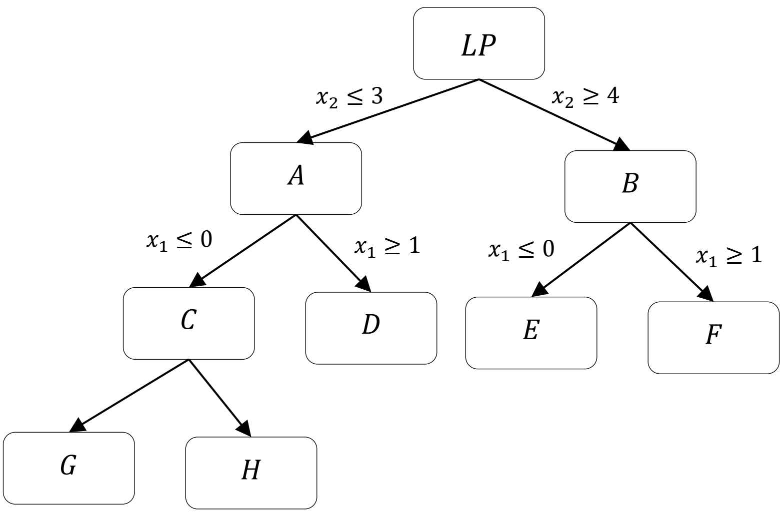


Figure 1. Branch and Bound Method Representation with Enumeration Trees

The steps in the Branch and Bound Method are as follows:

1. LP Relaxation. The KP is part of Integer Linear Programming (ILP). ILP is transformed into LP. LP is solved using the Simplex Method to obtain a solution. If the obtained solution is a fraction, branching is then initiated.

2. Branching. Fractional solutions are branched into integer values above and below them. These integer values are then inserted into the constraints of the original problem and the solutions are checked. If a fractional value is obtained, branching continues until an optimal integer value is achieved.

The search strategy in the Branch and Bound Algorithm determines the order in which unexplored subproblems in the enumeration tree are explored. One of the search strategies that can be used is Breadth-First Search (BrFS) [29]. BrFS can find the optimal solution closest to the root. In this article, it is described how to obtain an integer solution from the subproblem closest to the root. For other solutions, the process can be continued in the same way until all integer solutions are found using the Branch and Bound method.

In this article, parts of the enumeration trees in the Branch and Bound method are highlighted in red to indicate no feasible solution, while those highlighted in yellow represent a feasible solution. "No feasible solution" means that there is no solution that satisfies all the constraints in the given model. Unhighlighted parts imply that the obtained result is not an integer.

2.3 Data

The research data for this study utilized information from the pencak silat championship in the District-level Pencak Silat Championship in Sleman Regency in 2023. The acquired data includes the match categories, number of match sessions, duration of each match session, and the match schedule for the first day. There are 2 arenas, each arena is allocated 450 minutes for matches on the first day, resulting in a total of 900 minutes. The competition comprises three levels: Elementary School (SD), Junior High School (SMP), and Senior High School (SMA). Each level consists of two categories: Tanding and Artistic. Details regarding the number of match sessions and the duration of each match session can be found in Table 1.

Table 1. Number and Duration of Each Match Categories

Match Categories		Number of Matches	Duration of Match
Elementary School	Tanding	154	5 minutes
	Artistic	32	6 minutes
Junior High School	Tanding	237	8 minutes
	Artistic	13	6 minutes
Senior High School	Tanding	187	8 minutes
	Artistic	12	9 minutes

2.4 Formulation of Bounded Knapsack Problem

We assume that the notations utilized in formulating the maximization problem of the number of pencak silat match sessions are as follows.

x_1 : Tanding category for Elementary School
x_2 : Artistic category for Elementary School
x_3 : Tanding category for Junior High School
x_4 : Artistic category for Junior High School
x_5 : Tanding category for Senior High School
x_6 : Artistic category for Senior High School
w_j : Duration for each match session of $x_j; j = 1, 2, \ldots, 6$
b_j : Availability of $x_j; j = 1, 2, \ldots, 6$
v_j : Value of $x_j; j = 1, 2, \ldots, 6$. In this case $v_1 = v_2 = \cdots = v_6 = 1$.
W : Match time on the first day

The BKP formulation of maximizing the number of pencak silat match sessions can be seen as follows:

$$\max(x_1 + x_2 + x_3 + x_4 + x_5 + x_6) \quad (5)$$

subject to:

$$5x_1 + 6x_2 + 8x_3 + 6x_4 + 8x_5 + 9x_6 \leq 900 \quad (6)$$

$$x_1 \leq 154 \quad (7)$$

$$x_2 \leq 32 \quad (8)$$

$$x_3 \leq 237 \quad (9)$$

$$x_4 \leq 13 \quad (10)$$
$$x_5 \leq 187 \quad (11)$$
$$x_6 \leq 12 \quad (12)$$
$$x_j \in N_0 \quad (13)$$

3. RESULTS AND DISCUSSION

In this section, we discuss the solution of maximizing the number of match sessions in pencak silat competitions using the Branch and Bound method. The solution to the model maximizing the number of pencak silat match sessions on the first day begins by formulating the LP relaxation of the model as follows: Maximizing **Equation (5)**, subject to **Equation (6)**-**(12)** and

$$x_j \in \mathbb{R} \quad (14)$$

The solution of the LP relaxation using LINDO 6.1 is obtained, $x_1 = 154, x_2 = 21.667, x_3 = 0, x_4 = 0, x_5 = 0, x_6 = 0$ and $Z = 175.667$. The result obtained is not an integer, so the next step is adding constraints: $x_2 \leq 21$ for part A and $x_2 \geq 22$ for part B. This is done to obtain all potential solutions to the problem. The solutions in parts A and B were executed using LINDO 6.1. The solution diagram for the Branch and Bound method can be seen in **Figure 2**.

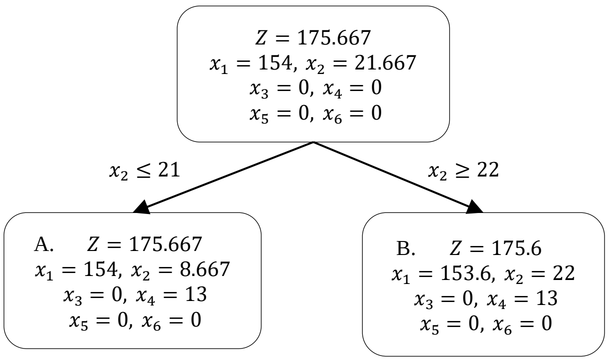
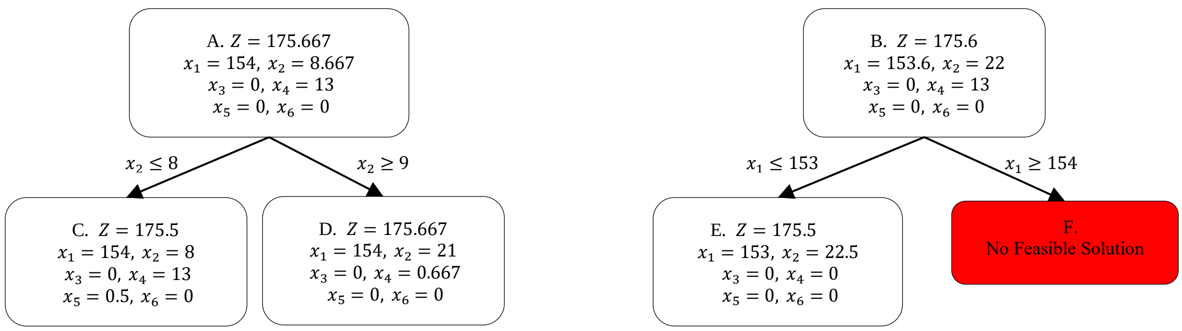


Figure 2. The Solution Diagram for the Initial Branching Step in the Branch and Bound method.

Based on the diagram in **Figure 2**, the solutions for parts A and B are not integers. In part A, an additional constraint of $x_2 \leq 8$ for part C and $x_2 \geq 9$ for part D is added. In part B, an additional constraint of $x_1 \leq 153$ for part E and $x_1 \geq 154$ for part F is added. The solutions in parts C, D, E, and F were executed using LINDO 6.1. The solution diagram for the Branch and Bound method can be seen in **Figure 3.**



(a) **(b)**

Figure 3. The Solution Diagram for the Second Branching Step in the Branch and Bound method, (a) Diagram for part A, (b) Diagram for part B.

Based on the diagram in **Figure 3**, the solutions for parts C, D, and E are not integers. Constraints are then added to these sections following the same procedure as earlier. The addition of these constraints results in parts G, H, I, J, K, and L. These parts are executed using LINDO 6.1. The solution diagram for the Branch and Bound method can be seen in **Figure 4**.

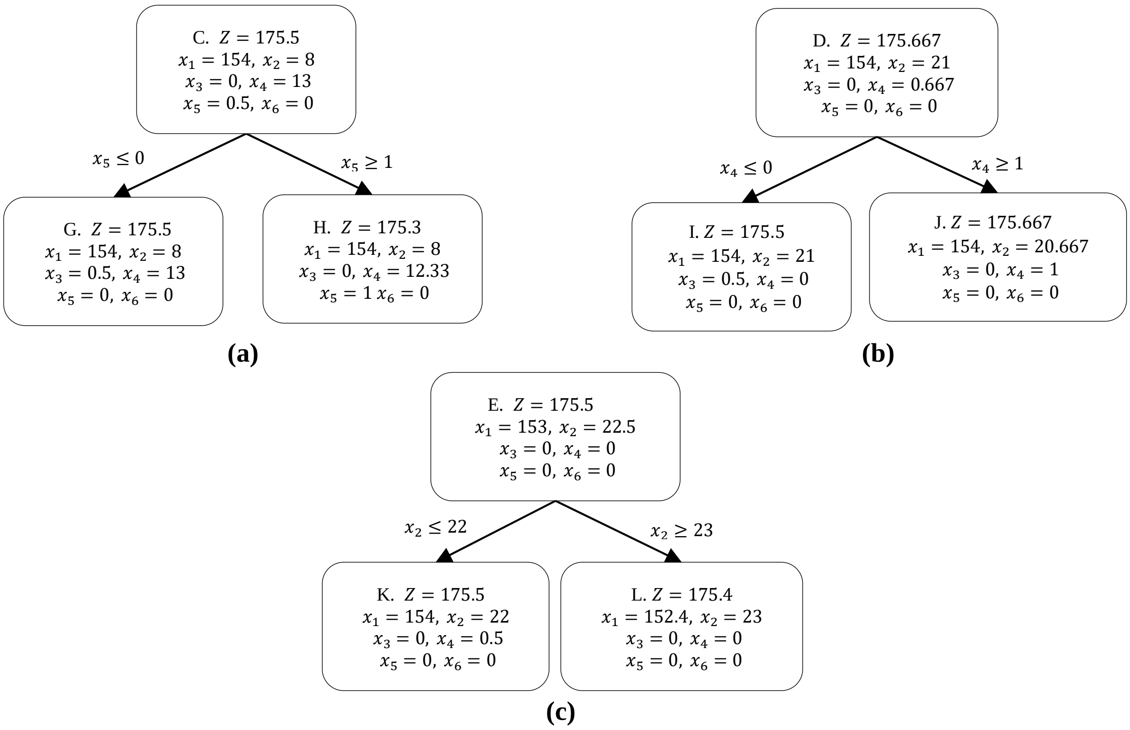
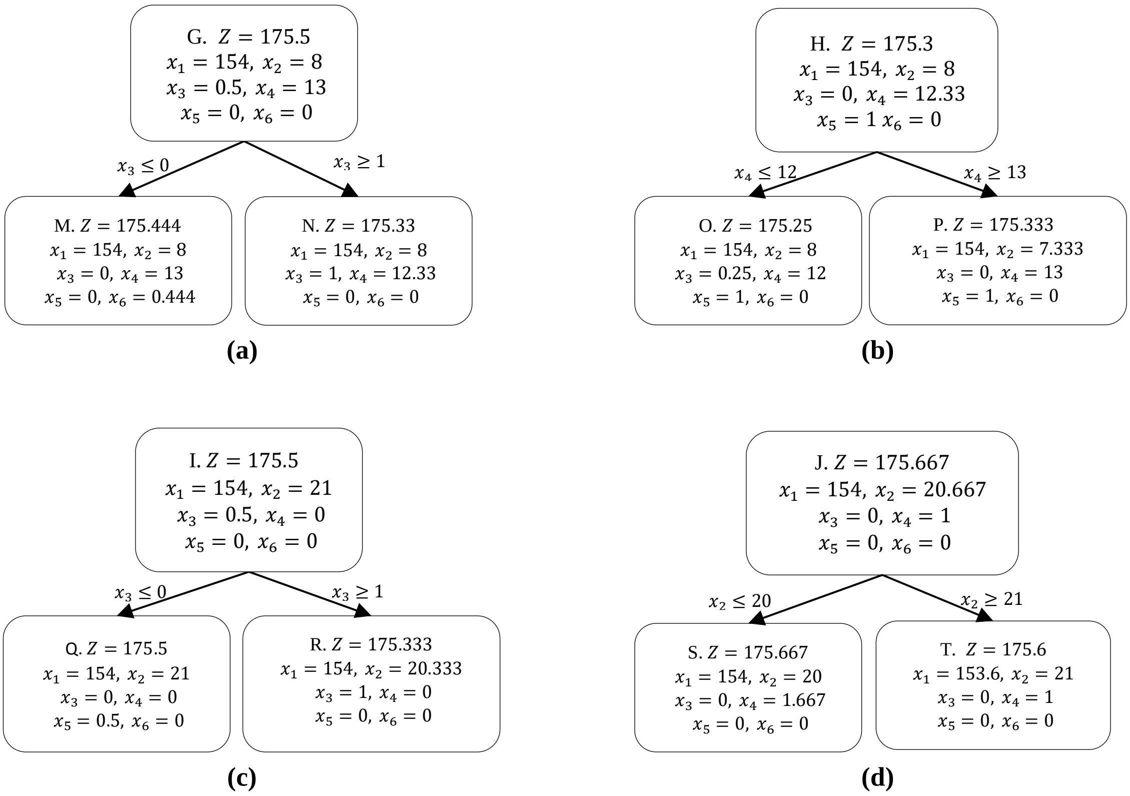


Figure 4. The Solution Diagram for the Third Branching Step in the Branch and Bound method, (a) Diagram for part C, (b) Diagram for part D, (c) Diagram for part E.

Based on the diagram in Figure 4 the solutions for parts G, H, I, J, K, and L are not integers. In these sections, additional constraints are added following the same procedure as before. The outcome of adding these constraints leads to parts M, N, O, P, Q, R, S, T, U, V, W, and X. Subsequently, these parts are solved using LINDO 6.1. The solution diagram for the Branch and Bound method can be seen in Figure 5.



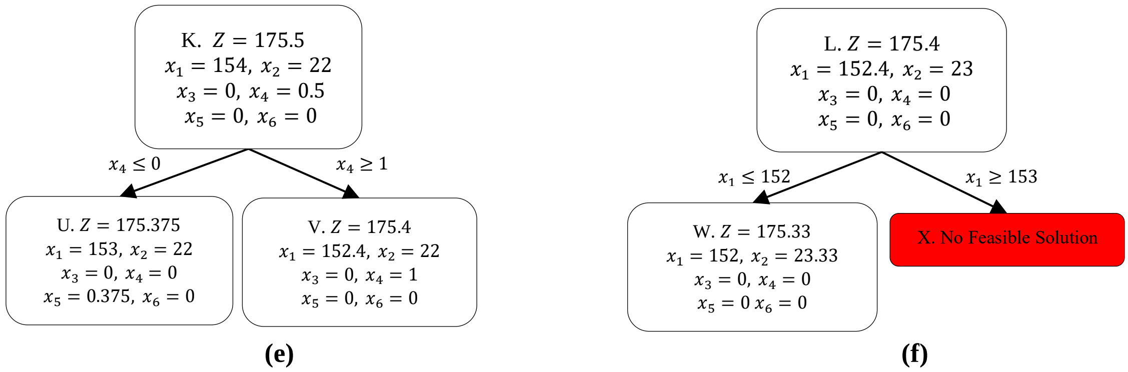
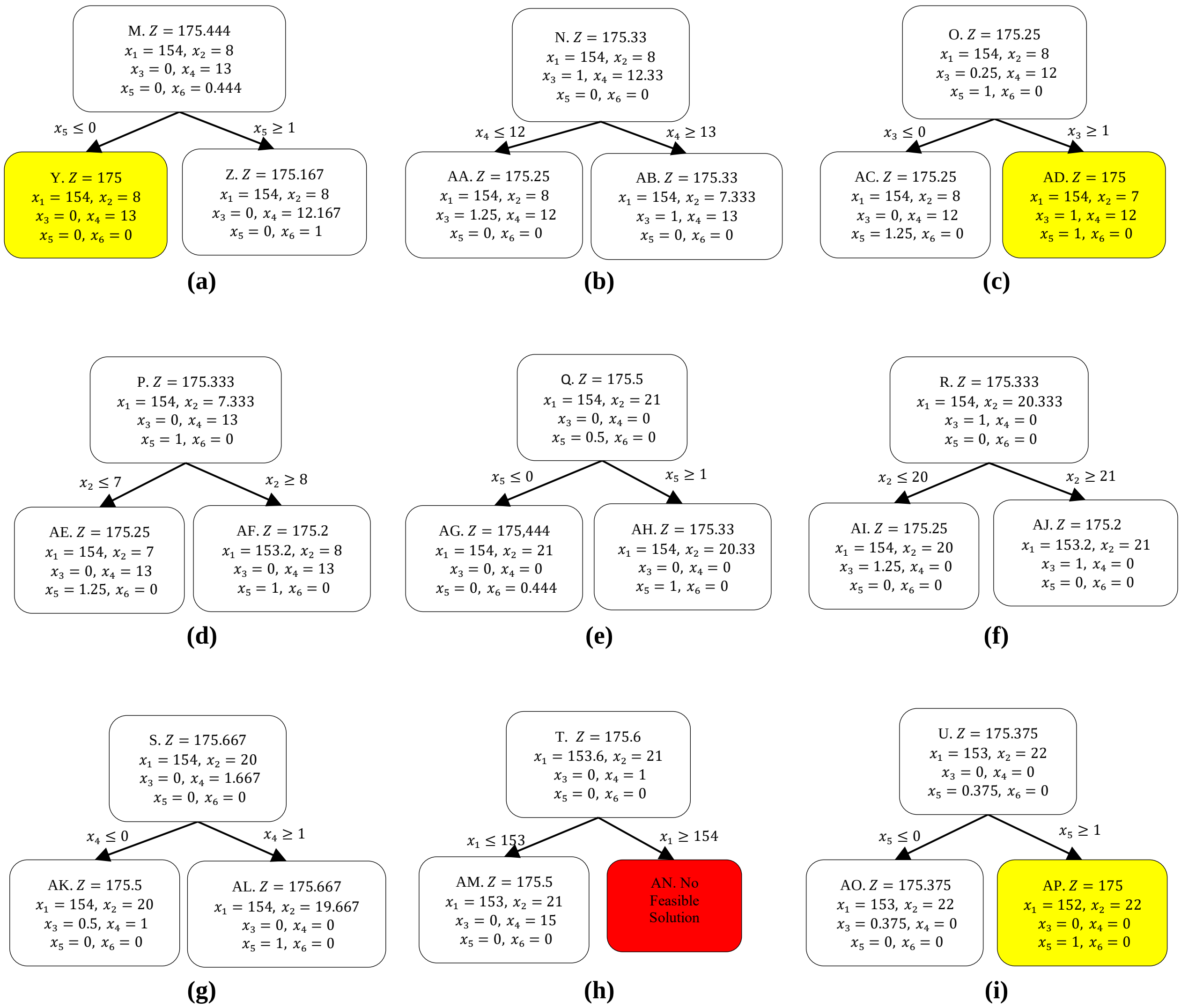


Figure 5. The Solution diagram for the fourth branching step in the Branch and Bound method, (a) Diagram for part G, (b) Diagram for part H, (c) Diagram for part I, (d) Diagram for part J, (e) Diagram for part K, (f) Diagram for part L.

Based on the diagram in **Figure 5**, the solutions for parts M, N, O, P, Q, R, S, T, U, V, W, and X are not integers. Additional constraints are introduced in these sections following the same steps as previously. These constraints result in the creation of parts Y, Z, AA, AB, AC, AD, AF, AG, AH, AI, AJ, AK, AL, AM, AN, AO, AP, AQ, AR, AS, and AT. These parts are then resolved using LINDO 6.1. The solution outcomes from the branch and bound method are in **Figure 6**.



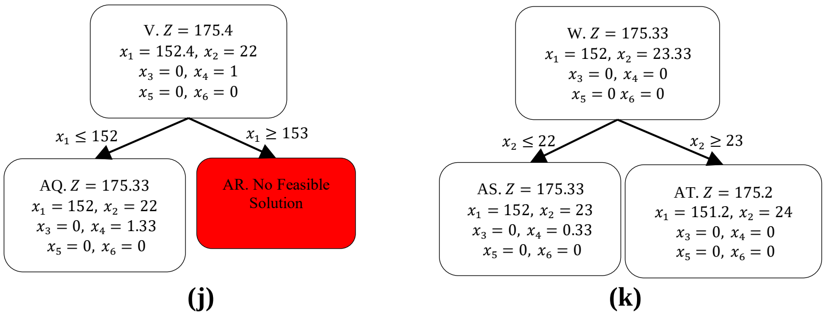


Figure 6. The Solution Diagram for the Fifth Branching Step in the Branch and Bound Method, (a) Diagram for part M, (b) Diagram for part N, (c) Diagram for part O, (d) Diagram for part P, (e) Diagram for part Q, (f) Diagram for part R, (g) Diagram for part S, (h) Diagram for part T, (i) Diagram for part U, (j) Diagram for part V, (k) Diagram for part W.

Based on the diagram in Figure 6, there are 3 feasible solutions. Next, each part that is not integer will be treated similarly to the previous ones. Overall, all parts are depicted as shown in Figure 7.

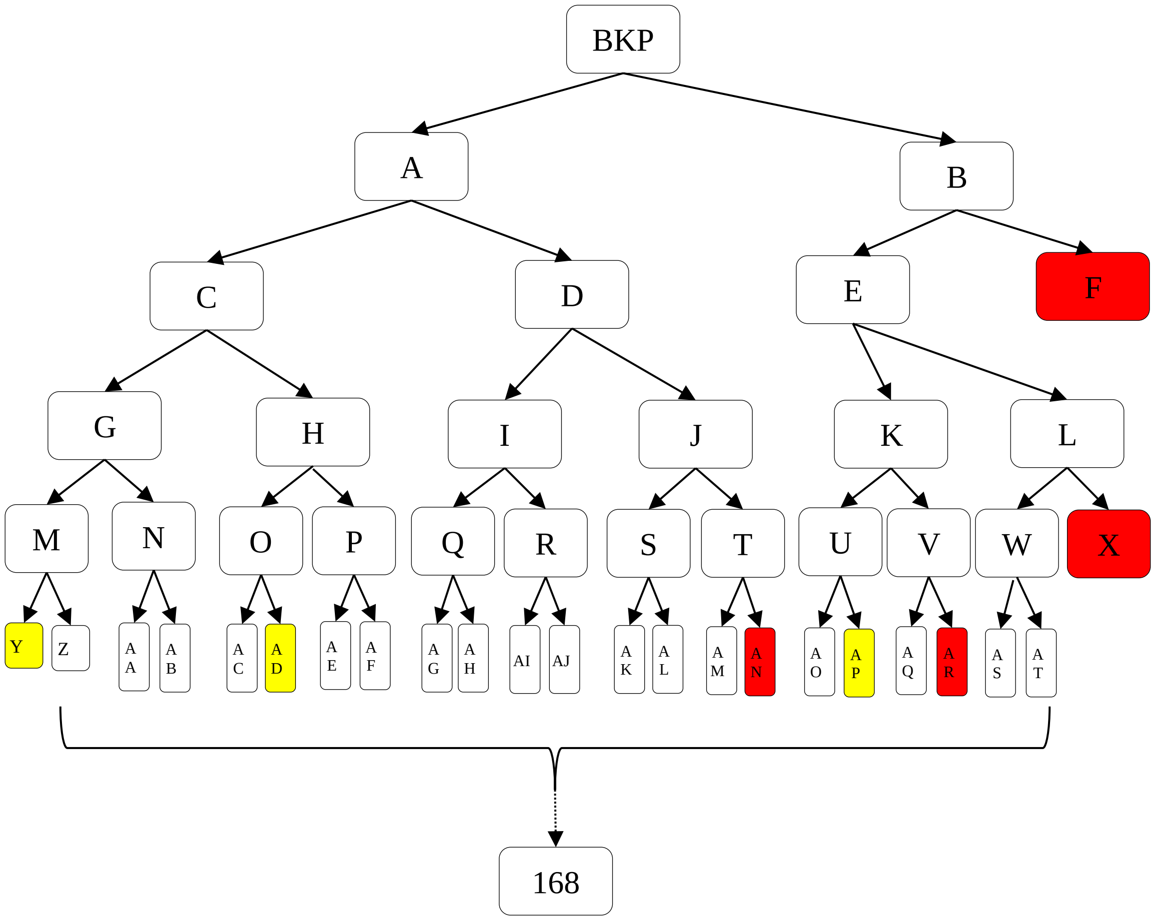


Figure 7. Diagram for all Parts of Branch and Bound Method

Based on the potential solutions in Figure 7, the potential solutions obtained are parts Y, AD, and AP with the maximum value $Z = 175$. In part Y, the solution is $x_1 = 154,\ x_2 = 8,\ x_3 = 0,\ x_4 = 13,\ x_5 = 0,\ x_6 = 0$. In part AD, the solution is $x_1 = 154,\ x_2 = 7,\ x_3 = 1,\ x_4 = 12,\ x_5 = 1,\ x_6 = 0$. In part AP, the solution is $x_1 = 152,\ x_2 = 22,\ x_3 = 0,\ x_4 = 0$, and $x_5 = 1,\ x_6 = 0$. For the remaining parts that are not integers, the branching process continues until all solutions are integers. The resulting minimum value of Z obtained is 168. The problem can be solved using integer programming directly in LINDO. The output from the integer programming in LINDO will be part of the solution set generated by the Branch and Bound method

4. CONCLUSIONS

Based on the result, the conclusions that can be drawn specifically and generally from this study are as follows:

1. The application of the Branch and Bound method in maximizing the number of pencak silat match sessions in the District-level Pencak Silat Championship in Sleman Regency in 2023 on the first day gets the optimal result. The first three combination solutions are obtained, namely sections Y, AD, and AP with a total of 175 match sessions. In part Y, the solutions are 154 matches of the Tanding category for Elementary School, 8 matches of the Artistic Category for Elementary School, and 13 matches of the Artistic Category for Junior High School. In part AD, the solutions are 154 matches of the Tanding category for Elementary School, 7 matches of the Artistic Category for Elementary School, 1 match Tanding Category for Junior High School, 12 matches Artistic Category for Junior High School, and 1 match for the Tanding Category for Senior High School. In part AP, the solutions are 152 matches of the Tanding category for Elementary School, 22 matches the Artistic Category for Elementary School, and 1 matches of the Tanding Category for Senior High School. Other combinations of solutions can be obtained by completing the branch and bound method, including the solution obtained from integer programming with LINDO.
2. The problem of maximizing the number of pencak silat match sessions on the first day, modeled using the Bounded Knapsack Problem, can be addressed using the Branch and Bound method. The result obtained through the Branch and Bound method confirms the feasibility of conducting the maximum number of match sessions. The objective value obtained from the Branch and Bound method decreases as it descends, indicating a diminishing maximum value. This is attributed to the equal importance values across each category. Subsequent research may involve assigning different importance values to each matched category, applying it to other cases, or using other resolution methods.

REFERENCES

[1] P. C. Gilmore and R. E. Gomory, "The theory and computation of knapsack functions," *Operations Research,* vol. 14, no. 6, pp. 1045-1074, 1966.

[2] U. Pferschy, J. Schauer and C. Thielen, "Approximating the product knapsack problem," *Optimization Letters,* vol. 15, no. 8, pp. 2529-2540, 2021.

[3] M. Assi and R. A. Haraty, "A survey of the knapsack problem," in *In 2018 International Arab Conference on Information Technology (ACIT),* 2018.

[4] J. Jooken, P. Leyman and P. De Causmaecker, " A new class of hard problem instances for the 0–1 knapsack problem," *European Journal of Operational Research,* vol. 301, no. 3, pp. 841-854, 2022.

[5] J. Jooken, P. Leyman and P. De Causmaecker, "Features for the 0-1 knapsack problem based on inclusionwise maximal solutions," *European Journal of Operational Research,* vol. 311, no. 1, pp. 36-55, 2023.

[6] V. Cacchiani, M. Iori, A. Locatelli and S. Martello, "Knapsack problems—An overview of recent advances. Part II: Multiple, multidimensional, and quadratic knapsack problems," *Computers & Operations Research,* p. 143, 2022.

[7] D. Pisinger, "A minimal algorithm for the bounded knapsack problem," *INFORMS Journal on Computing,* vol. 12, no. 1, pp. 75-82., 2000.

[8] Z. Li, Y. He, H. L. Y. Li and X. Guo, "A novel discrete grey wolf optimizer for solving the bounded knapsack problem," in *In Computational Intelligence and Intelligent Systems: 10th International Symposium, ISICA 2018*, Jiujiang, China,, 2019.

[9] S. Martello and P. Toth, Knapsack problems: algorithms and computer implementations, John Wiley & Sons, Inc., 1990.

[10] H. Becker and L. S. Buriol, "An empirical analysis of exact algorithms for the unbounded knapsack problem," *European Journal of Operational Research,* vol. 277, no. 1, pp. 84-99, 2019.

[11] M. Jemmali, "An optimal solution for the budgets assignment problem.," *RAIRO-Operations Research,* vol. 55, no. 2, pp. 873-897., 2021.

[12] T. Fukunaga, "Approximation algorithms for a generalization of the maximum budget allocation," *Journal of the Operations Research Society of Japan,* vol. 64, no. 1, pp. 31-44, 2021.

[13] R. Piedra de la Cuadra, "Knapsack models applied to the solution of complex problems in transport planning," 2023.

[14] K. T. Nguyen and H. M. Le, "An integrated approach for allocation and scheduling-location problems on graphs," *Computational and Applied Mathematics,* vol. 43, no. 3, p. 147, 2024.

[15] R. Kumar and R. Kleinberg, "Non-monotonic resource utilization in the bandits with knapsacks problem," *Advances in Neural Information Processing Systems,* vol. 35, pp. 19248-19259, 2022.

[16] B. Sun, A. Zeynali, T. Li, M. Hajiesmaili, A. Wierman and D. H. Tsang, "Competitive algorithms for the online multiple knapsack problem with application to electric vehicle charging.," *Proceedings of the ACM on Measurement and Analysis of Computing Systems,* vol. 4, no. 3, pp. 1-32, 2020.

[17] V. K. Tchendji and Y. F. Yankam, "Dynamic resource allocations in virtual networks through a knapsack problem's dynamic programming solution," *Revue Africaine de Recherche en Informatique et Mathématiques Appliquées,*, vol. 31, 2020.

[18] C. Hertrich and M. Skutella, "Provably good solutions to the knapsack problem via neural networks of bounded size," *INFORMS journal on computing,* vol. 35, no. 5, pp. 1079-1097, 2023.

[19] L. Han, C. Hao, C. Wu and Z. Zhang, "Approximation algorithms for the lower-bounded knapsack median problem," *In International Conference on Algorithmic Applications in Management.,* pp. 119-130, 2020 August.

[20] N. Ferdosian, M. Othman, B. M. Ali and K. Y. Lun, "Greedy–knapsack algorithm for optimal downlink resource allocation in LTE networks," *Wireless Networks,* vol. 22, pp. 1427-1440, 2016.

[21] M. Aïder, O. Gacem and M. Hifi, "Branch and solve strategies-based algorithm for the quadratic multiple knapsack problem," *Journal of the Operational Research Society,* vol. 73, no. 3, pp. 540-557, 2022.

[22] S. Coniglio, F. Furini and P. San Segundo, "A new combinatorial branch-and-bound algorithm for the knapsack problem with conflicts," *European Journal of Operational Research,* vol. 289, no. 2, pp. 435-455, 2021.

[23] S. Coniglio, F. Furini and P. San Segundo, "A new combinatorial branch-and-bound algorithm for the knapsack problem with conflicts.," *European Journal of Operational Research,* vol. 289, no. 2, pp. 435-455, 2021.

[24] M. Masmoudi, Y. Adouani and B. Jarboui, "LP relaxation and dynamic programming enhancing VNS for the multiple knapsack problem with setup," *International Transactions in Operational Research,* vol. 31, no. 3, pp. 1890-1916, 2024.

[25] M. Al-Rabeeah, E. Munapo, A. Al-Hasani, S. Kumar and A. Eberhard, "Computational enhancement in the application of the branch and bound method for linear integer programs and related models," *International Journal of Mathematical, Engineering and Management Sciences,* vol. 4, no. 5, pp. 1140-1153, 2019.

[26] R. M. Kolpakov, "Optimal strategy for solving a special case of the knapsack problem by the branch and bound method," *Moscow University Mathematics Bulletin,* pp. 97-106, 2021.

[27] M. Cho, "The knapsack problem and its applications to the cargo loading problem," *Anal. Appl. Math,* vol. 13, pp. 48-63, 2019.

[28] W. L. Winston, Operations research: applications and algorithms, Cengage Learning, 2022.

[29] D. R. Morrison, S. H. Jacobson, J. J. Sauppe and E. C. Sewell, "Branch-and-bound algorithms: A survey of recent advances in searching, branching, and pruning," *Discrete Optimization,* vol. 19, pp. 79-102, 2016.

[30] L. A. Wolsey, Integer programming, John Wiley & Sons., 2020.

[31] E. S. Kriswanto, Pencak Silat, Yogyakarta: Pustaka Baru Press, 2015.

[32] PERSILAT, Peraturan Pertndingan Pencak SIlat, Persilat, 2022.