6-1988

Graph and Directed Graph Augmentation Problems

Zhuguo Mo
Western Michigan University

Follow this and additional works at: http://scholarworks.wmich.edu/dissertations

Part of the Computer Sciences Commons

Recommended Citation
Mo, Zhuguo, "Graph and Directed Graph Augmentation Problems" (1988). Dissertations. 2168.
http://scholarworks.wmich.edu/dissertations/2168
GRAPH AND DIRECTED GRAPH AUGMENTATION PROBLEMS

by

Zhuguo Mo

A Dissertation
Submitted to the
Faculty of The Graduate College
in partial fulfillment of the
requirements for the
Degree of Doctor of Philosophy
Department of Mathematics & Statistics

Western Michigan University
Kalamazoo, Michigan
June 1988
INFORMATION TO USERS

The most advanced technology has been used to photograph and reproduce this manuscript from the microfilm master. UMI films the original text directly from the copy submitted. Thus, some dissertation copies are in typewriter face, while others may be from a computer printer.

In the unlikely event that the author did not send UMI a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyrighted material had to be removed, a note will indicate the deletion.

Oversize materials (e.g., maps, drawings, charts) are reproduced by sectioning the original, beginning at the upper left-hand corner and continuing from left to right in equal sections with small overlaps. Each oversize page is available as one exposure on a standard 35 mm slide or as a 17" × 23" black and white photographic print for an additional charge.

Photographs included in the original manuscript have been reproduced xerographically in this copy. 35 mm slides or 6" × 9" black and white photographic prints are available for any photographs or illustrations appearing in this copy for an additional charge. Contact UMI directly to order.

UMI
Accessing the World's Information since 1938
300 North Zeeb Road, Ann Arbor, MI 48106-1346 USA

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Graph and directed graph augmentation problems

Mo, Zhuguo, Ph.D.
Western Michigan University, 1988
For my parents,
whose love and encouragement keep me smiling,
and for Sue-Firm,
who understands.
ACKNOWLEDGEMENTS

I wish to thank Professor Kenneth Williams, my dissertation advisor, for his guidance and encouragement. My thanks also go to many others in the Department of Mathematics and Statistics and the Department of Computer Science of Western Michigan University who have stimulated my interest in mathematics and computer science and have given their time and friendship freely. Finally, I would like to thank Professor Abdol Esfahanian of Michigan State University, Professor Alfred Boals, Professor Gary Chartrand, and Professor John Petro of Western Michigan University for their many helpful suggestions in the preparation of this manuscript.

Zhuguo Mo
# TABLE OF CONTENTS

CHAPTER

ACKNOWLEDGEMENTS ............................................. ii

I. PRELIMINARIES .................................................. 1
   Introduction .................................................. 1
   Basic Definitions and Notation .......................... 4

II. RADIUS r DOMINATION AND AUGMENTATIONS .... 10
   Radius r Domination ....................................... 11
   m-Centrix Radius r Augmentation ....................... 26

III. K DISJOINT MATCHINGS AND AUGMENTATIONS ... 35
   r-Factorability .............................................. 37
   K Disjoint Maximum Matchings .......................... 39
   K Disjoint 1-Factors Augmentation
     for Forests .............................................. 48

IV. (M, N, R1, R2)-TRANSITIVE AUGMENTATIONS .... 58
    Results on Tournaments .................................. 59
    Some Results on Graphs and Digraphs ............... 62
    (M, 1, R1, =)-Transitive Augmentations ............. 68

V. (r, s)-DOMINATION AND AUGMENTATIONS ............ 75
   Bounds on (r, s)-Domination Number .................. 76
   Computational Complexities ............................ 92

VI. ON BLOCK-COMPLETE GRAPHS ............................ 102

VII. SUMMARY ................................................... 112

BIBLIOGRAPHY .................................................. 114

iii

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
CHAPTER I
PRELIMINARIES

Introduction

Let $G^* = (V, E^*)$ be a complete weighted graph with non-negative real-valued cost function $c$ on the edges of $G^*$, $G$ be a spanning subgraph of $G^*$, and $P$ be a property defined for graphs. One may ask the following questions: "Is there a graph $H$ such that $G \subseteq H \subseteq G^*$ and $H$ has property $P$?" If the answer is yes, "Is there an efficient algorithm for finding such a graph $H$ with a minimum-cost set of edges?" The problem of finding a spanning supergraph $H$ of $G$ with minimum-cost set of edges so that $H$ has property $P$ is called a (weighted) graph augmentation problem with respect to $G^*$ and $P$. For unweighted graphs, the problem of finding a minimum spanning supergraph $H$ of $G$ having the desired property $P$ is called an (unweighted) graph augmentation problem (with respect to $P$). Digraph augmentation problems can be defined similarly. Alternatively, graph (digraph) augmentation problems are sometimes called graph (digraph) completion problems. This dissertation presents solutions for unweighted augmentation problems, and it provides some general techniques for solving augmentation problems.

Graph and digraph augmentation problems have been studied by many people. In 1973, Eswaran [15] solved the problem of
adding a minimum-cost set of arcs to a digraph $D_0$ so that there is a cycle which contains all the arcs of $D_0$. In the same year, Goodman and Hedetniemi [24] gave an $O(|V|^2)$ time algorithm to find the minimum number of edges which must be added to a tree so that the resulting graph has a hamiltonian cycle. Later, they improved the algorithm to $O(|V|)$. In 1976, Eswaran and Tarjan [16] proposed a theoretical framework for studying graph (di-graph) augmentation problems, gave well-known examples of such problems, and analyzed in detail the strong connectivity, bridge connectivity and biconnectivity augmentation problems. Since then a number of results about augmentation problems have been obtained. In 1977, Boesch, Suffel, and Tindell [7] characterized those graphs which span eulerian graphs and gave exact formulas for the number of edges which must be added to such graphs in order to obtain eulerian graphs. Further results about the three connectivity augmentation problems considered by Eswaran and Tarjan [16] were obtained by others. In 1977, Rosenthal and Goldner [46] presented an $O(|V| + |E|)$ time algorithm which, given a graph $G$, finds a smallest biconnectivity augmentation of $G$. Frederickson and Ja'Ja' [18] showed that the three augmentation problems considered in [16] are NP-complete in the restricted case of the graph being initially connected. They also obtained fast approximation algorithms which have constant worst-case performance ratio. Triangulation augmentation has been studied extensively and has not yet been solved (see [41][44][45]). In recent years, many augmentation problems
have been investigated, among them are $K$-edge-connected digraph augmentation for trees [35], mixed graph augmentation [26], interval graph augmentation [42], $(M, N)$-transitive augmentation [4][27][52], and $(M, N, R_1, R_2)$-transitive augmentation [5] etc.

Note that if the complete graphs $K_n$ have property $P$, then for any graph $G$ there exists a minimum spanning supergraph $H$ of $G$ having property $P$. In this dissertation, we focus on the properties of the complete graphs (digraphs). Given an arbitrary graph $G = (V, E)$ and a graphical property $P$, let $Aug(G, P)$ be the minimum number of edges to be added to $G$ so that the resulting graph has property $P$. $Aug(G, P)$ is called the augmentation number of $G$ with respect to the property $P$. We will write $Aug(G)$ for $Aug(G, P)$ if no confusion arises. By a minimum augmentation of a graph $G$ with respect to a property $P$ sometimes is meant a minimum set of edges to be added to $G$ so that the resulting graph has property $P$ and sometimes is meant a graph obtained from $G$ by adding a minimum set of edges with respect to the property $P$ depending on the context. Many definitions and notation defined for graphs in this dissertation can be modified in a natural way to give corresponding definitions and notation for digraphs. In the rest of the dissertation, we deal with only those properties $P$ for which $Aug(G, P)$ are well defined.

Given an arbitrary graph $G = (V, E)$ and a graphical property $P$, if the problem of determining whether $G$ has property $P$ is NP-hard, then the problem of finding $Aug(G, P)$ is NP-hard since
G has property P if and only if $\text{Aug}(G, P) = 0$. In the case that the problem of determining whether an arbitrary graph G has property P is NP-complete (or NP-hard), we may provide heuristic approximation algorithms for the corresponding augmentation problem. We may also restrict the domain of the graphs to some classes of graphs, say trees, bipartite graphs, planar graphs, triangulated graphs, block-complete graphs etc.

Basic Definitions and Notation

For terminologies and notation, we follow Behzad, Chartrand, and Lesniak-Foster [3] unless defined here. A graph G is a finite, nonempty set V together with a set E of two element subsets of (distinct) elements of V. Each element of V is referred to as a vertex and V itself as the vertex set of G; the members of the edge set E are called edges. As usual, $|S|$ denotes the cardinality of a set S. For a graph G, $|V|$ and $|E|$ are referred to as the order and size of G, respectively. The maximum degree and minimum degree of G are denoted by $\Delta(G)$ and $\delta(G)$ respectively. There is only one graph of order one (up to isomorphism) which is referred to as the trivial graph. A non-trivial graph then has order at least 2. The edge $e = \{u, v\}$ is said to join the vertices u and v. If $e = \{u, v\}$ is an edge of a graph G, then u and v are adjacent vertices while u and e are incident as are v and e. Furthermore, if $e_1$ and $e_2$ are distinct edges of G incident with a common vertex, then $e_1$ and $e_2$ are ad-
jacent edges. Sometimes, an edge \( e = \{u, v\} \) is denoted by \( uv \) or \( vu \).

A graph is complete if every two of its vertices are adjacent. A complete graph of order \( n \) is denoted by \( K_n \). An empty graph is a graph which has no edges. A graph \( G = (V, E) \) is bipartite, if it is possible to partition \( V \) into subsets \( V_1 \) and \( V_2 \) (called partite sets) such that every element of \( E \) joins a vertex of \( V_1 \) to a vertex of \( V_2 \). A complete bipartite graph \( G \) is a bipartite graph with the added property that if \( u \in V_1 \) and \( v \in V_2 \) then \( uv \in E(G) \). If \( |V_1| = n_1 \) and \( |V_2| = n_2 \), then this graph is denoted by \( K(n_1, n_2) \) or \( K_{n_1, n_2} \).

Let \( u \) and \( v \) be (not necessary distinct) vertices of a graph \( G \). By a \( u-v \) walk of \( G \) is meant a finite, alternating sequence of vertices and edges of \( G \), beginning with \( u \) and ending with \( v \), such that every edge is immediately preceded and succeeded by the two vertices with which it is incident. A \( u-v \) walk is closed or open depending on whether \( u = v \) or \( u \neq v \). A \( u-v \) path is a \( u-v \) walk in which no vertex is repeated; a single vertex \( u \) forms the trivial \( u-u \) path. A hamiltonian path of \( G \) is a path which contains all vertices of \( G \). In a graph \( G \), a closed walk \( v_1, v_2, \ldots, v_n, v_1 \) \((n \geq 3)\) whose \( n \) vertices \( v_i \) are distinct is called a cycle of \( G \). A cycle edge is an edge that lies on a cycle. An acyclic graph has no cycles. A hamiltonian cycle of \( G \) is a cycle which contains all vertices of \( G \). A graph \( G \) is said to be hamiltonian if it has a hamiltonian cycle.
Two distinct vertices or edges in a graph $G$ are independent if they are not adjacent in $G$. A set of pairwise independent edges of $G$ is called a matching in $G$, while a matching of maximum cardinality is a maximum matching in $G$. Let $M$ be a specified matching in a graph $G$. An edge $e$ of $G$ that is not in $M$ is called a weak edge (with respect to $M$). A weak vertex (with respect to $M$) is a vertex of $G$ incident only with weak edges. In a graph $G$, a nonempty set $U_1$ of $V(G)$, is said to be matched to a subset of $U_2$ of $V(G)$ disjoint from $U_1$ if there exists a matching $M$ in $G$ such that each edge of $M$ is incident with a vertex of $U_1$ and a vertex of $U_2$ and every vertex of $U_1$ is incident with an edge of $M$, as is every vertex of $U_2$. Let $U$ be a nonempty set of vertices of a graph $G$ and let its neighborhood $N(U)$ denote the set of all vertices of $G$ adjacent with at least one element of $U$. Then the set $U$ is said to be nondeficient (relative to $G$) if $|N(S)| > |S|$ for every nonempty subset $S$ of $U$. Otherwise, $U$ is said to be deficient.

A factor of a graph $G$ is a (possibly empty) spanning subgraph of $G$. If $G$ is expressed as the edge sum of factors of $G$, then this sum is called a factorization of $G$. An $r$-regular factor of a graph $G$ is referred to as an $r$-factor of $G$. If there exists a factorization of $G$ such that each factor is an $r$-factor (for a fixed $r$), then $G$ is $r$-factorable.

The set of positive integers is denoted by $\mathbb{I}^+$. Note that an unweighted graph is a weighted graph with all the weights of the edges being one. For a weighted graph $G = (V, E)$ with
weight function \( c(e) \in \mathbb{I}^+ \), \( e \in E \), the distance \( d_G(u, v) \) between two vertices \( u \) and \( v \) is the minimum of the weighted lengths of the \( u-v \) paths of \( G \) if there exists a \( u-v \) path in \( G \), otherwise the distance \( d_G(u, v) \) is defined to be infinity. The eccentricity \( e(v) \) of a vertex \( v \) of a weighted or unweighted connected graph \( G \) is the number \( \max_{u \in V(G)} d(u, v) \), where the max is taken over all the vertices \( u \in V(G) \). The radius, \( \text{rad } G \), is defined as \( \min_{v \in V} e(v) \) while the diameter, \( \text{diam } G \), is \( \max_{u, v \in V} d(u, v) \). If two graphs \( G_1 \) and \( G_2 \) are isomorphic, we write \( G_1 = G_2 \).

A tree is an acyclic connected graph and a forest is an acyclic graph. A rooted tree, \( T \), is a finite set of one or more vertices such that

(1) there is a specially designated vertex called the root of \( T \);

(2) the remaining vertices are partitioned into \( n \geq 0 \) disjoint sets \( T_1, T_2, \ldots, T_n \) where each of these sets is a rooted tree with the root of each of \( T_1, T_2, \ldots, T_n \) adjacent to the root of \( T \).

\( T_1, T_2, \ldots, T_n \) are called the subtrees of the root. The roots of subtrees of a vertex \( v \) are called the children of \( v \) and \( v \) is the parent of the children.

In a rooted tree, the level of a vertex is defined by initially letting the root be at level one. If a vertex is at level \( p \), then its children are at level \( p + 1 \). The height or depth of a rooted tree is defined to be the maximum level of any vertex in the rooted tree.
An assignment of colors to the edges of a nonempty graph $G$ so that adjacent edges are colored differently is an edge coloring of $G$ (an $n$-edge coloring if $n$ colors are used). The graph $G$ is $n$-edge colorable if there exists an $m$-edge coloring of $G$ for some $m \leq n$. The minimum $n$ for which a graph $G$ is $n$-edge colorable is its edge chromatic number (or chromatic index).

A directed graph or digraph $D$ is a finite nonempty set $V$ (of vertices) together with a set $A$ (disjoint from $V$) of ordered pairs of distinct elements of $V$. The elements of $A$ are called arcs of $D$. Let $u$ and $v$ be two distinct vertices of a digraph $D$. A $u$-$v$ path of $D$ means a finite alternating sequence

$$u = u_0, a_1, u_1, a_2, \ldots, u_{n-1}, a_n, u_n = v$$

of distinct vertices and arcs such that $a_i = (u_{i-1}, u_i)$ for $1 \leq i \leq n$. A vertex $v$ is said to be reachable from a vertex $u$ in a digraph $D$ if $D$ contains a $u$-$v$ path. For a (weighted) digraph $D = (V, A)$ with weight function $c(e) \in \mathbb{I}^+$, $e \in A$, the distance $d_D(u, v)$ is defined to be the minimum of the (weighted) lengths of the $u$-$v$ paths of $D$ if there exists a $u$-$v$ path in $D$, otherwise the distance $d_D(u, v)$ is defined to be infinity. The eccentricity $e(v)$ of a vertex $v$ of $D$ is defined as $e(v) = \max_{u \in V(D)} d(v, u)$. The radius $\text{rad } D$ is defined by $\text{rad } D = \min_{v \in V(D)} e(v)$. If a given digraph $D$ is weighted, the distance of two vertices of $D$ means the weighted distance unless otherwise specified.

A digraph $D$ is strongly connected or strong if for every two distinct vertices of $D$, each vertex is reachable from the other. A digraph $D$ is called symmetric if whenever $(u, v)$ is an
arc of D so too is \((v, u)\). There is a natural one-to-one correspondence between the set of symmetric digraphs and the set of graphs. A digraph is asymmetric or oriented if whenever \((u, v)\) is an arc of D, then \((v, u)\) is not an arc of D. A digraph D is said to be complete if for every two distinct vertices u and v of D, both \((u, v)\) and \((v, u)\) are arcs of D. By the underlying graph of a digraph D is meant the graph G obtained from D by deleting all directions from the arcs of D and deleting an edge from a pair of multiple edges if multiple edges should be produced. A tournament is an asymmetric digraph whose underlying graph is complete. An oriented tree is an oriented digraph whose underlying graph is a tree. An arborescence is an acyclic digraph with one vertex, the root, having no entering arcs, and all other vertices having exactly one entering arc.

For graphs \(G = (V, E)\), a linear time algorithm is an algorithm having time complexity \(O(|E| + |V|)\). Linear time algorithms for digraphs are defined similarly.
CHAPTER II
RADIUS r DOMINATION AND AUGMENTATIONS

Let $V$ be the vertex set of a (weighted) graph or (weighted) digraph, $v$ be a vertex in $V$, and $U$ be a nonempty subset of $V$. Then the distance from the set $U$ to the vertex $v$ is defined as $d(U, v) = \min_{u \in U} d(u, v)$. Decision problems studied or referenced in this chapter are stated as follows:

**Vertex Cover**
Instance: Graph $G = (V, E)$, positive integer $K \leq |V|$.
Question: Is there a vertex cover of size $K$ or less for $G$, i.e., a subset $V' \subseteq V$ with $|V'| \leq K$ such that for each edge $uv \in E$ at least one of $u$ and $v$ belongs to $V'$?

**Graph Dominating Set of Radius $r$**
Instance: Graph $G = (V, E)$, weight function $c(e) \in \mathbb{I}^+$ for $e \in E$, positive integers $r, K \leq |V|$.
Question: Is there a dominating set of radius $r$ with size $K$ or less for $G$, i.e., a subset $V' \subseteq V$ with $|V'| \leq K$ such that for all $u \in V - V'$ there is a $v \in V'$ for which $d(v, u) \leq r$?

**Digraph Dominating Set of Radius $r$**
Instance: Digraph $D = (V, A)$, weight function $c(e) \in \mathbb{I}^+$ for $e \in A$, positive integers $r, K \leq |V|$.
Question: Is there a dominating set of radius $r$ with size $K$ or less for $D$, i.e., a subset $V' \subseteq V$ with $|V'| \leq K$ such that for all
For unweighted graphs, the Dominating Set of Radius 1 problem is referred to as the Dominating Set problem. Garey and Johnson [21][22] have shown that the Vertex Cover problem and the Dominating Set problem are NP-complete even when G is a (connected) planar graph of maximum degree at most 3 and of minimum degree 1. Kariv and Hakimi [36] have proved that (the optimization version of) the Dominating Set of Radius r problem is NP-hard which follows directly from the NP-completeness of the Dominating Set problem. We show that the Dominating Set of Radius r problem and the Digraph Dominating Set of Radius r problem are NP-complete for any fixed r. Both problems can be
solved in linear time if the graphs (digraphs) are restricted to trees (oriented trees).

**Theorem 2.1** For any fixed positive integer \( r \), the Dominating Set of Radius \( r \) problem is NP-complete even in the case when the graph is an unweighted connected planar graph of maximum degree 3 and of minimum degree 1.

**Proof:** The Dominating Set of Radius \( r \) problem is in NP since a nondeterministic algorithm need only guess a subset \( V' \) of \( V \) and check in polynomial time that \( |V'| \leq K \) and that for all \( u \in V - V' \) there is a \( v \in V' \) for which \( d_G(v, u) \leq r \).

We transform the Vertex Cover problem to the Dominating Set of Radius \( r \) problem. Let \( G_1 = (V_1, E_1) \) be an unweighted connected planar graph of maximum degree at most 3 and of minimum degree 1, and \( K \) be a positive integer in an instance of the Vertex Cover problem. We will construct a connected planar graph \( G = (V, E) \) of maximum degree 3 and of minimum degree 1 such that there exists a subset \( V' \subseteq V \) with \( |V'| \leq |E_1| + K \) and for all \( u \in V - V' \) there is a \( v \in V' \) for which \( d_G(v, u) \leq r \) if and only if there is a subset \( V_0 \subseteq V_1 \) with \( |V_0| \leq K \) such that for each edge \( \{u, v\} \in E_1 \) at least one of \( u \) and \( v \) belongs to \( V_0 \).

Let \( G \) be the graph obtained from \( G_1 \) by replacing each edge \( x_iy_i \) by the graph \( H_i \) (see Figure 2.1) which can be obtained from a path \( P \) of length \( 3r + 1 \) with \( x_i \) and \( y_i \) as the end vertices by adding a \( u_i - v_i \) path of length \( r + 1 \), where \( u_i \) and \( v_i \) are vertices on \( P \) such that \( d_P(x_i, u_i) = d_P(y_i, v_i) = r \). Clearly \( G \) is a connected
planar graph of maximum degree 3 and of minimum degree 1 and for \( i = j \), \( V(H_i) \cap V(H_j) = \{x_i, y_i\} \cap \{x_j, y_j\} \).

\[
\begin{align*}
C_{2r+2} & \quad P_{r+1} & \quad x_i & \quad \cdots & \quad u_i & \quad \cdots & \quad v_i & \quad \cdots & \quad y_i & \quad P_{r+1}
\end{align*}
\]

Figure 2.1 Local replacement used in the proof of Theorem 2.1

Suppose that there is a subset \( V_0 \) of the vertices of \( G_1 \) with \(|V_0| \leq K\) such that each edge of \( G_1 \) is incident with a vertex of \( V_0 \). Then for each edge \( x_iy_i \in E_i, 1 \leq i \leq |E_i| \), either \( x_i \in V_0 \) or \( y_i \in V_0 \). Define vertex \( w_i \) to be \( v_i \) if \( x_i \in V_0 \), else to be \( u_i \). Let \( V' = V_0 \cup \{w_i \mid 1 \leq i \leq |E_i|\} \), then \(|V'| = |V_0| + |E_i| \leq K + |E_i|\).

For \( 1 \leq i \leq |E_i| \), either \( \{x_i, v_i\} \subseteq V' \) or \( \{y_i, u_i\} \subseteq V' \). It follows that for all \( u \in V - V' \), there is a \( v \in V' \) for which \( d_G(v, u) \leq r \).

Conversely, suppose that there exists a subset \( V' \subseteq V \) with \(|V'| \leq |E_i| + K\) such that for all \( u \in V - V' \), there is a \( v \in V' \) for which \( d_G(v, u) \leq r \). Without loss of generality, assume that \(|V'|\) is minimum. We claim that \( G_1 \) has a vertex cover of size at most \( K \), i.e., there is a subset \( V_0 \subseteq V_1 \) with \(|V_0| \leq K\) such that each edge of \( G_1 \) is incident with a vertex of \( V_0 \).

Since \( d_{H_i}(x_i, u_i) = d_{H_i}(y_i, v_i) = r \), each vertex in the cycle \( C_{2r+2} \) of length \( 2r + 2 \) in \( H_i \) has to be dominated within distance \( r \) by some vertex in \( V(H_i) \cap V' \). Note that at least two vertices are
needed to dominate the vertices in $C_{2r+2}$ within distance $r$. Therefore $|V(H_i) \cap V'| \geq 2$, for all $i$, $1 \leq i \leq |E_1|$. The equality $d_G(x_i, y_i) = 3r + 1$ implies that $|(V(H_i) - \{x_i, y_i\}) \cap V'| \geq 1$, $1 \leq i \leq |E_1|$. Observe that $\{x_i, y_i, u_i\}$ is a dominating set of radius $r$ for the graph $H_i$. By the minimality of $|V'|$, $|V(H_i) \cap V'| \leq 3$.

Furthermore, in the case that $|V(H_i) \cap V'| = 3$, we may assume that $\{x_i, y_i, u_i\} \subseteq V'$; for otherwise, we could modify $V'$ to have the desired property since $\{x_i, y_i, u_i\}$ is one of the triples of vertices in $H_i$ which has best potential to cover more vertices in $G$.

Now we show that for each $i$, $1 \leq i \leq |E_1|$, such that $|V(H_i) \cap V'| = 2$, we may modify $V'$ so that $V(H_i) \cap V' = \{x_i, v_i\}$ or $V(H_i) \cap V' = \{u_i, y_i\}$. Let $V(H_i) \cap V' = \{s_i, t_i\}$. We need to show that the set $\{s_i, t_i\}$ can be chosen in such a way that either $\{s_i, t_i\} = \{x_i, v_i\}$ or $\{s_i, t_i\} = \{u_i, y_i\}$. By the structure of the graph $H_i$, without loss of generality, suppose that $s_i$ is on the $x_i-u_i$ path while $t_i$ is on the $v_i-y_i$ path. (If necessary, $V'$ could be modified in this way, the modified $V'$ would be taken as $V'$.) Note that $|\{s_i, t_i\} \cap \{x_i, y_i\}| \leq 1$. If $|\{s_i, t_i\} \cap \{x_i, y_i\}| = 1$, for simplicity, suppose that $\{s_i, t_i\} \cap \{x_i, y_i\} = x_i$, then $\{s_i, t_i\} = \{x_i, v_i\}$. So we may assume that $\{s_i, t_i\} \cap \{x_i, y_i\} = \emptyset$. Note that $x_i$ is an end-vertex in $G$ if and only if $x_i$ is an end-vertex in $G_1$. If $x_i$ is an end-vertex in $G_1$, then we may choose the set $\{s_i, t_i\}$ to be $\{u_i, y_i\}$. Suppose that $x_i$ is not an end-vertex in $G_1$. For each edge $x_jy_j$ ($j \neq i$) that is incident with $x_i$ in $G_1$, we have $|V(H_j) \cap V'| = 2$. Without loss of generality, suppose that $x_j = x_i$. Let $V(H_j) \cap V'$
= \{s_j, t_j\}$. Then either $s_j$ or $t_j$ is on the $x_j-u_j$ path. Therefore \{s_i, t_i\} could be replaced by \{u_i, y_i\} if necessary.

Thus there exists a subset $V' \subseteq V$ with $|V'| \leq |E_i| + K$ such that for all $u \in V-V'$, there is a $v \in V'$ for which $d_G(v, u) \leq r$. Furthermore for all $i$, $1 \leq i \leq |E_i|$, $|(V(H_i) - \{x_i, y_i\}) \cap V'| = 1$ and either $\{x_i, v_i\} \subseteq V(H_i) \cap V'$ or $\{u_i, y_i\} \subseteq V(H_i) \cap V'$. Let $V_0 = V' \cap V(G_i) = V' \cap \{x_i, y_i \mid 1 \leq i \leq |E_i|\}$. Then $V_0 \subseteq V(G_i)$, $|V_0| = |V'| - |E_i| \leq K$, and each edge of $G_i$ is incident with a vertex of $V_0$, i.e., $G_i$ has a vertex cover $V_0$ of size at most $K$. □

Theorem 2.2 For any fixed positive integer $r$, the Digraph Dominating Set of Radius $r$ problem is NP-complete even in the case when the digraph is unweighted, symmetric, and its underlying graph is connected planar of maximum degree 3 and of minimum degree 1.

Proof: It is easy to see that the Digraph Dominating Set of Radius $r$ problem is in NP. Let $G = (V, E)$ be an unweighted connected planar graph of maximum degree 3 and of minimum degree 1 in an instance of the Dominating Set of Radius $r$ problem and let $D = (V, A)$ be the symmetric digraph corresponding to $G$. Then $D$ is an unweighted symmetric digraph whose underlying graph $G = (V, E)$ is a connected planar graph of maximum degree 3 and of minimum degree 1. We show that a subset $V'$ of $V$ is a dominating set of radius $r$ for $D$ if and only if $V'$ is a dominating set of radius $r$ for $G$. Let $V' \subseteq V$ be a dominating set of radius $r$ for the digraph $D$. Then by definition for all $u \in V-V'$ there is a $v \in V'$ for which $d_D(v, u) \leq r$. Since $G$ is the underlying graph of $D$, we
have \( d_G(v, u) \leq r \). Therefore \( V' \) is also a dominating set of radius \( r \) for \( G \).

Conversely, suppose that \( V' \) is a dominating set of radius \( r \) for the graph \( G \). Then for all \( u \in V - V' \) there is a \( v \in V' \) for which \( d_G(v, u) \leq r \). Let \( P \) be a shortest \( v-u \) path in \( G \). Since \( G \) is the underlying graph of the symmetric digraph \( D \), there is a directed \( v-u \) path \( P' \) in \( D \) having the same length as that of \( P \) in \( G \). Therefore \( V' \) is also a dominating set of radius \( r \) for \( D \). Thus a subset \( V' \) of \( V \) is a dominating set of radius \( r \) for \( D \) if and only if \( V' \) is a dominating set of radius \( r \) for \( G \). The result follows from Theorem 2.1.

Algorithm 2.1 Finds a minimum dominating set of radius \( r \) in a weighted oriented tree.

Procedure DOMINATE\((T, \text{ROOT}, r, v, U, D, K)\)

INPUT:

\( T \) is a weighted oriented tree.

\( \text{ROOT} \) is the root of the tree \( T \).

\( r \) is a positive integer, called a dominating radius.

\( v \) is a vertex of \( T \).

OUTPUT:

\( U \) is a subset of \( V(T) \). At return, \( U \) is a minimum dominating set of radius \( r \) for \( T \) if \( v = \text{ROOT} \).

Otherwise, \( U \) is a subset of a minimum dominating set of radius \( r \) for the subtree of \( T \) rooted at \( v \).

\( D \) is a boolean variable. Let \( T_v \) be the subtree of \( T \) rooted at \( v \). If each vertex in \( V(T_v) \) is dominated by \( U \) within
distance \( r \), then \( D \) has value TRUE. Otherwise, \( D \) has value FALSE.

\( K \) is a nonnegative integer. If \( D \) has value TRUE, then \( K = d(U, v) \). Otherwise there exists a vertex \( u \) in the subtree rooted at \( v \) such that \( K = d(v, u) < r \) and \( u \) needs to be dominated by \( v \) or an ancestor (if any) of \( v \).

**VARIABLES:**

COVERED is a boolean variable. If vertex \( v \) has been dominated by \( U \) within distance \( r \), then \( \text{COVERED} \) has value TRUE; otherwise it has value FALSE.

\( Q \) is a temporary variable, its value is the maximum distance of \( v \) from a descendant which has not been dominated yet.

\(/\ast\) In this procedure, an arc \((u, v)\) is denoted by \( uv \). */

**begin**

if (\( v \) is an end-vertex) then

1) \( D \) \( \leftarrow \) FALSE; \( K \) \( \leftarrow \) 0

else /* \( v \) is not an end-vertex */

Let \( v_1, v_2, \ldots, v_m \) be the children of \( v \)

2) for \( i \leftarrow 1 \) to \( m \) loop

\( \text{DOMINATE}(T, \text{ROOT}, r, v_i, U, D_i, K_i) \)

3) end loop

/* Determine some children of \( v \) which need to be included in the dominating set \( U \) */

4) for \( i \leftarrow 1 \) to \( m \) loop

...
If (not $D_i$) and ($v_i v \in A(T)$) then
$$U \leftarrow U \cup \{v_i\}; \quad D_i \leftarrow \text{TRUE}; \quad K_i \leftarrow 0$$
end if

5) end loop

6) $K \leftarrow r; \quad \text{COVERED } \leftarrow \text{FALSE}$

/* Determine whether the vertex $v$ has been dominated by some of its descendants */

7) for $i \leftarrow 1$ to $m$ loop

if $D_i$ and ($v_i v \in A(T)$) and ($K_i + c(v_i v) \leq r$) then
$$\text{COVERED } \leftarrow \text{TRUE}; \quad K \leftarrow \min \{K, K_i + c(v_i v)\}$$
end if

8) end loop

if (not COVERED) then

9) $D \leftarrow \text{FALSE}; \quad K \leftarrow 0$

10) for $i \leftarrow 1$ to $m$ loop

/* Determine some children of $v$ which need to be included in the dominating set $U$ */

11) if (not $D_i$) and ($K_i + c(vv_i) > r$) then
/* not $D_i$ implies $vv_i$ is an arc of $T$ */
$$U \leftarrow U \cup \{v_i\}$$
end if

12) if (not $D_i$) and ($K_i + c(vv_i) \leq r$) then
/* Update the maximum distance of $v$ to some vertex $w$ in $T_v$ such that $w$ has not been dominated yet */

13) \[ K \leftarrow \max \{K, K_i + c(vv_i)\} \]

end if

end loop

if \( v = \text{ROOT} \) then

14) \[ U \leftarrow U \cup \{v\}; \quad D \leftarrow \text{TRUE} \]

end if

end if

if \( \text{COVERED} \) then

15) \[ Q \leftarrow 0 \]

for \( i \leftarrow 1 \) to \( m \) loop

/* determines some children of \( v \) which are required to be in the set \( U \). */

16) if \( \text{not} \ D_i \) and \( (c(vv_i) + K_i > r) \) then

\[ U \leftarrow U \cup \{v_i\}; \quad D_i \leftarrow \text{TRUE} \]

17) end if

/* Updates the maximum distance of \( v \) from a descendant which has not been dominated yet. */

18) if \( \text{not} \ D_i \) and \( (K + c(vv_i) + K_i > r) \) then

\[ Q \leftarrow \max \{Q, c(vv_i) + K_i\} \]

19) end if

end loop

20) if \( Q > 0 \) then

\[ D \leftarrow \text{FALSE}; \quad K \leftarrow Q \]

else

\[ D \leftarrow \text{TRUE}; \]
end if
end if
end if
end Procedure DOMINATE

begin /* Algorithm 2.1 */
U ← ∅
Select a vertex v as the root of T
ROOT ← v
if |V(T)| = 1 then
    U ← {v}
else
    DOMINATE(T, ROOT, r, v, U, D, K)
end if
end Algorithm 2.1

Example 2.1 Let T be the oriented tree in Figure 2.2, and let
$r = 2$. The vertex $u$ is the root of $T$ and the numbers in the figure are the weights of the arcs of $T$. If for each vertex $v$, Algorithm 2.1 calls procedure DOMINATE for the children of $v$ in left to right order, then the minimum dominating set of radius $r$ found is the set of solid vertices as indicated in Figure 2.2.

Note that the distance matrix of an oriented tree can be calculated in linear time.

**Theorem 2.3** Algorithm 2.1 finds a minimum dominating set of radius $r$ in a weighted oriented tree $T = (V, A)$ in $O(|V|)$ time.

**Proof:** If $T$ is of order one, the result is clear. So we suppose that the oriented tree $T$ is of order at least two. Algorithm 2.1 selects a vertex as the root of $T$ and calls procedure DOMINATE. Procedure DOMINATE uses depth first search to find a set $U$ of $V$. We will show that $U$ is a minimum dominating set of radius $r$ of $T$.

Note that procedure DOMINATE is recursive. We need only show that for each call DOMINATE($T$, ROOT, $r$, $v$, $U$, $D$, $K$), an optimal decision is made, i.e., a required number of vertices is selected into the set $U$ so that each vertex in the subtree rooted at $v$ is either dominated by some vertex in the current set $U$ within distance $r$ or dominated by some ancestor of $v$, and the selected set of vertices has "best" potential of covering more vertices that are not in the subtree rooted at $v$ within distance $r$. In the following "domination" means "domination of radius $r$", and $T_v$ is the subtree of $T$ with root $v$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
If $v$ is an end-vertex, then by step 1, vertex $v$ is not selected into $U$ currently. Note that $v$ might be selected into $U$ later. So we may assume that $v$ is not an end-vertex and let $v_1, v_2, \ldots, v_m$ be the children of $v$. The loop from step 2 to step 3 calls procedure DOMINATE for each child of $v$. By the loop from step 4 to step 5, for each child $v_i$ of $v$, if $v_i$ is not dominated yet and $v_i,v \in A(T)$, then $v_i$ is selected into $U$. This is necessary since no other vertices not in $T_v$ can dominate $v_i$. Step 6 initializes $K$ and $COVERED$. The loop from 7 to 8 determines whether the vertex $v$ has been dominated by some of its descendants. At this point we consider two cases depending on whether the vertex $v$ has been dominated.

**Case 1: $COVERED = FALSE$.**

Step 9 initializes $D$ and resets $K$. Note that since $D$ has value $FALSE$, $K$ is the maximum distance of $v$ to some vertex $w$ in $T_v$ such that $w$ has not been dominated yet. Observe that at this point, if $v_i$ is not dominated, then $(v, v_i) \in A(T)$. The if statement from step 11 to 12 determines some children of $v$ which are required to be in the set $U$. Step 13 updates $K$. Step 14 selects $v$ into $U$, since at this point $v$ is the root of the tree $T$ and $v$ needs to be dominated.

**Case 2: $COVERED = TRUE$.**

Step 15 sets $Q$ to 0. Steps 16 to 17 determine some children of $v$ which are required to be in the set $U$. Steps 18 to 19 update $Q$. Steps 20 to 21 update $D$ and $K$. □
Theorem 2.4 Algorithm 2.1 can be used to find a minimum dominating set of radius \( r \) for a weighted undirected tree.

**Proof:** Let \( T \) be an weighted undirected tree and \( r \) be a vertex of \( T \). Orient the tree \( T \) so that \( T \) becomes an arborescence \( T' \). Then use Algorithm 2.1 to find a minimum dominating set, say \( U \), of radius \( r \) of \( T' \). \( U \) is also a minimum dominating set of radius \( r \) of \( T \). \( \square \)

The cardinality of a smallest dominating set of radius \( r \) of an unweighted graph \( G \) (digraph \( D \)) is called the r-domination number of \( G \) (of \( D \)) and is denoted by \( \varphi_r(G) (\varphi_r(D)) \).

**Theorem 2.5** Let \( G \) be a connected graph. Then

\[
\varphi_r(G) = \min \varphi_r(T),
\]

where the minimum is taken over all spanning trees \( T \) of \( G \).

**Proof:** Let \( G \) be a connected graph and \( T \) be a spanning tree of \( G \). Then any dominating set of radius \( r \) of \( T \) is also a dominating set of radius \( r \) of \( G \). Therefore,

\[
\varphi_r(G) \leq \varphi_r(T).
\]

It follows that,

\[
\varphi_r(G) \leq \min \varphi_r(T),
\]

where the minimum is taken over all spanning trees \( T \) of \( G \).

Now we show the reverse inequality. If \( G \) is a tree, the theorem holds trivially. So we may assume that \( G \) is a connected non-acyclic graph. Let \( U \) be a minimum dominating set of radius \( r \) of \( G \) and \( C \) be a smallest cycle in \( G \). Note that if we can show that \( U \) is a dominating set of radius \( r \) of \( G - e \), for some cycle edge \( e \), then \( \varphi_r(G - e) \leq |U| = \varphi_r(G) \). By applying this
result a finite number of times, we will have $\mathcal{V}_r(T) \leq \mathcal{V}_r(G)$ for some spanning tree $T$ of $G$, thus

$$\mathcal{V}_r(G) \geq \min \mathcal{V}_r(T),$$

where the minimum is taken over all spanning trees $T$ of $G$.

Now we show that $U$ is a dominating set of radius $r$ of $G - e$, for some cycle edge $e$. Select a vertex $x$ in $V(C)$ such that $d_G(x, U) = \max \{d_G(v, U) \mid v \in V(C)\}$. Let $y$ and $z$ be the two vertices in $V(C)$ which are adjacent to $x$. Without loss of generality, suppose that $d_G(z, U) \leq d_G(y, U)$. Let $e = xy$. Then $e$ is a cycle edge. We will show that the edge $e$ has the desired property.

Let $u \in U$ such that $d_G(y, u) = d_G(y, U) \leq r$ and let $P$ be a $y-u$ path of length $d_G(y, u)$ in $G$. By the way in which $x$ was chosen, $d_G(y, U) \leq d_G(x, U)$. It follows that the edge $e = xy$ is not on the path $P$. Therefore $d_{G-e}(y, U) = d_G(y, U)$. Similarly, since $d_G(z, U) \leq d_G(y, U)$ and $d_G(z, U) \leq d_G(x, U)$, $d_{G-e}(z, U) = d_G(z, U)$. This implies that $d_{G-e}(x, U) = d_G(x, U)$. Therefore for any $v \in V(G)$, $d_{G-e}(v, U) = d_G(v, U) \leq r$. Thus $U$ is a dominating set of radius $r$ of $G - e$, where $e = xy$. □

**Corollary 2.5** Let $G$ be a connected graph of order $n$ with at most $K$ cycles, where $K$ is a positive integer. Then the time complexity for computing the $r$-domination number $\mathcal{V}_r(G)$ is $O(nK^+1)$.

**Proof:** Since $G$ has at most $K$ cycles, the number of spanning trees of $G$ is bounded above by $O(n^K)$. For each spanning tree $T$ of $G$, a minimum dominating set of radius $r$ of $T$ can be found in $O(n)$ time, using Algorithm 2.1. It follows that $\mathcal{V}_r(T)$ can be
found in $O(n)$ time. By Theorem 2.5, $\delta_r(G)$ can be found in $O(n^k)\cdot O(n) = O(n^{k+1})$ time.

**Theorem 2.6** Let $D$ be a connected digraph. Then

$$\delta_r(D) = \min \delta_r(T),$$

where the minimum is taken over all spanning oriented trees $T$ of $D$.

**Proof:** By a proof similar to Theorem 2.5, we have

$$\delta_r(D) \leq \min \delta_r(T),$$

where the minimum is taken over all spanning oriented trees $T$ of $D$.

Now we show the reverse inequality. If $D$ is an oriented tree, the theorem holds trivially. So we may assume that $D$ contains a smallest subdigraph $D_1$ whose underlying graph is a cycle $C$. Then $V(D_1) = V(C)$. Let $U$ be a minimum dominating set of radius $r$ of $D$. Note that if we can show that $U$ is a dominating set of radius $r$ of $D - e$, for some arc $e$ in $D_1$, then $\delta_r(D - e) \leq |U| = \delta_r(D)$. So by applying this result a finite number of times, we will have $\delta_r(T) \leq \delta_r(D)$ for some spanning oriented tree $T$ of $D$, thus

$$\delta_r(D) \geq \min \delta_r(T),$$

where the minimum is taken over all spanning oriented trees $T$ of $D$.

Now we show that $U$ is a dominating set of radius $r$ of $D - e$, for some arc $e$ in $D_1$. Select a vertex $x$ in $V(D_1)$ such that $d_D(U, x) = \max \{d_D(U, v) \mid v \in V(D_1)\}$. Let $y$ and $z$ be the two ver-
vertices in $V(D_i)$ such that $x$ is adjacent to $y$ and $z$ in the undirected cycle $C$. We consider two cases:

Case 1: $(x, y) \in A(D_i)$ or $(x, z) \in A(D_i)$

For simplicity, suppose that $e = (x, y) \in A(D_i)$. Since $d_{D_i}(U, y) \leq d_{D_i}(U, x)$, $d_{D_i-\{e\}}(U, v) = d_{D_i}(U, v)$ for all $v \in V(D_i)$. It follows that $U$ is a dominating set of radius $r$ of $D - e$.

Case 2: $(y, x) \in A(D_i)$ and $(z, x) \in A(D_i)$.

Without loss of generality, we may assume that $d_{D_i}(U, z) \leq d_{D_i}(U, y)$. Let $e = (y, x) \in A(D_i)$. Then $d_{D_i-\{e\}}(U, v) = d_{D_i}(U, v)$ for all $v \in V(D_i)$. Therefore $U$ is a dominating set of radius $r$ of $D - e$.

A proof similar to Corollary 2.5 gives

**Corollary 2.6** Let $D$ be a connected digraph of order $n$ whose underlying graph has at most $K$ cycles, where $K$ is a positive integer. Then the time complexity for computing the $r$-domination number $\delta_r(D)$ is $O(n^{K+1})$.

---

**m-Centrix Radius $r$ Augmentation**

In this section, the $m$-Centrix Radius $r$ Augmentation problem and the Digraph $m$-Centrix Radius $r$ Augmentation problem are shown to be NP-complete for fixed $m$ and $r$ such that $r \geq 2$. Both problems can be solved in $O(m \cdot |V|)$ time for $r = 1$ and for trees (oriented trees).

**Theorem 2.7** For $r = 1$, the problem of finding a minimum $m$-centrix radius $r$ augmentation can be solved in $O(m \cdot |V|)$ time.
Proof: Let \( G = (V, E) \) be a graph, \( m \) be a positive integer, \( C \subseteq V \) be an \( m \)-centrix of \( G \), and \( r = 1 \). The vertices in \( C \) are called central vertices. Note that a spanning supergraph \( G' \) of \( G \) has the property that \( d_{G'}(C, v) \leq 1 \) for all \( v \in V(G') \) if and only if each vertex in \( V(G') - C \) is adjacent to some vertex in \( C \) in the graph \( G' \).

Let \( C = \{c_1, c_2, \ldots, c_m\} \). For each central vertex \( c_i \), \( 1 \leq i \leq m \), find the set \( V_i \) of vertices adjacent to \( c_i \). Let \( V' = V_1 \cup V_2 \cup \ldots \cup V_m \) and \( V'' = V - (V' \cup C) \). Then \( V'' \) is the set of vertices in \( V(G) - C \) which are not adjacent to vertices in \( C \). Let \( E' = \{c_i v \mid v \in V''\} \). Then \( E' \) is a minimum set of edges to be added to \( G \) so that the resulting graph \( G' \) satisfies \( d_{G'}(C, v) \leq 1 \) for all \( v \in V(G') \). Note that \( E' \) can be found in \( O(m \cdot |V|) \) time.

Corollary 2.7 For \( r = 1 \), the \( m \)-Centrix Radius \( r \) Augmentation problem can be solved in \( O(m \cdot |V|) \) time.

By an analogous proof, we have

Theorem 2.8 For \( r = 1 \), the problem of finding a minimum digraph \( m \)-centrix radius \( r \) augmentation can be solved in \( O(m \cdot |V|) \) time.

Corollary 2.8 For \( r = 1 \), the Digraph \( m \)-Centrix Radius \( r \) Augmentation problem can be solved in \( O(m \cdot |V|) \) time.

Lemma 2.1 Let \( G = (V, E) \) be a graph and \( C = \{c_1, c_2, \ldots, c_m\} \) be a subset of \( V \), called an \( m \)-centrix. The vertices in \( C \) are called central vertices. Suppose that \( E' \) is a minimum set of edges such that for the graph \( G' = (V, E \cup E') \), \( d_{G'}(C, v) \leq r \) for all \( v \in V \). Then there exists a set \( E'' \) of edges such that \( |E''| = |E'| \), each edge of
E" joins a central vertex and a noncentral vertex, and for the graph $G'' = (V, E \cup E'')$, $d_{G''}(C, v) \leq r$ for all $v \in V$.

**Proof:** Let $G = (V, E)$ be a graph and $C = \{c_1, c_2, \ldots, c_m\} \subseteq V$ be a subset of vertices. Suppose that $E'$ is a minimum set of edges such that for the graph $G' = (V, E \cup E')$, $d_{G'}(C, v) \leq r$ for all $v \in V$. Observe that edges joining pairs of central vertices (i.e., vertices in $C$) do not affect the distance $d(C, v)$ for any $v \in V$. Since $E'$ is minimum, no edge in $E'$ joins a pair of central vertices.

Now, suppose that $e = uv \in E'$ is an edge joining two noncentral vertices. Without loss of generality, suppose that $d_{G'}(C, u) < d_{G'}(C, v)$. Let $E_1 = E' - uv + c_1v$ (see Figure 2.3), then for the graph $H = (V, E \cup E_1)$, $d_H(C, v) \leq d_{G'}(C, v) \leq r$ for all $v \in V$, $|E_1| = |E'|$, and the number of edges joining pairs of noncentral vertices in $H$ is less than that of $G'$.

![Figure 2.3 Edge replacement used in Lemma 2.1](image)

If no edges in $H$ join a pair of noncentral vertices, then we set $E'' = E_1$. Otherwise, we replace edges joining pairs of noncentral vertices for $H$ as we did for $G'$. After at most $O(|V|^2)$ steps a desired set $E''$ will be obtained. 

Similarly, we have

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Lemma 2.2 Let $D = (V, A)$ be a digraph and $C = \{c_1, c_2, \ldots, c_m\}$ be a subset of $V$, called an $m$-centrix. The vertices in $C$ are called central vertices. Suppose that $A'$ is a minimum subset of $V \times V - A$ such that for the digraph $D' = (V, A \cup A')$, $d_{D'}(C, v) \leq r$ for all $v \in V$. Then there exists a subset $A''$ of $V \times V - A$ such that $|A''| = |A'|$, each arc of $A''$ joins a central vertex and a non-central vertex, and for the digraph $D'' = (V, A \cup A'')$, $d_{D''}(C, v) \leq r$ for all $v \in V$.

Proof: Let $D = (V, A)$ be a digraph and $C = \{c_1, c_2, \ldots, c_m\} \subseteq V$ be a subset of vertices. Suppose that $A'$ is a minimum subset of $V \times V - A$ such that for the digraph $D' = (V, A \cup A')$, $d_{D'}(C, v) \leq r$ for all $v \in V$. Observe that arcs joining pairs of central vertices do not affect the distance $d(C, v)$ for any $v \in V$. Since $A'$ is minimum, no arc in $A'$ joins a pair of central vertices.

Now, suppose that $e = (u, v) \in A'$, where $u$ and $v$ are non-central vertices. Let $A_1 = A' - (u, v) + (c_1, v)$ (Figure 2.4), then for the digraph $D_1 = (V, A \cup A_1)$, $d_{D_1}(C, w) \leq d_{D'}(C, w) \leq r$ for all $w \in V$, $|A_1| = |A'|$, and the number of arcs joining pairs of non-central vertices in $D_1$ is less than that of $D'$. 

![Figure 2.4 Arc replacement used in Lemma 2.2](image-url)
If no arcs in $D_i$ join a pair of noncentral vertices, then we set $A'' = A_i$. Otherwise, we replace arcs joining pairs of noncentral vertices for $D_i$ as we did for $D'$. After at most $O(|V|^2)$ steps a desired set $A''$ will be obtained.

**Theorem 2.9** For any fixed positive integers $m$ and $r$ such that $r \geq 2$. The $m$-Centrix Radius $r$ Augmentation problem is NP-complete even in the case that the graph is connected planar of maximum degree 3.

**Proof:** It is easy to see that the $m$-Centrix Radius $r$ Augmentation problem is in NP, since a nondeterministic algorithm need only guess a set of edges $E'$ such that $E' \cap E = \emptyset$ and check in polynomial time that $|E'| \leq B$ and for the graph $G' = (V, E \cup E')$, $d(C, v) \leq r$ for all $v \in V$.

We will transform the Dominating Set of Radius $r$ problem to the $m$-Centrix Radius $r$ Augmentation problem. Let $H = (V_1, E_1)$ be a connected planar graph of maximum degree 3 and of minimum degree 1, $r$ and $K$ be two integers, $1 \leq r < K \leq |V_1|$ in an instance of the Dominating Set of Radius $r$ problem. Set $r' = r + 1$. We will construct a connected planar graph $G = (V, E)$ of maximum degree 3 such that there exists an edge set $E'$ with $|E'| \leq K$ and for $G' = (V, E \cup E')$, $d_G(C, v) \leq r'$ for all $v \in V$ if and only if there is a subset $V_0 \subseteq V_1$ such that $|V_0| \leq K$ and $d_H(V_0, v) \leq r$ for all $v \in V_1$.

Let $V(G) = V_1 \cup V_2 \cup C$, where $V_2 = \{v_1, v_2, \ldots, v_{r+1}\}$, $C = \{c_1, c_2, \ldots, c_m\}$, and $V_1, V_2$, and $C$ are pairwise disjoint. There
exists a vertex, say $v_{r+2}$, of degree 1 in $H$. Let $E(G) = E_1 \cup E_2$, where $E_2 = \{c_i c_{i+1} | i = 1, 2, \ldots, m-1\} \cup \{c_m v_i\} \cup \{v_i v_{i+1} | 1 \leq i \leq r+1\}$.

![Diagram](image)

Figure 2.5 The graph induced by $E_2$

The graph $G$ constructed above is clearly connected planar and of maximum degree 3. In graph $G$, only vertices in $C \cup V_2$ are within distance $r + 1$ from the m-centrix $C$. Suppose that $H$ has a dominating set $V_0$ of radius $r$ such that $V_0 \subseteq V_1$, $|V_0| \leq K$, and $d_H(V_0, v) \leq r$ for all $v \in V_1$. Let $E' = \{c_i v | v \in V_0\}$, then $|E'| = |V_0| \leq K$ and for $G' = (V, E \cup E')$, $d_{G'}(C, v) \leq r' = r + 1$ for all $v \in V$.

On the other hand, suppose that there exists an edge set $E'$ with $|E'| \leq K$ and for $G' = (V, E \cup E')$, $d_{G'}(C, v) \leq r' = r + 1$ for all $v \in V$. By Lemma 2.1, we may assume that each edge $e \in E'$ joins a central vertex $c_i \in C$ and a noncentral vertex. Let $V_0 = \{v | \exists e \in E' \text{ such that } e \text{ is incident with } v \text{ and } v \notin C\}$. Then clearly $|V_0| \leq |E'| \leq K$. Since for any $v \in V(H)$, $d(C, v) > r+1$ in $G$, and for any $v_0 \in V_0$, $d(C, v_0) = 1$ in $G'$, we have $d_H(V_0, v) \leq r$ for all $v \in V_1$. This completes the proof. $\square$

**Theorem 2.10** For any fixed positive integers $m$ and $r$, $r \geq 2$, the Digraph m-Centrix Radius r augmentation problem is NP-
complete even in the case that the digraph is symmetric and its underlying graph is planar of maximum degree 3.

Proof: In the proof of Theorem 2.9, replace each graph considered by the corresponding symmetric digraph. □

In the next algorithm, we use the fact that there is a linear time algorithm which finds the connected components of a graph [32].

Algorithm 2.2 m-Centrix Radius r augmentation for oriented trees.

INPUT: An oriented tree \( T = (V, A) \), \(|V| = n\), a subset of vertices \( C = \{c_1, c_2, \ldots, c_m\} \subseteq V\), called an m-centrix, positive integer \( r \). \( T \) is represented by its list structure, i.e., for each vertex, a list of vertices to which it is adjacent to is stored.

OUTPUT: A minimum set \( A' \) of arcs such that for the digraph \( D' = (V, A \cup A') \), \( d_{D'}(C, v) \leq r \) for all \( v \in V \).

VARIABLES

- \( d(v_i) \) is the minimum distance of vertex \( v_i \) to the m-centrix \( C \) after the algorithm
- \( W \) is the set of vertices in \( V \) such that \( A' = \{c_iw \mid w \in W\} \)

begin

/* initialize \( d(v_i) \) */
for \( i \leftarrow 1 \) to \( n \) loop
    \( d(v_i) \leftarrow n \)
end loop

/* For each vertex \( v_i \), determine the distance of \( v_i \) from the m-centrix \( C \) */
1) for $j \leftarrow 1$ to $m$ loop
   Let $c_j$ be the root of the tree.
2) Use Breadth First Search Method to determine
   the distances $d(c_j, v_i)$ in $T$ for $1 \leq i \leq n$.
   /* update $d(v_i)$ */
   for $i \leftarrow 1$ to $n$ loop
     if ($d(c_j, v_i) < d(v_i)$) then
       $d(v_i) \leftarrow d(c_j, v_i)$
     end if
   end loop
   end loop
   /* Finds the set $V_1$ of vertices, the distance from the
   $m$-centrix $C$ to each vertex of $V_1$ is greater than $r$ */
   $V_1 \leftarrow \emptyset$
3) for $i \leftarrow 1$ to $n$ loop
   if ($d(v_i) > r$) then
     $V_1 \leftarrow V_1 \cup \{v_i\}$
   end if
4) end loop
   if ($r = 1$) then $W \leftarrow V_1$
   else
     $W \leftarrow \emptyset$
     Let $F$ be the induced graph $\langle V_1 \rangle$.
     /* $F$ is represented by its list structure. */
5) Find the connected components of $F$.
   Let $T_1, T_2, \ldots, T_s$ be the components of $F$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
6) for i ← 1 to s loop
7) Find a minimum dominating set \( W_i \) of radius \( r - 1 \) for \( T_i \) by Algorithm 2.1
8) \( W ← W \cup W_i \)
end loop
end if
9) \( A' ← \{ (c_i, w) | w ∈ W \} \)
end Algorithm 2.2

Theorem 2.11 Let \( T = (V, A) \) be an oriented tree of order \( n \), then Algorithm 2.2 finds a minimum \( m \)-centrix radius \( r \) augmentation of \( T \) in \( O(m \cdot n) \) time.

Proof: In step 2, Algorithm 2.2 uses a Breadth First Search Method to determine the distances \( d(v_i, c_j) \), for each fixed \( c_j \in C \). Step 2 can be done in \( m \) times. Therefore the time required for step 2 is of order \( O(m \cdot n) \). The loop from step 3 to step 4 finds the set \( V_1 \) of vertices each of which has distance at least \( r + 1 \) from the \( m \)-centrix \( C \). For \( r = 1 \), the result is trivial. So we assume that \( r ≥ 2 \). Let \( F \) be the induced graph \( <V_1> \). Step 5 finds the connected components of \( F \) which can be done in \( O(n) \) time [32]. Steps 7 to 8 use Algorithm 2.1 to find \( W \) which is a minimum dominating set of radius \( r - 1 \) of \( F \). By Theorem 2.3, step 7 requires time \( O(n) \). Therefore the time complexity of Algorithm 2.2 is \( O(m \cdot n) \).

Let \( A' \) be the set of arcs returned from step 9, \( D' = (V, A \cup A') \), and \( v ∈ V \). If \( v ∈ V_1 \), then \( d_D(C, v) ≤ d_D(C, v) ≤ r \). Otherwise let \( T_i \) be the component of \( F \) containing \( v \), then
\( d_{D'}(W_i, v) \leq r - 1 \) since \( W_i \) is a dominating set of radius \( r - 1 \) of \( T_i \).

From the way \( W \) was defined, \( c_i \) must be adjacent to all the vertices in \( W_i \). It follows that \( d_{D'}(c_i, v) \leq 1 + (r - 1) = r \). It only remains to show that \( |A'| \) is minimum. Let \( A" \) be a minimum set of arcs such that for all \( v \in V \), \( d_{D"}(C, v) \leq r \), where \( D" = (V, A \cup A") \).

By Lemma 2.2, we may assume that each arc in \( A" \) joins a central vertex \( c \in C \) and a noncentral vertex. Let \( V' = \{ v \mid \exists e \in A", \exists c \in C \text{ such that } e = (c, v) \} \). Then \( V' \) is a dominating set of radius \( r - 1 \) of \( F \). By the minimality of \( W \), \( |W| \leq |V'| \). Therefore \( |A'| = |W| \leq |V'| \leq |A"| \). Thus \( |A'| \) is minimum.

An algorithm for finding minimum \( m \)-centrrix radius \( r \) augmentations for trees can be designed similarly.

**Theorem 2.12** There is an \( O(m-n) \) time algorithm which finds a minimum \( m \)-centrrix radius \( r \) augmentation for an arbitrary tree \( T \) of order \( n \). □

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
CHAPTER III

K DISJOINT MATCHINGS AND AUGMENTATIONS

The following decision problems will be studied or referenced in this chapter. It can be shown easily that these problems are in NP.

Chromatic Index
Instance: Nonempty graph $G = (V, E)$, positive integer $K$ such that $K \leq \Delta(G) + 1$.
Question: Does $G$ have chromatic index $K$ or less, i.e., Is there an assignment of $k$ colors to the edges in $E$, with $k \leq K$, such that adjacent colors are colored differently?

$r$ -Factorability
Instance: Graph $G = (V, E)$, positive integer $r \leq \Delta(G)$.
Question: Is $G$ $r$ -factorable, i.e., Is there a factorization of $G$ such that each factor is an $r$ -factor?

K Disjoint Maximum Matchings
Instance: Graph $G = (V, E)$, positive integer $K \leq \Delta(G)$.
Question: Does $G$ contain $K$ disjoint maximum matchings?

K Disjoint 1-Factors Augmentation
Instance: Graph $G = (V, E)$ of even order, positive integers $B$ and $K$ such that $B \leq |V|^2 - |V|$ and $K \leq \Delta(G)$.
Question: Is there a set $E'$ of edges such that $|E'| \leq B$ and the graph $G' = (V, E \cup E')$ contains $K$ disjoint 1 -factors?
r-Factorability

By Vizing's theorem [51][3], the chromatic index of a nonempty graph $G$ is either $\Delta(G)$ or $\Delta(G) + 1$, where $\Delta(G)$ is the maximum degree in $G$. It follows (see [3]) that for $r \geq 1$, an $r$-regular graph has chromatic index $r$ if and only if it is 1-factory. Holyer [30] has shown that the Chromatic Index problem is NP-complete even for cubic graphs. Thus we have

**Theorem 3.1** The 1-Factorability problem is NP-complete even for cubic graphs.

**Proof:** Let $G$ be a cubic graph. Then $G$ is 1-factorable if and only if $G$ has chromatic index 3. Thus, for cubic graphs, the NP-completeness of the 1-Factorability problem follows from that of the Chromatic Index problem. □

**Corollary 3.1** The 1-Factorability problem is NP-complete even for cubic graphs having a 1-factor.

**Proof:** There is an $O(|V|^3)$ algorithm which finds a maximum matching in an arbitrary graph (e.g., Edmonds[13]). A 1-factor in a graph is a perfect matching. Thus there is an $O(|V|^3)$ time algorithm which determines whether an arbitrary graph has a 1-factor. If a graph $G$ does not have a 1-factor, then $G$ is not 1-factorable.

We transform the 1-Factorability problem for cubic graphs to 1-Factorability problem for cubic graphs having a 1-factor. Let $G = (V,E)$ be a cubic graph. Then $G$ is of even order. Use an $O(|V|^3)$ time algorithm to determine whether $G$ has a 1-factor.
If \( G \) has a 1-factor, \( G \) is mapped to \( G' \) which is isomorphic to \( G \) (an identity map). Otherwise, \( G \) is mapped to \( G' \) which is the Petersen graph (see Figure 3.1). The Petersen graph is cubic and has a 1-factor but is not 1-factorable. Clearly, this is a polynomial transformation of order \( O(|V|^3) \). \( G \) is 1-factorable if and only if \( G' \) is 1-factorable.

\[\text{Figure 3.1 The Petersen graph}\]

By Theorem 3.1, the r-Factorability problem is NP-complete for an arbitrary \( r \). A natural question is "For a fixed value \( r \), can the r-Factorability problem be solved in polynomial time?" For \( r = 2 \), the answer is affirmative.

The 2-Factorable graphs have been characterized by Petersen [43][3]: A nonempty graph \( G \) is 2-factorable if and only if \( G \) is 2n-regular for some \( n \geq 1 \).

**Theorem 3.2** The 2-Factorability problem can be solved in \( O(|V|^2) \) time.

**Proof:** Let \( G = (V, E) \) be a graph which is represented by its adjacency matrix. The degree of the \( i \)-th vertex in \( G \) is the number of "one" entries in the \( i \)-th row of the adjacency matrix.
Thus the degrees of the vertices of $G$ can be determined in $O(|V|^2)$ time. It follows that the problem of deciding whether a graph $G$ is $2n$-regular for some $n \geq 1$ can be solved in $O(|V|^2)$ time. By Petersen's characterization for 2-factorable graphs, the 2-Factorability problem can be solved in $O(|V|^2)$ time. □

**K Disjoint Maximum Matchings**

The majority of published results on matchings have focused on the problem of characterizing and finding maximum matchings in a graph. Notable among these are results of Edmonds [13], which gave rise to an $O(n^3)$ algorithm for finding a maximum weighted matching in an arbitrary graph $G$ of order $n$, and Hopcroft and Karp [31], which include an $O(n^{5/2})$ algorithm for finding a maximum matching in a bipartite graph. We consider disjoint maximum matchings in graphs. Note that in a bipartite graph, the two partite sets might represent personnel and jobs. A maximum matching in such a bipartite graph represents a maximum assignment of people to jobs. The existence of $K$ disjoint maximum matchings in such a graph would imply that on $K$ successive days, maximum assignments could be scheduled in such a way that no person performs the same job twice. Cockayne, Hartnell, and Hedetniemi [10] have developed a linear algorithm for determining whether an arbitrary tree $T$ contains two disjoint maximum matchings and finding them if they exist. A constructive characterization for a tree $T$ containing $K$ disjoint maximum matchings has been obtained by Slater (see [10]). We
show that the $K$ Disjoint Maximum Matchings problem is NP-complete and present a linear algorithm for determining whether an arbitrary forest $F$ contains $K$ disjoint maximum matchings and finding them if they exist.

**Theorem 3.3** The $K$ Disjoint Maximum Matchings problem is NP-complete even for cubic graphs having a perfect matching and $K = 3$.

**Proof:** Clearly, the $K$ Disjoint Maximum Matchings problem is in NP. Let $G = (V, E)$ be a cubic graph having a perfect matching (i.e., a 1-factor) and $K = 3$. Then $G$ has $K$ disjoint maximum matchings if and only if $G$ is 1-factorable. By Corollary 3.1, the result follows. □

By a result due to König [37], the chromatic index of a bipartite graph is its maximum degree. Since a tree is a bipartite graph, the chromatic index of a tree $T$ is $\Delta(T)$.

**Algorithm 3.1** Colors the edges of a tree $T$ using $\Delta(T)$ colors.

**INPUT:** A tree $T = (V, E)$ with maximum degree $K \geq 1$. The tree is represented in list structure, i.e., for each vertex, a list of vertices to which it is adjacent to is stored.

**OUTPUT:** $K$ color sets $S_1, S_2, \ldots, S_k$, edges in the set $S_i$ are colored with color $i$, $1 \leq i \leq K$.

**Procedure** COLOR($v, c_0$)

**INPUT**

- $v$ is a vertex of $T$.
- $c_0$ is the color that is used in coloring the edge $uv$, where $u$ is the parent (if any) of $v$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
VARIABLES

S is the set of colors used in coloring the edges adjacent with v in T.

c(u_j) is the color that is used to color the edge vu_j, where u_j is a child of v, j ≥ 1.

begin /* Procedure COLOR */

If (v has a child ) then

S ← {c_0}

Let u_1, u_2, ..., u_m be the children of v

/* Color the edges vu_j, 1 ≤ j ≤ m */

for j ← 1 to m loop

Select a color i_0 in {1, 2, ..., K} - S

c(u_j) ← i_0; S_{i_0} ← S_{i_0} U \{vu_j\}; S ← S U \{i_0\}

end loop

/* recursive calls */

for j ← 1 to m loop

COLOR(u_j, c(u_j))

end loop

end if

end Procedure COLOR

begin /* Algorithm 3.1 */

for i ← 1 to K loop

S_i ← ∅

end loop

/* handle the first call to the procedure COLOR */

c_0 ← -1
Select a vertex \( r \in V \)

Transform the tree \( T \) into a rooted tree having root \( r \)

\[ \text{COLOR}(r, c_0) \]

end Algorithm 3.1

**Theorem 3.4** Let \( T = (V, E) \) be a tree with maximum degree \( \Delta(T) \geq 1 \). Then Algorithm 3.1 colors the edges of \( T \) using \( \Delta(T) \) colors in linear time \( O(|V|) \).

**Proof:** The proof proceeds by induction on the depth \( \text{dep}(T) \) of the rooted tree \( T \) with root \( r \) and maximum degree \( \Delta(T) \geq 1 \). If \( \text{dep}(T) = 2 \), then \( T \) is a star. The number of children of \( r \) is \( \Delta(T) \), each \( S_i, 1 \leq i \leq K \), returns exactly one edge. So Theorem 3.4 is true for \( \text{dep}(T) = 2 \). Suppose that Theorem 3.4 holds for all rooted tree \( T' \) such that \( 2 \leq \text{dep}(T') < d \) and let \( T \) be a rooted tree of depth \( d \) (\( \geq 3 \)) with root \( r \). Let \( u_1, u_2, \ldots, u_m \) be the children of \( r \). Algorithm 3.1 colors the edges \( ru_j \) first, \( 1 \leq j \leq m \). Since \( m \leq \Delta(T) \), the edges \( ru_j, 1 \leq j \leq m \), can be colored differently. The color of the edge \( ru_j \) is stored in \( c(u_j) \). Let \( T_j \) be the subtree of \( T \) rooted at \( u_j, 1 \leq j \leq m \). By inductive hypotheses, for each \( j, 1 \leq j \leq m \), the edges of \( T_j \) can be colored with \( \Delta(T_j) \) (\( \leq \Delta(T) \)) colors in linear time such that the color \( c(u_j) \) is not used for those edges in \( T_j \) which are incident with \( u_j \). It follows that the edges of \( T \) can be colored with \( \Delta(T) \) colors in linear time \( O(|V|) \). \( \square \)

**Corollary 3.4** Let \( F = (V, E) \) be a forest with maximum degree \( \Delta(F) \geq 1 \). Then the edges of \( F \) can be colored using \( \Delta(F) \) colors in linear time \( O(|V|) \).
Given a forest $F$ and a positive integer $K$ such that $1 \leq K \leq \Delta(F)$, Algorithm 3.2 finds a maximum spanning subforest $F_1$ of $F$ having a $K$-edge coloring. The algorithm first selects a vertex as the root of a component (which is a tree) of $F$. Then Algorithm 3.2 starts from the leaves going up to the root. Based on the number $K$ of colors given, the algorithm determines the edges not to be colored at each vertex along the way from the leaves to the root. We consider deleting edges incident with a vertex $v$ only after all the children (if any) of $v$ have been considered. At the time when the edge which joins a vertex $v$ and its parent and some other edges that are incident with $v$ are considered to be deleted, $\text{deg}(v)$ is the number of edges incident with $v$ that have not been deleted yet. If $\text{deg}(v) > K$, then the edge that joins $v$ and the parent of $v$, together with $\text{deg}(v) - K - 1$ additional (arbitrary) edges that are incident with $v$ are deleted. The process continues until a forest $F_1$ satisfying $\Delta(F_1) \leq K$ is obtained. At the end of Algorithm 3.2, $F_1$ is a maximum subforest of $F$ such that $\Delta(F_1) \leq K$.

Algorithm 3.2

INPUT: Forest $F = (V, E)$ with maximum degree $\Delta(F) \geq 1$, positive integer $K$ such that $1 \leq K \leq \Delta(F)$.

OUTPUT: A maximum spanning subforest $F_1$ of $F$ such that the edges of $F_1 = (V, E_1)$ can be colored using $K$ colors.

Procedure DELETE_EDGES($T$, $v$, $K$, $E_0$)

/* This procedure deletes a minimum set of edges from a tree
\[ T = (V, E) \] rooted at \( v \) such that the set of edges \( E_0 \) of the resulting forest \( F_0 = (V(T), E_0) \) can be colored using \( K \) colors */

begin
\[ E_0 \leftarrow E; \ d \leftarrow \deg_{F_0}(v) \]
1) \( \text{if } d > K \text{ then} \)
2) \( \text{for each child } u \text{ of } v \text{ loop} \)

\[ \text{DELETE\_EDGES}(T, u, K, E_0) \]
3) \( \text{end loop} \)
4) \( \text{Let } w \text{ be the parent of } v \text{ and let } v_1, v_2, \ldots, v_{d-K-1} \text{ be} \)
5) \( \text{arbitrary children of } v \text{ such that for each} \)
6) \( i, 1 \leq i \leq d-K-1, vv_i \in E_0. \)
7) \( E_0 \leftarrow E_0 - \{wv \cup \{vv_i \mid 1 \leq i \leq d-K+1\}\} \)
end if
end Procedure DELETE\_EDGES

begin /* Algorithm 3.2 */
/* S is the set of vertices that have not been traversed */
\[ S \leftarrow \emptyset; \ E_1 \leftarrow \emptyset \]
while \( V - S \neq \emptyset \) loop
begin
Select a vertex \( v \in V - S \) such that \( \deg_F v = 1. \)
\\9) Find the connected component of the forest \( F \)
containing \( v \), say \( T_v = (V_v, E_v) \).
\\7) \( \text{DELETE\_EDGES}(T_v, v, K, E_0) \)
\\8) \( E_1 \leftarrow E_1 \cup E_0 \)
\\9) \( S \leftarrow S \cup V_1 \)
end loop
/* F_1 = (V, E_1) */
for each component T of F_1 loop
    Use Algorithm 3.1 to color the edges of T using colors {1, 2, ..., K}
end loop
end Algorithm 3.2

The following example illustrates Algorithm 3.2 for a tree.

Example 3.1 Let T be the tree in Figure 3.2 and K = 2. Suppose that v is the root of T. If Algorithm 3.2 is used to find a maximum spanning subforest F_1 of T such that F_1 has a K-edge coloring, then edge e_1 is deleted first, then e_2, then one of the edges e_3, e_4, and e_5 is deleted, finally e_6 is deleted.

Theorem 3.5 Let F = (V, E) be a forest with maximum degree \( \Delta(F) \geq 1 \) and K be a positive integer such that K \( \leq \Delta(F) \). Then Algorithm 3.2 finds a K-edge colorable maximum spanning subforest F_1 of F in linear time.
Proof: For each component of the forest \( F \), Algorithm 3.2 calls procedure \textsc{delete-edges} to delete a set of edges from \( F \) (step 6 to step 7), returning the remaining edges (step 8). So it is sufficient to show that procedure \textsc{delete-edges} deletes a minimum set of edges for each component such that the resulting forest has a \( K \)-edge coloring. Note that procedure \textsc{delete-edges} is a recursive procedure (step 2 to step 3). Since all leaves of a tree have degree 1 and \( K \geq 1 \), by the condition in step 1, edges incident with leaves are not deleted.

Suppose that \( v \) is not a leaf. We consider the effect of calling procedure \textsc{delete-edges}(T, v, K, E). Let \( F_0 \) be the current forest, i.e., the remaining forest after all the calls to the procedure \textsc{delete-edges} for the children of \( v \). Only edges in \( F_0 \) are considered at this point. If \( \deg_{F_0}(v) \leq K \), no edge incident with \( v \) needs to be deleted. If \( d = \deg_{F_0}(v) > K \), then at least \( d - K - 1 \) number of edges incident with \( v \) need to be deleted. The procedure \textsc{delete-edges} deletes the edge joining \( v \) with its parent first, followed by the \( d - K \) other edges incident with \( v \). Note that this is an optimal way of deleting the required number of edges so that the resulting forest is maximum and the edges incident with \( v \) can be colored with \( K \) colors in the resulting forest.

Algorithm 3.2 uses a depth first search method. It can be seen that the time required for finding \( F_1 \) is \( O(|V|) \). □

For \( K = 1 \), Algorithm 3.2 finds a maximum matching of an arbitrary forest in linear time.
Let $m$ be the size of a maximum matching of $F$ and $K \leq \Delta(F)$. Then $F$ contains $K$ disjoint maximum matchings if and only if $|E(F_1)| = K \cdot m$, where $F_1$ is the subforest of $F$ found by Algorithm 3.2. Thus we have

**Theorem 3.6** There is a linear time algorithm which determines whether a forest contains $K$ disjoint maximum matchings and finds them if they exist. □

**Corollary 3.6** There is a linear time algorithm which determines whether a forest contains $K$ disjoint 1-factors and finds them if they exist. □

**Lemma 3.1** Let $M$ be a maximum matching of a graph $G = (V, E)$, $e \notin E$, and $M'$ be a maximum matching of $G' = (V, E \cup \{e\})$. Then $|M'| \leq |M| + 1$.

**Proof:** Note that $M'-e$ is a matching of $G$. Since $M$ is a maximum matching of $G$, $|M'-e| \leq |M|$. It follows that $|M'| \leq |M| + 1$. □

**Theorem 3.7** Let $G = (V, E)$ be a graph of even order $2n$, and let $m$ be the cardinality of a maximum matching of $G$. Then the 1-factor augmentation number of $G$ is $n - m$.

**Proof:** Let $\text{Aug}(G)$ be the 1-factor augmentation number of $G$. Since a 1-factor in a graph of order $2n$ contains $n$ edges, by Lemma 3.1, $\text{Aug}(G) \geq n - m$. On the other hand, let $M$ be a maximum matching of $G$, then $|M| = m$. Let $V_1$ be the set of weak vertices in $G$ with respect to the matching $M$. Then $|V_1| = 2(n - m)$ and the induced subgraph $H = \langle V_1 \rangle$ is empty. Let $E'$ be a 1-factor of the complement graph of $H$, then the graph $G' = (V, E \cup E')$ con-
tains a 1-factor and \(|E'| = n - m\). Therefore \(\text{Aug}(G) \leq n - m\).

Hence \(\text{Aug}(G) = n - m\). □

K Disjoint 1-Factors Augmentation for Forests

Theorem 3.3 implies that the problem of deciding whether a graph has 3 disjoint 1-factors is NP-complete even for cubic graphs having a 1-factor. Therefore the K Disjoint 1-Factors Augmentation problem is NP-complete even for cubic graphs having a 1-factor. We prove that the K Disjoint 1-Factors Augmentation problem can be solved in polynomial time if the graphs are restricted to be forests and the positive integer K is small. Specifically, let F be a forest of even order 2n and K be a positive integer such that \(K \leq \max \{5, 2n-1\}\). Then we can determine \(\text{Aug}(F)\), the minimum number of edges to be added to F so that the resulting graph H contains K disjoint 1-factors. Furthermore, the K disjoint 1-factors in the graph H may be found in polynomial time. In order to present this result, we establish some other results first.

Theorem 3.8 and Theorem 3.9 provide sufficient conditions for a bipartite graph to contain a 1-factor.

Theorem 3.8 Let G be a bipartite graph with partite sets \(V_1\) and \(V_2\), n and K be positive integers such that \(|V_1|=|V_2|=n\), \(n \geq 2K\), and \(\delta(G) \geq n - K\). Then G contains a 1-factor.

Proof: By a theorem of König [38] and Hall([28]), \(V_1\) can be matched to a subset of \(V_2\) if and only if \(V_1\) is nondeficient.
Since $|V_1| = |V_2|$, $V_1$ cannot be matched to a proper subset of $V_2$. Thus $G$ contains a 1-factor if and only if $V_1$ is nondeficient.

Suppose, to the contrary, that $G$ does not contain a 1-factor. Then $V_1$ is deficient, i.e., there exists a nonempty subset $U$ of $V_1$ such that $|N(U)| < |U|$, where $N(U)$ is the set of all vertices of $G$ adjacent with at least one element of $U$. It follows that $V_2 - N(U) \neq \emptyset$. Let $v \in V_2 - N(U)$. Since $\delta(G) \geq n-K$, $|U| \leq n - \deg(v) \leq n - (n - K) = K$. It follows that $|N(U)| < |U| \leq K$. On the other hand, $|N(U)| \geq \delta(G) \geq n - K \geq K$, since $n \geq 2K$. Thus we have a contradiction. Hence $G$ contains a 1-factor. □

For any fixed positive integer $K$, the condition $n \geq 2K$ cannot be improved upon. This is shown by the class of graphs $\{G_m \mid m \geq 1\}$, where $G_m = 2Km, m+1$. Let $n = 2m+1$, $K = m+1$. Then $G_m$ is a bipartite graph with partite sets of cardinality $n$, $\delta(G_m) = m = n-K$, $n = 2K-1$, and $G_m$ does not contain a 1-factor.

**Theorem 3.9** Let $G$ be a bipartite graph with partite sets $V_1$ and $V_2$, $n$ and $K$ be positive integers such that $n \geq 2K$, $K \geq 2$, $|V_1| = |V_2| = n$, $u_1 \in V_1$, $u_2 \in V_2$, $\deg(v_i) \geq n-K$ for $v_i = u_1, u_2$, $\deg(u_1), \deg(u_2) \geq n - K - 1$. Then $G$ contains a 1-factor.

**Proof:** By a similar argument to that of Theorem 3.8, $G$ contains a 1-factor if and only if $V_1$ is nondeficient. Suppose, to the contrary, that $G$ does not contain a 1-factor. Then $V_1$ is deficient, i.e., there exists a nonempty subset $U$ of $V_1$ such that $|N(U)| < |U|$. It follows that $V_2 - N(U) \neq \emptyset$.

**Case 1:** There exists $v_i \in V_2 - N(U)$ such that $\deg(v_i) \geq n-K$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
In this case, \(|U| \leq n - \deg(v_i) \leq n - (n - K) = K\). It follows that \(|N(U)| < |U| \leq K\). On the other hand, if there exists \(v_j \in U\) such that \(\deg(v_j) \geq n - K\), then \(|N(U)| \geq \deg(v_j) \geq n - K \geq K\), which is a contradiction. So we may assume that \(U = \{u_i\}\) and \(\deg(u_i) = n - K - 1\). Thus \(|N(U)| = |N(u_i)| = \deg(u_i) = n - K - 1 \geq 2K - K - 1 = K - 1\), and \(|U| \geq |N(U)| + 1 = K \geq 2\), contradicting that \(|U| = |\{u_i\}| = 1\).

Case 2: \(V_2 - N(U) = \{u_2\}\) and \(\deg(u_2) = n - K - 1\).

Note that \(|N(U)| = |V_2 - \{u_2\}| = n - 1\) and \(|N(U)| < |U| \leq |V_2| = n\).

This implies that \(|U| = n\) and \(U = V_1\). Since \(u_2 \notin N(V_1)\), \(\deg(u_2) = 0\). But \(\deg(u_2) = n - K - 1 \geq 2K - K - 1 = K - 1 \geq 1\) by the assumption that \(K \geq 2\). Again, we arrive at a contradiction.

Hence, \(G\) contains a 1-factor. \(\square\)

Lemma 3.2 For \(n \geq 2\), the graphs \(K_{2n} - nK_2\) and \(K_{2n} - C_{2n}\) are 1-factorable.

Proof: Let \(n\) be a positive integer such that \(n \geq 2\). The complete graphs \(K_{2n}\) are 1-factorable (see [3]). Let \(V(K_{2n}) = \{v_0, v_1, \ldots, v_{2n-1}\}\). Arrange the vertices \(v_1, v_2, \ldots, v_{2n-1}\) in a regular \((2n-1)\)-gon in clockwised order, and place \(v_0\) in the center. Join every two vertices by a straight line segment. For \(i = 1, 2, \ldots, 2n-1\), define the edge set of the factor \(F_i\) to be the edge \(v_0v_i\) together with all those edges perpendicular to \(v_0v_i\). Then \(K_{2n}\) has a 1-factorization with 1-factors \(F_i, 1 \leq i \leq 2n - 1\). It follows that the graph \(K_{2n} - nK_2\) has a 1-factorization with
1-factors $F_i$, $1 \leq i \leq 2n - 2$, and $K_{2n} - C_{2n}$ has a 1-factorization with 1-factors $F_i$, $1 \leq i \leq 2n - 3$, since the edge sum of $F_{2n-2}$ and $F_{2n-3}$ is a Hamiltonian cycle $C_{2n}$.

Lemma 3.2 is illustrated in Figure 3.3 for $n = 3$.

![Figure 3.3 A 1-factorization of $K_6$.](image)

Note that every 1-factorable graph is regular, so we may speak of the regularity of a 1-factorable graph.

**Theorem 3.10** Let $F = (V, E)$ be a nonempty forest of even order $2m$ with maximum degree $d$ such that $1 \leq d \leq 5$. Then $F$ is a spanning subforest of some 1-factorable graph $G$ of regularity $d$. Furthermore, there is an $O(m^{5/2})$ time algorithm which finds a set $E'$ of edges such that $|E'| = d \cdot m - |E|$ and the graph $G = (V, E \cup E')$ is 1-factorable and of regularity $d$.

**Proof:** By Corollary 3.6, there is a linear time algorithm which determines whether a forest is 1-factorable. If $F$ is 1-factorable, then $E' = \emptyset$. So we may assume that $F$ is not...
1-factorable in the following.

Case 1: \( d = 1 \).

Let \( V_1 \) be the set of isolated vertices in \( F \). Since \( |V| \) is even and each edge of \( F \) is incident with exactly two vertices, \( |V_1| \) is even, say \( V_1 = \{v_1, v_2, \ldots, v_{2k}\} \). Let \( E' = \{v_i v_{i+k} | 1 \leq i \leq k\} \), then \( |E'| = m - |E| \) and \( G = (V, E \cup E') \) is a \( 1 \)-factor.

Case 2: \( d = 2 \).

Let \( G_1, G_2, \ldots, G_k \) be the components of \( F \). Then each \( G_i, 1 \leq i \leq k \), is a path. Let \( u_i \) and \( v_i \) be the end vertices of \( G_i \) (if \( G_i = K_1 \), then \( u_i = v_i \)), and \( E' = \{u_i v_{i+1} | 1 \leq i \leq k-1\} \cup \{v_k u_1\} \). Then \( G = (V, E \cup E') \) is isomorphic to the cycle \( C_{2m} \) which is \( 1 \)-factorable and of regularity 2.

Case 3: \( d = 3, 4, 5 \).

\( F \) is a forest of maximum degree \( d \). By Corollary 3.4, the edges of \( F \) can be colored with \( d \) colors in linear time such that adjacent edges are colored differently. Thus the edges of \( F \) are partitioned into \( d \) sets, each set of edges is colored with one color. Let \( E_1 \) be such a set of edges with minimum cardinality, and \( n \) be an integer such that \( |E_1| = m - n \), then \( m - n \leq \lfloor (2m-1)/d \rfloor \), i.e., \( n \geq \lceil ((d-2)m+1)/d \rceil \). Note that for \( m \geq d \), \( n \geq d - 1 \).

We will use this fact for each of the following subcases. Let \( F_1 = F - E_1 \), then \( F_1 \) is a forest of order \( 2m \) with maximum degree \( d - 1 \).

Subcase 3.1: \( d = 3 \).

In this case \( m \geq 2 \). If \( m = 2 \), then \( F = K_{1,3} \) and \( F \) is a spanning subgraph of \( K_4 \) which is \( 1 \)-factorable. \( E' = E(K_4) - E(F) \).
So we may assume that \( m \geq 3 \). It then follows that \( n \geq 2 \). By Case 2, there is a linear time algorithm which finds a set \( E_2 \) of edges such that \( |E_2| = 2m - |E(F_i)| \) and the graph \( G_1 = (V, E(F_i) \cup E_2) \) is the hamiltonian cycle \( C_{2m} \) which is 1-factorable.

Let \( H = (V, E(F) \cup E_2)) = (V, E(F_i) \cup E_1 \cup E_2) \), then the vertices of \( H \) are of degree 2 or degree 3 and the number of vertices of degree 2 is \( 2n \). Let the vertices of degree 2 in \( H \) be \( v_1, v_2, \ldots, v_{2n} \) in clockwise order along the hamiltonian cycle \( C_{2m} \), \( E_3 = \{v_iv_{n+1}, v_nv_{2n}\} \cup \{v_iv_{2n-i+1} \mid 2 \leq i \leq n-1\} \), and \( E' = E_2 \cup E_3 \). Then the graph \( G = (V, E \cup E') \) is 1-factorable and of regularity 3.

Subcase 3.2: \( d = 4 \).

In this case \( m \geq 3 \). If \( m = 3 \), then \( F = K_{1,4} \cup K_1 \) or \( F \) is the tree of order 6 with maximum degree 4.

![Figure 3.4 Tree of order 6 with maximum degree 4](image)

In either case \( F \) is a spanning subgraph of \( K_6 - 3K_2 \) which is 1-factorable and of regularity 4 by Lemma 3.2. If follows that \( E(K_6 - 3K_2) - E(F) \) may be taken as the desired set \( E' \) of edges. So we may assume that \( m \geq 4 \). Then \( n \geq 3 \). By Subcase 3.1, there is a linear time algorithm which finds a set \( E_2 \) of edges such that \( |E_2| = 3m - |E(F_i)| \) and the graph \( G_1 = (V, E(F_i) \cup E_2) \) is 1-factorable,
hamiltonian, and of regularity 3.

Let \( H = (V, E(F) \cup E_2) = (V, E(F_1) \cup E_1 \cup E_2) \), then the vertices of \( H \) are of degree 3 or degree 4 and the number of vertices of degree 3 is 2n. Note that \( H \) contains a hamiltonian cycle \( C_{2m} \), so there exists an edge \( e \) on \( C_{2m} \) such that \( e \) is adjacent to a vertex of degree 3 and a vertex of degree 4 in \( H \). Without loss of generality, suppose that \( e = uv_1 \), \( \deg_H u = 4 \), \( \deg_H v_1 = 3 \), and \( v_1 \) is the next vertex of \( u \) along the hamiltonian cycle in clockwise direction. Let the vertices of degree 3 in \( H \) be \( v_1, v_2, \ldots, v_{2n} \) in clockwise order along the hamiltonian cycle \( C_{2m} \) starting from the vertex \( v_1 \). Let \( V_1 = \{v_1, v_2, \ldots, v_n\} \), \( V_2 = \{v_{n+1}, v_{n+2}, \ldots, v_{2n}\} \), \( E_3 = \{e \in E(H) \mid e = uv, u \in V_1, v \in V_2\} \), and \( H_1 = K_{n,n} - E_3 \). Note that \( v_nv_{n+1} \) may or may not be an edge in \( E_3 \), but in any case \( E_3 - v_nv_{n+1} \) is a set of independent edges in \( K_{n,n} \), so \( E_3 \) is contained in a 1-factor, say \( F_0 \), of \( K_{n,n} \). By a theorem of König [37], every regular bipartite graph of degree \( r \geq 1 \) is 1-factorable. \( K_{n,n} - F_0 \) is regular bipartite of degree \( n-1 \geq 2 \) and therefore is 1-factorable. This implies that \( K_{n,n} - F_0 - v_nv_{n+1} \) contains \( n-2 \geq 1 \) disjoint 1-factors. Thus \( H_1 \) contains a 1-factor. A 1-factor in a bipartite graph of order \( m \) can be found in \( O(m^{5/2}) \) time [31]. So a 1-factor of \( H_1 \) can be found in \( O((2n)^{5/2}) = O(n^{5/2}) \) time. The set of edges of a 1-factor in a graph is a perfect matching of the graph. Let \( E_4 \) be a perfect matching of \( H_1 \) and \( E' = E_2 \cup E_4 \), then the graph \( G = (V, E \cup E') \) is 1-factorable and of regularity 4.
Subcase 3.3: $d = 5$.

In this case, $m \geq 3$. If $m = 3$, then $F = K_{1,5}$, $F$ is a spanning subgraph of $K_8$ which is 1-factorable. $E' = E(K_8) - E(F)$. If $m = 4$, then there are three trees $T_1, T_2,$ and $T_3$ of order 8 with maximum degree 5 (see Figure 3.5). It is easy to verify that these trees are spanning subgraphs of $K_8 - C_8$.

![Figure 3.5 Trees of order 8 with maximum degree 5](image)

Note that any forest of order 8 with maximum degree 5 is a subgraph of $T_i$, for some $i, 1 \leq i \leq 3$. It follows that $F$ is a spanning subgraph of $K_8 - C_8$ which is 1-factorable and of regularity 5 by Lemma 3.2. $E' = E(K_8 - C_8) - E(F)$. So we may assume that $m \geq 5$, thus $n \geq 4$. By Subcase 3.2, there is an $O(m^{5/2})$ time algorithm which finds a set $E_2$ of edges such that $|E_2| = 4m - |E(F)|$ and the graph $G_1 = (V, E(F) \cup E_2)$ is 1-factorable, hamiltonian and of regularity 4.

Let $H = (V, E(F) \cup E_2) = (V, E(F_1) \cup E_1 \cup E_2)$, then the vertices of $H$ are of degree 4 or degree 5 and the number of vertices of degree 4 is $2n$. Note that $H$ contains a hamiltonian cycle $C_{2m}$, so there exists an edge $e$ on $C_{2m}$ such that $e$ is adjacent to a ver-
tex of degree 4 and a vertex of degree 5 in $H$. Without loss of
generality, suppose that $e=uv_1$, $\deg_H u=5$, $\deg_H v_1=4$, and $v_1$ is
the next vertex of $u$ along the hamiltonian cycle in clockwise
direction. Let the vertices of degree 4 in $H$ be $v_1, v_2, \ldots, v_{2n}$ in
clockwise order along the hamiltonian cycle $C_{2m}$ starting from
the vertex $v_1$ and let $V_1 = \{v_1, v_2, \ldots, v_n\}$, $V_2 = \{v_{n+1}, v_{n+2}, \ldots,$
$v_{2n}\}$, $E_3 = \{e \in E(H) | e = uv, u \in V_1, v \in V_2\}$, and $H_1 = K_n, n-E_3$.
We show that $H_1$ satisfies the conditions of Theorem 3.9. Let
$u_1 = v_n$, $u_2 = v_{n+1}$, and $k=2$, then $n \geq 4 = 2k$, $\deg_H(v_1) \geq n-K$ for
$v_i = u_1, u_2$, and $\deg_H(u_1), \deg_H(u_2) \geq n-K-1$. By Theorem 3.9,
$H_1$ contains a 1-factor. A 1-factor of $H_1$ can be found in
$O((2n)^{5/2}) = O(n^{5/2})$ time. Let $E_4$ be a perfect matching of $H_1$ and
$E' = E_2 \cup E_4$, then the graph $G=(V, E \cup E')$ is 1-factorable and of
regularity 5.

In case above, the desired set $E'$ can be found in $O(n^{5/2})$
time. □

Theorem 3.11 Let $F=(V, E)$ be a forest of even order $2n$ and $K$
be a positive integer such that $K \leq \max\{5, 2n-1\}$. Then there is
an $O(n^{5/2})$ time algorithm which finds a minimum set $E'$ of edges
so that the graph $G'=(V, E \cup E')$ contains $K$ disjoint 1-factors.
Proof: Use Algorithm 3.2 to find a maximum subforest $F_1$ of $F$
such that $F_1$ is $K$-edge colorable. If the maximum degree $d$ of $F$
is less than $K$, let $E_1$ be a set of $K - d$ edges such $E_1 \cap E(F_1) = \emptyset$
and the edges in $E_1$ are incident to some fixed vertex of $F_1$ having
degree $d$. Otherwise, let $E_1 = \emptyset$. By Theorem 3.10, there is an
$O(n^{5/2})$ time algorithm which finds a minimum set $E_2$ of edges so
that the graph $H = (V, E(F_1) \cup E_1 \cup E_2)$ is 1-factorable and of regularity $K$. Let $E' = E_1 \cup E_2$, then the graph $G' = (V, E(F) \cup E')$ contains $K$ disjoint 1-factors and $|E'|$ is minimum.

Based on Theorem 3.10, the following conjecture is made.

**Conjecture 3.1** Every nonempty forest $F$ of even order is a spanning subgraph of some 1-factorable graph of regularity $\Delta(F)$.
CHAPTER IV

(M, N, R₁, R₂)-TRANSITIVE AUGMENTATIONS

In a communications network it may be desirable to enforce some type of redundancy constraint on the paths that exist between vertices. This consideration leads to the following definition. Let M and N be two positive integers such that M > N, and let R₁ ∈ {=, ≥}, and R₂ ∈ (≤, ≤). A digraph D is (M, N, R₁, R₂)-transitive if for each u-v path P of length L in D such that L R₁ M there is a u-v path P' of length L' in D such that L' R₂ N and V(P') ⊆ V(P). A digraph D is free (M, N, R₁, R₂)-transitive if for each u-v path P of length L in D such that L R₁ M there is a u-v path P' of length L' in D such that L' R₂ N. (M, N, R₁, R₂)-transitivity for graphs and free (M, N, R₁, R₂)-transitivity for graphs can be defined similarly.

(2, 1, =, =)-transitivity is the normal transitivity. (M, N, =, =)-transitivity is sometimes called (M, N)-transitivity. Boals and Williams [4][52] have developed some algorithms for finding minimum and minimal (M, N, =, =)-transitive augmentations (for digraphs). Gyárfás, Jacobson, and Kinch [27] have obtained several results about (M, N, =, =)-transitive tournaments. In this chapter, a number of characterization results for tournaments as well as for graphs and digraphs are derived. Then some efficient (M, N, R₁, R₂)-transitive augmentation algo-
rithms are presented. Finally heuristic algorithms are provided for free \((d+1, d, \geq, \leq)\)-augmentation for graphs.

Results On Tournaments

Gyárfás et al. [27] have proved the following three theorems:

**Theorem 4.1** A tournament \(T\) is \((M, 1, =, =)\)-transitive if and only if it contains no cycles of length at least \(M+1\). □

**Theorem 4.2** If a tournament \(T\) is \((M, 1, =, =)\)-transitive, then \(T\) is \((M, K, =, =)\)-transitive for \(K = 1, 2, \ldots, M-1\). □

**Theorem 4.3** A tournament \(T\) is \((3, 1, =, =)\)-transitive if and only if \(T\) is \((3, 2, =, =)\)-transitive. □

By observation, we have

**Lemma 4.1** Let \(G\) be a graph or digraph.

If \(G\) is \((M, N, \geq, =)\)-transitive, then \(G\) is \((M, N, =, =)\)-transitive.

If \(G\) is \((M, N, =, =)\)-transitive, then \(G\) is \((M, N, =, \leq)\)-transitive.

If \(G\) is \((M, N, \geq, =)\)-transitive, then \(G\) is \((M, N, \geq, \leq)\)-transitive.

If \(G\) is \((M, N, \geq, \leq)\)-transitive, then \(G\) is \((M, N, =, \leq)\)-transitive. □

**Theorem 4.4** A tournament \(T\) is \((M, 1, \geq, =)\)-transitive if and only if it contains no cycles of length at least \(M+1\).

**Proof:** Let \(T\) be a \((M, 1, \geq, =)\)-transitive tournament. We show that \(T\) contains no cycles of length at least \(M+1\). Suppose, to
the contrary, that $T$ contains a cycle $C_L$ of length $L$ such that $L \geq M+1$. Let $(u, v)$ be an arc on $C_L$. Then $C_L - (u, v)$ is a $v-u$ path of length $L-1 \geq M$. Since $T$ is $(M, 1, \geq, =)$-transitive, $(v, u) \in A(T)$, contradicting that $T$ is a tournament. Therefore $T$ contains no cycles of length at least $M+1$.

On the other hand, suppose that tournament $T$ contains no cycles of length at least $M+1$. Let $P$ be a $u-v$ path of length at least $M$ in $T$, then $(v, u) \notin A(T)$; for otherwise $T$ would contain a cycle of length at least $M+1$. Since $T$ is a tournament, $(u, v) \in A(T)$. It follows that $T$ is $(M, 1, \geq, =)$-transitive. □

By Theorem 4.1 and Theorem 4.4, we have

**Theorem 4.5** Let $T$ be a tournament. Then $T$ is $(M, 1, \geq, =)$-transitive if and only if $T$ is $(M, 1, =, =)$-transitive. □

By a result due to Camion [8][3], a tournament of order at least 3 is hamiltonian if and only if it is strong. This result is used in the following theorem.

**Theorem 4.6** If a tournament $T$ is $(M, 1, \geq, =)$-transitive, then $T$ is $(M, K, \geq, =)$-transitive for $K = 1, 2, \ldots, M-1$.

**Proof:** Let $T$ be a $(M, 1, \geq, =)$-transitive tournament, $P = v_0, v_1, \ldots, v_L$ be a path of length $L$, $L \geq M$, $K$ be a positive integer, $1 \leq K \leq M-1$, $n$ be the order of $T$. If $n \leq 2$, the result is trivial. Thus we assume that $n \geq 3$. Let $T_1$ be the tournament $<V(P)>$ which is induced by the set of vertices in $P$. Then $T_1$ is not hamiltonian since $T$ is $(M, 1, \geq, =)$-transitive and $L \geq M$. Therefore $T_1$ is not strong. Then $V(T_1)$ can be partitioned as $W_1 \cup W_2 \cup \ldots \cup W_K$, $K \geq 2$, such that $W_i$ is a strong tournament for each $i$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
and if \( u_i \in W_i \) and \( u_j \in W_j \), where \( i < j \), then \( (u_i, u_j) \in A(T_i) \). Let \( V_1 = W_1 \) and \( V_2 = W_2 \cup \ldots \cup W_k \), then \( u_1u_2 \in A(T_1) \) for all \( u_1 \in V_1 \) and \( u_2 \in V_2 \). Let \( p = |V_1| \). Since \( V_0, V_1, \ldots, V_L \) is a hamiltonian path of \( T_1 \), \( \{v_0, v_1, \ldots, v_{p-1}\} = V_1 \) and \( \{v_p, \ldots, v_L\} = V_2 \) follow.

To see that \( T \) is \((M, K, \geq, =)\)-transitive, we exhibit a suitable path of length \( K \), \( P_{k+1} \); from \( v_0 \) to \( v_k \):

\[
P_{k+1} = V_0, V_1, \ldots, V_{K-1}, V_L \quad \text{if} \ K \leq p,
\]

\[
= V_0, V_1, \ldots, V_{p-1}, v_q, v_{q+1}, \ldots, v_L \quad \text{if} \ K > p,
\]

where \( q = P+L-K \).

\[ \square \]

**Theorem 4.7** A tournament \( T \) is \((3, 1, \geq, =)\)-transitive if and only if \( T \) is \((3, 2, \geq, =)\)-transitive.

**Proof:** Theorem 4.6 implies that if \( T \) is \((3, 1, \geq, =)\)-transitive, then \( T \) is \((3, 2, \geq, =)\)-transitive.

Let \( T \) be a \((3, 2, \geq, =)\)-transitive tournament. Then \( T \) is \((3, 2, =, =)\)-transitive by Lemma 4.1. By successively using Theorem 4.3 and Theorem 4.5, we have \( T \) is \((3, 1, =, =)\)-transitive and \((3, 1, \geq, =)\)-transitive.

\[ \square \]

**Theorem 4.8** A tournament \( T \) is \((3, 2, =, =)\)-transitive if and only if \( T \) is \((3, 2, =, \leq)\)-transitive.

**Proof:** Let \( T \) be a \((3, 2, =, =)\)-transitive tournament, then \( T \) is \((3, 2, =, \leq)\)-transitive by Lemma 4.1, hence it only remains to show that if \( T \) is a \((3, 2, =, \leq)\)-transitive tournament, then \( T \) is \((3, 2, =, =)\)-transitive. Let \( T \) be a \((3, 2, =, \leq)\)-transitive tournament, we show that \( T \) is \((3, 2, =, =)\)-transitive by showing that \( T \) is \((3, 1, =, =)\)-transitive. Suppose, to the contrary, that \( T \) is not \((3, 1, =, =)\)-transitive. By definition, it must be
the case that \( T \) contains a 4-cycle, say \( v_0v_1v_2v_3 \). Since \( T \) is \((3, 2, =, \leq)\)-transitive, either \( v_1v_3 \) or \( v_0v_2 \in A(T) \). Without loss of generality, suppose \( v_0v_2 \in A(T) \). By considering the path \( v_2v_3v_0v_1 \) it must be the case that \( v_2v_1 \in A(T) \). Now consider the path \( v_1v_2v_3v_0 \); a contradiction results since there is no path \( P \) of length 1 or 2 in \( T \) such that \( V(P) \subseteq \{v_0, v_1, v_2, v_3\} \). Thus \( T \) is \((3, 1, =, \leq)\)-transitive. By Theorem 4.3, \( T \) is \((3, 2, =, \leq)\)-transitive. □

Some Results on Graphs and Digraphs

**Theorem 4.9** For any positive integer \( N \), \((N+1, N, =, \leq)\)-transitivity is equivalent to \((N+1, N, \geq, \leq)\)-transitivity for digraphs.

**Proof:** Let \( D \) be an \((N+1, N, \geq, \leq)\)-transitive digraph. Since each path of length \( M \) is a path of length at least \( M \), \( D \) is \((N+1, N, =, \leq)\)-transitive.

Now suppose that \( D \) is an \((N+1, N, =, \leq)\)-transitive digraph. Let \( P \) be a \( u-v \) path of length \( L \) in \( D \) such that \( L \geq N+1 \). We show that there is a \( u-v \) path \( P' \) of length \( \leq N \) in \( D \) such that \( V(P') \subseteq V(P) \) by induction on \( L \). If \( L = N+1 \), the result is true since \( D \) is \((N+1, N, =, \leq)\)-transitive. Assume that the result is true for each \( u-v \) path \( P \) of length \( i \) satisfying \( N+1 \leq i < L \), i.e., there is a \( u-v \) path \( P' \) of length \( \leq N \) in \( D \) such that \( V(P') \subseteq V(P) \). Let \( P \) be a \( u-v \) path of length \( L \) in \( D \), \( w \) be the vertex in \( P \) which is adjacent to \( v \), and \( P_1 \) be the \( u-w \) subpath of \( P \). The path \( P_1 \) is of length \( L-1 \geq N+1 \), by induction, there is a \( u-w \) path \( P_2 \) of length \( \leq N \) in \( D \).
such that $V(P_2) \subseteq V(P_1)$. Let $P_3$ be the path formed by concatenating $P_2$ and the arc $wv$, and let $L_3$ be the length of $P_3$. Then $V(P_3) \subseteq V(P)$. If $L_3 \leq N$, then $P' = P_3$ is a desired path. Otherwise $N + 1 \leq L_3 < L$. By the inductive hypothesis, there is a $u$-$v$ path $P'$ of length $\leq N$ in $D$ such that $V(P') \subseteq V(P_3) \subseteq V(P)$. Therefore the digraph $D$ is $(N+1, N, \geq, \leq)$-transitive. This completes the proof. □

By a proof similar to that of Theorem 4.9, we have

Theorem 4.10 $(N+1, N, =, \leq)$-transitivity is equivalent to $(N+1, N, \geq, \leq)$-transitivity for graphs.

Williams and Boals [52] have shown that the intersection of two $(M, 1, =, =)$-transitive digraphs is $(M, 1, =, =)$-transitive and every digraph $D$ has a unique minimum $(M, 1, =, =)$-transitive augmentation. Results similar to these are proved in Lemma 4.2 and Theorem 4.11.

Lemma 4.2 (1) The intersection of two $(M, 1, =, =)$-transitive graphs (digraphs) is $(M, 1, =, =)$-transitive.

(2) The intersection of two $(M, 1, \geq, =)$-transitive graphs (digraphs) is $(M, 1, \geq, =)$-transitive.

Proof: (1) Let $G_1$ and $G_2$ be two $(M, 1, =, =)$-transitive graphs. Suppose that $P$ is a $u$-$v$ path of length $M$ in $G_1 \cap G_2$. Since $G_1$ and $G_2$ are $(M, 1, =, =)$-transitive, the edge $uv$ must be in each, hence in the intersection. Therefore $G_1 \cap G_2$ is $(M, 1, =, =)$-transitive. Proof for the digraph case can be found in [52].

(2) can be proved similarly. □

Theorem 4.11 Let $G$ be a graph or digraph. Then
(1) G has a unique minimum \((M, 1, =, =)-\)transitive augmentation.

(2) G has a unique minimum \((M, 1, \geq, =)-\)transitive augmentation.

**Proof:** (1) Suppose that \(G_1\) and \(G_2\) are \((M, 1, =, =)-\)transitive augmentations of \(G\). Then both \(G_1\) and \(G_2\) are \((M, 1, =, =)-\)transitive. By Lemma 4.2, \(G_1 \cap G_2\) is \((M, 1, =, =)-\)transitive. Since \(G \subseteq G_1\) and \(G \subseteq G_2\), \(G \subseteq G_1 \cap G_2\). Therefore \(G_1 \cap G_2\) is an \((M, 1, =, =)-\)transitive augmentation of \(G\). By the minimality of \(G_1\), \(|E(G_1 \cap G_2)| = |E(G_1)|\). This implies that \(E(G_1) \subseteq E(G_2)\). Similarly, \(E(G_2) \subseteq E(G_1)\). Hence \(G_1 = G_2\).

(2) can be proved similarly. \(\square\)

**Lemma 4.3** Let \(G\) be an \((M, 1, =, =)-\)transitive graph, \(M \geq 2\), and \(P\) be a path of length \(L\) in \(G\) such that \(L \geq M + 1\). Then for each pair of vertices \(u\) and \(v\) in \(P\) such that \(d_P(u, v) = 3\), \(u\) and \(v\) are adjacent in \(G\).

**Proof:** Let \(G = (V, E)\) be an \((M, 1, =, =)-\)transitive graph, \(M \geq 2\), and \(P\) be a path of length \(L\) in \(G\) such that \(L \geq M + 1\). Note that for each pair of vertices \(u\) and \(v\) in \(P\) such that \(d_P(u, v) = 3\), there exists a subpath \(P'\) of \(P\) such that \(d_{P'}(u, v) = 3\) and \(P'\) is of length \(M + 1\). So it is sufficient to prove this lemma for \(L = M + 1\).

Let \(P = v_0, v_1, \ldots, v_{M+1}\) be a path of length \(M + 1\) in \(G\). We need to show that \(v_iv_{i+3} \in E\) for all \(i\), \(0 \leq i \leq M - 2\).

If \(M = 2\), then \(v_0v_2 \in E\) since \(G\) is \((2, 1, =, =)-\)transitive and \(v_0, v_2\) is a path of length 2 in \(G\). Now \(v_0, v_2, v_3\) is a path of length 2 in \(G\), therefore \(v_0v_3 \in E\).
If $M = 3$, then $v_0v_3 \in E(G)$ since $G$ is $(3, 1, =, =)$-transitive and $v_0$, $v_1$, $v_2$, $v_3$ is a path of length 3 in $G$. Similarly, $v_1v_4 \in E(G)$.

So we assume that $M \geq 4$. Note that $v_0$, $v_1$, ..., $v_M$ is a path of length $M$ in $G$. Since $G$ is $(M, 1, =, =)$-transitive, $v_0v_M \in E$. Similarly, $v_1v_M \in E$. Now we show that $v_i v_{i+3} \in E$, $0 \leq i \leq M - 2$, by presenting a $v_i - v_{i+3}$ path of length $M$ in $G$. All of the following paths are of length $M$.

$v_0 - v_3$: $v_0$, $v_1$, $v_{M+1}$, $v_M$, ..., $v_4$, $v_3$.
$v_1 - v_4$: $v_1$, $v_2$, $v_3$, $v_0$, $v_M$, $v_{M-1}$, ..., $v_4$.
$v_2 - v_5$: $v_2$, $v_3$, $v_4$, $v_1$, $v_0$, $v_M$, $v_{M-1}$, ..., $v_5$.
$v_3 - v_6$: $v_3$, $v_4$, $v_5$, $v_2$, $v_1$, $v_0$, $v_M$, $v_{M-1}$, ..., $v_6$.

.....................
$v_{M-4} - v_{M-1}$: $v_{M-4}$, $v_{M-3}$, $v_{M-2}$, $v_{M-5}$, $v_{M-6}$, ..., $v_0$, $v_M$, $v_{M-1}$.
$v_{M-3} - v_M$: $v_{M-3}$, $v_{M-2}$, $v_{M-1}$, $v_{M-4}$, $v_{M-5}$, ..., $v_0$, $v_M$.
$v_{M-2} - v_{M+1}$: $v_{M-2}$, $v_{M-3}$, ..., $v_0$, $v_M$, $v_{M+1}$.

Theorem 4.12 If $G$ is an $(M, 1, =, =)$-transitive graph, $M \geq 2$, then $G$ is $(M + 2K, 1, =, =)$-transitive for all nonnegative integers $K$.

Proof: The proof is by induction on $K$. For $K = 0$, the result holds trivially. Suppose that the result is true for $K < L$ and let $G$ be an $(M, 1, =, =)$-transitive graph. We show that $G$ is $(M + 2L, 1, =, =)$-transitive. Let $P = v_0$, $v_1$, ..., $v_{M+2L}$ be a path of length $M + 2L$ in $G$. Since $G$ is an $(M, 1, =, =)$-transitive graph, $M \geq 2$, $d_p(v_0, v_3) = 3$, and $M + 2L > M + 1$, by Lemma 4.3, $v_0v_3 \in E(G)$. Observe that $v_0$, $v_3$, $v_4$, ..., $v_{M+2L}$ is a path of
length $M + 2(L - 1)$ in $G$. By the inductive hypotheses, $G$ is $(M + 2(L - 1), 1, =, =)$-transitive. Therefore $v_0 v_{M + 2L} \in E(G)$. Hence $G$ is $(M + 2L, 1, =, =)$-transitive. By mathematical induction, the proof is completed.

**Theorem 4.13** A graph $G$ is $(M, 1, \geq, =)$-transitive if and only if $G$ is $(M, 1, =, =)$-transitive and $(M + 1, 1, =, =)$-transitive.

**Proof:** Clearly, $(M, 1, \geq, =)$-transitivity implies $(M, 1, =, =)$-transitivity and $(M + 1, 1, =, =)$-transitivity for graphs.

On the other hand, suppose that $G$ is an $(M, 1, =, =)$-transitive and $(M + 1, 1, =, =)$-transitive graph. By Theorem 4.12, $G$ is $(M + 2K, 1, =, =)$-transitive and $(M + 1 + 2K, 1, =, =)$-transitive for all nonnegative integers $K$. It follows that $G$ is $(M, 1, \geq, =)$-transitive.

**Lemma 4.4** Let $G = (V, E)$ be a graph, $V = \{v_1, v_2, \ldots, v_n\}$ and $E = \{v_i v_{i+1} | 1 \leq i < n\} \cup \{v_i v_{i+3} | 1 \leq i \leq n - 3\}$, where $n (\geq 2)$ is even. Then $G$ contains a $v_1 - v_n$ hamiltonian path.

**Proof:** The proof proceeds by induction on $n$. For $n = 2, 4,$ and $6$, the paths $P_2: v_1, v_2; P_4: v_3, v_2, v_1, v_4; and P_6: v_5, v_4, v_1, v_2, v_3, v_6$ are $v_1 - v_n$ hamiltonian paths in $G$ respectively. Suppose that the result holds for all even positive integers $n$ such that $n < k$ and let $G$ be a graph with vertex set $V(G) = \{v_1, v_2, \ldots, v_k\}$ and edge set $E = \{v_i v_{i+1} | 1 \leq i < k\} \cup \{v_i v_{i+3} | 1 \leq i \leq k - 3\}$, where $k (\geq 8)$ is even. Let $H$ be the subgraph of $G$ induced by the set of vertices $(v_1, v_2, \ldots, v_{k-6})$. Then $E(H) = \{v_i v_{i+1} | 1 \leq i < k - 6\} \cup \{v_i v_{i+3} | 1 \leq i \leq k - 9\}$. By the inductive hypothe-
ses, H contains a \( v_{k-7} - v_{k-6} \) hamiltonian path \( P' \). Note that the following two paths \( P_1: v_{k-1}, v_{k-2}, v_{k-5}, v_{k-4}, v_{k-7} \) and \( P_2: v_{k-6}, v_{k-3}, v_k \) are in \( G \). The path \( P \) obtained by taking the path \( P_1 \) followed by \( P' \), then followed by \( P_2 \) is a \( v_{k-1} - v_k \) hamiltonian path in \( G \). □

Theorem 4.14 \((2K, 1, =, =)-\)transitivity is equivalent to \((2K, 1, \geq, =)-\)transitivity for graphs.

Proof: By Theorem 4.13, we need only show that \((2K, 1, =, =)-\)transitivity implies \((2K + 1, 1, =, =)-\)transitivity for graphs.

Suppose that \( G = (V, E) \) is a \((2K, 1, =, =)-\)transitive graph and \( P: v_0, v_1, \ldots, v_{2K+1} \) is a path of length \( 2K + 1 \) in \( G \). We will show that \( v_0 v_{2K+1} \in E \). If \( K = 1 \), then \( P \) is a \( v_0-v_3 \) path of length 3, by Lemma 4.3, \( v_0 v_3 \in E \). Therefore \( G \) is \((3, 1, =, =)-\)transitive. So we may assume that \( K \geq 2 \). Since \( G \) is \((2K, 1, =, =)-\)transitive, \( v_0 v_{2K} \in E \). By Lemma 4.3, \( v_i v_{i+3} \in E \), \( 0 \leq i \leq 2K - 2 \). Let \( H = \langle v_1, v_2, \ldots, v_{2K-2} \rangle_G \), then \( \{v_i v_{i+1} \mid 1 \leq i < 2K - 3\} \cup \{v_i v_{i+3} \mid 1 \leq i \leq 2K - 5\} \subseteq \mathcal{E}(H) \). By Lemma 4.4, \( H \) contains a \( v_{2K-3} - v_{2K-2} \) hamiltonian path \( P' \). Let \( E' = \{v_0 v_{2K}, v_{2K} v_{2K-3}, v_{2K-2} v_{2K+1}\} \cup \mathcal{E}(P') \). Then \( \langle E' \rangle_G \) induces a \( v_0-v_{2K+1} \) path of length \( 2K \) in \( G \). Therefore \( v_0 v_{2K+1} \in E \). Thus \( G \) is \((2K + 1, 1, =, =)-\)transitive. □

In general, an \((M, 1, =, =)-\)transitive graph is not neces-

\[ \text{Figure 4.1} \]
sarily (M + 1, 1, =, =)-transitive. For example, the graph repre-
icted by Figure 4.1 is (3, 1, =, =)-transitive but is not (4,
1, =, =)-transitive.

(M, 1, R1, =)-Transitive Augmentations

A number of results on the complexities of algorithms to
compute certain minimum (M, N, R1, R2)-transitive augmentations
for digraphs are known (e.g., see [4], [5], [52]). Algorithms for
finding the minimum (M, 1, =, =)-transitive augmentations for
digraphs have been developed by Williams and Boals [52]. Here
we present algorithms for computing the minimum (M, 1, R1, =)-
transitive augmentations for graphs, where R1 ∈ {=, ≥}.
Algorithm 4.1 Computes the minimum (M, 1, =, =)-transitive
augmentation of a graph.
INPUT:
Graph G = (V, E) of order p, positive integer M > 1.
OUTPUT:
The minimum (M, 1, =, =)-transitive augmentation G' =
(V, E') of G.
begin
1) E' ← E; DONE ← FALSE
2) while not DONE loop
3) ADD_EDGE ← FALSE
4) for i ← 1 to p loop
5) Find all paths of length M in G' with vi as an
   end-vertex.
for each path $v_i - v$ of length $M$ in $G$ such that $v_i v \notin E'$ loop

$E' \leftarrow E' \cup \{v_i v\}$

6) $ADD\_EDGE \leftarrow TRUE$

end loop

7) end loop

8) $DONE \leftarrow not\ ADD\_EDGE$

9) end loop

end Algorithm 4.1

Theorem 4.15 Algorithm 4.1 computes the minimum $(M, 1, =, =)$-transitive augmentation for any graph $G$ in $O(p^{M+3})$ time, where $p$ is the order of $G$.

Proof: Note that in step 8 $DONE$ is TRUE if and only if no edges were added to $G'$ during a loop from step 4 to step 7. Thus $G'$ must be $(M, 1, =, =)$-transitive. $G'$ is the minimum $(M, 1, =, =)$-transitive augmentation of $G$ since each edge in $E(G') - E(G)$ is required by the graph $G$ essentially.

Now we show that the time complexity of Algorithm 4.1 is $O(p^{M+3})$. Since there are at most $O(p^2)$ number of edges in $G$, the loop from step 2 to step 9 can be done at most $O(p^2)$ times. Note that the loop from step 4 to step 7 can be done at most $O(p^2) \cdot O(p) = O(p^3)$ times. For a fixed vertex $v_i$, all paths of length $M$ in $G'$ with $v_i$ as an end-vertex can be found in $O(p^M)$ times. Therefore the time complexity of Algorithm 4.1 is $O(p^{M+3})$. □
Algorithm 4.2 Computes the minimum \((M, 1, \geq, =)\)-transitive augmentation of a graph.

INPUT:

Graph \(G = (V, E)\) of order \(p\), positive integer \(M > 1\).

OUTPUT:

The minimum \((M, 1, \geq, =)\)-transitive augmentation \(G' = (V, E')\) of \(G\).

begin

1) Compute the minimum \((M, 1, =, =)\)-transitive augmentation \(G'\) of \(G\).

If \(M\) is odd then

DONE ← FALSE

2) while not DONE loop

3) Compute the minimum \((M + 1, 1, =, =)\)-transitive augmentation \(G_1\) of \(G'\).

4) Compute the minimum \((M, 1, =, =)\)-transitive augmentation \(G_2\) of \(G_1\).

5) If \(G_1 = G_2\) then

DONE ← TRUE

end if

\(G' ← G_2\)

6) end loop

end if

end Algorithm 4.2

Theorem 4.16 Algorithm 4.2 computes the minimum \((M, 1, \geq, =)\)-transitive augmentation for any graph \(G\) in \(O(p^{M+6})\) time,
where \( p \) is the order of \( G \).

**Proof:** If \( M \) is even, by Theorem 4.14, Theorem 4.15, and step 1, the result is true. Suppose that \( M \) is odd and let \( G'' \) be the minimum \((M, 1, \geq, =)-\)transitive augmentation of \( G \), \( G' \) be the minimum \((M, 1, =, =)-\)transitive augmentation of \( G \), and \( G_1 \) be the minimum \((M + 1, 1, =, =)-\)transitive augmentation of \( G' \). Then \( G'' \) is an \((M, 1, =, =)-\)transitive augmentation of \( G \). It follows that \( G' \subseteq G'' \). Note that \( G'' \) is an \((M + 1, 1, =, =)-\)transitive augmentation of \( G' \). Therefore \( G_1 \subseteq G'' \). Similarly, it can be shown that, for any graph \( H \) obtained from \( G \) by taking arbitrary number of \((M, 1, =, =)-\)transitive augmentations and \((M + 1, 1, =, =)-\)transitive augmentations in any order, \( H \subseteq G'' \). Therefore, the graph \( G' \) returned from Algorithm 4.2 is a subgraph of \( G'' \).

Since the number of edges in the graph \( G \) is at most \( O(p^2) \), the loop from step 2 to step 6 can be done at most \( O(p^2) \) times. Therefore Algorithm 4.2 is guaranteed to halt. DONE has value TRUE if and only if Algorithm 4.2 finds an \((M, 1, =, =)-\)transitive and \((M + 1, 1, =, =)-\)transitive graph \( G' \). By Theorem 4.13, \( G' \) is \((M, 1, \geq, =)-\)transitive.

We may use Algorithm 4.1 to compute minimum \((K, 1, =, =)-\)transitive augmentations, where \( K = M, M + 1 \). By Theorem 4.15, we have that the time complexity of Algorithm 4.2 is \( O(p^2) \cdot O(p^{M+1} + 3) = O(p^{M+6}) \).

Let \( G \) be a graph, \( M (> 1) \) be an odd positive integer, \( G'' \) be the minimum \((M, 1, \geq, =)-\)transitive augmentation of \( G \), \( G' \) be
the minimum \((M, 1, =, =)\)-transitive augmentation of \(G\), and \(G_i\) be the minimum \((M + 1, 1, =, =)\)-transitive augmentation of \(G'\).

In the proof of Theorem 4.16, we showed that \(G_i \subseteq G''\), it remains an open problem whether \(G_i = G''\).

For relational operators \(R_1 \in \{=, \geq\}\) and \(R_2 \in \{=, \leq\}\), and positive integers \(M\) and \(N\) such that \(M > N\), we consider the following decision problem.

\((M, N, R_1, R_2)\)-Transitive Augmentations (for Graphs)

Instance: Graph \(G = (V, E)\), positive integer \(K \leq |V|^2 - |V|\).

Question: Does there exist a set of edges \(E'\) with \(|E'| \leq K\) such that the graph \(G' = (V, E \cup E')\) is \((M, N, R_1, R_2)\)-transitive?

It follows from Theorem 4.15 and Theorem 4.16 that the \((M, 1, R_1, =)\)-Transitive Augmentation problems can be solved in polynomial time. By similar proofs of corresponding results in [52], it can be shown that the \((M, N, =, R_2)\)-Transitive Augmentation problems are in \(NP\). Then, it follows from Theorem 4.10, that the \((N+1, N, \geq, \leq)\)-Transitive Augmentation problem is also in \(NP\). For \(N \geq 2\) and \(M \geq N + 2\), we do not know whether the \((M, N, \geq, R_2)\)-Transitive Augmentation problems are in \(NP\).

The following conjecture is made:

Conjecture 4.1 For \(N \geq 2\), the \((M, N, \geq, R_2)\)-Transitive Augmentation problems are \(NP\)-hard.

Note that a graph \(G\) is free \((d + 1, d, \geq, \leq)\)-transitive if and only if the diameter of \(G\) is less than or equal to \(d\). We consider the following minimization problem: Given a graph \(G\) and a positive integer \(d\), find a minimum spanning supergraph \(G^*\) of \(G\) hav-
This problem is called diameter d augmentation problem. Note that a graph G has diameter 1 if and only if G is a complete graph of order at least 2. Therefore, for d = 1, the problem can be solved easily. For d ≥ 2, no significant results are known except heuristic algorithms for solving the problem. Two heuristic algorithms are presented here, to be compared.

**Algorithm 4.3**

**INPUT**
A graph G and a positive integer K.

**OUTPUT**
A supergraph G* of G such that diam(G*) ≤ K.

begin
    G* ← G
    while diam(G*) > K loop
        Find two vertices u and v such that d(u, v) = diam(G*)
        G* ← G* + uv
    end loop
end Algorithm 4.3

**Algorithm 4.4**

**INPUT**
A graph G and a positive integer K.

**OUTPUT**
A supergraph G* of G such that diam(G*) ≤ K.

begin
    Find a vertex u in the center of G.

\[ G^* \leftarrow G \]

while \( \text{rad}(G^*) > K/2 \) loop

Find a vertex \( v \) such that \( d(u, v) = e_{G^*}(u) \)

\[ G^* \leftarrow G^* + uv \]

end loop

end Algorithm 4.4

Clearly, Algorithm 4.3 has time complexity \( O(p^2) \) while Algorithm 4.4 has linear time complexity \( O(p) \).
CHAPTER V

(r,s)-DOMINATION AND AUGMENTATIONS

Cockayne, Dawes, and Hedetniemi [9] initiated the study of total dominating sets in graphs in 1980. In this chapter, we generalize the concept of total dominating set to (r, s)-dominating set and obtain some results about the (r, s)-domination number of a graph. Some of these results are generalizations of those in [9]. In addition, results parallel to Chapter II are obtained.

A total dominating set of a graph $G = (V, E)$ is a subset $U$ of $V$ such that each vertex in $V$ is adjacent to some vertex in $U$. Let $G = (V, E)$ be a graph, and $r$ and $s$ be two positive integers. A subset $U$ of $V$ is called an $(r, s)$-dominating set of $G$ if for any $v \in V - U$ there exists $u \in U$ such that $d_G(u, v) \leq r$, and for any $u_1 \in U$ there exists $u_2 \in U (u_2 \neq u_1)$ such that $d_G(u_1, u_2) \leq s$. Similarly, let $D = (V, A)$ be a digraph, and $r$ and $s$ be two positive integers. A subset $U$ of $V$ is called an $(r, s)$-dominating set of $D$ if for any $v \in V - U$ there exists $u \in U$ such that $d_D(u, v) \leq r$, and for any $u_1 \in U$ there exists $u_2 \in U (u_2 \neq u_1)$ such that $d_D(u_2, u_1) \leq s$. Clearly, an $(r, s)$-dominating set is a dominating set of radius $r$. Note that, if a digraph $D$ has an $(r, s)$-dominating set, then no vertex of $D$ has in-degree 0. Also a total dominating set is the same as a $(1, 1)$-dominating set for graphs.
Bounds on \((r, s)\)-Domination Number

The cardinality of a smallest \((r, s)\)-dominating set in a graph \(G\) is called the \((r, s)\)-domination number and is denoted by \(\gamma_{r,s}(G)\). We note that this parameter is only defined for graphs without isolated vertices and \(\gamma_{r,s}(G) \geq 2\). In the case that \(r = s = 1\), \(\gamma_{r,s}(G)\) is the same as \(\gamma_t(G)\) which is the total domination number.

Before presenting bounds on the \((r, s)\)-domination number of a graph, an important equality is proved.

**Theorem 5.1** Let \(G\) be a nontrivial connected graph, and \(r\) and \(s\) be two positive integers. Then

\[
\gamma_{r,s}(G) = \min \gamma_{r,s}(T),
\]

where the minimum is taken over all spanning trees \(T\) of \(G\).

**Proof:** Let \(G\) be a nontrivial connected graph and \(T\) be a spanning tree of \(G\). Then any \((r, s)\)-dominating set of \(T\) is also an \((r, s)\)-dominating set of \(G\). Therefore

\[
\gamma_{r,s}(G) \leq \gamma_{r,s}(T).
\]

It follows that,

\[
\gamma_{r,s}(G) \leq \min \gamma_{r,s}(T),
\]

where the minimum is taken over all spanning trees \(T\) of \(G\).

Now we show the reverse inequality. If \(G\) is a tree, the theorem holds trivially. So we may assume that \(G\) is a connected non-acyclic graph. Let \(U\) be a minimum \((r, s)\)-dominating set of \(G\) and \(C\) be a smallest cycle in \(G\). If we can show that \(U\) is an \((r, s)\)-dominating set of \(G - e\) for some cycle edge \(e\), then
\( \gamma_{r,s}(G-e) \leq |U| = \gamma_{r,s}(G) \). By applying this result a finite number of times, we have \( \gamma_{r,s}(T) \leq \gamma_{r,s}(G) \) for some spanning tree \( T \) of \( G \). Thus

\[ \gamma_{r,s}(G) \geq \min \gamma_{r,s}(T), \]

where the minimum is taken over all spanning trees \( T \) of \( G \).

Select two adjacent vertices \( x \) and \( y \) in \( V(C) \) such that

\[ d_G(x, U) + d_G(y, U) = \max \{ d_G(u, U) + d_G(v, U) | uv \in E(C) \}. \]

We will show that \( U \) is an \( (r, s) \)-dominating set of \( G-e \), where \( e = xy \).

Note that for any two adjacent vertices \( u \) and \( v \) in \( G \), the difference of \( d_G(u, U) \) and \( d_G(v, U) \) is at most one. This implies that for \( t = x \) or \( y \), \( d_G(t, U) = \max \{ d_G(v, U) | v \in V(C) \} \). Without loss of generality, suppose that \( d_G(x, U) = \max \{ d_G(v, U) | v \in V(C) \} \).

Let \( z \) be the vertex in \( V(C) \) such that \( zx \in E(C) \) and \( z = y \). By the way in which \( x \) and \( y \) were chosen, \( d_G(z, U) \leq d_G(y, U) \). Since an \( (r, s) \)-dominating set is a dominating set of radius \( r \), by the proof of Theorem 2.5, \( U \) is a dominating set of radius \( r \) of \( G-e \).

In addition, \( d_{G-e}(v, U) = d_G(v, U) \), for all vertices \( v \) in \( V(G) \). This equality will be used frequently in the rest of the proof.

Now it only remains to show that for any \( u \in U \), there exists \( u_2 \in U \) \( u_2 \neq u_1 \) such that \( d_{G-e}(u_1, u_2) \leq s \). Suppose, to the contrary, that there exists \( u_1 \in U \) such that \( d_{G-e}(u_1, U - u_1) > s \).

Let \( x' \) and \( y' \) be vertices in \( U \) such that \( d_{G-e}(x, x') = d_{G-e}(x, U) \) and \( d_{G-e}(y, y') = d_{G-e}(y, U) \). Since \( U \) is an \( (r, s) \)-dominating set of \( G \), there exists \( u_2 \in U \) \( u_2 \neq u_1 \) for which \( d_G(u_1, u_2) = d_G(u_1, U - u_1) \leq s \). Let \( P \) be a \( u_1-u_2 \) path of length \( d_G(u_1, u_2) \) in \( G \). Clearly, \( e \in
E(P). Observe that either the $u_1-x$ subpath of $P$ or the $u_1-y$ subpath of $P$ is in $G-e$. Thus we consider two cases:

Case 1: The $u_1-y$ subpath $P_1$ of $P$ is in $G-e$.

In this case, we may choose $u_2$ to be $x'$. For simplicity, we assume that $u_2 = x'$.

Let $n$ and $n_1$ be the lengths of the paths $P$ and $P_1$ respectively. Then $n = n_1 + 1 + d_{G-e}(x,U)$. If $u_1 = y'$, then

$$d_{G-e}(u_1, U - u_1) \leq d_{G-e}(u_1, y')$$

$$\leq d_{G-e}(u_1, y) + d_{G-e}(y, y')$$

$$= d_{G-e}(u_1, y) + d_{G-e}(y, U)$$

$$= n_1 + d_G(y, U)$$

$$\leq n_1 + d_G(x, U)$$

$$= n_1 + d_{G-e}(x, U)$$

$$< n$$

$$\leq s,$$

which is a contradiction.

So we may assume that $u_1 = y'$ (see Figure 5.1). Let $P_e$ be

![Figure 5.1](image-url)
the path obtained from $C$ by removing the edge $e$ and let $w$ be the vertex in $V(P_e)$ such that $d_{G-e}(w, U) = d_{G-e}(w, w')$, for some $w \in U$, $w \neq y$, $w' \neq y'$, and $d_{p}(w, y)$ is the smallest. The existence of the vertex $w$ is provided by the fact that $x \in V(P_e)$ and $d_{G-e}(x, U) = d_{G-e}(x, x')$, where $x' \in U$ and $x' = u_2 = u_1 = y'$. Let $w_1$ be the vertex in $V(P_e)$ such that $d_{P_e}(w_1, y) = d_{P_e}(w, y) - 1$. Then $w$ and $w'$ are adjacent and $d_{G-e}(w_1, U) = d_{G-e}(w_1, y')$.

By the way in which $x$ and $y$ were chosen,

\[
d_{G-e}(u_1, w') = d_{G-e}(y', w') \\
\leq d_{G-e}(w_1, y') + d_{G-e}(w, w') + 1 \\
= d_{G-e}(w_1, U) + d_{G-e}(w, U) + 1 \\
= d_G(w_1, U) + d_G(w, U) + 1 \\
\leq d_G(x, U) + d_G(y, U) + 1 \\
= d_{G-e}(x, U) + d_{G-e}(y, U) + 1 \\
= d_{G-e}(x, U) + d_{G-e}(y, u_1) + 1 \\
= d_{G-e}(x, U) + n_1 + 1 \\
= n \\
\leq s,
\]

which contradicts $d_{G-e}(u_1, U - u_1) > s$.

Case 2: The $u_1-x$ subpath $P_2$ of $P$ is in $G - e$.

The proof of this case is similar to Case 1. Without loss of generality, suppose that $u_2 = y'$. Let $n$ and $n_2$ be the lengths of the paths $P$ and $P_2$ respectively. Then $n = n_2 + 1 + d_{G-e}(y, U)$. If $u_1 \neq x'$, then

\[
d_{G-e}(u_1, U - u_1) \leq d_{G-e}(u_1, x')
\]
\[ d_{G-e}(u_1, x) + d_{G-e}(x, x') \]
\[ = n_2 + d_{G-e}(y, U) + 1 \]
\[ = n \]
which is a contradiction. So we may assume that \( u_1 = x' \).

The rest of the proof is exactly the same as the second part of Case 1 where \( u_1 = y' \), except we replace \( x, x', n_1 \) by \( y, y', n_2 \) respectively and vice versa.

A contradiction arises for Case 2.

Therefore in either case, a contradiction arises. Thus for any \( u_1 \in U \), there exists \( u_2 \in U \) such that \( d_{G-e}(u_1, u_2) \leq s \).

In addition we have established that \( U \) is a dominating set of radius \( r \) of \( G - e \). Therefore, \( U \) is an \((r, s)\)-dominating set of \( G - e \). This completes the proof. □

**Lemma 5.1** Let \( G = (V, E) \) be a nontrivial connected graph, and \( r \) and \( s \) be two positive integers. If \( \text{rad} \ G \leq r \), then \( \varphi_{r,s}(G) = 2 \).

**Proof:** Let \( v \) be a vertex in the center of \( G \) and \( u \) be a vertex adjacent to \( v \). Since \( \text{rad} \ G \leq r \), \( \{u, v\} \) is an \((r, s)\)-dominating set of \( G \). Therefore \( \varphi_{r,s}(G) = 2 \). □

**Lemma 5.1** is useful when establishing certain upper bounds on \( \varphi_{r,s}(G) \).

**Lemma 5.2** Let \( G \) be a graph without isolated vertices, \( r_1, s_1, r_2, \) and \( s_2 \) be positive integers such that \( r_1 \leq r_2 \) and \( s_1 \leq s_2 \). Then

\[ \varphi_{r_2, s_2}(G) \leq \varphi_{r_1, s_1}(G). \]
Proof: Lemma 5.2 follows from the fact that an \((r_1, s_1)\)-dominating set of \(G\) is also an \((r_2, s_2)\)-dominating set of \(G\), where \(r_1\), \(s_1\), \(r_2\), and \(s_2\) are positive integers such that \(r_1 \leq r_2\) and \(s_1 \leq s_2\).

Lemma 5.3 Let \(G=(V, E)\) be a graph, and \(r\) and \(s\) be two positive integers such that \(s \geq 2r+1\). A subset \(U\) of \(V\) is an \((r, s)\)-dominating set of \(G\) if and only if \(U\) is an \((r, 2r+1)\)-dominating set of \(G\).

Proof: It is clear that an \((r, 2r+1)\)-dominating set of \(G\) is an \((r, s)\)-dominating set of \(G\) for \(s \geq 2r+1\). Now suppose that \(U\) is an \((r, s)\)-dominating set of a graph \(G\), where \(s \geq 2r+1\). Then \(U\) is a dominating set of radius \(r\) of \(G\). For any vertex \(u_i \in U\), there exists \(u_2 \in U\) such that \(d_G(U\setminus\{u_i\}, u_i) = d_G(u_2, u_i)\). Denote \(d_G(u_2, u_i)\) by \(n\). Let \(P\) be a \(u_2-u_i\) path of length \(n\) in \(G\) and let \(v \in V(P)\) such that \(d_G(v, u_i) = \lfloor n/2 \rfloor\). If \(d_G(u_2, u_i) > 2r+1\), then \(d_G(U, v) > r\), contradicting that \(U\) is a dominating set of radius \(r\) of \(G\). Therefore \(d_G(u_2, u_i) \leq 2r+1\). Thus \(U\) is an \((r, 2r+1)\)-dominating set of \(G\).

By Lemma 5.3, for graphs, we need only consider \((r, s)\)-dominating sets and \((r, s)\)-domination numbers for \(s \leq 2r+1\).

The next algorithm will be used by Theorem 5.3.

Algorithm 5.1 SUBTREE_RS_DOMINATION(T, v, r, s, P, U, j)
/* This algorithm finds a minimum \((r, s)\)-dominating set for some subtree of \(T\), where \(s \leq r + 1\). */

INPUT

\(T\) is a tree with root \(v\) such that \(\text{rad } T > r\).
r and s are positive integers such that s ≤ r + 1.
P is a longest path in T with end-vertices u and v.
x and y are the vertices on P such that d(x, u) = r and
d(y, u) = r + s.
The x−y subpath of P is:
\[ X = v_0, v_1, \ldots, v_s = y. \]

OUTPUT

j is the index of vertex \( v_j \).
U is a minimum \((r, s)\)-dominating set for the subtree of T
with root \( v_j \).

begin

\[ U \leftarrow \{v\} \]

For i = 1 to s loop

For each child w ( ≠ \( v_{i-1} \)) of \( v_i \) loop

Let \( T_w \) be the subtree of T having root w
and let \( w' \) be a vertex in \( T_w \) such that
\[ e(w) = d(w, w'), \] where \( e(w) \) is the eccentricity of \( w \) in \( T_w \).

if \( e(w) ≥ r \) then

Let \( w'' \) be a vertex in \( T_w \) such that
\[ d(w', w'') = r \] and \[ d(w'', w) = e(v) - r. \]
\[ U \leftarrow U \cup \{w''\} \]

else

if \( e(w) = r - 1 \) then

\[ U \leftarrow U \cup \{v_i\} \]

endif
end loop

Let $T_{v_i}$ be the subtree of $T$ having root $v_i$.

if ($\exists z \in V(T_{v_i})$ such that $d(U, z) > r$ or ($v_i \in U$) then

$U \leftarrow U \cup \{v_i\}; \quad j \leftarrow i$

exit loop

endif

if ($i = s$) then

if ($|U| = 1$) or ($\exists z \in U$ such that $d(U-\{z\}, z) > s$) then

$U \leftarrow U \cup \{v_i\}$

endif

$j \leftarrow i$

endif

end loop

end Algorithm 5.1

Theorem 5.2 If $T$ is a tree and $r$ and $s$ are two positive integers such that $s \leq r + 1$, Algorithm 5.1 finds a minimum $(r, s)$-dominating set for some subtree of $T$.

Proof: Note that each vertex $u$ in $U - \{v_j\}$ is required to be in $U$ by an end-vertex descendant of $u$. If $v_j \in U$, then $v_j$ is required to be in $U$ to insure that $U$ is an $(r, s)$-dominating set of the subtree $T_{v_j}$ of $T$ with root $v_j$. \qed

The $(r, s)$-domination number of a disconnected graph can be very large, for example, $\delta_{r,s}(G) = |V(G)|$ for $G = mK_2$, $m \geq 1$. It is easy to see that $mK_2$ is the only graph with this property.
Cockayne et al. [9] have shown that for a connected graph of order \( n \geq 3 \),
\[
\varphi_t(G) \leq 2n / 3.
\]
Before presenting a generalization of this result, we define an \( r \)-star. An \( r \)-star is a graph which can be obtained from a set of disjoint paths of length \( r \) by identifying one end-vertex of each path to some fixed end-vertex of a path in the set. Thus each star is a 1-star.

**Theorem 5.3** Let \( G \) be a connected graph of order \( n \geq 2 \), and \( r \) and \( s \) be two positive integers such that \( s \leq r + 1 \). Then
\[
\varphi_{r,s}(G) \leq \max \{ 2n / (r + s + 1), 2 \}.
\]
Furthermore, this bound is sharp.

**Proof:** By Theorem 5.1, we need only show that for any tree \( T \) of order \( n \geq 2 \) and \( s \leq r + 1 \),
\[
\varphi_{r,s}(T) \leq \max \{ 2n / (r + s + 1), 2 \}.
\]
The proof is by induction on \( n \). Let \( T = (V, E) \) be a tree of order \( n \geq 2 \). If \( \text{rad} \, T \leq r \), then by Lemma 5.1,
\[
\varphi_{r,s}(T) = 2 \leq \max \{ 2n / (r + s + 1), 2 \}.
\]
Consequently,
\[
\varphi_{r,s}(T) = 2,
\]
for any nontrivial tree of order at most \( 2r + 1 \).

Now suppose that for any tree \( T' \) of order \( m \), \( 2 \leq m < n \),
\[
\varphi_{r,s}(T') \leq \max \{ 2m / (r + s + 1), 2 \},
\]
and \( T \) is a tree of order \( n \) such that \( \text{rad} \, T > r \). Let \( P \) be a longest path in \( T \), \( u \) and \( v \) be the end-vertices of \( P \), and \( k \) be the length of \( P \). Since \( \text{rad} \, T > r \), \( k \geq 2r + 1 \).
Let $x$ and $y$ be the vertices of $P$ such that $d(x, u) = r$ and $d(y, u) = r + s$, and the $x$-$y$ subpath of $P$ be:

$$x = v_0, v_1, \ldots, v_s = y.$$  

In the following, the tree $T$ is treated as a rooted tree with root $v$.

Use Algorithm 5.1 to find a minimum $(r, s)$-dominating set $U$ of some subtree $T_{v_j}$ of $T$ with root $v_j$, where $j$ is the integer returned from Algorithm 5.1. For each vertex $v$ in $U$, there is a set $S_v$ of vertices such that $|S_v| \geq r + 1$ for $v = v_s$, and $|S_v| \geq s$ for $v = v_s$ if $v_s \in U$. Each vertex in $S_v$ is within distance $r$ from $v$, and $S_v \cap S_{v'} = \emptyset$ if the corresponding vertices $v$ and $v'$ in $U$ are different. Let $t = d_T(U, v_j)$ and $d = |U| = \sigma_{r,s}(T_{v_j})$. If $t = 0$, then $|V(T_{v_j})| \geq (d - 1) \cdot (r + 1) + s$; otherwise $|V(T_{v_j})| \geq d(r + 1) + t$.

Let $T'$ be the subtree of $T$ obtained from $T$ by removing the subtree rooted at $v_j$ (including vertex $v_j$) from $T$, and let $n'$ be the order of $T'$. Then $n' < n$, by the inductive hypotheses,

$$\sigma_{r,s}(T') \leq \max \{ 2n' / (r + s + 1), 2 \}.$$  

We consider three cases:

Case 1: $2n' / (r + s + 1) < 1$.

Since $s \leq r + 1$, $n' < (r + s + 1) / 2 \leq r + 1$, it follows that $n' \leq r$.

If $t + n' \leq r$, then $U$ is an $(r, s)$-dominating set of $T$.

Since $U$ is a minimum $(r, s)$-dominating set of $T_{v_j}$, $U$ is necessarily a minimum $(r, s)$-dominating set of $T$. By the inductive hypotheses,
\[ \sigma_{r,s}(T) = \sigma_{r,s}(T_{v_j}) \]
\[ \leq 2(n - n') / (r + s + 1) \]
\[ < 2n / (r + s + 1). \]

Otherwise, \( t + n' \geq r + 1 \). Since \( n' \leq r, t \geq 1 \). This implies that \(|V(T_{v_j})| \geq d \cdot (r + 1) + t\). Therefore \( n = |V(T)| = |V(T_{v_j})| + n' \geq (d + 1) \cdot (r + 1) \). Note that \( U' = U \cup \{v_j\} \) is an \((r, s)\)-dominating set of \( T \). Since \( s \leq r+1 \), we have
\[ \sigma_{r,s}(T) \leq |U'| \]
\[ = d + 1 \]
\[ = (d + 1) \frac{r + s + 1}{r + s + 1} \]
\[ \leq 2(d + 1) (r + 1) / (r + s + 1) \]
\[ \leq 2n / (r + s + 1). \]

Case 2: \( 1 \leq 2n' / (r + s + 1) < 2 \).

In this case \( n' < r + s + 1 \). Since \( n' \) is a positive integer, \( n' \leq r + s \).

If \( t + n' \leq r + s \), let \( S' = \{ v \in V(T') \mid d_{T}(U, v) = s \} \). If \( S' = \emptyset \), then let \( u' \) be a vertex in \( S' \), otherwise let \( u' \) be any fixed vertex in \( T' \). Then \( U \cup \{u'\} \) is an \((r, s)\)-dominating set of \( T \), thus
\[ \sigma_{r,s}(T) \leq 1 + |U| \]
\[ \leq 2n' / (r + s + 1) + |U| \]
\[ \leq 2n' / (r + s + 1) + 2(n - n') / (r + s + 1) \]
\[ = 2n / (r + s + 1). \]

Otherwise, \( t + n' \geq r + s + 1 \). Since \( n' \leq r + s \), \( t \geq 1 \).

Therefore \(|V(T_{v_j})| \geq d \cdot (r + 1) + t\). It follows that \( n = |V(T)| = |V(T_{v_j})| + n' \geq (d + 1) \cdot (r + 1) + s \). Let \( S' = \{ v \in V(T') \mid d_{T}(v_j, v) = s \} \). If \( S' = \emptyset \), then let \( u' \) be a vertex in \( S' \), otherwise let \( u' \) be a
fixed vertex in the center of \( T' \). Then \( U \cup \{v_j, u'\} \) is an \((r, s)\)-dominating set of \( T \), thus

\[
\sigma_{r,s}(T) \leq |U| + 2
\]

\[
= d + 2
\]

\[
= d \frac{r + s + 1}{r + s + 1} + 2
\]

\[
\leq 2d \frac{r + 1}{r + s + 1} + 2
\]

\[
= 2(d + 1) \frac{r + 1 + s}{r + s + 1}
\]

\[
\leq 2n \frac{d + 1}{r + s + 1}
\]

where \( s \leq r + 1 \).

**Case 3**: \( 2n' / (r + s + 1) \geq 2 \).

In this case \( \sigma_{r,s}(T') \leq 2n' / (r + s + 1) \). Note that the union of an \((r, s)\)-dominating set of \( T' \) and an \((r, s)\)-dominating set of \( T_{v_j} \) is an \((r, s)\)-dominating set of \( T \). Thus, by induction

\[
\sigma_{r,s}(T) \leq \sigma_{r,s}(T') + \sigma_{r,s}(T_{v_j})
\]

\[
\leq 2n' / (r + s + 1) + 2(n - n') / (r + s + 1)
\]

\[
= 2n / (r + s + 1).
\]

By mathematical induction,

\[
\sigma_{r,s}(T) \leq \max \left\{ 2n / (r + s + 1), 2 \right\},
\]

for all trees \( T \) of order \( n \geq 2 \) and \( s \leq r + 1 \). Thus

\[
\sigma_{r,s}(G) \leq \max \left\{ 2n / (r + s + 1), 2 \right\},
\]

for all connected graphs \( G \) of order \( n \geq 2 \) and positive integers \( r \) and \( s \) such that \( s \leq r + 1 \).

Now we show that this bound is sharp. We need only show that the bound \( 2n / (r + s + 1) \) is obtainable under the assumption that \( n \geq r + s + 1 \). Let \( \mathcal{G}_{r+s+2} \) be the set of graphs each of which
can be obtained by taking an end-vertex from an \((r + s + 2)\)-star graph. By observation, for \(n \geq r + s + 1\), the upper bound 
\[\frac{2n}{r + s + 1}\] is obtainable by all the graphs in \(\mathcal{S}_{r+s+2}\). Thus the bound is sharp. □

The graph \(G\) represented by Figure 5.2 is a graph in \(\mathcal{S}_{r+s+2}\), where \(r = 3\) and \(s = 2\). \(G\) has order \(n = 24\). The set of solid vertices is an \((r, s)\)-dominating set of \(G\) of cardinality \(\mathcal{X}_{r,s}(G)\) which is equal to \(\frac{2n}{r + s + 1}\).

![Figure 5.2](image)

As a consequence, we have that for any \((r + s + 2)\)-star \(G\) of order \(n\) and positive integers \(r\) and \(s\) such that \(s \leq r + 1\),
\[\mathcal{X}_{r,s}(G) = \frac{2(n - 1)}{r + s + 1}\]

**Theorem 5.4** Let \(G\) be a connected graph of order \(n \geq 2\), and \(r\) and \(s\) be two positive integers such that \(s \geq r + 1\). Then
\[\mathcal{X}_{r,s}(G) \leq \max \{n / (r + 1), 2\}\] Furthermore, this bound is sharp.
Proof: Since \( s \geq r + 1 \), by Lemma 5.2 and Theorem 5.3, we have

\[
\gamma_{r,s}(G) \leq \gamma_{r,r+1}(G) \\
\leq \max \left\{ \frac{n}{r + 1}, 2 \right\}.
\]

To show this bound is sharp, we need only show that the bound \( n/(r + 1) \) is obtainable under the assumption that \( n \geq 2(r + 1) \). Let \( \mathcal{B}_{r+1} \) be the set of graphs each of which can be obtained by taking an end-vertex from an \( (r + 1) \)-star graph. By observation, for \( n \geq 2(r + 1) \), the upper bound \( n/(r + 1) \) is obtainable by all the graphs in \( \mathcal{B}_{r+1} \). Thus the bound is sharp. \( \square \)

Let \( G = (V, E) \) be a graph and \( r \) be a nonnegative integer. Define \( \text{End}_r(G) = \{ v \in V | \exists \text{ an end-vertex } u \in V \text{ such that } d(u, v) < r \} \). Note that \( \text{End}_1(G) \) is the set of end-vertices in \( G \).

Theorem 5.5 Let \( G \) be a connected graph of order \( n \geq 2 \), and \( r \) and \( s \) be two positive integers. Then

\[
\gamma_{r,s}(G) \leq \max \{ 2, \min \{ n - |\text{End}_r(T)| \} \}
\]

where the minimum is taken over all spanning trees \( T \) of \( G \).

Proof: Let \( T \) be a spanning tree of \( G = (V, E) \) and \( U = V - \text{End}_r(T) \). Then \( |U| = n - |\text{End}_r(T)| \). If \( |U| \geq 2 \), define \( U' \) to be the set \( U \), otherwise define \( U' \) to be the union of \( U \) and \( 2 - |U| \) end-vertices of \( T \). Clearly, \( U' \) is an \( (r, s) \)-dominating set of \( T \). Therefore,

\[
\gamma_{r,s}(T) \leq |U'| \leq \max \{ 2, n - |\text{End}_r(T)| \}.
\]

Thus

\[
\gamma_{r,s}(G) \leq \min \gamma_{r,s}(T) \\
\leq \min \max \{ 2, n - |\text{End}_r(T)| \} \\
= \max \{2, \min \{ n - |\text{End}_r(T)| \} \},
\]
where the minimum is taken over all spanning trees \( T \) of \( G \). \( \square \)

The following result has been obtained by Cockayne et al. [9]: If \( G \) is a connected graph of order \( n \) such that \( \Delta(G) < n - 1 \), then \( \sigma_t(G) \leq n - \Delta \). This result is generalized by Theorem 5.6.

Denote the set of end-vertices of a tree \( T \) by \( \text{End}(T) \). By observation, \( |\text{End}(T)| \geq \Delta(T) \). Theorem 5.6 relates the \((r, s)\)-domination number and the maximum degree of a graph.

**Theorem 5.6** Let \( G \) be a connected graph of order \( n \geq 2 \) with maximum degree \( \Delta = \Delta(G) \), and \( r \) and \( s \) be two positive integers such that \( s \leq r + 1 \). Then

\[
\sigma_{r,s}(G) \leq \max \{ 2, n - (r + s + \Delta) + 2 \}.
\]

**Proof:** Let \( r \) and \( s \) be two positive integers such that \( s \leq r + 1 \).

By Theorem 5.1, it is sufficient to show that

\[
\sigma_{r,s}(T) \leq \max \{ 2, n - (r + s + \Delta) + 2 \},
\]

for any tree \( T \) of order \( n \geq 2 \) with maximum degree \( \Delta = \Delta(T) \).

If \( \text{rad} \, T \leq r \), then by Lemma 5.1, \( \sigma_{r,s}(T) = 2 \). So we may assume that \( \text{rad} \, T > r \). Let \( P \) be a longest path in \( T \) with end-vertices \( u \) and \( v \). Then there exist vertices \( x \) and \( y \) of \( P \) such that \( d(x, u) = r \) and \( d(y, u) = r + s \). Let \( P' \) be the \( u-y \) subpath of \( P \), \( V' = V(P') - \{x, y\} \), and \( U = V(T) - (V' \cup \text{End}(T)) \). Then \( \{x, y\} \subseteq U \), it follows that \( |U| \geq 2 \). Thus \( U \) is an \((r, s)\)-dominating set of \( T \).

Since \( u \in V' \cap \text{End}(T) \) and \( |\text{End}(T)| \geq \Delta(T) \), where \( \text{End}(T) \) is the set of end-vertices of \( T \), we have

\[
\sigma_{r,s}(T) \leq |V(T)| - |V' \cup \text{End}(T)|
\]
\[
\leq |V(T)| - |V'| - |\text{End}(T)| + 1
\]
\[
\leq n - (r + s - 1) - \Delta(T) + 1.
\]
Corollary 5.6 Let $G$ be a graph of order $n$ which contains no isolated vertices and there exists a component $C$ of $G$ such that $\Delta(C) = \Delta(G)$ and $|V(C)| \geq r + s + \Delta(G)$, where $r$ and $s$ are positive integers such that $s \leq r + 1$. Then

$$\sigma_{r,s}(G) \leq n - (r + s + \Delta) + 2.$$

Proof: Let $U$ be a minimum $(r, s)$-dominating set in a component $C$ of $G$ such that $\Delta(C) = \Delta(G)$ and $n' = |V(C)| \geq r + s + \Delta(G)$. Since $G$ contains no isolated vertices, $(V(G) - V(C)) \cup U$ is an $(r, s)$-dominating set of $G$. By Theorem 5.5,

$$\sigma_{r,s}(G) \leq |V(G) - V(C)| + |U|$$

$$= n - n' + \sigma_{r,s}(C)$$

$$\leq n - n' + n' - (r + s + \Delta(C)) + 2$$

$$= n - (r + s + \Delta(G)) + 2 \quad \Box$$

The following theorem gives a lower bound for $\sigma_{r,s}(G)$ in terms of the diameter of a graph $G$ and $r, s$.

Theorem 5.7 Let $G$ be a graph which contains no isolated vertices, and $r$ and $s$ be two positive integers. Then

$$\sigma_{r,s}(G) \geq 2 \left\lfloor \frac{\text{diam}(G) + 1}{2r + s + 1} \right\rfloor.$$

Proof: Let $u$ be a vertex in $G = (V, E)$ such that $d(u) = \text{diam}(G)$ and $U$ be a minimum $(r, s)$-dominating set of $G$. Then $\sigma_{r,s}(G) = |U|$. Define $L_i = \{v \in V \mid d(u, v) = i\}$, $0 \leq i \leq \text{diam}(G)$, and $L_j = \emptyset$, $j > \text{diam}(G)$. A set $L_i$ is said to be dominated by some set $S$ if each element in $L_i$ is dominated by $S$. Observe that any two vertices in $U$ alone can dominate at most $2r+s+1$ $L_i$'s in $G$ within distance $r$. Therefore, $|U \cap (L_k \cup L_{k+1} \cup \ldots \cup L_{k+2r+s})| \geq 2$, for $k$
It follows that
\[ \varphi_{r,s}(G) = |U| \geq 2 \left\lfloor \frac{(\text{diam}(G)+1)}{(2r+s+1)} \right\rfloor. \]
\[ \square \]

Computational Complexities

\( (r,s) \)-Dominating Set

Instance: Graph \( G = (V, E) \) which does not contain isolated vertices, positive integers \( r, s \), and \( K \) such that \( s \leq 2r + 1 \) and \( K \leq |V| \).

Question: Is there an \( (r, s) \)-dominating set of size \( K \) or less, i.e., a subset \( U \subseteq V \) such that for any \( v \in V - U \), there exists \( u \in U \) for which \( d(u, v) \leq r \), and for any \( u_1 \in U \), there exists \( u_2 \in U \) (\( u_2 \neq u_1 \)) such that \( d(u_2, u_1) \leq s \)?

Digraph \( (r, s) \)-Dominating Set

Instance: Digraph \( D = (V, A) \) each vertex of which has in-degree \( > 0 \), positive integers \( r, s \), and \( K \) such that \( K \leq |V| \).

Question: Is there an \( (r, s) \)-dominating set of size \( K \) or less, i.e., a subset \( U \subseteq V \) such that for any \( v \in V - U \), there exists \( u \in U \) for which \( d(u, v) \leq r \), and for any \( u_1 \in U \), there exists \( u_2 \in U \) (\( u_2 \neq u_1 \)) such that \( d(u_2, u_1) \leq s \)?

\( m \)-Centrix \( (r, s) \)-Domination Augmentation

Instance: Graph \( G = (V, E) \), a subset \( C \) of \( V \) of cardinality \( m \geq 2 \), called an \( m \)-centrix, positive integers \( B, r, \) and \( s \) such that \( s \leq 2r + 1 \).

Question: Is there a set \( E' \) of unordered pairs of vertices from \( V \)
such that $|E'| \leq B$ and $C$ is an $(r, s)$-dominating set of $G' = (V, E \cup E')$, i.e., $d_{G'}(C, v) \leq r$ for all $v \in V$, and for any $c \in C$, there exists $c' \in C$ ($c' \neq c$) such that $d_{G'}(c', c) \leq s$?

**Digraph $m$-Centrix $(r, s)$-Domination Augmentation**

**Instance:** Digraph $D = (V, A)$, a subset $C$ of $V$ of cardinality $m \geq 2$, called an $m$-centrix, positive integers $B$, $r$, and $s$.

**Question:** Is there a set $A'$ of ordered pairs of vertices from $V$ such that $|A'| \leq B$ and $C$ is an $(r, s)$-dominating set of $D' = (V, A \cup A')$, i.e., $d_{D'}(C, v) \leq r$ for all $v \in V$, and for any $c \in C$, there exists $c' \in C$ ($c' \neq c$) such that $d_{D'}(c', c) \leq s$?

We consider $(r, s)$-Dominating Set and Digraph $(r, s)$-Dominating Set problems first.

**Theorem 5.8** Let $G = (V, E)$ be a connected graph, $r$ be a positive integer, and $U \subseteq V$ such that $|U| > 2$. Then $U$ is a dominating set of radius $r$ of $G$ if and only if $U$ is an $(r, 2r + 1)$-dominating set of $G$.

**Proof:** Clearly if $U$ is an $(r, 2r + 1)$-dominating set of $G$, then $U$ is a dominating set of radius $r$ of $G$.

On the other hand, assume that $G$ is a connected graph, $r$ is a positive integer, and $U$ is a dominating set of radius $r$ of $G$ such that $|U| \leq 2$. We show that $U$ is an $(r, 2r + 1)$-dominating set of $G$. Suppose, to the contrary, that there exists $u_i \in U$ such that $d_G(U - \{u_i\}, u_i) > 2r + 1$. Since $|U| \leq 2$, there exists $u_2 \in U$ such that $d_G(U - \{u_i\}, u_i) = d_G(u_2, u_i)$. Denote $d_G(u_2, u_i)$ by $n$. Let $P$ be a $u_2 - u_1$ path of length $n$ in $G$ and let $v \in V(P)$ such that $d_G(v, u_i) = \lfloor n/2 \rfloor$. Then $d_G(U, v) > r$, contradicting that $U$ is a dominating...
set of radius $r$ of $G$. Therefore $d_G(U-(u), u) \leq 2r + 1$ for all $u \in U$.

Thus $G$ is an $(r, 2r+1)$-dominating set of $G$. □

**Corollary 5.8** Let $G = (V, E)$ be a connected graph and $r$ be a positive integer such that $\delta_r(G) \geq 2$. Then $\delta_{r, 2r+1}(G) = \delta_r(G)$. □

Let $G$ be a connected graph or digraph and $r$ be an arbitrary positive integer. Observe that $\delta_r(G) = 1$ if and only if $\text{rad } G \leq r$.

It can easily be shown that the problem of determining whether $\text{rad } G \leq r$ can be solved in polynomial time. Thus we have

**Lemma 5.4** Let $G$ be a connected graph or digraph and $r$ be an arbitrary positive integer. The problem of determining whether $\delta_r(G) = 1$ can be solved in polynomial time. □

By Corollary 5.8, Lemma 5.4, and Theorem 2.1, we have

**Theorem 5.9** For any fixed positive integer $r$, the $(r, s)$-Dominating Set problem is NP-complete even in the case that the graph is a connected planar graph of maximum degree 3 and of minimum degree 1. □

By an argument similar to that of Theorem 5.8, it can be shown that for a connected symmetric digraph $D = (V, A)$, a positive integer $r$, and $U \subseteq V$ such that $|U| \geq 2$, $U$ is a dominating set of radius $r$ of $D$ if and only if $U$ is an $(r, 2r+1)$-dominating set of $D$. This result together with Lemma 5.4 and Theorem 2.2 give

**Theorem 5.10** For any fixed positive integer $r$, the Digraph $(r, s)$-Dominating Set problem is NP-complete even in the case that the digraph is symmetric and its underlying graph is connected planar of maximum degree 3 and of minimum degree 1. □
Let $T$ be an oriented tree no vertex of which has in-degree 0. An algorithm for finding a minimum $(r, s)$-dominating set of $T$ can be designed in a manner similar to Algorithm 2.1.

**Theorem 5.11** There is a linear time algorithm which finds a minimum $(r, s)$-dominating set of an oriented tree for which no vertex has in-degree 0. □

Similarly,

**Theorem 5.12** There is a linear time algorithm which finds a minimum $(r, s)$-dominating set of an arbitrary nontrivial tree. □

Now we consider $m$-Centrix $(r, s)$-Domination Augmentation and Digraph $m$-Centrix $(r, s)$-Domination Augmentation problems.

**Theorem 5.13** For $r = 1$, the problem of finding a minimum digraph $m$-centrix $(r, s)$-domination augmentation can be solved in $O(m^2|V|^2)$ time.

**Proof:** Let $D = (V, A)$ be a nontrivial digraph and $C = \{c_1, c_2, \ldots, c_m\} \subseteq V$ be an $m$-centrix, $m \geq 2$. Denote the minimum number of arcs to be added to the digraph $D$ so that $C$ is a $(1, s)$-dominating set of the resulting digraph by $\text{Aug}(D)$. We consider two cases:

Case 1: $s = 1$.

In this case, the problem is to find a minimum set of arcs $A'$ to be added to $D$ so that in the resulting digraph $D' = (V, A \cup A')$, every vertex in $V$ is adjacent from some vertex in the $m$-centrix $C$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
For each vertex $c_i$, $1 \leq i \leq m$, consider the set $V_i$ of vertices adjacent from $c_i$. Let $V' = V_1 \cup V_2 \cup \ldots \cup V_m$ and $V'' = V - V'$. Then $V''$ is the set of vertices which are not adjacent from vertices in $C$. For each vertex $v$ in $V''$, find a vertex $c_v$ in $C$ such that $v \neq c_v$. Let $A' = \{(c_v, v) \mid v \in V''\}$. Then in the digraph $D' = (V, A \cup A')$, every vertex in $V$ is adjacent from some vertex in $C$ and $|A'| = |V''|$.  

On the other hand, to ensure that each vertex in $V''$ is adjacent from some vertex in $C$, at least $|V''|$ number of arcs need to be added to $D$. Therefore Aug$(D) \geq |V''|$. Thus $A'$ is a desired set of arcs.

Case 2: $s \geq 2$.

For each $i$, $1 \leq i \leq m$, find the set $V_i$ of vertices adjacent from $c_i$. Let $V' = V_1 \cup V_2 \cup \ldots \cup V_m$ and $V'' = V - (V' \cup C) = \{v_1, v_2, \ldots, v_k\}$. Then $V''$ is the set of vertices in $V - C$ which are not adjacent from $C$. Clearly Aug$(D) \geq |V''|$. We consider three subcases.

Subcase 2.1: For each $v_i \in V''$, $1 \leq i \leq k$, there exists $c_{j_i} \in C$ such that $d_{C}(v_i, c_{j_i}) \geq s$.

Let $A_1 = \{(c_{j_i}, v_i) \mid 1 \leq i \leq k\}$, $D_1 = (V, A \cup A_1)$, and let $C'$ be the subset of $C$ such that for each $c' \in C'$, $d_{C'}(C - c', c') > s$. Note that $C'$ can be found in $O(m^2|V|^2)$ time. Clearly, in the digraph $D_1$, every vertex in $V - C$ is adjacent from some vertex in $C$. If $C' = \emptyset$, then Aug$(D) = |V''| = |A_1|$. So we may assume that $C' = \{c_1', c_2', \ldots, c_t'\} = \emptyset$. For each $j$, $1 \leq j \leq t$, let $c_{j''}$ be a vertex in $C$ such that $c_{j''} = c_j'$. Then for the digraph $D' =$
\[(V, A \cup A'), \text{ where } A' = A_1 \cup \{(c_j^n, c_j') | 1 \leq j \leq t\}, \text{ C is a } (1, s)\text{-dominating set of } D'.\]

We show that \(|A'| = \text{Aug}(D)\). Suppose that \(A_2\) is a set of arcs such that \(C\) is a \((1, s)\)-dominating set of \(D_2 = (V, A \cup A_2)\).

Then \(A_2\) contains a set \(A_3\) of arcs of cardinality \(k (= |V''|)\) such that each arc of \(A_3\) joins a central vertex with some \(v_i, 1 \leq i \leq k,\) and no arc in \(A_3\) is incident to more than one \(v_i, 1 \leq i \leq k.\) We consider the digraph \(D_3 = (V, A \cup A_3).\) Let \(C''\) be the subset of \(C\) such that for each \(c'' \in C'',\) \(d_{D_3}(C - c'', c'') > s.\) By the way in which the arcs in \(A_1\) were chosen, we have \(|C'| \leq |C''|\). Note that in the digraph \(D_3,\) each vertex \(v \in V - C\) is adjacent from some vertex in \(C.\) This implies that for each \(c'' \in C'',\) there exists an arc \(e'' = (u'', c'') \in A_2 - A_3, \) for some \(u'' \in V; \) for otherwise, we would have \(d_{D_2}(C - c'', c'') > s,\) which is a contradiction. Therefore \(|A_2 - A_3| \geq |C''|\). Since \(A_3\) is a subset of \(A_2,\) we have

\[
|A'| = |A_1| + |C'|
\]

\[
\leq |A_3| + |C''|
\]

\[
\leq |A_3| + |A_2 - A_3|
\]

\[
= |A_2|.
\]

Thus \(|A'| = \text{Aug}(D)\).

In the following subcases, there exists \(v_{i_0} \in V'', 1 \leq i_0 \leq k,\) such that \(d_D(v_{i_0}, c) \leq s - 1, \) for all \(c \in C.\)

Subcase 2.2: There exist \(v_{i_0}, v_{j_0} \in V''\) and \(c' \in C, 1 \leq i_0, j_0 \leq k,\) \(i_0 \neq j_0,\) such that \(d_D(v_{i_0}, c) \leq s - 1\) for all \(c \in C,\) and \(d_D(v_{j_0}, c') \leq s - 1.\)

Let \(c''\) be a vertex in \(C\) such that \(c'' = c'\) and let \(A' = (c'', v_{j_0}) \cup \)
\{(c', v_i) \mid i = j_0, 1 \leq i \leq k\}, \text{ then } |A'| = |V''| \leq \text{Aug}(D)$. We show that $C$ is a $(1, s)$-dominating set of $D'$, where $D' = (V, A \cup A')$. Clearly $C$ is a dominating set of radius 1 of $D'$. Since for all $c \in C$ (c.e. $c'$), $d_{D'}(c', c) \leq d_{D'}(c', v_{i_0}) + d_{D'}(v_{i_0}, c) \leq 1 + (s - 1) = s$, and $d_{D'}(c'', c') \leq d_{D'}(c'', v_{j_0}) + d_{D'}(v_{j_0}, c') \leq s$, $c'' = c'$, $C$ is a $(1, s)$-dominating set of $D'$. Therefore $A'$ is a desired set of arcs.

Subcase 2.3: For any $v_{i_0} \in V''$, $1 \leq i_0 \leq k$, such that $d_{D}(v_{i_0}, c) \leq s - 1$ for all $c \in C$, there does not exist $i, i \neq i_0, 1 \leq i \leq k$, such that $d_{D}(v_i, C) \leq s - 1$.

Note that in this subcase, there is only one vertex $v_{i_0} \in V''$, $1 \leq i_0 \leq k$, such that $d_{D}(v_{i_0}, c) \leq s - 1$ for all $c \in C$. Let $A' = \{(c_1, v_i) \mid 1 \leq i \leq k\} \cup (c_2, c_1)$ and $D' = (V, A \cup A')$. Then $|A'| = |V''| + 1$ and $C$ is a $(1, s)$-dominating set of $D'$. We show that $\text{Aug}(D) = |A'|$. Suppose that $A_1$ is a set of arcs such that $C$ is a $(1, s)$-dominating set of $D_1 = (V, A \cup A_1)$. Then $A_1$ contains a set $A_2$ of arcs of cardinality $k = |V''|$ such that each arc of $A_2$ joins a central vertex with some $v_i, 1 \leq i \leq k$, and no arc in $A_2$ is incident to more than one $v_i, 1 \leq i \leq k$. We consider the digraph $D_2 = (V, A \cup A_2)$. Let $(c', v_{i_0})$ be the arc in $A_2$ incident to $v_{i_0}$. Since for all $i = i_0, 1 \leq i \leq k$, $d_{D}(v_i, C) \geq s$, we have $d_{D_2}(C - c', c') > s$, it follows that $|A_1| \geq |A_2| + 1 = |A'|$. Thus $\text{Aug}(D) = |A'|$.

The time complexity to find $A'$ is $O(m^2|V|^2)$. \hfill \Box

Note that if the distance matrix of the digraph is given, then the problem of finding a minimum digraph $m$-centrix $(1, s)$-domination augmentation can be solved in $O(m^2|V|)$ time.
Corollary 5.13 For $r = 1$, the Digraph $m$-Centrix $(r, s)$-Domination Augmentation problem can be solved in $O(m^2 \cdot |V|^2)$ time.

The next theorem presents a corresponding result for graphs.

Theorem 5.14 For $r = 1$, the problem of finding a minimum $m$-centrix $(r, s)$-domination augmentation can be solved in $O(m^2 \cdot |V|^2)$ time.

Proof: Let $G = (V, E)$ be a nontrivial graph and $C = \{c_1, c_2, \ldots, c_m\}$ $\subseteq V$ be an $m$-centrix, and let $\text{Aug}(G)$ be the $(1, s)$ augmentation number of $G$ with respect to $C$. We consider two cases.

Case 1: $s = 1$.

In this case, the problem is equivalent to finding a minimum set of edges $E'$ to be added to $G$ so that $C$ is a total dominating set of the resulting graph $G' = (V, E \cup E')$.

For each $i$, $1 \leq i \leq m$, we consider the set $V_i$ of vertices adjacent to $c_i$. Let $V' = V_1 \cup V_2 \cup \ldots \cup V_m$ and $V'' = V - V'$. Then $V''$ is the set of vertices which are not adjacent to $C$. For each vertex $v$ in $V''$, find a vertex $c_v$ in $C$ such that $v = c_v$. Let $E' = \{c_vv \mid v \in V''\}$. Then $|E'| = |V''|$ and $C$ is a total dominating set of $G' = (V, E \cup E')$.

To ensure that each vertex in $V''$ is adjacent to some vertex in $C$, at least $|V''|$ number of edges must be added to $G$. Therefore $E'$ is a minimum augmentation. Note that $E'$ can be found in $O(m \cdot |V|)$ time.

Case 2: $s \geq 2$. 

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
For each $i$, $1 \leq i \leq m$, find the set $V_i$ of vertices adjacent from $c_i$. Let $V' = V_1 \cup V_2 \cup \ldots \cup V_m$ and $V'' = V - (V' \cup C) = \{v_1, v_2, \ldots, v_k\}$. Then $V''$ is the set of vertices in $V - C$ which are not adjacent from $C$. Clearly $\text{Aug}(G) \geq |V''|$. We consider three subcases.

Subcase 2.1: For each $v_i \in V''$, $1 \leq i \leq k$, there exists $c_j \in C$ such that $d_{G}(v_i, c_j) \geq s$.

Let $E_1 = \{c_j, v_i | 1 \leq i \leq k\}$, $G_1 = (V, E \cup E_1)$, and let $C'$ be the subset of $C$ such that for each $c' \in C'$, $d_{G_1}(C - c', c') > s$. Note that $C'$ can be found in $O(m^2 |V|^2)$ time. Clearly, in the graph $G_1$, every vertex in $V - C$ is adjacent from some vertex in $C$. If $C' = \emptyset$, then $\text{Aug}(G) = |V''| = |E_1|$. So we may assume that $C' = \{c_1', c_2', \ldots, c_t'\} \neq \emptyset$. Define $c_{t+1}'$ to be some fixed vertex in $C$ other than $c_{\lceil t/2 \rceil}'$ and let $E' = E_1 \cup \{c_i'c_{i+\lceil t/2 \rceil}' | 1 \leq i \leq \lceil t/2 \rceil\}$. Then $C$ is a $(1, s)$-dominating set of $G' = (V, E \cup E')$.

We show that $|E'| = \text{Aug}(G)$. Suppose that $E_2$ is a set of edges such that $C$ is a $(1, s)$-dominating set of $G_2 = (V, E \cup E_2)$. Then $E_2$ contains a set $E_3$ of edges of cardinality $k (= |V''|)$ such that each edge of $E_3$ joins a central vertex with some $v_i$, $1 \leq i \leq k$, and no edge in $E_3$ is incident to more than one $v_i$, $1 \leq i \leq k$.

We consider the graph $G_3 = (V, E \cup E_3)$. Let $C''$ be the subset of $C$ such that for each $c'' \in C''$, $d_{G_3}(C - c'', c'') > s$. By the way in which the edges in $E_1$ were chosen, we have $|C'| \leq |C''|$. Note that in the graph $E_3$, each vertex $v \in V - C$ is adjacent from some vertex in $C$. This implies that for each $c'' \in C''$, there exists an edge $e'' = u''c'' \in E_2 - E_3$, for some $u'' \in V$; for otherwise, we
would have $d_{G_2}(C - c'', c'') > s$, which is a contradiction. Therefore $|E_2 - E_3| \geq \lceil |C''|/2 \rceil$. By a similar argument as in Subcase 2.1 of Theorem 5.13, we have $|E'| \leq |E_2|$. Thus $|E'| = \text{Aug}(G)$.

Subcases 2.2 and 2.3 correspond to Subcases 2.2 and 2.3 in Theorem 5.13 and can be proved similarly.

In each case above, the desired set $E'$ of edges can be found in $O(m^2 \cdot |V|^2)$ time.

**Corollary 5.14a** For $r = 1$, the $m$-Centrix $(r, s)$-Domination Augmentation problem can be solved in $O(m^2 \cdot |V|^2)$ time.

**Corollary 5.14b** Let $G = (V, E)$ be a nontrivial graph and let $C$ be a subset of $V$, $|C| = m \geq 2$. The problem of finding a minimum set of edges $E'$ to be added to $G$ so that $C$ is a total dominating set of the resulting graph $G' = (V, E \cup E')$ can be solved in $O(m \cdot |V|)$ time.

By Corollary 5.8, Lemma 5.4, Theorem 2.9 and Theorem 2.10, we have

**Theorem 5.15** For any fixed positive integers $m$ and $r$, $r \geq 2$, the $m$-Centrix $(r, s)$-Domination Augmentation problem is NP-complete even in the case that the graph is connected planar of maximum degree 3.

**Theorem 5.16** For any fixed positive integers $m$ and $r$, $r \geq 2$, the Digraph $m$-Centrix $(r, s)$-Domination Augmentation problem is NP-complete even in the case that the digraph is symmetric and its underlying graph is connected planar of maximum degree 3.
CHAPTER VI

ON BLOCK-COMPLETE GRAPHS

The class of graphs whose blocks are complete subgraphs is denoted by $\mathcal{B}_c$. Graphs in $\mathcal{B}_c$ are called block-complete graphs.

There are a number of graph problems which are not known to be polynomial, but if the graphs are restricted to be trees (or forests), many of these problems can be solved in polynomial time. In this chapter, polynomial time algorithms are presented for solving some graph problems on the domain $\mathcal{B}_c$.

A graph $H$ of order $n$ is complete if and only if $H$ has $n(n - 1)/2$ edges. Thus the completeness of a graph can be determined in linear time. Therefore, given a graph $G$ and a block $B$ of $G$, the completeness of the block $B$ can be determined in linear time. Hopcroft and Tarjan [32] have proved that the components of a graph and the blocks of a connected graph $G = (V, E)$ can be found in linear time $O(|E| + |V|)$. Thus we have Theorem 6.1 There is a linear time algorithm which recognizes the class $\mathcal{B}_c$ of graphs.

Since each block in a forest is either $K_1$ or $K_2$, the class of forests is a proper subclass of $\mathcal{B}_c$. A triangulated graph is a graph which does not contain a cycle of length at least four as an induced subgraph. It is clear that $\mathcal{B}_c$ is a proper subclass of the
class of triangulated graphs.

The block-cutvertex graph of a graph $G$, denoted $bc(G)$, is defined as follows. Each block and each cut-vertex of $G$ is represented by a vertex of $bc(G)$. We call the vertices of $bc(G)$ which represent blocks its $b$-vertices, and those representing cut-vertices its $c$-vertices. The vertices $u, v$ of $bc(G)$ are adjacent if and only if $u$ is a cut-vertex contained in the block of $G$ corresponding to $v$, or vice versa. It has been shown [29] that $bc(G)$ is always a forest; it will be known as the bc-tree of $G$ when $G$ is connected.

Lemma 6.1 Let $G$ be a graph and $bc(G)$ be the block-cutvertex graph of $G$. Then

1. Every end-vertex of $bc(G)$ is a $b$-vertex.
2. Two vertices in $bc(G)$ have odd distance if and only if one is a $b$-vertex and the other is a $c$-vertex.
3. If $G$ is a block-complete graph, then for each pair of $c$-vertices having distance two in $bc(G)$, the corresponding pair of vertices in $G$ are adjacent.

*Proof:* (1) Observe that each end-vertex of $bc(G)$ corresponds to an end-block of $G$.

(2) Let $F = bc(G)$, and $u$ and $v$ be two vertices in $F$ such that $d_F(u, v)$ is odd, say $d_F(u, v) = 2n + 1$. By the definition of block-complete graphs, each $b$-vertex in $F$ is adjacent to $c$-vertices only and vice versa. Let $P: u = v_0, v_1, \ldots, v_{2n}, v_{2n+1} = v$ be a $u$-$v$ path in $F$. If $u$ is a $b$-vertex, then $v_1, v_3, \ldots, v_{2n+1}$ are $c$-vertices and $v_2, v_4, \ldots, v_{2n}$ are $b$-vertices.
This implies that \( v \) is a \( c \)-vertex. Similarly, if \( u \) is a \( c \)-vertex, then \( v \) is a \( b \)-vertex. Therefore, if two vertices in \( T \) have odd distance, then one is a \( b \)-vertex and the other is a \( c \)-vertex. By a similar proof, it can be shown that if two vertices in \( F \) have even distance, then either both vertices are \( b \)-vertices or both vertices are \( c \)-vertices.

(3) Let \( G \) be a block-complete graph, and \( u \) and \( v \) be two \( c \)-vertices in \( \text{bc}(G) \) having distance two. Then there exists a \( b \)-vertex \( w \) in \( \text{bc}(G) \) such that \( w \) is adjacent to both \( u \) and \( v \). Let \( u' \) and \( v' \) be the vertices in \( G \) corresponding to \( u \) and \( v \) in \( \text{bc}(G) \) respectively and let \( B_w \) be the block in \( G \) corresponding to \( w \) in \( \text{bc}(G) \). Then \( u' \) and \( v' \) are vertices in \( B_w \). Since \( G \) is a block-complete graph, \( u' \) and \( v' \) are adjacent in \( G \). □

\textbf{Block-Complete Augmentation}

\textbf{Instance:} Graph \( G = (V, E) \), positive integer \( B \leq |V|^2 - |V| \).

\textbf{Question:} Is there a set \( E' \) of unordered pairs of vertices from \( V \) such that \( |E'| \leq B \) and the graph \( G' = (V, E \cup E') \) is block-complete?

\textbf{Theorem 6.2} The Block-Complete Augmentation problem can be solved in linear time.

\textbf{Proof:} Let \( G \) be a graph. The blocks of \( G \) can be found in linear time [32]. For each block which is not complete, add those edges which are not present in the block to obtain a complete block. Clearly, the number of edges to be added to \( G \) is minimum so that the resulting graph is block-complete. The complete process requires only linear time. □
The technique for designing Algorithm 6.1 and Algorithm 6.2 comes from Kariv and Hakimi [36]. We know that the vertex cover number for the complete graph $K_n$ is $\alpha(K_n) = n - 1$. Algorithm 6.1 finds a minimum vertex cover $U$ in an arbitrary connected block-complete graph $G$. The algorithm is carried out through a search on the blocks of $G$, starting from the end-blocks and moving toward the "middle." During this search, we locate the vertices of the desired vertex cover set in an "optimal fashion" until all edges in the graph are covered by some minimum set of vertices.

Algorithm 6.1 finds a minimum vertex cover $U$ in a connected block-complete graph $G$.

begin

1) $G' \leftarrow G$; $U \leftarrow \emptyset$
2) while $G'$ is not complete loop
3) Find an end-block $B$ of $G'$.
4) Let $v$ be the cut-vertex of $B$ in $G'$.
5) if $(|V(B) \cap U| < |V(B)| - 1)$ then
6) Select a vertex, say $u$, in $V(B) - \{U \cup \{v\}\}$.
7) $U \leftarrow U \cup (V(B) - \{u\})$
8) end if
9) $G' \leftarrow G' - \{V(B) - \{v\}\}$
10) end loop

/* At this point, $G'$ is a complete graph */

11) if $(|U \cap V(G')| < |V(G')| - 1)$ then
12) Find a set, say W, consisting of \(|V(G')| - 1 - |U \cap V(G')|\) vertices in \(V(G') - U\).

13) \(U \leftarrow U \cup W\)

14) end if

end Algorithm 6.1

**Theorem 6.3** Algorithm 6.1 finds a minimum vertex cover \(U\) in a connected block-complete graph \(G\).

**Proof:** If \(G\) is a complete graph of order \(n\), then \(|U| = |V(G)| - 1\) by steps 11 to 13. So we assume that \(G\) is not complete. An end-block \(B\) is found with cut vertex \(v\).

If \(|V(B) \cap U| \geq |V(B)| - 1\), then all edges in the block \(B\) are already covered by vertices in the vertex cover which have been located so far, so there is no need to select vertices of \(B\) into \(U\). Otherwise, \(|V(B)| - 1 - |V(B) \cap U|\) additional vertices of \(B\) must be added to \(U\). By steps 11 to 13, the vertex that is not selected into \(U\) is not the cut-vertex \(v\) of \(B\). This implies that the number of new vertices selected into \(U\) is required and the set of vertices selected is "optimal". \(\Box\)

Recall that Algorithm 2.1 finds a minimum dominating set of radius \(r\) in a weighted oriented tree and that Algorithm 2.1 may also be used to find a minimum dominating set of radius \(r\) in a weighted tree. Next we present a linear time algorithm for finding a minimum dominating set of radius \(r\) in an (unweighted) block-complete graph. The algorithm finds a minimum dominating set of radius \(r\), \(U\), in an arbitrary connected block-complete graph \(G\). The process is carried out through a search on the
blocks of G, starting from the end-blocks and moving toward the "middle." During this search, we locate the vertices of the desired dominating set of radius r in an "optimal fashion" until all vertices in the graph are dominated by vertices in U within distance r. To do this, we use a copy G' of the original graph as an auxiliary graph on which the algorithm is carried out, and we attach two variables C(v) and R(v) to each vertex v of G'. C(v) is a boolean variable which has value TRUE if v is already dominated (within distance r) by some vertex in U has value FALSE otherwise. (The interpretation of R(v) will be given later.) If B is an end-block of the auxiliary graph G' and v_c is the cut-vertex of B in G', then we update the variables C(v_c) and R(v_c) and remove all vertices in V(B) - v_c (and incident edges) from G'. As a result, a new block may become an end-block of G', and the process is repeated until the graph G' becomes a complete graph. Then an appropriate set of vertices is added to U if necessary.

The variable R(v) has the following interpretation (based on C(v)):

Case 1: If the vertex v is already dominated by one of the vertices in U which have been located so far, then R(v) is the distance between v and the nearest located vertex in U.

Case 2: If the vertex is not yet dominated, then let S(v) be the set of all the vertices of the original graph G which are not yet dominated, and for which v is the nearest vertex in the auxiliary graph G'. Notice that v is the only vertex in S(v) which belongs to G'; in fact, S(v) is the set consisting of the vertex v
and all those vertices which have already been removed from $G'$ and are to be dominated by the same vertex of the dominating set of radius $r$ as $v$. $R(v) = \max \{d_G(u, v) \mid u \in S(v)\}$.

Algorithm 6.2 finds a minimum dominating set of radius $r$ for a connected block-complete graph $G$.

begin

/* initializations */
1) $G' \leftarrow G$; $U \leftarrow \emptyset$
   for each vertex $v$ in $G'$ loop
2) $C(v) \leftarrow$ FALSE; $R(v) \leftarrow 0$
   end loop
3) while ($G'$ is not complete) loop
4) Find an end-block $B$ in $G'$.
5) Let $v_c$ be the cut-vertex of $B$ in $G'$.
   /* For convenience, define $\min \{R(v) \mid v \in \emptyset\} = r + 1$
   and $\max \{R(v) \mid v \in \emptyset\} = 0 */
   $R_t \leftarrow \min \{R(v) \mid v \in V(B) \text{ and } C(v) = \text{TRUE}\}$
   $R_f \leftarrow \max \{R(v) \mid v \in V(B) \text{ and } C(v) = \text{FALSE}\}$
   $R' \leftarrow \min \{R(v) \mid v(\neq v_c) \in V(B) \text{ and } C(v) = \text{TRUE}\}$
   $R'' \leftarrow \max \{R(v) \mid v(\neq v_c) \in V(B) \text{ and } C(v) = \text{FALSE}\}$
   /* Update $C(v_c)$ and $R(v_c)$ */
6) if $C(v_c)$ then
   if ($R_t + R_f + 1 \leq r$) then
      $R(v_c) \leftarrow \min \{R' + 1, R(v_c)\}$
   else
      $C(v_c) \leftarrow$ FALSE; $R(v_c) \leftarrow R'' + 1$

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
end if
else
  if \((R_t + R_f + 1 \leq r)\) then
    \(C(v_c) \leftarrow \text{TRUE};\ R(v_c) \leftarrow R' + 1\)
  else
    \(R(v_c) \leftarrow \max\{R^n + 1, R(v_c)\}\)
  end if
end if

/* Select vertex into \(U\) in an optimal way */

8) if \((C(v_c) = \text{FALSE})\) and \((R(v_c) = r)\) then
   \(U \leftarrow U \cup \{v_c\};\ C(v_c) \leftarrow \text{TRUE};\ R(v_c) \leftarrow 0\)
end if

/* \(R(v_c) = r+1\) only if \(C(v_c) = \text{TRUE}\) */

10) if \((R(v_c) = r+1)\) then
    \(C(v_c) \leftarrow \text{FALSE};\ R(v_c) \leftarrow 0\)
end if

end if

G' \leftarrow G' \setminus \{v(B) \setminus \{v_c\}\}

end loop

12) \(R_t \leftarrow \min\{R(v) \mid v \in V(G')\ \text{and}\ C(v) = \text{TRUE}\}\)
    \(R_f \leftarrow \max\{R(v) \mid v \in V(G')\ \text{and}\ C(v) = \text{FALSE}\}\)
    if \((R_t + R_f \geq r)\) then
      Select a vertex \(u\) in \(G'\)
      \(U \leftarrow U \cup \{u\}\)
end if

end Algorithm 6.2
Theorem 6.4 Algorithm 6.2 finds a minimum dominating set of radius \( r \) in an (unweighted) connected block-complete graph \( G \) in linear time \( O(|E|+|V|) \).

Proof: Steps 1 and 2 initialize \( G' \), \( U \), \( C(v) \), and \( R(v) \) for \( v \in V(G') \). The interpretations for these symbols have already been given. If \( G \) is a complete graph, then steps 12 to 13 find a desired set \( U \). Suppose that \( G \) is not complete. Steps 4 and 5 find a block in \( G' \) with cut-vertex \( v_c \). It is easy to verify that the if statement from steps 6 to 7 updates the two variables \( C(v_c) \) and \( R(v_c) \) correctly. Steps 8 to 9 select vertex into \( U \) in an "optimal fashion". The if statement from steps 10 to step 11 resets \( C(v_c) \) and \( R(v_c) \) since the vertex \( v_c \) is not yet dominated. Finally, steps 12 to 13 handle the resulting complete graph.

It can be seen that the time complexity of Algorithm 6.2 is \( O(|E|+|V|) \). □

Algorithm 6.3

Given a block-complete graph \( G = (V,E) \) with \( m \)-centrix \( C = \{c_1, c_2, \ldots, c_m\} \subseteq V \), and positive integer \( r \). This algorithm finds a minimum \( m \)-centrix radius \( r \) augmentation of \( G \).

begin

Step 1 Find the set \( V_1 = \{v \in V \mid d_G(c,v) \leq r\} \), \( V_2 = V - V_1 \) and \( H = \langle V_2 \rangle_G \).

Step 2 If \( r = 1 \), \( E' = \{c_1v \mid v \in V_2\} \) is a desired set of edges.

Go to step 5.

/ * Let the components of \( H \) be \( C_1, C_2, \ldots, C_k \). The blocks of \( C_i \) are complete, \( 1 \leq i \leq k \). */

Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.
Step 3 For each component $C_i$ of $H$, use Algorithm 6.2 to find a minimum dominating set, $U_i$, of radius $r$ in $H$.

Step 4 $U \leftarrow U \cup \{U_i \mid 1 \leq i \leq k\}$

$E' = \{c_i v \mid v \in U\}$ is a desired set of edges.

Step 5 Stop.

end Algorithm 6.3

By a proof similar to Theorem 2.11, we have

Theorem 6.5 Algorithm 6.3 finds a minimum $m$-centrix radius $r$ augmentation of $G$. □
CHAPTER VII

SUMMARY

This dissertation has initiated the study of a number of graph and directed graph augmentation problems and has provided solutions for most of the augmentation problems and related problems investigated.

In Chapter II, the study of m-Centrix Radius r Augmentation problem and the Digraph m-Centrix Radius r Augmentation problem was initiated. These two problems were shown to be NP-complete for any fixed positive integers m and r such that \( r \geq 2 \). Both problems can be solved in \( O(m|V|) \) time for \( r = 1 \) and for trees (oriented trees). The related Graph and Digraph Dominating Set of Radius r problems were investigated and were proved to be NP-complete for any fixed r. A linear time algorithm was presented which can be used to find a minimum dominating set of radius r of a weighted tree or oriented tree.

Chapter III has explored the computational complexity of the following problems: r-Factorability, K Disjoint Maximum Matchings, and K Disjoint 1-Factors Augmentation. It was shown that these three problems are NP-complete. Results obtained include a linear time algorithm for deciding whether a forest contains K disjoint maximum matchings and for finding them if they exist, and an \( O(n^{5/2}) \) time algorithm which finds a
minimum $K$ disjoint 1-factors augmentation of a forest of even order where $K \leq \max\{5, 2n-1\}$.

Chapter IV has introduced the generalization of $(M, N)$-transitivity to $(M, N, R_1, R_2)$-transitivity for graphs and digraphs in order to better model redundancy constraints on networks. The study of $(M, N, R_1, R_2)$-transitivity has been extended through provision of a number of characterization results on tournaments as well as on graphs and digraphs. This chapter has provided an efficient minimum $(M, 1, R_1, =)$-transitive augmentation algorithms for graphs.

Chapter V has extended the definition of total dominating sets to $(r, s)$-dominating sets in graphs and digraphs. Various bounds on the $(r, s)$-domination number of a graph have been investigated. Computational complexity results concerning the $(r, s)$-Dominating Set and $m$-Centrix $(r, s)$-Domination Augmentation problems have been obtained in parallel to Chapter II.

Finally, Chapter VI has provided linear time algorithms for solving some graph problems for graphs whose blocks are complete.
BIBLIOGRAPHY


Reproduced with permission of the copyright owner. Further reproduction prohibited without permission.


