PARAMETER
JURNAL MATEMATIKA, STATISTIKA DAN TERAPANNYA

e-ISSN. 2829-6303
Vol. 04 No. 3, December 2025
Page 409 - 428

Check for updates

Optimization of Assignment Problems in Private Class Scheduling Using Graph Application

Hanifah Felisia Wati[1], Sapti Wahyuningsih[2*], Muhammad Nur Ramadhan[3]

[1,2,3] Department of Mathematics, Faculty of Mathematics and Natural Sciences, University of Malang
Semarang Street, No. 5, Malang, 65145, East Java, Indonesia

E-mail Correspondence Author: *sapti.wahyuningsih.fmipa@um.ac.id*

Abstract

The tutoring institution PT. Inspirasi Mandiri Nusantara (PINTARA) provides various types of learning services for students of various levels. The services offered include regular, intensive, exam preparation, and private classes. Private class services face scheduling problems due to the limited number of tutors and incompatibility between the availability of tutors and the subjects offered. This article discusses the optimization of tutor assignment using the maximum matching algorithm on the bipartite graph and the Hungarian algorithm. The research uses a mathematical approach and the data is obtained through direct observation and modeled in the form of graphs, then solved with the help of the Python program. The results show that optimal assignments can be achieved by using the maximum matching algorithm, the Hungarian algorithm, and the Python program tools with the same and optimal values. This approach has proven to be effective and can be the basis for the development of automated scheduling systems in the future.

Keywords: *Bipartite Graph, Hungarian Algorithm, Maximum Matching Algorithm, Private Class Scheduling, Python.*

: https://doi.org/10.30598/parameterv4i1pp409-428

Research Ariticle ***Open Acces***
Email: jurnalparameter@gmail.com
Homepage: https://ojs3.unpatti.ac.id/index.php/parameter

Submitted: June 2025
Accepted: December 2025

1. INTRODUCTION

In the world of education, especially in tutoring institutions such as PT. Inspirasi Mandiri Nusantara (PINTARA), the efficiency of scheduling private classes is the main problem that affects the quality of academic services. Suboptimal scheduling often creates various obstacles such as schedule clashes between tutors and students, classes that are canceled due to unilateral absences, and mismatches between the availability of tutors and class requests from students. In the tutoring system, there will definitely be scheduling used to allocate time and human resources to get optimal results [1]. Therefore, a systematic approach is needed to address the issue of assignment in private class scheduling to improve service quality and operational efficiency.

In determining the right method, the problem of assignment is one of the problems that are often faced. The assignment problem is a problem that often arises in many cases of assignment optimization from a decision-making, for example optimization minimizes cost, time, distance and so on [2]. Recent research on traffic assignment issues shows that there are attempts to improve static models by considering residual queues during peak hours [3]. Meanwhile, in the world of retail and warehousing, assignment issues are more applied in order fulfillment management and goods placement. Recent research trends show an increased focus on assignment issues that take into account the variability of demand and consumer behavior, especially as demand increases online and the use of hybrid distribution strategies [4].

Modeling with graph theory can be used to solve everyday problems, where graphs consist of a set of vertices and a set of sides (connecting two vertices in a graph) [5]. Graph theory is widely applied in various fields. Recent research shows the development of the application of graph theory to various practical problems. In theory, recent research on linear algebra showing the problem of determining the greatest minimum distance in troubleshooting on locally recoverable code can be attributed to graph theory [6]. In the medical context, the application of graph theory is used to analyze changes in brain tissue through EEG signals in severely depressed patients, so that it can help predict the effectiveness of antidepressant treatment [7]. This trend suggests that graph theory is increasingly evolving as an adaptive approach in modern research.

The maximum matching algorithm and the Hungarian algorithm are two optimization approaches widely used in problems. Recent research shows that the maximum bipartite matching variant with separation constraints is quite complex, requiring an approximation algorithm with a constant factor to achieve optimal results [8]. In addition, the development of matching algorithms is also continuously carried out for applications on GNSS data, where the map matching algorithm is used to map travel data from smartphones to road networks, in order to improve the accuracy of driver behavior monitoring [9]. On the other hand, the Hungarian algorithm is a method widely applied in solving assignment problems, namely pairing elements from two sets with optimal results. Recent research trends indicate the combination of matching methods with other techniques to improve accuracy and efficiency. In the field of optics, the Hungarian algorithm is combined with denoising methods to improve the accuracy of lens measurements [10]. The advantages of these two algorithms are considered for use in assignment problems in private class scheduling as an efficient and adaptive solution.

This study uses the bipartite graph maximum matching algorithm and the Hungarian algorithm to optimize the scheduling of private classes at PT. Inspirasi

Mandiri Nusantara. The maximum matching algorithm aims to maximize the number of subjects that can be scheduled without conflict, while the Hungarian algorithm ensures an efficient allocation of tutors based on the availability, expertise, and desires of each tutor in each subject. In the process of both algorithms, calculations are also carried out using the Python program tool. Python is a popular programming language to date that was created by Guido van Rossum and released in 1991. The Python programming language can read and modify files, handle big data in complex mathematical problems, and software development. In the process Python runs on an interpreter system, i.e. the code can be executed as soon as it is written. Python can be written in an integrated development environment, such as Jupyter, Code Visual, Pycharm, Thonny, or Eclipse [11]. With this approach, this research is expected to produce a more efficient scheduling system by combining mathematical analysis and computational implementation.

The purpose of this study is to identify scheduling problems that occur in the field through direct observation data collection, conduct literature studies related to assignment problems and the application of graphs in class scheduling, and build mathematical modeling with maximum matching algorithms based on actual conditions in PINTARA. Furthermore, the problem was solved by applying the maximum matching algorithm to the bipartition graph, the Hungarian algorithm, and the Python program tool. The interpretation of the results of the three ways of completing the case study is described in this article.

2. METHOD

2.1 Type of Research

This research begins by identifying the scheduling problem of private classes at PT. Inspirasi Mandiri Nusantara (PINTARA), which is experiencing conflict due to limited tutors and uneven distribution of teaching. To solve this problem, a literature study was conducted related to bipartite graph theory, maximum matching algorithm, and Hungarian algorithm as the basis for modeling and mathematical approaches. Data were obtained through direct field observations during the period of February 10, 2025, to April 23, 2025, including tutor names, subject names, and tutor preferences for each subject. The data was then modeled into a bipartite graph with nodes representing tutors and subjects. Matching calculations and analysis were also carried out computationally using the Python program to perform the matching process and validate the results. The results of this matching were analyzed to obtain an optimal scheduling solution, then reinterpreted against the initial conditions to assess the effectiveness of the given solution.

2.2 Optimization Model

The scheduling optimization problem in this study is formulated using the assignment problem approach, which aims to determine the optimal assignment of tutors to students based on cost or suitability values. This mathematical model ensures that each student receives exactly one tutor and each tutor is assigned to at most one student in a single scheduling period.

In this study, the objective function Z represents the total assignment cost (or suitability value), which must be minimized to obtain the best matching results. The decision variable x_{ij} indicates whether tutor i is assigned to subject j. A value of $x_{ij} = 1$ means that tutor i teaches subject j, while $x_{ij} = 0$ indicates no assignment. The parameter

C_{ij} denotes the preference or suitability score given by a tutor for a specific subject, where lower values represent more preferred or more suitable assignments. Mathematically, the assignment problem can be formulated in the following form [12].
Minimize:

$$Z = \sum_{i=1}^{n}\sum_{j=1}^{n} C_{ij}x_{ij}$$

$$Z = C_{11}x_{11} + C_{22}x_{22} + \cdots + C_{nm}x_{nm} \quad (1)$$

with constraints:

$$\sum_{i=1}^{n} x_{ij} = 0 \; or \; 1, \;\; i = 1, 2, \ldots, n \quad (2)$$

$$\sum_{i=1}^{m} x_{ij} = 0 \; or \; 1, \;\; j = 1, 2, \ldots, m \quad (3)$$

$$x_{ij} \in \{0, 1\}, i = 1, 2, \ldots, n; j = 1, 2, \ldots, m \quad (4)$$

with:
Z : objective function (total assignment cost)
C_{ij} : cost or suitability value of assigning tutor i to subject j
x_{ij} : binary decision variable (1 = if tutor i assigned to subject j, 0 = otherwise)
i : index of tutors ($i = 1, 2, \ldots, n$)
j : index of subjects ($j = 1, 2, \ldots, m$)

These constraints guarantee a one-to-one matching structure that fits the scheduling needs at PT. Inspirasi Mandiri Nusantara (PINTARA), where each tutor can only handle a single subject and each subject must be taught by exactly one tutor during a session. Together, the objective function and constraints form a complete optimization model that reflects the actual scheduling rules and operational limitations in the institution. This clarification strengthens the mathematical formulation and ensures alignment with the context of the problem.

3. RESULTS AND DISCUSSION

3.1. Identify Tutor Assignment Problems for Subjects

PT. Inspirasi Mandiri Nusantara (PINTARA) is a tutoring institution that provides various types of learning services for students from various levels of education. The services offered include regular classes, intensive classes, exam preparation classes, and private classes. Of these various options, private classes are one of the superior services because they are personal and tailored to the needs and time of students and tutors. However, in its operational practice, PINTARA faces challenges in the private scheduling process. The main problem faced is the limited number of tutors compared to the number of subjects offered. There are 11 subjects, but there are only 8 tutors available and each tutor only has the ability to teach a few specific subjects. Therefore, optimal assignment mapping is needed so that all subjects can be carried out as much as possible. Based on the data on the list of tutors, the names of available subjects, and the availability of tutors in each subject that has been obtained are calculated in the form of a scale. The following is a table of representation of tutor preferences for each subject.

Table 1. Tutor Preference Data for Each Subject

S / T	MAT	FIS	KIM	BIO	PK	PM	PU	PBM	PPU	LBI	LBE
A	10	5	4	3	10	10	10	4	4	4	4
B	5	10	3	3	5	3	3	3	2	3	4
C	10	4	4	4	10	10	10	4	3	3	5
D	4	3	3	3	3	5	4	10	10	10	5
E	4	3	10	3	4	4	3	3	3	3	3
F	3	3	4	3	3	2	4	10	10	5	10
G	4	3	4	10	3	3	3	3	3	3	3
H	4	3	3	10	3	3	3	4	3	3	4

Source: PT. Inspirasi Mandiri Nusantara

From the table above, for the S code is the subject and the T code is the tutor. The order of subjects from S1 to S11 is mathematics, physics, chemistry, biology, quantitative knowledge (PK), mathematical reasoning (PM), general reasoning (PU), reading and writing comprehension (PBM), general knowledge and comprehension (PPU), Indonesian literacy (LBI), and English literacy (LBE). The scores in the table are the preferences of each tutor for each subject. It can be seen that not every tutor has the desire or availability in all subjects where the greater the score score, the more priority his assignment is given to those subjects. This indicates the need for assignment optimization to avoid overlapping loads. There are two sets where the assembly states the tutor and the assembly expresses the subject $V_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ and $V_2 = \{y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}\}$.

3.2. Bipartite Graph Matching Algorithm Analysis and Graph Modeling

In optimizing tutor assignment to subjects at PT. Inspirasi Mandiri Nusantara, the maximum matching algorithm for bipartite graphs is applied based on the preference data in **Table 1**. This approach uses a bipartite graph to model the graph between tutors and subjects, with the goal of maximizing the number of subjects that can be scheduled using the 8 available tutors.

1. Graf Matching

   In graph theory, matching is a set of sides in a graph that do not have the same vertices **[13]**. Matching in a G graph is a set of parts of the sides in such a way that no G node intersects with two sides **[14]**. In other words, each node in the matching is only connected to at most one side in the set. Many sides of the matching M are symbolized by $|M|$ and read cardinality matching M **[15]**. In matching, maximum matching and perfect matching are known. On a graph G, if $|M| \geq |M'|$, in other words the cardinality of M is the greatest of all the matching in M **[16]**. Perfect matching when matching on a graph involves all points on graph G **[17]**. In the context of a bipartite graph, which consists of two sets of nodes between a tutor and a subject, matching represents an assignment in which each tutor is assigned to at most one subject and each subject is assigned to at most one tutor. In this study, a bipartite graph matching was used to ensure that tutors were allocated to subjects according to their preferences (score ≥ 5 in **Table 1**), thereby reducing the risk of assignments that do not match the tutor's expertise or availability.

2. Maximum Matching Algorithm for Bipartite Graphs

Basically the maximum matching search on the graph is focused on the bipartite graph [13]. Maximum matching is matching with the largest possible number of sides in a bipartite graph, i.e. assignments that include as many pairs of tutors and subjects as possible without violating the matching rules. The following are the steps to complete the matching graph algorithm [17].

1. The first step is the labeling of the l.
   1a. $\forall v \in V_1$, for example $l = max_u \in V_2(v, u)$.
   1b. $\forall v \in V_2$, missal $l = 0$.
   1c. H_l is a subgraph of G_l that contains the sides of E_l.
   1d. For example, G_l is the basic graph of H_l.
2. Select a random matching from M in G_l.
3. If there is a maximum match in G_l, check if the match is perfect. If yes, then it is the maximum matching in G'_l. If you haven't already, create an alternating T tree that starts from an unsaturated node to help labeling the node l'.
   3a. If all the nodes in V_1 are saturated nodes in the matching M, then M is the maximum matching and the algorithm is complete. If not, then proceed to the next stage. A saturated node is a node that intersects (incident) with an edge matching M [18].
   3b. Suppose x is an unsaturated node in V_1. An unsaturated node is an incident node with an M matching edge [18].
   3c. Build an alternating tree from M with a root node x. If an augmenting-P trajectory is found in G_l, replace M with a new matching M′. If there is no augmenting-P trajectory, and T is an alternating tree of M that can no longer be expanded on G_l, then proceed to update the l' node label for the next process.
      a. Alternating trajectories are trajectories in a graph whose sides alternate between sides that are part of the matching and those that are not [18].

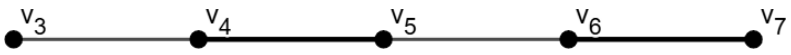


Figure 1. Alternating Path

      b. An augmenting trajectory is an alternating trajectory whose initial and final nodes are nodes that do not yet have an unsaturated [18].

Figure 2. Augmenting Path

4. Calculating the labeling of node l'. Let $m_l = min\{l(v) + l(u) - w(v, u) | v \in V_1 \cap V(T)$ and $u \in V_2 - V(T)\}$.
5. If the new labeling l' does not cover all the vertices in V_1, then the process returns to step 3c and is repeated.

3.3. Analysis of Hungarian Assignment Problems and Algorithms

The assignment problem is a problem that often arises in many cases of assignment optimization from a decision-making, for example optimization minimizes cost, time, distance and so on [2]. In its solution, this problem is generally expressed in the form of a matrix that contains the value of the preference entry to be assigned (tutor)

on the row, and the tasks or jobs available in column [19]. In this case, what is set is a number of tutors on a number of subjects in the most efficient way. One of the most well-known algorithms for solving assignment problems is the Hungarian algorithm. This algorithm works based on the principle of one-to-one optimal matching between two sets [10]. The following are the steps of the Hungarian algorithm for solving assignment problems [20].

1. Identify and simplify problems into the form of a matrix that represents preferences.
2. Determine the maximum value on each row, and then subtract that maximum value from all entries on that row.
3. Check each row and column to see if there is a value of 0. If it has, then it will be continued. If there is no value of 0 in a row or column, then subtract all entries by the smallest value in that row or column.
4. Create a closing line on rows and columns that contain a value of 0. Lines are drawn by selecting the row or column that has the most number of zeros to keep the number of lines as small as possible. If the number of lines drawn is equal to the number of rows or columns, then the solution is optimal. If not, proceed to step 5. This step is used to check if the optimal conditions have been achieved based on the required number of lines. Subtract all values that are not covered by the smaller value and add the value to the line intersection with the smaller value.
5. Subtract all the elements that are not covered by the line with the smallest value, and then add the smallest value to the elements that are at the intersection between the lines.
6. Repeat the line drawing process and checking until all rows and columns containing a value of 0 have been covered. If this condition is met, then the matrix is already in optimal shape.

3.4. Data Processing with Maximum Matching Algorithms

Based on the data in Table 1 related to tutor preferences for subjects, modeling was carried out in the form of a bipartite graph. This graph is formed from two sets of nodes, namely the tutor set $V_1 = \{x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8\}$ and a set of subjects $V_2 = \{y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}\}$. One side connects the tutor node and the subject node if there is a preference score of more than zero between the two. These formed graphs form a complete bipartition graph because the set of partitions X is connected to each of the nodes in Y by exactly one side [21].

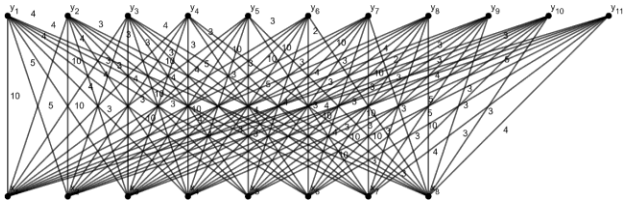


Figure 3. G-Weighted Bipartite Graph Assignment Problems in Private Class Scheduling

Before proceeding to the step by step process, it is necessary to describe how the maximum matching procedure is carried out on the constructed bipartite graph. One the graph has been formed based on tutor and subject preferences, the matching process begins by identifying initial labels, evaluating possible connections, and updating them systematically to obtain an optimal pairing. The following steps outline the matching process used in this study based on the established algorithmic framework.

1. The first step is the labeling of the l-node.

	y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8	y_9	y_{10}	y_{11}
x_1	10	5	4	3	10	10	10	4	4	4	4
x_2	5	10	3	3	5	3	3	3	2	3	4
x_3	10	4	4	4	10	10	10	4	3	3	5
x_4	4	3	3	3	3	5	4	10	10	10	5
x_5	4	3	10	3	4	4	3	3	3	3	3
x_6	3	3	4	3	3	2	4	10	10	5	10
x_7	4	3	4	10	3	3	3	3	3	3	3
x_8	4	3	3	10	3	3	3	4	3	3	4

1a. $\forall v \in V_1$, example $l = max_y \in V_2(v, y)$. So that, obtained value $l(x)$ = (10, 10, 10, 10, 10, 10, 10, 10)

1b. $\forall v \in V_2$, example l = 0. So that, the labeling value $l(y)$ = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0)

1c. H_l is the opposite subgraph of G_l with the side set of E_l.

1d. For example G_l is the basic graph of H_l.

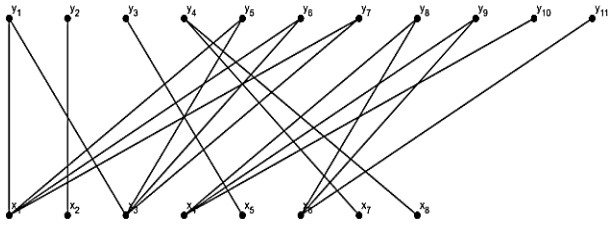


Figure 4. ***Equality*** **Subgraf di** G_l

2. Determine any matching pair M on G_l. Suppose you choose M_0 matching in the form of $(x_1, y_5), (x_2, y_2), (x_3, y_1), (x_4, y_{10}), (x_5, y_3), (x_6, y_9), (x_8, y_4)$.

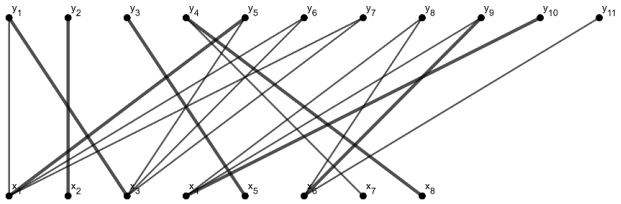


Figure 5. **Graf** ***Matching*** $\mathbf{M_0}$ **di** $\mathbf{G_l}$

3. If the maximum matching pair obtained from G_l is not perfect, so it is necessary to arrange an alternating tree T that starts from the unsaturated node to assign a new labeling to the node l'.

   3a. If a node is found V_1 which is an unsaturated node in M, Then the process continues.

   3b. For example, there is a node x from V_1 that are not unsaturated and when found x_7, then x_7 is an unsaturated node.

   3c. Build an alternating tree from M that starts from the node x_7. The result is an alternating tree T with $V(T)$ = { x_7, y_4, x_8}. Since no augmenting trajectory was found, the alternating tree T from M that comes from x_7 does not need to be expanded further in G_l. Therefore, node labeling l changed to new labeling l'.

4. Calculate the labeling of new nodes l'. $m_l = minimal\{l(x) + l(y) - w(x, y) \mid x \in V_1 \cap V(T)\ dan\ y \in V_2 - V(T)\}$ until it takes place l'.

   4a. $l(x) - m_l$ for $x \in V_1 \cap V(T)$

   4b. $l(x) - m_l$ for $y \in V_2 \cap V(T)$

   4c. $l(x)$ for others

Members of the assembly for $x \in V_1 \cap V(T)$ is $\{x_7, x_8\}$ and members of the assembly $y \in V_2 \cap V(T)$ is $\{y_1, y_2, y_3, y_4, y_5, y_6, y_7, y_8, y_9, y_{10}, y_{11}\}$. Where $S = V_1 \cap V(T)$ for the calculation of the value $m_l = min_{x \in S, y \notin T} l(x) + l(y) - w(x, y)$ are as follows.

$m_l = l(x_7) + l(y_1) - w(x_7, y_1) = 10 + 0 - 4 = 6$
$m_l = l(x_7) + l(y_2) - w(x_7, y_2) = 10 + 0 - 3 = 7$
$m_l = l(x_7) + l(y_3) - w(x_7, y_3) = 10 + 0 - 4 = 6$
$m_l = l(x_7) + l(y_5) - w(x_7, y_5) = 10 + 0 - 3 = 7$
$m_l = l(x_7) + l(y_6) - w(x_7, y_6) = 10 + 0 - 3 = 7$
$m_l = l(x_7) + l(y_7) - w(x_7, y_7) = 10 + 0 - 3 = 7$
$m_l = l(x_7) + l(y_8) - w(x_7, y_8) = 10 + 0 - 3 = 7$
$m_l = l(x_7) + l(y_9) - w(x_7, y_9) = 10 + 0 - 3 = 7$
$m_l = l(x_7) + l(y_{10}) - w(x_7, y_{10}) = 10 + 0 - 3 = 7$
$m_l = l(x_7) + l(y_{11}) - w(x_7, y_{11}) = 10 + 0 - 3 = 7$
$m_l = l(x_8) + l(y_1) - w(x_8, y_1) = 10 + 0 - 4 = 6$
$m_l = l(x_8) + l(y_2) - w(x_8, y_2) = 10 + 0 - 3 = 7$
$m_l = l(x_8) + l(y_3) - w(x_8, y_3) = 10 + 0 - 3 = 7$
$m_l = l(x_8) + l(y_5) - w(x_8, y_5) = 10 + 0 - 3 = 7$
$m_l = l(x_8) + l(y_6) - w(x_8, y_6) = 10 + 0 - 3 = 7$
$m_l = l(x_8) + l(y_7) - w(x_8, y_7) = 10 + 0 - 3 = 7$
$m_l = l(x_8) + l(y_8) - w(x_8, y_8) = 10 + 0 - 4 = 6$
$m_l = l(x_8) + l(y_9) - w(x_8, y_9) = 10 + 0 - 3 = 7$
$m_l = l(x_8) + l(y_{10}) - w(x_8, y_{10}) = 10 + 0 - 3 = 7$
$m_l = l(x_8) + l(y_{11}) - w(x_8, y_{11}) = \mathbf{10 + 0 - 4 = 6}$

Thus, a value is obtained from $m_l = 6$ from edge (x_8, y_{11}). Labeling value l' are as follows.

$l(x) - m_l = \{16, 10, 10, 10, 10, 10, 10, 10\}\ untuk\ x \in V_1 \cap V(T)$

$l'\ \{\ l(y) + m_l = \{0, 6, 0, 0, 0, 0, 0, 0, 0, 0, 0\}\ untuk\ y \in V_2 \cap V(T)$

$l(x)\ for\ the\ others$

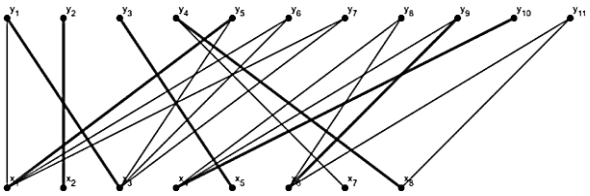


Figure 6. Matching M_0 dan Edge m_l

Because in the labeling of nodes l' Haven't loaded nodes yet l Then go back to step 3c.

3c. Arrange an alternating tree from M which began in x_7. An alternating tree is obtained T where $V(T) = \{ x_7, y_4, x_8, y_{11}\}$, can be found an augmenting-M trajectory i.e. $P_0 = \{(x_7, y_4), (x_8, y_4), (x_8, y_{11})\}$. Thus, the trajectory is augmenting P_0 can be used in the formation of new matching.

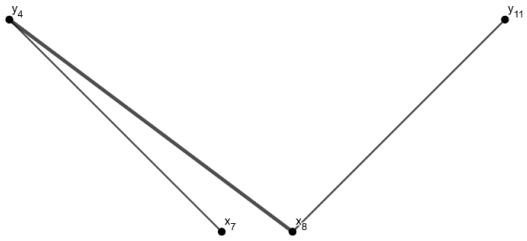


Figure 7. Augmenting-*M* Track

$M_1 = M_0 \otimes P_0$

$M_1 = \{(x_1, y_5), (x_2, y_2), (x_3, y_1), (x_4, y_{10}), (x_5, y_3), (x_6, y_9), (x_8, y_4)\}$

$\otimes \{(x_7, y_4), (x_8, y_4), (x_8, y_{11})\}$

$M_1 = \{(x_1, y_5), (x_2, y_2), (x_3, y_1), (x_4, y_{10}), (x_5, y_3), (x_6, y_9), (x_8, y_4)\}$

$\otimes \{(x_7, y_4), (x_8, y_4), (x_8, y_{11})\}$

$M_1 = \{(x_1, y_5), (x_2, y_2), (x_3, y_1), (x_4, y_{10}), (x_5, y_3), (x_6, y_9), (x_7, y_4), (x_8, y_{11})\}$

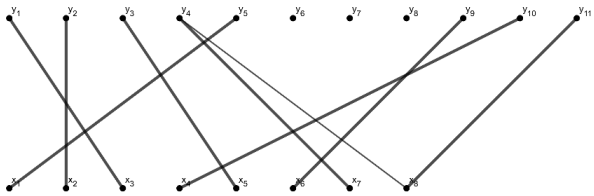


Figure 8. Maximum Matching M_1

If all vertices on the bundle V_1 has become a saturated node, then the matching search process is complete. Thus, the result of the matching graph is the final solution to the assignment problem that is being solved, and this condition is referred to as perfect matching.

$Z = w(x_1, y_5) + w(x_2, y_2) + w(x_3, y_1) + w(x_4, y_{10}) + w(x_5, y_3) + w(x_6, y_9)$

$+ w(x_7, y_4) + w(x_8, y_{11})$

$Z = 10 + 10 + 10 + 10 + 10 + 10 + 10 + 4$

$Z = 74$

With the placement of 8 tutors for each subject on the assignment problem is as follows.

Tutor x_1 teaching on subjects y_5

Tutor x_2 teaching on subjects y_2

Tutor x_3 teaching on subjects y_1

Tutor x_4 teaching on subjects y_{10}

Tutor x_5 teaching on subjects y_3

Tutor x_6 teaching on subjects y_9

Tutor x_7 teaching on subjects y_4

Tutor x_8 teaching on subjects y_{11}

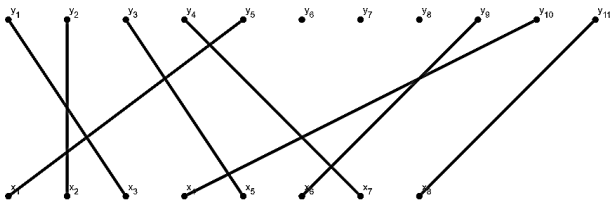


Figure 9. Tutor Placement Solutions for Every Subject

Based on the work carried out using the maximum matching algorithm on this bipartite graph, the optimal result was obtained, namely a total preference value of 74. Of the 8 tutors and 11 subjects have been adjusted, exactly one assignment per tutor in each subject. These results show that the matching of assignments in private class

scheduling is in the form of the total side weight between the tutor set and the most optimal subject set.

3.5. Data Processing with Hungarian Algorithms

In the data on assignment problems for scheduling PINTARA private classes shown in **Table 1**, there are 8 tutors and 11 subjects, so they must be added with a dummy variable.

Table 2. Tutor Preference Data for Each Subject with Dummy

T \ S	MAT	FIS	KIM	BIO	PK	PM	PU	PBM	PPU	LBI	LBE
A	10	5	4	3	10	10	10	4	4	4	4
B	5	10	3	3	5	3	3	3	2	3	4
C	10	4	4	4	10	10	10	4	3	3	5
D	4	3	3	3	3	5	4	10	10	10	5
E	4	3	10	3	4	4	3	3	3	3	3
F	3	3	4	3	3	2	4	10	10	5	10
G	4	3	4	10	3	3	3	3	3	3	3
H	4	3	3	10	3	3	3	4	3	3	4
Dummy1	0	0	0	0	0	0	0	0	0	0	0
Dummy2	0	0	0	0	0	0	0	0	0	0	0
Dummy3	0	0	0	0	0	0	0	0	0	0	0

Source: PT. Inspirasi Mandiri Nusantara

The steps in solving the Hungarian method assignment problem are as follows.

1. The initial stage involves identifying and simplifying the problem into an assignment matrix format.

$$\begin{bmatrix} 10 & 5 & 4 & 3 & 10 & 10 & 10 & 4 & 4 & 4 & 4 \\ 5 & 10 & 3 & 3 & 5 & 3 & 3 & 3 & 2 & 3 & 4 \\ 10 & 4 & 4 & 4 & 10 & 10 & 10 & 4 & 3 & 3 & 5 \\ 4 & 3 & 3 & 3 & 3 & 5 & 4 & 10 & 10 & 10 & 5 \\ 4 & 3 & 10 & 3 & 4 & 4 & 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 4 & 3 & 3 & 2 & 4 & 10 & 10 & 5 & 10 \\ 4 & 3 & 4 & 10 & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 3 & 3 & 10 & 3 & 3 & 3 & 4 & 3 & 3 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

2. Each row in the matrix is analyzed for the highest value, and then that value is subtracted from all elements in the same row.

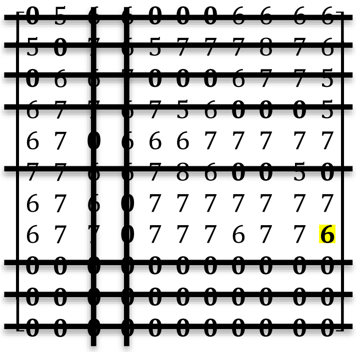
$$\begin{bmatrix} 10 & 5 & 4 & 3 & \mathbf{10} & 10 & 10 & 4 & 4 & 4 & 4 \\ 5 & \mathbf{10} & 3 & 3 & 5 & 3 & 3 & 3 & 2 & 3 & 4 \\ \mathbf{10} & 4 & 4 & 4 & 10 & 10 & 10 & 4 & 3 & 3 & 5 \\ 4 & 3 & 3 & 3 & 3 & 5 & 4 & 10 & 10 & \mathbf{10} & 5 \\ 4 & 3 & \mathbf{10} & 3 & 4 & 4 & 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 4 & 3 & 3 & 2 & 4 & 10 & \mathbf{10} & 5 & 10 \\ 4 & 3 & 4 & \mathbf{10} & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 3 & 3 & \mathbf{10} & 3 & 3 & 3 & 4 & 3 & 3 & 4 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

3. Confirm that each column has a value of 0.

$$
\begin{bmatrix}
\mathbf{0} & 5 & 6 & 6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 6 & 6 & 6 & 6 \\
5 & \mathbf{0} & 7 & 6 & 5 & 7 & 7 & 7 & 8 & 7 & 6 \\
\mathbf{0} & 6 & 6 & 7 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 6 & 7 & 7 & 5 \\
6 & 7 & 7 & 6 & 7 & 5 & 6 & \mathbf{0} & \mathbf{0} & \mathbf{0} & 5 \\
6 & 7 & \mathbf{0} & 6 & 6 & 6 & 7 & 7 & 7 & 7 & 7 \\
7 & 7 & 6 & 6 & 7 & 8 & 6 & \mathbf{0} & \mathbf{0} & 5 & \mathbf{0} \\
6 & 7 & 6 & \mathbf{0} & 7 & 7 & 7 & 7 & 7 & 7 & 7 \\
6 & 7 & 7 & \mathbf{0} & 7 & 7 & 7 & 6 & 7 & 7 & 6 \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\
\mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0}
\end{bmatrix}
$$

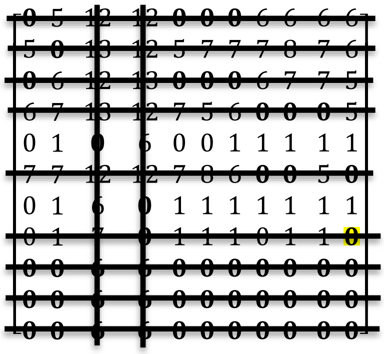
Since the value on the matrix already has a value of 0, it will be drawn.

4. Pulling After the value transformation, a line is drawn through a row or column containing element 0 by selecting the one with the most 0.



If the number of lines does not include all rows and columns, then the process is continued to the next stage.

5. The smallest value of the element that has not been crossed out is used to subtract all elements that have not been crossed out and added to the element at the point where the line intersects, the smallest value is found at the number 6.



Therefore, the number of lines and the number of rows and columns in the matrix have been equal.

6. When the number of lines is equal to the number of rows/columns, then the process is completed and the result of the matrix is considered optimal.

$$\begin{bmatrix} 0 & 5 & 12 & 12 & \mathbf{0} & 0 & 0 & 6 & 6 & 6 & 6 \\ 5 & \mathbf{0} & 13 & 12 & 5 & 7 & 7 & 7 & 8 & 7 & 6 \\ \mathbf{0} & 6 & 12 & 13 & 0 & 0 & 0 & 6 & 7 & 7 & 5 \\ 6 & 7 & 13 & 12 & 7 & 5 & 6 & 0 & \mathbf{0} & 0 & 5 \\ 0 & 1 & \mathbf{0} & 6 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\ 7 & 7 & 12 & 12 & 7 & 8 & 6 & 0 & \mathbf{0} & 5 & 0 \\ 0 & 1 & 6 & \mathbf{0} & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 7 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{6} & \mathbf{6} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{6} & \mathbf{6} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{6} & \mathbf{6} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \end{bmatrix}$$

Therefore, the optimal value is obtained from the bolded 0 element on the final matrix, where rows and columns are paired. Based on the values that have been determined on the row and column pairs, the total value is obtained to optimally solve the assignment problem.

$$\begin{bmatrix} 10 & 5 & 4 & 3 & \mathbf{10} & 10 & 10 & 4 & 4 & 4 & 4 \\ 5 & \mathbf{10} & 3 & 3 & 5 & 3 & 3 & 3 & 2 & 3 & 4 \\ \mathbf{10} & 4 & 4 & 4 & 10 & 10 & 10 & 4 & 3 & 3 & 5 \\ 4 & 3 & 3 & 3 & 3 & 5 & 4 & 10 & 10 & \mathbf{10} & 5 \\ 4 & 3 & \mathbf{10} & 3 & 4 & 4 & 3 & 3 & 3 & 3 & 3 \\ 3 & 3 & 4 & 3 & 3 & 2 & 4 & 10 & \mathbf{10} & 5 & 10 \\ 4 & 3 & 4 & \mathbf{10} & 3 & 3 & 3 & 3 & 3 & 3 & 3 \\ 4 & 3 & 3 & 10 & 3 & 3 & 3 & 4 & 3 & 3 & \mathbf{4} \\ 0 & 0 & 0 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mathbf{0} & 0 & 0 & 0 \end{bmatrix}$$

From the matrix, it is obtained that the maximum optimal value is

$Z = w(x_1, y_5) + w(x_2, y_2) + w(x_3, y_1) + w(x_4, y_{10}) + w(x_5, y_3) + w(x_6, y_9) + w(x_7, y_4) + w(x_8, y_{11}) + w(x_9, y_6) + w(x_{10}, y_7) + (x_{11}, y_8)$

$Z = 10 + 10 + 10 + 10 + 10 + 10 + 10 + 4 + 0 + 0 + 0$

$Z = 74$

Thus, the value on the assignment issue for the scheduling of PINTARA private classes has a maximum optimal of 74. The following are the results of tutor placement for each subject.

Table 3. Optimal Solution Allocation with Hungarian Algorithm

Tutors	Subjects	Preferences
x_1	y_5	10
x_2	y_2	10
x_3	y_1	10
x_4	y_{10}	10
x_5	y_3	10
x_6	y_9	10
x_7	y_4	10
x_8	y_{11}	4
$Dummy1$	y_6	0
$Dummy2$	y_7	0
$Dummy3$	y_8	0

With the placement of 8 tutors for each subject on the assignment problem is as follows.

Tutor x_1 teaching on subjects y_5

Tutor x_2 teaching on subjects y_2

Tutor x_3 teaching on subjects y_1

Tutor x_4 teaching on subjects y_{10}

Tutor x_5 teaching on subjects y_3

Tutor x_6 teaching on subjects y_9

Tutor x_7 teaching on subjects y_4

Tutor x_8 teaching on subjects y_{11}

For *Dummy* variables, it can be allocated to subject variables y6, y7, y8.

Based on the work carried out using the Hungarian algorithm, the optimal result was obtained, namely a total preference value of 74. These results show that the matching of tutor assignments and subjects in private class scheduling is in the form of the total entry values of columns and rows in the most optimal matrix.

3.5. Description of Calculations with Python Tools

Based on the optimization results of the manual calculation of the maximum matching algorithm for bipartite graphs and Hungarian algorithms, then the calculation is carried out using the Python tool. The Python program used is created by utilizing the networkx and pandas libraries, and refers to a bipartite graph structure between the tutor and subject sets. Each relationship defined in the preferences table is represented as an edge in a bipartite graph. Here are the calculation steps with the Python tool.

1. Import the required libraries

```
import networkx as nx
from networkx import bipartite
import matplotlib.pyplot as plt
import pandas as pd
```

In this calculation process, a Python program with several libraries such as networkx is used to form and analyze a bipartite graph that presents the relationship between the tutor and the subject. The bipartite library is used to find the maximum matching between nodes, while matplotlib.pyplot helps to visualize the graph graphically. The tutor's preference data is also compiled in the form of a matrix using pandas, making it easier to read each couple's score.

2. Define tutor and subject sets

```
tutors = [f"x{i}" for i in range(1, 9)] # x1 to x8
mapel = [f"y{i}" for i in range(1, 12)] # y1 to y11
```

Furthermore, the program defines two main sets, namely tutors and maple, which represent a list of x1 to x8 tutors, respectively and subjects y1 to y11. This set forms the basis for the formation of nodes on bipartite graphs.

3. Build tutor and subject pairs

```
edges = [
    ("x1", "y5"), ("x2", "y2"), ("x3", "y1"), ("x4", "y10"),
    ("x5", "y3"), ("x6", "y9"), ("x7", "y4"), ("x8", "y11")
]
```

4. Save preference scores

```
skor_preferensi={       ("x1",
     "y5"): 10,
     ("x2", "y2"): 10,
     ("x3", "y1"): 10,
     ("x4", "y10"): 10,
     ("x5", "y3"): 10,
     ("x6", "y9"): 10,
     ("x7", "y4"): 10,
     ("x8", "y11"): 4
  }
```

5. Build a preference matrix

```
def build_matrix(tutors, mapel, edges, scores):
    df = pd. DataFrame(0, index=tutors, columns=mapel)
    for tutor, pelajaran in edges:
         df.loc[tutor, pelajaran] = scores.get((tutor, pelajaran), 0)
    return df
```

6. Create and visualize bipartite graphs

```
def build_graph():
    G =
    nx.
    Graph
    ()
    G.add_nodes_from(tutors, bipartite=0)
    G.add_nodes_from(mapel, bipartite=1)
    G.add_edges_from(edges)
    return G

def plotGraph(graph):
    pos = {}
    pos.update((n, (1, i)) for i, n in enumerate([n for n, d in
graph.nodes(data=True) if d["bipartite"] == 0]))
    pos.update((n, (2, i)) for i, n in enumerate([n for n, d in
graph.nodes(data=True) if d["bipartite"] == 1]))
    nx.draw(graph,            pos,      with_labels=True, node_size=1000,
node_color="skyblue")
    plt.title("Graf Bipartit Tutor – Mata Pelajaran") plt.axis("off")
    plt.show()
```

7. Running the *maximum* matching algorithm

```
def enumMatchingUnoStyle(graph):
    top = [n for n, d in graph.nodes(data=True) if d['bipartite'] == 0]
    matching = bipartite.hopcroft_karp_matching(graph, top_nodes=top)
    match_clean = [(u, v) for u, v in matching.items() if u in top] return
    match_clean
```

8. Running a major program

```
If name == " main ":
    # Show preference matrix
    matrix   =   build_matrix(tutors,   maps,   edges,   skor_preferensi)
       print("Preference Matrix (Score):\n")
    print(matrix)

    # Create a graph and display
       G     =     build_graph()
       plotGraph(G)

    # Matching
    matches = enumMatchingUnoStyle(G)
    print("\nMaximum matching (Tutor → Subject):")
    for tutor, pelajaran in matches:
        print(f"{tutor} → {lesson}")

    # Total preference score of the found match
    total_skor = sum(skor_preferensi.get((tutor, pelajaran), 0) for tutor,
    pelajaran in matches)
    print(f"\nTotal Preference Score (Optimal): {total_skor}")
```

9. Output Results

Based on the *output results* of the Python program tool, a list of tutor pairs and subjects is obtained that represents the maximum adjustment results without clashing. Each tutor successfully paired exactly one subject according to their preferences, resulting in 8 unique pairs that do not overlap with each other. Here is a preference matrix view that shows the connectivity score between the tutor and the subject.

Preference Matrix (Score):

```
     y1  y2  y3  y4  y5  y6  y7  y8  y9  y10  y11
x1    0   0   0   0  10   0   0   0   0    0    0
x2    0  10   0   0   0   0   0   0   0    0    0
x3   10   0   0   0   0   0   0   0   0    0    0
x4    0   0   0   0   0   0   0   0   0   10    0
x5    0   0  10   0   0   0   0   0   0    0    0
x6    0   0   0   0   0   0   0   0  10    0    0
x7    0   0   0  10   0   0   0   0   0    0    0
x8    0   0   0   0   0   0   0   0   0    0    4
```

Of the eight pairs formed, each had a predetermined preference score based on the level of suitability or priority of the assignment. This score is then summed to calculate the total preference value of all the adjustments that are formed. The final result of this process shows that the total value achieved is 74 which means that the resulting adjustments are the optimal solution based on the preference parameters used.

Maximum matching (Tutors → Subjects):

```
x7 → y4
x6 → y9
x5 → y3
x1 → y5
x3 → y1
x8 → y11
x4 → y10
x2 → y2
Total Preference Score (Optimal): 74
```

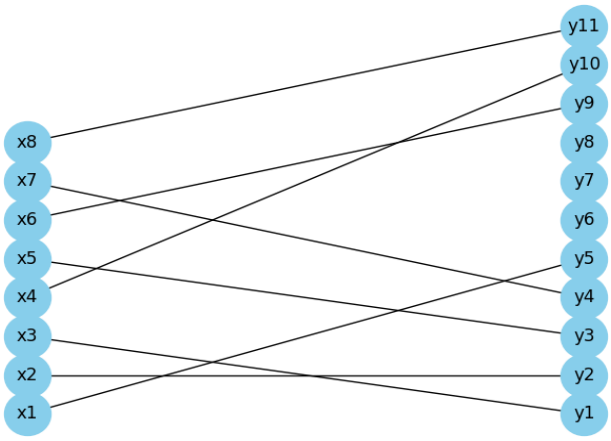


Figure 10. Maximum Matching Graph Output

Based on the results of the optimal assignment visualized on the graph, all tutors were successfully paired efficiently with one subject according to their preferences. Tutor A is assigned to teach quantitative knowledge (PK) subjects, tutor B teaches physics, tutor C teaches mathematics, tutor D teaches Indonesian literacy (LBI), tutor E teaches chemistry, tutor F teaches general knowledge and understanding, tutor G teaches biology, and tutor H teaches English literacy (LBE). Each pair reflects a non-overlapping allocation and according to the expertise of each tutor. The total value of the resulting preference score of 74 indicates that this match is the optimal solution based on the preference data used. This proves that the maximum matching algorithm model on the bipartition graph, the Hungarian algorithm, and using the Python program tool produces the matching results of tutor assignments and subjects of equal and optimal value. The findings align with previous studies that demonstrated the effectiveness of assignment problem based optimization models, including bipartite graph matching and the Hungarian method, in producing optimal and conflict free [20]. Other related works also indicated that assignment formulations and time minimization techniques consistently improve efficiency in practical allocation problems across various operational settings [2]. The consistency between the present results and those reported in earlier research strengthens the evidence that assignment models, whether solved through bipartite matching techniques, the Hungarian algorithm, or computational tools provide reliable solutions for one-to-one allocation problems such as tutors to subjects matching.

4. CONCLUSION

This study succeeded in identifying the problem of scheduling private classes at PT. Inspirasi Mandiri Nusantara (PINTARA) as an assignment problem. Through direct observation and literature studies, mathematical modeling was carried out with a maximum *matching* algorithm on the relationship between tutor sets and subject sets. Troubleshooting is done with the maximum matching algorithm, the Hungarian algorithm, and the Python program tool. The results show that all tutors can be optimally paired with a single subject. These results show that using three models of maximum matching algorithms, the Hungarian algorithm, and the Python program tool results in a matching of tutor and subject assignments of equal and optimal value. The applied mathematical and computational approaches are effective in minimizing schedule conflicts, optimizing tutor allocation, and improving the accuracy and efficiency of

scheduling modeling. Based on the results obtained, the assignment method in class scheduling can be used as an alternative to implement the completion of other assignments, because it has been proven to be able to produce optimal solutions. The research can also be expanded by including additional variables such as tutor availability time, learning location, and student preferences, and lead to large-scale scheduling automation using real-time Python program tools.

Acknowledgments

The authors express their heartfelt appreciation to **PT. Inspirasi Mandiri Nusantara (PINTARA)**, with special thanks to **Mr. Saiful (HRGA)** for their generous support and cooperation during the research. Special thanks are extended to the academic and administrative staff for providing valuable information and granting access to the required data. This acknowledgment is provided with their full consent.

Funding Information

The authors declare that no external funding was received for this research. All activities related to this study were fully supported and financed by the authors themselves.

Author Contributions Statement

Firts Author (Hanifah Felisia Wati): Conceptualization, methodology, software, formal analysis, investigation, data curation, writing–original draft, writing–review & editing, visualization, and project administration. Second Author (Sapti Wahyuningsih): Methodology, validation, resources, writing–review & editing, and supervision. Third Author (Muhammad Nur Ramdhan): Validation, resources, writing-review & editing, and supervision. All authors discussed the results and contributed to the final manuscript.

Conflict Of Interest Statement

In accordance with ethical standards and in support of transparency in academic publishing, we are the authors of the manuscript entitled "Optimization of Assignment Problems in Private Class Scheduling Using Graph Application", namely Hanifah Felisia Wati, Sapti Wahyuningsih, and Muhammad Nur Ramadhan from Universitas Negeri Malang.

We here by declare that we have **no affiliations or engagements** with any organization or entity that could present a financial interest in this work (such as honoraria, educational or research grants, speaker bureau involvement, employment, consultancies, stock or equity ownership, expert testimony, or patent-related arrangements). We also confirm that we have **no non-financial interests** such as personal, professional, ideological, academic, or institutional relationships that could influence the content or interpretation of the material presented in this manuscript. This statement is submitted accurately and with full acknowledgment of its implications.

Informed Consent

This study does not involve any personal, medical, or identifiable human data. All information used in the analysis consists solely of institutional, non-sensitive scheduling and assignment data provided by PT. Inspirasi Mandiri Nusantara (PINTARA). Access to these data was granted with the institution's permission for research purposes. Therefore, individual informed consent is not required or applicable to this study.

Ethical Approval

This study does not involve human participants, personal data, medical information, or animal subjects. All data used in this research consist of non-sensitive institutional scheduling information provided by PT. Inspirasi Mandiri Nusantara (PINTARA) for research purposes.

Therefore, ethical approval from an institutional review board or ethics committee is not required and does not apply to this study.

Data Availability

The data that support the findings of this study are held by the first author, Hanifah Felisia Wati, and may be obtained upon reasonable request. The dataset was provided by PT. Inspirasi Mandiri Nusantara (PINTARA) and is subject to institutional restrictions. As the data consist of internal scheduling and assignment information, they cannot be made publicly available. Access to the data may be granted upon request and with approval from the data-providing institution.

REFERENCES

[1] A. Y. Setiani, A. H. Mujianto, A. Andriani, and S. Widoyoningrum, "Sistem Penjadwalan Bimbingan Belajar Berbasis Web Menggunakan Algoritma Genetika," *Inovate: Jurnal Inovasi Teknologi Informasi*, vol. 9, no. 1, pp. 234–243, 2024.

[2] L. N. Rahman and W. Wahyudin, "Optimalisasi Penugasan Karyawan Jasa Ekspedisi Menggunakan Metode Hungarian (Studi Kasus CV. Anteraja Cabang Mekarmukti)," *JSE*, vol. 6, no. 3, pp. 2120–2127, 2021.

[3] H. Fu *et al.*, "Modeling the Residual Queue and Queue-dependent Capacity in a Static Traffic Assignment Problem," *Transportation Research Part B: Methodological*, vol. 192, p. 103158, 2025.

[4] S. Koehler and F. Theilacker, "The Backroom Assignment Problem for In-store Order Fulfillment," *Computers & Industrial Engineering*, vol. 202, p. 110912, 2025.

[5] R. Aryana and N. S. Sangkala, "Implementation of Set Theory in Developing the Definition of Graph Theory," *Integral: Journal of Mathematics Education and Learning*, vol. 2, no. 1, pp. 12–16, 2023.

[6] M. Khabbazian and M. Médard, "On Finding the Largest Minimum Distance of Locally Recoverable Codes: A Graph Theory Approach," *Discrete Mathematics*, vol. 348, no. 2, p. 114298, 2025.

[7] A. Davoodi *et al.*, "Response Prediction of Antidepressants: Using Graph Theory Tools for Brain Network Connectivity Analysis," *Biomedical Signal Processing and Control*, vol. 103, p. 107362, 2025.

[8] P. Manurangsi, E. Segal-Halevi, and W. Suksompong, "On Maximum Bipartite Matching with Separation," *Information Processing Letters*, vol. 182, p. 106388, 2023.

[9] J. Stipancic, N. Saunier, N. Navidi, E. B. Racine, L. Miranda-Moreno, and A. Labbe, "Evaluating Map Matching Algorithms for Smartphone GNSS Data: Matching Vehicle Trajectories to an Urban Road Network," *Transportation Research Procedia*, vol. 82, pp. 303–322, 2025.

[10] M. Chen, H. Zheng, H. Yang, W. Ding, and D. Zhang, "Combining Spot Image Denoising Network and Hungarian Matching Algorithm: Achieving High-Precision Measurement of Aspherical Morphology," *Optics and Laser Technology*, vol. 188, p. 112970, 2025.

[11] A. Ma'arif, Buku Ajar Pemrograman Lanjut: Bahasa Pemrograman Python, Yogyakarta: Universitas Ahmad Dahlan, 2020.

[12] F. S. Hillier and G. J. Lieberman, *Introduction to Operations Research*, Edition 9. New York: McGraw-Hill, Inc, 2008.

[13] A. B. Pratiwi and Triyani, "Matching Maksimum pada Graf Petersen Berarah Berdasarkan Multiplisitas Geometri Terbesar," *Jurnal Matematika Integratif*, vol. 19, no. 1, pp. 1–12, 2023.

[14] V. Mkrtchyan, "Three Results Towards Approximation of Special Maximum Matchings in Graphs," *Discrete Applied Mathematics*, vol. 371, pp. 127–136, Aug. 2025, doi: 10.1016/j.dam.2025.04.005.

[15] W. Kocay and D. L. Kreher, *Graphs, Algorithms, and Optimization (Discrete mathematics and its applications)*. New York Washington: CRC Press, 2005.

[16] J. A. Bondy and U. S. R. Murty, Graph Theory with Applications, New York: Mac Millan Press, 1976.

[17] G. Chartrand and U. S. R. Murty, Applied and Algoritmic Graph Theory, New York: Mc Graw Hill International Edition, 1993.
[18] M. Gondran and M. Minoux, Linear Algebra in Dioids: A Survey of Recent Results in Algebraic and Combinatorial Methods in Operations Research, Amsterdam: North-Holland Math, 1984.
[19] F. S. Hillier and G. J. Lieberman, Introduction to Operations Research, USA: Mcgraw-Hill, 1990.
[20]F. H. Sani and S. Sawaluddin, "Analisis Penyelesaian Masalah Penugasan Pada Algoritma Matching Graf Bipartit Dan Metode Hungarian," F.Jmpm, vol. 6, no. 1, pp. 89–96, 2023.
[21] Y. I. Runtunuwu, M. Mananohas, and C. Montolalu, "Derajat Laplacian dari Graf Lengkap, Graf Bipartisi Komplit, Graf Matahari dan Graf yang memiliki n - 1 Derajat berbeda," vol. 10, no. 1, 2021.