On the modification Highly Connected Subgraphs (HCS) algorithm in graph clustering for weighted graph

E R Albirri¹, K A Sugeng², D Aldila²
¹Mathematics Edu. Depart. University of Jember, Indonesia
²Mathematics Depart. Universitas Indonesia, Indonesia
E-mail: ermitara@unej.ac.id, kiki@sci.ui.ac.id, aldiladipo@sci.ui.ac.id

Abstract. Nowadays, in the modern world, since technology and human civilization start to progress, all city in the world is almost connected. The various places in this world are easier to visit. It is an impact of transportation technology and highway construction. The cities which have been connected can be represented by graph. Graph clustering is one of ways which is used to answer some problems represented by graph. There are some methods in graph clustering to solve the problem specifically. One of them is Highly Connected Subgraphs (HCS) method. HCS is used to identify cluster based on the graph connectivity \(k\) for graph \(G\). The connectivity in graph \(G\) is denoted by \(k(G)\). If \(k(G) > \frac{n}{2}\) that \(n\) is the total of vertices in \(G\), then it is called as HCS or the cluster. This research used literature review and completed with simulation of program in a software. We modified HCS algorithm by using weighted graph. The modification is located in the Process Phase. Process Phase is used to cut the connected graph \(G\) into two subgraphs \(H\) and \(\bar{H}\). We also made a program by using software Octave-401. Then we applied the data of Flight Routes Mapping of One of Airlines in Indonesia to our program.

1. Introduction
In this new world, technology, communication and human civilization are growing rapidly. Everyone can communicate and of course visits each other easily. All places in the world are almost connected. This connection can be represented by graph [3]. Graph Theory is a theory in discrete mathematics which discusses not only about theory but also the application. One of the graph theory application is graph clustering. There are many ways or methods to cluster the graph easily, one of them is Highly Connected Subgraphs (HCS) method.

Clustering is defined as a process of grouping element from a data into subset of its element of data (cluster) [6], [7]. Graph Clustering is a process of grouping elements in graph (edge or vertex) becoming subgraphs, it is called as cluster [9], [2]. Highly Connected Subgraphs (HCS), is a method of graph clustering to find out the cluster based on its edge connectivity. This method can be used on two kinds of graph. They are classified as weighted graph and non-weighted graph. The determination of the graph is done to get the result of minimum cut. Next, we look for the subgraphs which result in clusters, identified as Highly Connected Subgraphs (HCS). The edge connectivity of \(G\) (\(k(G)\)) is minimum of the sum of edges in \(G\) which is cut such that \(G\) becomes disconnected [6], [10]. Cut in \(G\) is a set of edge \(S\) which if it is deleted, then it will make \(G\) be disconnected [5].
Let $G$ be a simple, connected, and weighted graph which can be seen in Figure 1. Based on the label, $G$ can be labeled in three ways. It can be done in every vertex, edges, or both. Moreover, $G$ which has been labeled in the edges is called weighted graph [11],[8]. Weighted graph consists of two, weighted vertex graph and weighted edge graph. In this research, we use weighted edge graph and call it as weighted graph.

One of matrix which represents a graph is adjacency matrix. Let $G$ be a graph which has $V(G) = \{1, 2, \ldots, n\}$ and $E(G) = \{e_1, e_2, \ldots, e_n\}$. Adjacency matrix for non-weighted graph is denoted by $A(G) = [a_{ij}]$ [1]. $A(G)$ has size $n \times n$ with:

$$a_{i,j} = \begin{cases} 1 & \text{if } v_i \text{ adjacent } v_j; \\ 0 & \text{others}. \end{cases}$$

Next, adjacency matrix can contain edge weight of graph $G$ as it is defined above. It changes entry 1 with its edge weight.

2. Research Method

The research methods that we use is the literature review and completed with simulation by using software Octave−401. The literature review has some steps. It consists of:

(i) study of general graph
(ii) study of general graph clustering
(iii) study of HCS algorithm for weighted graph

Then, we modify HCS algorithm of weighted graph in Octave language by using software Octave−401.

Figure 1. Simple Connected Weighted Graph

Figure 2. The Example of HCS Subgraph
3. Result

In this section we partially discuss about Highly Connected Subgraphs, the algorithm of Highly Connected Subgraphs, the modification of algorithm for weighted graphs, and the application of Highly Connected Subgraphs for weighted graph in software Octave−401.

3.1. Highly Connected Subgraphs

**Definition 1** [6] Highly Connected Subgraphs (HCS) is a method of graph clustering which aims to look for cluster based on edge connectivity \( k(G) \) of graphs.

Algorithm of this method has the same name as the name of the method. It is Highly Connected Subgraphs (HCS) algorithm. This algorithm is found by Hartuv and Shamir [6]. HCS algorithm is used to do graph clustering in a graph data which produces one or more clusters. The cluster is identified as HCS if the total of edge connectivity \( k(G) > \frac{|V|}{2} \). For example, there is graph \( G = (V(G), E(G)) \) with \( V(G) = \{A, B, C, D, E, F\} \) and \( E(G) = \{f, g, h, i, j, k, l\} \). The graph in Figure 1 is not HCS because of \( k(G) = 3 \) and \( \frac{|V|}{2} = 3 \). Besides, subgraph \( H = (V(H), E(H)) \) with \( V(H) = \{A, D, C\} \) and \( E(H) = \{i, h, l\} \) in the Figure 1 is HCS because it has \( k(H) = 2 \) and \( \frac{|V|}{2} = \frac{3}{3} \). Subgraph \( H \) can be seen in Figure 2. The Algorithm 1 is HCS algorithm owned by Hartuv and Shamir [6].

\[
\begin{align*}
\text{HCS}(G(V, E)) & \quad \text{begin} \\
& \quad (H, \bar{H}, C) \leftarrow \text{MINCUT}(G) \\
& \quad \text{if } G \text{ is Highly Connected Subgraph (HCS) then} \\
& \quad \quad \text{return } G \\
& \quad \text{else} \\
& \quad \quad \text{HCS}\{H\}, \text{HCS}\{\bar{H}\} \\
\end{align*}
\]

**Algorithm 1:** HCS Algorithm

3.2. Algorithm of Highly Connected Subgraphs

HCS algorithm for non-weighted graph is started by looking for the total of minimum cut in graph. **Minimum Cut** in this graph is defined as minimum number of cut edges such that it becomes disconnected graph. Next, we recheck whether \( G \) is HCS cluster. If it is not, then the plot of this algorithm recurs in subgraph \( H \) till we get the HCS cluster, as well as in subgraph \( \bar{H} \). Figure 3 shows flowchart of using HCS in non-weighted graph [10].

3.3. Modification of HCS Algorithm for Weighted Graphs

It needs a modification to create a program by using HCS algorithm. It happens because the minimum cut used is the weighted graph. **Minimum Cut** is the minimum number of deleted graph edge which makes \( G \) be disconnected graph. In Figure 4 you can see the Process Phase \((G, w)\) on a blue color. On weighted graph, the determination of minimum cut is different. It is minimum number of the weight on the graph which is lost to make graph \( G \) be disconnected. Because we use the weighted graph, we need an additional phase to determine the minimum cut which will be cut. This phase is called Process Phase. We can see the flowchart of Process Phase \((G, w)\) which uses minimum cut algorithm for weighted graph in Figure 5. The implementation of this algorithm is problematic, that is why we do research on it.

The modification of HCS algorithm can be seen at Figure 4. Figure 4 is the center flowchart of HCS algorithm. There are other flowcharts in this Figure, those are Process Phase flowchart
Figure 3. HCS Flowchart in Non-weighted Graph

in Figure 5 and Sub$(H, \bar{H})$ Phase flowchart in Figure 7. The algorithm modification for weighted graphs is in the Process Phase flowchart in Figure 5 and of course we add another phase, that is the Minimum Cut Phase flowchart in Figure 6 to complete the algorithm. In the previous research [6] and [10], the previous researchers found how to get HCS from connected non-weighted graph. Since we have different object, weighted graph, so it can change the algorithm too. First, Process Phase $(G, w)$ in Figure 5, there is another process, that is Minimum Cut Phase flowchart which can be seen in Figure 6. Process Phase is used to find out the weight of mincut $w(\text{mincut})$ of graph $G$ or we can say that how many minimum cuts there are such that we get two disconnected subgraphs $H$ and $\bar{H}$. It starts by inputting graph $G$ data as weighted adjacency matrix in program (software), then do initialization of weight $w$ of mincut which is considered as $\infty$. It makes us be able to find minimum weight or minimum cut of graph $G$. Next, we make condition that the graph is not singleton (one-vertex-graph). If the graph is singleton, then the graph is not processed in graph clustering (clustering is done).

Next, if the graph is not singleton then it is continued to Minimum Cut Phase or $m = \text{Mincutphase}(G, w)$. The Minimum Cut Phase is used to find out where edges in $G$ will be cut based on the number of minimum cuts. Second, Sub$(H, \bar{H})$ Phase is used to repeat the program/algorithm for subgraph $H$ and $\bar{H}$ till we find the HCS or singleton. The purpose of this phase is getting phase-cut so that program can cut some edges and finally it becomes 2-disconnected-subgraphs. First, we choose any vertex $a$ in $G$. Then, $a$ is saved in new vertex set $A$. Since vertex set $A \neq V$ then the process will continue to look for vertex $v$ which is Most Tightly Connected Vertex (MCTV of vertex set $V - A$). Next, we get new set $A = A \cup \{v\}$. Its purpose is to look for two last vertices ($s$ and $t$) in $V$. If it is fulfilled and there is a vertex $v = t$ left, then we get phase cut = cut $(A - \{t\}, \{t\})$.

After we get the value of Minimum Cut Phase ($m$), the Figure 5 shows that given a condition when weight $w$ in Minimum Cut Phase is less than $w$ in Minimum Cut (which has been initialized before in the first time), thus weight $w$ in Minimum Cut which resulted is $w$ in Minimum Cut Phase ($w(st)$). Next, Merge process in new graph $G$ has new vertex set $V = V \setminus \{s, t\} \cup \{(st)\}$. Then, counting the total of weight $w(v, (st))$ in edges which is formed by Merge process. In this Phase, it will do looping till we have $|V| = 1$ or graph $G$ is singleton and obtain Minimum Cut of $G$.

3.4. The Application of Highly Connected Subgraphs (HCS) Algorithm for Weighted Graphs in Software Octave—401

Using Highly Connected Subgraphs (HCS) algorithm can be applied in simple graph and real life problem, for example: the case of mapping of flight routes one of airlines in Indonesia. The
3.4.1. The Application of HCS Algorithm of Simple Weighted Graph

To run the program, first we input graph data as weighted adjacency matrix and presume vertex $h$ in editor of software Octave−401. Then "call" command to run syntax of Minimum Cut and HCS. Note the Figure 8, graph $G$ has set of vertices $V = \{A, B, C, D, E, F, G\}$ which in syntax will define $A = 1, B = 2, \ldots, G = 7$. The following is an example to run the HCS algorithm is written in Octave software language version 401 in computer with spec processor AMD Dual Core E-450 APU (1.65 GHz), installed memory (RAM) 2 GB, and operating system Windows 7 Starter (32−bit).

**Figure 4.** HCS Flowchart in Weighted Graph

**Figure 5.** Process Phase
algorithm in Software Octave—401. Figure 9 is the result of running on the following Listing 1.

Listing 1. HCS Code of Graph G

```plaintext
n = 7;
g = zeros(n,n);
g(1,[2,4]) = [4 2];
g(2,3) = 4;
g(3,[4,6]) = [2 1];
g(4,[5,6]) = [2 1];
g(5,[6,7]) = [5 4];
g(6,7) = 3;

g = g + g';

(mv,mw) = mincut(g,[6 7])

(pb,qb) = HCS (g,mv,n)
```
Based on the result of the minimum-cut running \((mv, mw)\) in Figure 9, it can be seen that there is Minimum Cut, symbolized \(mw\), \(mw\) equals 4. Besides, \(mv\) has biper matrix which shows \(G\) has 7 vertices which is cut to separate edges in \(G\) into two subgraphs. They are \(pb\) subgraph which has 4 vertices and \(qb\) subgraph which has 3 vertices. Next, the command in HCS running \((pb, qb)\) will be called as in Figure 10. There are two matrices which show two different subgraphs which is cut based on its Minimum Cut.

In this research, the program result which we created by using Octave—401 is a matrix. The matrix represented by subgraph is the Highly Connected Subgraph (HCS). The first running result in matrix which has binomial number. It explains the position of the ”cut”. The ”cut” separates the graph \(G\) into two different subgraph \(H\) and \(\bar{H}\). The subgraph \(H\) in this program is stated as \(pb\) and another one is stated as \(qb\). Based on Listing 1 above, \(pb\) and \(qb\) are two subgraphs generated from cutting graph \(G\). Since graph \(G\) is not HCS, it will check subgraph \(pb\) and all subgraphs generated from cutting by using minimum cut (mincut) in subgraph \(pb\) till it generates in subgraph (HCS cluster) or singleton. Next, through the same way, it checks subgraph \(qb\). The command in this program uses mincut command and HCS command. So the result is that subgraph \(qb\) as HCS (cluster) of graph \(G\) with 7 vertices.

3.4.2. Application of Highly Connected Subgraphs (HCS) Algorithm of Flight Routes Mapping of One of Airlines in Indonesia

The application of HCS algorithm can be seen in a graph data in one of airlines in Indonesia. We use the distance of the vertices as the weight of edges in graph data. Based on Figure 11,
there are 60 vertices connected. Next we input the weighted adjacency matrix of graph data. The Listing 2 for inputting data is like Listing 1 but we change the weight as the graph data of Flight Routes.

**Listing 2. HCS Code of Flight Routes**

```matlab
clc; clear;
n = 60 ;
g = zeros(n,n);  % buat graf
p = zeros(n,n)

\[
\begin{align*}
g(1,[7,22,29]) &= [432.9 \ 1769.83 \ 2177.02]; \\
g(2,[7,10,13,16,22,31]) &= [635.59 \ 307.94 \ 681.16 \ 406.31 \ 838.77 \ 1314.42]; \\
g(3,[9,22]) &= [300.67 \ 607.89]; \\
g(4,7) &= 256.35; \\
g(5,[9,22]) &= [170.46 \ 661.62]; \\
g(6,7) &= 279.81; \\
g(7,[8,9,10,11,12,13,20,22]) &= [385.87 \ 877.12 \ 454.36 \ 223.7 \ 431.68 \ 1152.38 \ 2204.96 \ 1351.01]; \\
g(8,[10,22]) &= [144.6 \ 976.83]; \\
g(9,[13,14,16,20,22]) &= [338.01 \ 320.83 \ 149.08 \ 1334.36 \ 500.54]; \\
g(10,22) &= 896.69; \\
g(11,22) &= 1177.28; \\
g(13,22) &= 207.12; \\
g(14,[16,22]) &= [171.45 \ 453.85]; \\
g(15,22) &= 717.53; \\
g(16,22) &= 452.06; \\
g(17,[20,31]) &= [867.02 \ 577.88]; \\
g(18,31) &= 210.11; \\
g(19,[20,27,38]) &= [393.89 \ 271.41 \ 341.06];
\end{align*}
\]
```

Figure 10. Result of The \((pb,qb)\) Running
g(20,[21,22,24,25,27,28,29,31,32,33,38,40,55,59]) = [646 949.78 948.3 513.15 122.48 537.97 496.39 293.16 453.32 545.37 573.24 1463.33 2896.24 2437.42];
g(21,[24,25]) = [355.69 132.85];
g(22,[24,26,27,28,29,31,33,35,36,38,40,41,42,43,45,46,50,53,54,55,57,59]) = [1897.97 648.94 1069.5 412.18 453.41 661.94 404.88 1918.99 1768.28 1397.57 2144.21 1592.92 1241.74 921.37 901.28 722.44 2389.2 2374.26 327.85 3777.06 3709.48 3334.98];
g(23,31) = 152.66;
g(24,[25,31,32]) = [466.51 1239.31 494.99];
g(25,32) = 130.83;
g(27,[28,30,31,38]) = [658.43 121.52 409.27 478.68];
g(28,[29,31,38]) = [42.62 255.38 1019.17];
g(31,[33,38,42,43,50,55]) = [262.96 771.49 807.36 489.34 1766.68 3143.02];
g(33,[38,42,43]) = [1026.58 980.73 645.57];
g(34,38) = 350.59;
g(35,38) = 686.36;
g(36,38) = 407.66;
g(37,38) = 669.59;
g(38,[39,40,41,42,50,53,54,55,56,58,59,60]) = [338.23 897.85 558.17 529.02 996.55 1042.66 1890.14 2380.13 1717.13 1429.91 1938.25 326.07];
g(40,[42,53,55,58,59]) = [915.73 302.83 1790.95 726.11 1449.59];
g(42,[43,44,45,46,48]) = [335.32 338.35 345.08 842.44 452.03];
g(45,46) = 554.75;
g(46,[47,49]) = [399.17 238.37];
g(50,[51,52,53,58]) = [550.62 648.79 345.68 479.95];
g(54,55) = 532.93;
g(55,[56,57,58,59,60]) = [767.44 667.81 1065.54 474.94];
g(56,58) = 307.63;
g(58,59) = 764.28;

\[ g = g + g^\prime; \]
\[ (mv,mw) = \text{mincut}(g,[\ldots]) \]
\[ (pb,qb) = \text{HCS}(g,mv,n); \]

Based on Running Code in Listing 2 above, we get HCS cluster consisting 10 vertices (airports) in 10 cities. They are Denpasar, Jakarta, Surabaya, Makassar, Manado, Balikpapan, Palangkaraya, Pontianak, Jayapura, dan Timika. Figure 12 represents HCS cluster as a graph. There are some obstacles such that the graph data just gets a cluster. They are difference of degree in \( G \) and difference of edge weight in \( G \). The difference of degree means that there is vertex which has a degree consisting one and there is also vertex which has degree consisting tens. It is like the difference of degree, the difference of edge weight also looks significant. It can be seen through Running Code in Listing 2, stating that weight of edge has a range from hundreds kilometer up to four a hundred and more kilometers up to four hundred kilometers. Due to the great difference between degree and edge weight, then it makes the graph \( G \) data get unbalance and it is not suitable for this HCS Algorithm. We can see that when the program do cutting on graph \( G \), it results subgraph \( H \) and \( \bar{H} \). The subgraph \( H \) is the biggest one. Meanwhile, the subgraph \( \bar{H} \) is only singleton. Thus, the cluster result is only one-big-subgraph identified as HCS. Although the prediction results some subgraphs of HCS.
4. Conclusion

We have discussed about modification of Highly Connected Subgraphs (HCS) algorithm for weighted graph and get conclusion such as:

(i) Modification of Highly Connected Subgraphs (HCS) algorithm for weighted graph can be done by adding variable $h$ on algorithm to get two disconnected subgraphs by deleting some edges which have minimum weight ($\text{minimum cut}$).

(ii) Flight path data with edge weight as mileage is not recommended for implementation of Highly Connected Subgraphs (HCS) algorithm. The result is just has a cluster because there are significant differences. They are the difference of degree and the difference of edge weight.

Open Problem 1 Highly Connected Subgraphs (HCS) algorithm for weighted graph by using another data, for example: protein data.
Open Problem 2  Modification of Highly Connected Subgraphs (HCS) algorithm for other kinds of graphs.

Acknowledgments
This work was partially supported by the Mathematics Department, Universitas Indonesia 2018

References
[1] Bapat R B 2010 Graphs and Matrices New Delhi: Hindustan Book Agency
[2] Deo N 1990 Graph Theory with Application to Engineering and Computer Science New Delhi: Prentice Hall of India Private Limited
[3] Dafik, Miller M, Iliopoulos C, and Ryjacek Z 2009 On Diregularity of Digraphs of Defect at Most Two Journal of Combinatorial Mathematics and Combinatorial Computing 71 pp. 21-37
[4] Flight Domestic Route of Garuda Indonesia 2014 be quote April 15, 2016 from https://www.garuda-indonesia.com/id/id/destination/route-map/index-domestic.page
[5] Gross J L dan Yellen J 2006 Graph Theory and Its Application Boca Raton: Taylor & Francis Group
[6] Hartuv E and Shamir R 2000 A clustering algorithm based on graph connectivity Information Processing Letters 76(4-6) pp. 175-181
[7] Przulj N 2005 Knowledge Discovery in Proteomics: Graph Theory Analysis of Protein-Protein Interactions Toronto: CRC Press
[8] Ruohonen K 2013 Graph Theory Kanada: University of British Columbia
[9] Schaeffer S E 2007 Graph Clustering Journal of Computer Science Vol 1 pp. 27-64
[10] Stoer M F W 1997 A Simple Min Cut Algorithm Journal of ACM 44(4) pp. 585-591
[11] Sugeng K A, Slamet S, and Silaban D R 2014 Teori Graf dan Aplikasinya Textbook: Mathematics Department FMIPA Universitas Indonesia