Efficient Parallel Algorithms on Distance-Hereditary Graphs (Extended Abstract)

Sun-yuan Hsieh *

Dept. of Computer Science & Info. Eng.
National Taiwan University, Taiwan
e-mail: d3506013@csie.ntu.edu.tw

Tsan-sheng Hsu †

Institute of Information Science
Academia Sinica, Taiwan
e-mail: tshsu@iis.sinica.edu.tw

Chin-Wen Ho

Dept. of Computer Science & Info. Eng.
National Central University, Taiwan
e-mail: hocw@csie.ncu.edu.tw

Ming-Tat Ko

Institute of Information Science
Academia Sinica, Taiwan
e-mail: mtko@iis.sinica.edu.tw

Gen-Huey Chen

Dept. of Computer Science & Info. Eng.
National Taiwan University, Taiwan
e-mail: ghchen@csie.ntu.edu.tw

Abstract

In this paper, we present efficient parallel algorithms for finding a minimum weighted connected dominating set, a minimum weighted Steiner tree for a distance-hereditary graph which take $O(\log n)$ time using $O(n+m)$ processors on a CRCW PRAM, where n and m are the number of vertices and edges of a given graph, respectively. We also find a maximum weighted clique of a distance-hereditary graph in $O(\log^2 n)$ time using $O(n+m)$ processors on a CREW PRAM.

1 Introduction

A graph is *distance-hereditary* if every two vertices have the same distance in every connected induced subgraph containing both (where the *distance* between two vertices is the length of a shortest path connecting them). Properties and optimization problems in distance-hereditary graphs have been extensively studied during the past two decades [2, 7, 8, 9, 11].

A *dominating set* D of a graph $G = (V, E)$ is defined as $D \subseteq V$ such that every vertex in $V \setminus D$ is either in D or is adjacent to some vertex in D . A dominating set D is *connected* if the subgraph induced by D is connected. For a given graph G and a set $K \subseteq V$ (of *terminal vertices*), a *Steiner tree* is a tree which spans all vertices of K . The *connected dominating set problem* CD (respectively, *Steiner tree problem* ST) asks for a minimum cardinality connected dominating set (respectively, Steiner tree). In this paper, we consider the following weighted version of the CD and ST on distance-hereditary graphs. Let $w(v)$ be a non-negative weight associated with each vertex v in

G . We want to find a connected dominating set D (respectively, Steiner tree T) such that $\sum_{v \in D} w(v)$ (respectively, $\sum_{v \in T} w(v)$) equals the smallest possible value. We show that the above problems can be solved in $O(\log n)$ time using $O(n+m)$ processors on a CRCW PRAM, where n and m are the number of vertices and edges of a given graph, respectively.

A graph C is a *clique* if there is an edge between every pair of vertices. We say that C is a *clique* of G if C is an induced clique of G . If each vertex is assigned a non-negative weight, a *maximum weighted clique* is a clique of G with the maximum total weight. In this paper, we also compute a maximum weighted clique of a distance-hereditary graph in $O(\log^2 n)$ time using $O(n+m)$ processors on a CREW PRAM.

2 Preliminaries

This paper considers finite, simple loopless, undirected and connected graphs $G = (V, E)$, where V and E are the vertex and edge set of G , respectively. Let $n = |V|$ and $m = |E|$. The *distance* $d_G(x, y)$ or $d(x, y)$ between two vertices x and y in G is the length of a shortest x - y path in G . Let v be a vertex of G . We denote the *neighborhood* of v , consisting of all vertices adjacent to v , by $N(v)$, and the *closed neighborhood* of v , the set $N(v) \cup \{v\}$, by $N[v]$. Let S be a subset of V . We denote $N(S)$ the *open neighborhood* of S , that is the set of vertices in G , exclusive of S , which are adjacent to any vertex in S . We also denote $N[S] = N(S) \cup S$. The *subgraph induced by* S , denoted by $\langle S \rangle$, consists of the vertices of S and edges (x, y) with $x, y \in S$ and $(x, y) \in E$.

The *hanging* of a connected graph $G = (V, E)$ at a vertex $u \in V$, denoted as h_u , is the collection of sets $L_0(u)$, $L_1(u)$, ..., $L_t(u)$ (or simply $L_0, L_1, \dots, L_t$ when no ambiguity

*Supported in part by Institute of Information Science, Academia Sinica, Taipei, Taiwan.

†Supported in part by NSC Grant 86-2213-E-001-012.

arises), where $t = \max_{v \in V} d_G(u, v)$ and $L_i(u) = \{v \in V : d_G(u, v) = i\}$ for $0 \leq i \leq t$. For any vertex $v \in L_i$ and vertex set $S \subseteq L_i$, $1 \leq i \leq t$, let $N'(v) = N(v) \cap L_{i-1}$ and let $N'(S) = N(S) \cap L_{i-1}$. Any two vertices $x, y \in L_i$ ($1 \leq i \leq t-1$) are said to be *tied* if x and y have a common neighbor in L_{i+1} . A *homogeneous set* of a graph G is a set A of vertices such that every vertex in $V(G) - A$ is adjacent to either all or none of the vertices of A .

A graph is a *cograph* (also a P_4 -free graph) [5] if it is either a vertex, the complement of a cograph, or the union of two cographs (where by union of two graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ we mean the graph $G = (V_1 \cup V_2, E_1 \cup E_2)$). A cograph G has a parse tree representation called *cotree*, denoted by T_G , with (a) the leaves of T_G are the vertices of G ; (b) the internal nodes of T_G are labelled 0 or 1; (c) 0 nodes and 1 nodes alternate along every path starting from the root; (d) two vertices x and y of the cograph are adjacent if and only if the least common ancestor of x and y in T_G is a 1 node. In the following, we introduce properties of distance-hereditary graphs.

Proposition 2.1 [8] Suppose $h_u = (L_0, L_1, \dots, L_t)$ is the hanging at u of a connected distance-hereditary graph. If $x, y \in L_i$ ($1 \leq i \leq t$) are in the same component of $\langle L_i \rangle$, or x and y are tied, then $N'(x) = N'(y)$.

Fact 1 A hanging of a connected distance-hereditary graph can be computed in $O(\log n)$ time using $O(n + m)$ processors on a CRCW PRAM.

Proof: An implementation is described as follows. (a) We first find an arbitrary spanning tree S for G in $O(\log n)$ time using $O((m + n)\alpha(m, n)/\log n)$ processors on a CRCW PRAM, where α is the inverse Ackermann function [4], and then transform it into a rooted tree with the root u in $O(\log n)$ time using $O(n)$ processors on an EREW PRAM [10]. (b) For each vertex $v (\neq u)$, we associate it with a number $num(v)$ which is the distance from v to u in S (the computation of distances for those non-root vertices can be done in $O(\log n)$ time using $O(n/\log n)$ processors on an EREW PRAM [10]). (c) For each vertex x , let $(x, y) \in E(T)$ if $num(y)$ is the smallest value in $\{num(z) : z \in N(x)\}$. This can be done in $O(\log n)$ time using $O(n + m)$ processors on an EREW PRAM by the sorting algorithm described in [3]. It can be verified that $T = (V, E(T))$ is a quasi-breadth-first-search tree which is a hanging at u . It has been shown that a quasi-breadth-first-search tree is also a breadth first search tree on a distance-hereditary graph [6]. Note that L_i is the set of vertices $x \in T$, such that the unique path from x to u is of length exactly i . Q.E.D

Proposition 2.2 [2] Suppose $h_u = (L_0, L_1, \dots, L_t)$ is a hanging of a connected distance-hereditary graph at u . For any two vertices $x, y \in L_i$, $i \geq 1$, $N'(x)$ and $N'(y)$ are either disjoint, or one of the two sets is contained in the other.

We call a family of subsets *arboreal* if any two elements of the family are either disjoint or comparable (by set inclusion). For example, $\{\{1, 2, 3\}, \{1, 2\}, \{4\}\}$ is an arboreal family of subsets of $\{1, 2, 3, 4\}$. Given an arboreal family of distinct subsets, denoted by $S_1, S_2, \dots, S_r$, we can define a partial order $\preceq$ between S_i 's by $S_p \preceq S_q \Leftrightarrow S_p \subseteq S_q$,

where $1 \leq p, q \leq r$. An element S_p is said to be *minimal* in the given family if $S_p \not\supseteq S_q$, for all $q \neq p$ and $1 \leq q \leq r$.

In [8], Hammer and Maffray defines an equivalence relation $\equiv_i$ between vertices of L_i by $x \equiv_i y$ means x and y are in the same connected component of L_i or x and y are tied. Let $\equiv_a$ be defined on $V(G)$ by $x \equiv_a y$ means $x \equiv_i y$ for some i . They make explicit the structural aspects of distance-hereditary graphs by deriving the following characteristics.

Proposition 2.3 [8] Let u be a vertex of a connected distance-hereditary graph G and let h_u be the hanging at u of G . Let $R_1, R_2, \dots, R_r$ be the equivalence classes of the relation $\equiv_a$. Then, (a) the graph obtained from G by shrinking each R_j into one vertex is a tree rooted at u (for convenience, we call such a tree an equivalence-tree T_{h_u}); (b) each R_j induces a P_4 -free subgraph; (c) for each j , the family $\{N'(R_k) : N'(R_k) \subseteq R_j\}$ is an arboreal family of homogeneous subsets of R_j .

3 Minimum Weighted Steiner Tree

This section deals with the weighted Steiner tree problem on a distance-hereditary graph with the terminal vertex set K . Given a hanging h_u , where u is an arbitrary vertex of K , let T_{h_u} be the equivalence-tree defined in Proposition 2.3. Note that each node ω in T_{h_u} represents an equivalence class, denoted by R_ω . For each node ν in a rooted tree T , let $T(\nu)$ be the subtree rooted at ν . We present our algorithm below.

Algorithm WCD

Input: A distance-hereditary graph $G = (V, E)$ and a set $K \subset V$ composed of terminal vertices.

Output: A minimum weighted Steiner tree T on K .

Step 1. Let u be an arbitrary vertex in the target set K and let $U = K$. Determine a hanging h_u and construct the equivalence-tree T_{h_u} .

Step 2. For each node ω in T_{h_u} , if all of the following conditions (a) $\bigcup_{x \in V(T_{h_u}(\omega))} \{R_x \cap K\} \neq \emptyset$, (b) $N'(R_\omega)$ contains no vertex in K , and (c) $N'(R_\omega)$ contains no proper subset $N'(R_\nu)$, such that the node ν in T_{h_u} satisfies that $\bigcup_{x \in V(T_{h_u}(\nu))} \{R_x \cap K\} \neq \emptyset$; hold, then select a minimum weighted vertex $y \in N'(R_\omega)$ and add it to U .

Step 3. Find a spanning tree T for $\langle U \rangle$, which is the desired output.

Theorem 3.1 Algorithm WCD generates a minimum weighted Steiner tree for the distance-hereditary graph in $O(\log n)$ time using $O(n + m)$ processors on a CRCW PRAM.

Proof: For each vertex $w (\neq u) \in K$, it can be verified that there is a path, connecting w and u , in $\langle U \rangle$ after Step 2. Thus $\langle U \rangle$ is a connection of K . We then show that $\langle U \rangle$ has the minimum weight. Observe that for each node ω in T_{h_u} satisfying conditions (a), (b) and (c) in step 2 of algorithm WCD, a minimum weighted connection of K must contain a vertex of $N'(R_\omega)$. Since the algorithm selects a vertex with the minimum weight from ω , the weight of U is clearly minimum. After executing Step 3, the algorithm finds an arbitrary spanning tree T for $\langle U \rangle$. By definition, T is a minimum weighted Steiner tree.

We now analyse the time-processor complexities. In Step 1, the implementation of a hanging is described in Fact 1. We note that the computation of T_{h_u} can be done by finding the equivalence classes with respect to h_u . It takes $O(\log n)$ time using $O(n + m)$ processors on a CRCW PRAM. In step 2, associate each vertex v with $tag(v)$ such that $tag(v) = 1$ for $v \in K$ and $tag(v) = 0$ otherwise. Using the above information, the Euler-tour technique [10] can be applied to check the condition (a) which takes $O(\log n)$ time with $O(n/\log n)$ processors on an EREW PRAM. Condition (b) can be checked in $O(\log n)$ time using $O(n/\log n)$ processors on an EREW PRAM by slightly modifying optimal parallel prefix-sum computation [10]. A method to check condition (c) can be done similar to the implementation of the algorithm in Section 3 for finding a connected dominating set. In Step 3, finding a spanning tree can be done in $O(\log n)$ time using $O((m+n)\alpha(m, n)/\log n)$ processors on a CRCW PRAM [4].

Q.E.D.

4 Maximum weighted Clique

It is easy to see that any clique of a distance-hereditary graph G belongs to $L_i \cup L_{i+1}$ for some $0 \leq i \leq t-1$. Let R denote an equivalence class of G with respect to h_u .

Observation 1 *A maximum weighted clique of a distance-hereditary graph is a maximum weighted clique of some $\langle R \cup N'(R) \rangle$.*

We call $\langle R \cup N'(R) \rangle$ a candidate subgraph of a maximum weighted clique. For abbreviation, we also call it is a candidate subgraph. Since there are at most n equivalence classes for G , we have the following observation.

Observation 2 *There are $O(n)$ candidate subgraphs for a distance-hereditary graph.*

For two graphs G_1 and G_2 , the join of G_1 and G_2 , denoted by $G_1 \oplus G_2$, is the graph consists of $G_1 \cup G_2$ and all edges joining $V(G_1)$ and $V(G_2)$ [12]. From structure characteristics of G , every candidate subgraph $\langle R \cup N'(R) \rangle = \langle R \rangle \oplus \langle N'(R) \rangle$. If C_1 and C_2 are maximum weighted cliques of R and $N'(R)$ respectively, then $C_1 \oplus C_2$ is a maximum weighted clique of $\langle R \cup N'(R) \rangle$. By Observation 2 and 1, a maximum weighted clique of G is a clique with the maximum weight among $O(n)$ candidate subgraphs.

It is assured by Proposition 2.3 that $\langle R \rangle$ and $\langle N'(R) \rangle$ are cographs. In [1], Abrahamson, et. al. present an algorithm to find a maximum weighted clique of a cograph G : Given a cotree T of G a maximum weighted clique can be found by applying the tree contraction technique to T which takes $O(\log n)$ time using $O(n/\log n)$ processors on an EREW PRAM. Hence, if cotrees T_1 and T_2 for $\langle R \rangle$ and $\langle N'(R) \rangle$ are given, a maximum weighted clique of the candidate subgraph $\langle R \cup N'(R) \rangle$ can be computed by applying the algorithm in [1] to T_1 and T_2 . But it is necessary to construct cotrees for each R and $N'(R)$, the above algorithm takes a large number of processors to achieve the desired computation. Here we provide a different way (described latter) which requires fewer processors than the above one.

For any node ν in a tree T , let $leaf(\nu)$ be the leaves of $T(\nu)$ (recall that $T(\nu)$ is the subtree of T rooted at

ν). Let $\mathcal{S}_R = \{S \mid \text{there is an equivalence class } R' \text{ with } N'(R') = S \text{ and } S \subseteq R\}$. By the algorithm in [1], given a cotree T all maximum weighted cliques of cographs induced by $leaf(\nu)$, where ν is an internal node of T , can be found simultaneously in $O(\log n)$ time using $O(n/\log n)$ processors on an EREW PRAM (see [1] for details). Based on this result, we concentrate our effort to construct a special cotree T_R for $\langle R \rangle$ such that for each $S \in \mathcal{S}_R$ there exists a node ν in T_R with $leaf(\nu) = S$ and $T_R(\nu)$ is a cotree of $\langle S \rangle$. For convenience, we call T_R a canonical cotree.

Lemma 4.1 *If a canonical cotree of $\langle R \rangle$ is given, all maximum weighted cliques of $\langle S \rangle$, where $S \in \mathcal{S}_R$, can be computed in $O(\log |R|)$ time using $O(|R|/\log |R|)$ processors on an EREW PRAM.*

In the following, we show how to construct the canonical cotree for $\langle R \rangle$. By Proposition 2.3, $\mathcal{S}_R$ is the arboreal family of homogeneous subsets of R . Let $\mathcal{S}'_R = \mathcal{S}_R \cup \{R\}$. According to the partial order $\preceq$ defined in Section 2, we first construct an auxiliary tree structure A_R as follows. We create $|\mathcal{S}'_R|$ nodes where each node represents an element (a set) in $\mathcal{S}'_R$. Let S_ν denote the set represented by the node ν in A_R , and $par(\nu)$ (respectively, $child(\nu)$) be the parent (respectively, children) of ν . For any two created nodes α and β , there is a directed edge from α to β if $S_\alpha \preceq S_\beta$ and no other S_γ satisfies $S_\alpha \preceq S_\gamma \preceq S_\beta$.

Lemma 4.2 *The auxiliary tree structure A_R can be constructed in $O(\log p)$ time using $O(p + q)$ processors on an EREW PRAM, where $p = |R|$ and $q = |E(\langle R \rangle)|$.*

Proof: Suppose $R = \{v_1, v_2, \dots, v_p\}$ and $\mathcal{S}_R = \{S_1, S_2, \dots, S_r\}$. Let $S_i = N'(R_i)$ where R_i is an equivalence class. For ease of description, we assume the vertices are represented by their indices. So $R = \{1, 2, \dots, p\}$. We construct A_R as follows. Let u be any vertex of R_i and $S_i = \{t_1, t_2, \dots, t_k\}$, where $t_i \in \{1, 2, \dots, p\}$. Note that $N'(u) = S_i$. We assign $|S_i|$ processors to the edges $(u, t_1), (u, t_2), \dots, (u, t_k)$ for creating tuples $(t_1, i), (t_2, i), \dots, (t_k, i)$. Thus there are totally $\sum_{i=1}^r |S_i|$ created tuples that are stored in an array B .

We first sort those tuples according to their first indices. Then B is divided into p blocks (recall that $|R| = p$) $B_1, B_2, \dots, B_p$, where B_i records those tuples (i, a) 's for all a such that $i \in S_a$. We then sort the tuples in each B_i according to the cardinalities of those sets indicated by second indices. Suppose $(i, a_1), (i, a_2), \dots, (i, a_k)$ are the sorted elements in B_i and let α_i be the created node representing S_{a_i} . Set $par(\alpha_i) = \alpha_{i+1}$, $1 \leq i \leq k-1$ and let $par(\alpha_k)$ be the node representing R .

After executing the above algorithm, A_R can be constructed. The correctness follows directly from the property that if $S_p \preceq S_q$ then: (i) $|S_p| < |S_q|$ and (ii) $x \in S_p$ implies $x \in S_q$. Using the sorting algorithm described in [3], the implementation can be done in $O(\log p)$ time using $O(p + q)$ processors on an EREW PRAM. **Q.E.D.**

Lemma 4.3 *A canonical cotree T_R can be generated in $O(\log^2 p)$ time using $O(p + q)$ processors on a CREW PRAM, where $p = |R|$ and $q = |E(\langle R \rangle)|$.*

Proof: Given the auxiliary tree structure A_R , we generate a tree T_R in the following way. Let us consider the cograph $\langle S_\nu \rangle$ for each node $\nu \in A_R$. We first shrink each $\langle S_\chi \rangle$ in $\langle S_\nu \rangle$, where $\chi \in \text{child}(\nu)$, into a vertex s_χ called *shrunked vertex*. Since S_χ is a homogeneous set of $\langle S_\nu \rangle$, the resulting graph, denoted by H , is still a cograph. Applying the algorithm in [6] to H , a cotree T_H can be constructed for H in $O(\log^2 p)$ time using $O(p+q)$ processors on a CREW PRAM. Note that each shrunked vertex is a leaf in T_H . There are totally $|S'_R|$ constructed cotrees. For each leaf of the generated cotrees, if it is a shrunked vertex s_χ , we replace it with the "corresponding" cotree T_χ . It takes a constant time using $O(p)$ processors.

We now show that T_R is the canonical cotree of $\langle R \rangle$. Clearly, T_R has the following properties: (1) the leaves of T_R are all vertices of R , (2) each internal nodes are either 0 node or 1 node, and (3) if two vertices x and y of $\langle R \rangle$ are adjacent, the least common ancestor of x and y in T_R is a 1 node (this property can be verified from the fact that S_R is a family of homogeneous sets of R). Moreover, from the construction of T_R , for each $S \in S_R$ there exists a node ν in T_R such that $\text{leaf}(\nu) = S$. Hence T_R is a canonical cotree of $\langle R \rangle$. **Q.E.D.**

Remark: Note T_R is not the standard form of the cotree since it violates the definition (c) (refer to Section 2.1) that 0 nodes and 1 nodes alternate along every path starting from the root. For an internal node ν of a cotree with children $\mu_1, \mu_2, \dots, \mu_k$, let $G(\nu)$ be the cograph induced by $\text{leaf}(\nu)$ and G_{μ_i} be the cograph induced by $\text{leaf}(\mu_i)$. From the definition (d), if ν is a 1 (respectively, 0) node, $G(\nu) = G(\mu_1) \oplus G(\mu_2) \oplus \dots \oplus G(\mu_k)$ (respectively, $G(\nu) = G(\mu_1) \cup G(\mu_2) \cup \dots \cup G(\mu_k)$). According to the above observation, the definition (c) can be relaxed (note that this kind of cotree representation is no longer unique).

Theorem 4.4 *The maximum weighted clique problem for distance-hereditary graphs can be solved in $O(\log^2 n)$ time using $O(n+m)$ processors on a CREW PRAM.*

Proof: It takes $O(\log n)$ time using $O(n+m)$ CRCW processors to determine a hanging and compute the equivalence classes by Fact 1 and Lemma 5.2. By Observation 2, we have $O(n)$ candidate subgraphs, i.e., $O(n)$ candidate cliques. According to Lemma 4.3, we compute canonical cotrees for those equivalence classes of G in $O(\log n^2)$ time using $O(n+m)$ processors on a CREW PRAM. By applying the algorithm in [1] to canonical cotrees, $O(n)$ candidate cliques can be found in $O(\log n)$ time using $O(n/\log n)$ processors on an EREW PRAM. Thus we can find the one with the largest total weight in $O(\log n)$ time using $O(n/\log n)$ processors [10]. **Q.E.D.**

5 Minimum Weighted Connected Dominating Set

Given a hanging h_u , let $R_1, R_2, \dots, R_r$ be those equivalence classes of the relation $\equiv_{l+1}$ (see Proposition 2.3). Let $\mathcal{S} = \{S \subseteq L_l \mid \text{there exists an } R_i \text{ with } N'(R_i) = S\}$. For convenience, we call $\mathcal{S}$ the *upper-neighborhood system* of L_{l+1} . By Proposition 2.3, $\mathcal{S}$ is arboreal. So the partial order $\preceq$ described in Section 2 can be defined on $\mathcal{S}$. Let $\mathcal{U} = \{S \in \mathcal{S} \mid \text{there exists no } S' \in \mathcal{S}, S \not\preceq S'\}$, i.e., $\mathcal{U}$

is the set composed of those minimal elements of $\preceq$. We call $\mathcal{U}$ the *minimal upper-neighborhoods* of L_{l+1} . For each $S \in \mathcal{U}$, we select a vertex $x_s \in S$ with the minimum weight. Let $D_l = \{x_s \mid S \in \mathcal{U}\}$.

Lemma 5.1 [11] *Suppose $G = (V, E)$ is a connected distance-hereditary graph with a non-negative weight function w on vertices. Let u be a vertex of minimum weight and $h_u = (L_0, L_1, \dots, L_t)$ be the hanging at u . Then, one of the following three sets of vertices is a minimum weighted connected dominating set of G : (1) $\bigcup_{i=1}^{t-1} D_i$; (2) $\bigcup_{i=1}^{t-1} D_i \cup \{u\}$; (3) some vertex of G .*

Lemma 5.2 *Given a hanging h_u , D_l , for all $1 \leq l \leq t-1$, can be computed in $O(\log n)$ time using $O(n+m)$ processors on a CRCW PRAM.*

Proof: We first compute the auxiliary tree structure A_R for each $R \subseteq L_l$ (see Lemma 4.2). Let $\text{leaf}(A_R)$ denote the leaves of A_R . So $\mathcal{U} = \{S \mid S \in \text{leaf}(A_R), R \subseteq L_l\}$ (recall that $\mathcal{U}$ is the minimal upper-neighborhood of L_{l+1}). Clearly, D_l can be determined from $\mathcal{U}$. **Q.E.D.**

Theorem 5.3 *A minimum weighted connected dominating set for distance-hereditary graphs can be found in $O(\log n)$ time using $O(n+m)$ processors on a CRCW PRAM.*

References

- [1] K. Abrahamson, N. Dadoun, D. G. Kirkpatrick, and T. Przytycka, "A simple parallel tree contraction algorithm," *Journal of Algorithms*, vol. 10, no. 2, pp. 287-302, 1989.
- [2] H. J. Bandelt and H. M. Mulder, "Distance-hereditary graphs," *Journal of Combinatorial Theory Series B*, vol. 14, no. 1, pp. 182-208, 1986.
- [3] R. Cole, "Parallel merge sort," *SIAM Journal on Computing*, vol. 17, no. 4, pp. 770-785, 1988.
- [4] R. Cole and R. Thurimella, "Approximate parallel scheduling, II: application to optimal parallel graph algorithms in logarithmic time," *Information and Computation*, vol. 92, no. 1, pp. 1-47, 1991.
- [5] D. G. Corneil, H. Lerchs, and L. S. Burlingham, "Completeness reducible graphs," *Discrete Applied Mathematics*, vol. 3, pp. 163-174, 1981.
- [6] E. Dahlhaus, "Efficient parallel recognition algorithms of cographs and distance-hereditary graphs," *Discrete Applied Mathematics*, vol. 57, pp. 29-44, 1995.
- [7] A. D'atri and M. Moscarini, "Distance-hereditary graphs, steiner trees, and connected domination," *SIAM Journal on Computing*, vol. 17, no. 3, pp. 521-538, 1988.
- [8] P. L. Hammer and F. Maffray, "Complete separable graphs," *Discrete Applied Mathematics*, vol. 27, pp. 85-99, 1990.
- [9] E. Howorka, "A characterization of distance-hereditary graphs," *Quarterly Journal of Mathematics Oxford Series 2*, vol. 28, pp. 417-420, 1977.
- [10] J. Ja'Ja', *An Introduction to Parallel Algorithms*, Addison Wesley, 1992.
- [11] Hong-Gwa Yeh and Gerard J. Chang, "Weighted connected domination and Steiner trees in distance-hereditary graphs," manuscripts, 1994.
- [12] A. A. Zykov, "On some properties of linear complexes," *Mathematics Sbornik*, vol. 24, pp. 163-188, 1949.