Sciencia Acta Xaveriana
An International Science Journal
ISSN. 0976-1152

Vol. 8 No. 2
pp. 33-41
September 2017

ALGORITHMS TO FIND CLIQUE-TO-VERTEX DETOUR DISTANCE IN GRAPHS

I. Keerthi Asir[1*], S. Athisayanathan[2] and A. Antonysamy S.J.[3]

[1,2,3]Department of Mathematics, St. Xavier's College (Autonomous),
Palayamkottai - 627 002, Tamil Nadu, India.

E-mail: asirsxc@gmail.com
E-mail: athisxc@gmail.com
E-mail: antonysamy.fr@gmail.com

Received : May 2017 *Accepted : August* 2017

Abstract. Let C be a clique and j a vertex in a connected graph G. A clique-to-vertex $C-j$ path P is an $i-j$ path, where i is a vertex in C such that P contains no vertices of C other than i. The clique-to-vertex detour distance, $D(C,j)$ is the length of a longest $C-j$ path in G. The clique-to-vertex detour eccentricity $e_{D2}(C)$ of a clique C in G is the maximum clique-to-vertex detour distance from C to a vertex $i \in V$ in G. The clique-to-vertex detour radius R_2 of G is the minimum clique-to-vertex detour eccentricity among the cliques of ζ in G, where ζ is the set of all cliques in G, while the clique-to-vertex detour diameter D_2 of G is the maximum clique-to-vertex detour eccentricity among the cliques of ζ in G. The clique-to-vertex detour center of G is the set of all cliques having minimum clique-to-vertex detour eccentricity of G and the clique-to-vertex detour periphery of G is the set of all cliques having maximum clique-to-vertex detour eccentricity of G. It is given that the algorithms to find the clique-to-vertex detour distance $D(C,j)$, the clique-to-vertex detour eccentricity $e_{D2}(C)$, the clique-to-vertex detour radius R_2, the clique-to-vertex detour diameter D_2, the clique-to-vertex detour center $C_{D2}(G)$, and the clique-to-vertex detour periphery $P_{D2}(G)$ of a graph G using BC representation.

Keywords: Binary count, clique-to-vertex detour distance, clique-to-vertex detour center, clique-to-vertex detour periphery.

2010 Mathematics Subject Classification: 05C12

1. Introduction

By a graph $G = (V, E)$, we mean a finite undirected connected simple graph. For basic graph theoretic terminologies, we refer [3, 4]. If $X \subseteq V$, then $\langle X \rangle$ is the subgraph induced by X. A clique C of a graph G is a maximal complete subgraph, denoted by its vertices. In 1964, Hakimi [5] considered the facility location problems as vertex-to-vertex distance in graphs. For any two vertices u and v in a connected graph G, the distance $d(u, v)$ is the length of a shortest $u - v$ path in G. A $u - v$ path of length $d(u, v)$ is called a $u - v$ geodesic in G. Also they defined the eccentricity $e(v)$ of a vertex v, the radius r, diameter d, the center $C(G)$, and the periphery $P(G)$. The distance matrix $D(G) = [d_{ij}]$ of G is a $n \times n$ matrix, where n is the order of G, and $[d_{ij}] = d(v_i, v_j)$, the distance between v_i and v_j in $G(1 \leq i \leq n, 1 \leq j \leq n)$.

In 2005, Chartrand, Escuadro and Zhang [2] introduced and studied the concepts of detour distance in graphs. For any two vertices u and v in a connected graph G, the detour distance $D(u, v)$ is the length of a longest $u - v$ path in G. A $u - v$ path of length $D(u, v)$ is called a $u - v$ detour in G. Also they defined the detour eccentricity $e_D(v)$ of a vertex v, the detour radius R, detour diameter D, the detour center $C_D(G)$, and the detour periphery $P_D(G)$. The detour distance matrix $D(G) = [D_{ij}]$ of G is a $n \times n$ matrix, where n is the order of G, and $[D_{ij}] = D(v_i, v_j)$, the detour distance between v_i and v_j in $G(1 \leq i \leq n, 1 \leq j \leq n)$.

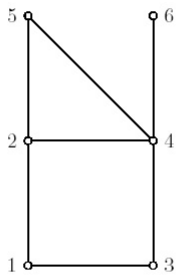

Fig 1.1: G

Ashok kumar, Athisayanathan and Antonysamy [1] introduced the algorithms to find clique-to-vertex structures in a graph using BC-representation. Correspondingly they defined a method to represent a subset of a set which is called binary count (or BC) representation. For example, the graph G given in Fig. 1.1, the set of all cliques in G is $\zeta = \{\{1, 2\}, \{1, 3\}, \{2, 4, 5\}, \{3, 4\}, \{4, 6\}\}$ and the set ζ of all cliques in G in BC representation is $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$. Note that if C is the clique $\{3, 4\}$, then the BC representation of C is $BC(C) = (001100)$, and further $BC(C(1)) = BC(C(2)) = BC(C(5)) = BC(C(6)) = 0$, and $BC(C(3)) = BC(C(4)) = 1$. That is, $BC(C(i))(1 \leq i \leq n)$ denotes the integer 1 or 0 in the i^{th} place in the BC representation of the clique C in the graph G. For our convenience, we define if $C = (010110)$ then $|C| = 0 + 1 + 0 + 1 + 1 + 0 = 3$. Also the detour distance matrix $D(G)$ of G is

$$D(G) = \begin{bmatrix} 0 & 4 & 4 & 3 & 4 & 4 \\ 4 & 0 & 3 & 3 & 4 & 4 \\ 4 & 3 & 0 & 4 & 4 & 5 \\ 3 & 3 & 4 & 0 & 4 & 1 \\ 4 & 4 & 4 & 4 & 0 & 5 \\ 4 & 4 & 5 & 1 & 5 & 0 \end{bmatrix}$$

Keerthi Asir and Athisayanathan [6] introduced and studied the concepts of clique-to-vertex detour distance in graphs. In this paper we introduce and study algorithms to find the clique-to-vertex detour distance $D(C, j)$, the clique-to-vertex detour eccentricity $e_{D2}(C)$, the clique-to-vertex detour radius R_2, the clique-to-vertex detour diameter D_2, the clique-to-vertex detour center $C_{D2}(G)$ and the clique-to-vertex detour periphery $P_{D2}(G)$ of a graph G using BC representation. Throughout this paper, G denotes a connected graph with at least two vertices.

2. Clique-to-Vertex Detour Distance

First, we introduce an algorithm to find the clique-to-vertex detour distance $D(C, j)$ between a clique C and a vertex j in a graph G using BC representation.

Definition 2.1. *Let C be a clique and j a vertex in a connected graph G. A clique-to-vertex $C - j$ path P is a $i - j$ path, where i is a vertex in C such that P contains no vertices of C other than i. The clique-to-vertex detour distance $D(C, j)$ is the length of a longest $C - j$ path in G.*

In particular it may be a vertex, say $x \in C$ such that the $x - j$ path is unique and contains no vertices of C other than x.

Algorithm 2.2. *Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C : C$ is a clique inBC representation$\}$.*

(1) *Let $D(G) = [D_{ij}]$ be the detour distance matrix of G.*
(2) *Let $C \in \zeta$*
(3) *Let $j \in V$*
(4) *If $BC(C(j)) = 1$ then $D(C, j) = 0$; goto step (10)*
(5) *For $i = 1$ to n*
(6) *If $BC(C(i)) = 0$ then $D(i, j) = 0$*
(7) *If $BC(C(i)) = 1$ then $D(i, j) = D_{ij}$*
(8) *Next i*
(9) *Find $D(C, j)$*
  - $D(C, j) = \max\{D(i, j) : 1 \leq i \leq n, i \neq x\} - |C| + 1$
  - $D(C, j) = D(x, j)$

(10) *Return $D(C, j)$*
(11) *Stop*

Theorem 2.3. *For every clique C and a vertex j in a connected graph G, the Algorithm 2.2 finds the clique-to-vertex detour distance $D(C, j)$.*

Proof. Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$, $\zeta = \{C : C$ is a clique in BC representation$\}$ and $D(G)$ the detour distance matrix of G. Let $C \in \zeta$ and $j \in V$. We consider the following two cases:
Case 1. If $j \in C$ then $BC(C(j)) = 1$ so that the clique-to-vertex detour distance $D(C, j) = 0$.
Case 2. If $j \notin C$ then $BC(C(j)) = 0$ so that the steps (5) to (8) of the Algorithm 2.2 finds the detour distance $D(i, j)$ from the vertex $i(1 \leq i \leq n)$ to the vertices j as follows.
Subcase 1 of Case 2. If $i \notin C$ then $BC(C(i)) = 0(1 \leq i \leq n)$ so that the detour distance $D(i, j) = 0$.
Subcase 2 of Case 2. If $i \in C$ then $BC(C(i)) = 1(1 \leq i \leq n)$ so that the detour distance $D(i, j) = D_{ij}$.
Then step (9) of the Algorithm 2.2 finds the clique-to-vertex detour distance $D(C, j)$ by either $D(C, j) = \max\{D(i, j) : 1 \leq i \leq n, i \neq x\} - |C| + 1$ or $D(C, j) = D(x, j)$, where x is a vertex in C such that the $x - j$ path is unique and contains no vertices of C other than x. □

In the Algorithm 2.2, the step (4) is executed in $O(1)$ time, the steps (5) to (8) are executed in $O(n)$ time, and the step (9) is executed in $O(n)$ time, we have the following theorem.

Theorem 2.4. *The clique-to-vertex detour distance $D(C, j)$ between the clique C and the vertex i in a graph G can be found in $O(n)$ time.*

Example 2.5. *Consider the graph G given in Fig. 1.1, the set ζ of all cliques in G in BC representation is $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$. Let $D(G)$ be the detour distance matrix of G. Now using Algorithm 2.2, let us find the clique-to-vertex detour distance $D(C, j)$ between the clique $C = \{1, 2\}$ and the vertex $j = 1$. Clearly $BC(C) = (110000)$. Since $BC(C(j)) = 1$, the Algorithm 2.2 returns clique-to-vertex detour distance $D(C, j) = 0$. Again using the Algorithm 2.2, let us find the clique-to-vertex detour distance $D(C, j)$ between the clique $C = \{1, 2\}$ and the vertex $j = 6$. Sine $BC(C(j)) = 0$ and there is no vertex x in C such that the $x - j$ path is unique and contains no vertices of C other than x. Then the Algorithm 2.2 finds the clique-to-vertex detour distance $D(C, j) = \max\{D(i, j) : 1 \leq i \leq n, i \neq x\} - |C| + 1$. For the vertices $i = 3, 4, 5, 6$, $BC(C(i)) = 0$ and also for the vertices $i = 1, 2$, $BC(C(i)) = 1$ so that $D(1, j) = D_{1j} = 4$ and $D(2, j) = D_{2j} = 4$. Now the Algorithm 2.2 returns the clique-to-vertex detour distance $D(C, j) = \max\{D(i, j) : 1 \leq i \leq n, i \neq x\} - |C| + 1 = \max\{D(1, j), D(2, j), D(3, j), D(4, j), D(5, j), D(6, j)\} - |C| + 1 = \max\{4, 4, 0, 0, 0, 0\} - |C| + 1 = 4 - 2 + 1 = 3$. Also for the clique $C = \{2, 4, 5\}$ and the vertex $j = 6$, $BC(C(j)) = 0$. Here there is a vertex 4 in C such that the $4 - j$ path is unique and contains no vertices of C other than 4. Then the Algorithm 2.2 finds the clique-to-vertex detour distance $D(C, j) = D(4, j) = D_{4j} = 1$.*

3. Clique-To-Vertex Detour Eccentricity

Next, we introduce an algorithm to find the clique-to-vertex detour eccentricity $e_{D2}(C)$ of a clique C in a graph G using BC representation.

Definition 3.1. *The clique-to-vertex detour eccentricity $e_{D2}(C)$ of a clique C in a connected graph G is defined as $e_{D2}(C) = \max\{D(C, j) : j \in V\}$.*

Algorithm 3.2. *Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C : C$ is a clique in BC representation$\}$.*

(1) *Let $C \in \zeta$.*
(2) *Let $j \in V$*
(3) *For $j = 1$ to n*
(4) *Find $D(C, j)$, (By Calling Algorithm 2.2)*
(5) *Next j*
(6) *Find $e_{D2}(C) = \max\{D(C, j) : 1 \leq j \leq n\}$*
(7) *Return $e_{D2}(C)$*
(8) *Stop*

Theorem 3.3. *For every clique C and the set of all vertices V in a connected graph G, the Algorithm 3.2 finds the clique-to-vertex detour eccentricity $e_{D2}(C)$.*

Proof. Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C_1, C_2, \ldots, C_m\}$ the set of all cliques in BC representation in G. Let $i \in V$. Then the step (4) of the Algorithm 3.2 finds the clique-to-vertex detour distance $D(C, j)$ between the clique C and every vertex $j(1 \leq j \leq n)$ in G, and the step (6) of the Algorithm 3.2 finds the clique-to-vertex detour eccentricity $e_{D2}(C)$ by $e_{D2}(C) = \max\{D(C, j) : 1 \leq j \leq n\}$. □

In the Algorithm 3.2, the step (4) is executed in $O(n)$ time, the steps (3) to (5) are executed in $O(n^2)$ time, and the step (6) is executed in $O(n)$ time, we have the following theorem.

Theorem 3.4. *The clique-to-vertex detour eccentricity $e_{D2}(C)$ of a clique C in a graph G can be found in $O(n^2)$ time.*

Example 3.5. *For the graph G given in Fig. 1.1, the set ζ of all cliques in G in BC representation is $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$. Let $C = (110000) \in \zeta$. Now using Algorithm 3.2, we find the clique-to-vertex detour eccentricity $e_{D2}(C)$. By calling the algorithm 2.2 n times, the step (4) of Algorithm 3.2 finds the clique-to-vertex detour distances $D(C, 1) = 0$, $D(C, 2) = 0$, $D(C, 3) = 3$, $D(C, 4) = 2$, $D(C, 5) = 3$ and $D(C, 6) = 3$. Finally the step (6) of Algorithm 3.2 finds the clique-to-vertex detour eccentricity $e_{D2}(C) = \max\{0, 0, 3, 2, 3, 3\} = 3$.*

4. Clique-To-Vertex Detour Radius

Next, we introduce an algorithm to find the clique-to-vertex detour radius R_2 of a graph G using BC representation.

Definition 4.1. *The clique-to-vertex detour radius R_2 of a connected graph G is defined as, $R_2 = rad_{D2}(G) = \min\{e_{D2}(C) : C \in \zeta\}$.*

Algorithm 4.2. *Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C : C$ is a clique in BC representation$\}$.*

(1) *Let $\zeta = \{C_1, C_2, \ldots, C_m\}$*
(2) *For $i = 1$ to m*

(3) *Find* $e_{D2}(C_i)$*, (By Calling Algorithm 3.2)*
(4) *Next* i
(5) *Find* $R_2 = \min\{e_{D2}(C_i) : 1 \leq i \leq m\}$
(6) *Return* R_2
(7) *Stop*

Theorem 4.3. *For a connected graph* G*, the Algorithm 4.2 finds the clique-to-vertex detour radius* R_2 *of* G*.*

Proof. Let G be a non-trivial connected graph with $\zeta = \{C_1, C_2, \ldots, C_m\}$ and $C \in \zeta$. Then the steps (2) to (4) of the Algorithm 4.2 finds the clique-to-vertex detour eccentricity $e_{D2}(C_i)$ for every clique C_i, and the step (5) of the Algorithm 4.2 finds the clique-to-vertex detour radius R_2 of G by $R_2 = \min\{e_{D2}(C_i) : 1 \leq i \leq m\}$. □

In the Algorithm 4.2, the step (3) is executed in $O(n^2)$ time, the steps (2) to (4) are executed in $O(mn^2)$ time, and the step (5) is executed in $O(m)$ time, we have the following theorem.

Theorem 4.4. *The clique-to-vertex detour radius* R_2 *of* G *can be found in* $O(mn^2)$ *time.*

Example 4.5. *For the graph* G *given in Fig. 1.1, the set* ζ *of all cliques in* G *in BC representation is* $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$*. Now using Algorithm 4.2, we find the clique-to-vertex detour radius* R_2 *of* G*. By calling the algorithm 3.2* m *times, the step (3) of Algorithm 4.2 finds the clique-to-vertex detour eccentricities* $e_{D2}(C_1) = 3$*,* $e_{D2}(C_2) = 4$*,* $e_{D2}(C_3) = 2$*,* $e_{D2}(C_4) = 3$ *and* $e_{D2}(C_5) = 4$*. Finally step (5) of Algorithm 4.2 finds the clique-to-vertex detour radius* $R_2 = \min\{3, 4, 2, 3, 4\} = 2$*.*

5. Clique-To-Vertex Detour Center

Next, we introduce an algorithm to find the clique-to-vertex detour center $C_{D2}(G)$ of a graph G using BC representation.

Definition 5.1. *Let* G *be a connected graph. A clique* C *in* G *is called a clique-to-vertex detour central clique if* $e_{D2}(C) = R_2$ *and the clique-to-vertex detour center* $C_{D2}(G)$ *of* G *is defined as,* $C_{D2}(G) = Cen_{D2}(G) = \langle\{C \in \zeta : e_{D2}(C) = R_2\}\rangle$*.*

Algorithm 5.2. *Let* G *be a non-trivial connected graph with* $V = \{1, 2, 3, ..., n\}$ *and* $\zeta = \{C : C$ *is a clique in BC representation*$\}$*.*

(1) *Let* $\zeta = \{C_1, C_2, ..., C_m\}$*.*
(2) *Let* $C_{D2}(G) = \langle\phi\rangle$
(3) *For* $i = 1$ *to* m
(4) *Find* $e_{D2}(C_i)$*, (By Calling Algorithm 3.2)*
(5) *Next* i
(6) *Find* R_2*, (By Calling Algorithm 4.2)*
(7) *For* $i = 1$ *to* m
(8) *If* $e_{D2}(C_i) = R_2$ *then* $C_{D2}(G) = C_{D2}(G) \cup \{C_i\}$
(9) *Next* i

(10) *Stop*

Theorem 5.3. *For a connected graph G, the Algorithm 5.2 finds the clique-to-vertex detour center C_{D2} of G.*

Proof. Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C_1, C_2, \ldots, C_m\}$ the set of all cliques in BC representation in G. Then the steps (3) to (5) of the Algorithm 5.2 finds the clique-to-vertex detour eccentricity $e_{D2}(C_i)$ for every clique $C_i \in \zeta (1 \leq i \leq m)$, the step (6) of the Algorithm 5.2 finds the clique-to-vertex detour radius R_2 of G by $R_2 = \min\{e_{D2}(C_i) : 1 \leq i \leq m\}$, and the steps (7) to (9) of the Algorithm 5.2 finds the clique-to-vertex detour center $C_{D2}(G)$ of G by $C_{D2}(G) = Cen_{D2}(G) = \langle\{C_i \in \zeta : e_{D2}(C_i) = R_2\}\rangle$. □

In the Algorithm 5.2, the step (4) is executed in $O(n^2)$ time, the steps (3) to (5) are executed in $O(mn^2)$ time, the step (6) is executed in $O(m)$ time, and the steps (7) to (9) are executed in $O(m)$ time, we have the following theorem.

Theorem 5.4. *The clique-to-vertex detour center $C_{D2}(G)$ of G can be found in $O(mn^2)$ time.*

Example 5.5. *For the graph G given in Fig. 1.1, the set ζ of all cliques in G in BC representation is $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$. Now using Algorithm 5.2, we find the clique-to-vertex detour center $C_{D2}(G)$. By calling the algorithm 3.2 m times, the step (4) of Algorithm 5.2 finds the clique-to-vertex detour eccentricities $e_{D2}(C_1) = 3$, $e_{D2}(C_2) = 4$, $e_{D2}(C_3) = 2$, $e_{D2}(C_4) = 3$ and $e_{D2}(C_5) = 4$. By Calling Algorithm 4.2 m times, step (6) of Algorithm 5.2 finds the clique-to-vertex detour radius $R_2 = \min\{3, 4, 2, 3, 4\} = 2$. Finally step (8) of Algorithm 5.2 finds the clique-to-vertex detour center*
$C_{D2}(G) = \langle\{C \in \zeta : e_{D2}(C) = R_2\}\rangle = \langle\{C_3\}\rangle$.

6. Clique-To-Vertex Detour Diameter

Next, we introduce an algorithm to find the clique-to-vertex detour diameter D_2 of a graph G using BC representation.

Definition 6.1. *The clique-to-vertex detour diameter D_2 of a connected graph G is defined as,*
$D_2 = diam_{D2}(G) = \max\{e_{D2}(C) : C \in \zeta\}$.

Algorithm 6.2. *Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C : C \text{ is a clique in BC representation}\}$.*

(1) *Let* $\zeta = \{C_1, C_2, \ldots, C_m\}$
(2) *For* $i = 1$ *to* m
(3) *Find* $e_{D2}(C_i)$, *(By Calling Algorithm 3.2)*
(4) *Next* i
(5) *Find* $D_2 = \max\{e_{D2}(C_i) : 1 \leq i \leq m\}$
(6) *Return* R_2
(7) *Stop*

Theorem 6.3. *For a connected graph G, the Algorithm 6.2 finds the clique-to-vertex detour diameter D_2 of G.*

Proof. Let G be a non-trivial connected graph with $\zeta = \{C_1, C_2, \ldots, C_m\}$ and $C \in \zeta$. Then the steps (2) to (4) of the Algorithm 6.2 finds the clique-to-vertex detour eccentricity $e_{D2}(C_i)$ for every clique C_i, and the step (5) of the Algorithm 6.2 finds the clique-to-vertex detour diameter D_2 of G by $D_2 = \max\{e_{D2}(C_i) : 1 \leq i \leq m\}$. □

In the Algorithm 6.2, the step (3) is executed in $O(n^2)$ time, the steps (2) to (4) are executed in $O(mn^2)$ time, and the step (5) is executed in $O(m)$ time, we have the following theorem.

Theorem 6.4. *The clique-to-vertex detour diameter D_2 of G can be found in $O(mn^2)$ time.*

Example 6.5. *For the graph G given in Fig. 1.1, the set ζ of all cliques in G in BC representation is $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$. Now using Algorithm 6.2, we find the clique-to-vertex detour diameter D_2 of G. By calling the algorithm 3.2 m times, the step (3) of Algorithm 6.2 finds the clique-to-vertex detour eccentricities $e_{D2}(C_1) = 3$, $e_{D2}(C_2) = 4$, $e_{D2}(C_3) = 2$, $e_{D2}(C_4) = 3$ and $e_{D2}(C_5) = 4$. Finally step (5) of Algorithm 6.2 finds the clique-to-vertex detour diameter $D_2 = \max\{3, 4, 2, 3, 4\} = 4$.*

7. Clique-To-Vertex Detour Periphery

Next, we introduce an algorithm to find the clique-to-vertex detour periphery $P_{D2}(G)$ of a graph G using BC representation.

Definition 7.1. *Let G be a connected graph. A clique C in G is called a clique-to-vertex detour peripheral clique if $e_{D2}(C) = D_2$ and the clique-to-vertex detour periphery $P_{D2}(G)$ of G is defined as, $P_{D2}(G) = Per_{D2}(G) = \langle\{C \in \zeta : e_{D2}(C) = D_2\}\rangle$.*

Algorithm 7.2. *Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C : C \text{ is a clique in BC representation}\}$.*

(1) *Let $\zeta = \{C_1, C_2, ..., C_m\}$.*
(2) *Let $P_{D2}(G) = \langle\phi\rangle$*
(3) *For $i = 1$ to m*
(4) *Find $e_{D2}(C_i)$, (By Calling Algorithm 3.2)*
(5) *Next i*
(6) *Find D_2, (By Calling Algorithm 6.2)*
(7) *For $i = 1$ to m*
(8) *If $e_{D2}(C_i) = D_2$ then $P_{D2}(G) = P_{D2}(G) \cup \{C_i\}$*
(9) *Next i*
(10) *Stop*

Theorem 7.3. *For a connected graph G, the Algorithm 7.2 finds the clique-to-vertex detour periphery $P_{D2}(G)$ of G.*

Proof. Let G be a non-trivial connected graph with $V = \{1, 2, 3, ..., n\}$ and $\zeta = \{C_1, C_2, \ldots, C_m\}$ the set of all cliques in BC representation in G. Then the steps (3) to (5) of the Algorithm 7.2 finds the clique-to-vertex detour eccentricity $e_{D2}(C_i)$ for every clique $C_i \in \zeta (1 \leq i \leq m)$, the step (6) of the Algorithm 7.2 finds the clique-to-vertex detour diameter D_2 of G by $D_2 = \max\{e_{D2}(C_i) : 1 \leq i \leq m\}$, and the steps (7) to (9) of the Algorithm 7.2 finds the clique-to-vertex detour periphery $P_{D2}(G)$ of G by $P_{D2}(G) = Per_{D2}(G) = \langle\{C_i \in \zeta : e_{D2}(C_i) = D_2\}\rangle$. □

In the Algorithm 7.2, the step (4) is executed in $O(n^2)$ time, the steps (3) to (5) are executed in $O(mn^2)$ time, the step (6) is executed in $O(m)$ time, and the steps (7) to (9) are executed in $O(m)$ time, we have the following theorem.

Theorem 7.4. *The clique-to-vertex detour periphery $P_{D2}(G)$ of G can be found in $O(mn^2)$ time.*

Example 7.5. *For the graph G given in Fig. 1.1, the set ζ of all cliques in G in BC representation is $\zeta = \{(110000), (101000), (010110), (001100), (000101)\}$. Now using Algorithm 5.2, we find the clique-to-vertex detour periphery $P_{D2}(G)$. By calling the algorithm 3.2 m times, the step (4) of Algorithm 7.2 finds the clique-to-vertex detour eccentricities $e_{D2}(C_1) = 3$, $e_{D2}(C_2) = 4$, $e_{D2}(C_3) = 2$, $e_{D2}(C_4) = 3$ and $e_{D2}(C_5) = 4$. By calling the algorithm 6.2 m times, step (6) of Algorithm 7.2 finds the clique-to-vertex detour diameter $D_2 = \max\{3, 4, 2, 3, 4\} = 4$. Finally step (8) of Algorithm 7.2 finds the clique-to-vertex detour periphery $P_{D2}(G) = \langle\{C \in \zeta : e_{D2}(C) = D_2\}\rangle = \langle\{C_2, C_5\}\rangle$.*

ACKNOWLEDGMENTS

The author wish to thank God.

REFERENCES

[1] A. Ashok Kumar, S. Athisayanathan and A. Antonysamy, *Algorithms to find Clique-To-Vertex Structures in a Graph using BC-representation*, Int. J. Research in Computer Applications and Robotics, **1**, (2013), 1-5.

[2] G. Chartrand, H. Escuadro, and P. Zhang, *Detour Distance in Graphs*, J.Combin.Math.Combin.Comput., **53**, (2005), 75-94.

[3] G. Chartrand and Ortrud R. Oellermann, *Applied and Algorithmic Graph Theory*, McGrraw-Hill International Editions (1993).

[4] G. Chartrand and P. Zhang, *Introduction to Graph Theory*, Tata McGraw-Hill, New Delhi, (2006).

[5] S. L. Hakimi, *Optimum location of switching centers and absolute centers and medians of a graph*, Operations Research, **12**, (1964).

[6] I. Keerthi Asir and S. Athisayanathan, *Clique-To-Vertex Detour Distance in Graphs*, Journal of Prime Research In Mathematics, **11**, (2015), 42-54.