An $O(n)$-time algorithm to compute minimum $k$-hop connected dominating set of interval graphs

Sambhu Charan Barman$^1$, Madhumangal Pal$^2$ and Sukumar Mondal$^3$

Abstract
For a graph $G = (V,E)$ and a fixed natural number $k$, a subset $D_k$ of $V$ is called a $k$-hop connected dominating set of $G$ if every node $a \in V$ is within $k$-steps from at least one vertex $b \in D_k$, i.e., $d(a,b) \leq k$, and the subgraph of $G$ induced by $D_k$ is connected. A dominating set $D_k$ is known to be minimal if there does not exist any $H \subset D_k$ such that $H$ is a dominating set of $G$. A dominating set $D_k$ is said to be minimum $k$-hop connected dominating set, if it is minimal as well as it is $k$-hop connected dominating set. In this paper, we propose an optimal algorithm to obtain a minimum $k$-hop connected dominating set of connected interval graphs having $n$ nodes that runs in $O(n)$ time.

Keywords
Design and analysis of algorithms, $k$-hop connected domination, interval graphs.

AMS Subject Classification
05C30, 05C12, 68R10, 68Q25.

1. Introduction

Interval graphs are more complex than a tree but simpler than a general graph. This graph is one of the most useful mathematical structures for modeling real life problems in graph theory. This graph is a subclass of intersection graph, chordal graph and perfect graph and superclass of permutation, circular-arc and trapezoid graph. Suppose $I = \{I_1, I_2, \ldots, I_n\}$, where $I_j = [a_j, b_j], r = 1, 2, \ldots, n$, be a family of intervals on a real line, where $a_j$ and $b_j$ represent respectively the left and right endpoints of the interval $I_j$. Also, an interval graph can be considered as an intersection graph of $I$. Interval graph has lots of applications in archeology, protein sequencing [13], molecular biology, file organization [5], sociology, circuit routine, genetics, job scheduling [5], traffic planning, VLSI design, circuit routing, transportation etc. A deep study on interval graphs is available in [9].

We represent a graph by the algebraic structure $G = (V,E)$, where $V$ and $E$ represent respectively the node/vertex set with cardinality $n$ and edge set with cardinality $m$. A subset $B \subseteq V$ is called a dominating set of a graph $G$ if each node in $V$ is either in $B$ or adjacent with at least one node in $B$. A dominating set with minimum cardinality is known as minimal dominating set. A dominating set $B$ of $G$ is known to be a connected dominating set if the subgraph made by $B$ is connected. In graph theory, Domination problems has several variations like connected domination, secure domination, edge domination, roman domination, weighted domination, independent domination, perfect domination, locating domination, paired-domination, $k$-tuple domination and $k$-hop domination [8, 14, 21], etc.

1. Introduction
graphs are found in [2, 15, 21]. A dominating set \( D_k \) is said to be minimum \( k \)-hop connected dominating set (in short \( MHCDS \)), if it is minimal as well as it is \( kHDS \) and the subgraph of \( G \) induced by \( D_k \) is connected.

Hop domination has several elegant applications in the real life like facility location problems, land surveying, kernels of games and communication networks, ad hoc networks [16, 23].

Domination is an important problem in graph theory. Claude Berge first introduce the concept of the domination number of a graph. A book on the topic of domination [10] lists over 1,200 papers related to domination in graphs. Many researchers have been studied domination and its variations for its versatile applications in real life.

Among the variations of domination, the \( k \)-hop domination has been briefly discussed in the literature, see [2, 15]. This problem is NP-complete in general graphs and some special subclass of graphs [4, 11, 12, 16, 23]. Demaine et al. [8] proposed an \( O(n^4) \)-time algorithm to compute \( k\)HDS of planar graphs, where \( n \) is the number of nodes. Furthermore, Ayyaswamy et al. [2] discussed about Bounds on the hop domination number of a tree. Also, Rana et al. [19] have planed an efficient algorithm to solve the distance \( k \)-domination problem on permutation graphs. Kundu et al. [14] have designed an \( O(n) \) time algorithm for computing an optimal \( k\)HDS of a tree. Barman et al. [3] have designed an \( O(n) \) time algorithm to compute minimum \( k\)HDS of interval graphs. Recently Adhya et al. [1] have designed an algorithm for computing minimum \( k \)-hop connected dominating set of permutation graphs in \( O(n) \) time.

In this paper, we propose an optimal algorithm to compute a minimum \( k \)-hop connected dominating set of the simple, undirected and connected interval graph in \( O(n) \) time, where \( n \) is the number of nodes of the interval graph. This problem is similar with the problem of computing a connected dominating set, but dissimilar in terms of number of hops required to reach all the nodes. We have generalized the problem as \( k \)-hop connected dominating set problem on interval graphs.

2. Preliminaries

An interval graph can be displayed by an interval representation where intervals are drawn in a real line. Without affecting the generality, we presume that each interval consist of both of its extremities/endpoint and that no two intervals share a common extremity/endpoint. If in some particular cases the intervals share common endpoints then we apply the algorithm CONVERT (see [18]) to convert the intervals of \( I \) into intervals with distinct endpoints. We also presume that the graph \( G \) is simple, connected & the sorted endpoints list is given. We have labeled the intervals in \( I \) according to the increasing right end points, that is \( b_1 < b_2 < \ldots < b_n \). This indexing is called interval graph (IG) ordering. An interval graph and its interval representation are displayed in Figure 1. For arbitrary interval graph \( G = (V, E) \), we recall a very important result presented in below.

Lemma 2.1. [17] In interval graph ordering, if the nodes \( x, y, z \in V \) are such that \( x \) is adjacent to \( z \) and \( x < y < z \), then \( y \) must be adjacent to \( z \). But \( y \) is not necessarily adjacent to \( x \).

![Figure 1](image)

Figure 1. An interval graph and its interval representation on real line.

3. Formation of BFS-tree & main path

In graph theory, there exist several important graph BFS is an important graph traveling technique. Among them BFS is very useful and popular graph traveling process. Using this process we can form a BFS-tree on arbitrary graphs. The BFS-tree can be formed on general graphs in \( O(n+m) \) time, where \( n \) and \( m \) indicate respectively the number of nodes and number of edges [22]. Recently, Barman et al. [3] have designed an Algorithm IBFS to construct a BFS tree \( T^*(1) \), with root as 1 of interval graphs in \( O(n) \) time, where \( n \) is the number of vertices. The BFS tree \( T^*(1) \) rooted at 1 of the interval graph of Figure 1 is shown in Figure 2. The level of each node on BFS-tree \( T^*(1) \) can be assigned by the BFS algorithm of Chen and Das [6].

Lemma 3.1. [6] The level of each node of the BFS tree \( T^*(1) \) can be computed, using BFS, in \( O(n) \) time.

In BFS tree \( T^*(1) \) with root as 1, let \( u_{last} \) be any vertex at level \( h \), where \( h \) is the height of tree \( T^*(1) \). Also we assume that the shortest path between 1 and \( u_{last} \) which is \( u_{last} \rightarrow parent(u_{last}) \rightarrow parent(parent(u_{last})) \rightarrow \cdots \rightarrow 1 \) as the main
path between 1 and \(u_{last}\).

Also, we assume that the vertex whose distance from 1 is \(r\) along the main path, is denoted by \(u_r^*\).

Now, we are going to present some notations that are needed through the paper.

\(u_r^*\): The node lying on the main path of \(T^*(1)\) at level \(r\).

\(N(z) = N(z) = \{i : (i, z) \in E\} \cup \{z\}\). Hence, the result.

\(N(z) = N(z) = \{i : (i, z) \in E\} \cup \{z\}\).

\(N_i\): The set of nodes at level \(r\) on \(T^*(1)\).

\(h\): The height of the BFS-tree \(T^*(1)\).

\(k\): A fixed natural number.

### 4. Main Results

Here we present some important results relating to MkHCDS of interval graphs.

**Lemma 4.1.** Each member of \(N_1\) is adjacent to other members of \(N_1\).

*Proof.* To prove this lemma we construct Figure 3. Let \(i\) and \(j\) be any two members of \(N_1\). Obviously, parent\((i) = parent(j) = 1\). So, \(a_i, a_j < b_1, b_i, b_j\). Hence, interval \(i\) intersects interval \(j\) in interval representation. So, \(i, j\) are adjacent. Therefore, each member of \(N_1\) is adjacent to other members of \(N_1\).

\[
\begin{array}{c}
  a_i \quad a_j \quad b_j \\
  \hline
  a_i \quad a_j \quad b_j \\
  \hline
  b_i
\end{array}
\]

**Figure 3.** A part of an interval representation on real line.

**Lemma 4.2.** If \(h \leq k\), then the subset \(D_k\) contains only one arbitrary node of \(V\) is a MkHCDS of interval graph \(G = (V, E)\).

*Proof.* Since \(h\) is the height of the BFS tree \(T^*(1)\) of the interval graph \(G = (V, E)\), \(h\) is the diameter of \(G\) Also, \(h \leq k\). So, \(d(u, v) \leq h \leq k\), for all \(u, v \in V\). Therefore, the subset \(D_k\) which contains only one node of \(V\) is a MkHCDS of \(G\). Hence, the result.

**Lemma 4.3.** If \(h \leq 2k\), then \(D_k = \{u_h^*\}\) is a MkHCDS of interval graph \(G = (V, E)\).

*Proof.* Let \(x_i\) be any member of \(X_i\) for \(r = 0, 1, 2, \ldots, h\). Now, we know by Lemma 3, that \(d(u_r^*, x_i) = 1\). So, \(d(u_r^*, x_i) \leq k\) (as \(u_r^* \rightarrow u_{r+1}^* \rightarrow \cdots \rightarrow u_i^* \rightarrow x_i\) or \(u_r^* \rightarrow u_{r+1}^* \rightarrow \cdots \rightarrow u_{r+k}^* \rightarrow \cdots \rightarrow u_i^* \rightarrow x_i\)). Also, \(d(u_r^*, x_i) \leq k + 2\) (as \(u_r^* \rightarrow u_{r+1}^* \rightarrow \cdots \rightarrow u_i^* \rightarrow x_i\) or \(u_r^* \rightarrow u_{r+1}^* \rightarrow \cdots \rightarrow u_{r+k}^* \rightarrow \cdots \rightarrow u_i^* \rightarrow x_i\), for \(i = 2, 3, 4, \ldots, (k - 1)\)). Again, \(d(u_r^*, u_h^*) = k\) (as \(u_r^* \rightarrow u_{r+k}^* \rightarrow \cdots \rightarrow u_h^*\)). Also, \(d(u_r^*, x_i) \leq 2\) (as \(u_r^* \rightarrow u_{r+k}^* \rightarrow x_i\) or \(u_k^* \rightarrow u_h^* \rightarrow x_i\)). Hence, \(d(u_r^*, x_i) \leq k\), for all \(x_i \in U_{i=0}^{h}N_i\). Again, we know that parent\((x) = u_{r-1}^*\), for all \(x_i \in N_i\), \(i = 1, 2, \ldots, h\). So, \(d(u_r^*, x_i) \leq h - k \leq k\), for all \(x_i \in U_{i=0}^{h}N_i\). Therefore, \(d(u_r^*, x_i) \leq k\), for all \(x_i \in U_{i=0}^{h}N_i\). Hence, \(D_k = \{u_h^*\}\) is a MkHCDS of interval graph \(G = (V, E)\).

**Lemma 4.4.** If \(h \geq 2k\), then \(d(u_r^*, x_i) \leq k\) for all \(x_i \in U_{i=0}^{h}N_i\).

*Proof.* Let \(x_i\) be any member of \(X_i\) for \(r \geq 1, 2, \ldots, h\). Now, \(d(u_r^*, x_i) = k\) (as \(u_0^* \rightarrow u_1^* \rightarrow u_2^* \rightarrow \cdots \rightarrow u_k^*\)). Also, we know that \(x_i \in U_{i=0}^{h}N_i\) for all \(x_i \in X_i\) (by Lemma 4.1), so, \(d(u_r^*, x_i) \leq k\) (as \(x_i \rightarrow u_0^* \rightarrow u_1^* \rightarrow u_2^* \rightarrow \cdots \rightarrow u_k^*\)). Furthermore, \(d(u_r^*, x_i) \leq k + 2\) (as \(x_i \rightarrow u_0^* \rightarrow u_1^* \rightarrow u_2^* \rightarrow \cdots \rightarrow u_k^*\) or \(x_i \rightarrow u_0^* \rightarrow u_1^* \rightarrow u_2^* \rightarrow \cdots \rightarrow u_k^*\) or \(x_i \rightarrow u_0^* \rightarrow u_1^* \rightarrow u_2^* \rightarrow \cdots \rightarrow u_k^*\) for \(i = 2, 3, \ldots, (k - 1)\). Again, \(d(u_r^*, x_i) \leq 2\) (as \(x_i \rightarrow u_k^* \rightarrow u_{k-1}^*\) or \(x_i \rightarrow u_k^* \rightarrow u_{k-1}^*\)). Therefore, \(d(u_r^*, x_i) \leq k\) for all \(x_i \in U_{i=0}^{h}N_i\).

Also from the above result one may conclude another result which is stated below.

**Corollary 4.5.** If \(h \geq 2k\), then \(u_h^*\) is a possible member of MkHCDS.

**Lemma 4.6.** If \(u_h^*\) is selected as a member of \(D_k\) and \(h > 2k\), then \(u_{h-k}^*\) is a possible member of \(D_k\).

*Proof.* From the Lemma 4.4, it is clear that \(d(u_r^*, x_i) \leq k\) for all \(x_i \in U_{i=0}^{h}N_i\). But, \(h > 2k\), so, the vertices of the set \(U_{i=0}^{h}N_i\) are not within \(k\) steps from \(u_r^*\). Therefore, we have to select one or more other vertices (excluding \(u_r^*\)) of \(V\) to find \(D_k\). Let us assume that \(x_i\) is any member of \(N_i\) for \(i = 1, 2, \ldots, h\). Now, we know that parent\((x_i) = u_{r-1}^*\) for \(i = 1, 2, \ldots, h\). So, \(d(u_r^*, x_i) \leq k\) (as \(u_r^* \rightarrow u_{r+k}^* \rightarrow \cdots \rightarrow u_h^* \rightarrow \cdots \rightarrow u_h^* \rightarrow x_i\) for \(i = 1, 2, \ldots, k\). Therefore, the vertices of \(N_h\) are within \(k\) steps from \(u_{r-k}^*\). Also, \(d(u_r^*, x_i) \leq k\) for all \(x_i \in U_{i=0}^{h}N_i\). So, if \(h > 2k\), then \(u_{r-k}^*\) is a possible member of \(D_k\).

**Lemma 4.7.** If \(h > 2k\), then \(D_k = \{u_r^*, u_{r+1}^*, u_{r+2}^*, \ldots, u_{r-k}^*\}\) is a \(h\)-hop connected dominating set of the interval graph \(G = (V, E)\).

*Proof.* We know by Lemma 4.3, that if \(h \leq 2k\) then \(D_k = \{u_r^*\}\) is a minimum \(h\)-hop connected dominating set of interval graph \(G = (V, E)\). Again, by Corollary 4.5, if \(h \geq 2k\), then \(u_r^*\) is a possible member of \(D_k\). Also, by Lemma 4.6, it is clear that if \(h < 2k\), then \(u_{r-k}^*\) is a possible member of \(D_k\). So, to compute a minimum \(h\)-hop dominating set \(D_k\), we have to select other vertices of \(V\) in such a way that the subgraph of the interval graph \(G\) is induced by \(D_k\) is connected. Now, it is obvious that \(u_r^* \rightarrow u_{r+k}^* \rightarrow \cdots \rightarrow u_{r-k}^*\) is a shortest path between \(u_r^*\) and \(u_{r-k}^*\). So, if we select \(u_r^*, u_{r+1}^*, u_{r+2}^*, \ldots, u_{r-k}^*\) as the member of \(D_k\), then all the vertices of \(V\) are within \(k\)
steps from at least one vertex of $D_k$. Also, the subgraph of the interval graph $G$ is induced by $D_k$ is connected. Hence, $D_k = \{u_k^*, u_{k+1}^*, u_{k+2}^*, \ldots, u_{n-k}^*\}$ is a $k$-hop connected dominating set of $G$. □

5. Algorithm and its Complexity

From the results discussed in Section 4, it is observed that if $h \geq k$, then $u_k^*$ is selected as a member of $D_k$. Also, we observed that we can take the vertex $u_{h-k}^*$ as a member of $D_k$. All possible cases for selection of the members of $D_k$ are already presented in terms of lemmas. Now, we are ready to present the complete algorithm to find a M\&HCDS $D_k$ of interval graphs.

Algorithm MINKHCDS

Input: Interval representation of an interval graph $G$.

Output: Minimum $k$-hop connected dominating set $D_k$ of the interval graph $G$.

Initially $D_k = \emptyset$ (empty set) and $u_0^* = 1$, the root of the BFS-tree $T^*(1)$.

Step 1: Construct the BFS-tree $T^*(1)$.

Step 2: Compute the nodes on the main path of the BFS-tree $T^*(1)$ and let them be $u_r^*, r = 0, 1, 2, \ldots, h$.

Step 3: Compute the sets $N_i, i = 0, 1, 2, \ldots, h$.

Step 4: If $h \leq k$ then $D_k = D_k \cup \{z\}$, where $z$ is only one arbitrary node of $V$. (by Lemma 4.2)

else if $h \leq 2k$ then $D_k = D_k \cup \{u_k^*\}$, (by Lemma 4.3)

else $D_k = D_k \cup \{u_k^*, u_{k+1}^*, \ldots, u_{n-k}^*\}$. (by Lemma 4.7)

end MINKHCDS.

Applying the Algorithm MINKHCDS, we obtain a minimum $3$-hop dominating set $D_k = \{8, 10, 11\}$ of the interval graph of Figure 1.

Lemma 5.1. The set $D_k$ is a minimum $k$-hop connected dominating set.

Proof. In Algorithm MINKHCDS, we find $D_k = \{x\}$, where $x$ is any member of $V$, when $h \leq k$. So, in that case, $D_k$ is a M\&HCDS of $G$. Again, if $h \leq 2k$ then we compute $D_k = \{u_k^*\}$, which also a M\&HCDS of $G$. Furthermore, when $h > 2k$, then, we find $D_k = \{u_k^*, u_{k+1}^*, u_{k+2}^*, \ldots, u_{n-k}^*\}$. In that case, we first select $u_k^*$ as a first member of $D_k$ in such a way that maximum number of nodes (union $\bigcup_{i=0}^{h} N_i$) are within $k$-steps from $u_k^*$ (by Lemma 4.4). Again, we select $u_{h-k}^*$ as a last member of $D_k$ because $d(u_{h-k}^*, z) = k$, for all $z \in N_h$. Now, we select the other vertices of $D_k$ (excluding $u_k^*$ or $u_{k-1}^*$ and $u_{h-k}^*$) in such a way that each vertex of $V$ is within $k$ steps from at least one vertex of $D_k$ and the subgraph of $G$ induced by $D_k$ is connected. Therefore, $D_k$ is a minimum $k$-hop connected dominating set. □

Theorem 5.2. The time complexity to compute a minimum $k$-hop connected dominating set $D_k$ of an undirected and connected interval graph is $O(n)$, where $n$ is the cardinality of vertex set of interval graph.

Proof. In Step 1, the BFS-tree $T^*(1)$ can be constructed in $O(n)$ time [3]). Since the main path contains only $h + 1$ vertices, so the main path (in Step 2) can be marked in $O(n)$ time. Step 3, i.e., the sets $N_i, i = 0, 1, 2, \ldots, h$ can be computed in $O(n)$ time as $N_i, i = 0, 1, 2, \ldots, h$ are mutually exclusive. Time complexity is needed to execute Step 4 is $O(n)$ as the vertices of $D$ are some vertices of the main path. Hence, overall time complexity is $O(n)$.

6. Conclusion

Domination plays an important role in graph theory. Among the variations of dominations, connected domination has many real life applications. Now a days, minimum $k$-hop connected domination problem is an important research topic in graphs theory because it has applications in communication networks like, wireless mobile networks. In this paper, we proposed an $O(n)$ time algorithm to obtain a M\&HCDS of undirected and connected interval graphs. We feel that it would be interesting to design an $O(n)$ time algorithm to find a minimum $k$-hop connected dominating set of weighted interval graphs or trapezoid graphs or social network graphs or other intersection graphs.

Acknowledgment

We thank the editors and anonymous reviewers for their valuable comments and suggestions to improve our paper.

References

[1] A. S. Adhya, S. Mondal and S. C. Barman: An optimal algorithm to find minimum $k$-hop connected dominating set of permutation graphs, Asian-European Journal of Mathematics, (2020) doi:10.1142/S17935557121500492 (2020).

[2] S. K. Ayyaswamy, B. Krishnakumari, C. Natarajan and Y. B. Venkatakrishnan, Bounds on the hop domination number of a tree, Proceedings of Mathematical Sciences, Indian Academy of Science, 125 (4)(2015), 449-455.

[3] S. C. Barman, M. Pal and S. Mondal, An optimal algorithm to find minimum $k$-hop dominating set of interval graphs, Discrete Mathematics, Algorithms and Applications, 11(2) 1950016(2019) DOI:10.1142/S1793830919500162.

[4] P. Basuchowdhuri and S. Majumder, Finding influential nodes in social networks using minimum $k$-hop dominating set, in Proceedings of Int. Conference on Appl. Algorithms, 8321(2014) 137 - 151.

[5] M. c. Carlisle and E. l. Lloyd, On the $k$-coloring of intervals, Discrete Applied Mathematics, 59 (1995), 225 - 235.

[6] C. C. Chen and S. K. Das, Breadth-first traversal of trees and integer sorting in parallel, Infomation Processing Letters, 41 (1992), 39 - 49.
An $O(n)$-time algorithm to compute minimum $k$-hop connected dominating set of interval graphs

[7] C. F. De Jaenisch, *Trait des applications de l’analyse mathematique au jeu des ecchecs*, Petrograd, (1862).

[8] E. D. Demaine, M. Hajaghay and D. M. Thilikos, Fixed parameter algorithm for $(k, r)$ center in planar graphs and map graphs, *ACM trans. Algorithm*, 1(1)(1862), 1 - 16.

[9] M. C. Golumbic, *Algorithmic Graph Theory and Perfect Graphs*, Academic Press, New York, (1980).

[10] T. W. Haynes, S. T. Hedetniemi and P. J. Slater, *Fundamentals of Domination in Graphs*, Marcel Dekker, New York, (1998).

[11] M. A. Henning and N. Jafari Rad, On 2-Step and Hop Dominating Sets in Graphs, *Graphs and Combinatorics*, 33 (4) (2017), 913 - 927.

[12] M. F. Jalavand and N. Jafari, On the complexity of $k$-step and $k$-hop dominating sets in graphs, *Mathematica Montisnigri Mathematics*, 12 (2017), 36 - 41.

[13] J. R. Jungck, G. Dick and A. G. Dick, Computer assisted sequencing, interval graphs and molecular evolution, *Biosystem*, 15 (3)(1982), 259-273.

[14] S. Kundu and S. Majumder, A linear time algorithm for optimal $k$-hop dominating set of a tree, *Information Processing Letters*, 116(2016), 197 - 202.

[15] C. Natarajan and S.K. Ayyaswamy, Hop Domination in Graphs-II, *An. Stt. Univ. Ovidius Constanta*, 23 (2)(2015), 187 - 199.

[16] T. N. Nguyen and D. T. Huynh, Connected d-hop dominating sets in mobile ad hoc networks. *Proceedings of Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks*, (2006), 1-8.

[17] S. Olariu, An optimal greedy heuristic to color interval graphs, *Information Processing Letters*, 37(1991) 21 - 25.

[18] M. Pal and G. P. Bhattacharjee, The parallel algorithms for determining edge-packing and efficient edge dominating sets in interval graphs, *Parallel Algorithms and Applications*, 7(1995), 193 - 207.

[19] A. Rana, A. Pal and M. Pal, An efficient algorithm to solve the distance $k$-domination problem on permutation graphs, *J. Discrete Mathematical Sciences and Cryptography*, 19(2)(2016), 241-255.

[20] M. Q. Rieck, S. Pai and S. Dhar, Distributed routing algorithms for multi-hop ad hoc networks using d-hop connected d-dominating sets. *Computer networks*, 47(6) (2005), 785-799.

[21] P. J. Slater, R-domination in graphs, *J. ACM*, 23 (2) (1976), 446 - 450.

[22] R. E. Tarjan, Depth first search and linear graph algorithm, *SIAM J. Comput.*, 2 (1972), 146 - 160.

[23] T. H. P. Vuong and D. T. Huynh, Adapting d-hop dominating sets to topology changes in ad hoc networks. *Proceedings NinthInternational Conference on Computer Communications and Networks*, (2000), 348-353.