International Journal of Social Science and Human Research

ISSN (print): 2644-0679, ISSN (online): 2644-0695

Volume 08 Issue 07 July 2025

DOI: 10.47191/ijsshr/v8-i7-54, Impact factor- 8.007

Page No: 5359-5371

Application of Kuhn-Munkres Algorithm to Multi-Objective Assignment Problems

Ni Luh Gede Izta Mawarni[1], Ni Ketut Tari Tastrawati[2], Kartika Sari[3]

[1,2,3]Departement of Mathematics, Udayana University, Bali, Indonesia

ABSTRACT: The assignment problem is a problem concerning the arrangement of a group of individuals in carrying out tasks with the aim of determining the right tasks so that the total use of resources can be minimized. Assignment problems that have several resources that must be optimized are called multi-objective assignment problems. The purpose of this study is to determine how to solve multi-objective assignment problems related to costs, completion time, and the number of defective products in the ABC confection business using the Kuhn-Munkres Algorithm with a graph theory approach. It is obtained that a daily wage savings of 3.47%, savings in the completion of the product 1.88%, and the reduced defect product produced by 33.33%.

KEYWORDS: Assignment Problem; Kuhn-Munkres Algorithm; Multi-Objective

I. INTRODUCTION

The rapid development of the era requires humans to be ready to anticipate all changes in various fields, one of which is the field of production business, both production of goods and services. In the business world, in addition to marketing strategies and good product quality, strategies in managing production aspects are also important. One aspect of production in question is the placement of workers properly or known as the problem of assignment.

Assignment problems are a special type of linear programming problems. According to Ristono & Puryani (2011), assignment problems are problems regarding the arrangement of a group of individuals to carry out tasks so that the use of resources to complete all tasks can be minimized. In general, assignment problems are where there are n tasks assigned to m workers with different competencies possessed by each worker.

Several studies have been conducted to solve the assignment problem, but most of them only consider one optimization objective. For example, the assignment problem by minimizing production costs. This kind of problem is called a single-objective assignment problem. However, in the assignment problems found in the field, it is not enough to optimize only one resource. This is because in managing production aspects there are several resources that must be optimized in order to obtain optimal assignments, such as minimizing costs, time, and the number of defective products simultaneously. Assignment problems that have several optimization objectives for several types of resources owned by workers in completing tasks are called multi-objective assignment problems (D.J. White, 1984)

The assignment problem can be solved using a graph theory approach. In the assignment problem, each worker can complete all tasks. Therefore, the form of the assignment problem if represented in the form of a graph will form a graph with two subsets of points, namely the set of points consisting of workers and the set of points consisting of tasks, where each point of each set is connected to each other. In graph theory, a graph whose set of points can be grouped into two subsets, for example V_1 and V_2 where each point in V_1 is connected to all points in V_2 is called a complete bipartition graph or complete bipartition graph (Munir, 2010). Because one worker and another have different competencies in completing tasks, the assignment problem can be expressed as a complete weighted bipartition graph where the weight in question is the competency possessed by each worker. Then the solution to the assignment problem is sought by determining the perfect matching with the maximum weight

A perfect matching on a complete weighted bipartition graph is a matching that has the maximum number of weights. In finding a perfect matching with maximum weight, you can use the Kuhn-Munkres Algorithm developed by Kuhn (1955) and Munkres (1957) (Bondy & Murty, 1976). This algorithm is also known as the Hungarian Algorithm. Before finding a solution to the assignment problem using the Kuhn-Munkres Algorithm, the multi-objective objective function is first converted into a single-objective objective function by giving weights to each objective function. Then continue by finding a solution to the assignment problem using the Kuhn-Munkres Algorithm.

Several studies have been conducted in relation to solving multi-objective assignment problems. Bao et al., (2007) provided a new approach in solving multi-objective assignment problems related to cost, time, and quality in the form of simulation data with the 0-1 programming method. Raharjo (2010) solved the optimization of multi-objective assignment problems using the Hungarian method with matrices in a guitar craft business in Ngrombo Baki Sukoharjo. The multi-objective assignment problem studied considered cost, time, and quality with a minimum total cost of Rp. 15,070,000, -, a minimum total time of 29 days, and a minimum quality score of 10. Then Utami (2018) applied the Kuhn-Munkres Algorithm with a graph theory approach in solving multi-objective assignment problems in the DP. SPORTY bag convection industry. The assignment problem studied considered cost, time, and quality with a minimum total cost of Rp. 326,000,-, minimum total time 1587 seconds, and minimum quality score 601.

The assignment problem is also experienced by ABC's clothing business. ABC is a clothing business engaged in the field of clothing production located in Mengwi District, Badung Regency, Bali, Indonesia. The products produced are collared t-shirts, t-shirts, short-sleeved shirts, long-sleeved shirts, hats, and pants. Most of the orders received by ABC are school uniform orders and uniform orders for the general public. So far, ABC has not implemented a special method in placing workers, so an analysis is needed for the right placement of workers so that costs, completion time, and the number of defective products can be minimized.

Considering that the assignment problem in the ABC confection business is a multi-objective assignment problem and the assignment problem can be solved using the Kuhn-Munkres Algorithm, the aim of this study is to solve the multi-objective assignment problem using the Kuhn-Munkres Algorithm with a graph theory approach in the ABC confection business. The graph concept used is complete bipartite graph matching.

Graph matching is defined as a subset M of a an edge set ($E(G)$) in a graph G whose edges have distinct endpoints and no two edges are adjacent (Bondy & Murty, 1976). A matching M in graph G is said to be a perfect matching if M contains all points G. In other words, if M is a perfect matching in graph G, then every point in G is covered by a matching (M-saturated). A point v in graph G is said to be closed by matching M (M-saturated) if point v is the end point or end point of one of the edges in M. If M does not close v in graph G then point v is said to be not closed by matching (M-unsaturated) (Bondy & Murty, 1976).

Let M be a matching in graph G, E a set of edges in G that are not matching. An M-alternating path is a path whose sides alternate at $E\backslash M$ and at M. An M-alternating path whose starting point and ending point are not covered by M is called an M-augmenting path (Bondy & Murty, 1976). Next, given a graph tree T with M matchings and point u is a point that is not covered by M matching. M-alternating tree is a tree rooted at point u with the characteristic that all paths starting from point u are M-alternating paths (Bondy & Murty, 1976).

Given a graph G, $S \subseteq V(G)$ and $N(S)$, where $V(G)$ is the set of all points in graph G and $N(S)$ is the set of neighbors of S in G or N(S) is the set of all points of G that are neighbors of points in S (Bondy & Murty, 1976). Hall's Theorem is a necessary and sufficient condition for matching a bipartition graph G with a partition (X,Y) that covers all points in X.

Hall's Theorem (Hall, 1935) Let G be a bipartition graph with partitions (X,Y). The graph G contains a matching that covers all vertices X if and only if for each S⊆X holds $|N(S)| \geq |S|$.

Hall's Marriage Theorem (Hall, 1935) If G is a k-regular bipartition graph with k>0, then G contains a match that covers all vertices in X.

In a particular company, suppose there are n workers denoted by $x_1, x_2, x_3, \ldots, x_n$ and there are n jobs denoted by $y_1, y_2, y_3, \ldots, y_n$. Each worker has the ability to do one or more of these jobs. Based on this, a pairing can be made between workers and existing jobs so that all workers get jobs that match their abilities. This is called the personal assignment problem (Bondy & Murty, 1976).

The assignment problem can be modeled by forming a bipartition graph G with partitions (X,Y) where by $X = \{ x_1, x_2, x_3, \ldots, x_n\}$, $Y = \{ y_1, y_2, y_3, \ldots, y_n \}$, and x_i is adjacent to y_j if and only if worker x_i can do job y_j. Next, determine whether or not there is perfect matching in graph G. According to Hall's Theorem, there is perfect matching in graph G if for every $S \subseteq X$ holds $|N(S)| \geq |S|$. Conversely, if there exists $S \subseteq X$ such that $|N(S)| < |S|$ then G does not have perfect matching. In other words, the ultimate goal of solving the personal assignment problem is to find a match of G that covers all points or if no such match is found, we will find $S \subseteq X$ such that $|N(S)| < |S|$.

In the personal assignment problem, it is modeled by forming a bipartition graph G, while in the optimal assignment problem it is limited by a complete weighted bipartition graph. Given a complete weighted bipartition graph (X,Y) with $X = \{ x_1, x_2, x_3, \ldots, x_n\}$, $Y = \{ y_1, y_2, y_3, \ldots, y_n \}$ and edge $x_i y_j$ has a weight $w_{ij} = w(x_i y_j)$ which states the level of ability of worker x_i in completing work y_j. The solution to the optimal assignment problem is to find a perfect match with maximum weight in the complete weighted bipartition graph or what is called optimal match (Bondy & Murty, 1976).

In solving the optimal assignment problem, on a graph $G = K_{n,n}$ it is possible to create $n!$ perfect matchings and find the optimal one among the other perfect matchings. In the case of large n, listing all perfect matchings on the graph $K_{n,n}$ and finding the optimal one is very inefficient. (Bondy & Murty, 1976). To find the optimal matching on a complete weighted bipartition graph, the Optimal Matching Theorem is applied.

Optimal Matching Theorem Let l be a feasible vertex labeling in a weighted complete bipartition graph G and G_l is a spanning part graph G formed by edges E_l . If the graph G_l contains a perfect match M* then M* is the optimal match in graph G (Bondy & Murty, 1976).

Optimal Matching Theorem is used as the basis for the creation of an algorithm by Kuhn (1955) and Munkres (1957) to determine a perfect matching in a complete weighted bipartition graph (Bondy & Murty, 1976). The Kuhn-Munkres algorithm was published by an American mathematician named Harold Kuhn in 1955 to honor two Hungarian mathematicians, D. Kõnig and E. Egerváry. Then in 1957, James Raymond Munkres, a Professor Emeritus of Mathematics, revised the Kuhn algorithm. This algorithm is also known as the Hungarian Algorithm (Bondy & Murty, 1976).

II. METHODS

An This study uses secondary data, namely data obtained from the ABC confection business from March to May 2021. The data used are data on the number of workers, the number of types of products, total wages, the number of products produced, completion time, and the number of defective products. The data is quantitative data in the form of data on total worker wages in rupiah, the number of products produced in units of units, completion time in units of days (1 working day is 8 hours), and the number of defective products in units of units.

This study uses the Kuhn-Munkres Algorithm in solving multi-objective assignment problems. The steps taken in this study are as follows (Bondy & Murty, 1976):

1. Collecting data on the number of workers, the number of product types, worker wages, total products produced, completion time, and the number of defective products.
2. Compiling a multi-objective assignment problem model based on equation (1).

Minimize

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

with constraints

$\sum_{i=1}^{n} x_{ij} = 1$, for every $j = 1,2, \dots, n$

$\sum_{j=1}^{n} x_{ij} = 1$, for every $i = 1,2, \dots, n$

$x_{ij} = \begin{cases} 1, & \\ 0, & \text{if task } j \text{ is not assigned to worker } i. \end{cases}$

where Z is objective function; n is the number of workers who will be given tasks and the number of tasks that must be completed; x_{ij} is an assignment from worker i to task j.

In this study, 3 objective functions are optimized simultaneously, namely the average wage of workers per day in rupiah, the average completion time per product in seconds, and the number of defective products in unit. The optimization objective is minimization.

3. Reconstructing the objective function coefficients.

Reconstruction is carried out on the objective function coefficients which aim to minimize by finding the maximum value of the data for each objective, then the maximum value is subtracted from each data value from the corresponding objective.

4. Performing data normalization.

Data normalization, namely data equalization, is carried out by dividing each objective data value to be optimized by the maximum data value of each objective.

5. Changing the multi-objective objective function to a single-objective form.

Changing the multi-objective objective function to a single-objective form is done by giving weights to each objective. It is assumed that each objective function has the same level of importance, so it is given the same weight. After being given weights, the objective function is formulated in equation (2).

Minimize (maximize):

$$Z(X) = a_1 \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}^{1} x_{ij} + a_2 \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}^{2} x_{ij} + \dots + a_p \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}^{p} x_{ij} \tag{2}$$

6. Determining the solution to the assignment problem using the Kuhn-Munkres Algorithm . The steps to obtain a perfect matching with maximum weight on a complete weighted bipartition graph with the Kuhn-Munkres Algorithm are as follows (Bondy & Murty, 1976):

Step 1: Determine the labeling of vertex l and form the spanning part graph G_l .
Determine the labeling of vertex in graph G, name the labeling l which is defined as follows:

a. For every $x_i \in X$, $l(x_i) = \max \{w(x_i y_j) | y \in Y\}$, $i = 1, 2, 3, \ldots, n; j = 1, 2, 3, \ldots, n.$

b. For every $y_j \in Y$, $l(y_j) = 0$

c. Form G_l which is the spanning part graph of G with edge set E_l

$E_l = \{ x_i y_j \in E(G) | \, x_i \in X, y_j \in Y \text{ and } l(x_i) + l(y_i) = w(x_i y_j)\}$

Step 2: choose any M maching on G_l

Step 3: determine whether a perfect match has been obtained or not.

a. Determines whether a perfect match has been obtained. If all points X are covered by M, then M is a perfect match such that M matching is optimal on G_l , and the algorithm stops. If there is a point in X is not closed by matching M, then M is not a perfect match in G_l, so the algorithm continues to step 3.b
b. Determine the root point, for example point u, which is a point on X which is not covered by matching M on G_l to form M-alternating tree T.
c. Forms an M-alternating tree T from matching M to G_l which is rooted at point u. Then it is determined whether there is an M-augmenting path in the M-alternating tree T. If there is an M-augmenting path, then we obtain an extension of M along the M-augmenting path to get a new match M* by replacing the non-matching edge with a match (and vice versa) contained in the M-augmenting path. Then go back to step 3.a. If there is no M-augmenting path, then proceed to step 4, namely replacing the point l labeling with a new point labeling called l ' with the property that M and T are contained in the graph G_l.

Step 4: calculate the new vertex labeling l '.
This step uses the alternating tree T to calculate the new vertex labeling l '. First, the value of $m_\ell =$ min $\{\ell(x_i) + \ell(y_j) - w(x_i y_j) | x_i \in X \cap V(T) \text{ and } y_j \in Y - V(T)\}$ is found. After obtaining the value of m_ℓ, l ' is determined which is defined as written in equation (3)

$$\ell' = \begin{cases} \ell(x_i) - m_\ell, & \text{for } x_i \in X \cap V(T) \\ \ell(y_j) + m_\ell, & \text{for } y_j \in Y \cap V(T) \\ \ell(x_i), \ell(y_j), & \text{others} \end{cases} \qquad (3)$$

with $m_\ell > 0$

Step 5: form a spanning section graph $G_{l'}$
After it is obtained a new vertex labeling l ', then form a graph $G_{l'}$ with a set of edges $E_{l'} = \{ x_i y_j \in E(G) \,|\, x_i \in X, y_j \in Y \text{ and } l'(x_i) + l'(y_j) = w(x_i y_j)\}$, then return to step 3.b.

7. Interpretation of result

III.RESULT AND DISCUSSION

The data used in this study are secondary data obtained from the ABC confection business. Data collection was carried out from March to May 2021. The data are in the form of data on the number of workers (namely D, E, F, G, H, J) and the number of types of products as many as 6, total worker wages in rupiah, the number of products produced in units, completion time in days (1 working day is 8 hours), and the number of defective products in units during the period March-May 2021. Based on these data, the average daily wages (c) , average completion time per product (t) , and number of defective (q) products during the research period were calculated for each worker. The calculation results are presented in Table I.

Table I Data of the Average Daily Wages (c) , Average Completion Time per Poduct (t) , and Number of Defective products (q) during During the Research Period

Job / Worker		Collared Shirts	T-shirts	Short-sleeved shirts	Long-sleeved shirts	Hats	Pants
D	*c* (rupiahs)	102800	114423	83077	200000	38000	105000
	t (second)	1681	881	4160	2160	7579	1920
	q (unit)	2	2	1	1	1	2
E	*c* (rupiahs)	133412	98667	96000	159375	37500	115111
	t (second)	1295	1021	3600	2711	7680	1751

	q (unit)	2	2	1	1	1	1
F	*c* (rupiahs)	123818	93800	108000	182143	50000	103091
	t (second)	1396	1075	3200	2372	5760	1956
	q (unit)	2	2	1	1	1	2
G	*c* (rupiahs)	97600	90563	103200	168750	40000	112778
	t (second)	1770	1113	3349	2560	7200	1788
	q (unit)	2	1	1	2	1	1
H	*c* (rupiahs)	104000	95278	98182	153333	45000	124250
	t (second)	1662	1058	3520	2817	6400	1623
	q (unit)	1	2	1	1	1	1
J	*c* (rupiahs)	98143	107917	110667	168750	45000	93333
	t (second)	1761	934	3123	2560	6400	2160
	q (unit)	2	2	2	1	1	2

Based on Table I and equation (1), a multi-objective assignment problem model is formed as written in equation (2).
Minimize

$Z_1(X) = \sum_{i=1}^{6}\sum_{j=1}^{6} c_{ij}\, x_{ij}$

$= 102800x_{11} + 114423x_{12} + 83077x_{13} + 200000x_{14} + 38000x_{15} + 105000x_{16} + 133412x_{21} + 98667x_{22} + 96000x_{23} + 159375x_{24} + 37500x_{25} + 115111x_{26} + 123818x_{31} + 93800x_{32} + 108000x_{33} + 182143x_{34} + 50000x_{35} + 103091x_{36} + 97600x_{41} + 90563x_{42} + 103200x_{43} + 168750x_{44} + 40000x_{45} + 112778x_{46} + 104000x_{51} + 95278x_{52} + 98182x_{53} + 153333x_{54} + 45000x_{55} + 124250x_{56} + 98143x_{61} + 107917x_{62} + 110667x_{63} + 168750x_{64} + 45000x_{65} + 93333x_{66}$

$Z_2(X) = \sum_{i=1}^{6}\sum_{j=1}^{6} t_{ij}\, x_{ij}$

$= 1681x_{11} + 881x_{12} + 4160x_{13} + 2160x_{14} + 7579x_{15} + 1920x_{16} + 1295x_{21} + 1021x_{22} + 3600x_{23} + 2711x_{24} + 7680x_{25} + 1751x_{26} + 1396x_{31} + 1075x_{32} + 3200x_{33} + 2372x_{34} + 5760x_{35} + 1956x_{36} + 1770x_{41} + 1113x_{42} + 3349x_{43} + 2560x_{44} + 7200x_{45} + 1788x_{46} + 1662x_{51} + 1058x_{52} + 3520x_{53} + 2817x_{54} + 6400x_{55} + 1623x_{56} + 1761x_{61} + 934x_{62} + 3123x_{63} + 2560x_{64} + 6400x_{65} + 2160x_{66}$

$Z_3(X) = \sum_{i=1}^{6}\sum_{j=1}^{6} q_{ij}\, x_{ij}$

$= 2x_{11} + 2x_{12} + x_{13} + x_{14} + x_{15} + 2x_{16} + 2x_{21} + 2x_{22} + x_{23} + x_{24} + x_{25} + x_{26} + 2x_{31} + 2x_{32} + x_{33} + x_{34} + x_{35} + 2x_{36} + 2x_{41} + x_{42} + x_{43} + 2x_{44} + x_{45} + x_{46} + x_{51} + 2x_{52} + x_{53} + x_{54} + x_{55} + x_{56} + 2x_{61} + 2x_{62} + 2x_{63} + x_{64} + x_{65} + 2x_{66}$

With constrain

$\sum_{i=1}^{6} x_{ij} = 1$, for every j = 1, 2, 3, 4, 5, 6

$\sum_{j=1}^{6} x_{ij} = 1$, for every i = 1, 2, 3, 4, 5, 6 (2)

$$x_{ij} = \begin{cases} 1, \text{if task } j \text{ assigned to worker } i \\ 0, \text{if task } j \text{ did not assign to worker } i \end{cases}$$

Next, reconstruction is carried out by finding the maximum value of each data on average worker wages, average completion time, and number of defective products, then the maximum value is reduced by each value of each corresponding type of data. The reconstruction results are presented in Table II.

Table II

Table II Data on Average Daily Worker Wages, Average Completion Time Per Product, and Number of Defective Products During the Research Period After Reconstruction

Job / Worker		Collared Shirts	T-shirts	Short-sleeved shirts	Long-sleeved shirts	Hats	Pants
D	*c'* (rupiahs)	97200	85577	116923	0	162000	95000
	t' (second)	5999	6799	3520	5520	101	5760
	q' (unit)	0	0	1	1	1	0
E	*c'* (rupiahs)	66588	101333	104000	40625	162500	84889
	t' (second)	6385	6659	4080	4969	0	5929
	q' (unit)	0	0	1	1	1	1
F	*c'* (rupiahs)	76182	106200	92000	17857	150000	96909

	t' (second)	6284	6605	4480	5308	1920	5724
	q' (unit)	0	0	1	1	1	0
G	c' (rupiahs)	102400	109438	96800	31250	160000	87222
	t' (second)	5910	6567	4331	5120	480	5892
	q' (unit)	0	1	1	0	1	1
H	c' (rupiahs)	96000	104722	101818	46667	155000	75750
	t' (second)	6018	6622	4160	4863	1280	6057
	q' (unit)	1	0	1	1	1	1
J	c' (rupiahs)	101857	92083	89333	31250	155000	106667
	t' (second)	5919	6746	4557	5120	1280	5520
	q' (unit)	0	0	0	1	1	0

Because the workers' wages, completion time, and number of defective products are different units, it is necessary to carry out data normalization, namely equalizing the data by dividing each data on the average worker's wages, average completion time, and number of defective products by the maximum data value of each corresponding data type. The results of data normalization in Table II are presented in Table III.

Table III Normalization Results of Average Daily Wages, Average Completion Time Per Product, and Number of Defective Products During Research Periods

Job / Worker		Collared Shirts	T-shirts	Short-sleeved shirts	Long-sleeved shirts	Hats	Pants
D	c'' (rupiahs)	0.60	0.53	0.72	0.00	0.99	0.58
	t'' (second)	0.88	1.00	0.52	0.81	0.01	0.85
	q''(unit)	0.00	0.00	1.00	1.00	1.00	0.00
E	c'' (rupiahs)	0.41	0.62	0.64	0.25	1.00	0.52
	t'' (second)	0.94	0.98	0.60	0.73	0.00	0.87
	q''(unit)	0.00	0.00	1.00	1.00	1.00	1.00
F	c'' (rupiahs)	0.47	0.65	0.57	0.11	0.92	0.60
	t'' (second)	0.92	0.97	0.66	0.78	0.28	0.84
	q''(unit)	0.00	0.00	1.00	1.00	1.00	0.00
G	c'' (rupiahs)	0.63	0.67	0.60	0.19	0.98	0.54
	t'' (second)	0.87	0.97	0.64	0.75	0.07	0.87
	q''(unit)	0.00	1.00	1.00	0.00	1.00	1.00
H	c'' (rupiahs)	0.59	0.64	0.63	0.29	0.95	0.47
	t'' (second)	0.89	0.97	0.61	0.72	0.19	0.89
	q''(unit)	1.00	0.00	1.00	1.00	1.00	1.00
J	c'' (rupiahs)	0.63	0.57	0.55	0.19	0.95	0.66
	t'' (second)	0.87	0.99	0.67	0.75	0.19	0.81
	q''(unit)	0.00	0.00	0.00	1.00	1.00	0.00

The next step is to change the multi-objective objective function into a single-objective form which is done by giving weights to each objective. It is assumed that each objective function has the same level of importance, so that the same weight is given to each objective. In this study, there are three objective functions so that each weight is worth 1/3. Based on Table III and equation (2) by giving a weight of 1/3, the objective function is obtained as written in equation (4)

Maximize

$$
\begin{aligned}
Z(X) &= \frac{1}{3}\sum_{i=1}^{6}\sum_{j=1}^{6} c''_{ij}\,x_{ij} + \frac{1}{3}\sum_{i=1}^{6}\sum_{j=1}^{6} t''_{ij}\,x_{ij} + \frac{1}{3}\sum_{i=1}^{6}\sum_{j=1}^{6} q''_{ij}\,x_{ij} \\
&= 0.49x_{11} + 0.51x_{12} + 0.75x_{13} + 0.60x_{14} + 0.67x_{15} + 0.48x_{16} \\
&\quad +0.45x_{21} + 0.53x_{22} + 0.75x_{23} + 0.66x_{24} + 0.67x_{25} + 0.80x_{26} \\
&\quad +0.46x_{31} + 0.54x_{32} + 0.74x_{33} + 0.63x_{34} + 0.74x_{35} + 0.48x_{36} \\
&\quad +0.50x_{41} + 0.88x_{42} + 0.74x_{43} + 0.32x_{44} + 0.69x_{45} + 0.80x_{46} \\
&\quad +0.83x_{51} + 0.54x_{52} + 0.75x_{53} + 0.67x_{54} + 0.71x_{55} + 0.79x_{56} \\
&\quad +0.50x_{61} + 0.52x_{62} + 0.41x_{63} + 0.65x_{64} + 0.71x_{65} + 0.49x_{66} \qquad (4)
\end{aligned}
$$

The coefficients of equation (4) are summarized in Table IV.

Tabel IV Average Daily Worker Wages, Average Completion Time Per Product, and Number of Defective Products During the Research Period After Being Converted to Single-Objective Form

Job / Worker	Collared Shirts	T-shirts	Short-sleeved shirts	Long-sleeved shirts	Hats	Pants
D	0.49	0.51	0.75	0.60	0.67	0.48
E	0.45	0.53	0.75	0.66	0.67	0.80
F	0.46	0.54	0.74	0.63	0.74	0.48
G	0.50	0.88	0.74	0.32	0.69	0.80
H	0.83	0.54	0.75	0.67	0.71	0.79
J	0.50	0.52	0.41	0.65	0.71	0.49

After obtaining the objective function in single-objective form, the solution of the assignment problem is determined using the Kuhn-Munkres Algorithm. In this study, the assignment problem consists of 6 workers and 6 tasks that will be modeled with a graph. Name the graph G for the graph that will be used in this modeling. Furthermore, the definition of points on the graph G is grouped into 2 subsets, namely $X = \{ x_1, x_2, x_3, x_4, x_5, x_6 \}$ is a set of points representing workers and $Y = \{ y_1, y_2, y_3, y_4, y_5, y_6 \}$ is a set of points representing tasks. In this case, X and Y are mutually exclusive and for each point in X (as well as in Y) are not connected by an edge. The edge in the graph G is defined as worker x_i member of X who can do task y_j member of Y. The weight contained in each edge connecting workers and tasks is the competence possessed by the worker in completing the task where in this study the weight in question is based on the data in Table IV. In this study, all workers can complete all tasks so that each vertex in X is connected by an edge to each vertex in Y. Thus, the graph modeling used in this assignment problem is a complete weighted bipartition graph G as shown in Figure 1.

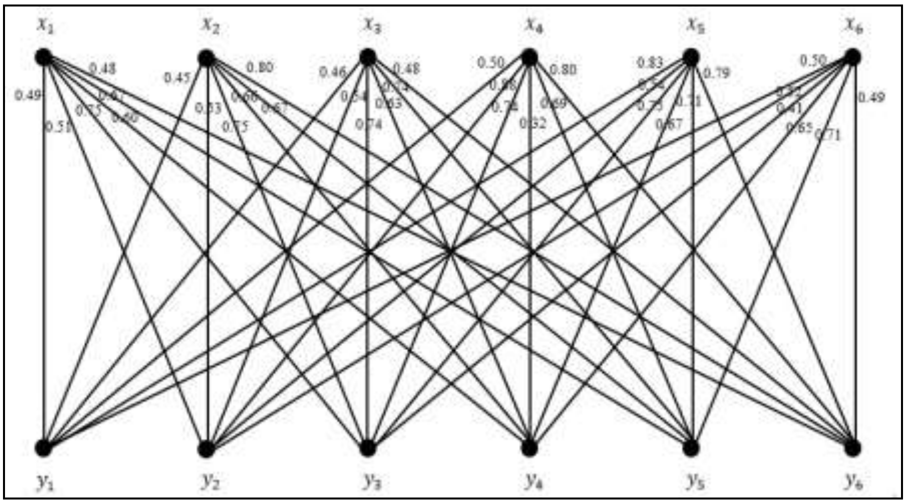

Figure 1 Complete Weighted Bipartition Graph G

The solution of assignment problems with this graph approach is done by finding the optimal matching on a weighted bipartition graph G with the following steps.

Step 1: find the labeling of vertex l and form the spanning part graph G_l of graph G with the following steps.

First, the graph is represented in the form of an adjacency matrix.

$$\begin{array}{c} \\ x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{array} \begin{array}{c} \begin{array}{cccccc} y_1 & y_2 & y_3 & y_4 & y_5 & y_6 \end{array} \\ \begin{bmatrix} 0.49 & 0.51 & 0.75 & 0.60 & 0.67 & 0.48 \\ 0.45 & 0.53 & 0.75 & 0.66 & 0.67 & 0.80 \\ 0.46 & 0.54 & 0.74 & 0.63 & 0.74 & 0.48 \\ 0.50 & 0.88 & 0.74 & 0.32 & 0.69 & 0.80 \\ 0.83 & 0.54 & 0.75 & 0.67 & 0.71 & 0.79 \\ 0.50 & 0.52 & 0.41 & 0.65 & 0.71 & 0.49 \end{bmatrix} \end{array}$$

Then determine a labeling point G, name the labelling l which is defined as follows:

a. For each $x_i \in X$, $l(x_i) = \text{maximum } \{w(x_i y_j) | y_j \in Y\}$, $i, j = 1, 2, 3, 4, 5, 6$

b. For each $y_j \in Y$, $l(y_j) = 0$.

The labeling of point l is obtained as follows:

$$\begin{array}{c|cccccc|c} & y_1 & y_2 & y_3 & y_4 & y_5 & y_6 & \ell(x) \\ x_1 & 0.49 & 0.51 & 0.75 & 0.60 & 0.67 & 0.48 & 0.75 \\ x_2 & 0.45 & 0.53 & 0.75 & 0.66 & 0.67 & 0.80 & 0.80 \\ x_3 & 0.46 & 0.54 & 0.74 & 0.63 & 0.74 & 0.48 & 0.74 \\ x_4 & 0.50 & 0.88 & 0.74 & 0.32 & 0.69 & 0.80 & 0.88 \\ x_5 & 0.83 & 0.54 & 0.75 & 0.67 & 0.71 & 0.79 & 0.83 \\ x_6 & 0.50 & 0.52 & 0.41 & 0.65 & 0.71 & 0.49 & 0.71 \\ \ell(y) & 0 & 0 & 0 & 0 & 0 & 0 & \end{array}$$

Thus, the point labeling is obtained:

$\ell(x_1) = 0.75, \ell(x_2) = 0.80, \ell(x_3) = 0.74, \ell(x_4) = 0.88, \ell(x_5) = 0.83, \ell(x_6) = 0.71.$

$\ell(y_1) = 0, \ell(y_2) = 0, \ell(y_3) = 0, \ell(y_4) = 0, \ell(y_5) = 0, \ell(y_6) = 0.$

c. From the labeling value l, the edge set $E_l = \{x_i y_j \in E(G) \mid x_i \in X, y_j \in Y$ and $l(x_i)+l(y_j)=w(x_i y_j)\}$ is obtained, namely $\{x_1y_3, x_2y_6, x_3y_3, x_3 y_5, x_4 y_2, x_5 y_1, x_6 y_5\}$. Next, graph G_l is formed which is the spanning part graph of G with the edge set E_l as in Figure 2.

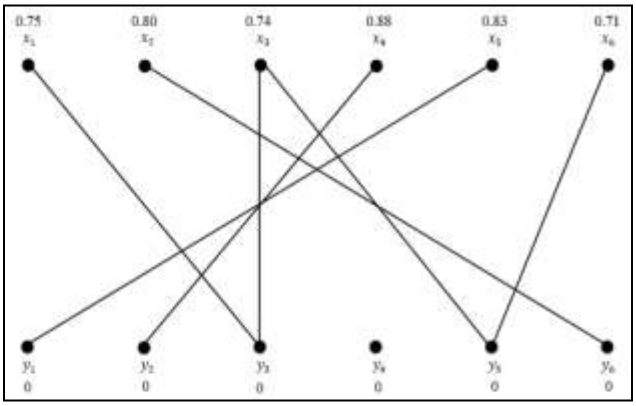

Figure 2 Spanning Section Graph G_l

Step 2: choose any match M on G_l.

After the graph G_l is formed, then any matching M in G_l is selected. For example, the matching M in the selected G_l is $\{x_1y_3, x_2y_6, x_4y_2, x_5 y_1, x_6 y_5\}$ as shown in Figure 3.

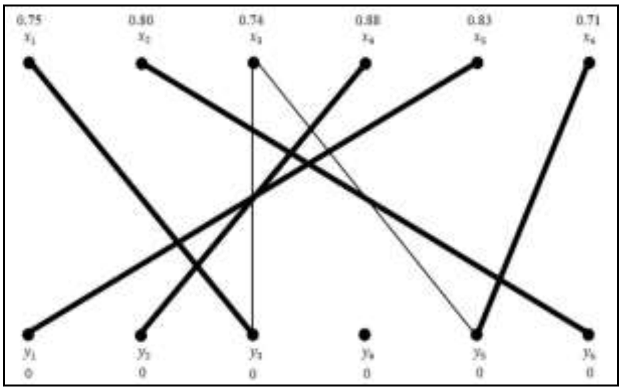

Figure 3 Matching M on Graph G_l

Step 3: Determine whether a perfect match has been obtained or not, with the following stages.

a. After selecting any match M to G_l, it is then determined whether a perfect match has been obtained. If X is covered by M, then a perfect match is obtained, and the algorithm stops. If there is a point in X that is not covered by the matching M, then the algorithm continues. Based on Figure 3, there is a point in X which is not yet covered by the matching M is point x_3, then the algorithm continues to step 3.b.
b. Because there are points on X which is not yet covered by the matching, then determine the root point, namely the point on X which is not covered by the matching M in G_l. This root point is the root point of the alternating tree T that will be formed. Based on Figure 3, point x_3 is a point on X that is not covered by M, so x_3 is the root point.
c. After obtaining x_3 as the root point, then form an alternating tree T from matching M to G_l which is rooted at point x_3 as in Figure 4.

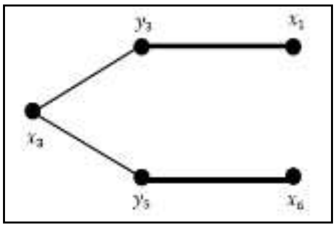

Figure 4. Alternating tree *T* from Matching *M*

Based on Figure 4, the set of points from the Alternating tree T is obtained, namely $V(T)=\{ x_1, x_3, x_6, y_3, y_5\}$.
Then it is determined whether there is an M-augmenting path. If there is an M-augmenting path, then return to step 3.a. If there is no M-augmenting path, then proceed to step 4.
As seen in Figure 4.4, no M-augmenting path is found in the alternating tree T, then proceed to step 4, namely replacing the labeling of point l with a new labeling of point l '.

Step 4: Determine the labeling of new point l '.
This step uses the alternating tree T to calculate the labeling of new point l '. First, find the value of
m_l = min $\{l(x_i) + l(y_j) - w(x_i y_j) | x_i \in X \cap V(T)$ and $y_j \in Y - V(T)$.
Since $X = \{ x_1, x_2, x_3, x_4, x_5, x_6 \}$, $Y = \{ y_1, y_2, y_3, y_4, y_5, y_6\}$, and $V(T)=\{ x_1, x_3, x_6, y_3, y_5\}$, so $X \cap V(T) = \{ x_1, x_3, x_6\}$ and $Y - V(T) = \{ y_1, y_2, y_4, y_6\}$. The value of m_l is the minimum of the results of the following calculations:

$$\ell(x_1) + \ell(y_1) - w(x_1y_1) = 0.75 + 0 - 0.49 = 0.26$$
$$\ell(x_1) + \ell(y_2) - w(x_1y_2) = 0.75 + 0 - 0.51 = 0.24$$
$$\ell(x_1) + \ell(y_4) - w(x_1y_4) = 0.75 + 0 - 0.60 = 0.15$$
$$\ell(x_1) + \ell(y_6) - w(x_1y_6) = 0.75 + 0 - 0.48 = 0.27$$
$$\ell(x_3) + \ell(y_1) - w(x_3y_1) = 0.74 + 0 - 0.46 = 0.28$$
$$\ell(x_3) + \ell(y_2) - w(x_3y_2) = 0.74 + 0 - 0.54 = 0.20$$
$$\ell(x_3) + \ell(y_4) - w(x_3y_4) = 0.74 + 0 - 0.63 = 0.11$$
$$\ell(x_3) + \ell(y_6) - w(x_3y_6) = 0.74 + 0 - 0.48 = 0.26$$
$$\ell(x_6) + \ell(y_1) - w(x_6y_1) = 0.71 + 0 - 0.50 = 0.21$$
$$\ell(x_6) + \ell(y_2) - w(x_6y_2) = 0.71 + 0 - 0.52 = 0.19$$
$$\ell(x_6) + \ell(y_4) - w(x_6y_4) = 0.71 + 0 - 0.65 = 0.06$$
$$\ell(x_6) + \ell(y_6) - w(x_6y_6) = 0.71 + 0 - 0.49 = 0.22$$

Thus, the minimum value of m_l =0.06 is obtained. Next, the labeling of the new point l ' is determined based on equation (3) and the new labeling of l ' is obtained as follows:

$$\ell' = \begin{cases} \ell(x_i) - 0.06, & for\ x_i \in X \cap V(T) \\ \ell(y_j) + 0.06, & \text{for } y_j \in Y \cap V(T) \\ \ell(x_i), \ell(y_j), & \text{others} \end{cases}$$

$$\begin{array}{c|c|cc} & \begin{matrix} \boldsymbol{y_1} & \boldsymbol{y_2} & \boldsymbol{y_3} & \boldsymbol{y_4} & \boldsymbol{y_5} & \boldsymbol{y_6} \end{matrix} & \boldsymbol{\ell(x)} & \boldsymbol{\ell'(x)} \\ \begin{matrix} \boldsymbol{x_1} \\ \boldsymbol{x_2} \\ \boldsymbol{x_3} \\ \boldsymbol{x_4} \\ \boldsymbol{x_5} \\ \boldsymbol{x_6} \end{matrix} & \begin{bmatrix} 0.49 & 0.51 & 0.75 & 0.60 & 0.67 & 0.48 \\ 0.45 & 0.53 & 0.75 & 0.66 & 0.67 & 0.80 \\ 0.46 & 0.54 & 0.74 & 0.63 & 0.74 & 0.48 \\ 0.50 & 0.88 & 0.74 & 0.32 & 0.69 & 0.80 \\ 0.83 & 0.54 & 0.75 & 0.67 & 0.71 & 0.79 \\ 0.50 & 0.52 & 0.41 & 0.65 & 0.71 & 0.49 \end{bmatrix} & \begin{matrix} 0.75 \\ 0.80 \\ 0.74 \\ 0.88 \\ 0.83 \\ 0.71 \end{matrix} & \begin{matrix} 0.69 \\ 0.80 \\ 0.68 \\ 0.88 \\ 0.83 \\ 0.65 \end{matrix} \\ \boldsymbol{\ell(y)} & \begin{matrix} 0 & 0 & 0 & 0 & 0 & 0 \end{matrix} & & \\ \boldsymbol{\ell'(y)} & \begin{matrix} 0 & 0 & 0.06 & 0 & 0.06 & 0 \end{matrix} & & \end{array}$$

Step 5: Forming the spanning part graph $G_{l'}$ of graph G.
From the new labeling value l ', the set $E_{l'} = \{ x_i y_j \in E(G) \mid x_i \in X, y_j \in Y$ and $l'(x_i) + l'(x_i) = w(x_i y_j) \}$ is obtained,
that is $\{ x_1y_3, x_2y_6, x_3y_3, x_3y_5, x_4y_2, x_5y_1, x_6y_4, x_6y_5\}$. Next, the spanning part graph $G_{l'}$ of graph G is formed with the edge set $E_{l'}$ as in Figure 5, then return to step 3.b.

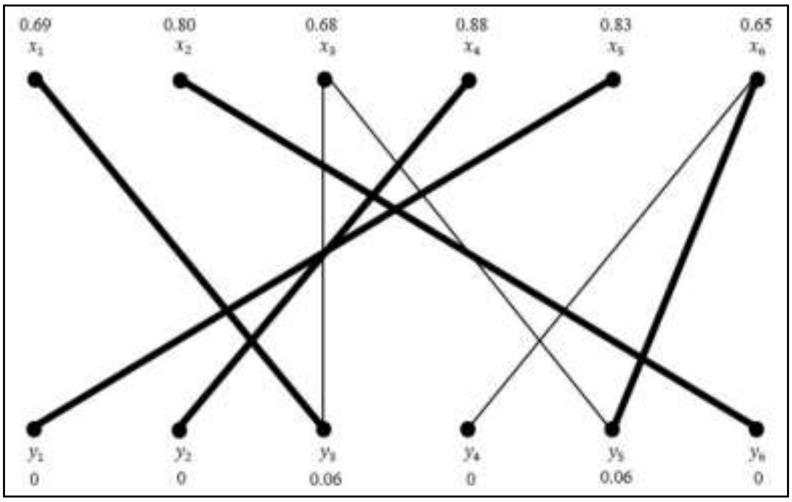

Figure 5. Spanning Section Graph $G_{l'}$ and Matching *M*

Step 3:

b. Based on Figure 5, point x_3 is a point on X that is not covered by M, so x_3 is the root point.

c. After obtaining x_3 as the root point, then form an alternating tree T from matching M to $G_{l'}$ which has its roots at point x_3. Based on Figure 5, an alternating tree T from M is formed which has its roots at point x_3 as in Figure 6.

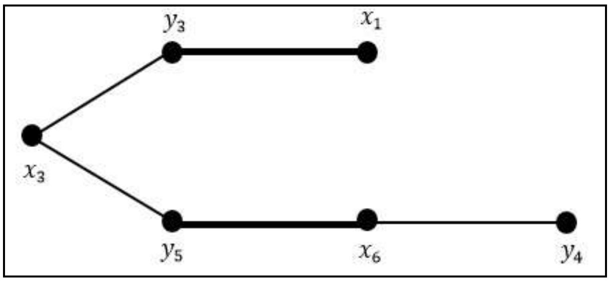

Figure 6. Alternating tree T from Matching M

Based on Figure 6, the set of points obtained from the Alternating tree T is $V(T) = \{x_1, x_3, x_6, y_3, y_4, y_5\}$.

Next, determine whether there is an *M*-augmenting path. If there is an M-augmenting path, then return to step 3.a. If there is no M-augmenting path, then proceed to step 4.

It can be seen in Figure 6 that the *M*-augmenting path is found in the alternating tree *T* as in Figure 7. So we get an extension of *M* along the *M*-augmenting path to get a new matching *M** by changing the side that is not a match to a match (and vice versa) contained in the M-augmenting path as in Figure 4.8. Thus, a new matching *M** is obtained as in Figure 9, then return to step 3.a.

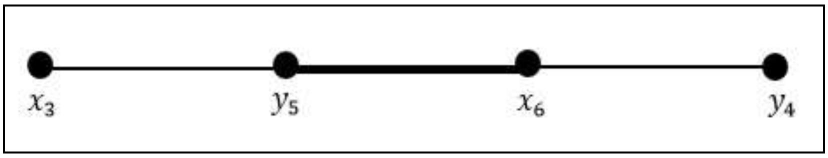

Figure 7 M-augmenting Path on Alternating Tree T

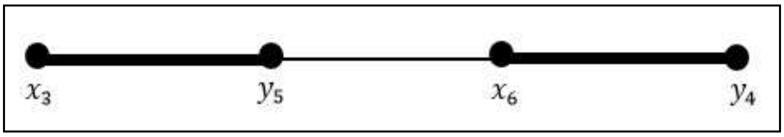

Figure 8 New Matching from M-augmenting Path

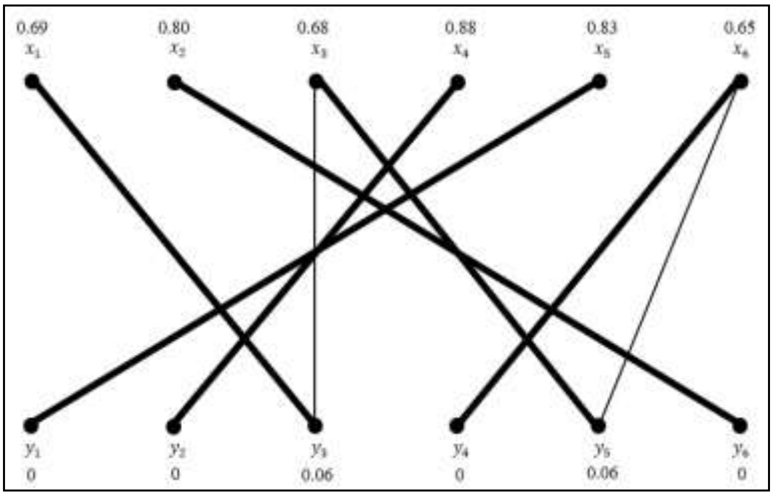

Figure 9 Graph of the Spanning Section $G_{l'}$ and Matching *M**

Step 3:

a. After obtaining a new match M^*, it is then determined whether a perfect match has been obtained. Based on Figure 9, it can be seen that all points on X is covered by M^* therefore M^* is a perfect match on graph $G_{l'}$ and M^* is an optimal match on graph G.

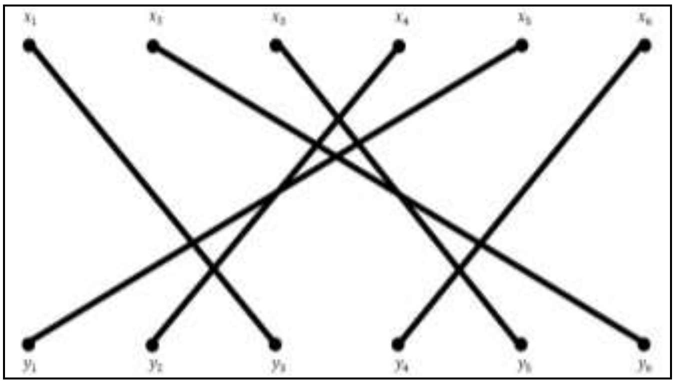

Figure 10 Perfect Matching M* on Graph G

Based on Figure 10, a perfect matching is obtained $M^*=\{x_1y_3, x_2y_6, x_3y_5, x_4y_2, x_5y_1, x_6y_4\}$. Thus, the right worker placement is obtained, namely worker x_1 works on task y_3, worker x_2 works on task y_6, worker x_3 works on task y_5, worker x_4 works on task y_2, worker x_5 works on task y_1, and worker x_6 works on task y_4.

Based on the perfect matching obtained from the Kuhn-Munkres Algorithm and by substituting the values in Table I, the best placement of workers for the ABC confection business is:

a. Worker x_1 (D) is assigned to work on job y_3 (short-sleeved shirts), with an average wage of Rp. 83,077.00 per day, an average completion time of 4160 seconds per product, and 1 unit defective product.
b. Worker x_2 (E) is assigned to work on job y_6 (pants), with an average wage of Rp. 115,111.00 per day, an average completion time of 1751 seconds per product, and 1 unit defective product.
c. Worker x_3 (F) is assigned to work on job y_5 (hats), with an average wage of Rp. 50,000.00 per day, an average completion time of 5760 seconds per product, and 1 unit defective product.
d. Worker x_4 (G) is assigned to work on job y_2 (t-shirts), with an average wage of Rp. 90,563.00 per day, an average completion time of 1113 seconds per product, and 1 unit defective product.
e. Worker x_5 (H) is assigned to work on job y_1 (collared shirts), with an average wage of Rp. 104,000.00 per day, an average completion time of 1662 seconds per product, and 1 unit defective product.
f. Worker x_6 (J) is assigned to work on job y_4 (long-sleeved shirts), with an average wage of Rp. 168,750.00 per day, an average completion time of 2560 seconds per product, and 1 unit defective product.

Thus, the total average wage that must be paid by the garment business is Rp. 611,501.00 per day, with a total average completion time of 17006 seconds per product, and a total of 6 unit defective products.

Based on Table I, the average wage per day to complete each task, before the assignment is carried out using the Kuhn Munkres algorithm, is presented in Table V.

Table V Average Daily Wage for Every Job, average completion time per product (second) and Average Number of Defective products Before Applying Kuhn-Munkres Algorithm

Job	Average Daily Wage (Rp)	Average Completion Time per Product (second)	Average Number of Defective products (unit)
Collared Shirts	109,962.2	1,594.167	$1.8 \approx 2$
T-shirts	100,108.00	1,013.667	$1.8 \approx 2$
Short-sleeved shirts	99,854.33	3,492	$1.2 \approx 1$
Long-sleeved shirts	172,058.5	2,530	$1.2 \approx 1$
Hats	42,583.33	6,836.5	1
Pants	108,927.2	1,866.333	$1.5 \approx 2$
Total	633,493.5	17,332.67	9

Based on Table V and previous results after the assignment using the Kuhn Munkres Algorithm, Table VI was formed which contains a total daily wage ratio, the total number of settlement per product and the total defect product before and after the Kuhn Munkres algorithm was applied.

Table VI. Total Daily Wage Average, Total Settlement per Product and Total Defective Products, Before and After the Kuhn Munkres algorithm is applied

Situation / Viewpoint	Before the Kuhn Munkres algorithm is applied	After the Kuhn Munkres algorithm is applied	The difference
Total Average Daily Wage (Rp)	633,493.5	611,501.00	Cost effective 3.47 %
Total Average Completion Time per Product (second)	17,332	17,006	Save time 1.88 %
Average Number of Defective Products	9	6	Reduce 33.33%

Thus, in this study through the application of the Kuhn Munkres algorithm obtained a daily wage savings of 3.47%, savings in the completion of the product 1.88%, and the reduced defect product produced by 33.33%. This means that the Kuhn Munkres algorithm is effective and efficient in solving the assignment problem in the case in this study.

In this study, the number of workers and the number of tasks are assumed to be the same. But in reality, the number of workers and the number of tasks are often not the same. Therefore, in further research, it is recommended to solve multi-objective assignment problems where the number of workers is not the same as the number of tasks.

IV. CONCLUSION

Based on the discussion, it can be concluded that the Kuhn Munkres algorithm is effective and efficient in solving the assignment problem in the ABC confection business.

ACKNOWLEDGMENT

Thanks to ABC confectionery business (in this study, the name is disguised) who provided the data for analysis.

REFERENCES

1) Bao, C. P., Tsai, M. c., & Tsai, M. i. 2007. A New Approach to Study The Multi-objective Assignment Problem. WHAMPOA - An Interdisciplinary Journal 53, 123-132.
2) Bondy, J. A., & Murty, U. S. 1976. Graph Theory with Applications. New York: Elsevier Science Publishing Co., Inc.
3) Hilier, F. S., & Lieberman, G. J. 2015. Introduction to Operation Research (10th edition). New York: McGraw-Hill Education.
4) Kuhn, H. W. 1955. The Hungarian Method for the Assignment Problem. Journal of Naval Research Logistics Quarterly, 83-97.
5) Liana, L. 2017. Pencapaian Biaya Minimum Menggunakan Metode Hungarian dan Daftar Kombinasi. JBE (Jurnal Bingkai Ekonomi), 2(2), 36-37.
6) Munir, R. 2010. Matematika Diskrit (edisi ketiga). Bandung: Informatika Bandung.

7) Munkres, J. 1957. Algorithms for the Assignment and Transportation Problems. Journal of the Society for Industrial and Applied Mathematics Vol. 5, No. 1, 32-38.
8) Raharjo, D. 2010. Proses Optimasi dan Idealisasi Masalah Penugasan Multi-Objective Menggunakan Metode Hungaria pada Contoh Kasus Usaha Kerajinan Gitar di Ngrombo Baki Sukoharjo. Skripsi. Surakarta: Universitas Sebelas Maret.
9) Ristono, A., & Puryani. 2011. Ekonomi Teknik. Yogyakarta: Graha Ilmu.
10) Swadian, A. 2007. Penentuan Matching Maksimum Pada Bipartite Graph (Bigraph). Thesis. Malang: Universitas Brawijaya.
11) Utami, T. B., Rosyida, I., & Mulyono. 2018. Penerapan Algoritma Kuhn-Munkres dalam Penyelesaian Masalah Penugasan Multi-objective pada Industri Konveksi Tas DP.SPORTY. Prosiding Seminar Nasional Matematika, 765-773.
12) White, D. J. 1984. A Special Multi-Objective Assignment Problem. Journal of the Operational Research Society Vol. 35, No. 8, 759-767.

There is an Open Access article, distributed under the term of the Creative Commons Attribution – Non Commercial 4.0 International (CC BY-NC 4.0) (https://creativecommons.org/licenses/by-nc/4.0/), which permits remixing, adapting and building upon the work for non-commercial use, provided the original work is properly cited.