ON HAMILTONIAN QUASI-THRESHOLD GRAPHS

S.D. Nikolopoulos

34 – 2001

Preprint, no 34 – 01 / 2001

Department of Computer Science
University of Ioannina
45110 Ioannina, Greece

On Hamiltonian Quasi-threshold Graphs

Stavros D. Nikolopoulos

Department of Computer Science, University of Ioannina,
P.O. Box 1186, GR-45110 Ioannina, Greece
email: stavros@cs.uoi.gr

Abstract — In this paper we prove structural and algorithmic properties on the class of quasi-threshold graphs, or QT-graphs for short, and show that a QT-graph G has a unique tree representation; that is, a tree structure, we call it *cent-tree*, which meets the structural properties of G. Based on the structure of the cent-tree, we prove necessary and sufficient conditions for a QT-graph to be Hamiltonian. By taking advantage of these properties and conditions, we construct linear-time algorithms for finding a Hamiltonian cycle and computing the Hamiltonian completion number and the Hamiltonian completion edge set in a QT-graph; our algorithms take $O(n + m)$ time on graphs with n vertices and m edges. This implies linear-time recognition algorithms for this class of graphs. We also present a linear-time algorithm for constructing the cent-tree of a QT-graph.

Keywords: Quasi-threshold graphs, tree representation, Hamiltonian cycles, Hamiltonian completion number, complexity, linear algorithms.

1. Introduction

We consider finite undirected graphs with no loops nor multiple edges. Let G be such a graph with vertex set $V(G)$ and edge set $E(G)$. We say that G is a Hamiltonian graph if it has a spanning cycle (as opposed to the more usual definition which refers to spanning path); such a cycle is called a Hamiltonian cycle. The *Hamiltonian completion number* of G is the minimum number of edges which need to be added to $E(G)$ to make G Hamiltonian; the set of such edges is called *Hamiltonian completion edge set* of G [2, 12]. We denote the Hamiltonian completion number of a graph G as $hcn(G)$ and its Hamiltonian completion edge set as $CE(G)$. If G is a Hamiltonian graph, then $hcn(G) = 0$.

Given a graph G, an edge $(x, y) = (y, x)$ of G can be classified as follows according to the relationship of closed neighbourhoods [13, 18]: (x, y) is *free* if $N[x] = N[y]$; (x, y) is *semi-free* if $N[x] \subset N[y]$ (or $N[y] \subset N[x]$); and (x, y) is *actual* otherwise. Obviously, $E(G)$ can be partitioned into the three subsets of free edges, semi-free edges and of actual edges, respectively.

A graph G is called a *quasi-threshold graph*, or QT-graph for short, if every edge of G is either free or semi-free. Thus G is a QT-graph if and only if for every edge (x, y) of G, we have $N[x] \subseteq N[y]$ or $N[x] \supseteq N[y]$; equivalently, G is a QT-graph if and only if G has no induced subgraph isomorphic to P_4 or C_4 [3, 11, 16]. The class of QT-graphs is a subclass of the class of cographs [6, 8] and contains the class of threshold graphs [7].

Many researchers have devoted their work to the study of QT-graphs. Wolk [21] called these graphs *comparability graphs of trees* and gave characterization of them. Golumbic [10] called them *trivially* *perfect* graphs in respect to a concept of "perfection". Ma, Wallis and Wu. [15] called them *quasi-threshold* graph (*QT*-graphs) and studied algorithmic properties.

The class of *QT*-graphs is a subclass of the well-known class of perfect graphs [3, 11, 16]; it is a very important class of graphs, since a number of problems, which are NP-complete in general, can be solved in polynomial time on its members. For the class of *QT*-graphs, Ma et. al. [15] presented polynomial algorithms for a number of optimization problems. In particular, they gave an $O(nm)$ time algorithm for the recognition problem, and polynomial algorithms for the Hamiltonian cycle problem and the bandwidth problem. They also gave a formula for the clique covering number and conditions for a *QT*-graph to be Hamiltonian. Yan et. al [24] stated important characterizations of these graphs and presented a linear-time algorithm, that is, $O(n + m)$, for the recognition problem. They also gave linear-time algorithms for the edge domination problem and the bandwidth problem in this class of graphs.

In this paper we study the class of *QT*-graphs in farther details and provide structural and algorithmic properties on its members. We show that a *QT*-graph G has a unique tree representation; that is, a tree structure which meets the structural properties of G; we refer to this tree as *cent-tree* of the graph G and denote it by $T_c(G)$. Based on the structure of the cent-tree, we prove necessary and sufficient conditions for a *QT*-graph to be Hamiltonian. From these properties and conditions, we first design a linear-time algorithm for constructing the cent-tree representation of a *QT*-graph G and, then, we construct linear-time algorithms for finding a Hamiltonian cycle and computing the Hamiltonian completion number of G; our algorithms take $O(n + m)$ time on graphs with n vertices and m edges. Note that the linear-time algorithm for the Hamiltonian completion number is also a recognition algorithm. We also show that the Hamiltonian completion edge set of a *QT*-graph can be computed in linear time and propose an $O(n + m)$-time algorithm.

We should point out that, to the best of my knowledge, the study of the Hamiltonian completion number on *QT*-graphs has not received much attention. On the other hand, this problem on other classes of graphs has been extensively studied (see, [1, 9, 14, 19]).

The paper is organized as follows. In Section 2 we characterize the class of *QT*-graphs in details and show that a *QT*-graph has a unique tree representation; that is, the cent-tree. In Section 3 we give necessary and sufficient conditions for a *QT*-graph to be Hamiltonian. In Section 4 we present a linear-time algorithm for constructing the cent-tree of a *QT*-graph. Based on the structural properties of the cent-tree and the conditions of Section 3, we present the main results of the paper in Section 5; we design and analyze linear-time algorithms for finding a Hamiltonian cycle and computing the Hamiltonian completion number and the Hamiltonian completion edge set of a *QT*-graph. Finally, in Section 6 we conclude with a summary of our results and extensions.

2. Quasi-threshold Graphs and their Structures

Let G be a graph with vertex set $V(G)$ and edge set $E(G)$. The neighbourhood of a vertex x is the set $N(x) = N_G(x)$ consisting of all the vertices of G which are adjacent with x. The closed neighbourhood of x is defined by $N[x] = N_G[x] := \{x\} \cup N(x)$. The subgraph of a graph G induced by a subset $S \subseteq V(G)$ is denoted by $G[S]$. Let X and Y be two subsets of a certain set. Then $X \subset Y$ means that X is a proper subset of Y, and if $Y \subseteq X$, then let X - Y denote $X \setminus Y$.

For a vertex subset S of a graph G, we define $G - S$ by $G[V(G) - S]$. The following lemma follows immediately from the fact that for every subset $S \subset V(G)$ and for a vertex $x \in S$, we have $N_{G[S]}[x] = N[x] \cap S$ and that $G - S$ is an induced subgraph.

Lemma 1 ([18]). If G is a QT-graph, then for every subset $S \subset V(G)$, both $G[S]$ and $G - S$ are also QT-graphs.

The following theorem provides important properties for the class of QT-graphs. For convenience, we define

$$cent(G) = \{x \in V(G) \mid N[x] = V(G)\}.$$

Theorem 1 ([18]). The following three statements hold.

(i) A graph G is a QT-graph if and only if every connected induced subgraph $G[S]$, $S \subseteq V(G)$, satisfies $cent(G[S]) \neq \emptyset$.

(ii) A graph G is a QT-graph if and only if $G - cent(G)$ is a QT-graph.

(iii) Let G be a connected QT-graph. If $G - cent(G) \neq \emptyset$, then $G - cent(G)$ contains at least two connected components.

Let G be a connected QT-graph. Then $V_1 := cent(G)$ is not an empty set by Theorem 1. Put $G_1 := G$, and $G - V_1 = G_2 \cup G_3 \cup \ldots \cup G_r$, where each G_i is a connected component of $G - V_1$ and $r \geq 3$. Then since each G_i is an induced subgraph of G, G_i is also a QT-graph, and so let $V_i := cent(G_i) \neq \emptyset$ for $2 \leq i \leq r$. Since each connected component of G_i-$cent(G_i)$ is also a QT-graph, we can continue this procedure until we get an empty graph. Then we finally obtain the following partition of $V(G)$.

$$V(G) = V_1 + V_2 + \ldots + V_k, \quad \text{where } V_i = cent(G_i).$$

Moreover we can define a partial order $\leq$ on $\{V_1, V_2, \ldots, V_k\}$ as follows:

$$V_i \leq V_j \quad \text{if} \quad V_i = cent(G_i) \text{ and } V_j \subseteq V(G_i).$$

It is easy to see that the above partition of $V(G)$ possesses the following properties.

Theorem 2 ([18]). Let G be a connected QT-graph, and let $V(G) = V_1 + V_2 + \ldots + V_k$ be the partition defined above; in particular, $V_1 := cent(G)$. Then this partition and the partially ordered set $(\{V_i\}, \leq)$ have the following properties:

(P1) If $V_i \leq V_j$, then every vertex of V_i and every vertex of V_j are joined by an edge of G.

(P2) For every V_j, $cent(G[\{\cup V_i \mid V_i \leq V_j\}]) = V_j$.

(P3) For every two V_s and V_t such that $V_s \leq V_t$, $G[\{\cup V_i \mid V_s \leq V_i \leq V_t\}]$ is a complete graph. Moreover, for every maximal element V_t of $(\{V_i\}, \leq)$, $G[\{\cup V_i \mid V_1 \leq V_i \leq V_t\}]$ is a maximal complete subgraph of G.

(P4) Every edge with both endpoints in V_i is a free edge.

(P5) Every edge with one endpoint in V_i and the other endpoint in V_j, where $V_i \neq V_j$, is a semi-free edge.

The results of Theorem 2 provide algorithmic and structural properties for the class of QT-graphs. A typical structure of such a graph is shown in Figure 1. We shall refer to the structure which meets the properties of Theorem 2 as *cent-tree* $T_c(G)$. The cent-tree is a rooted tree with root V_1; every node V_i of the tree $T_c(G)$ is either a leaf or has at least two children. Moreover, $V_s \leq V_t$ iff V_s is an ancestor of V_t.

If V_i and V_j are disjoint vertex sets of a *QT*-graph G such that $V_i \leq V_j$ or $V_j \leq V_i$, we say that V_i and V_j are *clique-adjacent* and denote $V_i \approx V_j$.

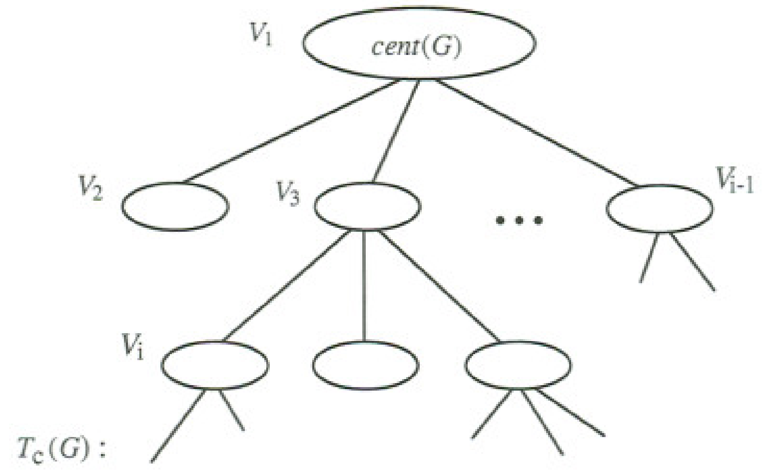

Figure 1. The typical structure of the *cent-tree* $T_c(G)$ of a *QT*-graph G.

Let G be a *QT*-graph and let $V = V_1 + V_2 + ... + V_k$ be the above partition of $V(G)$; $V_1 := cent(G)$. Let $S = \{v_s, v_{s+1}, ..., v_t, ..., v_q\}$ be a stable set such that $v_t \in V_t$ and V_t is a maximal element of $(\{V_i\}, \leq)$ or, equivalently, V_t is a leaf node of $T_c(G)$, $s \leq t \leq q$. It is easy to see that S has the maximum cardinality $\alpha(G)$ among all the stable sets of G. On the other hand the sets $\{\cup V_i \mid V_1 \leq V_i \leq V_t\}$, for every maximal element V_t of $(\{V_i\}, \leq)$, provide a clique cover of size $\kappa(G)$ which has the property to be a smallest possible clique cover of G; that is $\alpha(G) = \kappa(G)$. Based on the Theorem 2 or, equivalently, on the cent-tree of G, it is easy to show that the clique number $\omega(G)$ equals the chromatic number $\chi(G)$ of G; that is, $\chi(G) = \omega(G)$.

3. Hamiltonian Quasi-threshold Graphs

Let $V_1, V_2, ..., V_k$ be the nodes of the cent-tree $T_c(G)$ of a *QT*-graph G rooted at $r_c = V_1$, and let $V_{i1}, V_{i2}, ..., V_{ip}$ be the children of the node V_i ($1 \leq i \leq k$); note that $p \geq 2$ if V_i is not a leaf of the cent-tree. We assign a label *H-label*(V_i) to each node V_i of the cent-tree $T_c(G)$, which we compute as follows:

$$H\text{-}label(V_i) = \begin{cases} |V_i| - p & \text{if } V_i \text{ is the root of the tree,} \\ |V_i| - p + 1 & \text{if } V_i \text{ is an internal node, and} \\ 0 & \text{if } V_i \text{ is a leaf,} \end{cases}$$

where p is the number of children of the node V_i ($1 \leq i \leq k$); see also [17]. Figure 2 depicts a node V_i of a cent-tree along with its four children V_{i1}, V_{i2}, V_{i3} and V_{i4}; here we have *H-label*(V_i) = 2 if V_i is an internal node or *H-label*(V_i) = 1 if V_i is the root of the tree, *H-label*(V_{i1}) = 1, *H-label*(V_{i2}) = -1, *H-label*(V_{i3}) = 0, and H-$label(V_{i4}) = 0$. We shall show that G is a Hamiltonian QT-graph if H-$label(V_i) \geq 0$ for each node $V_i \in T_c(G)$.

Let $V_{i1}, V_{i2}, \ldots, V_{ip}$ be the children of an internal node V_i of the cent-tree $T_c(G)$ such that H-$label(V_i) \geq 0$, and let $list(V_i) = (v_{i1}, \ldots, v_{i(p-1)}, v_{ip}, \ldots, v_{is})$ be the list of the vertices of the node V_i, where $p \geq 2$ and $s \geq p-1$. Let a-$vertices(V_i) = (v_{ip}, v_{i(p+1)}, \ldots, v_{is})$; the elements of this list a-$vertices(V_i)$ are called *available vertices* of the node V_i. If V_i is the root of the cent-tree then a-$vertices(V_i) = (v_{i(p+1)}, v_{i(p+2)}, \ldots, v_{is})$. In Figure 2, for the internal node V_i we have a-$vertices(V_i) = \{u, v\}$.

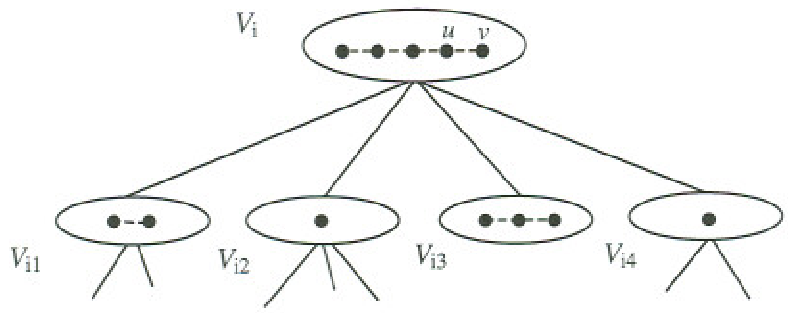

Figure 2. A node of the *cent-tree* $T_c(G)$ of a QT-graph G along with its four children; the lists of the vertices of G that correspond to these nodes.

Let $V_{f(1)}, V_{f(2)}, \ldots, V_{f(t)}$ be the left-to-right order listing of the leaves of the cent-tree $T_c(G)$, and let $V_{a(i)}$ be the lowest common ancestor of the nodes $V_{f(i)}$ and $V_{f(i+1)}$, where $2 \leq f(i) \leq k$ and $1 \leq i \leq t$-1. We define the *h-sequence* of the cent-tree $T_c(G)$ to be the following sequence:

$$h\text{-sequence}(T_c(G)) = (V_{f(1)}, V_{a(1)}, V_{f(2)}, V_{a(2)}, \ldots, V_{f(t-1)}, V_{a(t-1)}, V_{f(t)}, V_1)$$

where V_1 is the root of the tree $T_c(G)$ and t is the number of leaves in $T_c(G)$; the length of the h-sequence is $2t$; see also [17].

By definition there exists no pair $V_{f(i)}, V_{f(j)}$ of elements $V_{f(1)}, V_{f(2)}, \ldots, V_{f(t)}$ of the h-sequence$(T_c(G))$ such that $V_{f(i)} = V_{f(j)}$ for $i \neq j$, $1 \leq i, j \leq t$. On the other hand, may exist elements $V_{a(i1)}, V_{a(i2)}, \ldots, V_{a(iq)}$ such that $V_{a(i1)} = V_{a(i2)} = \ldots = V_{a(iq)} = V_i$, where V_i is an internal node $T_c(G)$; q is equal to the number of children of V_i minus 1. Let $a(i1)$ and $a(iq)$ be the indices of the leftmost and rightmost appearance of V_i in h-sequence$(T_c(G))$, and let $a(i1) < a(i2) < \ldots < a(iq)$. We say that $V_{a(i1)}$ is the first appearance of V_i, $V_{a(i2)}$ is the second appearance of V_i, and so on; $V_{a(iq)}$ is the last appearance of V_i in h-sequence$(T_c(G))$. Based on the structure of the cent-tree $T_c(G)$ and the fact that each internal node of $T_c(G)$ has at least two children we can easily conclude that each internal node of $T_c(G)$ appears at least once in the h-sequence. Thus, we have the following result.

Proposition 1. All the nodes of the cent-tree $T_c(G)$ of a QT-graph G are appeared in h-sequence$(T_c(G))$. Moreover, two consecutive node in h-sequence$(T_c(G))$ are clique-adjacent.

Let G be a QT-graph and h-sequence$(T_c(G))$ be its h-sequence. We next use a depth-first search (dfs) traversal strategy for searching the graph G and building a spanning tree of G. We shall use the h-sequence for the process of selecting the next unvisited vertex; note that in the standard dfs traversal when we have a choice of vertices to visit, we select them in alphabetical order. Based on the h-sequence for the selection process, we describe a dfs traversal, which, hereafter, we shall call h-dfs; it works as follows:

Traversal strategy h-dfs:

(i) Select an arbitrary vertex v from $V_{f(1)}$ as starting vertex; visit v and mark v visited (initially, all vertices of G are marked unvisited);

(ii) If v is visited and v is a vertex of $V_{f(i)}$, then visit in turn each unvisited vertex of $V_{f(i)}$ $(1 \leq i \leq t)$;

(iii) Once all the vertices of $V_{f(i)}$ have been visited, select an unvisited vertex u from the leftmost set $V_{a(j)} = V_i$ which lie on the right of $V_{f(i)}$ in h-sequence($T_c(G)$) $(i \leq j \leq t\text{-}1)$; visit u, mark u visited and if $V_{a(j)}$ is the last appearance of V_i in h-sequence, then visit in turn each unvisited vertex of $V_{a(j)}$; otherwise, does to $V_{f(i+1)}$ and select an unvisited vertex from this set and visit it.

(iv) Visit all the unvisited vertices of the last set V_1 of h-sequence($T_c(G)$);

It is well known that if G is a connected undirected graph, then the dfs forest of G contains only one tree. Moreover, it is obvious that if each node of the dfs tree rooted at $v \in V(G)$ has at most one child, then G contains a Hamiltonian path beginning with vertex v (it is the path from the root v to the unique leaf); G contains a Hamiltonian cycle if the root of the dfs tree and the unique leaf are adjacent in G. We next prove the following result.

Lemma 3. A QT-graph G is a Hamiltonian graph if $H\text{-}label(V_i) \geq 0$ for each node $V_i \in T_c(G)$.

Proof. Let $V_1, V_2, ..., V_k$ be the nodes of the cent-tree $T_c(G)$ of the QT-graph G rooted at V_1, and let $(V_{f(1)}, V_{a(1)}, V_{f(2)}, \ldots, V_{f(t)}, V_1)$ be the h-sequence of $T_c(G)$. Let $q(i)$ be the number of all the nodes of h-sequence(T_c), say, $V_{a(i1)}, V_{a(i2)}, ..., V_{a(iq)}$, such that $V_{a(i1)} = V_{a(i2)} = ... = V_{a(iq)} = V_i$, where V_i is an internal node $T_c(G)$; in fact, V_i is the *lca* of some pairs of leaves of $T_c(G)$. By definition, $q(i)$ is equal to the number of children of V_i minus 1.

Let $V_{i1}, V_{i2}, ..., V_{ip}$ be the children of the node V_i and let $list(V_i) = (v_{i1}, ..., v_{i(p-1)}, v_{ip}, ..., v_{is})$. Then, $q(i) = p - 1$. Since $H\text{-}label(V_i) \geq 0$, it follows that the V_i contains at least p-1 vertices; it contains p vertices if V_i is the root V_1 of the cent-tree $T_c(G)$.

We select a vertex v from $V_{f(1)}$ and apply h-dfs traversal to G starting at v. Since each V_i contains at least p - 1 vertices (it contains p vertices if $V_i = V_1$) and $q(i) = p - 1$ ($q(i) = p$ if $V_i = V_1$ because the last element of the h-sequence is the root V_1 of the cent-tree), it follows that after visiting the vertices of the node $V_{f(i)}$ there exists at least one unvisited vertex in $V_{a(i)}$ and, thus, the h-dfs always selects the next vertex from $V_{a(i)}$, $1 \leq i \leq t\text{-}1$; this is also true for the nodes $V_{f(t)}$ and V_1. On the other hand, the nodes $V_{a(i)}$ and $V_{f(i+1)}$ are clique-adjacent. Thus, the h-dfs tree of G has the property that each node has at most one child; that is, G contains a Hamiltonian path. Moreover, $V_{f(1)}$ and V_1 are clique-adjacent. Thus, G contains a Hamiltonian cycle. □

We conceder now the case where the cent-tree $T_c(G)$ of a Hamiltonian QT-graph has nodes, say, V_i and V_j, such that $V_i \leq V_j$ and $H\text{-}label(V_i) > 0$ and $H\text{-}label(V_j) < 0$. Let u be an available vertex of the node V_i. We define an operation that moves the available vertex u from the node V_i to node V_j. We call this operation *vertex-move*, or *v-move* for short.

From the structure of the cent-tree $T_c(G)$ of a QT-graph, it is easy to see that if we apply a v-move operation to nodes V_i and V_j, then the resulting tree has the Property (P3): for every two nodes V_s and V_t

such that $V_s \le V_t$, $G[\{\cup V_i \mid V_s \le V_i \le V_t\}]$ is a complete graph. Obviously, if V_t is a maximal element of $(\{V_i\}, \le)$, then after applying a *v*-move operation the graph $G[\{\cup V_i \mid V_1 \le V_i \le V_t\}]$ may not be a maximal complete subgraph of G.

Consider the tree that results from the cent-tree $T_c(G)$ of a QT-graph after applying some *v*-move operations on appropriate nodes so that each node V_i of that tree has *H-label* greater than or equal to 0; we call such a tree *h-tree* and denote it by $T_h(G)$. Then, we prove the following result.

Theorem 3. Let G be a QT-graph and let $T_c(G)$ be the cent-tree of G. The graph G is a Hamiltonian QT-graph if and only if either $H\text{-}label(V_i) \ge 0$ for each node $V_i \in T_c(G)$ or we can construct an h-tree $T_h(G)$ such that $H\text{-}label(V_i) \ge 0$ for each node $V_i \in T_h(G)$.

Proof. The if implication follows directly from Lemma 1 since $H\text{-}label(V_i) \ge 0$ for each node $V_i \in T_h(G)$. Note that Proposition 1 also holds for the h-tree $T_h(G)$.

Suppose now that there exist nodes in $T_h(G)$ with negative H-labels. Let V_i be such a node and let each ancestor V_{ij} of V_i has $H\text{-}label(V_{ij}) \ge 0$; note that the leaves of the tree $T_h(G)$ have zero H-labels. Since $H\text{-}label(V_i) < 0$, it follows that there exists no predecessor V_{ik} of V_i with available vertices; that is, $H\text{-}label(V_{ik}) \le 0$.

Let $V_{i1}, V_{i2}, \ldots, V_{ip}$ be the children of V_i and let q be the number of vertices $v_{i1}, v_{i2}, \ldots, v_{iq}$ of V_i, where $q < p - 1$ if V_i is an internal node and $q < p$ if V_i is the root of the tree $T_h(G)$. We construct the h-dfs tree of G rooted at vertex v; recall that the starting vertex v belongs to $V_{f(1)}$. We consider the following two cases: (i) $q < p - 1$. It is easy to see that the vertex v_{iq} has at least two children in the h-dfs tree. (ii) $q = p - 1$. In this case V_i is the root of the tree $T_h(G)$; that is, $V_i = V_1$, and each vertex in the h-dfs tree has only one child except, of course, of the unique leaf u. The vertices v and u are not adjacent in G since they do not belong to the same connected comport of the graph $G - V_1$. Thus, in both cases the graph G does not contain a Hamiltonian cycle. □

4. The Cent-tree of a Quasi-threshold Graph

The characterizations provided by Theorem 2 enable us to describe a linear-time algorithm for constructing the cent-tree of a QT-graph.

Let G be a QT-graph and let $T_c(G)$ be its cent-tree with node set $\{V_1, V_2, \ldots, V_k\}$ and root V_1. We have shown that if node V_i is an ancestor of node V_j in the cent-tree of G, then V_i and V_j are clique-adjacent. Thus, if $(V_1, V_2, \ldots, V_i)$ is a path from the root V_1 of the cent-tree to a node V_i, then $deg(V_1) > deg(V_2) > \ldots > deg(V_i)$, where $deg(V_i)$ denotes the degree of the vertices of G that belong to node V_i; note that all the vertices of G that belong to node V_i have the same degree and that each internal node of the cent-tree of G has at least two children. It follows that if $\{v_1, v_2, \ldots, v_p\}$ is a clique in a QT-graph, then $deg(v_1) \ge deg(v_2) \ge \ldots \ge deg(v_p)$, $1 \le p \le n$; see also [24].

Based on this property, we describe a method which produces a rooted tree representation of a QT-graph; see also [24]. We call this tree *degree-tree* of G, or d-tree for short, and we denote it by $T_d(G)$. The method is as follows. First, sort the vertices $v_1, v_2, \ldots, v_n$ of G according to their degrees; let $D = (v_1, v_2, \ldots, v_n)$ be a sequence such that $deg(v_1) \ge deg(v_2) \ge \ldots \ge deg(v_n)$. Then, construct the tree T_g with vertex set $\{v_1, v_2, \ldots, v_n\}$ in the following manner: for every vertex $v_i \in D$, $2 \le i \le n$, find the vertex v_k, if it exists, such that k is the maximum index satisfying $1 \le k < i$ and (v_k, v_i) is an edge in G; add the edge (v_k, v_i) into $E(T_g)$. Finally, root the tree T_g at vertex $r = v_1$. The resulting tree is the *d*-tree $T_d(G)$ of the *QT*-graph *G*.

We next describe the above method in a more formal way. Note that we do not need to compute the degree sequence of the input graph. The algorithm is a modification of the recognition algorithm presented in [24]; it takes as input a *QT*-graph *G* and produces the *d*-tree $T_d(G)$.

Algorithm D-Tree-Construction (DTC):

(1) For each edge $(v_i, v_j) \in E(G)$ do the following:
  If $deg(v_i) > deg(v_j)$ or ($deg(v_i) = deg(v_j)$ and $i < j$)
    then $level(v_j) \leftarrow level(v_j) + 1$
    else $level(v_i) \leftarrow level(v_i) + 1$;
 Initially, $level(v_i) = 0$ for every vertex $v_i \in V(G)$;

(2) Construct the tree T_g with vertex set $V(T_g) = \{v_1, v_2, ..., v_n\}$ as follows:
  For each edge $(v_i, v_j) \in E(G)$ do the following:
  If $level(v_i) = level(v_j) + 1$ then add the edge (v_i, v_j) in $E(T_g)$;

(3) Root the tree T_g at vertex $r = v_i$ such that $level(v_i) = 0$;
 The resulting tree is the *d*-tree $T_d(G)$ of the *QT*-graph *G*;

Let us now compute the complexity of the above construction algorithm. The degree $deg(v_i)$ of the vertex v_i can be computed in $O(d_i)$ time. Thus, since $\Sigma_{v_i \in V} d_i = O(m)$, it takes $O(n + m)$ time. Obviously, both steps (1) and (2) are executed in $O(m)$ time. It is well known that we can root an *n*-node tree in $O(n)$ time. Thus, the *d*-tree of a *QT*-graph can be constructed in linear time.

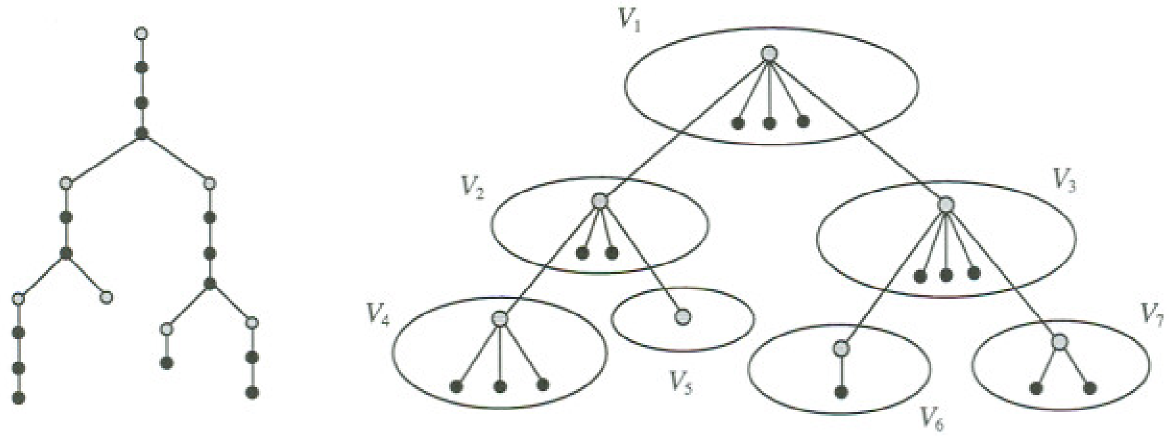

Figure 3. The degree-tree $T_d(G)$ and the *cent-tree* $T_c(G)$ of a *QT*-graph *G*.

Based on the structural properties of the *d*-tree $T_d(G)$ of a *QT*-graph, we next present a linear-time algorithm for the construction of the cent-tree $T_c(G)$ of the graph *G*.

We observe that, a vertex *u* and its parent $p(u)$ belong to the same node set V_i of the cent-tree of *G* iff *u* is a unique child of the vertex $p(u)$ in the *d*-tree $T_d(G)$; see Figure 3. Let $u_2, ..., u_k$ be the vertices of the *d*-tree $T_d(G)$ with the property that their parents have at least two children and let $R = \{r = u_1\ u_2, ..., u_k\}$, where *r* is the root of the *d*-tree. It is easy to see that, the cent-tree $T_c(G)$ has nodes $V_1, V_2, ..., V_k$ and

$u_i \in V_i$, $1 \leq i \leq k$. The node V_1 is the root of the cent-tree and the node $V_i = \{u_i\}$ has parent the node $V_j = \{u_j\}$ in $T_c(G)$ if u_j is the least ancestor of u_i in $T_d(G)$ that belongs to R. The vertex $u \notin R$ of the graph G belongs to the node set V_i if the least ancestor of u in $T_d(G)$ that belongs to R is the vertex u_i, $1 \leq i \leq k$; see Figure 3.

More precisely, we have the following algorithm. It takes as input a *QT*-graph G and produces the cent-tree of the graph G.

Algorithm Cent-Tree-Construction (CTC):

(1) Compute the *d*-tree $T_d(G)$ of the input *QT*-graph G using Algorithm DTC;

(2) For each vertex v_i of the *d*-tree $T_d(G)$, $1 \leq i \leq n$, do the following
  If v_i is the root r of the tree or its parent $p(v_i)$ has more than one child, then
  set $color(v_i) \leftarrow$ red; otherwise $color(v_i) \leftarrow$ black;
  Let $r = u_1, u_2, ..., u_k$ be the red vertices of $T_d(G)$, $k \geq 1$;

(3) For each red vertex u_i do the following
  Construct a node set V_i and set $V_i \leftarrow \{u_i\}$ and $label(u_i) \leftarrow i$, $1 \leq i \leq k$,

(4) Visit (preorder) each vertex v_i of the *d*-tree $T_d(G)$ and do the following
  If v_i is a black vertex then set $label(v_i) \leftarrow label(p(v_i))$;

(5) Construct the tree T_g as follows:
  Set $V(T_g) \leftarrow \{r = V_1, V_2, ..., V_k\}$;
  For each black vertex $v_i \in T_d(G)$, do the following
    if $label(v_i) = j$, then add vertex v_i in V_j, $1 \leq j \leq k$;
  For each read vertex $u_i \in V_i$, $1 \leq i \leq k$, do the following
    if $label(p(u_i)) = j$, then add the edge (V_i, V_j) in $E(T_g)$;

(6) Root the tree T_g at node V_1;
  The resulting tree is the cent-tree $T_c(G)$ of the graph G;

We next compute the complexity of the proposed algorithm for the construction of the cent-tree of a *QT*-graph. *Step 1*: The rooted *d*-tree $T_d(G)$ is constructed in $O(n + m)$ time using Algorithm DTC. *Steps 2* through *5*: It is easy to see that all these steps are executed in $O(n)$ time. *Step 6*: The process of rooting an n-node tree takes $O(n)$ time.

Therefore, from the above analysis, it follows that the construction algorithm CTC runs in $O(n + m)$ time. Thus, we have the following result.

Theorem 4. The cent-tree of a *QT*-graph on n vertices and m edges can be constructed in $O(n + m)$ time.

5. The Main Results

It is well known that it is NP-complete to recognize whether a graph is Hamiltonian [2]. In Section 3 (Theorem 3) we give necessary and sufficient conditions for a *QT*-graph to be Hamiltonian. From this condition, a linear-time algorithm can be constructed for finding a Hamiltonian cycle in a Hamiltonian *QT*-graph G, and also linear-time algorithms for computing the Hamiltonian completion number $hcn(G)$ and the Hamiltonian completion edge set $CE(G)$ of G. Obviously, if $hcn(G) = 0$ then G is a Hamiltonian graph. Thus, the algorithm for the computation of the number $hcn(G)$ is also a recognition algorithm.

Let G be a Hamiltonian QT-graph and let $T_c(G)$ be its cent-tree with nodes $V_1, V_2, ..., V_k$ and root V_1. Suppose that $H\text{-}label(V_i) \geq 0$ for each node $V_i \in T_c(G)$. Then, by Theorem 3, the graph G has a Hamiltonian cycle. Conceder the h-sequence $(V_{f(1)}, V_{a(1)}, \ldots, V_1)$ of the cent-tree of G and construct the h-dfs tree of the graph G using the h-dfs traversal strategy on the sent-tree $T_c(G)$; see Section 3. We select an arbitrary vertex v from the set $V_{f(1)}$ as start point. Since $H\text{-}label(V_i) \geq 0$ for each node $V_i \in T_c(G)$, it is easy to see that each node of the h-dfs tree rooted at $v \in V_{f(1)}$ has at most one child; its unique leaf u belongs to node V_1 and, thus, $(v, u) \in E(G)$; see Figure 4. It follows that we can find a Hamiltonian cycle of the graph G from its h-dfs tree.

The cent-tree and the h-sequence of a QT-graph are constructed in linear time. Thus, in the case where $H\text{-}label(V_i) \geq 0$ for each node $V_i \in T_c(G)$, a Hamiltonian cycle of G can be constructed in linear time.

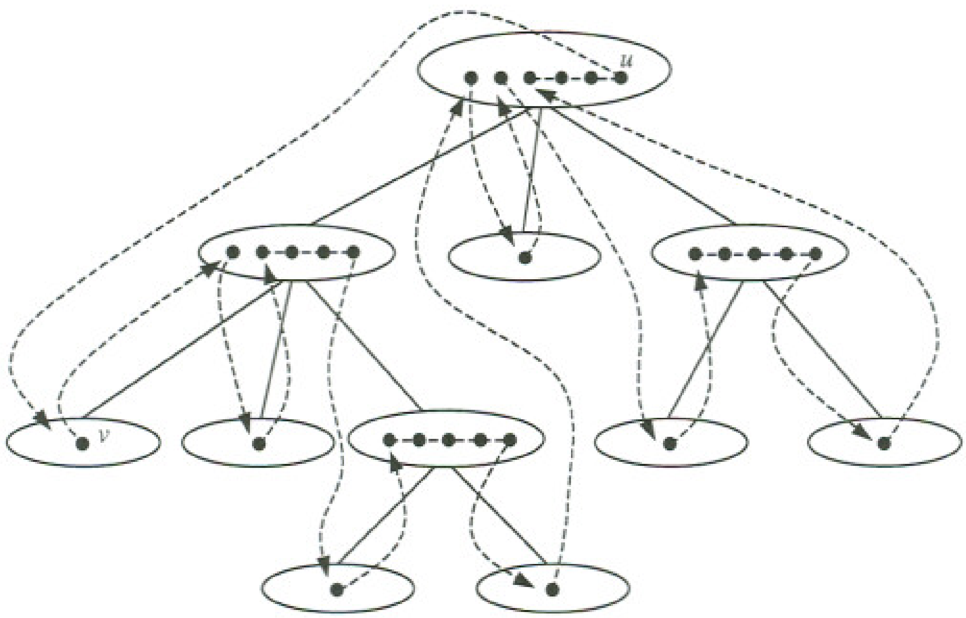

Figure 4. The structure of a Hamiltonian cycle of a QT-graph G; it is produced using the h-dfs traversal strategy on the h-tree $T_h(G)$.

Conceder now the case where the cent-tree of a Hamiltonian QT-graph has nodes, say, V_i and V_j, such that V_i is an ancestor of V_j, and $H\text{-}label(V_i) > 0$ and $H\text{-}label(V_j) < 0$. In this case we first construct the h-tree $T_h(G)$ of the cent-tree $T_c(G)$, and then use the h-dfs traversal strategy on the h-tree to construct the h-dfs tree of the graph G. Since $H\text{-}label(V_i) \geq 0$ for each node $V_i \in T_h(G)$, it follows that the h-dfs tree has the same structure as that of the previous case. Thus, the h-dfs tree gives us all the necessary information to compute a Hamiltonian cycle of the graph G.

We next present an algorithm which takes as input the cent-tree $T_c(G)$ of a Hamiltonian QT-graph and produces the h-tree $T_h(G)$ of the cent-tree of G. Since G is a Hamiltonian QT-graph, the algorithm always succeed in computing the h-tree $T_h(G)$.

Algorithm H-tree-Construction (HTC):

(1) If $H\text{-}label(V_i) \geq 0$ for each node $V_i \in T_c(G)$, $1 \leq i \leq k$, then
$T_c(G)$ is an h-tree; set $T_h(G) \leftarrow T_c(G)$ and exit;

(2) For each node $V_i \in T_c(G)$, $1 \leq i \leq k$, such that $H\text{-}label(V_i) < 0$, do the following
Find the least ancestor V_p of V_i in $T_c(G)$ such that $H\text{-}label(V_p) > 0$;
Move available vertices from V_p to V_i until $H\text{-}label(V_i) = 0$ or $H\text{-}label(V_p) = 0$;

We can easily show that the H-labels of the nodes of the cent-tree $T_c(G)$ can be computed in $O(n)$ time; the $H\text{-}label(V_i)$ of the node V_i can be computed in $O(n_i + p_i)$ time, where n_i is number of vertices in V_i and p_i is the number of children of node V_i in the cent-tree $T_c(G)$. The least ancestor V_p of a node V_i such that $H\text{-}label(V_p) > 0$ can also be computed in $O(n + m)$ time using similar techniques to that of computing the cent-tree $T_c(G)$ tree from the d-tree $T_d(G)$; see Algorithm CTC. Since each available vertex is moved at most once, we conclude that the h-tree $T_h(G)$ can be constructed from the cent-tree $T_c(G)$ in linear time. Thus, we obtain the following result.

Theorem 5. Let G be a Hamiltonian QT-graph on n vertices and m edges. A Hamiltonian cycle of G can be constructed in $O(n + m)$ time.

We consider now the case where G is not a Hamiltonian QT-graph. Let G be such a graph and let $T_c(G)$ be its cent-tree. From Theorem 3, we have that there are nodes in the cent-tree $T_c(G)$ with negative H-labels and the h-tree of $T_c(G)$ does not exist; that is, Algorithm HTC fails to construct the h-tree $T_h(G)$. That means, there is a node $V_i \in T_c(G)$ such that $H\text{-}label(V_i) < 0$ for which the there exists no ancestor V_p of V_i in $T_c(G)$ such that $H\text{-}label(V_p) > 0$.

In this case we are interested in computing the minimum number of edges which need to be added to $E(G)$ to make the graph G Hamiltonian. To this end, we construct a Hamiltonian QT-graph D from the graph G by adding dummy vertices in the set $V(G)$ and appropriate edges in the set $E(G)$. We shall define the graph D through its corresponding h-tree; we call dh-tree the h-tree of the graph D. The dh-tree is constructed from the cent-tree $T_c(G)$ of the graph G; the construction algorithm is as follows.

Algorithm DH-tree-Construction (DHTC):

(1) Step (1) of Algorithm H-tree-Construction (HTC);

(2) For each node $V_i \in T_c(G)$, $1 \leq i \leq k$, such that $H\text{-}label(V_i) < 0$, do the following
If there exist a least ancestor V_p of V_i in $T_c(G)$ such that $H\text{-}label(V_p) > 0$
then move available vertices from V_p to V_i until $H\text{-}label(V_i) = 0$ or $H\text{-}label(V_p) = 0$
else add dummy vertices $dv_{i1}, dv_{i2}, \ldots, dv_{ih}$ in node set V_i until $H\text{-}label(V_i) = 0$;

The dh-tree constructed from the above algorithm and the cent-tree $T_c(G)$ have the same structures. Moreover, the dh-tree has the property that $H\text{-}label(V_i) \geq 0$ for each its node V_i, $1 \leq i \leq k$; note that the node V_i of the dh-tree contains the vertices of the node V_i of the cent-tree along with, probably, some dummy vertices. Thus, the graph D which corresponds to dh-tree is a Hamiltonian QT-graph.

More precisely, let $v_1, v_2, \ldots, v_n$ be the vertices of G and let $dv_1, dv_2, \ldots, dv_h$ be the dummy vertices added to $T_c(G)$ by the dh-tree construction algorithm. Then, the graph D is the following:

(i) $V(D) = \{v_1, v_2, \ldots, v_n, dv_1, dv_2, \ldots, dv_h\}$, and

(ii) $E(D)$ contains all the edges of G and edge of the form (u, dv_j) if $u \in V_i$ and $dv_j \in V_j$ and the node sets V_i, V_j have the property: V_i (V_j) is an ancestor of V_j (V_i) or $V_i = V_j$ in the cent-tree $T_c(G)$; the vertex u is either a vertex of G or a dummy vertex.

By construction, the Hamiltonian QT-graph D contains the QT-graph G as an induced subgraph. Moreover, it is easy to see that if we remove a vertex from D then it is no longer a Hamiltonian QT-graph. This proves the following result.

Lemma 4. The graph D constructed by Algorithm DHTC is a minimum order Hamiltonian QT-graph which contains the QT-graph G as an induced subgraph.

Let G be a non Hamiltonian QT-graph and let $T_c(G)$ be its cent-tree. We conceder the Hamiltonian QT-graph D and its dh-tree constructed from $T_c(G)$ by Algorithm DHTC. We have shown that, a Hamiltonian cycle of the graph D can be produced using the h-dfs tree of the graph D; recall that the h-dfs tree is constructed by the h-dfs traversal strategy on the dh-tree. We prove the following result.

Lemma 5. The Hamiltonian completion number $hcn(G)$ of a non Hamiltonian QT-graph G equals the number of dummy vertices in the dh-tree of the graph D computed by Algorithm DHTC.

Proof. Consider the h-sequence $(V_{f(1)}, V_{a(1)}, \ldots, V_1)$ of the dh-tree and the h-dfs tree of the graph D. Let $v \in V_{f(1)}$ be the root of the h-dfs tree and let $HC = (v, \ldots, v_i, dv_j, v_k, \ldots, v)$ be the Hamiltonian cycle which is produced by the h-dfs tree, where dv_j is a dummy vertex. Note that HC is a cycle on $n + h$ vertices, where n is the number of vertices in G and h is the number of dummy vertices in the dh-tree. By construction, the cycle HC has the following properties:

(i) If dv_j is a dummy vertex in HC, then its two adjacency vertices, say, v_i and v_k, are not dummy vertices; that is, $v_i, v_k \in V(G)$, and

(ii) if $v_i \in V_i$ and $v_k \in V_k$, then both nodes V_i and V_k are leaves in $T_c(G)$ and $V_i \neq V_k$.

Thus, if we remove each dummy vertex dv_j from HC and makes the vertices v_i and v_k to be adjacent, the resulting structure HC^* is a cycle on n vertices $v_1, v_2, \ldots, v_n$.

Since the vertices v_i and v_k belong to deferent leaves in $T_c(G)$, it follows that v_i and v_k are not adjacent in the graph G. Thus, if we add the edges (v_i, v_k) in $E(G)$, then the resulting graph G^* is Hamiltonian and the cycle HC^* is a Hamiltonian cycle of it.

We have proved that the number h of dummy vertices which need to be added to the nodes of $T_c(G)$ to produce the dh-tree is minimum. The graph G^* is Hamiltonian, $V(G^*) = V(G)$, $E(G^*) \supset E(G)$ and $|E(G^*)| = |E(G)| + h$. Therefore, $h = hcn(G)$ and the lemma is proved. □

The preceding lemma provides a linear-time algorithm for computing the Hamiltonian completion number of a QT-graph on n vertices since $h < n$. Moreover, it provides a linear-time algorithm for computing the Hamiltonian completion edge set $CE(G)$; that is, the set of edge which need to be added to $E(G)$ to make G Hamiltonian. Note that, $|CE(G)| = hcn(G)$. Thus, we have the following theorem.

Theorem 6. The Hamiltonian completion number of a QT-graph G on n vertices and m edges can be computed in $O(n + m)$ time. Moreover, the Hamiltonian completion edge set of G can be computed within the same time bound.

The method we have described for the computation of the number $hcn(G)$ and the edge set $CE(G)$ of a QT-graph G based on the construction of the hd-tree and the computation of a Hamiltonian cycle of the graph D.

The computation of the Hamiltonian completion number of a QT-graph G can also be done in linear time in a much simpler way using only the structural properties of the cent-tree $T_c(G)$ and the H-labels of its nodes. This computation is described in the following algorithm.

Algorithm Hamiltonian-Completion-Number (HCN):

(1) Compute the cent-tree $T_c(G)$ of G using Algorithm CTC;
Let $V_1, V_2, ..., V_k$ be the nodes of the tree $T_c(G)$ and let $r_c = V_1$ be its root;

(2) For each node $V_i \in T_c(G)$ compute the label $H\text{-}label(V_i)$, $1 \leq i \leq k$;

(3) Contract the cent-tree $T_c(G)$ into a 1-node tree by applying delete operations on the leaves of the $T_c(G)$; when a leaf node V_i ($V_i \neq V_1$) is subject to a deletion operation, we adjust the H-label of the parent $p(V_i)$ of the node V_i, as follows:

If $H\text{-}label(V_i) < 0$ then

$H\text{-}label(p(V_i)) \leftarrow H\text{-}label(p(V_i)) + H\text{-}label(V_i)$;

Delete the leaf node V_i;

Let V_1 be the root of the resulting 1-node tree;

(4) Compute the Hamiltonian completion number $hcn(G)$ of the input graph G as follows:

If $H\text{-}label(V_1) < 0$ then $hcn(G) \leftarrow \mid H\text{-}label(V_1) \mid$

else $hcn(G) \leftarrow 0$; that is, G is a Hamiltonian QT-graph;

Let us now determine the time required for the execution of the proposed algorithm for the computation of the number $hcn(G)$ of a QT-graph. *Step1*: The cent-tree of a QT-graph can be constructed in $O(n + m)$ time by Algorithm CTC. *Step2*: This step takes $O(n)$ time; see the analysis of Algorithm HTC. *Step3*: The contraction process on an n-node tree can be completed in $O(n)$ time. *Step4*: Obviously, this step is executed in $O(1)$ time.

From the previous step-by-step analysis, it follows that Algorithm HCR runs in $O(n + m)$ time. Thus, we have proved the following result.

Theorem 7. The Hamiltonian completion number of a QT-graph on n vertices and m edges can be computed in $O(n + m)$ time.

Corollary 1. It can be decided whether a QT-graph on n vertices and m edges is a Hamiltonian graph in $O(n + m)$ time.

6. Concluding Remarks

In this paper we studied the class of QT-graphs and proved structural and algorithmic properties on its members. We showed that a QT-graph G has a unique tree representation, the *cent-tree*, and proved necessary and sufficient conditions for a QT-graph to be Hamiltonian. Based on these properties and conditions, we constructed linear-time algorithms for finding a Hamiltonian cycle and computing the Hamiltonian completion number of G. It is obvious that the linear-time algorithm for the Hamiltonian completion number is also a recognition algorithm. We also showed that the Hamiltonian completion edge set of a *QT*-graph can be computed in linear time and proposed an $O(n + m)$-time algorithm.

Based on the structure of the cent-tree of a *QT*-graph, we can also design linear-time algorithms for some well-known optimization problems on *QT*-graphs. For example, the maximum clique problem, the maximum independent set problem, the clique cover problem and the coloring problem cam be solved in linear time.

Different problems can be foreseen for further research. An interesting optimization problem is the construction of a Hamiltonian cycle of a *QT*-graph G in the weighted case: each vertex and/or edge of G has certain weight and we wish to minimize the total weight of edges in a Hamiltonian cycle. A second problem that is worth studying is the weighted version of the Hamiltonian completion edge set problem: we wish to minimize the total weight of the edges (with respect to the weights of its end-vertices) of the set $CH(G)$. We pose these as open problems for algorithmic study.

A topic for further research is the study of problems on the line graph of a *QT*-graph (for results on line graphs, see [4, 5, 20, 22, 23]). One can work towards the identification of structural and algorithmic properties of such graphs, which may lead to linear-time algorithms for the Hamiltonian problems we conceder here as well as for other combinatorial and optimization problems.

References

[1] A. Agnetis, P. Detti, C. Meloni and D. Pacciarelli, A linear algorithm for the Hamiltonian completion number of the line graph of a tree, *Inform. Process. Lett.* 79 (2001) 17-24.

[2] A.A. Bertossi, The edge Hamiltonian problem is NP-hard, *Inform. Process. Lett.* 13 (1981) 177-159.

[3] A. Brandstädt, B. Le and J.P. Spinrad, *Graph Classes: A Survey*, Siam Monographs on Discrete Math. and Appl., 1999.

[4] H.J. Broersma, Subgraphs conditions for dominating circuits in graphs and pancyclicity in line graphs, *Ars Combinatoria* 23 (1987) 5-12.

[5] R.A. Brualdi and R.F. Shong, Hamiltonian line graphs, *J. of Graph Theory* 5 (1981) 304-307.

[6] D.G Corneil, H. Lerches and L. Burlingham, Complement reducible graphs, *Discrete Appl. Math.* 3 (1981) 163-174.

[7] V. Chvatal and P.L. Hammer, Set-packing and threshold graphs, Res. Report CORR 73-21, University of Warerloo, 1973.

[8] D.G. Corneil, Y. Perl and L.K. Stewart, A linear recognition algorithm for cographs, *SIAM J. Comput.* 14 (1985) 926-934.

[9] P. Detti and C. Meloni, A linear algorithm for the Hamiltonian completion number of the line graph of a cactus, *Proc. CTW'2001*, in: *Electronic Notes in Discrete Math.* (Elsevier), vol. 8, 2001.

[10] M.C. Golumbic, Trivially perfect graphs, *Discrete Math.* 24 (1978) 105-107.

[11] M.C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, Inc., New York, 1980.

[12] S. Goodman and S. Hedetniemi, On the Hamiltonian completion problems, in: A. Dold, B. Eckman (Eds), Graphs and Combinatorics, Lecture Notes in Math. (Springer, Berlin), Vol. 406, pp. 262-272, 1974.

[13] F. Harary and S.D. Nikolopoulos, On complete systems of invariants for small graphs, *Inter. J. of Comput. Math.* 64 (1997) 35-46.

[14] S. Kundu, A linear algorithm for the Hamiltonian completion number of a tree, *Inform. Process. Lett.* 5 (1976) 55-57.

[15] S. Ma, W.D. Wallis and J. Wu, Optimization problems on quasi-threshold graphs, *J. Comb. Inform. & Syst. Sciences*. 14 (1989) 105-110.

[16] T.A. McKee and F. R. McMorris, *Topics in Intersection Graph Theory*, Siam Monographs on Discrete Math. and Appl., 1999.

[17] S.D. Nikolopoulos, Hamiltonian cycles in quasi-threshold graphs, *Proc. CTW'2001*, in: *Electronic Notes in Discrete Math.* (Elsevier), vol. 8, 2001.

[18] S.D. Nikolopoulos, Recognizing cographs and threshold graphs through a classification of their edges, *Inform. Process. Lett.* 75 (2000) 129-139.

[19] A. Raychaudhuri, The total interval number of a tree and the Hamiltonian completion number of its line graph, *Inform. Process. Lett.* 56 (1995) 299-306.

[20] H.J. Veldman, A result on Hamiltonian line graphs involving restrictions on induced subgraphs, *J. of Graph Theory* 12 (1988) 413-420.

[21] E.S. Wolk, The comparability graph of a tree, *Proc. Amer. Math, Soc.* 3 (1962) 789-795.

[22] L. Xiong, The Hamiltonian index of a graph, *Graphs and Combinatorics*, to appear.

[23] L. Xiong, H.J. Broersma, C. Hoede and X. Li, Degree sum and subpancyclicity in line graphs, *Discrete Math.*, to appear.

[24] J-H. Yan, J-J. Chen and G.J. Chang, Quasi-threshold graphs, *Discrete Appl. Math.* 69 (1996) 147-255.