Combinatorics of Degree Sets and a Study of the Complexity of Degree Set Constrained Reachability Problem

A Project Report

submitted by

SAURABH SUNIL SAWLANI

in partial fulfilment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

DEPARTMENT OF ELECTRICAL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY MADRAS.

May 2013

THESIS CERTIFICATE

This is to certify that the thesis titled **Combinatorics of Degree Sets and a Study of the Complexity of Degree Set Constrained Reachability Problem**, submitted by **Saurabh Sunil Sawlani**, to the Indian Institute of Technology, Madras, for the award of the degree of **Bachelor of Technology**, is a bona fide record of the research work done by him under our supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Prof. Jayalal Sarma M. N.
Research Guide
Assistant Professor
Dept. of Computer Science and Engineering
IIT-Madras, 600 036

Place: Chennai

Date: 14th May 2013

ACKNOWLEDGEMENTS

First of all, I would like to express my gratitude to my guide, Prof. Jayalal Sarma M.N., for giving me the opportunity to work with him through this project. I would also like to thank him for his patience, enthusiasm and for the guidance and motivation he has given me during the course of the project. Secondly, I would like to thank my co-guide, Prof. Radhakrishna Ganti, for providing me any kind of support I needed. I would also like to thank Prasun Kumar, for his invaluable inputs throughout the project.

ABSTRACT

KEYWORDS: Degree sequence ; Degree set; Reachability

The degree sequence D of a graph $G(V, E)$ is defined as the sequence of degrees of all the vertices of G, arranged in non-increasing order. The *degree set* S of a graph $G(V, E)$ is just the set of degrees of the vertices of G. In our attempt to use degree-set constraints in computational complexity, we attempt to answer a slightly modified version of the Directed Graph Reachability problem through this point of view. This problem is known to be in the complexity class NL. We look at a slightly constrained version of this problem, i.e., by predetermining the degree set of the underlying undirected graph. In addition, we investigate various degree set-variants for directed graphs and also arrive at certain degree set realizability conditions for directed graphs with a given girth k.

TABLE OF CONTENTS

LIST OF FIGURES

ABBREVIATIONS

Digraph	Directed graph
DAG	Directed acyclic graph
DTM	Deterministic Turing Machine
NDTM	Non-deterministic Turing Machine
DREACH	Directed Graph Reachability Problem

NOTATION

$\mu(S)$	Minimum order for a graph realising S
$\mu_o(S)$	Minimum order for a digraph realising S as its outdegree
$\mu_\wedge(S)$	Minimum order for a digraph $wedge$-realising S
$\mu_\vee(S)$	Minimum order for a digraph vee-realising S
$\mu_A(S)$	Minimum order for an asymmetric digraph $wedge$-realising S
$\mu_k(S)$	Minimum order for a digraph of girth k $wedge$-realising S

CHAPTER 1

INTRODUCTION

1.1 The Reachability Problem

The Directed-Graph Reachability problem is stated as follows: "Given a directed graph $G(V, E)$ and two vertices $(s, t) \in V$, determine whether there is a path from s to t in G."

The Directed-Graph Reachability problem is solvable in polynomial time and is known to be in the complexity class NL. Also, the problem is known to be hard for NL. While this problem is known to be NL-complete, the undirected version of the same is known to be in L.

1.2 Degree Sets of Graphs

In the context of complexity-related questions, representation of graphs can be a hugely important aspect. While the standard methods like adjacency matrix and adjacency list representations use $O(|V|^2)$ space to represent a graph $G(V, E)$, degree sets and degree sequences can be used to represent graphs to an extent using just $O(|V|log|V|)$ space.

Realizability of degree sets has also been an important line of research in the past years. In Kapoor *et al.* (1977), it is shown that any set $S = a_1 < a_2 < \ldots < a_n$ of positive integers can always be realized by a simple graph with $a_n + 1$ vertices. They also gave minimum realizability conditions for trees, planar graphs etc. Similarly, Chartrand *et al.* (1976) showed such realizability conditions for directed graphs, given their outdegree set.

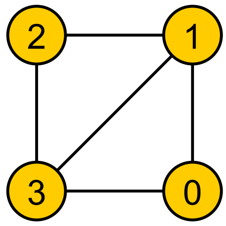

Figure 1.1: An undirected graph of 4 vertices with degree set $\{2, 3\}$. Note that 4 is the minimum number of vertices required to realise this degree set.

1.3 Motivation

The Directed Graph Reachability problem is a well-studied problem in terms of its space complexity (Reachability is NL-complete). Hence, we try to constrain the problem, by specifying the graph's degree set beforehand. It is interesting to see if this constrained version remains as hard as the original, or if it falls down to deterministic log-space.

The results regarding asymmetric (Girth=3) digraphs in Kumar *et al.* (2013), gave rise to the question if the result could be generalized to any girth. We develop a similar proof for this general case, which also gives us an idea about degree set realization for DAGs.

1.4 Overview

In the chapter 2, we provide the basic concepts in Graph Theory and Complexity Theory used in our results, and familiarize the reader with the terminology used throughout.

Chapter 3 gives a broad view of the literature related to our work. We state and also reproduce some of the work done in the field previously, and which is needed as a background to grasp the essence of our work here.

Chapters 4 and 5 contain our contribution in the thesis. Chapter 6 outlines the open problems and future research which could help expand and better the results in this thesis.

CHAPTER 2

PRELIMINARIES

2.1 Graph Theoretic Preliminaries

Undirected Graphs

An **undirected graph** is an ordered pair $G(V, E)$, where V is a finite non-empty set of elements (known as *vertices*), and E is a finite set of unordered pairs of vertices (known as *edges*). For some $e = (u, v) \in E$, u and v are known as the *end-vertices* of e and the edge e is said to be incident with the vertices u and v. Also, u and v are called adjacent vertices. Two edges, (u_1, v_1) and (u_2, v_2), are called *parallel edges* if $u_1 = v_1$, and $u_2 = v_2$. When $u = v$, the edge (u, v) is called a *self loop*.

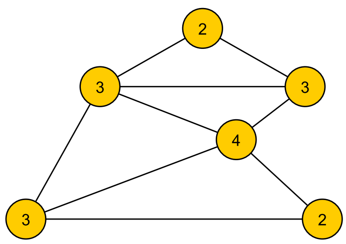

Figure 2.1: An example undirected graph of 6 vertices.

In an undirected graph, the *degree* of a vertex v (deg(v))is defined as the number of edges incident with it. A vertex with degree 0 is called an *isolated* vertex. For a graph $G(V, E)$, $\sum_{v \in V} deg(v) = 2|E|$. In the graph in Figure 2.1,

- $deg(v_0) = 3; deg(v_1) = 6; deg(v_2) = 2; deg(v_3) = 4; deg(v_4) = 2; deg(v_5) = 3$

Directed Graphs

A **directed graph** is an ordered pair $G(V, A)$, where V is a finite non-empty set of elements (known as *vertices*), and A is a finite set of *ordered* pairs of vertices (known as *arcs* or directed edges). For some $e = (u, v) \in A$, u and v are known as the *start-vertex* and *end-vertex* of e respectively. Also, v is said to be a direct successor of u, and u is said to be a direct predecessor of v. Two arcs, (u_1, v_1) and (u_2, v_2), are called *parallel edges* if $u_1 = v_1$, and $u_2 = v_2$. When $u = v$, the arc (u, v) is called a *self loop*.

In a directed graph $G(V, A)$, the *outdegree* of a vertex $u \in V$, denoted by $d^+(u)$, is defined as $d^+(u) = |(u, v) \in A|v \in V|$ and the *indegree* of a vertex $u \in V$, denoted by $d(u)$, is defined as $d(u) = |(v, u) \in A|v \in V|$. A vertex with positive outdegree and indegree 0 is called a *source* vertex. A vertex with positive indegree and outdegree 0 is called a *sink* vertex. For a graph $G(V, A)$, $\sum_{v \in V} d^+(v) = \sum_{v \in V} d^-(v) = |E|$.

Simple Graphs

A graph G(V,E) is said to be simple if it does not contain self loops and parallel edges. It may be directed or undirected. Henceforth, whenever we use the term graph, we implicitly mean simple undirected graph.

In Figure 2.2, only the graphs on the right are simple graphs.

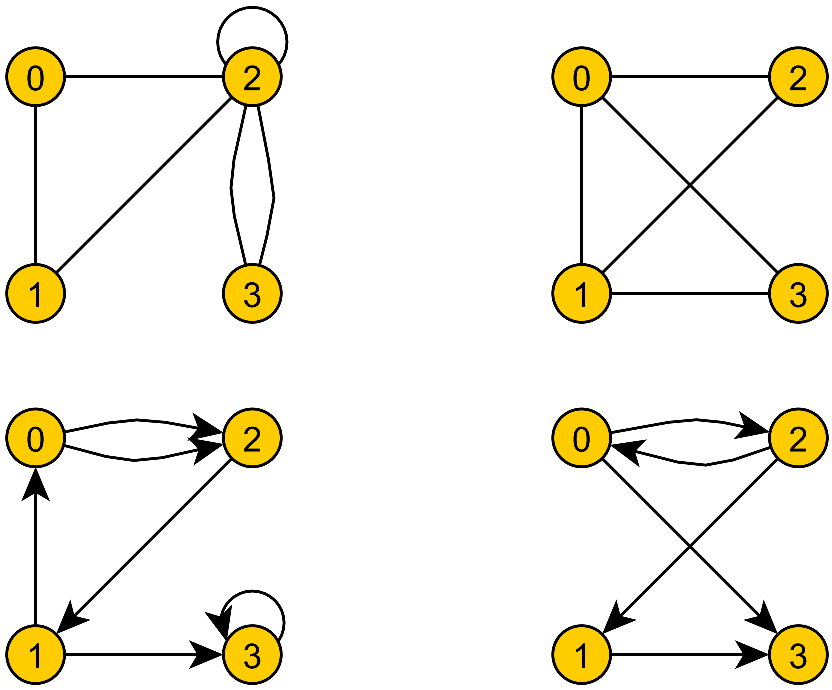

Figure 2.2: An example to illustrate simplicity of graphs.

Degree Sequences and Degree Sets

The *degree sequence* of an undirected graph $G(V, E)$, $D(G)$ is defined as the sequence of degrees of all the vertices of G arranged in non-increasing order.

The *degree set* of an undirected graph $G(V, E)$, $S(G)$ is defined as the set of degrees of all the vertices of G.

For the graph in Figure 2.1, the degree sequence is $(6, 4, 3, 3, 2, 2)$ and the degree set is $\{2, 3, 4, 6\}$

Although generally undefined, we will define notions of degree sequences and sets for directed graphs as required in future sections.

Subclasses of graphs

There are many subclasses of both directed and undirected graphs have applications in various fields. Some of the subclasses are:

- *Tree*: A connected undirected graph with no cycles is called a tree.
- *Planar Graph*: A planar graph is a graph which can be embedded in a plane.
- *Directed Acyclic Graph*: A directed acyclic graph (DAG) is a digraph without directed cycles.
- *Directed Tree* : A digraph whose underlying undirected graph is a tree.

Other Graph Properties

The *order* of a graph is the number of vertices in the graph. The *size* of a graph is the number of edges in the graph.

The *girth* of a graph is the length of the shortest cycle contained in the graph. For directed graphs, it is the length of the shortest directed cycle.

The *complement* $\overline{G}(V', E')$ of a graph $G(V, E)$ is that graph for which $V' = V$ and for any $u, v \in V, (u, v) \in E'$ if and only if $(u, v) \notin E$.

The *union* $G \cup H$ of two disjoint graphs $G(V, E)$ and $H(V', E')$ is that graph whose vertex set is $V \cup V'$ and whose edge set is $E \cup E'$.

The *join* $G + H$ of two disjoint graphs $G(V, E)$ and $H(V', E')$ is that graph whose vertex set is $V \cup V'$ and whose edge set is $E \cup E' \cup X$, where $X = \{(u, v)|u \in V and v \in V'\}$.

2.2 Complexity Theoretic Preliminaries

The *complexity* of a given problem is a measure of how hard it is to solve the problem, with respect to the amount of resources required to solve it. The resources usually used as complexity measures are *time* and *space*. A complexity class, however, is defined as a collection of problems that, using some model of computation, have the same resource based constraints . We will concentrate on space as a resource, as it is relevant to our work. Also, we will use a *single-tape Turing Machine* as our primary model of computation.

Space complexity

The space complexity of a problem is a measure of the amount of space, or memory required by an algorithm to solve it.

Space Complexity Classes

- DSPACE$(f(n))$: Set of languages that are decidable by a DTM using $O(f(n))$ tape cells.
- NSPACE$(f(n))$: Set of languages that are decidable by a NDTM using $O(f(n))$ tape cells.
- PSPACE$= \bigcup_{c\geq 0}$DSPACE(n^c)
- NPSPACE$= \bigcup_{c\geq 0}$NSPACE(n^c)
- L$=$DSPACE$(log(n))$
- NL$=$NSPACE$(log(n))$

Class L is a set of languages that are decidable by a DTM using logarithmic number of tape cells with respect to input length. Similarly, class NL is a set of languages that are decidable by a NDTM using logarithmic number of tape cells with respect to input length.

CHAPTER 3

Literature Review

3.1 Space complexity of the Directed Reachability Problem (DREACH)

In our bid to see if degree set bounds really help in bettering algorithms, we first look at some of the established results regarding Reachability in graphs. Directed Reachability (DREACH) DREACH is a well-studied problem, and is known to be in the complexity class NL, i.e., it takes logarithmic space for a NDTM to solve DREACH. We will see the result proved here:

3.1.1 DREACH $\in$ NL

This part is fairly simple. To see that it is in NL, we need to show a non-deterministic algorithm using log-space that never accepts if there is no path from s to t, and that sometimes accepts if there is a path from s to t. The following simple algorithm achieves this:

```
input <G,s,t>
if s = t ACCEPT
set v := s
for i = 1 to n:
guess a vertex vnext
if there is no edge from v to vnext, REJECT
if vnext = t, ACCEPT
v := vnext
if i = n and no decision has yet been made, REJECT
```

The above algorithm needs to store i(using $logn$ bits), and at most the labels of two vertices v and $vnext$ (using $O(logn)$ bits).

Interestingly, the problem DREACH is also known to be *hard* for NP, i.e., if DREACH can be solved by a DTM in logarithmic space, then NL will collapse to L. The following is a proof for this:

3.1.2 DREACH is hard for NL

For this, we will choose an arbitrary language A $\in$ NL and reduce it in logspace to DREACH.

- Let A be a language in NL
- Let M be a non-deterministic Turing Machine that decides A with space complexity $logn$
- Choose an encoding for the computation $M(x)$ that uses $klog(|x|)$ symbols for each configuration.
- Let C_0 be the initial configuration, and C_a be the accepting configuration.
- We represent $M(x)$ by giving first the list of vertices, and then a list of edges by doing the following:
  - We go through all possible strings of length $klog(|x|)$ and, if the string properly encodes a configuration of M, prints it on the output tape.
  - We go through all possible pairs of strings of length $klog(|x|)$. For each pair (C_i, C_j), it checks if both strings are legal encodings of configurations of M, and if C_i can yield C_j. If yes then it prints out the pair on the output tape.
  - Both these require only log-space, as the strings are lexicographically ordered and only the current string needs to be stored.
- M accepts x if and only if there is a C_0 is reachable from C_m in the graph $M(x)$.

Although the undirected Reachability problem is now known to be in L, there is a much simpler algorithm to compute the Reachability problem for *acyclic graphs*. We will give a gist of the algorithm here:

- The input is an acyclic graph F and two vertices s and t.
- We first form a cyclic order beginning from s by looking at the adjacency list.
- We traverse this order, and store the edge through which we first left s. Name this edge e.
- If t is encountered at any point, stop and accept.
- If we reach s through some edge e', and e follows e' in the cyclic order, then stop and reject.

- It takes more than logarithmic space to assign a cyclic order, but if we assign the order for a vertex' neighbours when we are at the vertex itself, then at any instant we do not need more than logarithmic space. Also, remembering e requires log-space.

In figure 3.1, we can see the path taken by the algorithm to decide that t is not reachable from s.

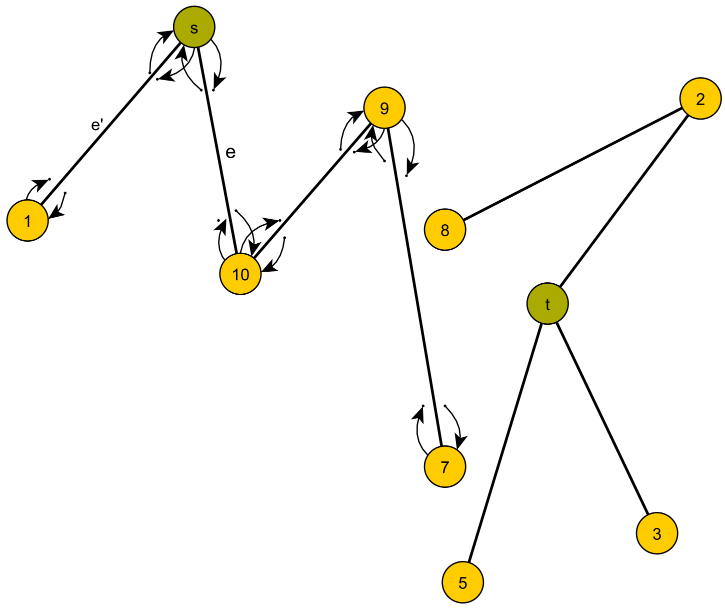

Figure 3.1: The log-space algorithm on a forest F

With this background of the reachability problem, we attempt to look at the problem within degree set constraints. Our attempts in this regard are seen in Chapter 4.

3.2 Degree Set Characterizations of Undirected and Directed Graphs

Given a set $S = \{a_1 < a_2 < \ldots < a_n\}$, it is interesting to see if S can be realized by a graph. If yes, how many vertices does it need to realize it? To answer this question, Kapoor *et al.* (1977) give some interesting results, where it is shown that there always exists a simple graph $G(V, E)$ with $|V| = a_n + 1$ vertices realizing the set $S = a_1 < a_2 < \ldots < a_n$ of positive integers. This implies that $\mu(S) = a_n + 1$. We reproduce the proof as below:

Theorem 1 *Any set $S = \{a_1 < a_2 < \ldots < a_n\}$ of positive integers is realized by a simple graph and $\mu(S) = a_n + 1$.*

Proof

Proof is by induction on $|S|$. For $|S| = 1$, $S = \{a1\}$ and the corresponding graph is K_{a_1+1}, the complete graph on $a_1 + 1$ vertices. For $|S| = 2$, $S = \{a1\}$ and the corresponding graph is $K_{a_1+1} + (\overline{K}_{a_2-a_1+1}$ (Note that it has exactly $a_2 + 1$ vertices).

Now assume that, $\forall m \leq n$ for every set S with m number of elements, $\mu(S) = a_m + 1$, where a_m is the largest element of S. Let $S' = \{b_1 < b_2 < \ldots < b_{n+1}\}$ By the induction hypothesis, $\mu(\{b_2 - b_1 < b_3 - b_1 < \ldots < b_n - b_1\}) = b_n - b_1 + 1$. Let H be some graph which satisfies this degree set with $b_n - b_1 + 1$ vertices. Thus, the graph $K_{b_1} + (\overline{K}_{b_{n+1}-b_n} \cup H)$ has the degree set S' and order b_{n+1}. Thus, using the principles of mathematical induction, $\mu(a_1 < a_2 < \ldots < a_n) = a_n + 1$. ■

This result was extended to asymmetric directed graphs in Chartrand *et al.* (1976), wherein they only restricted the outdegree set of the digraph.

Lemma 2 *For $a \geq 0, \mu_o(a) = 2a + 1$*

A graph for $a = 3$ with $2(3) + 1 = 7$ vertices can be seen in Figure 3.2.

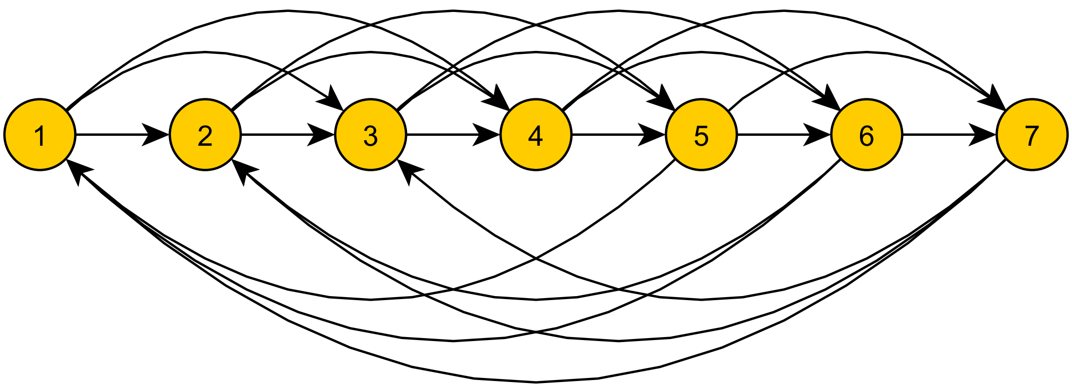

Figure 3.2: An example for a minimum order graph for a singleton degree set.

Theorem 3 *Let $\{0 \leq a_1 < a_2 < \ldots < a_n\}, (n \geq 2)$ be a set of non-negative integers, and let t be the least integer exceeding 1 for which $(n + t - 2)a_1 + \binom{t}{2} \geq \sum_{i=2}^{n} a_i$. Then,*

$$\mu_o(a_1, a_2, \ldots, a_n) = \begin{cases} a_n + 1 & \textit{if}\, a_n \geq \mu_0(a_1, a_2, \ldots, a_{n-1}) \\ 2a_1 + t & \textit{if}\, a_n < \mu_0(a_1, a_2, \ldots, a_{n-1}) \end{cases}$$

Kapoor *et al.* (1977) also extended their result for minimum order for undirected graphs to trees and planar graphs. For trees, they showed the following result:

Theorem 4 *Let* $S = \{a_1 < a_2 < \ldots < a_n\}, n \geq 1$ *be a set of positive integers. There exists a tree* T *with degree set as* S *if and only if* $a_1 = 1$*. Moreover, the minimum order of a tree realizing* S *(*$\mu_T(S)$*) is* $\sum_{i=1}^{n} a_i - 1 + 2$

The minimum order tree construction in the proof of Theorem 4 uses only 1 vertex each for each of a_2 to a_n and a large number of pendant (degree 1) vertices. So naturally, the question arose if one could reduce the multiplicity of pendant vertices in a tree, compromising on the minimum order. However, Kumar *et al.* (2013) answered this in the negative with the following proof:

Theorem 5 *The minimum multiplicity of* pendant vertices *in any tree realization for the degree set* $S = \{1 = a_1 < a_2 < \ldots < a_n\}$ *is* $\sum_{i=1}^{n} a_i - 2n + 3$.

Proof The set $S = \{1 = a_1 < a_2 < \ldots < a_n\}$ can be realized by a tree due to Kapoor *et al.* (1977). The minimum order of such a tree is $\sum_{i=1}^{n}(a_i - 1) + 2$. In the minimum order tree construction in Kapoor *et al.* (1977), each a_i is connected with exactly $a_i - 2$ pendant vertices for $i = 3, 4, \ldots, n-1$ and for $i = 2$ *and* n, a_i's are connected with $a_i - 1$ pendant vertices and then a_i is connected with a_{i+1} for $i = 2, \ldots, n-1$.

Let m_i be the multiplicity of a_i in a tree realization T. Then, $(a_n^{m_n}, a_{n-1}^{m_{n-1}}, \ldots, 1^{m_1})$ will be the degree sequence of T.

Case 1 when $a_2 \geq 3$. If degree sequence $D = (d_1 \geq d_2 \geq \ldots \geq d_n)$ is being realized by a tree then number of pendant vertices in any tree realization Arikati and Maheshwari (1996) of D is $\sum_{i=1}^{k}(d_i - 2) + 2$ where k is the largest index such that $d_k \geq 3$. Hence, m_1 = $2 + (a_2 - 2)m_2 + (a_3 - 2)m_3 + \ldots + (a_n - 2)m_n$, $\forall i\ m_i \geq 1$. m_1 will be minimum if $m_i = 1$ for each $i = 2, 3, \ldots, n$ and the tree construction described above satisfy the required conditions. Hence, the minimum value $m_1 = 2 + (a_2 - 2) + (a_3 - 2) + \ldots + (a_n - 2) = \sum_{i=1}^{n} a_i - 2(n-1) + 1$ $= \sum_{i=1}^{n} a_i - 2n + 3$

Case 2 : when $a_2 = 2$. We first construct the tree for the degree set S_1 = $\{1 = a_1 < a_3 < \ldots < a_n\}$ in the way mentioned above and then introduce a vertex v. Now make v adjacent to any one pendant vertex,say u,so that v becomes the new pendant vertex and $d(u) = 2$. Degree set of this modified tree is S and number of pendant vertices is same as that in the tree realization of D_1 which is same as $m_1 = 2+(a_3-2)+(a_4-2)+\ldots+(a_n-2 = 2+(a_2-2)+(a_3-2)+\ldots+(a_n-2)$ $= \sum_{i=1}^{n} a_i - 2(n-1) + 1 = \sum_{i=1}^{n} a_i - 2n + 3$

■

However, a similar characterization for planar graphs was not as straightforward. The following result gives a necessary and sufficient condition for a degree set to be realized by a planar graph.

Theorem 6 *Let $S = \{a_1 < a_2 < \ldots < a_n\}, n \geq 1$, be a set of positive integers. Then there exists a planar graph G with degree set S if and only if* $1 \leq a_1 \leq 5$.

With respect to the minimum number of vertices required for a planar graph to realize an integer set (μ_p, they only showed results for sets of maximum size 2:

Theorem 7 *Let a_1 and a_2 be positive integers with $a_1 < a_2$. Then,*

1. $\mu_p(a_1, a_2) = \begin{cases} a_2 + 1 & for 1 \leq a_1 \leq 3 \\ a_2 + 2 & for a_1 = 4 \end{cases}$
2. $\mu_p(a_1, a_2) \leq 2a_2 + 2 for a_1 = 5$

Apart from the notions of outdegree and indegree sets, several other variants of degree sets exist for digraphs:

- A digraph is said to $\wedge$**-realize** an integer set if all the elements of the set appear both as indegree and outdegree at least once, and the indegree and outdegree of all the vertices belong to the set.
- A digraph is said to $\vee$**-realize** an integer set if all the elements of the set appear either as indegree and outdegree at least once, and either the indegree or outdegree of every vertex belongs to the set.
- The **underlying degree set** of a digraph is the degree set of the underlying undirected graph for any given digraph.

Kumar *et al.* (2013) give $\wedge$-realizability constraints for asymmetric (Girth = 3) digraphs. We will state the result, but not prove it, as the proof is generalized for a girth k digraph in Chapter 5.

Theorem 8 *If $S = \{a_1 < a_2 < \ldots < a_n\}, n \geq 2$ is a set of positive integers then*

$$a_1 + a_n + 1 \leq \mu_\wedge(S) \leq a_{n-1} + a_n + 1$$

After directed graphs, the natural direction was to move on to directed trees. Surprisingly, minimum order realizability takes a much simpler turn in the domain of directed tree, be it $\wedge$-realizability or vee-realizability, as shown in Kumar *et al.* (2013):

Theorem 9 *For the degree set $S = \{1 = a_1 < a_2 \ldots < a_n\}$, minimum order of a directed tree $T(V, E)$ vee-realizing S, is same as the minimum order undirected tree realizing S,i.e. $\sum_{i=1}^{n} a_i - 1 + 2$.*

Proof To show that it is upper bounded by $\sum_{i=1}^{n} a_i - 1 + 2$, we construct a graph with those many vertices. Consider an undirected tree T_u realizing S. We know that every tree can be expressed as a bipartite graph. Consider the bipartite version of T_u and add directions to all edges from left to right. It can be seen that this digraph vee-realizes S.

For each i, $a_i \in S$ will appear as both (a_i, a_j) and (a_k, a_i) at least once, where $a_j, a_k \in S$. Thus, $1 \leq a_i + a_j \leq 2a_n$. Let $T(V, E)$ be a directed tree for S satisfying the constraints. We have,

$$\sum_{v \in V} (d^-(v) + d^+(v)) = 2|E| = 2(V - 1) \geq \sum_{i=1}^{n} a_i + (V - n)$$

This implies the lower bound $|V| \geq \sum_{i=1}^{n} a_i - 1 + 2$.

■

In case of $\wedge$-realizability, a necessary condition is that $0 \in S$, since the tree's leaves will have either indegree or outdegree as 0.

Theorem 10 *For the degree set $S = \{0 < 1 < a_3 < \ldots < a_n\}$, the minimum order of a directed tree T which $\wedge$-realizes the degree set S, is $2\sum_{i=1}^{n} a_i - 1 + 2$.*

Proof We prove the upper bound by constructing the directed tree. Construct a path with $2(n-1)$ number of vertices, say $u_1, u_2, \ldots, u_{2n-2}$. Now add $(a_2 - 1)$ pendant vertices to u_1. For each $2 \leq i \leq 2n-1$, add $a_{\lceil \frac{i}{2} \rceil + 1} - 2$ pendant vertices to u_i. Add $a_n - 1$ pendant vertices to the u_{2n-2}.

In this tree, the first 2 vertices have degree a_2, the next 2 vertices have degree a_3 and so on. Now, we assign directions. Start with the first vertex u_1 in the path. Direct all edges connected with u_1 towards u_1. For the next vertex in the path u_2 assign directions to all adjacent edges away from u_2. Repeat this process to assign direction to all edges. Since each a_i, for $i = 2, 3, \ldots, n$, appears exactly twice and because of the way we are assigning directions to edges, a_i once appears as $(a_i, 0)$ and once as $(0, a_i)$ in final directed tree. For all the pendant vertices, indegree and outdegree pair occurs as either $(1, 0)$ or $(0, 1)$. This can be seen in Figure 3.3

To prove the minimality, we consider that each degree needs to appear twice(both as indegree and outdegree). However we group the indegrees and outdegrees, we cannot decrease the number of pendant vertices, and thus the underlying undirected tree will have at least $2\sum_{i=1}^{n} a_i - 1 + 2$ vertices. ■

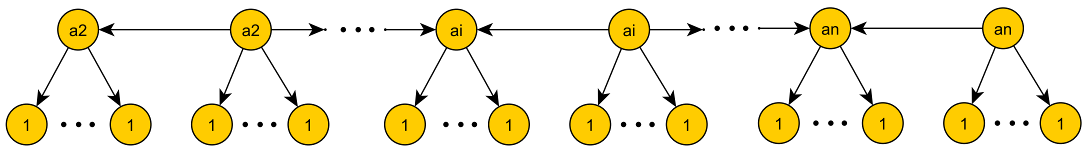

Figure 3.3: A directed tree $\wedge$-realizing S.

CHAPTER 4

Effect of Degree Set Constraints on Reachability

We have already discussed the space complexity of the Directed Graph Reachability problem, and have established its hardness for the class NL. We will now attempt to constrain this problem using the Degree set handle, and check if we can either prove this constrained version to be in L, or prove that it is as hard as the original problem.

We shall denote the degree set of the underlying graph of a directed graph G as $S_u(G)$.

First, we consider a simplified version of this problem, by restricting the degree set to be a singleton. Clearly, the problem is solvable in constant time (and in constant space) if the degree set is $\{1\}$. However, for degree set $\{2\}$, we have the following result.

Lemma 11 *The Directed Graph Reachability problem on a graph $G(V, A)$ can be solved in deterministic log-space if $S_u(G) = \{2\}$.*

Proof The following log-space algorithm achieves this:

```
if s = t ACCEPT
set v := s
while outdegree of v is not 0
choose a vertex which has an edge from v as vnext
if vnext = t, ACCEPT
delete the edge (v,vnext)
if outdegree of vnext is 0,
v := s
else
v := vnext
REJECT
```

Correctness:

1. If the outdegree of s is 0, t is obviously not reachable from s, and the algorithm rejects immediately without entering the loop.
2. If the outdegree of s is 1 and t is not reachable from s, the algorithm reaches a sink (may be even s as the algorithm deletes edges already traversed) and hence exits the loop and rejects.
3. If the outdegree of s is 1 and t is reachable from s, then the algorithm follows the only possible path and accepts on reaching t.
4. If the outdegree of s is 2 and t is not reachable from s, then the algorithm reaches a sink, starts again from s and once more reaches a sink (could be the same one). It then exits the loop and rejects.
5. If the outdegree of s is 2 and t is reachable from s, then the algorithm goes in one direction and accepts if it reaches t. If it doesn't, it restarts from s in the opposite direction and reaches t and accepts.

Space complexity: The above algorithm needs to store at most the labels of two vertices v and $vnext$ (using $O(logn)$ bits).

■ It is simple to see that this result also holds for any graph with underlying degree set $\{1, 2\}$. Naturally, we try to extend the result. Interestingly, the problem stops being in log-space for other choices of the degree set. Allowing the singleton set to contain *any positive odd integer greater than* 1, we show the following result.

Lemma 12 *Every digraph $G(V, A)$ has an equivalent digraph $G'(V', A')$ with $S_u(G') = \{k\}$, (k is any odd integer greater than 1), such that for every $s, t \in V$, $\exists\, s', t' \in V'$ such that t is reachable from s in G if and only if t' is reachable from s' in G'.*

Proof Consider a given graph $G(V, E)$. For every vertex $v \in V$, we will construct a connected subgraph H_v in G' with only vertices of degree k, such that H_v is functionally similar to v.

Let $S_u(G) = \{a_1, a_2, \ldots, a_n\}$. Then, the construction of H_v for some $v \in V$ with degree a_i is as follows:

Case 1 $a_i < k$, a_i is even.
Construct K_{k+1}. Now, we obtain the subgraph H_v by deleting any $a_i/2$ edges from it. Fig. 4.1 illustrates this for $a_i = 2$ and $k = 3$.

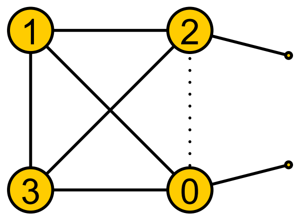

Figure 4.1: H_v for $a_i = 2$ and $k = 3$

Case 2 $a_i < k$, a_i is odd.
Construct K_{k+1}. Now, delete $(k - a_i)/2$ disconnected edges from the graph. (Can be shown that this can always be done) Then, add another vertex v' and connect it to the vertices from which the edges were deleted. This gives H_v. Fig. 2 illustrates this for $a_i = 1$ and $k = 3$.

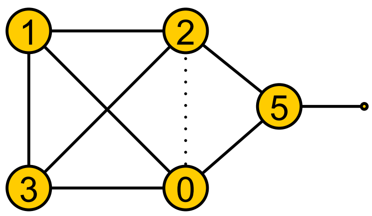

Figure 4.2: H_v for $a_i = 1$ and $k = 3$

Case 3 $a_i > k$.
First, we find integers y and z which satisfy the equation,

$$a_i = z(k - 2) + y, 0 \leq y < k - 2$$

If we get $z = 1$, we construct a vertex of degree k connected to a vertex of degree $y + 1$. If we get $z = 2$, we construct two vertices of degree k, both connected to a vertex of degree $y + 2$. For any other value of z, we construct a cyclic graph of $z + 1$ vertices, such that z of the vertices have k as its degree and one vertex has degree $y + 2$. This can be seen clearly in Fig. 4.3.

This vertex (z+1) is then replaced with a subgraph of degree-k vertices by using the methods in Cases 1 and 2.

■

This essentially means that restricting the degree set to even a singleton such as $\{3\}$ does not make the problem any easier.

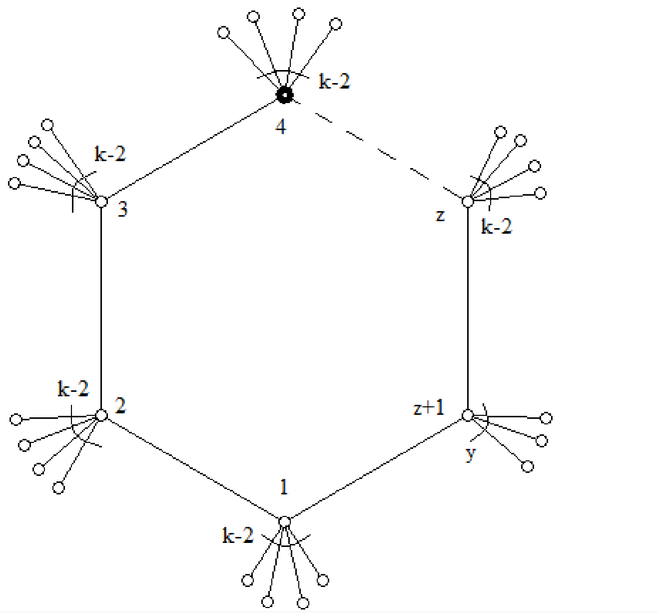

Figure 4.3: H_v for $a_i > k$

We now generalize this to a much broader case of degree sets.

Theorem 13 *Every graph $G(V, E)$ has an equivalent graph $G'(V', E')$ where $S_u(G')$ contains at least one odd integer (greater than 1) and for every $s, t \in V, \exists s', t' \in V'$ such that t is reachable from s if and only if t' is reachable from s'.*

Proof For this construction, we only need a slight tweak of the previous case. Take any odd integer greater than 1, say $k \in S_u(G')$. Now, construct an equivalent graph H for G, where $S_u(H) = \{k\}$. Then, construct a minimum-order graph H' on $S_u(G')$ (as in Kapoor *et al.* (1977)). We give G' as $G' = H \bigcup H'$. ■

So, essentially, we can produce a reduction from the original DREACH to the Degree-set constrained version. Since the degree set is given beforehand, we can convert each vertex into its corresponding subgraph in log-space (as the subgraphs will be ready during preprocessing).

This method does not work for degree sets with only even numbers as elements, and hence it still remains to be seen if restricting the problem with an all-even degree set makes it solvable in deterministic log-space.

CHAPTER 5

Degree set realization for directed graphs with specified girth

We have seen the results in Kapoor *et al.* (1977) and Chartrand *et al.* (1976) for minimum-order realizability conditions in undirected graphs, trees, planar graphs and even asymmetric directed graphs (realising outdegree set). If we observe closely, the minimum order conditions for general directed graphs to $\wedge$-realise degree sets (without asymmetric conditions, i.e., having Girth $= 3$) are exactly same as the conditions for undirected graphs. All we need to do is replace every undirected edge with a pair of arcs going opposite ways, to achieve the minimum possible order.

Kumar *et al.* (2013) then gave us conditions to be able to $wedge$-realise degree sets by asymmetric (Girth $= 3$)digraphs.

From Kumar *et al.* (2013), we know the following theorem:

Theorem 14 *If* $S = \{a_1 < a_2 < \ldots < a_n\}, n \geq 2$ *is a set of positive integers then*

$$a_1 + a_n + 1 \leq \mu_A(S) \leq a_{n-1} + a_n + 1$$

It was only natural, then, to try and generalise this to Girth k digraphs. We will use a similar construction to generalise this result for girth $= k$ digraphs. Let $\mu_k(S)$ be the minimum order of a girth k digraph $\wedge$-realising S.

Lemma 15 *If* $S = \{a\}$ *where* a *is a non-negative integer, then* $\mu_k(S) = (k-1)a + 1$.

Proof This case is similar to the one in Chartrand *et al.* (1976). When $a = 0$ the graph is an isolated vertex, the result is obvious. For $a \geq 1$, all vertices in a directed graph with degree set $\{a\}$ must have both indegree and outdegree equal to a. Consider a vertex v, since the graph is asymmetric, v is connected to $2a$ distinct vertices. Any vertex out of the a vertices which receive an edge from v connects to a vertices distinct from any already existing vertex. This continues for $k-3$ steps before one connect a vertex to an already existing one. Accounting for these vertices and v, we have $(k-1)a+1$ vertices. Hence, $\mu_k(S) \geq (k-1)a+1$. To complete the proof, we need to prove that $\mu_k(S) \leq (k-1)a+1$. To do this, we will come up with a construction of a directed graph with degree set $\{a\}$ and order $(k-1)a+1$.

We define G to be the directed graph with the vertex set $\{v_1, v_2, \ldots, v_{(k-1)a+1}\}$. The edges are as follows: $\{(v_i, v_j) | 1 \leq i \leq (k-1)a+1$ and $i+1 \leq j \leq i+a\}$ (where subscripts are modulo $(k-1)a+1$). Clearly, G has girth k and has $(k-1)a+1$ vertices with degree set $\{a\}$. Hence the proof. ■ The Figure 5.1 shows this construction for

$k = 4$ and $a = 2$

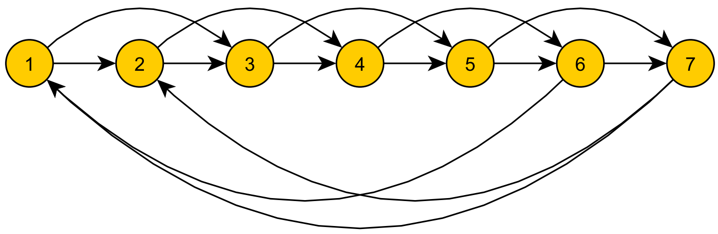

Figure 5.1: An example for a minimum order graph of given girth $k = 4$ for a singleton degree set $\{2\}$.

We now build on this base case, with a construction similar to the one used in Kumar *et al.* (2013).

Theorem 16 *If* $S = \{a_1 < a_2 < \ldots < a_n\}, n \geq 2$ *is a set of positive integers then*

$$(k-2)a_1 + a_n + 1 \leq \mu_k(S) \leq (k-3)a_1 + a_{n-1} + a_n + 1.$$

Proof We know that there is at least one vertex v of G with either indegree or outdegree equal to a_n. Without loss of generality, let us assume that $d^+(v) = a_n$. Now, we know that $d^-(v) \geq a_1$. Therefore, $d^+(v) + d^-(v) \geq a_n + a_1$. Since G has girth k, any vertex v' which has an edge from v cannot have an edge from itself to any of the vertices already present. This process has to repeat itself $k-3$ number of times for a vertex to be able to connect back to an already existing vertex. Thus, including the original vertex v, the minimum number of vertices in the graph equals $(k-2)a_1 + a_n + 1$.

To prove that $\mu_A(S) \leq (k-3)a_1 + a_{n-1} + a_n + 1$, we proceed by induction. By Lemma 15, we know that $\mu_k(\{a_1\}) = (k-1)a_1 + 1$. Let the graph representing this be G_1. Divide G_1 into k components, C_1 to C_{k-1} - each containing a_1 vertices, and C_k - containing the remaining vertex. From G_1, we obtain G_2, by adding a new component A_1 containing $a_2 - a_1$ vertices and adding the following edge set $E = \{(u,v)|u \in C_1, v \in A_1\} \cup \{(u,v)|u \in A_1, v \in C_2\}$. Thus, we have a girth-$k$ directed graph for the degree set $\{a_1 < a_2\}$ with order $(k-2)a_1 + a_2 + 1$.

Now consider that there exists a girth-k directed graph G_m with degree set $\{a_1 < a_2 < \ldots < a_m\}$, with order $(k-3)a_1 + a_{m-1} + a_m + 1$. G_m contains a total of $2m+k-3$ components :

- A_{m-1}, containing $a_m - a_{m-1}$ vertices with outdegree and indegree equal to a_1.
- A_i, for i from 1 to $m-2$, each containing $a_{i+1} - a_i$ vertices with outdegree a_1 and indegree a_{m-1-i}.
- B_j, for j from 1 to $m-2$, each containing $a_{j+1} - a_j$ vertices with outdegree a_{m-1-j} and indegree a_1.
- C_1, containing a_1 vertices with outdegree a_m and indegree a_{m-1}.
- C_2, containing a_1 vertices with outdegree a_1 and indegree a_m.
- C_{k-1}, containing a_1 vertices with outdegree a_{m-1} and indegree a_1
- C_k, containing 1 vertex with outdegree and indegree a_1.

From G_m, we obtain G_{m+1}, by adding two new components - A_m containing $a_{m+1} - a_m$ vertices, and B_{m-1} containing $a_m - a_{m-1}$ vertices, and adding the edge set $E = E_1 \cup E_2 \cup E_3$, where

- $E_1 = \{(u,v)|u \in C_1, v \in A_m\} \cup \{(u,v)|u \in A_m, v \in C_2\}$
- $E_2 = \{(u,v)|u \in C_{n-1}, v \in B_{m-1}\} \cup \{(u,v)|u \in B_{m-1}, v \in C_1\}$
- $E_3 = \{(u,v)|u \in B_{m-1-i}, v \in A_i\}$, where $i \in \{1, 2, \ldots, m-2\}$

Figure 5.2 illustrates the inductive step. In the figure,

- every node represents a component of the graph.
- an edge from a component to another represents an edge from every vertex in the first to every vertex in the second.
- a dashed-line edge from a component to another represents that there exist some edges from the first to the second.

- a darker edge just indicates that it is added during the present iterative step.
- both C_1s are the same. It has been repeated only to make the graph look cleaner.

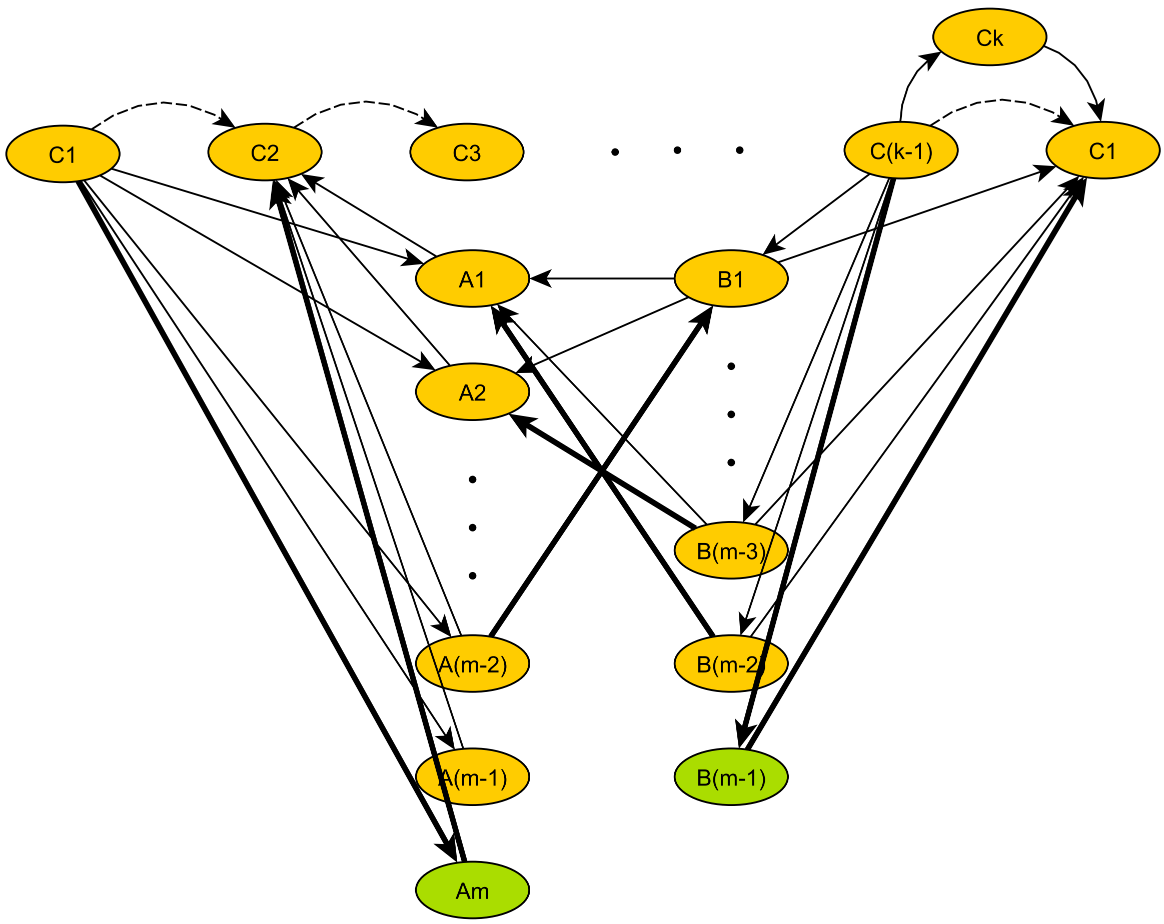

Figure 5.2: Construction of G_{m+1} from G_m

We can observe that G_{m+1} resembles G_m if m is replaced with $m+1$. Thus, through this construction, we have proved that there always exists a girth-k directed graph G with degree set $(a_1 < a_2 < \ldots < a_n)$, of order $a_{n-1} + a_n + 1$. Hence, the minimum order $\mu_k(S) \leq (k-3)a_1 + a_{n-1} + a_n + 1$. ■

Thus, we have now generalised this result for graphs of any girth. (Note that even for Girth $= 2$, the equation holds and both lower and upper bounds equate to $a_n + 1$). Also, we can notice that, for graphs with extremely high girth, a characterization might only be possible with a huge number of vertices. This leads to the following lemma for DAGs:

Lemma 17 *For a DAG, it is impossible to find a realization of a degree set with only positive numbers.*

Proof Only if 0 belongs to the degree set, can source and sink vertices be added to the graph. Since we know that DAGs require at least one of each, we can say that a DAG cannot $\wedge$-realize a positive degree set. ■

This should not be a surprise, as for DAGs ($k = \infty$), the minimum order to realise a degree set S, $\mu_k(S) \geq (k-2)a_1 + a_n$. Since $k = \infty$, $\mu_k(S) = \infty$.

CHAPTER 6

Summary and Future Work

6.1 Summary

A summary of our work is as follows:

- The Directed Graph Reachability problem falls to logspace when we confine the degree set to be $\{1\}, \{2\}$ or $\{1, 2\}$
- For all sets with at least one odd integer greater than 1, defining them as degree sets does not affect the complexity of the Directed Graph Reachability problem.
- For a set $S = \{a_1 < a_2 < \ldots < a_n\}$ of positive integers to be realised by a digraph of girth k, the minimum number of vertices required is given by,

$$(k-2)a_1 + a_n + 1 \leq \mu_k(S) \leq (k-3)a_1 + a_{n-1} + a_n + 1$$

- A DAG cannot $\wedge$-realize a positive degree set.

6.2 Future Work

- Settle the Reachability problem even when confined by an even-degree set.
- Attempt to use degree sets as a handle for other algorithmic problems such as the Graph Equivalence Problem.
- Close the gap between the upper and lower bounds for the minimum order for girth k digraphs.
- Extend these degree set results to include 0 in the degree sets.

REFERENCES

1. **Arikati, R., Srinavasa** and **A. Maheshwari** (1996). Realizing degrees sequences in parallel. *SIAM Journal of Discrete Mathematics*, **9**, 317–338.

2. **Chartrand, G.**, **L. Lesniak**, and **J. Roberts** (1976). Degree sets for digraphs. *Periodica Mathematica Hungarica*, **7**, 77–85. ISSN 0031-5303. URL `http://dx.doi.org/10.1007/BF02019997`.

3. **Kapoor, S.**, **A. Polimeni**, and **C. wall** (1977). Degree sets for graphs. *Fundamental Mathematics*, **95**, 189–194.

4. **Kumar, P.**, **M. Jayalal Sarma**, and **S. Sawlani**, On directed tree realizations of degree sets. *In* **S. Ghosh** and **T. Tokuyama** (eds.), *WALCOM: Algorithms and Computation*, volume 7748 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-36064-0, 274–285. URL `http://dx.doi.org/10.1007/978-3-642-36065-7`$_2$6.