



Problem of Maximum Matching in Non-Bipartite Graph Using Edmonds' Cardinality Matching Algorithm and Its Application in the Battle of Britain Case

Muchammad Abrori¹, Mohammad Imam Jauhari²

^{1,2} Department of Mathematics
State Islamic University Sunan Kalijaga, Indonesia

Email: borymuch@yahoo.com, jonycrisha@gmail.com

ABSTRACT

Matching is a part of graph theory that discusses pair. A matching M is called to be maximum if M has the highest number of elements. A blossom which is encountered in non-bipartite graph can cause failure in process of finding the maximum matching in non-bipartite graph. One of the algorithms that can be used to find a maximum matching in non-bipartite graph is Edmonds' Cardinality Matching Algorithm. Shrinking process is done in each blossom B_i that is encountered to become pseudovortex b_i , in a way that each blossom does not interfere the process of finding a maximum matching in non-bipartite graph. In order to accelerate the finding, simple greedy method is used to perform initialization of matching and BFS algorithm is also used in constructing an alternating tree in a non-bipartite graph. The research discussed the finding of maximum matching in non-bipartite graph using Edmonds' cardinality matching algorithm. In addition, this research gave a sample of its application in the resolution of The Battle of Britain case. The result obtained is a maximum matching in non-bipartite graph. The maximum matching obtained is a solution to the case of The Battle of Britain.

Keywords: Edmonds' cardinality *matching*, *maximum matching*, The Battle of Britain

INTRODUCTION

A matching M in G is a subset of $E(G)$ where there are no two edges in M that meet in the same vertex [3]. Matching concept can be used in many things especially for the problem of pairing. In the discussion about matching, there is a term called maximum matching. Maximum matching is matching M with the highest cardinality [1]. Generally, the finding of maximum matching is performed in bipartite graph, but it can also be executed in non-bipartite graph.

The process of finding the maximum matching in non-bipartite graph is a bit different from that in bipartite graph where in non-bipartite graph, it should contain at least an odd cycle that issues a blossom. A blossom can cause a failure in some algorithms of finding the maximum matching or it cannot be engaged in finding the maximum matching in non-bipartite graph.

One of the algorithms that can be used in finding maximum matching in non-bipartite graph is Edmonds' cardinality matching algorithm. This algorithm can handle a

blossom found in non-bipartite graph. Hence the process of finding maximum matching can be done without any obstacle even though there is a blossom while finding the maximum matching.

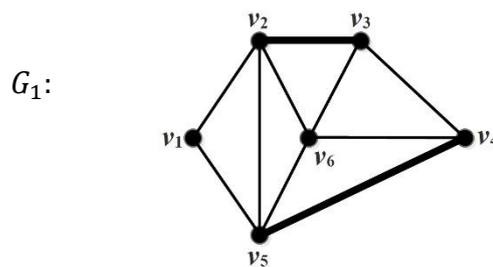
METHODS

The research was begun with studying the basic concept about graph theory which cover definition of basic graph, types of graph, subgraph and skelter subgraph, operation of graph, and basic concept about tree graph and matching concept in G graph. Besides, the research also explains about Breadth-First Search (BFS) algorithm to arrange tree construction of alternating T . Then, it discusses Edmonds' cardinality matching algorithm and its application in finding a maximum matching in a graph and in the last chapter, there is a sample of pairing a pilot in a war of Britain in 1940 which was taken from $M. Gondran$ and $Minoux's$ book entitled "*Graphs and Algorithms*".

RESULTS AND DISCUSSION

Definition: Matching [6]

A matching M in G is a subset of a set $E(G)$ where there are no two edges in M that meet at the same vertex. A G graph with matching $M \subseteq E(G)$ can be written in (G, M) . Example: A graph G_1 with matching $M = \{v_2v_3, v_4v_5\}$.



There are two vertex terms in matching graph:

1. Saturated vertex is a vertex which is incident with the edges in matching M . [8]
Example: In G_1 , the saturated vertex is vertex v_2, v_3, v_4, v_5 .
2. Exposed vertex is a vertex which is not incident with each edge in matching M . [8].
Example: In G_1 , the exposed vertex is vertex v_1 and v_6 .

Apart from vertex, path in the matching graph is also divided into two kinds:

1. Alternating path is a path with the edges that alternate between the edge that is included in M and not included in M . [7]
Example: In G_1 path $P: v_1 - v_2 - v_3 - v_4 - v_5$ is an alternating path



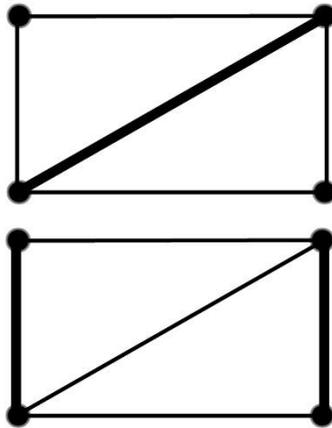
2. Augmenting path is a path with the beginning and ending vertex that are exposed vertex. [7]
Example: In G_1 path $P: v_1 - v_2 - v_3 - v_4 - v_5 - v_6$ is an augmenting path



Maximum Matching

Matching M in a G graph is called maximum if there is no matching M' in G with $|M'| > |M|$. [1]

Example:



Graph with a matching that is not maximum yet



Graph with a matching that is already maximum

Matching Initialization

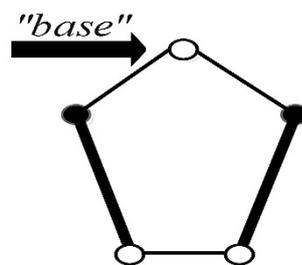
Initialization process is used to shorten the finding of maximum matching in a graph. This process uses a help of simple greedy method as follows: [9]

1. $M = \emptyset$
2. If there is no more e edge which can be added into M , then stop
3. Otherwise, select e edge that does not have the same vertex with the edges in M
 - 3.1 $M = (M \cup e)$
4. Turn back to M .

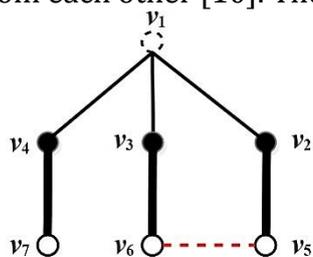
Blossom

A blossom B in a matching graph (G, M) is an odd cycle with $M \cap B$ which is the maximum matching in B with base which is an exposed vertex or outer vertex for $M \cap B$. [3]

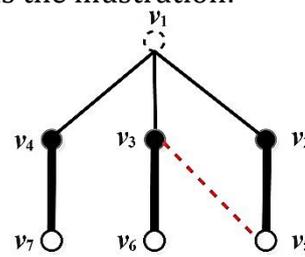
Example:



In a case of bipartite graph, it is not possible to find a blossom while constructing alternating tree T because bipartite graph does not have two vertices that are adjacent to each other at the same partition. Therefore, it is impossible for two vertices at the same layer to join each other [10]. The following below is the illustration:

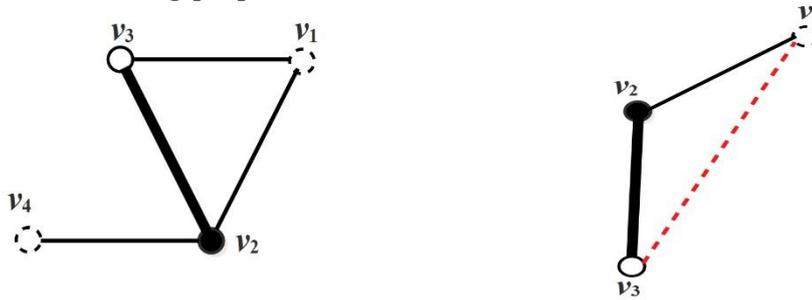


In non-bipartite graph



In bipartite graph

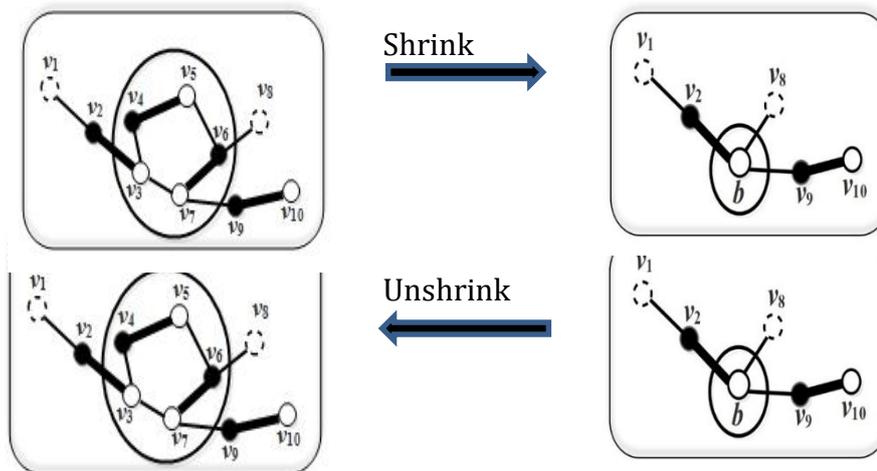
A blossom in non-bipartite graph can raise a possibility of not finding an augmenting path while constructing alternating tree T so that it causes a failure in the finding process of maximum matching [10].



Shrinking Blossom

To prevent the problem caused by a blossom B , the process of shrinking and unshrinking should be done to a blossom B that is encountered when the alternating tree T is shrinking. A shrinking blossom is a shrinking process to odd cycle in (G, M) by replacing all of the vertices and edges in blossom B with a labeled vertex, for example b with $b = B/B$. Vertex b is called pseudovertex. When the unshrink process toward b is not executed yet (by deleting label b then replacing it with B as it was), ignore all of the vertices in B and all of the edges which join one vertex to the other in B . Besides, the unshrink process toward b is a returning process of all vertices and edges included in blossom B [3].

Example:



Below is a lemma which ensures that shrinking and unshrink process can be used in finding a maximum matching.

Lemma of Shrinking Cycle [7]

Supposed G is a graph with matching M , and r is an exposed vertex in (G, M) . Supposed that during the process of establishing an alternating tree T with the root r , it encounters blossom B . If vertex w is a base of blossom B and graph $G' = G/B$ is a graph produced from the shrinking of blossom B into pseudovertex b . Graph (G, M) will contain an augmenting path that begins in vertex r if and only if graph (G', M') with matching $M' = M/B$ contains an augmenting path that starts in vertex r .

Alternating Tree

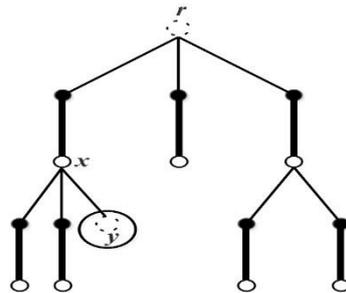
An alternating tree T is a tree whose edges alternate between the edge that is included in M and not included in M and are rooted in an exposed vertex of which each edge relates an inner vertex with an outer vertex thus every inner vertex in T is exactly incident with two edges in T . Inner vertex is a vertex that occupies odd layer (1,3,5, ...), while outer vertex is a vertex that occupies even layer (0,2,4, ...) [3].

The following is the steps to construct an alternating tree T using BFS: [7]

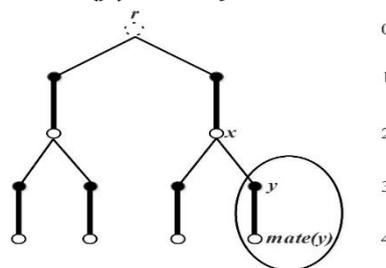
(Root): Choose an exposed vertex r in graph (G, M) . Make the exposed vertex r as the root (layer 0) in alternating tree T .

1. In layer 1, add all the vertices $a_1, a_2, a_3, \dots, a_p$ which are adjacent to the root r . Keep in mind that all of the vertices that are added are only saturated ones.
2. Add vertex $b_i = \text{mate}(a_i)$ on layer 2 (even) in T .
3. For the next layer, add all the vertices $c_1, c_2, c_3, \dots, c_p$ where c_i is adjacent to at least one b_i , while the edge that relates c_i with b_i is not contained in M . Continue step 1, 2, and 3 until finding one of these two possibilities as follows: (a) an augmenting path, (b) an odd cycle.
4. If x is a vertex in layer $2i$, and $y \neq \text{mate}(x)$ is a vertex that is adjacent to x . So, there are four possibilities:

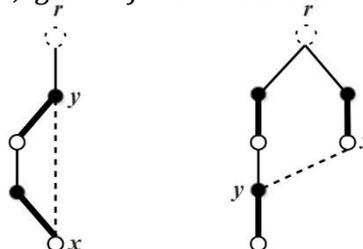
Case 1: y is an exposed vertex (and not yet contained in T). So, an augmenting path has been found. Therefore, the constructing process of alternating tree T is discontinued.



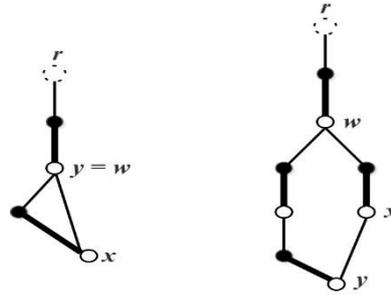
Case 2: y is not an exposed vertex; neither y nor $\text{mate}(y)$ is contained in T . So, add y into layer $2i + 1$ and $\text{mate}(y)$ into layer $2i + 2$.



Case 3: y is already contained in T as an inner vertex. So, a cycle with an even length has been found. Therefore, ignore if the condition below is found.



Case 4: y is already contained in T as an outer vertex. Blossom has been found. Stop the constructing process of alternating tree T and do the shrinking process of the blossom.



The followings are theorems that become the basis of Edmonds' cardinality matching algorithm in finding a maximum matching in a graph [7].

Theorem: [2]

M_1 and M_2 are matching in graph G . If H is a spanning subgraph of G , then in every component of H with $(E(H) = (M_1 - M_2) \cup (M_2 - M_1))$ thing that should fulfill the following requirements:

1. An isolated vertex,
2. An even cycle in which the edge of the cycle alternates in M_1 dan M_2 ,
3. A path in which the edge of the path alternates in M_1 dan M_2 thus every end of the vertex is unsaturated in M_1 or M_2 but not in both.

Theorem: [7]

Matching M in graph G is called maximum if and if only there is no augmenting path in G with respect to M .

Edmonds' Cardinality Matching Algorithm

The followings are the steps from Edmonds' cardinality matching algorithm: [4]

Step 1 (initialization)

Graph G is a non-bipartite graph with matching $M_0 = \emptyset$. Do the matching initialization in G by using greedy algorithm.

Step 2 (An Examination of exposed vertex)

- 2.1 If all vertices in (G, M) are incident with a matching edge, go to step 5.
- 2.2 If there is only one unexamined exposed vertex in (G, M) , go to step 5.
- 2.3 Otherwise, select any exposed vertex r in (G, M) as a root and construct alternating tree T :
 - 2.3.1 If the alternating tree T ends with an exposed vertex s wheres $\neq r$, an augmenting path from r to s has been found. Go to step 3.
 - 2.3.2 If the alternating tree T encountered an odd cycle, stop the constructing process and go to step 4.

Step 3 (Augmenting Path):

- 3.1 If the augmenting path P obtained from the alternating tree T contains pseudovertex b_i ($i = 1, 2, 3 \dots$), then unshrink the pseudovertex b_i into a blossom B_i as previously. By using path P , trace backwards starting from the exposed vertex s to vertex r (root) through an alternating path with an even length in blossom B_i . Do the same steps until augmenting path in (G, M) is found in which it does not contain any pseudovertex b_i . Augment matching M in (G, M) with $M' = M \oplus P$. Go to step 3.2.

3.2 If the augmenting path P that has been obtained from the alternating tree T does not contain pseudovortex b_i ($i = 1,2,3 \dots$), then augment matching M in (G, M) with $M' = M \oplus P$. Go to step 2.

Step 4 (Odd cycle)

This step is reached only if the alternating tree T encounters a cycle with an odd length (called blossom). Suppose a blossom B_i ($i = 1,2,3 \dots$) is encountered in the constructing process of alternating tree T , then shrink blossom B_i into a pseudovortex b_i . Continue the construction process of alternating tree T with root r as step 2.

Step 5

The matching on graph G is maximum. Stop the finding process.

Sample Case (The Battle of Britain)

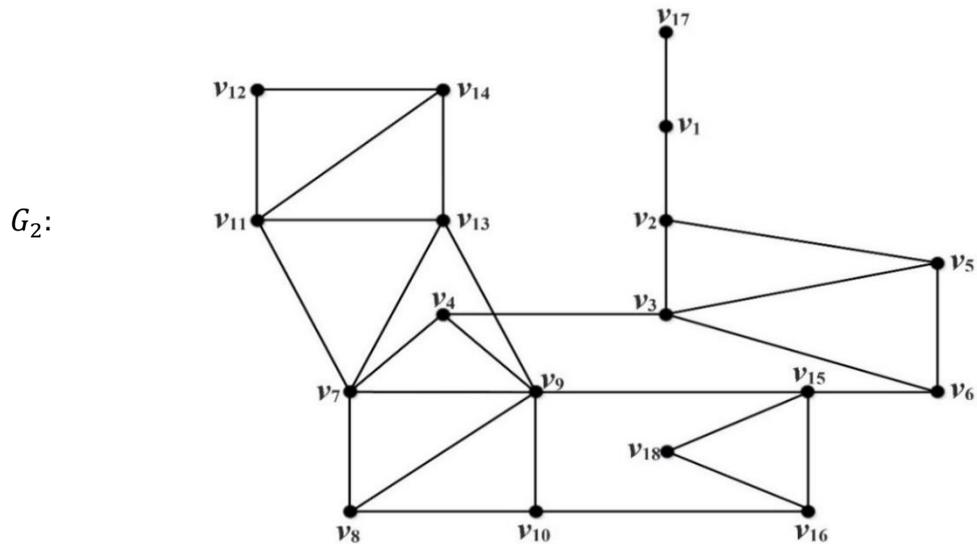
In the Battle of Britain (1940), The Royal Air Force provided several aircrafts to thwart the German air force attack. The aircrafts provided by the *Royal Air Force* could be flown only by two people as a pilot and co-pilot. To get the best results in the mission to thwart German air force attack, the *Royal Air Force* brought some of the best pilots from all regions in Britain. However, there was a problem to solve by the Royal Air Force: among all the pilots that were called, some of them could not fly together in one aircraft because there were differences in terms of language, the ability of flying techniques, as well as in terms of flying experience. Suppose the Royal Air Force obtained 18 best pilots from all regions in Britain that were ready to fly the aircraft. Based on the problem, the Royal Air Force wished to pair the 18 pilots where each pair consisted of two pilots in such a way that the number of all the possible pairs is the highest number. In order to solve such problem, the Royal Air Force conducted a series of tests to the 18 pilots including language test, flying technique test, and psychological test to determine the suitability of the 18 pilots to fly together in one aircraft as a pilot and co-pilot [5].

Table 1 Table of pairing between each aircraft pilot

The pilot's number $i = (1, 2, \dots, 18)$	The result of pairing between each aircraft pilot
1 st Pilot	Pilot: 2 nd and 17 th
2 nd Pilot	Pilot: 1 st , 3 rd , and 5 th
3 rd Pilot	Pilot: 2 nd , 4 th , 5 th , and 6 th
4 th Pilot	Pilot: 3 rd , 7 th , and 9 th
5 th Pilot	Pilot: 2 nd , 3 rd , and 6 th
6 th Pilot	Pilot: 3 rd , 5 th , and 15 th
7 th Pilot	Pilot: 4 th , 8 th , 9 th , 11 th , and 13 th
8 th Pilot	Pilot: 7 th , 9 th , and 10 th
9 th Pilot	Pilot: 4 th , 7 th , 8 th , 10 th , 13 th , and 15 th
10 th Pilot	Pilot: 8 th , 9 th , and 16 th
11 th Pilot	Pilot: 7 th , 12 th , 13 th , and 14 th
12 th Pilot	Pilot: 11 th and 14 th
13 th Pilot	Pilot: 7 th , 9 th , 11 th , and 14 th
14 th Pilot	Pilot: 11 th , 12 th , and 13 th
15 th Pilot	Pilot: 6 th , 9 th , 16 th , and 18 th
16 th Pilot	Pilot: 10 th , 15 th , and 18 th
17 th Pilot	Pilot 1 st
18 th Pilot	Pilot: 15 th and 16 th

By transforming the problem into the form of graph, where the pilots are represented as a vertex and the corresponding relationship between each pilot is

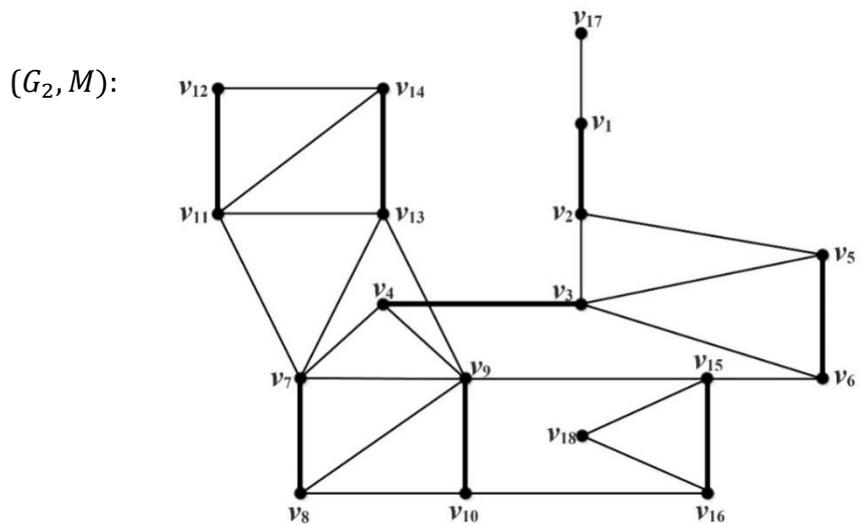
represented as an edge, the following graph is obtained: (The following form of graph is taken from [7]):



By following the steps from Edmonds' cardinality matching algorithm, we obtain maximum matching on graph G_2 as follows:

Step 1 (initialization):

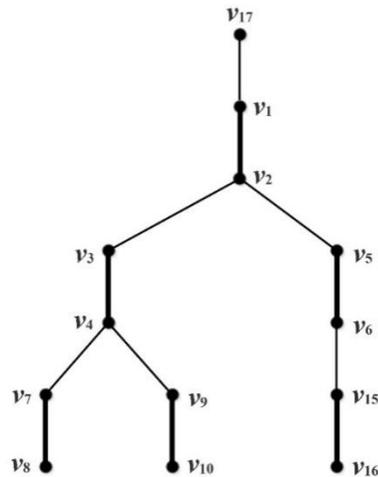
By doing the process of matching initialization to graph G_2 , we obtain graph (G_2, M) with $M = \{v_1v_2, v_3v_4, v_5v_6, v_7v_8, v_9v_{10}, v_{11}v_{12}, v_{13}v_{14}, v_{15}v_{16}\}$ as follows:



Since there are still two exposed vertices left that are v_{17} and v_{18} , then go to step 2.3
Step 2.3 :

Vertex v_{17} is selected as the root, by using BFS we obtain an alternating tree T_1 that corresponds to graph (G_2, M) as follows:

Tree T_1 :

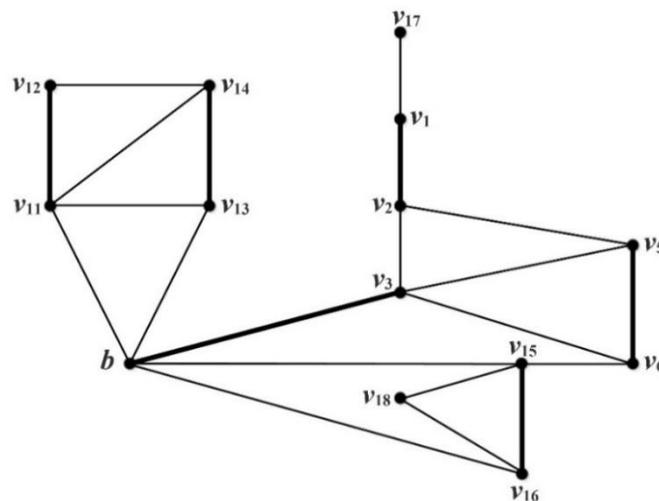


Since blossom $B = \{v_4, v_8, v_9, v_{10}\}$ has been encountered, then go to step 4.

Step 4:

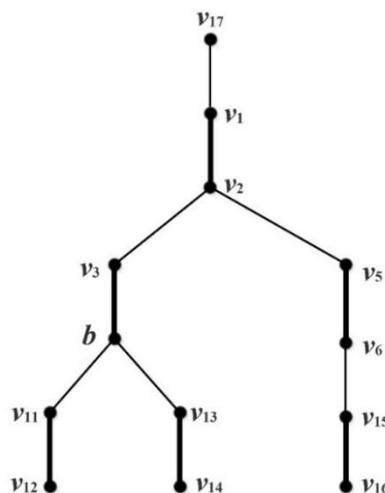
Shrink blossom $B = \{v_4, v_8, v_9, v_{10}\}$ to obtain a graph as follows:

(G'_2, M_1) :

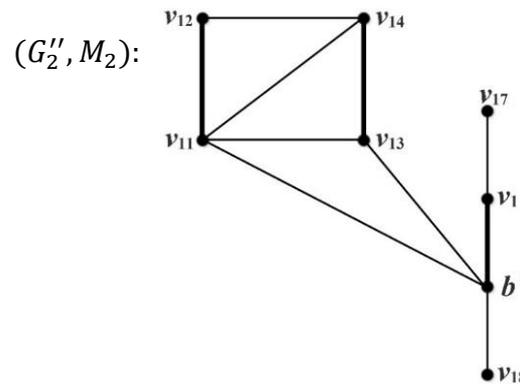


By continuing the constructing process of alternating tree that corresponds to graph (G'_2, M_1) , we obtain:

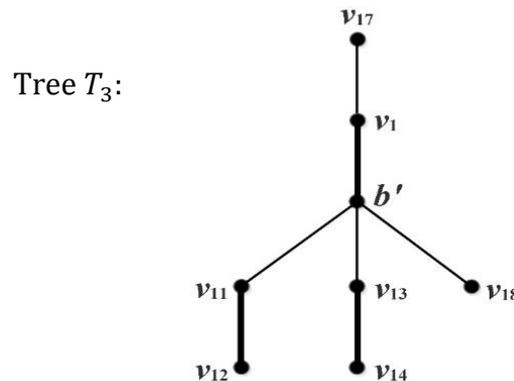
Tree T_2 :



Since blossom $B' = \{b, v_2, v_3, v_5, v_6, v_{15}, v_{16}\}$ has been encountered, then go to Step 4:
Shrink blossom $B' = \{b, v_2, v_3, v_5, v_6, v_{15}, v_{16}\}$ to obtain a graph as follows:



By continuing the constructing process of alternating tree that corresponds to graph (G_2'', M_2) , we obtain:



Since an augmenting path $P'' : v_{18} - b' - v_1 - v_{17}$ that contains *pseudovertex* b' has been obtained, then go to step 3.1

Step 3.1:

By unshrinking the pseudovertex b' , we obtain augmenting path $P' : v_{18} - v_{15} - v_{16} - b - v_3 - v_2 - v_1 - v_{17}$ that contains pseudovertex b , then go to step 3.1

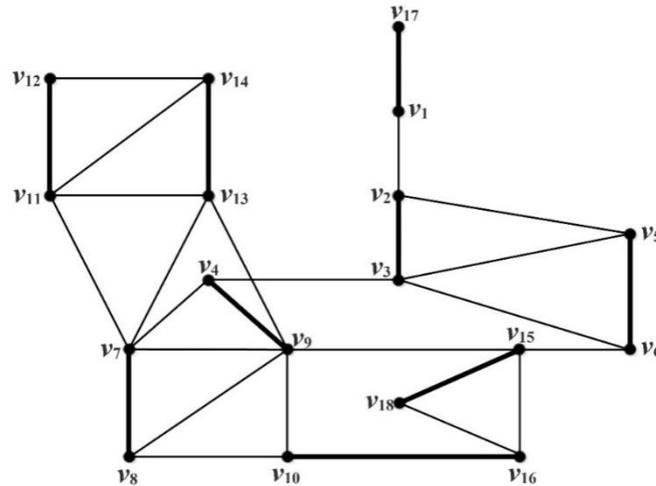
Step 3.1:

By unshrinking the pseudovertex b , we obtain augmenting path $P : v_{18} - v_{15} - v_{16} - v_{10} - v_9 - v_4 - v_3 - v_2 - v_1 - v_{17}$ which does not contain any pseudovertex, then go to step 3.2

Step 3.2:

Augment the initial matching M using augmenting path P , we obtain a new matching in (G_2, M) as follows:

(G_2, M') :



Since all vertices in (G_2, M') are incident with a matching edge, then go to step 5.

Step 5:

Matching M' in graph G_2 is maximum, therefore the finding process is discontinued.

From the steps that have been done, we obtain a maximum matching M' in G_2 with cardinality of matching M' that is $|M'| = 9$. Matching M' that has been obtained is the solution of the problem of pairing the 18 pilots as described above with details as follows:

1. Pilot number 1 pairs up with pilot number 17
2. Pilot number 2 pairs up with pilot number 3
3. Pilot number 4 pairs up with pilot number 9
4. Pilot number 5 pairs up with pilot number 6
5. Pilot number 7 pairs up with pilot number 8
6. Pilot number 10 pairs up with pilot number 16
7. Pilot number 11 pairs up with pilot number 12
8. Pilot number 13 pairs up with pilot number 14
9. Pilot number 15 pairs up with pilot number 18

And the maximum number of aircrafts that can be flown is as many as 9 aircrafts.

CONCLUSION

Matching M in graph G is a subset of $E(G)$ edge set where there are no two edges in Matching M that meet each other at the same vertex in graph G .

Generally, Edmonds' cardinality matching algorithm is divided into three steps where the first step is the process of matching initialization. The second step is shrinking and unshrinking of blossom B_i . The third step is to execute augmenting- M if the augmenting P path has been found.

Edmonds' cardinality matching algorithm can be applied to find for maximum matching especially towards non-bipartite graph. The process of matching initialization is the effort to find the first matching to fasten the finding of maximum matching. Alternating T tree is a T graph concept that is used to find an augmenting path to obtain a new matching that has bigger cardinality than the first matching from the process of matching initialization.

REFERENCES

- [1] Bondy, J. A., And Murty, U. S. R., 1976, *Graph Theory with Applications*, New York: Mac Millan Press.
- [2] Chartrand, G., And Lesniak, L., 1996, *graph dan digraphs (third edition)*, London: Chapman dan Hill.
- [3] Edmons, Jack, 1965, *paths, trees, and flowers*, canadian j.math., 17, 449-467.
- [4] Evans, James R., and Edward Minieka, 1992, *Optimization Algorithms for Networks and Graphs*, New York: Marcel Dekker, Inc.
- [5] Gondran, M., and M. Minoux, 1984, *Graphs and Algorithms*, New York: Jhon Wiley & Sons.
- [6] Gross, Jonathan L., and Jay Yellen (Editors), 1999, *Handbook of Graph Theory*, USA: CRC Press.
- [7] Jungnickel, Dieter, 2008, *Graphs, Networks, and Algorithms (Third Edition)*, Berlin: Springer - Verlag.
- [8] Lovasz, L., and M. D. Plummer, 1986, *Matching Theory*, USA: North - Holland.
- [9] Winter, 2005, *Maximum Matching*, CS105,
www.cs.dartmouth.edu/~ac/Teach/CS105.../kavathekar-scribe.pdf
- [10] Zwick, U., 2009, *Maximum Matching in Bipartite and Non-bipartite Graphs*,
<http://www.cs.tau.ac.il/~zwick/grad-algo-0910/match.pdf>