Large Networks Bounded in Degree and Diameter

Hebert Pérez-Rosés

Submitted in total fulfillment of the requirements of the degree of
Doctor of Philosophy in Computer Science

School of Electrical Engineering and Computer Science
Faculty of Engineering and Built Environment
The University of Newcastle
Callaghan, NSW 2308, Australia

March 2012

Certificate of Originality

I hereby certify that this thesis contains no material which has been accepted for the award of any other degree or diploma in any university or other tertiary institution and, to the best of my knowledge and belief, contains no material previously published or written by another person, except where due reference has been made in the text. I give consent to this copy of my thesis, when deposited in the University Library, being made available for loan and photocopying subject to the provisions of the Copyright Act 1968.

(Signed) _____

Hebert Pérez-Rosés

Dedication

To the memory of my cousin Eduardo René Verdecia Rosés (1972-2006), who is deeply missed.

To the memory of Prof. Herbert Saul Wilf (1931-2012), who has inspired my work through his personal example and his awesome books.

Acknowledgements

Trying to summarize in a few words all the help I have received from different people during the three and-a-half years of my candidature is an extremely difficult task, and I want to start by apologizing in advance for any omission or understatement.

My principal supervisor, Professor Mirka Miller, has given me all the support that a supervisor can give, both at the professional and the personal level. More than just supportive, she is one of the nicest persons I've ever met. To me it is remarkable that she is always in a good mood, with a word of encouragement and understanding for everyone, and she always manages to create a positive working environment. No wonder she won the Vice-Chancellor's Award for Supervision Excellence in 2011.

My co-supervisor, Dr. Joe Ryan, has also been very helpful. He is noted for his sharpness, and is very quick with words and numbers. He is a valuabe asset for any student.

I owe a great deal to my friend and external co-supervisor, Dr. Guillermo Pineda-Villavicencio. He used to be my student once, but he has also taught me quite a few things. His determination and generosity have been an example for me all the time, and during my candidature in particular.

Collaboration with other people has also been rewarding. In that respect I wish to thank Prof. Francesc Comellas, my external co-supervisor in Spain, Prof. Miguel A. Borges, my former supervisor in Cuba, Dr. Eyal Loz, and Dr. Anthony Dekker. I have especially enjoyed working with Dr. Přemysl Holub, and expect to continue this fruitful collaboration in the future.

Other people have contributed with valuable discussions, pointers to the literature, have assisted with the endless paperwork, or simply by providing much needed encouragement. I will try to remember them all in alphabetical order: Ms. Kathy Allan, Prof. Brian Alspach, Prof. Krassimir Atanassov, Dr. Franọis Bertault, Prof. Ljiljana Brankovič, Prof. Charles Delorme, Mr. Ramiro Feria-Purón, Dr. Joan Gimbert†, Prof. José Gómez, Ms. Chanel Hopkinson, Prof. Robert Jajcay, Dr. Thomas Kalinowski, Ms. Kathy Killen, Dr. Yuqing Lin, Prof. Pablo Moscato, Mr. Oudone Phanalasy, Prof. Marko Petkovšek, Prof. Filiberto Pla, Prof. Ricardo Quirós, Prof. Zdeněk Ryjáček, Dr. Alain Tamayo, Prof. Paul Watters, and Prof. Doron Zeilberger.

Special thanks go to the examiners of this thesis: Prof. Margarida Mitjana, Prof. Franya Franek, and Prof. Tomáš Kaiser. They read the manuscript very carefully in record time, and made valuable suggestions for improving it.

At the personal level, what could I say about my family? Words could not possibly convey my gratitude to them. My parents Rubi and Hebert have made me who I am, in the first place, or more precisely, they are responsible for all the good things that I may have – the bad things are of my own design.

There can be no words either to thank my wife Arelis – my graph theory widow, and my daughter Karla – my graph theory orphan. They have endured my absence *for years* with incredible courage and patience.

What can I say about my cousin Miguel, who saved my life? What can I say about my brother Renato, my best friend and supporter?

I've also got the priceless support of my sister-in-law Chelo, my parents-in-law, Josefina and Rolando, our dear sister Teresa, all my cousins and aunties in Cuba, and my friends Arturo, Alejandro, Edgar, Guillermo Matos, Donna, and Jude. I apologize if I have not given back a fair percentage of what I have received from them.

I thank you all, as well as all the people who have ever lent me a hand, either in Cuba, Spain, or Australia.

List of Publications

Publications arising from this thesis

1. M.A. Borges-Trenard, H. Pérez-Rosés, and L. Puente-Maury. "Elimination in String Rewriting Systems", *Proceedings of the VI JMDA Conference* (J.Gimbert, ed.), Lleida, Spain, 2008, pages 493–500. ISBN: 978-84-8409-263-6. http://www.jmda2008.udl.cat/fitxers/actasVIjmda.zip.

2. G. Pineda-Villavicencio, J. Gómez, M. Miller and H. Pérez-Rosés. "New Largest Known Graphs of Diameter 6", *Networks* 53, (2009), pages 315–328. DOI: 10.1002/net.20269.

3. A. Dekker, H. Pérez-Rosés, G. Pineda-Villavicencio, and P. Watters. "The Maximum Degree & Diameter-Bounded Subgraph and its Applications", *Journal of Mathematical Modelling and Algorithms*, 2012. DOI: 10.1007/s10852-012-9182-8.

4. M. Miller, H. Pérez-Rosés, and J. Ryan. "The Maximum Degree & Diameter-Bounded Subgraph in the Mesh", *Discrete Applied Mathematics*, 160 (2012), pages 1782—1790. DOI: 10.1016/j.dam.2012.03.035.

5. P. Holub, M. Miller, J. Ryan, and H. Pérez-Rosés. "The Degree-Diameter Problem in honeycomb networks", in preparation, 2012.

6. P. Holub, M. Miller, H. Pérez-Rosés, and J. Ryan. "The Maximum Degree & Diameter-Bounded Subgraph in the hypercube", in preparation, 2012.

7. C. Delorme, E. Loz, H. Pérez-Rosés, and G. Pineda-Villavicencio. "Some Properties of Metabelian Groups for Constructing Large Graphs by Voltage Assignment", in preparation, 2012.

Other publications produced during my candidature

1. A. Garcés, R. Quirós, M. Chover, H. Pérez-Rosés, I. Quinzán, P. Espinosa. "Denotational Semantics for Social Communication among Agents". *Proceedings of the 18th International Conference on Software Engineering and Data Engineering (SEDE-2009)*, Las Vegas, USA, pages 42–46.

2. 7. M. Castañeda, O. Belmonte, J. Gil, H. Perez, and I. Coma. "A method for real-time detection of photometric perturbations in projected images" (in Spanish). Proceedings of the Spanish Congress of Graphics (CEIG), San Sebastián, September 2009. DOI: 10.2312/LocalChapterEvents/CEIG/CEIG09/239-242.

3. F. Bertault, R. Feria-Purón, M. Miller, H. Pérez-Rosés, E. Vaezpour. "A Heuristic for Magic and Antimagic Labellings",*Proceedings of the VII Symposium on Methaheuristics and Evolutionary Agorithms (MAEB)*, Valencia, Spain, 2010. Ibergarceta Publicaciones, S.L., ISBN 978-84-92812-58-5, pages 677–684.

4. M. Castañeda, O. Belmonte, H. Pérez-Rosés, and A. Díaz. "Detection of Non-Stationary Photometric Perturbations on Projection Screens", *Journal of Research and Practice in Information Technology*, accepted, 2012.

5. Y. Lin and H. Pérez-Rosés. "Constructing the poset of heterofullerenes.", in preparation.

Abstract

In the design of large interconnection networks several (and sometimes conflicting) factors come into play. Two of the most common requirements are:

- an upper bound on the number of connections attached to each node (the degree of the node in question), and
- an upper bound on the diameter of the network, which is the distance between the two nodes that are farthest apart.

With these two constraints we can try to build a network that is as large as possible, in terms of the number of nodes. In Graph Theory this classic problem is known as the **Degree-Diameter Problem**, or DDP for short. Research on this problem began in the 60s, but there are fundamental questions that still remain unanswered. There is a well known upper bound (the Moore bound) on the maximum number of nodes that can be achieved, given a maximum degree and a diameter. However, the number of networks reaching the Moore bound (or even approaching it, in the undirected case) is very small. For most combinations of maximum degree and diameter there is a gap (that can be very large for undirected networks) between the largest constructed network and the theoretical upper limit.

Consequently, research in this area can be roughly classified into two main categories:

1. Lowering the theoretical upper bound, by proving the nonexistence of networks with a given number of nodes, for a given combination of maximum degree and diameter.
2. Increasing the lower bounds, by constructing ever larger networks, for each combination of maximum degree and diameter.

Our research in DDP falls entirely within the second category: we investigate and apply methods to construct large networks. More precisely, we investigate two kinds of methods:

- Graph compounding, which produces large networks of diameter 6.
- Algebraic methods, based on Cayley graphs and its generalization, voltage assignment. We have applied algebraic methods to obtain large bipartite networks.

Additionally, we introduce a generalization of the Degree-Diameter Problem, which we have called the **Degree-Diameter Subgraph Problem**, or DDS for short, consisting in finding the largest subnetwork of a given host network, again subject to constraints on the maximum

degree and the diameter. It is noteworthy that DDS had not been investigated before, in spite of the fact that it is a natural generalization of DDP, hence we regard this as the main contribution of the thesis.

Our research in DDS falls within the two main research directions enumerated above, i.e. lowering the upper bounds, and raising the lower bounds. We have focused on some host networks of practical interest (mainly for parallel computing): the mesh, the hexagonal grid, and the hypercube. For those host networks we have determined Moore-like upper bounds for the largest subnetwork of a given maximum degree and diameter. Then we have applied some *ad hoc* construction techniques that yield families of subnetworks, which in most cases come quite close to, or even reach the theoretical upper limits.

Finally, we discuss DDS from the computational viewpoint. As a combinatorial optimization problem it is $\mathcal{NP}-$hard, hence finding an *exact* solution for large instances is hopeless with the current state of the art. We propose a heuristic algorithm that approximates the solution in polynomial time, and we investigate its performance, both theoretically and empirically.

Our work opens up many new interesting research directions, and we briefly discuss some of them in the thesis.

Contents

Part I

Preliminaries

1 Introduction

1.1 The Degree-Diameter Problem

In the design of large communication networks several (and sometimes conflicting) factors come into play. For instance, it is desirable that the number of connections attached to each node, as well as the distance between any two nodes of the network, remain small. In graph theory this situation is known as the *Degree-Diameter Problem* (or *DDP* for short), which asks to construct a largest possible graph (i.e. one with the largest possible number of vertices) having a fixed maximum degree and diameter. This problem was first stated by Elspas in 1964 [56], and has been investigated by numerous researchers ever since.

The range of applications of this problem is very large, and we do not intend to cover the whole range in this introduction. The interested reader can find some of these applications in [103]. In Section 1.2 we discuss a straighforward generalization of DDP, with an even wider range of practical applications.

DDP comes in two flavours: directed and undirected. We will mainly be concerned here with the undirected case, but our theoretical framework requires some coverage of the directed version as well. Let D represent the diameter of the graph (digraph) G, and let Δ stand for its maximum degree (outdegree). Now let us denote by $N_{\Delta,D}$ the maximum number of vertices that G may have, keeping Δ and D fixed. It is well known that $N_{\Delta,D}$ has an upper bound, called the Moore bound[1], which is attained in only very few cases. For most combinations of Δ and D, the actual size of a largest known graph is still very far from the Moore bound.

Moreover, DDP can be specialized for particular classes of graphs, like planar graphs, vertex-transitive graphs, Cayley graphs, and bipartite graphs. For each one of these classes there is a Moore-like bound, and a body of related research. For more information about the Moore bound and related issues, we refer the reader to [119].

Therefore, one of the main research areas in extremal graph theory lies in the construction of ever larger graphs satisfying the degree-diameter constraints, and getting closer to the Moore

[1] The Moore bound is different in the directed and undirected cases.

bound. Several construction methods have been tried, ranging from *ad hoc* constructions to computer-based methods.
The most successful technique so far is the one known as *voltage assignment* [77, 78]; it is credited with about 60% percent of the entries in the current table of the largest known graphs [63]. Voltage assignment takes a digraph $G = (V, A)$ (called base digraph) and a group Γ (called voltage group), and builds a derived digraph (or lift) with $|V||\Gamma|$ vertices and $|A||\Gamma|$ arcs. In Section 2.5.1 we summarize the main concepts and results of that theory.
Our work in this area is a mixture of computational and algebraic methods. So far we have mainly worked on ad hoc analytic constructions (graph compounding), and voltage assignment, as well as some accessory techniques. The use of voltage assignment for constructing large graphs is not an original idea; it has been used with success in the past by several researchers. However, we have found that the current theory of voltage assignment, and its application to DDP, is not sufficiently developed.
For instance, the use of voltage assignment has been so far limited to judicious choices of base graphs and voltage groups, or to nearly brute-force computer search. Very little is known about the conditions, either necessary or sufficient, that a base graph, a voltage group, and a particular voltage assignment must satisfy, in order to obtain a graph with some desired properties, like planarity.
As another example of this oblivion, only ordinary voltages have been used in connection with the degree-diameter problem. We do not know of any work that exploits the potential of permutation voltages, which produce graphs with less regularity, and may thus be used for the design of real-life networks. In general, we dare to assert that the tremendous power of the group-theoretic machinery is far from being exploited.
Taking into account that even under these circumstances the voltage assignment technique has been so successful, we are totally convinced that its potential is still enormous. In our research we have envisioned a more systematic methodology that does not rely on ad-hoc constructions or nearly blind computer search.

1.2 A more general approach to the Degree-Diameter Problem

A broad class of network design problems consists of finding a subgraph with given properties inside a given *host* graph. In the famous book by Garey and Johnson [68], this kind of network design problem already occupies an important place, since it also turns out that many of these problems are computationally hard. Since then, a lot of work has been carried out on the subject; we will just mention [5, 7, 95, 131] as recent examples. Some of the typical properties that we would like to control are:

1. *Size* (number of nodes). In many applications we need the largest, or the smallest possible network, that satisfies other properties as well

2. *Maximum degree.* For practical reasons, it is impossible to have too many connections attached to a single node

3. *Diameter* (the shortest distance between two nodes that are farthest apart). The diameter is an upper bound on the distance that a message has to travel inside the network, so it is important to keep it as small as possible. A related metric is average path length, which is often a more reliable measure of network performance than the diameter [42].

4. *Connectivity* (the smallest number of nodes - or links - that have to be removed in order to disconnect the network).

5. *Fault tolerance* (a generalization of connectivity, has to do with the number of nodes or links that have to be removed in order to make the network dysfunctional in some way). Obviously, we would like our network to have a high fault tolerance

6. *Symmetry* is a desirable property for network designers because it allows the implementation of the same algorithm at each component of the network. Symmetry is also related to fault tolerance [42, 41, 43]. Symmetry can be defined by several parameters, usually (but not exclusively) linked to the group of automorphisms of the network's graph (i.e. the set of transformations that leaves the graph invariant).

What we are facing now is a particular problem of that type, namely finding the largest connected subgraph with given maximum degree and diameter, contained inside a given host graph. The Degree-Diameter Problem, as it has been known to-date, is merely a special case of this broader problem, which we may call the Degree-Diameter Subgraph Problem (DDS), or more precisely, the Maximal Degree-and-Diameter Bounded Subgraph Problem (MaxDDBS). The practical implications of MaxDDBS are diverse: For example, communication time plays a crucial role in parallel and distributed processing, hence it may be important to identify a sub-network of bounded degree and diameter within the parallel architecture, in order to perform the computation efficiently.

A similar situation occurs in *botnets*. A botnet is basically a network of *bots* (malicious programs carrying out tasks for other programs or users), and controlled by members of organized crime groups, or 'botmasters'. Bots belonging to a botnet can be hosted on almost any computer, and very few internet users are immune to becoming a host. Some of the common malicious activities performed by botnets are Distributed Denial-of-Service Attacks (DDoS), and the distribution of spam. In those activities, the botmaster may try to choose a sub-network with the criteria enumerated above, in order to inflict the greatest possible damage, and at the same time remain immune to detection and regulation, whereas our goal is to predict the parameters of the attacking network, and take defensive steps against it.

Sohaee and Forst [139] apply bounded-diameter subgraphs to protein interaction networks (such as the 453-vertex metabolic network of the *Caenorhabditis elegans* nematode [54]) in order to identify the 'core' of the network. However, this may result in a subgraph dominated by a single node of high degree. On the other hand, subgraphs bounded by both degree and diameter may display a richer pattern of interaction, which may consequently be of greater interest.

1.3 Goals

Our main goal in this project can be stated in a very general way as follows:

Develop original techniques and methodologies for constructing large subgraphs of a given host graph, subject to upper bounds on the degree and the diameter.[2]

[2]Here the host graph could be a complete graph.

In turn, this main goal can be decomposed into several sub-goals:

1. In general, we intend to make a contribution to the classical Degree-Diameter Problem. In particular, we aim at improving the lower bounds with the aid of construction techniques, and a mixture of computational and algebraic techniques.
2. In the case of algebraic techniques, we wish to contribute to the theory of voltage graphs, with the aid of the powerful tools provided by classical and computational group theory. More specifically, we want to obtain conditions on the base digraph and the voltage group, that enable us to predict properties of the derived graph, such as bipartiteness. These conditions can lay the foundations for a methodology to construct large graphs of some particular classes via voltage assignment.
3. In the case of the Degree-Diameter Subgraph Problem, we focus on establishing Moore-like bounds for different kinds of host networks of practical importance.
4. Finally, we develop a practical algorithm for dealing with MaxDDBS and analyse its performance.

In order to carry out these goals, we needed to address some secondary or accesory issues, which may be interesting in themselves:

1. Expand the theory of rewriting systems, to enable its use in connection with voltage assignment.
2. Develop computer tools to perform calculations with graphs and groups. For example, we have designed and implemented functions for Knuth-Bendix completion of string-rewriting systems, computing voltage assignment, computing the MaxDDBS in a given host graph, and constructing graphs of certain classes, thereby expanding existing software libraries.
3. Create, maintain, and improve the website http://combinatoricswiki.org/wiki, as a collaborative repository of trustworthy and up-to-date information for practitioners of combinatorics and graph theory.

1.4 Organization of the thesis

The thesis consists of three main parts:

1. **Preliminaries**, where we provide the basic concepts and notation, state our research problems more formally, and discuss the relevant literature. Besides this introductory chapter, we have a chapter on algebraic structures, graphs, and their relationship (Chapter 2), and another chapter on network construction and algorithms (Chapter 3). Besides laying down the basis for the rest of the thesis, these two chapters discuss some recent relevant results and research directions. Given the variety of topics that we touch upon in the thesis, we have chosen this heterodox approach, rather than having a separate chapter devoted to literature review, in order to give more unity to the thesis.

2. **The Degree-Diameter Problem**, where we improve the existing lower bounds with the aid of construction, algebraic, and computational techniques. This part consists of two chapters: First, in Chapter 4 we give a construction of large graphs of diameter 6. Then, in Chapter 5 we develop some algebraic and computational techniques for use in conjunction with voltage assignment, in order to obtain large graphs with given degree and diameter.

3. **The Degree-Diameter Subgraph Problem**, where we investigate this generalization of DDP in some host graphs of practical interest, and we address the problem computationally. This part is made up by three chapters: In Chapter 6 we focus on DDS in the k-dimensional mesh, and we obtain Moore-like upper bounds for the general case, and lower bounds for some specific cases. Then, in Chapter 7 we follow the same approach with other host networks of practical importance: the hexagonal grid, and the hypercube. In Chapter 8 we discuss MaxDDBS from the computational viewpoint: We design, implement, and test a practical algorithm to solve the problem. Finally, we have a Conclusion chapter to summarize all the above information.

The stronger dependencies among the chapters are shown in Figure 1.1.

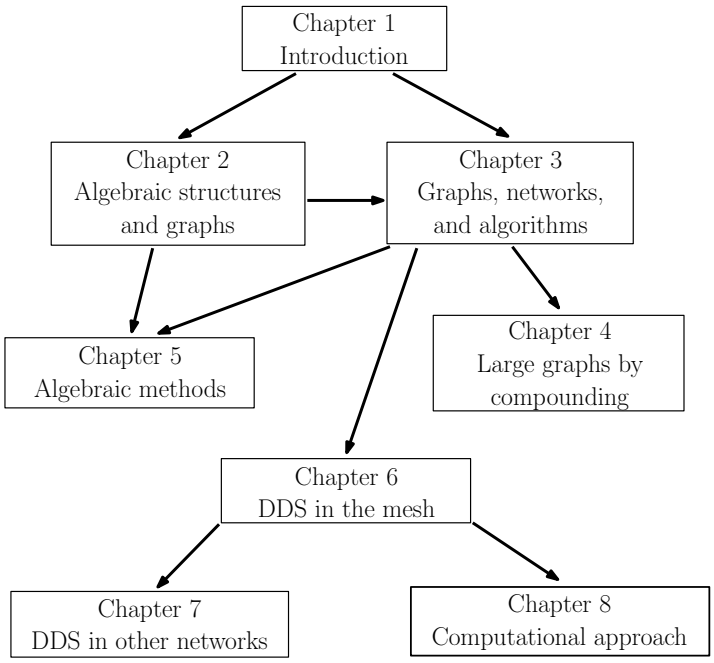


Figure 1.1: Chapter dependencies

Our original results are marked with the symbol ★, and the end of a proof is signalled with the symbol □. If the symbol □ appears right after an assertion, it means that the assertion in question is relatively straightforward to verify.

Personally, I feel that this thesis raises more question than it answers (which is positive in general, but can be frustrating at times). Consequently, I have set aside some space at the end of each chapter to discuss, or simply to enumerate, some of these open problems that I could not pursue to the end in the time frame that was allocated for the thesis.

2

Algebraic structures and graphs

This is the first of two chapters devoted to the basic concepts that will be used throughout the thesis. It fixes the notation and introduces the basic notions and results of algebraic structures (Section 2.1), graphs (Section 2.2), group presentations (Section 2.3), rewriting systems (Section 2.4), and the theory of voltage assignment (Section 2.5). Most of the results presented here are well known, and so they are stated without proof, with only the appropriate references. In Section 2.5 we also review some recent research directions that are directly connected with our work.

2.1 Basic algebraic structures

In this section we establish our set-theoretic notations, and review two fundamental algebraic structures: monoids and groups. It is impossible to give a complete overview of all, but the most basic group-theoretic concepts in such a short space. We will expand this foundation later in Sections 2.2.4, 2.3, 5.2, and 5.4, where the group-theoretic definitions are introduced in the context where they will be used. All this material can also be found in [132]. Personally, I am also very fond of [97] because it was a standard text in my undergraduate years. Another good text for group theory, that treats permutation groups in reasonable depth, is [99].

2.1.1 Sets, monoids, and groups

In what follows we will denote by $\mathbb{N}$ the set of natural numbers including 0, and by $\mathbb{Z}$ the set of all integers. The latter is, in turn, partitioned into $\mathbb{Z}^+$ (the set of positive integers), $\{0\}$ and $\mathbb{Z}^-$ (the set of negative integers). The sets of rational and real numbers are denoted by $\mathbb{Q}$ and $\mathbb{R}$, respectively. 2^S will stand for the power set of a set S, and $\binom{S}{k}$ will denote the set of all k-subsets of S.

Let Γ be a non-empty set, and $\circ$ a binary operation on Γ. The pair $(\Gamma, \circ)$ is a *monoid* if

(*i*) the operation $\circ$ is associative,

(*ii*) there exists an element $id \in \Gamma$, called *identity*, such that, for all $\gamma \in \Gamma$, $id \circ \gamma = \gamma \circ id = \gamma$

If additionally we have that

(*iii*) for each $\gamma \in \Gamma$, there exists an element $\gamma^{-1} \in \Gamma$, called the *inverse* of γ, such that $\gamma^{-1} \circ \gamma = \gamma \circ \gamma^{-1} = id$,

then the pair $(\Gamma, \circ)$ is called a *group*.
If the operation $\circ$ is commutative, the monoid (resp. group) is called *commutative*, or *abelian*. For abelian monoids and groups it is customary to use the symbol $+$ (additive notation) instead of $\circ$ (multiplicative notation). The multiplicative symbol $\circ$ is usually ommited if no confusions arise. Also, if the multiplicative notation is used, the identity element is usually denoted as 1_Γ (or just 1), and it is denoted as 0_Γ (or just 0) in the additive notation.
The *order* of a monoid (resp. group) Γ, denoted $|\Gamma|$, is the number of elements in Γ. In the case of groups, the order of a particular element $\gamma \in \Gamma$ is the smallest power n such that $\gamma^n = 1$. An element of order 2 is called an *involution* of Γ. In the remainder of this section we will be concerned mainly with groups, unless otherwise stated.
A subset S of Γ is a *subgroup* of Γ if S, together with the operation $\circ$, restricted to S, is itself a group. Given $X \subseteq \Gamma$, the *subgroup generated by* X, denoted $\langle X \rangle$, is the smallest possible subgroup S of Γ that contains X. $\langle X \rangle$ consists of all possible products $x_1 x_2 \cdots x_n$, where either $x_i \in X$, or $x_i^{-1} \in X$, for $1 \leq i \leq n$ and $n \in \mathbb{N}$. The set X is called a *generating set* of $\langle X \rangle$.
Given two groups $(\Gamma, \circ)$ and $(\Lambda, \diamond)$, a *group homomorphism* is a mapping $h : \Gamma \to \Lambda$ such that $h(\gamma_1 \circ \gamma_2) = h(\gamma_1) \diamond h(\gamma_2)$ for all $\gamma_1, \gamma_2 \in \Gamma$. If the homomorphism is onto, it is called an *epimorphism*, and if it is injective, then it is called a *monomorphism*. A homomorphism that is at the same time injective and surjective is an *isomorphism*, and if such a homomorphism exists between $(\Gamma, \circ)$ and $(\Lambda, \diamond)$, then they are *isomorphic groups*. If h is an isomorphism from $(\Gamma, \circ)$ and $(\Lambda, \diamond)$, then h^{-1} is also an isomorphism, this time from $(\Lambda, \diamond)$ to $(\Gamma, \circ)$.
Let h be a homomorphism between $(\Gamma, \circ)$ and $(\Lambda, \diamond)$. The following statements are immediate consequences of the group and homomorphism definitions:

- $h(1_\Gamma) = 1_\Lambda$
- $h(\gamma^{-1}) = h(\gamma)^{-1}$, for all $\gamma \in \Gamma$
- if S is a subgroup of $(\Gamma, \circ)$, then $h(S)$ is a subgroup of $(\Lambda, \diamond)$
- if S' is a subgroup of $(\Lambda, \diamond)$, then $h^{-1}(S')$ is a subgroup of $(\Gamma, \circ)$

The set $h^{-1}(1_\Lambda)$ is called the *kernel* of h, denoted $Ker(h)$.
An isomorphism from $(\Gamma, \circ)$ onto itself is an *automorphism* of $(\Gamma, \circ)$. The set of all automorphisms of $(\Gamma, \circ)$ is itself a group, together with the operation of composition of mappings, where the identity mapping is the group identity, and the inverse of a mapping is the inverse mapping. The automorphism group of Γ is denoted $Aut(\Gamma)$. Let $x \in \Gamma$; the automorphism ϕ_x defined as $\phi_x(\gamma) = x^{-1}\gamma x$, for all $\gamma \in \Gamma$, is the *inner automorphism* induced by x. The set $Inn(\Gamma)$ of inner automorphisms is a subgroup of $Aut(\Gamma)$.

A subgroup S of $(\Gamma, \circ)$ that remains invariant under every inner automorphism is called *normal*, and we denote it by $S \trianglelefteq \Gamma$. If h is a homomorphism between $(\Gamma, \circ)$ and $(\Lambda, \diamond)$, then $Ker(h)$ is a normal subgroup of $(\Gamma, \circ)$. The homomorphism theorem asserts that the converse is also true, i.e. every normal subgroup N of $(\Gamma, \circ)$ is the kernel of some homomorphism $h : (\Gamma, \circ) \to (\Lambda, \diamond)$. In this situation, the homomorphic image $(\Lambda, \diamond)$ is also called the *quotient group* of Γ by N, and is denoted Γ/N.

2.1.2 Permutation groups

The set of all bijections of an n-set X, denoted as $\mathcal{S}_X$, or simply as $\mathcal{S}_n$, is called the *symmetric group* on X. It is easy to show that if X_1 and X_2 are two n-sets, their corresponding symmetric groups are isomorphic, hence we can talk of the symmetric group on n elements, or the symmetric group of degree n (and order $n!$). A *permutation group* on X is a subgroup of $\mathcal{S}_X$. Let Γ be a permutation group on X, and $\gamma \in \Gamma$. Then γ can be represented as a set of disjoint permutation cycles. A *permutation cycle* is a sequence $\big(x, \gamma(x), \gamma^2(x), \ldots, \gamma^p(x)\big)$, where $x \in X$ and $\gamma^{p+1}(x) = x$. For instance, the permutation $\omega = \{2, 5, 4, 3, 1, 6\}$ can be represented as $(1, 2, 5)(3, 4)(6)$. The length of a cycle is the number of its elements.
A *permutation representation* of a group Γ is a homomorphism from Γ onto $\mathcal{S}_X$, for some set X. A permutation representation is also called an *action* of Γ on the set X, and Γ is said to act on X.
Let Γ be a group acting on a set X, then there is a homomorphism $h : \Gamma \to \mathcal{S}_X$. Let us set, for any $\gamma \in \Gamma$, $h(\gamma) = \pi_\gamma$. The *orbit* of an element $x \in X$, designated $Orb_\Gamma(x)$, is the set of all elements of X that are the image of x under some permutation $\pi_\gamma \in \mathcal{S}_X$. Formally, $Orb_\Gamma(x) = \{\pi_\gamma(x) | \pi_\gamma \in \mathcal{S}_X\}$. The orbit of x is a subset of X. All the orbits of Γ partition X, and the associated equivalence relation is defined by $x \sim y$ if, and only if, there exists $\gamma \in \Gamma$ such that $\pi_\gamma(x) = y$.
Let Γ be a permutation group acting on a set X. The group Γ is *transitive* on X if, for all $x, y \in X$, there exists $\gamma \in \Gamma$ such that $\gamma(x) = y$. That is, Γ is transitive on X if for all $x \in X$, $Orb_\Gamma(x) = X$. If moreover, the above permutation $\gamma \in \Gamma$, such that $\gamma(x) = y$, is unique, the group Γ is said to be *regular*. In a regular group Γ, no permutation $\gamma \neq id$ leaves any element of X fixed. Also, the order of a regular group on X is $|X|$.
The *stabilizer* of an element $x \in X$, denoted $Stab_\Gamma(x)$, is the set of all permutations $\gamma \in \Gamma$ mapping x to itself. Formally, $Stab_\Gamma(x) = \{\gamma \in \Gamma | \gamma(x) = x\}$. $Stab_\Gamma(x)$ is a subgroup of Γ. The following theorem relates orbits and stabilizers (Theorem 1.4A of [52]):

Theorem 2.1 (Orbit-Stabilizer) *Let Γ be a permutation group acting on a set X and let x be an element of X. Then for any $x \in X$,*

$$|Orb_\Gamma(x)||Stab_\Gamma(x)| = |\Gamma|$$

2.2 Graphs

This section contains the background material for graph theory. The terminology and notation introduced here are quite standard, though the approach we have used is somewhat unorthodox. We start by defining the basic concepts for directed graphs, and then we see

how they translate to the undirected case, which will be our main concern throughout this thesis. Unfortunately, no approach is perfect, and this one has its shortcomings too, but its merits outweigh any possible disadvantages, and ultimately, it is a matter of personal taste. The interested reader can find more information about directed graphs in [8], for instance, and some standard sources for undirected graphs are [21, 23, 30, 50, 78].

2.2.1 Directed graphs

Let V and A be two sets, such that $V \cap A = \emptyset$. A *directed graph* (or *digraph*, for short) is a pair $G = (V, A)$, where $A \subseteq V \times V$. The elements of V and A are called *vertices* and *arcs* of the digraph G, respectively. The vertex set V of a digraph G will be denoted by $V(G)$, and its arc set by $A(G)$. The number of vertices of G is called the *order* of G, and is denoted by $|G|$, as usual. A digraph of finite order is called *finite*. Here we will consider only finite digraphs. A digraph of order 0 is an *empty digraph* and is denoted by $\emptyset$. A *trivial* digraph is either an empty digraph or a digraph of order 1.
If $a = (u, v)$ is an arc, we say that u and v are its *endvertices*. More precisely, u is the *tail* and v is the *head* of a. We sometimes use the notation $u \to v$ for a. For $X, Y \subseteq V(G)$, $A(X, Y)$ is the set of all arcs going from a vertex in X to a vertex in Y. We say that X *dominates* u, denoted $X \to u$, if either $u \in X$ or there is an arc (x, u), with $x \in X$.
A digraph represents a binary relation on V. By the definition above, it is possible to have an arc $a = (v, v)$. In that case, we say that a is a *loop*. Even though not strictly allowed by the definition, sometimes it is also useful to consider *multiple* or *parallel* arcs, that are equal as ordered pairs. A digraph without multiple arcs or loops is called *simple*.
Let u be a vertex of G. The *out-degree* of u (denoted $\deg^+(u)$) is the number of arcs in G with tail u. The *in-degree* (denoted $\deg^-(u)$) is the number of arcs in G with head u. The set $N^+_\Lambda(x) = \{v|(u, v) \in A(G)\}$ is the *out-neighborhood* of u, whereas the set $N^-_\Lambda(x) = \{v|(v, u) \in A(G)\}$ is the *in-neighborhood* of u. If every vertex in a digraph G has the same in-degree (out-degree) then G is said to be *in-regular* (*out-regular*). If a digraph is in-regular of in-degree Δ and out-regular of out-degree Δ then it is called *diregular* of degree Δ.
Let G and G' be two digraphs. If $V(G') \subseteq V(G)$ and $A(G') \subseteq A(G)$ then G' is said to be a *subdigraph* of G, denoted $G' \subseteq G$. If $G' \subseteq G$ and $G' \neq G$, we say that G' is a *proper subdigraph* of G, denoted by $G' \subset G$. If $G' \subseteq G$ and G' contains all the arcs $(u, v) \in A(G)$ with $u, v \in V' = V(G')$ then G' is an *induced subdigraph* of G, denoted by $G[V']$. Given $G' \subseteq G$, G' is a *spanning subdigraph* of G if $V(G') = V(G)$.
Let $G = (V, A)$ be a digraph, where we allow multiple arcs and loops. A *walk* on G is an alternating sequence of vertices and arcs, $v_1, a_1, v_2, a_2, \ldots, v_k, a_k, v_{k+1}$, where v_1 is the *starting vertex*, v_{k+1} is the *end vertex*, and the arc a_i joins the vertices v_i and v_{i+1}, for $i = 1, 2, \ldots, k$. This walk is a $v_1 - v_{k+1}$ walk, since it goes from vertex v_1 to vertex v_{k+1}. Let X and Y be two sets of vertices; an $X - Y$ walk is a walk $v_1, a_1, v_2, a_2, \ldots, v_k, a_k, v_{k+1}$, where $x_1 \in X$, $x_{k+1} \in Y$ and $x_i \notin X, Y$, for $i = 2, \ldots, k$.
In most situations we can specify a walk by its arc sequence only, since the vertices can be inferred. Note also that we say nothing about whether $a_i = (v_i, v_{i+1})$ or $a_i = (v_{i+1}, v_i)$; that is, in a walk an arc can be traversed in reverse direction. Also, vertices and arcs can be traversed more than once. The *length* of the walk is k, the number of arcs. A walk of length k is called a $k-$*walk*. If $v_1 = v_{k+1}$ we say that the walk is *closed*, otherwise it is *open*.
A *natural walk* is a walk where arcs are traversed in the natural direction indicated by the

arrows, i.e. from tail to head. A natural walk can still repeat vertices and arcs. A *path* is a natural walk where all vertices are distinct, with the possible exception of v_1 and v_{k+1}. A closed path is called a *cycle*.
Let u, v be two vertices in a digraph G. The length of a shortest $u - v$ path in G is called the *distance* in G from u to v, denoted by $d_G(u, v)$. Note that in general $d(u, v) \neq d(v, u)$. If there is no $u - v$ path in G, then $d_G(u, v)$ is taken to be ∞.

A digraph G is *strongly connected* if for any two vertices u and v, there exists at least one $u - v$ path. G is said to be *(simply) connected* if for any two vertices u and v, there exists either an $u - v$ path, or a $v - u$ path.

2.2.2 Undirected graphs

An undirected graph (or simply *graph*) is a pair $G = (V, E)$ of sets satisfying $E \subseteq \binom{V}{2}$, where $V \cap E = \emptyset$. The elements of V and E are called *vertices* and *edges* of the graph G, respectively. It is customary to write the edges as pairs (u, v), even if they are unordered, although some authors prefer u, v or uv. We will mostly use the first and the third notation. It is possible to view an undirected graph as a special case of digraph, i.e. one where the relation A is symmetric. Therefore, most definitions that we have seen for digraphs in 2.2.1 carry over naturally for undirected graphs. For instance, the order of a graph is defined as in the directed case, as well as the concept of trivial graph. However, there are also new features that arise from the symmetry property, and we will review them shortly.
As in the case of digraphs, we will sometimes allow loops and multiple edges. A graph without multiple edges or loops is called *simple*. The vertex set V of a graph G will be denoted by $V(G)$, and its edge set by $E(G)$. If $e = (u, v)$ is an edge, then u and v are called the *endvertices* of a, and we also say that u and v are *adjacent* or *neighbours*, and also that they are *incident* with e, while e is said to be *incident* with u and v. Two edges $e \neq f$ are adjacent if they share an endvertex. If two vertices u and v are not adjacent then we write $u \nsim v$.
For $X, Y \subseteq V(G)$, $E(X, Y)$ is the set of all edges going from a vertex in X to a vertex in Y. The set of neighbours of a vertex v in G is denoted by $N(v)$. The set of all edges in $E(G)$ incident with a vertex v is denoted by $E_G(x)$. In the case of undirected graphs, the in-degree and out-degree of a vertex v are the same, and they are referred to as the *degree* of v, denoted $\deg(v)$. The number $\Delta_G = \max\{\deg(v) | v \in V\}$ is the *maximum degree* of G, while $\delta_G \min\{\deg(v) | v \in V\}$ is the *minimum degree*. We say that $X \subseteq V(G)$ *dominates* v (denoted $v \sim X$), if v either belongs to X or is adjacent to a vertex of X. Otherwise, we use the symbol $\nsim$.
A set of elements is called *independent* or *stable* if it contains no adjacent elements. A set of independent edges is called a *matching*. A *perfect matching* M of G is a matching such that, for any vertex $v \in G$, there is an edge in M incident with v.
The concepts *subgraph*, *proper subgraph*, *induced subgraph*, and *spanning subgraph*, are similar to their directed counterparts. In an undirected graph, all walks are natural walks, and the concepts *path* and *cycle* are defined in the same manner as in the directed case. The minimum length of a cycle in a graph G is the *girth* of G, and is denoted by $g(\Gamma)$. A graph G without cycles is called *acyclic*, and in this case, $g(G)$ is taken to be ∞.
The distance from u to v is also defined in the same manner, with the remark that in this case, $d(u, v)$ is always equal to $d(v, u)$. The *eccentricity* of a vertex v, denoted $e_G(v)$, is defined as the maximum distance between v and any other vertex. The *diameter* of a graph

G, denoted $D(G)$, is equal to the maximum eccentricity over all vertices of the graph, that is, the greatest distance between any two vertices in G. Formally, $e_G(v) = \max_{x \in V(G)} d_G(v, x)$ and therefore, $D(G) = \max_{v \in V(G)} \max_{x \in V(G)} d_G(v, x)$. The *radius* of a graph G, denoted $rad(\Gamma)$, is equal to the minimum eccentricity over all vertices of the graph. That is, $rad(G) = \min_{x \in V(G)} \max_{y \in V(G)} d_G(x, y)$.
The distinction between strongly connected and simply connected does not make sense in undirected graphs, therefore we talk only about *connected* graphs.

2.2.3 Some special graphs

Let $k \geq 0$. A *path* graph $P = P_k$ is a graph such that

$$V(P) = \{x_0, x_1, \ldots, x_k\} \text{ and } E(P) = \{x_0x_1, x_1x_2, \ldots, x_{k-1}x_k\}.$$

The vertices x_0 and x_k are the *endvertices* or *ends* of P, and the vertices $x_1, \ldots, x_{k-1}$ are its *inner* vertices. Two paths are called *independent* if they do not share an inner vertex. The subpath $P[x_i, x_j]$ denotes the induced subgraph $P[x_i, \ldots, x_j]$.
A path (cycle) in a graph G is *hamiltonian* if it contains all the vertices of G.

A *complete* graph of order n, denoted K_n, is a graph where every vertex is adjacent to every other one.
Let $r \geq 2$ be an integer. A graph G is called *r-partite* if it is possible to partition $V(G)$ into r independent sets (called *partite sets*). A 2-partite graph is called *bipartite*, and it is written $G = (V_1 \cup V_2, E)$, where V_1 and V_2 are the partite sets. A graph is bipartite if, and only if, it contains no odd cycle. Given a bipartite graph $G = (V_1 \cup V_2, E)$, if every vertex of V_1 is joined to every vertex of V_2, then G is a *complete bipartite* graph, and we denote it by $K_{m,n}$, where $m = |V_1|$ and $n = |V_2|$.

If all the vertices of graph G have the same degree Δ then G is called *Δ-regular*, or just *regular*. A 3−regular graph is called *cubic*.

Regarding directed graphs, there are two types of digraphs that are commonly used in the literature in connection with voltage assignment (see Section 2.5): The *bouquet* consists of a single vertex and several loops.[1] The *dipole* or *dumbbell* consists of two vertices joined by multiple arcs, and possibly with loops at each vertex. A dipole with three arcs and no loops is commonly referred to as a *theta-graph*.

2.2.4 Graph isomorphism

Let G and G' be two digraphs. $G = (V, A)$ and $G' = (V', A')$ are said to be *isomorphic* (denoted $G \cong G'$) if there are two bijections, $f_V : V \to V'$, and $f_A : A \to A'$, so that for every arc $a = (x, y) \in A$ we have $f_A(a) = (f_V(x), f_V(y))$. The pair of functions f_V, f_A is called an *isomorphism* of G onto G'.
If G and G' are simple digraphs, then we can rephrase this definition in terms of f_V only: $G \cong G'$ if there is a bijection $f_V : V \to V'$ such that $(x, y) \in A$ if, and only if, $(f_V(x), f_V(y)) \in A'$.

[1] Some authors also include semiedges.

Obviously, $\cong$ is an equivalence relation in the set of all digraphs. These definitions carry over to undirected graphs.

An isomorphism of G onto itself is called an *automorphism*. The set of automorphisms of a graph G forms a group with the operation of composition of functions. This group is called the *automorphism group* of G, and is denoted $Aut(\Gamma)$. The group $Aut(G)$ is a permutation group on $V(G)$.
We say that G is *vertex-transitive* if $Aut(G)$ acts transitively on $V(G)$, that is, for any two vertices, u and v, there is an automorphism $\omega \in Aut(G)$ such that $\omega(u) = v$. Analogously, we say that G is *edge-transitive* if for any two edges e and f in $E(G)$, there is an automorphism $\omega \in Aut(G)$ such that $\omega(e) = f$, where $\omega(x, y) = (\omega(x), \omega(y))$.
Finally, we say that a connected graph G is *distance-transitive* if, for all vertices u, v, x and y of G such that $d(u, v) = d(x, y)$, there is an automorphism $\omega \in Aut(G)$ satisfying $\omega(u) = x$ and $\omega(v) = y$.
Let G be a bipartite graph with partite sets V_1 and V_2. A *polarity* of G is an involutory element of $Aut(G)$ that interchanges V_1 and V_2.

2.3 Presentations

A *presentation* is an abstract way of defining an algebraic structure by means of generators and relations among them. A presentation is usually given as a set of equations on words. They are closely related to *Cayley graphs*. In this section we give an overview of those concepts, which can be found in more detail in [99, 38], and other sources.

2.3.1 Monoid and group presentations

Let Σ be a finite set of symbols; we shall denote by Σ^* the free monoid generated by Σ, i.e. the set of all words that can be constructed with letters of the alphabet Σ, together with the operation of concatenation. The identity element (the empty word) will be denoted either as $\mathbf{1}$ or λ. If we have a binary relation R on Σ^*, then $\equiv_R$ will stand for the congruence generated by R, i.e. the reflexive-symmetric-transitive closure of R, that is compatible with the concatenation operation. For the sake of simplicity, we can omit the subscript R whenever that does not lead to confusion. For a word $\alpha \in \Sigma^*$, $[\alpha]_R$ will be the equivalence class of α modulo $\equiv_R$.
The concatenation operation naturally induces an operation " $\circ$ " in the quotient

$$M = \frac{\Sigma^*}{\equiv_R},$$

namely, $[\alpha]_R \circ [\beta]_R = [\alpha\beta]_R$. This gives rise to the following definition:

Definition 2.1 (Monoid presentation) *The pair $\langle \Sigma : R \rangle$ is called* a presentation *of the monoid $\langle M, \circ \rangle$. The latter is, in turn, the* monoid presented *by $\langle \Sigma : R \rangle$, and is also denoted as $Mon(\Sigma, R)$. The elements of Σ are the* generators *of the monoid, and R is the set of* defining relations. *The identity element of this monoid is $[\mathbf{1}]_R$.*

Example 2.1

$$\langle a, b, c : a^2 = b^2 = c^2 = \mathbf{1},\ bab = aba,\ ca = ac,\ cbc = bcb,\ cbac = bcba \rangle,$$

is a monoid presentation (the monoid presented is the symmetric group of degree 4).

In the previous example we have followed the tradition of writing the pairs of words that make up a defining relation as equalities. Moreover, we will only be concerned with *finite presentations*, i.e. presentations where R is finite, like the one of example 2.1.
A fundamental algorithmic problem that arises from this definition is the so-called *word problem*, that can be defined as follows:

Definition 2.2 (Word problem in monoids) *Let $\langle \Sigma : R \rangle$ be a fixed monoid presentation.*
Instance: *Two words $\alpha, \beta \in \Sigma^*$.*
Question: $\alpha \equiv_R \beta$ *?*

In the uniform version of the word problem, the presentation becomes part of the problem instance:

Definition 2.3 (Uniform word problem)
Instance: *A monoid presentation: $\langle \Sigma : R \rangle$, and two words, $\alpha, \beta \in \Sigma^*$.*
Question: $\alpha \equiv_R \beta$ *?*

Both versions of the word problem are undecidable in general (Section 2.5 of [24]).
Let us now take Σ^{-1} as the set of inverse symbols of Σ, that is: $\Sigma^{-1} = \{x^{-1} : x \in \Sigma\}$, and let I be a binary relation on $(\Sigma \cup \Sigma^{-1})^*$ defined as

$$I = \{(xx^{-1}, \mathbf{1}), (x^{-1}x, \mathbf{1}) : x \in \Sigma\}.$$

Here too, a binary operation is induced in the quotient

$$F = \frac{(\Sigma \cup \Sigma^{-1})^*}{\equiv_I},$$

which we will represent again by means of the symbol " $\circ$ ".

Definition 2.4 (Free group) *The monoid $\langle F, \circ \rangle$ described above is the* free group *generated by Σ.*

Let us now consider a binary relation R on $(\Sigma \cup \Sigma^{-1})^*$, and the set

$$G = \frac{(\Sigma \cup \Sigma^{-1})^*}{\equiv_{(R \cup I)}},$$

which, together with the operation " $\circ$ " induced on it, becomes a group:

Definition 2.5 (Group presentation) *The pair* $\langle \Sigma : R \rangle$ *is called (a)* presentation *of the group* $\langle G, \circ \rangle$. *The latter is, in turn, the* group presented *by* $\langle \Sigma : R \rangle$, *and is denoted as* $Grp(\Sigma, R)$.

Clearly, every group presentation can be transformed into a monoid presentation. This can be done, for example, by taking $\Sigma := \Sigma \cup \Sigma^{-1}$, and $R := R \cup I$; but in general, it suffices to add a single extra generator [146]. Moreover, if G is a torsion group,[2] then any group presentation for G can be transformed to a monoid presentation for G without having to add any extra generators. In particular, that holds for all finite groups; the presentation of example 2.1 is a monoid presentation for the group S_4.
For group presentations we can also define the word problem in its two versions, and new decision problems arise, like the *generalized word problem*:

Definition 2.6 (The generalized word problem) *Let* $\langle \Sigma : R \rangle$ *be a fixed monoid presentation for a group* G.
Instance: *A set of words* $w_1, w_2, \ldots, w_k \in \Sigma^*$, *that define a subgroup* H *of* G, *and another word* $\alpha \in \Sigma^*$.
Question: *Does the element represented by* α *belong to* H*?*

Needless to say, the generalized word problem is undecidable, too.

2.3.2 Cayley graphs

Let Γ be a group, and $S \subset \Gamma$, where $1 \notin S$.[3] The *Cayley color digraph* on Γ with connection set S is a digraph with vertex set $V = \Gamma$, and there is an arc going from g to h, labeled s (where $s \in S$), if $sg = h$. The labels are the 'colors'; if we drop them, we get the *Cayley digraph* on Γ with connection set S.
Now, if we take $S := S \cup S^{-1}$, and we collapse two opposing arcs that have inverse labels, then we get the *Cayley color graph*, and the *Cayley graph*, respectively, with connection set S. The symbol $Cay(\Gamma, S)$ will denote the Cayley graph on Γ with connection set S. A nice introduction to Cayley graphs is given in [3, 4].
Cayley graphs are vertex-transitive, however the converse is not true in general. The following result gives a complete characterization of Cayley graphs:

Theorem 2.2 ([133]) *A graph* G *is a Cayley graph if, and only if,* $Aut(G)$ *contains a regular subgroup.*

Notable Cayley graphs include the complete graph K_n, the complete multipartite graph, and the $d-$dimensional cube Q_d. A Cayley graph on the cyclic group $\mathbb{Z}_n$ is called a *circulant graph*. Due to their symmetry properties, among others, Cayley graphs (including circulant graphs) have been widely studied as models of communication networks since the late 80's and early 90's (see [31, 114, 2, 37, 135, 101, 83, 154, 67], for instance).

[2] A torsion group is a group where every element has finite order.
[3] Some authors require that S be a generating set of Γ.

However, the main emphasis so far has been on Cayley graphs of permutation groups, most likely due to the fact that there exist efficient algorithms to deal with them. Our approach is rather different: we discuss tools and methods to deal with Cayley graphs of finitely presented groups.

2.4 String rewriting systems

We have said that it is customary to write the defining relations of a presentation as equations involving two words. This approach underlines the non-oriented or bidirectional nature of such relations (let us recall that we take the symmetric closure of R). Nevertheless, in order to tackle the algorithmic problems associated to presentations, we could consider giving an orientation to the defining relations, and always use them in one direction to transform words. That way, defining relations would become *string rewriting rules*. The advantages of using this oriented approach will become evident in this section, when we present the useful properties of string rewriting systems. The definitions and results given here have been taken mainly from the monograph by Book and Otto [24], and from the survey article by Cohen [32].

Definition 2.7 (String rewriting system) *Let Σ be a finite alphabet, and R a binary relation on Σ^*. The pair $\langle \Sigma : R \rangle$ is a* string rewriting system *on Σ. Every element (l, r) is a rewriting rule; l is called* left-hand side*, or* head *of the rule, whereas r is the* right-hand side*, or* tail.

It is customary to refer to the relation R as the rewriting system, since the alphabet can generally be deduced from R. We also say that R is a string rewriting system on Σ. Here too, unless otherwise stated, we will assume that R is finite.

Definition 2.8 (Reduction) *Let R be a string rewriting system on Σ. The* one-step reduction relation *induced by R, and denoted as $\to_R$, is defined as follows: for any two words $u, v \in \Sigma^*$, $u \to_R v$ if, and only if, there exists $(l, r) \in R$, and $x, y \in \Sigma^*$ such that $u = xly$ and $v = xry$. In this case we say that u* is reduced *to v (in one reduction step), and the word u is said to be* reducible*. If u is such that there does not exist v, with $u \to_R v$, then u is said to be* irreducible*. The reflexive and transitive closure of $\to_R$ is called* the reduction relation *induced by R, and is denoted $\overset{*}{\to}$.*

An immediate consequence of the above definition is that $l \to_R r$ for every $(l, r) \in R$, which enables us to specify rewriting rules with the notation $l \to_R r$. Here we can also omit the subscript R, whenever that omission poses no threats for the understanding of the text.
Besides the relation $\overset{*}{\to}$, rewriting theorists make abundant use of the relations $\leftrightarrow$ (the symmetric closure of $\to$), and $\overset{*}{\leftrightarrow}$ (its reflexive, symmetric, transitive closure); the latter coincides with our $\equiv$, defined in Section 2.3. In the same way that we have done before, $[\alpha]_R$ will be the equivalence class of α modulo $\equiv_R$.

Definition 2.9 (Properties of string rewriting systems) *We will say that two words, u, v are* joinable *(which is denoted as $u \downarrow v$) if there exists a word w such that $u \overset{*}{\to} w$ and $v \overset{*}{\to} w$. The rewriting system R is called* confluent *if $u \overset{*}{\to} v$ and $u \overset{*}{\to} w$ imply $v \downarrow w$, and*

it is called locally confluent *if* $u \to v$ *and* $u \to w$ *imply* $v \downarrow w$. *R is said to be* terminating, well founded *or* noetherian *if there does not exist an infinite sequence* $u_1 \to u_2 \to \ldots \to u_n \to \ldots$.

The relationship among the properties just defined are given by the following, classical result.

Lemma 2.1 (Newman) *If R is terminating, then local confluence amounts to confluence.*

Definition 2.10 (Complete rewriting system) *It is said that R has the* Church-Rosser property *if* $u \equiv_R v$ *implies* $u \downarrow v$. *If R is, at the same time, terminating and Church-Rosser, we say that it is* complete, canonical, *or* convergent.

In the definition of a complete system, the condition of having the Church-Rosser property can be replaced by confluence, since both properties turn out to be equivalent. Completeness is a very important property from the algorithmic viewpoint, for it guarantees the existence of a unique irreducible word in each equivalence class modulo $\equiv_R$.
Thus, if we use a complete string rewriting system to specify a monoid, then the number of irreducible words corresponds exactly to the order of the monoid. For an arbitrary string rewriting system, the number of irreducible words is an upper bound of the order of the monoid [69].
A presentation of a monoid M with the form of a complete string rewriting system is called a *complete presentation* of M. Such a presentation provides a natural algorithm for solving the word problem in M. Namely, let us denote by $NF(w)$ the only irreducible word in the equivalence class of w, which can be computed by virtue of the termination property. Let now α and β be two words in the generators of M: $\alpha \equiv_R \beta$ if, and only if, $NF(\alpha) = NF(\beta)$.
From now on, $Irr(R)$ will denote the set of irreducible words modulo R.

Definition 2.11 (Normalized systems) *The system R is said to be* normalized *if for every rule* $\alpha \to \beta$ *of R,* $\beta \in Irr(R)$, *and* $\alpha \in Irr(R \setminus \{\alpha \to \beta\})$.

Definition 2.12 (Equivalence of systems) *Two string rewriting systems, R and S, on the same alphabet Σ are said to be* equivalent *if they generate the same congruence $\equiv$; that is, if $\equiv_R$ coincides with $\equiv_S$.*

The underlying idea in normalized systems is that they do not have redundant rules. Given an arbitrary rewriting system R, we can always find a normalized system R', equivalent to R. Now, given an arbitrary rewriting system R, is it always possible to find a complete system $\overline{R}$, equivalent to R? Unfortunately, the answer to this question is negative. Moreover, given a system R, in general it is undecidable to determine whether R is terminating or confluent. In order to prove termination, we can use a partial ordering on Σ^* with certain properties.

Definition 2.13 (Term ordering) *Let $<$ be a strict well-founded partial ordering on Σ^*, and R a string rewriting system on Σ. We say that $<$ is* admissible *if* $u < v$ *implies* $xuy < xvy$, $\forall u, v, x, y \in \Sigma^*$. *Admissible well-founded orderings are called* term orderings.
We say that $<$ is compatible *with R if $\beta < \alpha$ for all rewriting rules $\alpha \to \beta$ in R.*

Theorem 2.3 *Let R be a rewriting system on Σ. R is noetherian if, and only if, there exists a strict partial ordering $<$ on Σ^*, which is admissible and compatible with R.*

One of the term orderings most frequently used is the so-called "ShortLex" or "DegLex" ordering, where comparison between two words is based in the first place on their lengths: if one of the words is longer than the other, then this longer word is taken to be the greater; if both words have the same length, then they are compared lexicographically. Besides the "ShortLex" ordering, we have used an ordering suggested by Mora [120], which is defined as follows:
Let us suppose that a total ordering has been defined on Σ, and let $S_{max}(w)$ be a maximal symbol occurring in the word $w \in \Sigma^*$. Given w_1 and w_2 with $w_1 \neq w_2$, if $S_{max}(w_1) < S_{max}(w_2)$, then $w_1 < w_2$. Otherwise, if $S_{max}(w_1) = S_{max}(w_2) = x$, then we can write:

$$w_1 = u_1 x u_2 x \dots u_{r-1} x u_r, \text{ y } w_2 = v_1 x v_2 x \dots v_{p-1} x v_p,$$

where the u_i and v_j do not contain any occurrence of x. In this case, if $r < p$ we take $w_1 < w_2$. Finally, if $r = p$ then we make $q = \max\{i / u_i \neq v_i\}$, and if $u_q < v_q$ then we'll also have $w_1 < w_2$.
Mora's ordering possesses interesting properties that we discuss in Chapter 5. We have used this ordering systematically for the computation of our complete presentations.
Once we have proved R to be noetherian, confluence can be verified with the aid of the *Knuth-Bendix criterion* [93], introduced in the sequel.
Let us denote by $LRED(\alpha)$ the irreducible word of $[\alpha]_R$ that is obtained from α by performing reduction steps to the leftmost possible subword of α that is amenable to reduction.
An *overlap ambiguity* is a triple of nonempty words, α, μ, β such that there exist rules $\alpha\mu \to \gamma_1$ and $\mu\beta \to \gamma_2$ in R. In this case, the pair of words $\gamma_1\beta$ and $\alpha\gamma_2$ is called a *critical pair*, since $\alpha\mu\beta$ could be reduced to them both.
The triple α, μ, β is called an *inclusion ambiguity* if there exist rules $\mu \to \gamma_1$ and $\alpha\mu\beta \to \gamma_2$ in R, where $\gamma_1 \neq \gamma_2$ if α and β are both empty. The pair of words $(\alpha\gamma_1\beta, \gamma_2)$ is the corresponding critical pair in this case.
Finally, we have the following

Theorem 2.4 *Let R be a terminating string rewriting system. R is locally confluent (and hence, complete) if, and only if, for any critical pair (δ_1, δ_2) we have $LRED(\delta_1) = LRED(\delta_2)$.*

Clearly, in order to verify that $LRED(\delta_1) = LRED(\delta_2)$ it suffices to check that $\delta_1 \downarrow \delta_2$. When this happens, we say that the critical pair *resolves*.
The above criterion suggests a computational procedure to 'complete'a finite noetherian string rewriting system R. That procedure (also known as the Knuth-Bendix procedure), when it does terminate, produces a complete string rewriting system $\overline{R}$, equivalent to R.
The procedure works by adding the critical pairs that do not resolve to the original system, as new rewriting rules. During this process, the underlying ordering $<$ is used in an essential way: the new rules are oriented in such a way that $<$ continues to be compatible with the resulting system, which guarantees the termination of the latter.
In Algorithm 1 we give the basic Knuth-Bendix procedure for string rewriting systems, as given in [24]. $PAC(R)$ denotes the set of critical pairs of the system R. In [152, 137] some modifications to the basic procedure are discussed, with the aim of improving its performance.

The KBMAG package (which is distributed with GAP) [85] contains an efficient implementation of this procedure. Additionally, we have our own implementation, written in GAP 3, which has less functionality than the KBMAG package, but is easier to use.

Algorithm 1: Knuth-Bendix procedure

Input : A rewriting system R on Σ and an admissible ordering $<$.
Output: A complete system $\overline{R}$, equivalent to R.

```
   /* Orient the system according to <                                  */
1  R_0 := {(l,r) : l > r and [(l,r) ∈ R or (r,l) ∈ R]};
2  i:=−1;
3  repeat
4  |  i := i + 1;
5  |  CP:=PAC(R_i);
6  |  while CP ≠ ∅ do
7  |  |  Choose (z_1, z_2) ∈ CP;
8  |  |  z_1 bar:= LRED(z_1);
9  |  |  z_2 bar:= LRED(z_2);
10 |  |  if z_1 bar > z_2 bar then
11 |  |  |  R_{i+1}:=R_{i+1} ∪ {(z_1 bar, z_2 bar)};
12 |  |  end
13 |  |  if z_2 bar > z_1 bar then
14 |  |  |  R_{i+1}:=R_{i+1} ∪ {(z_2 bar, z_1 bar)};
15 |  |  end
16 |  |  CP:= CP - {(z_1, z_2)};
17 |  end
18 |  if R_{i+1} ≠ ∅ then
19 |  |  R_{i+1}:=R_{i+1} ∪ R_i
20 |  end
21 |  ;
22 until R_{i+1} = ∅;
23 R bar:=⋃_{i≥0} R_i;
```

2.4.1 Application: Computing the diameter of Cayley graphs

In Section 2.4 we have seen that if we have a monoid or group specified by a complete rewriting system, then there is a one-to-one correspondence between the set of irreducible words and the set of elements of the monoid or group in question. We can use this correspondence to perform certain computations in the corresponding Cayley graph, e.g. to compute its diameter.

First let us remember that a Cayley graph is vertex-transitive. Therefore, in order to compute its diameter it suffices to compute the longest path from a fixed vertex, e.g. the vertex labelled with the identity element of the group. Suppose that S is a generating set of Γ and S is closed under the operation of taking inverses (so that $Cay(\Gamma, S)$ is connected and undirected). Then,

the diameter of $Cay(\Gamma, S)$ is $\max\{\mathrm{len}_S(\gamma)|\gamma \in \Gamma\}$, where $\mathrm{len}_S(\gamma)$ is the length of the shortest word in the alphabet S which represents γ.

Computing the diameter of $Cay(\Gamma, S)$ is, in general, a very difficult problem. Paraphrasing [86], Section 4.8.3, this problem "represents a failure of Computational Group Theory, in that no methods have been found that are significantly better than brute-force search".

Now, let us assume that Γ is presented by a complete string-rewriting system on the generating set S, with respect to the ShortLex ordering. Then, the most distant vertex corresponds to the longest irreducible word, since an irreducible word is at the same time the shortest word representing the element in question.

In order to find the longest irreducible word we can traverse the set of irreducible words in depth-first fashion (DFS), keeping a record of the longest irreducible word found so far. That is, for every new word generated, we have to check whether it is irreducible modulo the string rewriting system, and if so, compare its length with the longest so far. DFS traversal can be done very efficiently in terms of memory usage, since only one word has to be kept in memory at a time.

In many cases we can also save time by noting that the longest word is located at the end of the leftmost branch, i.e. it is the leaf of the DFS tree that appears right before the first bactrack. However, even this does not guarantee a significant saving in computing time in the worst case.

It is true that in order to apply this method we needed to compute a complete string-rewriting system first, and that is a costly process in itself, but on the other hand, this complete rewriting system could have been obtained by other means than the Knuth-Bendix procedure. For example, in Section 5.1.1 we explain a method that can be used to obtain a new complete string-rewriting system from a previous one, at a small computational cost.

If S does not contain the inverses of the elements, then $Cay(\Gamma, S)$ is directed, and the above discussion still applies.

Example 2.2 *As a simple example, Figure 2.1 shows the DFS tree of irreducible words of the quaternion group, given by the complete monoid presentation* $\langle a, b;\ b^2 = a^2, aba = b, ba^2 = a^2b, bab = a, a^4 = 1, a^3b = ba\rangle$. *As the generating set* $\{a, b\}$ *does not contain the inverses, the corresponding Cayley graph is directed. The longest words in the tree are* a^3 *and* a^2b, *hence the diameter of the Cayley graph is 4.*

Example 2.3 *For a more serious example, let* $\mathbf{F_4}$ *be the group of order 1152, with generating set* $\{a, b, c, d\}$ *and defining relations* $a^2 = b^2 = c^2 = d^2 = 1$, $bab = aba, ca = ac, cbcb = bcbc, da = ad, db = bd, dcd = cdc$. *Note that each generator is its own inverse, hence the corresponding Cayley graph is undirected. We compute a complete string-rewriting system with respect to the ShortLex ordering:*

$$a^2 = b^2 = c^2 = d^2 = 1,$$

$$ca = ac, da = ad, db = bd, bab = aba, dcd = cdc,$$

$$cbcb = bcbc, dcbd = cdcb, dcbad = cdcba, cbacba = bcbacb,$$

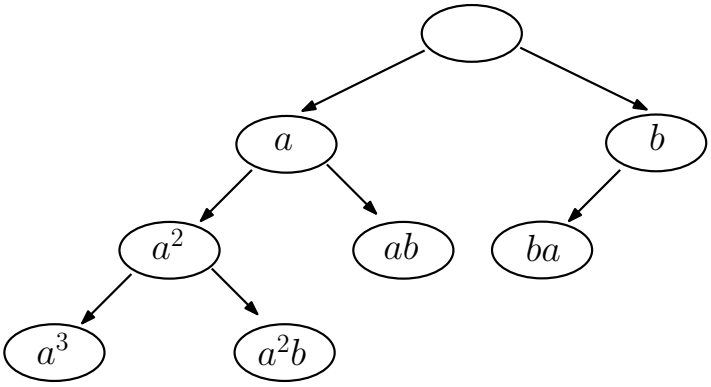


Figure 2.1: DFS tree of irreducible words of the quaternion group $\langle a, b;\ a^4 = 1, b^2 = a^2, ba = a^3b\rangle$

$$dcbcdc = cdcbcd, dcbacdc = cdcbacd, dcbacbdcbc = cdcbacbdcb,$$

$$dcbacbdcbac = cdcbacbdcba, dcbacbcdcbcd = cdcbacbcdcbc,$$

$$dcbacbcdcbacd = cdcbacbcdcbac,$$

$$dcbacbcdcbacbd = cdcbacbcdcbacb,$$

$$dcbacbcdcbacbcdc = cdcbacbcdcbacbcd.$$

With respect to this system, the longest word has length 24.

2.5 Voltage assignment

We have seen the connection between graphs and groups in Section 2.2.4, with the automorphism group of a graph, and in Section 2.3.2, with the Cayley graph of a group. In this section these two concepts meet again: we introduce voltage assignment, a technique that was originally used in connection with some topological problems in Graph Theory, but has recently found great application in other areas.

In its original form, voltage assignment takes a *base digraph* and a group to obtain a new, larger digraph. By dropping arc directions one can also construct undirected graphs; that is the way we will generally use it. Our theoretical framework is slightly different from the one that is customary in the literature: first we define the construction for natural walks (i.e. walks that follow arc directions), and then we extend the definition for arbitrary walks.

This section is divided into two subsections, the first one devoted to the definition and basic properties of the construction, and the second one to a short review of its connection with the Degree-Diameter Problem. For more detail we refer the reader to [77, 14, 28].

2.5.1 Definition and main properties

Let G be a digraph, Γ a finite group, and $\alpha : A \to \Gamma$ a labelling of the arcs with elements of Γ. The labels are usually called *voltages*, and α is a *voltage assignment*. Given G, Γ, and α, a new digraph G^α is constructed as follows: $V(G^\alpha) = V^\alpha = V \times \Gamma$, and $A(G^\alpha) = A^\alpha = A \times \Gamma$. If we have an arc $(u, v) \in A$ in G, with voltage $h \in \Gamma$, then (u, g) is joined to (v, gh) in G^α, for all $g \in \Gamma$. Quite often the vertex $(v, g) \in V^\alpha$ is written v_g. The digraph G^α is a topological covering of G, and it is commonly referred to as the the *derived digraph* or the *lift* of G by Γ and α. The digraph G is called the *base digraph*. A straightforward consequence of this definition is that both the number of vertices and the number of arcs of G^α are divisible by $|\Gamma|$. Figure 2.2 shows an example of a voltage assignment of a dipole, with voltages in $\mathbb{Z}_3$, and the resulting derived digraph.

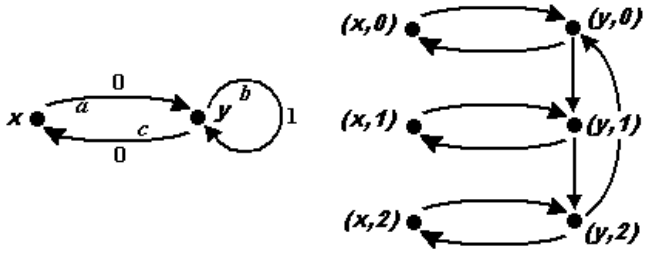


Figure 2.2: Voltage assignment of a dipole, with voltages in $\mathbb{Z}_3$, and the resulting derived digraph.

The function $\pi : G^\alpha \to G$ that erases the second coordinate of any vertex or arc of G^α, is called the *natural projection* of G^α onto G. The natural projection is defined for individual vertices or arcs of G^α, but it can be extended to subgraphs of G^α in two ways. The most obvious extension is the set-theoretic extension, i.e. $\pi(S) = \cup_{s \in S} \pi(s)$. Let us denote this extension by π_1. However, in some cases it is very useful to consider another extension π_2 of π, namely the one that allows repetition of elements.[4] For example, if we have a walk W in G^α, consisting of the arcs a_g, b_h, a_i, then $\pi_2(W) = a, b, a$, while $\pi_1(W) = \{a, b\}$. That is, $\pi_2(W)$ is also a walk in G, while $\pi_1(W)$ is just a set of arcs. If W is a natural walk in G^α, then $\pi_2(W)$ is also a natural walk (in G).
A lift of the walk $W = a_1, a_2, \ldots, a_k$, in G, is a walk $W' = a'_1, a'_2, \ldots, a'_k$ in G^α, such that $\pi_2(W') = W$. We say that W *lifts* to W'. On the other hand, we say that a walk W in G *expands* to a subgraph W' of G^α if $\pi_1(W') = W$, and W' is maximal with respect to that property. W' is the *expansion*[5] of W. In some sense, the lifting operation is the 'inverse' of π_2, and the expansion is the 'inverse' of π_1.
Given a walk $W = a_1, a_2, \ldots, a_k$, on G, whose arcs have voltages $g_1, g_2, \ldots, g_k$, the *net voltage* (or simply the *voltage*) of W, denoted $\alpha(W)$, is defined as the product $g_1^{e_1} g_2^{e_2}, \ldots, g_k^{e_k}$, where $e_i = 1$ if the corresponding arc a_i is traversed in the natural direction, and $e_i = -1$ if it is

[4] Most authors do not make any distinction between π_1 and π_2, but we believe that this omission may be a source of confusion.

[5] This should not be confused with other existing definitions of expansion graphs.

traversed in reverse direction. If W is a natural walk, then all the exponents are 1. With the aid of projection π_2 we can also define the net voltage of a walk W' on G^α, just as the net voltage of $\pi_2(W')$, which is a walk on G.
If x is a vertex (resp. an arc) of G, then $\pi_1^{-1}(x) = \{(x; g) : g \in \Gamma\}$ is called the vertex (resp. arc) *fibre* over x. There are several known facts concerning fibres:

1. If $a = (u, v)$ is an arc in G, then the fibre over a is a perfect matching between the fibre over u and the fibre over v (see 2.1.2 in [77], p 60).
2. The fibre over a loop is a set of cycles (Idem).
3. If the walk W starts at vertex $u \in G$, then for each vertex u_g in the fibre over u, there is a unique lift of W that starts at u_g (hence it can be denoted W_g). Moreover, if W ends at vertex v, and has net voltage h, then W_g terminates at v_{gh} (Theorems 2.1.1 and 2.1.2 of [77], p 62).

Now, given the set of voltages $X = \{x_1, x_2, \ldots, x_k\}$ on the arcs leaving a vertex v of G, and given an arbitrary $g \in \Gamma$, define $g\alpha$ to be the voltage assignment identical to α, such that the set U is modified as follows: $X = \{gx_1, gx_2, \ldots, gx_k\}$. The resulting derived graph $G^{g\alpha}$ is isomorphic to G^α. This implies that any voltage assignment α is equivalent to a voltage assignment $g\alpha$ with the identity element of G assigned to the arcs of any spanning tree of G. A voltage assignment a is said to be in *standard form* if a spanning tree of G is assigned the identity element of G. From the above results we conclude that for each voltage assignment α' we can find α in standard form. The relation defined by 'having the same standard form' is an equivalence relation on voltage assignments. It is possible, however, that two voltage assignments from two different equivalence classes will give rise to isomorphic lifts.

2.5.2 Cycle structure of derived digraphs

We now formalize the emergence of cycles in the derived digraph. The results presented in this subsection are usually taken for granted by the practitioners of topological graph theory, but are in no way trivial. In our opinion they have not been properly formalized in the literature.

The following lemma explains what happens to the cycles of G under voltage assignment.

Lemma 2.2 *Let C be a cycle of length k in G, with vertex sequence $u_0, u_1, u_2, \ldots, u_{k-1}, u_k = u_0$, and arc sequence $a_1 = (u_0, u_1), a_2 = (u_1, u_2), \ldots, a_k = (u_{k-1}, u_k)$. Let $g = \alpha(C)$ be the net voltage of C, and let n be the order of g in G. Then C expands to $\frac{|\Gamma|}{n}$ cycles of G^α, each of length kn. This result does not depend on the chosen initial vertex u_0.*

Proof See Theorem 2.1.3, and 2.1.5 of [77], p 63. □

This result can be slightly generalized to natural walks.

Lemma 2.3 *Let W be a natural closed walk of length k in G, with net voltage g, and let n be the order of g in Γ. Then C expands to $\frac{|\Gamma|}{n}$ natural closed walks of G^α, each of length kn.*

Proof The same reasoning as in the previous lemma. □

Now that we know that cycles of G are transformed into sets of cycles, and natural walks are transformed into sets of natural walks, a natural question arises: Does every cycle of G^α come from a cycle in G?. The answer to that question is given by the following

Lemma 2.4 *If C' is a cycle in G^α, then $\pi_2(C') = W$ for some natural closed walk W in G. That is, every cycle in G^α is the lift of some natural closed walk in G.* □

Corollary 2.1 *If C' is a cycle in G^α, then $\pi_1(C') = W$ for some natural closed walk W in G. That is, every cycle in G^α is also the expansion of some natural closed walk in G.* □

Lemma 2.5 *If W' is a natural closed walk in G^α , then $\pi_1(W')$ and $\pi_2(W')$ are natural closed walks in G. That is, every natural closed walk in G^α is the expansion of some natural closed walk in G, and also the lift of some natural closed walk in G.*

Proof The same reasoning as in Lemma 2.4 and Corollary 2.1. □

Lemma 2.6 *If C is a cycle in G^α then its net voltage is the identity element of G.*

Proof Let C' have length m, i.e, C' consists of the vertex sequence (u_0, g_0), (u_1, g_1), (u_2, g_2), $\ldots$, (u_{m-1}, g_{m-1}), $(u_m, g_m) = (u_0, g_0)$. By the definition of lift, the arc sequence of C' must have the form (a_1, g_0), (a_2, g_1), (a_3, g_2), $\ldots$, (a_m, g_{m-1}), where $a_1 = (u_0, u_1)$, $a_2 = (u_1, u_2)$, $a_3 = (u_2, u_3)$, $\ldots$, $a_m = (u_{m-1}, u_m)$, are arcs in G, and

$$\begin{aligned} g_1 &= g_0 \cdot \alpha(a_1) = g_0 \cdot x_1, \\ g_2 &= g_1 \cdot \alpha(a_2) = g_0 \cdot x_1 \cdot x_2, \\ &\vdots \\ g_{m-1} &= g_{m-2} \cdot \alpha(a_{m-1}) = g_0 \cdot x_1 \cdot x_2 \cdots x_{m-1}, \text{ and} \\ g_m = g_0 &= g_{m-1} \cdot \alpha(a_m) = g_0 \cdot x_1 \cdot x_2 \cdots x_m. \end{aligned}$$

Hence, $a_1, a_2, a_3, \ldots, a_m$, form a (natural) closed walk in G, and the net voltage $x_1 \cdot x_2 \cdots x_m$ of C' is the identity element of Γ. □

The converse is also true, as stated in the next lemma:

Lemma 2.7 *If W is a natural closed walk in G, with net voltage equal to the identity element of Γ, then W lifts to a natural closed walk W' in G^α.*

Proof It follows directly from Fact 3. □

A natural closed walk with net voltage equal to the identity element of Γ, is said to be *minimal* if it does not contain any proper closed sub-walk with net voltage equal to the identity. We have the following:

Theorem 2.5 *Every cycle C' of G^α is the* ***lift*** *of some minimal natural closed walk W in G, such that $\alpha(W)$ = identity of Γ.*

Proof By Lemma 2.4, C' is the lift of some natural closed walk W in G, and by Lemma 2.6, the net voltage of W is the identity element. Now let us suppose that W is not minimal. Then W contains some proper closed sub-walk with net voltage equal to the identity. By Lemma 2.7, this sub-walk must also lift to some natural closed walk, which is a proper sub-walk of C', a contradiction since C' is a cycle. Hence W must be minimal. □

In order to study certain properties of G^α, such as bipartiteness or planarity, the orientation of arcs is not important: we are interested to view G^α as a graph, not as a digraph. Hence, in those cases we must not restrict ourselves to cycles, or natural walks, but consider more general walks, where arcs can be traversed in the reverse direction. That is when the group structure of voltages comes into play.
With the aid of a simple trick we can easily extend all the previous results to general walks: Before we are about to traverse an arc, we check whether the direction of the intended path coincides with the orientation of the arc to be traversed; if not, then we can reverse the orientation of the arc, and replace its voltage by its inverse. By convention, we cannot reverse the orientation of an arc when we are already traversing it, or when we are at one of its end-vertices; the orientation has to be changed *before* arriving at the arc. This means that we cannot backtrack on the same arc that we have just traversed. With this convention, all the walks are natural walks, and all the previous results hold without modifications. Therefore, from now on we do not have to establish any distinction between general and natural walks. Note also that a cycle is now an undirected cycle.
As a consequence of Theorem 2.5, in order to enumerate all cycles of G^α we must enumerate all minimal closed walks in G, with net voltage equal to the identity element. However, from a computational viewpoint this is not trivial. We would like to relate cycles of G^α with cycles of G, that can be enumerated more easily.
Now let us see what happens when the voltage group is an (external) direct product of the form $\Gamma \times \Lambda$. For the sake of simplicity, we shall denote the operation in both factors with the same symbol "$\cdot$". Their identity elements will be denoted 1_Γ and 1_Λ, respectively, but we can use the same symbol 1 whenever it creates no confusion. If we have an arc $(u; v) \in A$ in G, with voltage $\langle h, i\rangle \in \Gamma \times \Lambda$, then $(u; \langle g, l\rangle)$ is joined to $(v; \langle gh, li\rangle)$ in G^α, for all $\langle g, l\rangle \in \Gamma \times \Lambda$. Obviously, all the properties studied so far for an arbitrary voltage group also hold for $\Gamma \times \Lambda$. Now let $\alpha : A \to \Gamma \times \Lambda$ be a voltage assignment. The function $\pi_\Gamma(\alpha) = \alpha_\Gamma$, defined as $\alpha_\Gamma(a) = \langle g, 1\rangle$, if $\alpha(a) = \langle g, l\rangle$, will be called the *restriction* or *projection* of α to Γ. The restriction $\pi_\Lambda(\alpha) = \alpha_\Lambda$ is defined in a similar fashion.
We have the following

★ **Proposition 2.1** *Let W be a minimal natural closed walk in G. Suppose that W expands under α_Γ to a cycle C_1 of length L_1, and expands under α_Λ to a cycle C_2 of length L_2. Then, W expands under α to a cycle C of length $L = lcm(L_1, L_2)$.*

Proof Let $\langle g, l\rangle$ be the net voltage of W. By Theorem 2.5 we have $g^{L_1} = 1_\Gamma$, and $l^{L_2} = 1_\Lambda$. Hence, L is the least number such that $\langle g, l\rangle^L = \langle g^L, l^L\rangle = \langle 1, 1\rangle$, which is $\mathrm{lcm}(L_1, L_2)$. □

Lemma 2.1 can of course be generalized to arbitrary direct products:

★ **Proposition 2.2** *Let $\Gamma = \Gamma_1 \times \Gamma_2 \times \cdots \times \Gamma_m$, and let W be a minimal natural closed walk in G. Suppose that W expands under α_{Γ_i} to a cycle C_i of length L_i, for $i = 1, \ldots, m$. Then, W expands under α to a cycle C of length $L = lcm(L_1, \ldots, L_m)$.* □

From the preceding discussion we have seen that if we want to control the length of cycles in the derived graph G^α, we should be able to predict the order of elements in Γ, that are the net voltages of cycles in G. This is in general a difficult group-theoretical question. In Section 5.2 we address the problem for the case when Γ is a metabelian group.

2.5.3 Constructing large graphs by voltage assignment

Voltage assignment has been the most successful technique so far for obtaining large graphs of given degree and diameter: It is responsible of about 60% of the largest known graphs of the table [63]. Therefore, this application of voltage assignment merits a separate subsection here.

Voltage assignment has been used in a variety of ways, ranging from ad hoc constructions, to general constructions, and in combination with computer search. For example, in a classic paper, McKay, Miller and Širáň describe a family of large vertex-transitive graphs with $\Delta = (3q-1)/2$, where q is a prime power congruent with 1 (mod 4), though the construction generalizes to all prime powers [112]. The base graphs used in [112] are complete bipartite graphs with loops. Later, Šiagiová showed that the McKay-Miller-Širáň graphs can also be obtained as lifts of dipoles [147].

Other authors played for some time with the possibility of using voltage assignment for obtaining large graphs (see [14], [28], [155]), until a breakthrough occurred in 2008, when Loz and Širáň published a large number of new record graphs found with the aid of voltage assignment and computer search [107]. Their algorithm explores a search space consisting of lifts of simple base graphs (bouquets, dipoles, etc.) by non-abelian groups. Their groups were taken among the ones suggested by Dinneen and Hafner for constructing large Cayley graphs [51], namely semidirect products of cyclic groups, such as $\mathbb{Z}_m \rtimes_r \mathbb{Z}_n$, $(\mathbb{Z}_m \times \mathbb{Z}_m) \rtimes_\psi \mathbb{Z}_n$, and $(\mathbb{Z}_m \rtimes_r \mathbb{Z}_n) \rtimes (\mathbb{Z}_m \rtimes_r \mathbb{Z}_n)$. Those were precisely the groups that were used later in [106] to enlarge the table of the largest known graphs up to degree 20 [63].

Some recent results show that in order to obtain good graphs by voltage assignment, the underlying group should be as far as possible from abelian [104, 60, 62]. Actually, this had already been noted by Dinneen and Hafner in [51], and even though the semidirect product of cyclic groups are not very far from abelian, they have the advantage that they are easy to generate and compute with, and they produced good results (probably because they were the first non-abelian groups that were tried).

Indeed, the method of [107] can be seen as a generalization of [51] inasmuch as Cayley graphs can be obtained as lifts of bouquets. The algorithm used by Loz and Širáň is a basic *random search*, combined with a number of clever *sieves*, designed to prune the search space. The algorithm starts with a base graph and a certain family of groups, and then iterates through the groups, generating random voltage assignments, and computing the diameter of the corresponding lift in each case. The sieves control the prospective voltage groups, the number of trials to be made with each group, and further computations with the derived graph.

The first sieve consists of sorting out the groups that do not meet certain criteria, like having a small center, for instance. Then, the *escalation sieve* controls the number of random trials

to be made with each voltage group. Initially, it assigns a predefined number of trials to each group, and then, the number of trials is modified during the search process, according to the prospects of finding a good derived graph with the given group. Finally, the *girth sieve* abandons the computation of the diameter, if the girth is found to be smaller than a predefined threshold.
Algorithm 2 describes the basic random search structure of this procedure, to which the sieves can be plugged in. Some more details concerning the sieves and the implementation of the algorithm are given in [103], but no detailed pseudo-code description of the algorithm has been published so far.

Algorithm 2: Basic random search

Input : A base graph G, and a family of groups Ω.
Output: A set (possibly empty) of large graphs, that are lifts of G by $\Gamma \in \Omega$.

```
1  Initialize the max. number of trials, m_{d,k};
2  Choose a BFS spanning-tree T of G, and label the arcs of T with the generic identity
   element;
3  foreach unexplored group Γ ∈ Ω do
4  |  for i:= 1 to m_{d,k} do
5  |  |  Generate a random voltage assignment α;
6  |  |  Compute the girth and diameter of G';
7  |  |  if diameter ≤ k then
8  |  |  |  Return Γ and α;
9  |  |  |  Break;
10 |  |  end
11 |  end
12 end
```

Recent progress in finding large graphs with given degree and diameter via voltage assignment includes [148, 110, 109].

Finally, it is worth mentioning that the voltage assignment technique has also been used to find other types of graphs, like cages [59, 103, 105]. In particular, the paper [59] explores some interesting methods to find the appropriate voltage group and voltage assignment. That paper contains the seeds of a theory that was developed later in [62]. For a comprehensive account see [61].

3

Graphs, networks, and algorithms

In this chapter we expand on the basic notions of graph theory, focusing on the algorithmic approach, and applications in network design. In particular, we introduce the Degree-Diameter Problem and its generalization, MaxDDBS , which make up the main subject of the thesis. In connection with these two problems we review the relevant literature. For completeness, we also review some important concepts and trends in algorithm design, which will be used later.

3.1 Graphs, revisited

This section contains some additional background material for graph theory, namely the notions that are more directly related to network design and the Degree-Diameter Problem.

3.1.1 Operations on graphs

Graph operations are a way to obtain new graphs from existing ones. The simplest graph operations are just generalizations of the set-theoretical operations (set union, intersection, etc.). For the sake of convenience, we keep the same symbol in graph and set operations. For instance, the *union* of G and G', denoted $G \cup G'$, is the graph with vertex set $V(G) \cup V(G')$ and edge set $E(G) \cup E(G')$. Graph *complement*, *intersection*, and *difference*, can be defined in a similar way.
The *join* $G + G'$ is obtained from $G \cup G'$ by adding all possible edges from vertices in G to vertices in G'.
The *cartesian product* $G \times G'$ has vertex set $V(G) \times V(G')$, and (v_i, w_j) is adjacent to (v_h, w_k) if either

- v_i is adjacent to v_h in G, and $w_j = w_k$, or

- $v_i = v_h$, and w_j is adjacent to w_k in G'

In the next section we see some examples of cartesian product of graphs.
The *line graph* of G, denoted $L(G)$, is the graph with vertex set E, in which $e, f \in E$ are adjacent as vertices of $L(G)$ if, and only if, they are adjacent as edges in G.

Another useful operation is *graph compounding*: Given a graph G, we replace every vertex $v \in G$ with a graph H_v, and all the edges incident with v are then joined to chosen vertices of H_v. Note that there may exist different ways of joining v to the vertices of H_v, hence the result of this operation is essentially ambiguous; different authors have adopted different compounding methods in order to meet their specific purposes. Here we have tried to encompass most of the graph compounding techniques that are used in practice.
Let $S = \{H_1, H_2, \ldots, H_k\}$ be a set of graphs. Each element of S is called a *source graph*. Now let G be another graph (called the *base graph*), and let $\hat{G}$ be a subgraph of G such that $E(\hat{G}) = \emptyset$, and $V(\hat{G})$ is formed by all those vertices of G to be replaced during the compounding process. We call $\hat{G}$ the *replaced graph*. Finally, let f be a mapping from $V(\hat{G})$ to S.
The compounding of S into G, denoted as $G(S)$, or $G(H_1, H_2, \ldots, H_k)$, is defined by means of the following two steps:

Step 1: Every vertex $x \in V(\hat{G})$ is replaced by the graph $f(x) \in S$. The set of added vertices is denoted by $\hat{V}(S)$; that is, $\hat{V}(S) = \bigcup_{x \in V(\hat{G})} V(f(x))$.

Step 2: The edges that were incident with $x \in V(\hat{G})$ are now distributed among the vertices of $f(x)$. This is the ambiguous step, for it can be done in several ways.

It is easy to see that

$$|G(S)| = |G| + \sum_{x \in V(\hat{G})} |f(x)| - |\hat{G}|.$$

To exemplify this, see Figure 3.1, where $S = \{C_3, C_4, K_4\}$ and the base graph is K_4. The replaced graph has vertex set $V(\hat{G}) = \{a, b, d\}$ and edge set $E(\hat{G}) = \emptyset$, and the mapping is $f(a) = C_3$, $f(b) = C_4$, and $f(d) = K_4$.

3.2 Basic network topologies

In this section we review some special families of graphs that appear frequently as network topologies in parallel and distributed computing. Besides their practical application, they have some interesting properties, and they will be the basis of subsequent chapters of this thesis. We start with the hypercube and its variants.

The *$d-$dimensional cube* or *hypercube* Q_d is the graph whose vertex set is the set of all binary $d-$tuples, and two tuples are joined by an edge iff they differ in exactly one position. Obviously, Q_d has order 2^d and is $d-$regular. It can be defined recursively as

$$\begin{aligned} Q_2 &= K_2 \\ Q_d &= Q_{d-1} \times K_2 \quad \text{for } d > 2. \end{aligned}$$

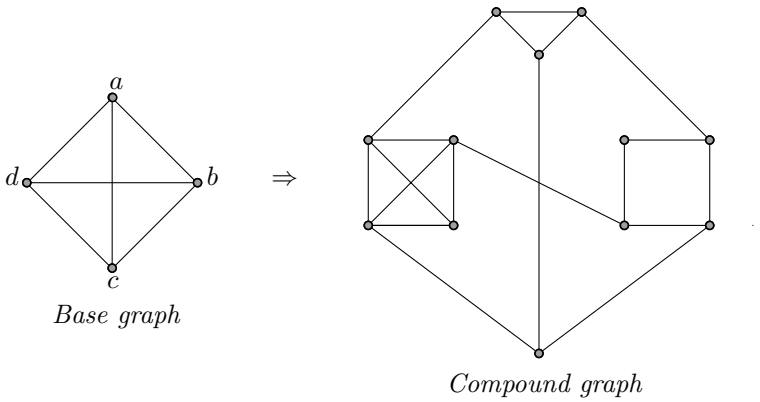


Figure 3.1: Compound graph.

Two different drawings of Q_4 can be seen in Figure 3.2.

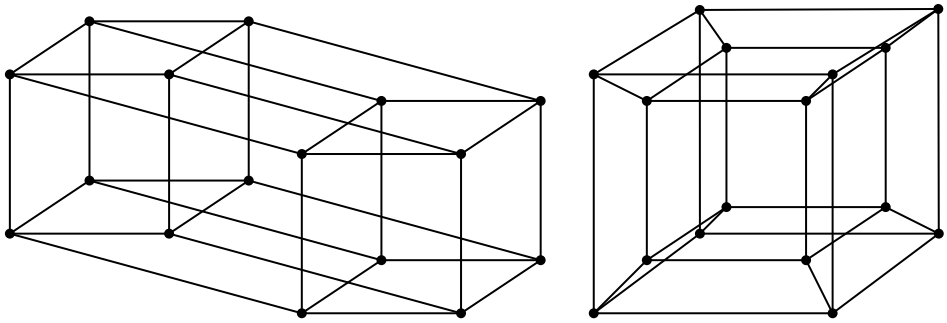


Figure 3.2: Two different representations of the 4-cube

Q_d has $2^{d-1}d$ edges and diameter k. It is bipartite and vertex-transitive for all $d \geq 1$, and hamiltonian for $d \geq 2$. Actually, Q_d is the Cayley graph of the abelian group $\mathbb{Z}_2^d$. While the hypercube has played an important historical role in the development of supercomputers (see [80], for example), it has a significant drawback: the degree of each node is too high (d), and it becomes impractical for large d. Therefore it has been superseded by some variants that are more *scalable*.

One of these variants of the hypercube is the *Cube-Connected Cycles*, or CCC, for short [129]. In particular, the cube-connected cycles of dimension d, denoted here CCC_d, are constructed by replacing each vertex v of the d-dimensional hypercube Q_d by a cycle of length d, consisting of new vertices $v_1, \ldots, v_d$, and then joining each v_i to one of the edges that was incident to the former vertex v. In other words, the graph CCC_d is constructed by compounding the cycle C_d into Q_d. Figure 3.3 depicts CCC_3.

The main advantage of CCC_d over Q_d is that CCC_d is always regular of degree 3, regardless of the dimension d, which makes it highly *scalable*. This comes at the price of increasing the diameter from d to $2d$, but it is still logarithmic as a function of the number of vertices.

A d-dimensional *butterfly* has $(d+1)2d$ nodes corresponding to pairs (w, i) where w is a d-bit binary number and $0 \leq i \leq d$ is the level. Two nodes $(w; i)$, $(w'; i')$ are linked iff $i' = i+1$ and

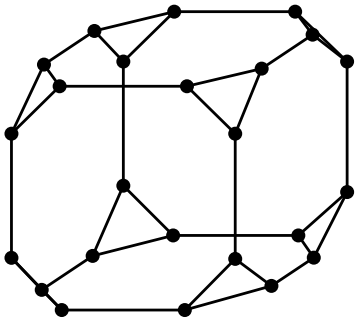


Figure 3.3: Cube-Connected Cycles of dimension 3

$w = w'$ or w and w' differ only in the i'-th bit. This makes $2^{d+1}d$ edges. Figure 3.4 depicts a butterfly of dimension 3. The butterfly is suitable for certain computations, like the Fast Fourier Transform (FFT).

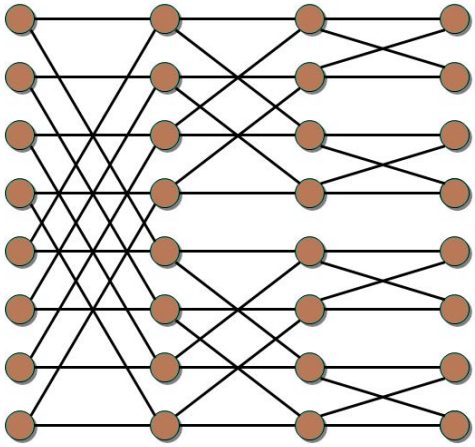


Figure 3.4: A butterfly of dimension 3

If we take two butterflies of dimension d and identify the vertex (w, d) of the first butterfly with the vertex (w, d) of the second butterfly, for w ranging among all possible d-bit binary numbers, we get the *Beneš network*. Figure 3.5 depicts a Beneš network of dimension 3.

Another important architecture (at least historically) is the *mesh*. An infinite *$d-$dimensional mesh* or *grid* is the graph M_d whose vertex set is the set of all $d-$tuples with integer coordinates, and two tuples are joined by an edge iff they are at distance one. The infinite $d-$dimensional mesh is the Cayley graph of the free abelian group $\mathbb{Z}^d$.

If we restrict the i-th coordinate to take values in the interval $[a_i, b_i]$, then we get a finite grid. By keeping all coordinates fixed, except the i-th coordinate, which is allowed to move freely in $[a_i, b_i]$, we get a row of the finite grid, with endvertices at $x_i = a_i$ and $x_i = b_i$. If we join the two endvertices of every row, we get a *toroidal grid*. An n by m toroidal grid is the Cayley graph of the group $\mathbb{Z}_n \times \mathbb{Z}_m$. Figure 3.6 depicts a 5×5 square toroidal grid.

Chapter 6 is devoted to the study of the grid, and in Section 7.1 we will see another type of

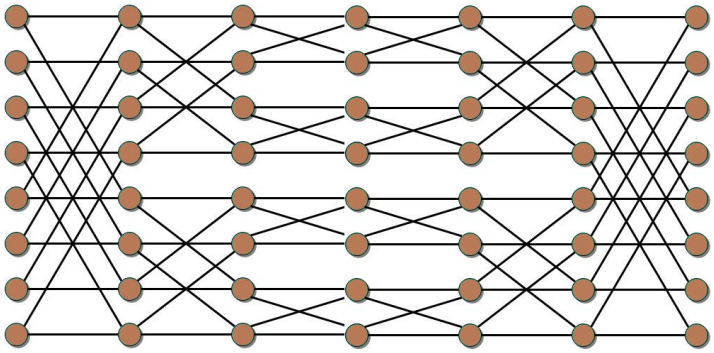


Figure 3.5: Beneš network of dimension 3

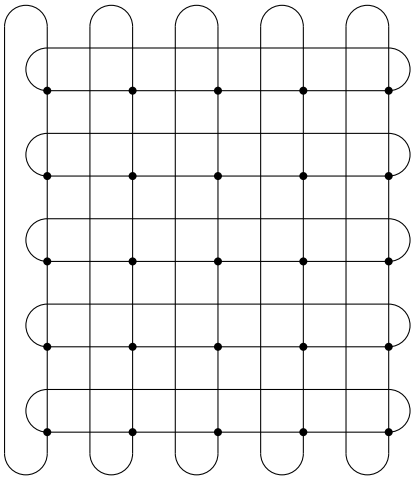


Figure 3.6: Square toroidal grid

grid: the hexagonal, or honeycomb grid.

3.3 Random graphs

In many practical situations we face graphs that are random in nature. The study of random graphs was started by Erdös and Rényi in the late 50's and early 60's, and they developed the first model of random graphs, that today bears their name. The *Erdös-Rényi random graph* $G(n,p)$ is a graph with n vertices, where each edge is chosen at random, independently from the others, with a probability p.

Other random graph models were developed later to describe real-life networks more accurately. For example, in the *Watts-Strogatz small world model* we start with a regular graph, and then rewire each edge, by re-connecting one end to another vertex chosen at random, with probability p. As p approaches 1, this construction behaves approximately like $G(n,p)$, but for intermediate values of p it has new interesting properties. The name 'small world' comes from the fact that the resulting graphs have low diameter.

Another popular model is the *Barabási-Albert preferential attachment model*, where new vertices are added, one by one, and connected at random to the existing vertices, with a probability that is proportional to the degree of each existing vertex. This process yields a graph where the degrees have a power-law distribution, whence the alternative name *scale-free graph*, for its properties remain invariant with changes of scale. The Barabási-Albert preferential attachment model and its variants describe many real-life networks quite accurately, such as the World-Wide Web, the scientific collaboration network, and others.

A good source of information about these basic random graph models and their variants is [55].

3.4 Moore graphs and the Degree-Diameter Problem

At this point it is natural to start asking questions about the interplay among different graph parameters. One such question is the *Degree-Diameter Problem* (or DDP, for short), which can be stated as:

Problem 3.1 (Degree-diameter problem for undirected graphs) *Given positive integers Δ and D, find the largest possible number of vertices $N_{\Delta,D}$ of a graph of maximum degree Δ and diameter D.*

It is easy to show that an upper bound for $N_{\Delta,D}$ is

$$M_{\Delta,D} = 1 + \Delta + \Delta(\Delta-1) + \cdots + \Delta(\Delta-1)^{D-1} = \begin{cases} 1 + \Delta\,\frac{(\Delta-1)^D - 1}{\Delta - 2} & \text{if } \Delta > 2 \\ 2D+1 & \text{if } \Delta = 2 \end{cases} \tag{3.1}$$

The number $M_{\Delta,D}$ is called the *Moore bound*, and a graph of order $M_{\Delta,D}$ is called a *Moore graph* [119]. Note that such a graph is necessarily regular of degree Δ.

Moore graphs exist only for a few combinations of Δ and D. For $D = 1$ and $\Delta \geq 1$, they are the complete graphs on $\Delta + 1$ vertices. For $D \geq 2$ and $\Delta = 2$, they are the cycles on $2D + 1$ vertices. For $D = 2$, Moore graphs exist for $\Delta = 2, 3, 7$, and possibly 57 [84]. There are no other Moore graphs.
The same problem can be stated for digraphs:

Problem 3.2 (Degree-diameter problem for directed graphs) *Given positive integers Δ and D, find the largest possible number of vertices $N^*_{\Delta,D}$ of a digraph of maximum out-degree Δ and diameter D.*

In this case, the Moore bound is given by

$$M^*_{\Delta,D} = 1 + \Delta + \Delta^2 + \cdots + \Delta^D = \frac{\Delta^{D+1} - 1}{\Delta - 1} \tag{3.2}$$

Moore digraphs exist only for the trivial cases: For $\Delta = 1$ they are the directed cycles with $D + 1$ vertices, and for $D = 1$ they are the complete digraphs on $\Delta + 1$ vertices.
For more information about the Degree-Diameter Problem, both in the directed and the undirected case, see [119].

3.5 Lower bounds

In this section we give a brief systematic account of methods for constructing large graphs with given degree and diameter. More detail can be found in [119, 126]. However, there is some material on lower bounds that is not systematically covered anywere, such as computer-based methods for constructing large graphs.

All research related to the Degree-Diameter Problem can be roughly classified in one of the following categories:

- **Upper bounds**: Given that the Moore bound can only be attained for a few combinations of degree and diameter, it is interesting to know what is the actual largest number of vertices that a graph can have, for other combinations of Δ and D. In other words, this line of research seeks to obtain tighter theoretical upper bounds for all combinations of Δ and D.
- **Lower bounds**: This second line of research, of more practical importance perhaps, is related with the construction of ever larger graphs, with order as close as possible to the theoretical upper bound.

The ideal situation is that the lower and upper bounds coincide, but that situation is again very rare for the moment. Only for a few combinations of degree and diameter, the order of the largest known graph matches the theoretical upper bound, while in the vast majority of cases, there is a gap between the lower and the upper bound, which varies according to Δ and D.
This situation is especially evident in the case of undirected graphs, where the gap is considerably large for most combinations of Δ and D. In the case of directed graphs the situation

is more favourable, since there exist constructions that yield graphs with order close to the theoretical upper bound for all combinations of Δ and D.
Upper bounds are treated in great detail in [126], and also surveyed in [119]. In this thesis we will mainly be concerned with lower bounds. The current knowledge about lower bounds is summarized in [63], that contains tables for the largest known general graphs, the largest known Cayley graphs, and the largest known bipartite graphs, for degrees 3 to 16, and diameters 2 to 10. The table of general graphs has been recently extended to degree 16 in [106]. The website http://combinatoricswiki.org/wiki/The_Degree/Diameter_Problem maintains tables of the best known upper bounds, and the percentage of the largest known graph orders in relation to those upper bounds. Dr. Eyal Loz has been the driving force behind this compilation, that continues the tradition started by Comellas [34], and is part of a wider project, which will also include tables of the largest known vertex-transitive and planar graphs, as well as information about other combinatorial problems.
Back to lower bounds, all the efforts to construct large graphs can in turn be roughly classified into two main categories:

- **Analytic constructions**: Graph operations, like compounding and voltage assignment, provide the means for building large graphs from smaller base graphs or blocks. A judicious choice of the building blocks and operations on them can yield large graphs for certain combinations of Δ and D. This is the principle that has been followed in [71, 127] (with graph compounding), and [112] (with voltage assignment), for instance.
- **Computer-based techniques**: This second approach relies on the power of electronic computers in order to find larger graphs, either using the standard combinatorial optimization techniques (as in [153]), or graph operations, like voltage assignment, coupled with computer search (as in [107]).

In terms of the number of graphs produced, the most successful techniques are graph compounding, polarity graphs of generalized polygons, and voltage assignment, the latter in conjunction with computer search. Of the 126 entries in the current table of largest known graphs, 67 have been found by voltage assignment (53%), while 19 have been found with the aid of graph compounding (15%), and 15 are somehow related to polarity graphs of generalized polygons (12%). Finally, 11 entries have been found by other computer-based search techniques (9%). Together, all these techniques account for 89% of the table.

3.6 Analytic constructions

In [126], constructions have been classified as

- **general constructions** (the ones that yield a large, usually infinite, class of graphs), and
- **ad hoc constructions**, where at least one of the parameters Δ and D is restricted to a few values.

General analytic constructions provide lower bounds for the order of the largest graphs, for many combinations of Δ and D. We review de Bruijn and Kautz graphs, as two relevant

examples of general constructions, and the bounds that they provide. Then we focus on graph compounding, which in principle is a general technique, but is usually applied as an ad hoc method. Finally, we review other ad hoc constructions.

3.6.1 Some general constructions

The *de Bruijn graph of type* (t, k) [40] is a graph $G = (V, E)$, such that V is formed by all words of length k over a finite alphabet A with cardinality t, and two vertices $a = (a_1, a_2, \ldots, a_k)$ and $b = (b_1, b_2, \ldots, b_k)$ are joined by an edge if either $a_i = b_{i+1}$ or $a_{i+1} = b_i$, for $1 \leq i \leq k-1$. In other words,

$$(a_1, a_2, \ldots, a_k) \sim \begin{cases} (a_0, a_1, \ldots, a_{k-1}) & \text{for any } a_0 \in A \\ (a_2, a_3, \ldots, a_{k+1}) & \text{for any } a_{k+1} \in A \end{cases}$$

For $t \geq 3$ and $k \geq 3$, the de Bruijn graph of type (t, k) has order t^k, diameter $D = k$ and maximum degree $\Delta = 2t$. That gives us a lower bound for the size of the largest graph with maximum even degree Δ, and diameter D:

$$N_{\Delta,D} \geq \left(\frac{\Delta}{2}\right)^D \tag{3.3}$$

Given a de Bruijn graph of type (t, k), a *Kautz graph of type* (t, k) is obtained by deleting words with two consecutive identical letters [91]. The Kautz graph is therefore an induced subgraph of the de Bruijn graph, and if $t \geq 3$ and $k \geq 3$, it has order $t(t-1)^{k-1}$, diameter k and maximum degree $2t - 2$. Again, if Δ is even we get the bound:

$$N_{\Delta,D} \geq \left(\frac{\Delta}{2}\right)^D + \left(\frac{\Delta}{2}\right)^{D-1}$$

which is an improvement over (3.3).
A further improvement of the previous bounds was obtained by Canale and Gómez [29]. They proved that there are two constants D_0 and $\alpha < 1.59$ such that, for each $D \geq D_0$ and infinitely many values of Δ, it is possible to construct a $(\Delta, D)-$graph of order $(\frac{\Delta}{\alpha})^D$.

3.6.2 Compounding

As mentioned in Section 3.5, graph compounding is one of the most successful techniques for the construction of large graphs of given maximum degree and diameter. Let us recall that the technique consists of replacing one or more vertices of a graph with elements of a set of graphs S, and then the edges of the resulting graph are rearranged appropriately. The idea is straightforward, the only drawback being the calculation of the diameter, which depends on the choice of the edges connecting the elements of S. The technique was introduced by Bermond, Delorme and Quisquater [18], and later it has been systematically used by other authors, either alone or in combination with other methods (see, for example, [16, 17, 29, 36, 45, 46, 47, 65, 70, 71, 73, 74, 75, 76, 130]).
In particular, compounding of complete graphs into bipartite Moore graphs has been used recently with good results. It was first suggested in [130], where a single vertex from a bipartite

Moore graph is replaced with a suitable complete graph. In a bipartite Moore graph it is even possible to replace several vertices with copies of suitable complete graphs without increasing the diameter of the bipartite Moore graph. This idea was first used in [74]. Subsequently, using bipartite Moore graphs of diameter 6, some improvements of this approach have been achieved in [36, 76], and more recently in [71, 127]. The results of [127] are the subject of Chapter 4.

3.6.3 Other ad hoc constructions

Many of the largest known graphs for small values of D have been obtained as graphs of incidence structures, or modifications thereof. Incidence geometries, and their applications to the construction of large graphs, have also been extensively reviewed in [126], therefore we will reproduce here only the main concepts and the most outstanding results.

An *incidence structure* is a triple $\Omega = (\mathcal{P}, \mathcal{L}, \mathcal{I})$, where $\mathcal{P} \neq \emptyset$ is a set of *points*, $\mathcal{L} \neq \emptyset$ is a set of *lines*, $\mathcal{P} \cap \mathcal{L} \neq \emptyset$, and $\mathcal{I} \subseteq \mathcal{P} \times \mathcal{L}$ is a relation, called the *incidence relation*. Given a point p and a line l, if $(p, l) \in \mathcal{I}$ then we say that p and l are *incident*. If $\mathcal{P}$ and $\mathcal{L}$ are finite sets, $\Omega = (\mathcal{P}, \mathcal{L}, \mathcal{I})$ is said to be finite.

Two points are *collinear*, if they are incident with at least one common line. Two lines are *concurrent*, if they share at least one point. Given an incidence structure Ω, if each line is incident with $s+1$ points, and each point is incident with $t+1$ lines, we say that Ω has *order* (s, t). If $s = t$ then Ω is said to have order s.

A number of different graphs can be derived from an incidence structure. For instance, let $\Omega = (\mathcal{P}, \mathcal{L}, \mathcal{I})$ be an incidence structure. The *incidence graph* G of Ω is the graph with vertex set $V(G) = \mathcal{P} \cup \mathcal{L}$, and $(x, y) \in E(G)$ if, and only if, $x\mathcal{I}y$ or $y\mathcal{I}x$, for $x, y \in V(G)$.

A *generalized D-gon* is an incidence structure whose incidence graph is a bipartite graph of diameter D and girth $2D$. It is common to use the standard geometric names for small polygons, i.e. generalized triangle instead of generalized 3-gon.

Now let $\Omega = (\mathcal{P}, \mathcal{L}, \mathcal{I})$ be an incidence structure with a polarity ω. We can now define the *polarity graph* of Ω with respect to ω, denoted G^ω, as the graph with vertex set $V(G) = \mathcal{P}$, and the following adjacency relation: $pp_1 \in E(G^\omega)$ if $p \neq p_1$ and $(p, \omega(p_1)) \in \mathcal{I}$.

For small values of the diameter D, some of the largest known graphs turn out to be precisely the polarity graphs, or modified polarity graphs, of the generalized D-gons. For instance, in the current table of largest known graphs [63], the entries (8,2), (10,2), (12,2), and (14,2), correspond to polarity graphs of generalized triangles. The graph in (9,2) is a modification of that in (8,2), while (15,2) is a modification of (14,2).

In diameter 3, the graph in (9,3) is a polarity graph of the generalized quadrangle of order 8, and (10,3), (11,3), (12,3), (14,3), (15,3), and (16,3), are modified polarity graphs. Finally, (4,5) and (16,5) are also derived from polarity graphs.

One of the most common operations applied to polarity graphs to obtain other large graphs is *vertex duplication*. Given a graph $G = (V, E)$ with maximum degree Δ, and diameter D, and given a vertex $x \in V$, the operation of duplicating x consists of adding a new vertex x', and joining it to x and all the neighbours of x. Obviously, the resulting graph will have maximum degree $\Delta + 1$ and diameter D. The most recent graphs constructed that way correspond to the entries (12,3), (13,3), and (14,3) (see [72]).

3.7 The Degree-Diameter Subgraph Problem

Given the time that the Degree-Diameter Problem has been investigated, it is quite surprising that the following straightforward generalization has not been considered before:

Problem 3.3 (Degree-Diameter Subgraph Problem) *Given a connected undirected host graph G, an upper bound Δ for the maximum degree, and an upper bound D for the diameter, find the largest connected subgraph S with maximum degree $\leq \Delta$ and diameter $\leq D$.*

In the context of computational complexity we will also refer to this problem as the *Largest*, or the *Maximum Degree & Diameter Bounded Subgraph problem*, in analogy with other well-known related algorithmic problems, such as the *Bounded Diameter Spanning Tree*, or the *Degree-Bounded Connected Subgraph*. For the sake of brevity we will use the acronyms DDS and MaxDDBS.

This problem is known to be $\mathcal{NP}-$hard, since it contains other well-known $\mathcal{NP}-$hard problems as subproblems. Actually, restricting the search to only one constraint (either on the degree or the diameter), is enough to ensure $\mathcal{NP}-$hardness. The Largest Degree-Bounded Subgraph Problem is $\mathcal{NP}-$hard as long as we insist that the subgraph be connected, but can be solved in polynomial time otherwise (see [89] and Problem ND1 of [68]). On the other hand, the Maximum Diameter-Bounded Subgraph becomes the Maximum Clique for $D = 1$, which was one of Karp's original 21 $\mathcal{NP}-$hard problems (see [90] and Problem GT19 of [68]).

Obviously, if we take G as the complete graph on $M_{\Delta,D}$ vertices in Problem 3.3, we get the Degree-Diameter Problem. Note that this does not imply that DDP is $\mathcal{NP}-$hard; actually, the complexity of DDP is not known to-date. The relationship between MaxDDBS and DDP can be exploited to get benchmarks for MaxDDBS. The Moore bound is also a theoretical upper bound for MaxDDBS, and the existing lower bounds constitute a benchmark to measure the performance of any algorithm attempting to solve MaxDDBS. A table of the largest known graphs for all $3 \leq \Delta \leq 20$, and all $2 \leq D \leq 10$ can be found in [63].

Now, in practical applications we often have weights or costs attached to the edges. We can generalize the definitions of distance and diameter to the case of weighted undirected graphs in a very straightforward manner:

Definition 3.1 *Let $G = (V, E)$ be a weighted connected undirected graph, with positive integral weights on the edges, and let $\omega(e)$ be the weight of the edge $e \in E$. The length of a path $e_1 e_2 \dots e_k$ is defined as $\sum_{i=1}^{k} \omega(e_i)$. The distance between two vertices $u, v \in V$ is the length of a shortest path between u and v, and the diameter of G is the distance between two vertices u and v that are farthest apart.*

This leads us to a more realistic formulation of Problem 3.3:

Problem 3.4 (MaxWDDBS) *Let G be a weighted connected undirected host graph, with positive integral weights on the edges, and suppose we are given an upper bound Δ for the maximum degree, and an upper bound D for the diameter. Find the largest connected subgraph S of G, with maximum degree $\leq \Delta$ and diameter $\leq D$.*

In the botnet and distributed computing scenarios described in Section 1.2, the weights could represent the time units that a message takes to get from one node to an adjacent one. Having integral weights does not imply any loss of generality here, since we can always use a time unit that is small enough, so that all weights are multiples of it, or the remainder is negligible. Moreover, we can also assume without loss of generality that Δ does not exceed the maximum degree of G.

The following relationship between MaxDDBS and MaxWDDBS is also straightforward:

Proposition 3.1 *Let G be a connected undirected graph, with positive integral weights on the edges, and let G' be the same graph with all the weights replaced by 1. Now, let Δ and D be given, and let S, with n vertices, be the solution of* MaxWDDBS *on G, and let S', with n' vertices, be the solution of* MaxDDBS *on G'. Then, $n \leq n' \leq N_{\Delta,D} \leq M_{\Delta,D}$.* □

Needless to say, MaxWDDBS is also $\mathcal{NP}-$hard, and therefore amenable to heuristic approaches. In Section 8.1 we investigate a heuristic algorithm to solve it, and Section 3.8 gives an overview of different heuristic techniques.

3.8 Algorithmic techniques

In this section we deal with some computer-based techniques for solving intractable combinatorial problems, such as DDP. Computer approaches range from systematic search techniques to heuristic methods, such as genetic algorithms and simulated annealing, which are now commonplace in Combinatorial Optimization. A *combinatorial optimization problem* consists of an *objective function* to be minimized or maximized, $f : D_1 \times \cdots \times D_k \to \mathbb{R}^+$, and a set of constraints among the variables $x_i \in D_i$. A feasible solution is a $k-$tuple $(x_1, \ldots, x_k)$, such that $x_i \in D_i$ and all constraints are satisfied. The set $\mathcal{S}$ of all feasible solutions is often called the *search space*, and it is usually endowed with a *neighborhood structure*, which is a function $\mathcal{N} : \mathcal{S} \to 2^{\mathcal{S}}$ that assigns a set of neighbours to every feasible solution $s \in \mathcal{S}$.
Combinatorial optimization problems include the famous Travelling Salesman Problem (TSP – Problem ND22 of [68]), the $0-1$ Knapsack Problem (Problem MP9 of [68]), various scheduling problems, and so on. In particular, MaxDDBS is a maximization problem, where the objective function is the number of vertices of the resulting subgraph. Many of these problems are $\mathcal{NP}-$hard, and hence, exhaustive exploration of the search space is not feasible. Approximate or heuristic methods sacrifice the guarantee of finding an optimal solution for the sake of lower computation overhead. The greatest danger for these methods is to be trapped in a *locally optimal solution*, which is an $\hat{s} \in \mathcal{S}$ such that $f(\hat{s}) \leq s$ (if we have a minimization problem), or $f(\hat{s}) \geq s$ (if we have a maximization problem), for all $s \in \mathcal{N}(\hat{s})$. Different methods are classified according to the strategy they use for coping with this problem.
In recent years, a set of strategies have been devised, which are very general in nature, and where particular problem-specific heuristics can be plugged in, in order to solve different combinatorial optimization problems. Due to their general nature, these methods are collectively called *metaheuristics*. They include greedy algorithms, basic local search, simulated annealing, tabu search, evolutionary algorithms (comprising genetic algorithms), and ant colony optimization, among others.
In the sequel we review some of these techniques, especially those that have been successfully

employed in DDP and related problems. W.l.o.g. we will assume that we have to *minimize* the objective function f. For more information about metaheuristics, including their taxonomy, and a framework for their analysis and design, we refer the reader to [20].

3.8.1 Greedy algorithms

The *greedy technique* is perhaps the most straightforward and most popular heuristic for dealing with combinatorial optimization problems. It constructs the solution $s^{(k)} = (x_1, \ldots, x_k)$ in stages, adding each new component x_i one by one, and leaving the previous components fixed. The i-th component is chosen as the $x_i \in D_i$ that optimizes the partial solution $s^{(i)} = (x_1, \ldots, x_i)$. Algorithm 3 describes the technique more formally.

Algorithm 3: The greedy method

Input : The objective function f and the set of constraints.
Output: A feasible solution $\hat{s}$.

```
for i := 1 to k do
    Choose x_i ∈ D_i such that s^(i) = (x_1, ..., x_i) is feasible,
    and f(s^(i)) is minimal;
end
Return ŝ = s^(k);
```

It is easy to see that the running time of Algorithm 3 is bounded above by $\sum_{i=1}^{k} |D_i|$, and hence polynomial in k, provided that the $|D_i|$ are polynomial in k. So simple as it looks, the greedy technique is the basis of many effient and popular algorithms, such as Dijkstra's algorithm, for computing single-source shortest paths in graphs, and Kruskal's algorithm, for constructing a minimum-weight spanning tree. Although in general the solution provided by a greedy algorithm is not optimal, it can be optimal if our problem has an underlying *matroid* or *greedoid* structure [96], such as in the two examples mentioned above. In Chapter 8 we employ the greedy technique to deal with MaxDDBS .

3.8.2 Basic local search and its derivatives

Local search starts with some initial feasible solution s, and iteratively tries to replace s by a better solution s' in the neighborhood of s, until a local minimum is reached. This process is also called *iterative improvement* for obvious reasons. The replacement of s by s' can be done in a number of ways. For instance, we can choose the best feasible solution in $\mathcal{N}(s)$, or we can take s' as the first element of $\mathcal{N}(s)$ that we come across, that is better than s, or any intermediate option.
The quality of the solutions obtained by basic local search is usually very poor, therefore a number of different strategies have been developed to reduce the risk of getting trapped in a local minimum. *Simulated annealing* is one of the first strategies in that sense. The main idea behind simulated annealing is to allow occasional moves resulting in a worse solution s' (uphill moves). This emulates a cooling process of metals, where the atoms strive to adopt a configuration of minimal potential energy (a crystal), but occasional rearrangements occur in

the opposite direction.
The process is depicted in Algorithm 4. The control parameter T (the temperature) regulates the speed of convergence and the probability of accepting worse solutions at any given moment. That probability is usually computed by the formula $p(T, s', s) = \exp(\frac{f(s')-f(s)}{T})$. The temperature is decreased during the search process, thus the probability of accepting uphill moves is higher at the beginning of the process, and gradually decreases, so that the process converges to basic local search. There is a trade-off associated with temperature regulation, namely computation time versus quality of solutions. If we decrease the temperature slowly, the chances of converging to a globally optimal solution are better, whereas computation time obviously increases. On the other hand, a quick temperature decrease leads to a smaller computation time, but the algorithm is more likely to get trapped in a local minimum. Therefore, the choice of the cooling schedule is critical for the performance of the algorithm.
Note that the probability of accepting an uphill move also depends on the absolute value of that move. Small uphill moves are more likely to be accepted than large ones. A variant of simulated annealing, called *threshold accepting* avoids the computation of probabilities in Step 8 of Algorithm 4, and simply compares the difference $f(s') - f(s)$ with a given threshold, to decide whether to accept it or not.

Algorithm 4: Simulated annealing

Input : The objective function f, the set of constraints, and the neighborhood function $\mathcal{N}$.
Output: A feasible solution $\hat{s}$ (that is in general a local minimum of f).

1 s:=Generate initial solution;
2 T:= T_0;
3 **while** *termination conditions not met* **do**
4 s':=PickAtRandom($\mathcal{N}(s)$);
5 **if** $f(s') < f(s)$ **then**
6 s:=s';
7 **else**
8 Accept s' with probability $p(T, s', s)$;
9 **end**
10 Update T;
11 **end**

Another metaheuristic that improves basic local search is *tabu search*. Simple tabu search keeps a list of the most recently visited solutions (tabu list), in order to avoid re-visiting them. In other words, it relies on *memory* to escape local minima and avoid cycles. The trade-off here involves the benefits provided by the tabu list, and the cost of maintaining and searching it.

3.8.3 Evolutionary algorithms

Evolutionary algorithms are inspired by natural evolution principles. The driving force of natural evolution is natural *selection*, survival of the fittest individuals among a diverse population. The main factors that contribute to population diversity are *recombination* and *mutation*. An

evolutionary algorithm emulates this process: it starts with an initial population of feasible solutions, and then iterates the three operators (recombination, mutation, and selection) in order to obtain increasingly better populations, where “better" here means having a lower value of the objective function f. Algorithm 5 describes a general evolutionary algorithm.

Algorithm 5: General evolutionary algorithm

Input : The objective function f, and the set of constraints.
Output: A good feasible solution $\hat{s}$.

```
P:=Generate initial population;
Evaluate P;
while termination conditions not met do
    P':=Recombine(P);
    P'':=Mutate(P');
    Evaluate(P'');
    P:=Select(P ∪ P'');
end
Return the 'best' individual of P;
```

In a variant of evolutionary algorithms called *memetic algorithms*, a local search is applied to every individual of the population [121, 122, 115]. Memetic algorithms have been used recently to address various network design problems, but not DDP specifically. For example, in [102] memetic algorithms are used to address two network design problems, namely *Vertex Biconnectivity Augmentation* (V2AUG), and the *Prize-Collecting Steiner Tree* problem (PCST). Also, the Distributed Algorithms Group at the University of Kaiserslautern has been consistently using memetic algorithms for the design of peer-to-peer and wireless networks (see for instance [66]).

3.8.4 Approximation algorithms

An *approximation algorithm* is any heuristic algorithm for which we can get a *performance guarantee*, i.e. we can *prove* that the solution given by the algorithm will always lie within a certain neighborhood of the optimal solution. Let us assume that we have a maximization problem, and let OPT be the value of the objective function f in the optimal solution of some problem instance, and ALG the value of f produced by the algorithm. The performance of the algorithm is measured by the ratio OPT/ALG, called the *approximation ratio*. Usually, the goal is to find a polynomial-time heuristic algorithm with the smallest approximation ratio OPT/ALG.

Combinatorial optimization problems can be classified according to the difficulty with which they can be approximated, i.e. the approximation ratio that can be reached in polynomial time, in the worst or average case. Our maximization problem is said to have a *Polynomial-Time Approximation Scheme (PTAS)* if for every $\epsilon > 0$ there exists a polynomial-time algorithm that computes a solution with worst-case approximation ratio $1 - \epsilon$ (or $1 + \epsilon$ for minimization problems).

In turn, the class of problems having a PTAS is included in the class Apx, that consists of all $\mathcal{NP}-$hard optimization problems for which there is a polynomial-time algorithm with a

constant approximation ratio OPT/ALG, for all problem instances. Loosely speaking, these are the problems that can be approximated in a reasonable time. Unfortunately, MaxDDBS does not belong to that class, as we will see in Section 8.1.

There are a number of books that treat the design of approximation algorithms in depth, e.g. [144] and [151]. In addition to that, [39] gives a list of optimization problems, and specifies which of them have a PTAS. [142] focuses on inapproximability results, and [116] investigates the conditions when the greedy technique yield good approximations.

3.9 Computer-based techniques in DDP

Since the number of graphs of a given maximum degree and diameter is very large, it is a natural idea to recourse to computers for help in the task of locating the largest instances. In Section 2.5.3 we have already seen how computer-guided search, combined with voltage assignment, has produced many instances of large graphs. However, as Δ and D grow, the search space becomes too large, even for computers, so that brute-force search is not feasible, and some heuristics are needed to guide the search, or prune the search space. In this section we briefly review some heuristic algorithms that have been used for that purpose.

The first serious attempt in that direction was made by Allwright, who used 2-opting, a local search technique developed for the Travelling Salesman Problem, to find large graphs with given degree and diameter [1].

In another landmark paper, Dinneen and Hafner used computer search and clever techniques to reduce the search space [51]. Their large graphs were Cayley graphs of non-abelian groups, like semidirect products of cyclic groups, for instance.

Around the same time, Mitjana and Comellas obtained the graph of order 253 at the entry (8,3) of the table, using a threshold accepting algorithm (see Section 3.8.2). It is also a Cayley graph of the semidirect product of two cyclic groups. The idea had been around since 1990, at least [35].

Later, Wohlmuth found another large Cayley graph with the aid of a genetic algorithm (see Section 3.8.3), that today still stands as the largest Cayley graph of degree 6 and diameter 3 [153]. Let us explain this algorithm briefly. The population consists of a set of Cayley graphs, that are subgroups of the symmetric group S_n, each one being represented by a generating set of $n-$permutations. Mutation of an individual Cayley graph is achieved by replacing one generator by the product of two different generators in the same generating set. Therefore, a subgroup of the former Cayley graph is obtained. The crossover operator exchanges generators from different generating sets. Finally, the fitness function to be minimized is defined by

$$f(x_1,x_2,x_3,x_4) = \begin{cases} 10x_2+10x_1+10 & \text{if } D \leq 2 \\ 10x_2+x_3 & \text{if } D = 3 \\ 10x_2-x_4 & \text{if } D \geq 3 \end{cases}$$

where x_i represents the number of vertices with distance i from any fixed vertex x (the choice of x is irrelevant, since we are working with a Cayley graph, which is vertex-transitive). That function was chosen so as to minimize the number of cycles of length ≤ 6, which correspond to large values of x_1, x_2, x_3, and to minimize the number of vertices at distance 4.

Sampels also used a genetic algorithm for constructing large Cayley graphs [134]. Some of the groups he used were specified by a finite presentation, but the large majority of them were semidirect products of cyclic groups, generated by a set of permutations.
Exoo argued that Cayley graphs are relatively rare, and therefore it makes sense to relax the symmetry conditions, and look for improvements in a larger search space. The graphs he considered have order that is a small integral multiple of the size of their respective automorphism group. Thus, he managed to obtain several large graphs in the low-order entries of the table, with the aid of an algorithm that is a synthesis of simulated annealing and tabu search (see Section 3.8.2), which had been previously used to construct Ramsey colourings. The algorithm is described in [57], and the graphs are given in [58].
Finally, [118] describes a hybrid heuristic algorithm (HSAGA) to solve a relaxed variant of the Degree-Diameter Problem. HSAGA stands for 'Hybrid Simulated Annealing Genetic Algorithm'.

Part II

The Degree-Diameter Problem

4

Large graphs by compounding

Our first original result in the Degree-Diameter Problem provides new constructions of large graphs of diameter 6 with the aid of the compounding operation described in 3.1.1. More specifically, by compounding complete graphs of order h (K_h), $h \leq \Delta - 1$, into a bipartite Moore graph of degree Δ and diameter 6 ($H_{\Delta-1}$), a family of large graphs $H_{\Delta-1}(K_h)$ of maximum degree Δ and diameter 6 is obtained. This family includes some of the largest known graphs of diameter 6 [34]. These results have been published in [127].

4.1 Compound Graphs $H_{\Delta-1}(K_h)$

Construction of the compound graphs $H_{\Delta-1}(K_h)$ is achieved in two stages. The first stage is the compounding operation itself, and the second consists of the addition of new edges.
In the compounding operation, if we use $S = \{K_h\}$ as the set of source graphs and G as the base graph, it is easy to see that the order of any compound graph $G(S)$ will be

$$|G(S)| = |G| + |\hat{W}|(h-1) \tag{4.1}$$

where $\hat{W}$ is the vertex set of the replaced graph $\hat{G} = (\hat{W}, \hat{E} = \emptyset)$.

Stage 1: The compounding operation

Source Graphs: K_h, the complete graph of order h, for $h \leq \Delta - 1$, and $\Delta \in \mathbb{Z}^+$.

Base Graph: A bipartite Moore graph $H_{\Delta-1}$ with partite sets V and W, edge set E, degree Δ and diameter 6. Recall that $\Delta - 1$ is a power of a prime.

Replaced Graph: Let R be the subgraph of $H_{\Delta-1}$ depicted in Figure 4.1. Let x be a vertex of V, and let $N(x) = \{x_0, x_1, \ldots, x_{\Delta-1}\}$. Now define the sets

(i) $W^0 = N(x) - \{x_{\Delta-1}\}$.

(ii) $V^0 = N(W^0) - \{x\} - \bigcup_{i\in\{0,\ldots,\Delta-2\}} \{x_{i\Delta-2}\}$.

(iii) $W' = N(V^0) - W^0 - \bigcup_{\substack{i\in\{0,\ldots,\Delta-2\}\\ j\in\{0,\ldots,\Delta-3\}}} \{x_{ij\Delta-2}\}$.

The set $W' \subset W$ is called the *set of replaceable vertices*, and is depicted in Figure 4.1. Given a subset $\hat{W}$ of W', the replaced graph will then be $\hat{G} = (\hat{W}, \hat{E} = \emptyset)$.

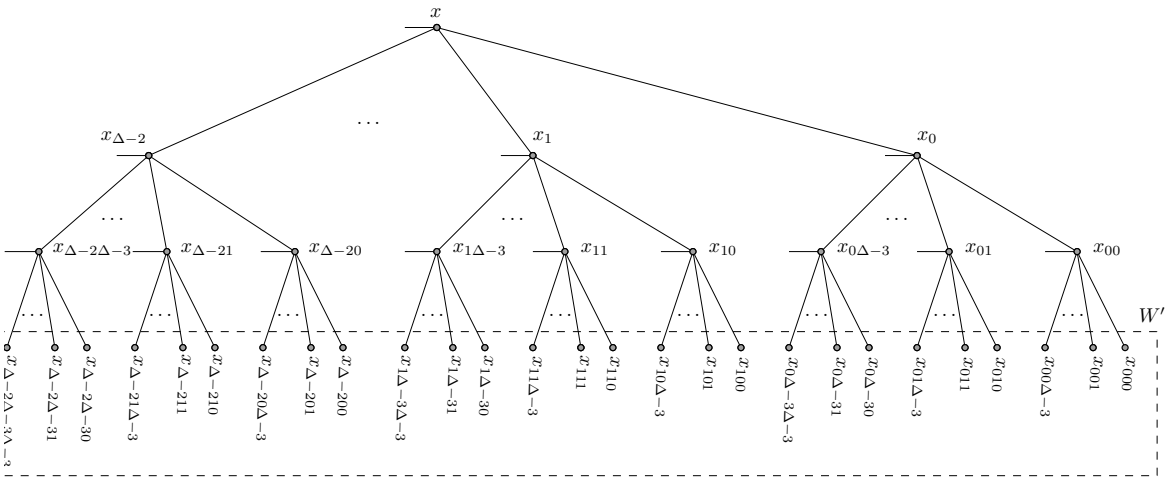


Figure 4.1: Subgraph R of $H_{\Delta-1}$ to be modified.

Step 1: Each vertex x_{ijk} of $\hat{W}$ is replaced by a copy of the complete graph K_h, denoted $K_h^{(ijk)}$, with vertex set $V(K_h^{(ijk)}) = \left\{y_0^{(ijk)}, \ldots, y_{h-1}^{(ijk)}\right\}$.

The set of added vertices will be denoted by $\hat{W}(K_h)$. Therefore, $\hat{W}(K_h) = \bigcup_{x_{ijk}\in\hat{W}} V(K_h^{(ijk)})$.

Step 2: At this step, the edges incident with each vertex $x_{ijk} \in \hat{W}$ will be joined to the vertices of $K_h^{(ijk)}$ so that each vertex of $K_h^{(ijk)}$ will be incident with at least one of these edges.

Given a vertex $x_{ijk} \in \hat{W}$ and the complete graph $K_h^{(ijk)}$, we denote by $y_0^{(ijk)} \in K_h^{(ijk)}$ the vertex that has x_{ij} as a neighbour. Then $y_0^{(ijk)}$ will be connected to at least one other neighbour of x_{ijk}.

The resulting graph will be denoted $H^0_{\Delta-1}(K_h)$.

Stage 2: Introducing new edges

If we want to ensure that the graphs $H_{\Delta-1}(K_h)$ will have diameter 6, we must introduce new edges. The set of edges that will be added at this stage will be denoted by $\hat{E}(K_h)$. Therefore,

we will define the graph $\hat{G}(K_h) = (\hat{W}(K_h), \hat{E}(K_h))$.
The edge set of the graph $\hat{G} = (\hat{W}, \hat{E})$ is defined as follows: For $x_{ijk}, x_{rst} \in \hat{W}$, $\{x_{ijk}, x_{rst}\} \in \hat{E} \leftrightarrow \exists\ \alpha, \beta \in \{0, \ldots, h-1\}$ such that $\{y_\alpha^{(ijk)}, y_\beta^{(rst)}\} \in E(V(K_h^{(ijk)}), V(K_h^{(rst)}))$, that is, $\{y_\alpha^{(ijk)}, y_\beta^{(rst)}\} \in \hat{E}(K_h)$. Figure 4.2 depicts a graph $\hat{G}(K_3)$ and its corresponding graph $\hat{G}$.

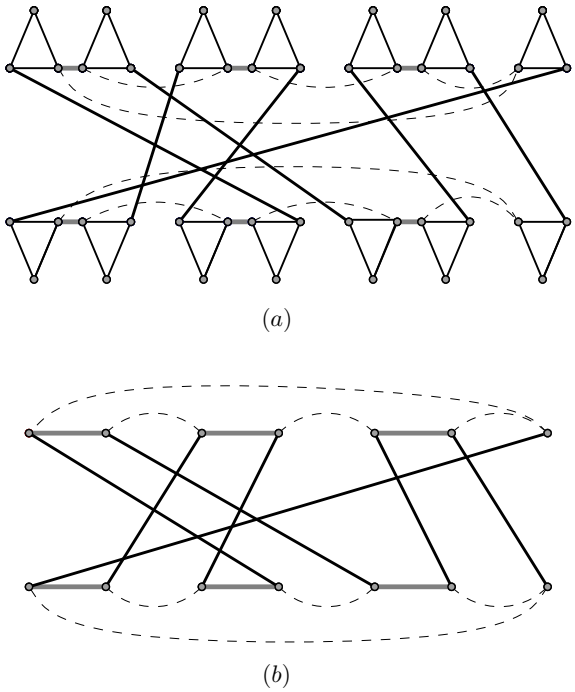


Figure 4.2: (a): Graph $\hat{G}(K_3)$ and (b): graph $\hat{G}$ obtained from $\hat{G}(K_3)$.

A *block* is a set of copies of $K_h^{(ijk)}$. In practice we will mainly use blocks made of copies of $K_h^{(ijk)}$ with i fixed. Let N_b denote the total number of blocks, and define the *margin* M as the number of edges that can be added to the vertices of each K_h-copy so that the maximum degree of $H_{\Delta-1}(K_h)$ remains at most Δ. It is not hard to see that $M = (\Delta - h)(h-1)$. Finally, $M' = M - \delta$ will denote the number of edges that can be added to each K_h-copy after the application of Condition 1.
Now, the new edges to be added to $H^0_{\Delta-1}(K_h)$ should meet the following conditions:

Condition 1: There should be at least one edge between each pair of copies $K_h^{(ijk)}$ and $K_h^{(ijt)}$, for $k \neq t$. The maximum number of edges joining a copy $K_h^{(ijk)}$ to any other copy $K_h^{(ijt)}$ is denoted by δ, for $k \neq t$.

Condition 2: The distance between any two copies of $K_h^{(ijk)}$ inside a block should be at most 3. In particular, the distance between any two copies $K_h^{(ijk)}$ and $K_h^{(ist)}$ should be at most 3.

Condition 3: The distance between any two copies $K_h^{(ijk)}$ and $K_h^{(rst)}$ should be at most 4, for $i \neq r$. Equivalently, the distance between any two copies of $K_h^{(ijk)}$ from different blocks should be at most 4.

Condition 3a: The distance between any vertex $y_\alpha^{(ijk)}$ and any copy of type $K_h^{(rst)}$, $i \neq r$, should be at most 5.

Condition 4: All these adjacencies are added in such a way that the maximum degree of $H_{\Delta-1}(K_h)$ remains at most Δ.

The graph so obtained after the addition of the edges is $H_{\Delta-1}(K_h)$.

Note that in order to show that the distance between any vertex $y_\alpha^{(ijk)} \in K_h^{(ijk)}$ and any vertex $y_\beta^{(rst)} \in K_h^{(rst)}$, $i \neq r$, is at most 6, we can use either Condition 3 or 3a. Note also that the cardinality of each block need not be the same.

Figure 4.3 illustrates two blocks with their intra and interconnections, satisfying Conditions 1, 2 and 3.

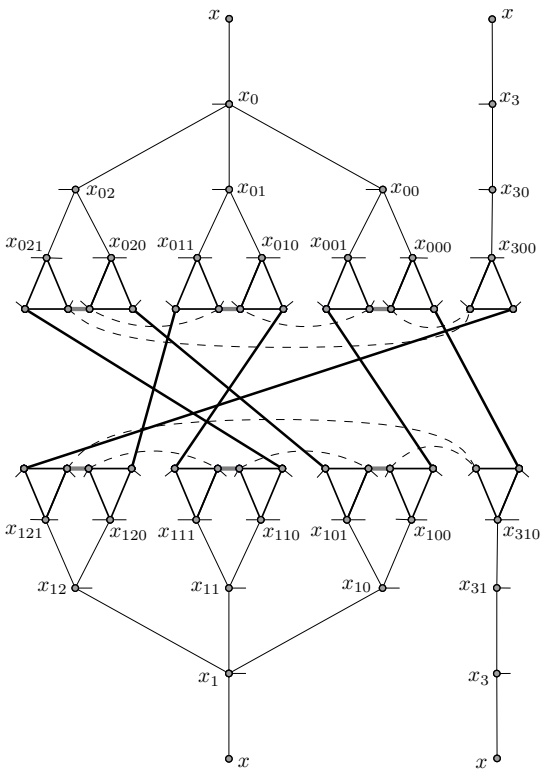


Figure 4.3: Two blocks, namely, $\{K_3^{(000)}, K_3^{(001)}, K_3^{(010)}, K_3^{(011)}, K_3^{(020)}, K_3^{(021)}, K_3^{(300)}\}$ and $\{K_3^{(100)}, K_3^{(101)}, K_3^{(110)}, K_3^{(111)}, K_3^{(120)}, K_3^{(121)}, K_3^{(310)}\}$, and their interconnections. The edges that ensure Condition 1 are colored in gray, and the edges that ensure Conditions 2 and 3 are highlighted by dashed and heavier lines, respectively.

★ **Lemma 4.1** *In $H_{\Delta-1}(K_h)$, the graph $\hat{\Gamma}(K_h) = (\hat{W}(K_h), \hat{E}(K_h))$ has diameter 6.*

Proof. Due to Condition 2, in the graph $\hat{\Gamma}(K_h)$, the distance between any two copies of $K_h^{(ijk)}$ inside a given block is 3. Hence, the distance between any two vertices $y_\alpha^{(ijk)}$ and $y_\beta^{(rst)}$

belonging to the same block is at most 5. Moreover, the distance between any two vertices $y_\alpha^{(ijk)}$ and $y_\beta^{(rst)}$ belonging to different blocks is at most 6, by virtue of Condition 3 or Condition 3a. If we take into account that $\hat{\Gamma}(K_h)$ is the union of the N_b blocks, the result follows. □

★ **Theorem 4.1** *Any graph $H_{\Delta-1}(K_h)$ constructed according to the Conditions 1-4 above has maximum degree Δ and diameter 6.*

Proof. Let $H_{\Delta-1}(K_h)$ be a graph satisfying Conditions 1-4. In such a graph, any path that does not contain vertices of $\hat{W}$ remains unaffected.

In a shortest path of any $H_{\Delta-1}(K_h)$ at most two vertices can be replaced − see the selected subgraph R of $H_{\Delta-1}$ (Figure 4.1), and recall that the girth of $H_{\Delta-1}$ is 12. Accordingly, the distance between any two vertices will increase by 2, at most.

Following Lemma 4.1, the distance between any two vertices $y_\alpha^{(ijk)}$ and $y_\beta^{(rst)}$ is at most 6.

Let us consider two unreplaced vertices z and t lying at distance 5 from one another in $H_{\Delta-1}$ such that a shortest path P joining them has two replaced vertices, say $P = zx_{ijk}x_{ij}x_{ijl}rt$. According to Condition 1, there should be an edge between each pair of copies $K_h^{(ijk)}$ and $K_h^{(ijl)}$, say $\{y_\alpha^{(ijk)}, y_\beta^{(ijl)}\} \in E(V(K_h^{(ijk)}), V(K_h^{(ijl)}))$. Then $zy_\theta^{(ijk)}y_\alpha^{(ijk)}y_\beta^{(ijl)}y_\gamma^{(ijl)}rt$ is a path of length at most 6 between z and t. Figure 4.4 shows an example of this situation.

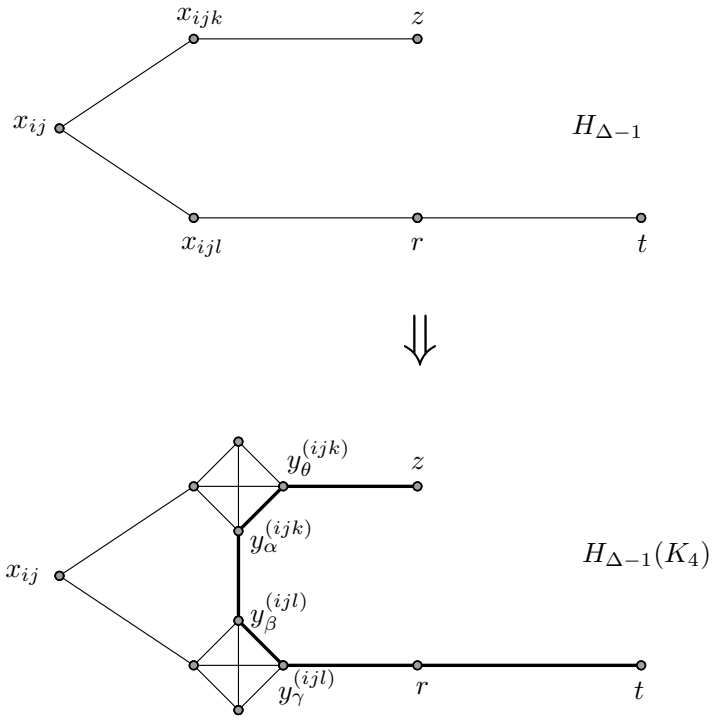


Figure 4.4: A path of length 6 between z and t in $H_{\Delta-1}(K_4)$

Consider now two vertices z and x_{ijk} at distance 5 from one another in $H_{\Delta-1}$, such that $z \in V$ and $x_{ijk} \in \hat{W}$. Let $P = zx_{imn}x_{im}x_ix_{ij}x_{ijk}$ be a shortest path joining z and x_{ijk}. According

to Condition 2, there should be a path of length at most 3 between each pair of copies $K_h^{(imn)}$ and $K_h^{(ijk)}$. Then there is a path of length ≤ 5 between any pair of vertices $y_\alpha^{(imn)}$ and $y_\beta^{(ijk)}$. Thus the distance between vertex z and the vertices $y_\alpha^{(ijk)} \in K_h^{(ijk)}$ is at most 6.

Suppose that we have two unreplaced vertices z and t at distance 6 in $H_{\Delta-1}$. Then, as there are Δ disjoint paths between them, there exists a 6-path between z and t, which is not affected by the replacements.

Finally, consider two vertices z and x_{ijk} at distance 6 from each other in $H_{\Delta-1}$, such that $z \notin \hat{W}$ and $x_{ijk} \in \hat{W}$. Then there exist Δ disjoint paths between them. We need to check that the distance between the vertex z and vertices $y_\alpha^{(ijk)} \in K_h^{(ijk)}$ is at most 6.

Due to the structure of the selected subgraph R of $H_{\Delta-1}$ and the fact that $g(H_{\Delta-1}) = 12$, a shortest path linking x_{ijk} to another replaced vertex passes through x_{ij}. Therefore, there is a unique shortest path P_6 in $H_{\Delta-1}$ that joins x_{ijk} and z, which has one further replaced vertex; the remaining $\Delta - 1$ paths, namely, $P_6^1, \ldots, P_6^{\Delta-1}$, have no replaced vertex other than x_{ijk}. Thus, there exists a unique vertex $y_\alpha^{(ijk)}$ of $K_h^{(ijk)}$ contained in $P_6' = zP_6x_{ij}y_\alpha^{(ijk)} \in H_{\Delta-1}(K_h)$. We define the 6-paths $P_6'^i = zP_6^i y_\theta^{(ijk)} \in H_{\Delta-1}(K_h)$ for $i = 1, \ldots, \Delta-1$ and any $y_\theta^{(ijk)} \in K_h^{(ijk)}$. Then the remaining $h-1$ vertices $y_\beta^{(ijk)} \neq y_\alpha^{(ijk)}$ of $K_h^{(ijk)}$ are connected to z through one of the 6-paths $P_6'^i$ for $i = 1, \ldots, \Delta-1$. Let us now consider the vertex $y_\alpha^{(ijk)} \in K_h^{(ijk)}$ contained in P_6'. Then $y_\alpha^{(ijk)}$ is adjacent to x_{ij}, and therefore the vertex $y_\alpha^{(ijk)}$ is the same as the vertex $y_0^{(ijk)}$. Hence, due to the compounding operation, the vertex $y_0^{(ijk)}$ is connected to x_{ij} and to at least another neighbour of x_{ijk}. Consequently, one of the 6-paths $P_6'^i \in H_{\Delta-1}(K_h)$, $i = 1, \ldots, \Delta-1$, connects $y_0^{(ijk)}$ to z. Figure 4.5 exemplifies this case. □

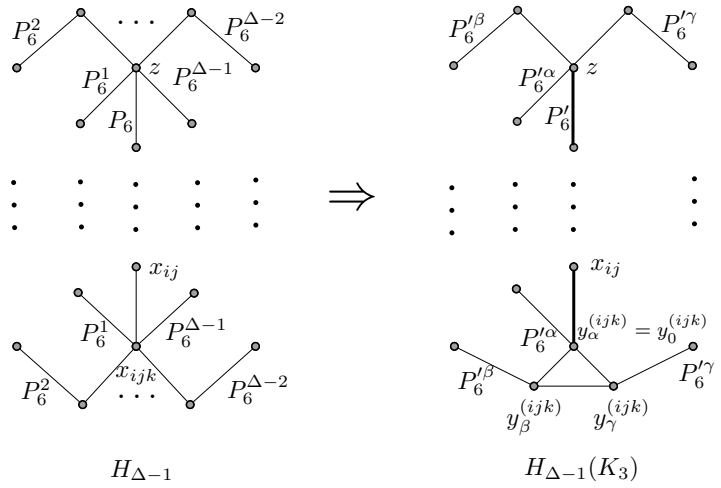


Figure 4.5: Paths of length 6 between z and vertices $y_0^{(ijk)}$ in $H_{\Delta-1}(K_3)$

Thus, in order to construct large $H_{\Delta-1}(K_h)$ graphs, we need to link the copies of K_h according to the structure of a certain graph, and following Conditions 1-4. Some authors have joined copies of K_h according to a special graph of diameter 2 [76], while still others have used a bipartite graph of diameter 3 [71]. In the next two sections we describe two additional variants

for joining the K_h-copies. This completes the construction of $H_{\Delta-1}(K_h)$, and results in some record graphs of diameter 6.

4.2 Multipartite replaced graphs

In this variant, copies of K_h are joined by following the structure of a modified N_b-partite graph, that is, $\hat{G}$ is a modified N_b-partite graph. To be more precise, $\hat{G}$ is constructed in the following way. We take N_b sets of vertices $\{A_1, \ldots, A_{N_b}\}$, join A_i to A_j, $i \neq j$, by means of a perfect matching, and finally, the vertices within each A_i are connected according to the structure of a diameter 3 graph.

The graph $H_4(K_3)$

The graph $H_4(K_3)$ is constructed by setting up the blocks $\left\{K_3^{(0jk)}, K_3^{(300)}\right\}$, $\left\{K_3^{(1jk)}, K_3^{(310)}\right\}$ and $\left\{K_3^{(2jk)}, K_3^{(320)}\right\}$ for $j \in \{0,1,2\}$ and $k \in \{0,1\}$.
Subsequently, we connect any two blocks as shown in Figure 4.6. Note that $\hat{\Gamma}$ is made up by three sets of seven vertices, and vertices belonging to the same set are connected by a 7-cycle. $\hat{\Gamma}$ is partially represented in Figure 4.2 (b).
The resulting graph $H_4(K_3)$ has degree 5 and diameter 6. After applying the compounding operation to these 21 vertices, we get $M = (5-3)(3-1) = 4$; hence we can add 4 edges to each K_3. Additionally, after applying Condition 1, we have $\delta = 1$, so $M' = 4 - 1 = 3$.
For verifying Condition 2, note that any two copies $K_3^{(ijk)}$ belonging to the same block are joined by a 7-cycle. Moreover, there is a 3-path between any two copies $K_3^{(3jk)}$. These connections are indicated by a dashed line in Figure 4.6.
Thereafter, it only remains to verify that the distance between any two copies $K_3^{(ijk)}$ and $K_3^{(opm)}$, $i \neq o$, is at most 4, or in other words, that the distance between any two copies of $K_3^{(ijk)}$ from different blocks is at most 4.
Figure 4.6 shows that the previous assertion is true for any copy $K_3^{(0jk)}$, $K_3^{(1mn)}$, $K_3^{(300)}$ and $K_3^{(310)}$. We can verify this fact by following the connections between such graphs. These connections are highlighted by a heavier line (Figure 4.6 only shows a part of these connections).
The vertices $y_1^{(ijk)}$ and $y_1^{(opm)}$ link any two blocks in $H_4(K_3)$. Note that each of these vertices has two connections for this purpose, and we only need one of these connections to set up a path of length ≤ 4 between any two copies $K_3^{(ijk)}$ and $K_3^{(opm)}$ belonging to two different blocks.
With the aid of Equation (4.1) we can compute the order of the new graph $H_4(K_3)$ as $|H_4| + 21 \times 2 = 2772$. This is an improvement over the previous value of the entry $(5,6)$ in the table of the largest known graphs, which was 2766.

The graph $H_5(K_4)$

In order to construct the graph $H_5(K_4)$, we organize the blocks as follows: $\left\{K_4^{(0jk)}, K_4^{(030)}\right\}$, $\left\{K_4^{(1jk)}, K_4^{(130)}\right\}$, $\left\{K_4^{(2jk)}, K_4^{(230)}\right\}$, $\left\{K_4^{(3jk)}, K_4^{(330)}\right\}$ and $\left\{K_4^{(4jk)}, K_4^{(430)}\right\}$, for $j \in \{0,1,2\}$

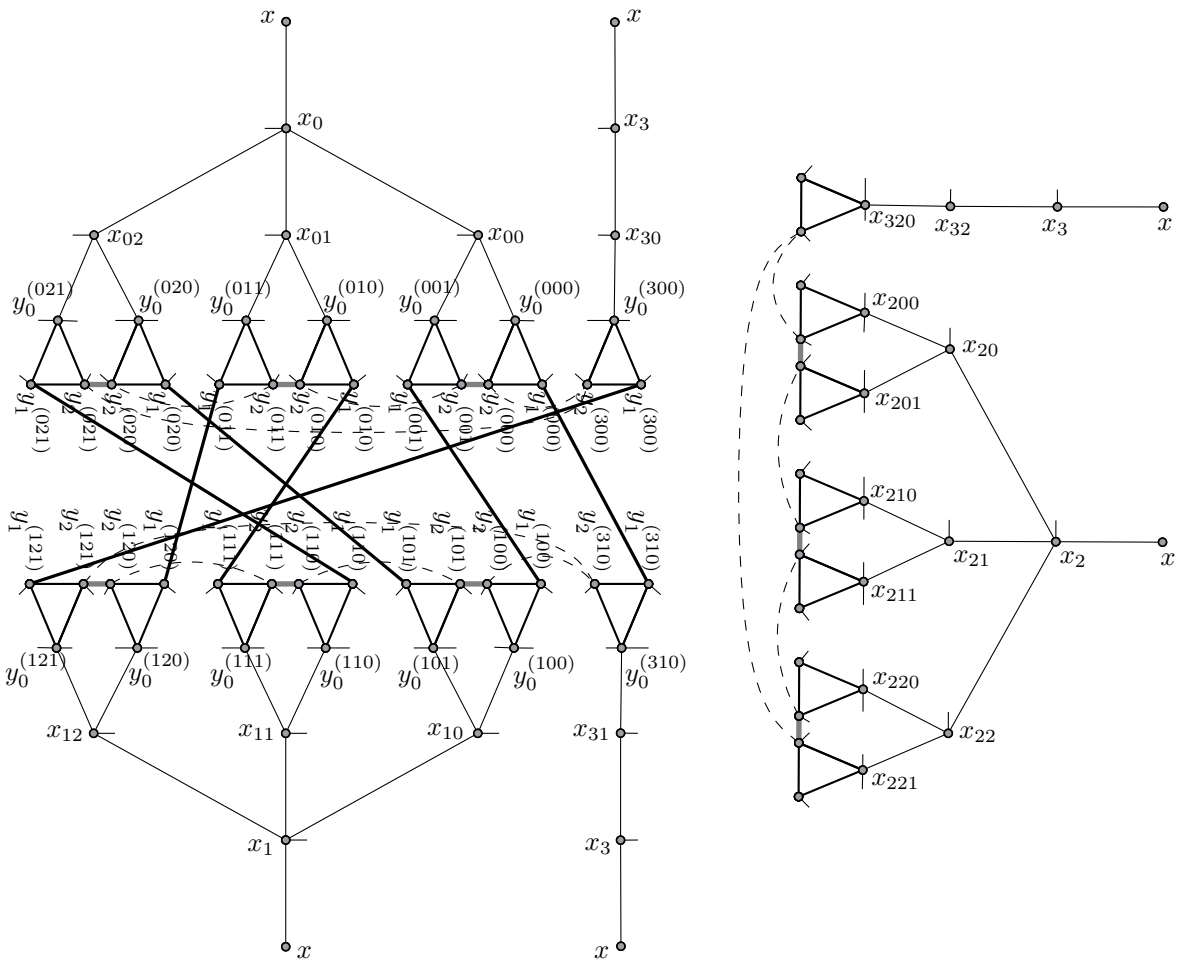


Figure 4.6: Basic configuration of a $H_4(K_3)$ graph of order 2772.

and $k \in \{0, 1\}$.

As before, two blocks are connected as in Figure 4.7. Note that, in this case, $\hat{\Gamma}$ is made up by five sets of seven vertices, and vertices of the same set are connected by a 7-cycle.
The graph $H_5(K_4)$ so constructed has degree 6 and diameter 6. After applying the compounding operation to its 35 vertices we have $M = (6-4)(4-1) = 6$, and we can add six edges to each K_4. Additionally, after applying Condition 1 we get $\delta = 1$, so $M' = 6 - 1 = 5$.
Inter-block connections are indicated by a heavier line in Figure 4.7. Furthermore, note that for each copy $K_4^{(ijk)}$ of a block, we have exactly two vertices (shown in white in Figure 4.7) with two connections, and we only need one of these to connect any two copies $K_4^{(ijk)}$ and $K_4^{(opm)}$ belonging to two different blocks, by a path of length ≤ 4.
With the aid of Equation (4.1), we get the order of the new graph $H_5(K_4)$: $|H_5|+35\times3 = 7917$. This is an improvement over the previous $(6, 6)$ entry, which was 7908.

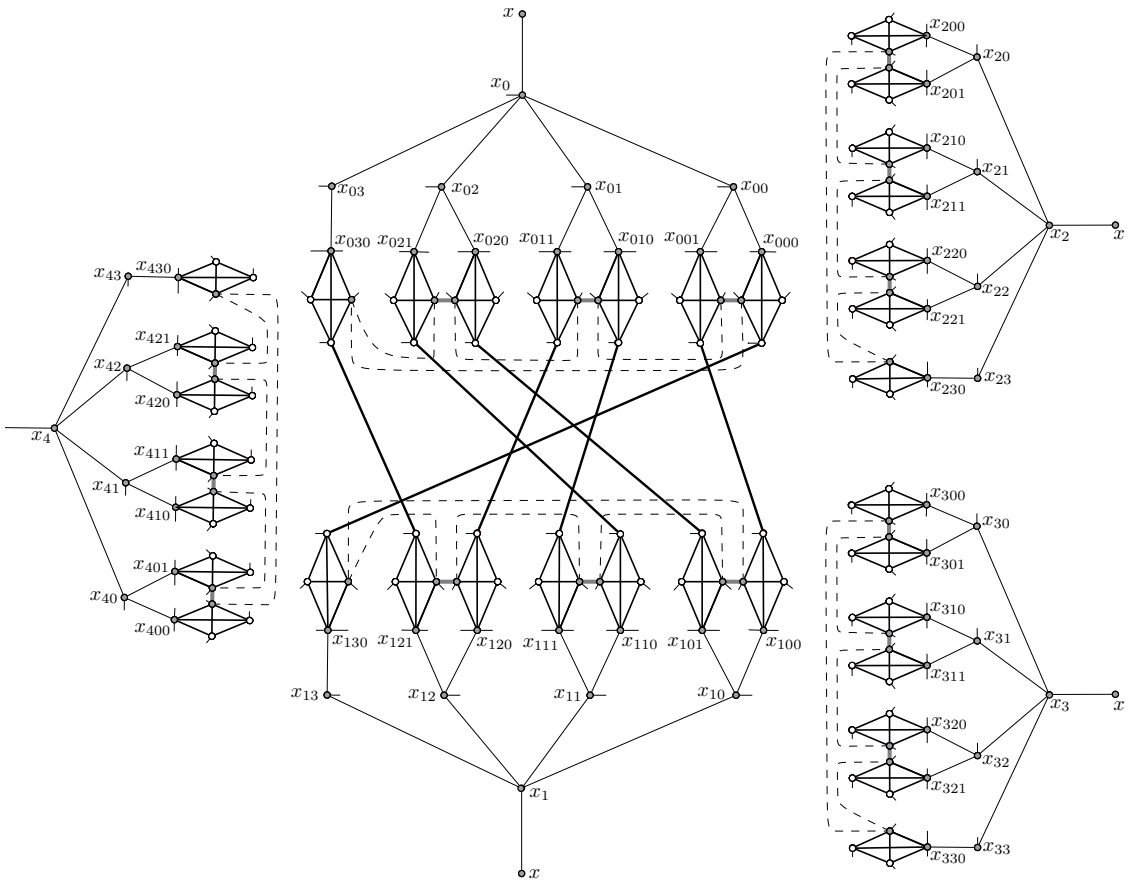


Figure 4.7: Basic configuration of a $H_5(K_4)$ graph of order 7917.

4.3 $G(B_\omega)$ as replaced graphs

In this second approach, copies of K_h are joined according to the structure of a compound graph of diameter 3, denoted $G(B_\omega)$, that is, $\hat{\Gamma}$ is isomorphic to $G(B_\omega)$. The graphs $G(B_\omega)$

are defined next.

4.3.1 Description of the graphs $G(B_\omega)$

The graphs $G(B_\omega)$ are constructed as follows: Let G be a graph, and $B_\omega = (V_1 \cup V_2, E)$ be a bipartite graph. Assume that B_ω has a polarity ω, so that we do not have to make any distinction between its partite sets, i.e. $V_1 = V_2 = V$. Let us now perform the compounding of B_ω into G by replacing each vertex u in G with $S_u = V$. If u and v are adjacent in G, then S_u and S_v are connected according to the structure of B_ω. A vertex t of $G(B_\omega)$ such that $t \in S_u$ will be denoted by the pair (t, S_u). Note that we can match the vertex (t, S_u) in $G(B_\omega)$ with the vertex t of V in B_ω. We may assume that $t \in V_1$ or $t \in V_2$, as needed. Figure 4.8 shows examples of graphs $G(B_\omega)$.

Let us now recall some known properties of bipartite graphs:

(*i*) If $B = (V_1 \cup V_2, E)$ is any bipartite graph of even diameter $D(B)$ then the distance between $x \in V_1$ and any $y \in V_2$ is at most $D(B) - 1$. If the diameter $D(B)$ of $B = (V_1 \cup V_2, E)$ is odd, then the distance between $x \in V_1$ and any $y \in V_1$ is at most $D(B)-1$. Consequently, from any vertex of V_i, $i = 1, 2$, we reach all vertices from either V_1 or V_2 by means of paths of lengths at most $D(B) - 1$.

(*ii*) Moreover, if we have two vertices u_0 and u_l joined by a path P of length $l = D(B)-1-2r$ for $r \geq 0$, say $P = u_0u_1 \dots u_l$, we can find a walk Z of length $D(B) - 1$ between u_0 and u_l. In fact, $Z = Z' \cup P$, where Z' is any closed walk of length $2r$ starting at u_0. Note that we can always obtain such a walk Z', for instance, we can take $Z' = \underbrace{u_0u_1u_0u_1 \dots u_1u_0}_{2r \text{ steps}}$.

Now, some properties of the graphs $G(B_\omega)$ are given in the next theorem. As usual, $D(G)$ denotes the diameter of the graph G.

★ **Theorem 4.2** *The graphs $G(B_\omega)$ have maximum degree $\Delta(G) \times \Delta(B_\omega)$, order $\frac{|G||B_\omega|}{2}$ and diameter $\max\{D(G), D(B_\omega)\}$.*

Proof. We concentrate on proving the assertion regarding the diameter, since the maximum degree and the order follow from the construction.

Let z_0 and z_k be vertices of G at distance k, and let $P = z_0z_1 \dots z_k$ be a shortest path joining them. Additionally, let u and v be vertices of $G(B_\omega)$ such that $u \in S_{z_0}$ and $v \in S_{z_k}$; see Figure 4.9.

We intend to prove that for any two vertices u and v in $G(B_\omega)$, $d_{G(B_\omega)}(u, v) = \max\{k, D(B_\omega)\}$. More precisely, we will take a path P of length k between two vertices z_0 and z_k in G, say $P = z_0z_1 \dots z_k$, and then we will show that $d_{G(B_\omega)}((u, S_{z_0}), (v, S_{z_k})) = \max\{k, D(B_\omega)\}$.

As $B_\omega = (V_1 \cup V_2, E)$ has a polarity, (u, S_{z_0}) corresponds to a vertex u in V_1.

When $k = 0, 1$, u and v are located in B_ω, so $d_{G(B_\omega)}(u, v) = d_{B_\omega}(u, v)$. Let us assume that $k > 1$; we distinguish two cases:

Case 1. $k \geq D(B_\omega)$.

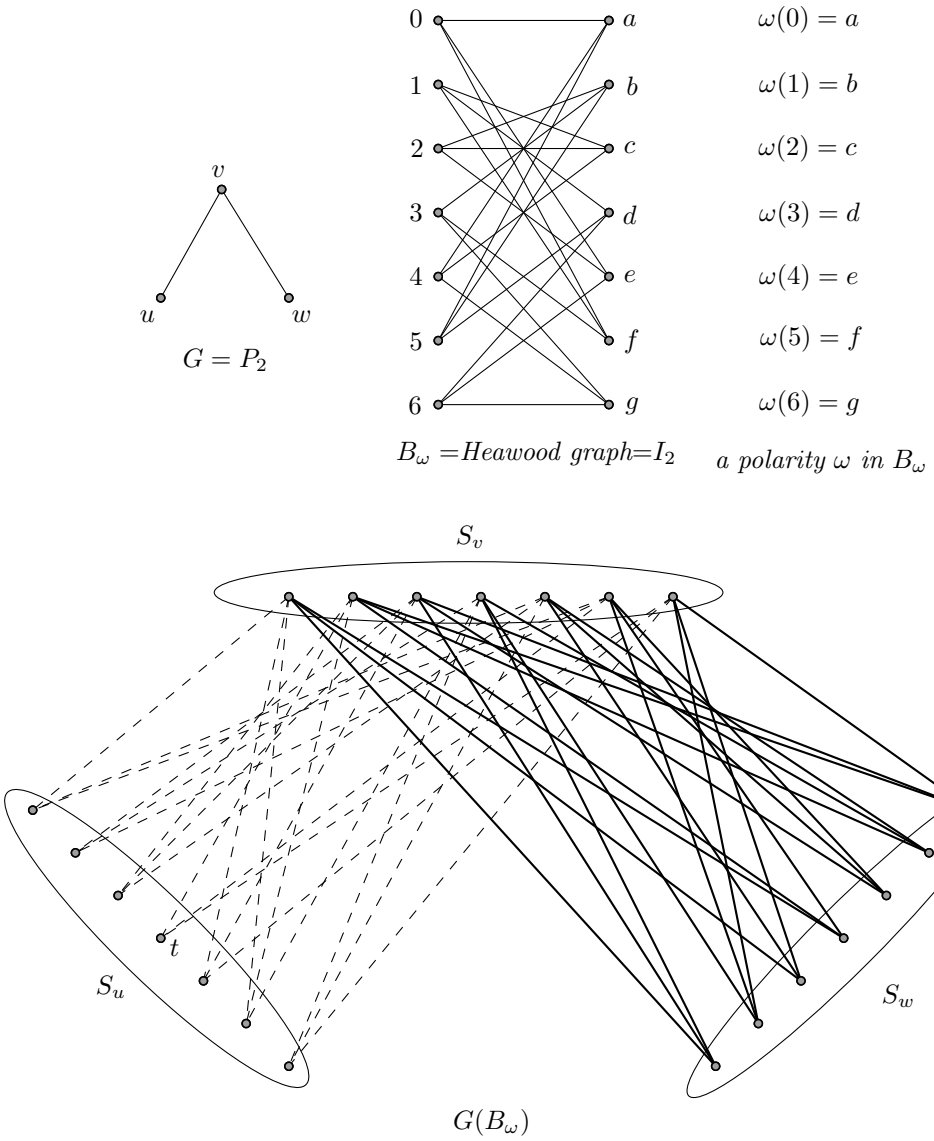


Figure 4.8: Example of a $G(B_\omega)$ graph, where $G = P_2$ and B_ω =*Heawood graph*.

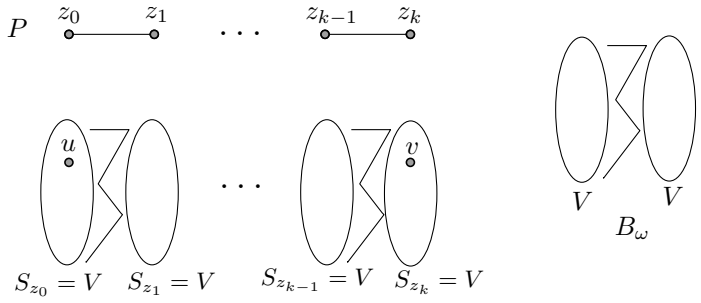


Figure 4.9: Path of length k in G and the corresponding structure in $G(B_\omega)$.

Let u_l be any vertex different from u in V_1 (V_2), provided that $D(B_\omega)$ is odd (even). Let Q be a shortest path in B_ω between u and u_l, say $Q = uu_1 \dots u_l$. Because of (i), we may assume that the length of Q has the form $l = D(B_\omega) - 1 - 2r$ for $r \geq 0$.

By (ii), we can construct a walk Z of length $D(B_\omega) - 1$ from u to u_l in B_ω. For example, we can take the path $Z = \underbrace{uu_1u \dots u}_{2r \text{ steps}} u_1u_2 \dots u_l$.

Consider now the vertex $(u_l, S_{D(B_\omega)-1})$.

As B_ω has a polarity, and using (i), we can reach $(u_l, S_{z_{D(B_\omega)-1}})$ from (u, S_{z_0}) through the $(D(B_\omega) - 1)$-path $(u, S_{z_0})(u_1, S_{z_1})(u, S_{z_2}) \dots (u, S_{z_{2r}})(u_1, S_{z_{2r+1}}) \dots (u_l, S_{z_{D(B_\omega)-1}})$.

Thus, there exist $(k - (D(B_\omega) - 1))$-paths from (v, S_{z_k}) to certain vertices in $S_{z_{D(B_\omega)-1}}$, hence there is a path from (u, S_{z_0}) to (v, S_{z_k}) of length k; see Figure 4.10.

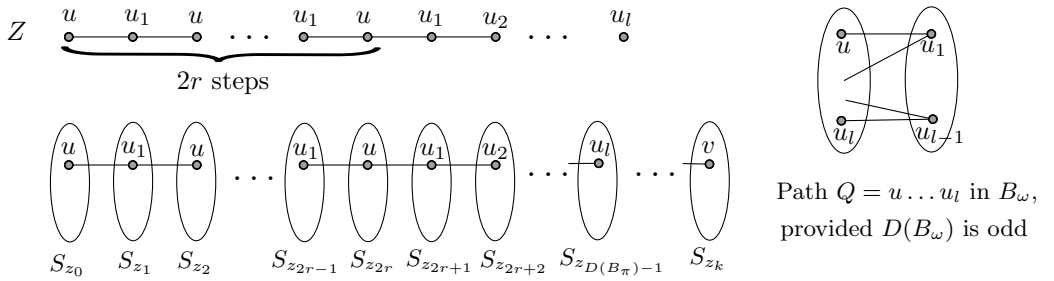


Figure 4.10: Path of length k between (u, S_{z_0}) and (v, S_{z_k}) in $G(B_\omega)$.

Case 2. $k < D(B_\omega)$.

This case is in turn split into two subcases:

Case 2 A $k + L_1 = D(B_\omega) - 1$, where L_1 is even.

Again, we can associate (v, S_{z_k}) with a vertex $v \neq u$, where $v \in V_1$ if $D(B_\omega)$ is odd, and $v \in V_2$ if $D(B_\omega)$ is even. Hence we can take Q as a path of length $k = D(B_\omega) - 1 - L_1$ in B_ω joining u with v, say $Q = uu_1u_2 \dots u_{k-1}v$, for $L_1 \geq 0$.

Now the vertices (u, S_{z_0}) and (v, S_{z_k}) are joined by the path $(u, S_{z_0})(u_1, S_{z_1})(u_2, S_{z_2}) \dots$ $\dots (u_{k-1}, S_{z_{k-1}})(v, S_{z_k})$ of length k.

Case 2 B $k + L_1 = D(B_\omega) - 1$, where L_1 is odd.

As $k > 1$, this subcase can be reduced to the previous subcase. As a matter of fact, $k-1+L_2 = D(B_\omega) - 1$, where $L_2 = L_1 + 1$. Hence, we will be able to reach any vertex in $S_{z_{k-1}}$ in at most $D(B_\omega) - 1$ steps. Consequently, since we have edges going from (v, S_{z_k}) to certain vertices in $S_{z_{k-1}}$, we will have a path from (u, S_{z_0}) to (v, S_{z_k}) of length at most $D(B_\omega)$.

In summary, $d_{G(B_\omega)}((u, S_{z_0}), (v, S_{z_k})) \leq D(B_\omega)$ in Case 2. □

4.3.2 Construction of $H_8(K_6)$, $H_{11}(K_8)$, and $H_{13}(K_{10})$

Now that we have defined graphs $G(B_\omega)$ and deduced some of their properties, we are prepared to present other record graphs $H_{\Delta-1}(K_h)$.

General Construction: Let G be a graph, and B_ω be a bipartite graph with a polarity ω. Assume that both G and B_ω have diameter three.

Step 1: Label the vertices of G in such a way that two vertices have the same label only if they are at distance 2 in G. A vertex in G can have several labels; for example, if $d(x,y) = d(y,z) = 2$ and $d(x,z) = 4$ then the labels of vertices x, y and z could be l_1, $\{l_1, l_2\}$ and l_2, respectively.

Step 2: Perform the compounding $G(B_\omega)$. Each vertex $(u, S_{z_i}) \in G(B_\omega)$ takes only one of the labels from the vertex $z_i \in G$, so we now denote the vertex (u, S_{z_i}) of $G(B_\omega)$ by the triple (u, S_{z_i}, l), where l is one of the labels of $z_i \in G$. The distribution of these labels in each S_{z_i} is done according to the specific $G(B_\omega)$. The graph $\hat{\Gamma}$ is isomorphic to $G(B_\omega)$ and retains the same labeling as the latter.

Step 3: In $\hat{G}(K_h)$, keep the same label as $x_{ijk} = (u, S_{z_i}, l)$ in $\hat{G}$ for the copy $K_h^{(ijk)}$.

Step 4: In $\hat{G}(K_h)$, a block is formed by copies of $K_h^{(ijk)}$, for a fixed i, all of them having the same label. The total number of labels must equal the total number of blocks, N_b.

Step 5: Apply condition 1 of Theorem 4.1.

★ **Theorem 4.3** *Let $H_{\Delta-1}(K_h)$ be a graph constructed according to Steps 1-4 above, and suppose that $H_{\Delta-1}(K_h)$ satisfies the requirements*

(i) The cardinality of any block is at most $(\Delta-2)^2$. For a fixed i, $(\Delta-2)^2$ is the number of vertices u in W' such that $d(x_i, u) = 2$ (see Figure 4.1), and

(ii) $M' = M - \delta \geq \Delta(G(B_\omega)) = \Delta(G) \times \Delta(B_\omega)$, and

(iii) $h \leq \Delta(B_\omega)+1$

then $H_{\Delta-1}(K_h)$ has maximum degree Δ and diameter 6.

Proof. The second condition of the theorem ensures that the maximum degree of $H_{\Delta-1}(K_h)$ will be at most Δ, hence, we only need to prove that Conditions 2 and 3 of Theorem 4.1 hold, or the diameter is 6 (in those cases where they do not hold).

By virtue of Step 4, and considering that blocks are formed by copies of K_h with the same label, these copies must be at distance three, i.e. Condition 2 of Theorem 4.1 holds.

Now let (x_{ijk}, S_{z_α}) and (x_{rst}, S_{z_β}) be vertices of $\hat{\Gamma}$ such that $d_{\hat{\Gamma}}(x_{ijk}, x_{rst}) = 3$. Then, in $\hat{\Gamma}(K_h)$, copies $K_h^{(ijk)}$ and $K_h^{(rst)}$ belong to different blocks.

Vertices from the same partite set of B_ω are at distance ≤ 2 from one another, since the diameter of (B_ω) is three.

Let us show that the distance between a vertex of $K_h^{(ijk)}$ and a vertex of $K_h^{(rst)}$ is at most 6. We distinguish two cases:

Case 1. $h \leq \Delta(B_\omega)$.

Let $P = (x_{ijk}, S_{z_\alpha})(u, S_{z_\delta})(v, S_{z_\theta})(x_{rst}, S_{z_\beta})$ be a 3-path between x_{ijk} and x_{rst} in $G(B_\omega)$. Then, since vertices from the same partite set of B_ω are at distance at most 2, there are $\Delta(B_\omega)$ paths of length three between x_{ijk} and x_{rst}, each one passing through a different neighbor of x_{ijk} in S_{z_δ} (see Figure 4.11 (a)).

Therefore, connections in $\hat{\Gamma}(K_h)$ can be arranged in such a way that every vertex $y_\gamma^{(ijk)} \in K_h^{(ijk)}$ is joined to a copy $K_h^{(opq)}$ lying at distance 3 from $K_h^{(rst)}$ (see Figure 4.11 (b)). This verifies Condition 3a of Theorem 4.1.

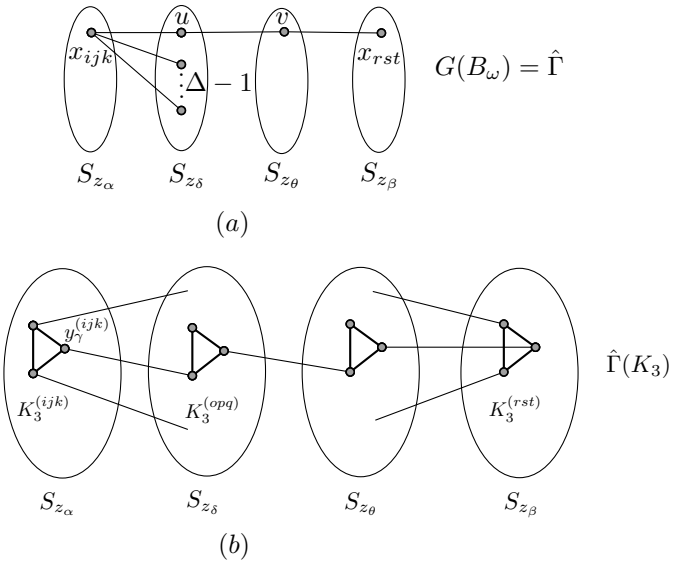


Figure 4.11: A path of length at most 6 between $y_\gamma^{(ijk)} \in K_3^{(ijk)}$ and any vertex in $K_3^{(rst)}$. Note that $d(K_3^{(opq)}, K_3^{(rst)}) = 3$.

Case 2. $h = \Delta(B_\omega) + 1$.

In this case we can arrange the connections in $\hat{\Gamma}(K_h)$ so as to have exactly one vertex in each copy $K_h^{(ijk)}$ not connected to any copy $K_h^{(opq)}$ at distance 3 from $K_h^{(rst)}$. For each copy $K_h^{(ijk)}$, let $y_0^{(ijk)}$ be such a vertex, hence there is a path P of length 6 between $y_0^{(ijk)}$ and $y_0^{(rst)}$ that traverses the subgraph R of $H_{\Delta-1}$ (see Figure 4.1), $P = y_0^{(ijk)} x_{ij} x_i x x_r x_{rs} y_0^{(rst)}$.

Finally, note that Condition 3a holds for all vertices in $K_h^{(ijk)}$ with the exception of $y_0^{(ijk)}$. □

Some of the graphs used to produce graphs $G(B_\omega)$ are:

$I_{\Delta-1}$**:** A bipartite Moore graph of diameter 3 and degree Δ. It is known that those graphs have a polarity [19].

$C_5 * F_4$**:** Maximal graph of degree 3, diameter 3 and order 20, found by Green [56].

BD**:** Bipartite graph of degree 7, diameter 3, and order 78, found by Bondy and Delorme [22]. It also has a polarity.

H**:** The unique graph of degree 3, diameter 3, and order 18, constructed by Faradzhev [64] (see also [128]).

Table 4.1 summarizes the features of each $H_{\Delta-1}(K_h)$ obtained by this method. The orders are computed with the aid of Equation (4.1).

Δ	h	M	δ^*	M'	G	B_ω	$H_{\Delta-1}(K_h)$	Order of $H_{\Delta-1}(K_h)$
9	6	15	3	12	C_7	I_5	$H_8(K_6)$	75893
12	8	28	7	21	$C_5 * F_4$	BD	$H_{11}(K_8)$	359772
14	10	36	9	27	H	I_8	$H_{13}(K_{10})$	816294

*δ is the quotient between the maximum cardinality of a block in $\hat{G}(K_h)$ and $\Delta - 2$ (the maximum number of copies $K_h^{(ijk)}$ having i and j fixed).

Table 4.1: Features of the graphs $H_{\Delta-1}(K_h)$ obtained by the second approach.

Δ	Previous Largest Known Order	New Largest Known Order	Graph
5	2766	**2772**	$H_4(K_3)$
6	7908	**7917**	$H_5(K_4)$
9	75828	**75893**	$H_8(K_6)$
12	359646	**359772**	$H_{11}(K_8)$
14	816186	**816294**	$H_{13}(K_{10})$

Table 4.2: Orders of the new largest known graphs $H_{\Delta-1}(K_h)$ for $\Delta \leq 14$.

4.4 Conclusions

In this chapter we have presented a new family of compound graphs of diameter 6, denoted by $H_{\Delta-1}(K_h)$. At least five members of this family—namely, $H_4(K_3)$, $H_5(K_4)$, $H_8(K_6)$, $H_{11}(K_8)$ and $H_{13}(K_{10})$—are the largest known graphs for their respective maximum degrees and diameter 6 (see Table 4.2). In order to obtain some of these new largest known graphs, we have defined a second new family of compound graphs, denoted by $\Gamma(B_\omega)$, and we proved that their maximum degree is $\Delta(\Gamma) \times \Delta(B_\omega)$, their order is $\frac{|\Gamma||B_\omega|}{2}$, and their diameter is $\max\{D(\Gamma), D(B_\omega)\}$.

5

Algebraic methods in the Degree-Diameter Problem

The main goal of this chapter is to expand on the existing algebraic methods and tools in connection with the Degree-Diameter Problem. In particular, in Section 5.1 we develop the theory of rewriting systems further, and we show how to use these tools in connection with Cayley graphs, which in principle can be generalized to voltage graphs. Then, in Sections 5.2 and 5.3 we derive conditions that enable us to narrow the search space in Algorithm 2, in order to obtain large graphs of different types (e.g. bipartite) by voltage assignment. Finally, in Section 5.4 we discuss some open research problems that we have not been able to cover in this thesis.

Most of the results of Section 5.1 have been published in [27], while the material of Sections 5.2 and 5.3 is the main subject of [49].

5.1 Elimination

In Section 2.4 we saw some of the nice properties of complete string rewriting systems, as well as some of its drawbacks, like the problems associated with computing a complete string rewriting system, which is not always possible. This section expands further on this topic. In particular, we deal with the issue of computing a complete string rewriting system from an existing one, which is a lot easier than computing it from scratch.
Our main tool for this purpose will be *elimination theory*, which describes the properties of some substructure of a given algebraic structure, where an *elimination ordering* has been defined. Elimination has been widely studied in commutative polynomial rings in connection with Gröbner bases, since it enables to obtain the Gröbner basis of an elimination ideal with little computational effort, provided that a Gröbner basis for a super-ideal is known. Given its importance, the concept has been generalized to other settings, like free associative algebras over a field [25]. In this section we formalize the notion of elimination in the context of string rewriting systems, and we show how to use this property to solve the *generalized word problem*. The results contained in this section have been published in our paper [27], and this section

follows approximately the same structure: Subsection 5.1 discusses the properties of elimination orderings in connection with string rewriting systems, and how to use this connection to solve the generalized word problem. Then, Subsection 5.1.2 briefly mentions some applications of the preceding theory to solve other computational problems in groups, such as the Discrete Logarithm Problem in finite groups, finding the inverse of an element in a group, and finding a new set of generators for a group. The latter application is of interest for dealing with Cayley graphs.

5.1.1 The elimination property

Let $\Sigma = \{x_1, x_2, \dots x_n\}$, and let Σ_k stand for $\{x_1, x_2, \dots x_k\}$, where $1 \leq k \leq n-1$. Throughout this chapter we will assume that $x_1 < x_2 < \dots x_n$.

Definition 5.1 (Term ordering with the elimination property) *Given the term ordering $<$ on Σ^*, we say that it has the elimination property at position k (with k as above) if, for every $v, w \in \Sigma^*$, $v < w$ and $w \in \Sigma_k^*$ implies $v \in \Sigma_k^*$. An ordering with the elimination property is called an* **elimination ordering**.

Mora's ordering (see Section 2.4) is an example of a term ordering with the elimination property; it possesses the property at position k, for all $1 \leq k \leq n-1$. The elimination property is very convenient when computing complete systems. Informally, if we have a complete rewriting system on Σ with respect to an elimination ordering, then it is easy to get a complete system on Σ_k: we simply remove the rules containing letters not in Σ_k, i.e. $x_{k+1}, \dots x_n$. We formalize this idea in the sequel.
Let us consider the restriction $<_k$ of $<$ to Σ_k^*, where $<$ is a term ordering on Σ^*, and let $R_k = R \cap (\Sigma_k^* \times \Sigma_k^*)$.

★ **Lemma 5.1** *As above, let $<$ be a term ordering on Σ^*, with the elimination property at position k. Then, $<_k$ is also a term ordering, and it is compatible with R_k.*

Proof. $<_k$ is obviously an admissible, well-founded partial ordering. Let $\alpha \to \beta$ be an arbitrary rule of $R_k \subseteq R$. We have that $\beta < \alpha$ and $\alpha, \beta \in \Sigma_k^*$, hence $\beta <_k \alpha$. □

★ **Lemma 5.2** *If $u \xrightarrow{*}_R v$ and $u \in \Sigma_k^*$ then $u \xrightarrow{*}_{R_k} v$.*

Proof. If $u \xrightarrow{*}_R v$ then $u \to_R u' \to_R u'' \to_R \cdots \to_R v$. Since $u > u' > u'' > \cdots > v$ and $u \in \Sigma_k^*$, we have that $u', u'', \dots, v \in \Sigma_k^*$, and hence $u \to_{R_k} u' \to_{R_k} u'' \to_{R_k} \cdots \to_{R_k} v$. □

★ **Theorem 5.1 (Elimination theorem)** *Let $\langle \Sigma : R \rangle$ be a complete rewriting system that is compatible with the term ordering $<$, which possesses the elimination property at position k. Let J denote the congruence $\equiv_R$ and $J_k = J \cap (\Sigma_k^* \times \Sigma_k^*)$. Then, $\langle \Sigma_k : R_k \rangle$ is also a complete rewriting system, and J_k coincides with the congruence $\equiv_{R_k}$.*

Proof Lemma 5.1 implies that R_k is noetherian; now let us check for local confluence. Assume that for $u, v, w \in \Sigma_k^*$, $u \to_{R_k} v$ and $u \to_{R_k} w$; this implies that $u \to_R v$ and $u \to_R w$, and since R is locally confluent, $v \downarrow_R w$, i.e. $v \xrightarrow{*}_R z$ and $w \xrightarrow{*}_R z$ for some $z \in \Sigma_k^k$. By Lemma 5.2, $v \xrightarrow{*}_{R_k} z$ and $w \xrightarrow{*}_{R_k} z$, i.e. $v \downarrow_{R_k} w$. Hence $\langle \Sigma_k : R_k \rangle$ is complete. The coincidence between J_k and $\equiv_{R_k}$ follows directly from the definition of J_k. □

In Theorem 5.1, we call R_k an *elimination system* of R, and J_k an *elimination congruence* of J. This theorem is the basis for a computational procedure that solves some instances of the generalized word problem (procedure IN-SUBGROUP).

Algorithm 6: Procedure IN-SUBGROUP

Input : A finite monoid presentation $\langle \Sigma : R \rangle$ for a group Γ, a set of words $W = \{w_1, w_2, \ldots, w_k\}$, with $w_i \in \Sigma^*$, defining $H \trianglelefteq \Gamma$, and another word $\alpha \in \Sigma^*$.

Output: The procedure decides whether $[\alpha]_R \in H$, and if so, it gives the expression of α in terms of the w_i.

1 Introduce a set of new letters, $Y = \{y_1, \ldots, y_k\}$, for tagging the elements of W ;
2 Choose a term ordering $<$ on $(\Sigma \cup Y)^*$, with the elimination property at position k, so that $Y < \Sigma$, i.e. for every $y \in Y$ and every $x \in \Sigma$, $y < x$;
3 Add to the presentation of Γ all the rules $y_i \to w_i$, and apply the Knuth-Bendix procedure to the resulting system, with respect to $<$;
4 Compute $NF(\alpha)$ modulo the complete system obtained ;
5 $[\alpha]_R \in H$ if, and only if, $NF(\alpha)$ depends solely on the y_i. The expression of α in terms of the w_i is obvious ;

A key step of our procedure is step 3: the procedure terminates if, and only if, Knuth-Bendix completion does. This is the case, for example, when the group Γ is finite or abelian. In general, it is impossible to predict beforehand whether the Knuth-Bendix completion procedure will terminate on a given input.

5.1.2 Some applications

In this subsection we show how to apply the procedure IN-SUBGROUP for solving three algorithmic problems related with finitely presented groups. They were first given in [26].

The *Discrete Logarithm Problem* for finitely presented groups (DLP for short) has been extensively studied, due to its applications in Cryptography, mainly in multiplicative groups of finite fields and groups of points of elliptic curves over finite fields [94]. The setting in which the problem is presented here is rather unusual. The DLP can be stated as follows:

Problem 5.1 (The Discrete Logarithm Problem in finitely presented groups) *Let $\langle \Sigma : R \rangle$ be a finite monoid presentation for a group Γ.*
Instance: *Two words $b, c \in \Sigma^*$ (b is called the base element).*
Question: *Does there exist a positive integer k such that $b^k = c$? If the answer is positive, find k.*

This problem has also been known in the literature as the **power problem** (see e.g. [111]). It is not difficult to realize that k exists if, and only if, $NF(c)$ is in the subgroup generated by b, and this fact can be tested with the aid of the procedure IN-SUBGROUP.

Another problem that appears in finitely presented groups in connection with the generalized word problem is the **Inverse Problem** (IP):

Problem 5.2 (The Inverse Problem) *Let $\langle \Sigma : R \rangle$ be a finite monoid presentation of a group G.*
Instance: *A word $\alpha \in \Sigma^*$.*
Question: *Is it possible to find a word $\beta \in \Sigma^*$ such that $[\beta]_R = [\alpha]_R^{-1}$?*

Procedure INVERSE leads to the solution of the problem:

Algorithm 7: Procedure INVERSE

1 Introduce a new symbol y that will act as a label for the inverse of α ;
2 Choose a term ordering $<$ on $(\Sigma \cup \{y\})^*$, with the elimination property at position n, so that $\Sigma < y$;
3 Perform a Knuth-Bendix completion on $R \cup \{y\alpha \to \mathbf{1}, \alpha y \to \mathbf{1}\}$;
4 Then β is the right-hand side of the rule whose head is y ;

The proof that this procedure works can be found in [26]; it makes use of the theory of Gröbner bases in non-commutative algebras.

Our last application deals with the problem of finding a different **Generating Set** for a given finitely presented group.

Problem 5.3 (Generation Problem) *Let a finite monoid presentation $\langle \Sigma : R \rangle$ of a group G be given.*
Instance: *$W = \{w_1, w_2, \dots w_k\}$, with $w_i \in \Sigma^*$ for all $1 \leq i \leq k$.*
Question: *Does W generate G?. If so, compute $\{z_1, z_2 \dots, z_n\} \subset W^*$ such that $[x_i]_R = [z_i]_R$ for all $1 \leq i \leq n$.*

This problem amounts to deciding, for each generator x_i of Γ, whether x_i lies in the subgroup of G generated by W. If W is to generate Γ, then, after the first three steps of the procedure IN-SUBGROUP, we should have each x_i as the left-hand side of some rule in the resulting complete system. Moreover, all the rules containing a symbol of Σ must be of the form $x_i \to q_i$, where $q_i \in Y^*$.

Thus, as a bonus we get rules for transforming any word of Σ^* to a word in the w_i, and we also get a complete presentation for the group in terms of the w_i. In Cayley graphs, this is useful for changing the generating set, e.g. to look for a generating set that yields a smaller diameter.

5.2 Finite metacyclic and metabelian groups

In [51], Dinneen and Hafner suggested the use of semidirect products of cyclic groups, like $\mathbb{Z}_n \ltimes_r \mathbb{Z}_m$, $\mathbb{Z}_n \ltimes_\psi (\mathbb{Z}_m \times \mathbb{Z}_m)$, and $(\mathbb{Z}_n \ltimes_r \mathbb{Z}_m) \ltimes (\mathbb{Z}_n \ltimes_r \mathbb{Z}_m)$ to obtain large Cayley graphs. Those are precisely the groups that were used in [107] and [106] to update the table of the largest known graphs by voltage assignment. Even though those groups are not very far from abelian, they have the advantage that they are easy to generate and use in computation.
In this section we investigate these groups. We are interested in the order of the elements, which influences the length of the cycles resulting in the voltage assignment construction, and in turn, determines other graph properties, like bipartiteness.
Given two groups, H and N, and a homomorphism $\psi : H \to Aut(N)$, the (external) semidirect product $H \ltimes_\psi N$ can be defined on the set $H \times N$ as $(h_1, n_1) * (h_2, n_2) = (h_1 h_2, n_1 \psi_{h_1}(n_2))$. So defined, N is a normal subgroup of $\Gamma = H \ltimes_\psi N$, and H is a subgroup of Γ. Every element of $\Gamma = H \ltimes_\psi N$ can be represented uniquely as hn, where $h \in H, n \in N$; this is the so-called internal semidirect product. The internal and external definitions are equivalent, and we may alternate between the two as needed.
If H and N are both cyclic, then the semidirect product $\Gamma = H \ltimes_\psi N$ is *metacyclic*. A group Γ is metacyclic if it has a cyclic normal subgroup N, such that Γ/N is also cyclic. The complete characterization of metacyclic groups was first given in [81]. We will first concentrate on metacyclic groups, so as to pave the way to extend our results to other classes of groups later.

Let $H = \mathbb{Z}_n$, and $N = \mathbb{Z}_m$; then $Aut(\mathbb{Z}_m)$ is isomorphic to the multiplicative group of units of the ring $(\mathbb{Z}_m, \oplus, \odot)$, which has order $\phi(m)$, where $\phi(m)$ denotes Euler's totient function, and $\oplus$, $\odot$ denote the addition and multiplication modulo m, respectively. Every automorphism of $\mathbb{Z}_m$ has the form $x \to x^r$, where r and m are relatively prime.
In additive notation, the operation $*$ in the group $\Gamma = \mathbb{Z}_n \ltimes_r \mathbb{Z}_m$ becomes

$$(\alpha_1, \beta_1) * (\alpha_2, \beta_2) = (\alpha_1 + \alpha_2 \pmod n, \quad \beta_1 + \beta_2 r^{\alpha_1} \pmod m) \tag{5.1}$$

and from there we can compute the k-th power of any element (α, β):

$$(\alpha, \beta)^k = \left(k\alpha \pmod n, \quad \beta \frac{r^{k\alpha} - 1}{r^\alpha - 1} \pmod m \right) \tag{5.2}$$

Note that, with the notation of Equation 5.1, $(\alpha, 0) * (0, \beta) = (\alpha, \beta r^\alpha)$, and $(0, \beta) * (\alpha, 0) = (\alpha, \beta)$.

The *multiplicative order* of r modulo m, denoted $\mathrm{ord}_m(r)$, is the smallest positive integer t such that $r^t \equiv 1 (\mathrm{mod}\ m)$. In this case, since we are assuming that $r^n \equiv 1 (\mathrm{mod}\ m)$, t must be a divisor of n.

We have the following

★ **Proposition 5.1** *Let $\Gamma = \mathbb{Z}_n \ltimes_r \mathbb{Z}_m$, where $r^n \equiv 1 (mod\ m)$, $t = ord_m(r)$, and let $o(\gamma)$ denote the order of $\gamma \in \Gamma$. Then,*

$$o(\alpha, \beta) \text{ divides } k = \mathrm{lcm}\left(\frac{n}{\gcd(n, \alpha)}, \frac{mt}{\gcd(m, \beta)\gcd(t, \alpha)} \right)$$

Proof: From Equation 5.2 we see that $(\alpha,\beta)^k$ is the identity of $\mathbb{Z}_n \ltimes_r \mathbb{Z}_m$ if, and only if, the following system of equations are satisfied:

$$k\alpha \equiv 0 \pmod{n}, \tag{5.3}$$

$$\beta\frac{r^{k\alpha}-1}{r^{\alpha}-1} \equiv 0 \pmod{m} \tag{5.4}$$

With the above definition of k, it is obvious that Equation 5.3 is satisfied, so we only have to check that Equation 5.4 is satisfied as well. We are going to show that $\beta\frac{r^{k\alpha}-1}{r^{\alpha}-1} \equiv 0 (\text{mod } m)$. Let $m' = \frac{m}{\gcd(m,\beta)}$. Note that α divides $\frac{t\alpha}{\gcd(t,\alpha)}$, and $\frac{m't\alpha}{\gcd(t,\alpha)}$ divides $k\alpha$, therefore

$$\frac{r^{\frac{m't\alpha}{\gcd(t,\alpha)}}-1}{r^{\frac{t\alpha}{\gcd(t,\alpha)}}-1} \text{ divides } \frac{r^{k\alpha}-1}{r^{\alpha}-1}$$

So, it suffices to show that m' divides $\frac{r^{\frac{m't\alpha}{\gcd(t,\alpha)}}-1}{r^{\frac{t\alpha}{\gcd(t,\alpha)}}-1}$. Now,

$$\frac{r^{\frac{m't\alpha}{\gcd(t,\alpha)}}-1}{r^{\frac{t\alpha}{\gcd(t,\alpha)}}-1} = \sum_{j=0}^{m'-1} r^{\frac{jt\alpha}{\gcd(t,\alpha)}}$$

Each exponent is divisible by t, hence each term in the sum is congruent with 1 (mod m), and congruent with 1 (mod m'). There are m' terms, hence the sum is divisible by m', and so is $\frac{r^{k\alpha}-1}{r^{\alpha}-1}$. Consequently, $\beta\frac{r^{k\alpha}-1}{r^{\alpha}-1}$ is divisible by m, as claimed.

□

Sometimes it is also useful to view the group in multiplicative notation. A complete monoid presentation of $\Gamma = \mathbb{Z}_n \ltimes_r \mathbb{Z}_m$ is $\langle a, b : a^n = b^m = \lambda, ba = ab^r\rangle$. Therefore, every element of $\mathbb{Z}_n \ltimes_r \mathbb{Z}_m$ has a unique normal form $a^\alpha b^\beta$, with $0 \le \alpha \le n-1$, and $0 \le \beta \le m-1$, modulo this presentation.

With this notation, the product of $a^\alpha \in \mathbb{Z}_n$ with $b^\beta \in \mathbb{Z}_m$ reduces to concatenation. The k-th power of $a^\alpha b^\beta$ is

$$(a^\alpha b^\beta)^k = a^{k\alpha} b^{\beta\frac{r^{k\alpha}-1}{r^{\alpha}-1}} \tag{5.5}$$

which agrees with Equation 5.2. From Equation 5.3 it is obvious that $o(a^\alpha)$ divides $o(a^\alpha b^\beta)$.

Proposition 5.1 gives an upper bound for the order of an element $(\alpha,\beta) \in \mathbb{Z}_n \ltimes_r \mathbb{Z}_m$. A general upper bound is given by the *exponent* of $\mathbb{Z}_n \ltimes_r \mathbb{Z}_m$, i.e. the largest order of any element, which is $\text{lcm}(n,m)$ (see [81], Lemma 2.1).

Metacyclic groups belong to the larger family of *metabelian groups*. A group Γ is metabelian if it has an abelian normal subgroup N such that Γ/N is also abelian. In particular, semidirect products of the form $H \ltimes_\psi N$, with H and N abelian, are metabelian. In turn, finitely generated metabelian groups are *polycyclic*. A group is polycyclic if it admits a *subnormal series*

$$\Gamma = \Gamma_1 \triangleright \Gamma_2 \triangleright \cdots \triangleright \Gamma_n \triangleright \Gamma_{n+1} = \{1\} \tag{5.6}$$

where each Γ_{i+1} is a normal subgroup of Γ_i, and the quotient Γ_i/Γ_{i+1} is cyclic, for all $1 \leq i \leq n$. The nice thing about polycyclic groups is that there exist efficient algorithms to handle them.

If every quotient Γ_i/Γ_{i+1} in the series 5.6 is abelian, then the group is *solvable*, and the length of the shortest such series (e.g. n in Equation 5.6) is the *derived length*. The derived length of a solvable group Γ is a measure of its commutativity: the longer the series, the less commutative Γ is.

Back to polycyclic groups, a *polycyclic generating sequence* is a sequence $(\gamma_1, \gamma_2, \ldots, \gamma_n)$ such that $\Gamma_i = \langle \Gamma_{i+1}, \gamma_i \rangle$, for all $1 \leq i \leq n$. Now let Γ be finite, and let r_i be the index of Γ_{i+1} in Γ_i. Then, every element $\gamma \in \Gamma$ can be represented uniquely in the normal form

$$\gamma_1^{e_1} \gamma_2^{e_2} \cdots \gamma_n^{e_n}, \tag{5.7}$$

where $0 \leq e_i \leq r_i - 1$. A complete presentation corresponding to the generators $(\gamma_1, \gamma_2, \ldots, \gamma_n)$ is

$$\gamma_i^{r_i} = \gamma_{i+1}^{a(i,i,i+1)} \cdots \gamma_n^{a(i,i,n)}, \quad \text{for } 0 \leq i \leq n \tag{5.8}$$

$$\gamma_i^{-1} \gamma_j \gamma_i = \gamma_{i+1}^{a(i,j,i+1)} \cdots \gamma_n^{a(i,j,n)}, \quad \text{for } 0 \leq i < j \leq n, \tag{5.9}$$

where $0 \leq a(i,j,k) \leq r_k - 1$, and the right-hand sides are words in normal form. The normal form of any word w can be obtained by repeated application of the rules above, however some shortcuts have been devised for that purpose; they are called *collection* algorithms (see [86], p. 280).

Now let us go back to the special case of finite metabelian groups. Let H and N be finite abelian groups, with generating sets $A = \{a_1, \ldots, a_n\}$ and $B = \{b_1, \ldots, b_m\}$, respectively, and let us assume that a semidirect product $H \ltimes_\psi N$ can be constructed. Assume also that we have polycyclic presentations of H and N on the generating sets A and B.

Every automorphism of N has the form $\psi_a : a^{-1}ba \to \omega_b$, where $\omega_b = b_1^{e_1} b_2^{e_2} \cdots b_m^{e_m}$ is a word in normal form with respect to the presentation of N. The union of the polycyclic presentations of H and N, with the relations $b_j a_i \to a_i \omega_{b_j}$, for $1 \leq i \leq n, 1 \leq j \leq m$, yields a polycyclic presentation of $H \ltimes_\psi N$. With the aid of this presentation, every element of $H \ltimes_\psi N$ can be represented uniquely as $a_1^{\alpha_1} a_2^{\alpha_2} \cdots a_n^{\alpha_n} b_1^{\beta_1} b_2^{\beta_2} \cdots b_m^{\beta_m}$, where the exponents are as in Equation 5.7.

The k-th power of an element has the form

$$(a_1^{\alpha_1} a_2^{\alpha_2} \cdots a_n^{\alpha_n} b_1^{\beta_1} b_2^{\beta_2} \cdots b_m^{\beta_m})^k = a_1^{k\alpha_1} a_2^{k\alpha_2} \cdots a_n^{k\alpha_n} b_1^{\beta'_1} b_2^{\beta'_2} \cdots b_m^{\beta'_m} \tag{5.10}$$

where β'_i is a multiple of β_i for all i. Therefore we have the following

★ **Proposition 5.2** *Let H and N be finite abelian groups, and let us assume that a semidirect product $H \ltimes_\psi N$ can be constructed. Also let $h \in H$ and $n \in N$. Then $o(h)$ divides $o(hn)$.*

□

Metabelian groups include the groups of the form $\mathbb{Z}_n \ltimes (\mathbb{Z}_m \times \mathbb{Z}_m)$, which were successfully used in [51], and later in [107], to obtain large graphs with given degree and diameter. Applying Proposition 5.2 to the product of several elements, we get the following

★ **Corollary 5.1** *Let H and N be abelian, and let us assume that a semidirect product $H \ltimes_\psi N$ can be constructed. Also let $(h, n) = (h_1, n_1) * (h_2, n_2) * \cdots * (h_k, n_k)$, with $h, h_1, \ldots, h_k \in H$ and $n, n_1, \ldots, n_k \in N$. Then $o(h_1 h_2 \cdots h_k)$ divides $o(hn)$.*

□

In Section 5.3.2 we discuss how to apply this corollary to obtain large bipartite graphs by voltage assignment.

5.3 Bipartite lifts

For practical applications, bipartite graphs form an important subclass of the class of all graphs. There are several papers dealing with properties and generation of large bipartite graphs with given degree and diameter, e.g. [145].

A graph is bipartite iff all its cycles are even. From the discussions in Section 2.5 it follows that G^α will be bipartite iff every minimal closed walk in G, with net voltage equal to the identity, has even length, or iff every odd closed walk in G has a net voltage with even order. However, these conditions are not very useful from the algorithmic point of view, so we will look into necessary and/or sufficient conditions for G^α to be bipartite.
If G^α is bipartite then, for every cycle C of G, of length k, net voltage g, and n being the order of g in Γ, the product kn is even. Yet, this condition is not sufficient. If G is a bouquet, this amounts to every voltage having even order in Γ, which again, is not sufficient, as shown by the following example.

Example 5.1 *Let G be a bouquet with two loops, with voltages 1 and 2 in $\mathbb{Z}_4$, as shown in Figure 5.1. Both elements, 1 and 2, have even order in $\mathbb{Z}_4$ (4 and 2, respectively). However, we have a closed walk with net voltage equal to the identity of $\mathbb{Z}_4$ and odd length, namely: 2+1+1. That results in an odd cycle in G^α.*

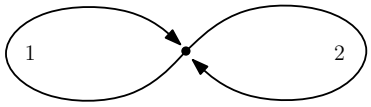


Figure 5.1: Voltage graph producing odd cycles

One sufficient condition for G^α to be bipartite is that G is bipartite. By imposing more restrictions on G and Γ we can get additional conditions, as in the next subsection.

5.3.1 Bipartite circulant graphs

Now let us concentrate on the case of bouquets as base digraphs. We already know that all the voltages must have even order in Γ, but that condition is not sufficient, except maybe for the trivial case, when we have only one loop. Let us start with the simplest non-trivial case, i.e. two loops, and voltages in the cyclic group $\mathbb{Z}_n$. Let the voltages on the loops be a and b. Since the voltage group is abelian, a closed walk is a sequence of p cycles of type a and q cycles of type b. We must find all such combinations that yield the identity element of $\mathbb{Z}_n$, i.e. we must solve the set of linear diophantine equations

$$pa + qb \equiv 0 \pmod{n}, \text{ or} \tag{5.11}$$
$$pa + qb = jn, \text{ with } p, q, j \in \mathbb{Z} \tag{5.12}$$

Let $d = \gcd(a, b)$. Equation (5.12) has a solution iff $d \mid jn$, and in that case it has infinitely many solutions of the form

$$p = \overline{p} + \frac{b}{d}t, \tag{5.13}$$
$$q = \overline{q} - \frac{a}{d}t, \tag{5.14}$$

where $(\overline{p}; \overline{q})$ is any particular solution and t runs through all the integers. For the expansion to be bipartite, $p + q$ must always be even. Adding Equations (5.13) and (5.14) we get

$$p + q = \overline{p} + \frac{b}{d}t + \overline{q} - \frac{a}{d}t = \overline{p} + \overline{q} + \frac{b-a}{d}t \tag{5.15}$$

Since t can be any integer, $\overline{p} + \overline{q}$ and $\frac{b-a}{d}$ must be both even. A particular solution $(\overline{p}; \overline{q})$ can be found by simply letting $\overline{q} = 0$, and taking $\overline{p}$ as the order of a (or viceversa). This implies that the order of both a and b must be even.

Example 5.2 (Example 5.1 revisited) *In Example 5.1 we had a cyclic group of even order, and both voltages had even order, but* $d = \gcd(1, 2) = 1$*, and hence* $\frac{2-1}{1} = 1$ *is not even.*

In the more general case, when we have $k > 2$ loops, with voltages $a_1, a_2, \ldots, a_k$, we get the set of linear diophantine equations

$$a_1x_1 + a_2x_2 + \cdots + a_kx_k \equiv 0 \pmod{n}, \text{ or} \tag{5.16}$$
$$a_1x_1 + a_2x_2 + \cdots + a_kx_k = jn, \text{ with } j \in \mathbb{Z} \tag{5.17}$$

As in the two-variable case, the *general linear diophantine equation* in k variables

$$a_1x_1 + a_2x_2 + \cdots + a_kx_k = c_k, \tag{5.18}$$

has a solution iff $\gcd(a_1, \ldots, a_k)$ divides c.
Nevertheless, a simpler way to approach this problem is by noting that the lift of a bouquet by $\mathbb{Z}_n$ is a *circulant graph*, and bipartiteness of circulant graphs was completely characterized in [82].

Definition 5.2 *Let $n, k, a_1, a_2, \ldots, a_k$ be positive integers, and let's assume that $a_i \not\equiv \pm a_j \pmod{n}$ for $i \neq j$. A graph with vertex set $V = \{0, 1, \ldots, n-1\}$ and edge set $E = \{(i;\ i + a_j \bmod n) : i = 0, \ldots, n-1,\ j = 1, \ldots, k\}$ is called a **circulant graph**, and is denoted $C_n(a_1, \ldots, a_k)$. A circulant graph is a Cayley graph of the cyclic group $\mathbb{Z}_n$.*

A circulant graph $G = C_n(a_1, \ldots, a_k)$ is connected iff $\gcd(n, a_1, \ldots, a_k) = 1$. In general, each connected component of G is circulant, and isomorphic to $C_{\frac{n}{d}}(\frac{a_1}{d}, \ldots, \frac{a_k}{d})$, where $d = \gcd(n, a_1, \ldots, a_k)$. Hence, in the sequel we can assume w.l.o.g. that G is connected.

Theorem 5.2 ([82]) *Let $G = C_n(a_1, \ldots, a_k)$ be a connected circulant graph. Then G is bipartite if, and only if,*

$$a_1, \ldots, a_k \text{ are odd, and } n \text{ is even.} \tag{5.19}$$

In order to apply Theorem 5.2 to our problem, just note that the integers $a_1, \ldots, a_k$ are precisely the voltages in our voltage graph. First we have to identify the connected components, and then apply Condition 5.19 in a straightforward manner to each connected component.

Example 5.3 (Example 5.1 revisited) *Going back to Example 5.1, we see that the derived graph is connected, hence we can apply Theorem 5.2 directly. The voltage group has even order, but there is one even voltage (2), hence the derived graph is not bipartite.*

A discussion about how to identify the connected components of the derived graph is given in [77], pp. 86-93.

5.3.2 Bipartite lifts by metabelian groups

In the case of a finite metabelian groups of the form $H \ltimes_\psi N$, with H and N finite abelian, Corollary 5.1 gives us a sufficient condition that we can use to determine *a priori* whether the resulting derived graph will be bipartite. Let W be a closed walk in G of length k (odd), with net voltage $(h, n) = (h_1, n_1) * (h_2, n_2) * \cdots * (h_k, n_k)$, where $h, h_1, \ldots, h_k \in H$ and $n, n_1, \ldots, n_k \in N$. If $h_1 h_2 \cdots h_k$ has even order in H (and consequently in Γ), then the order of (h, n) is also even.

Such conditions have been integrated into *sieves*, which can be plugged in as additional if-statements in Step 7 of Algorithm 2 (Section 2.5.3). The sieves prune the search space, and thus make the search tractable. Some more details concerning the sieves and the implementation of the algorithm are given in [103], but these particular sieves to get bipartite lifts are not discussed.

In principle, this technique can be adapted to obtain graphs of a particular type (e.g. bipartite, planar, vertex-transitive, etc.). We just have to add a function that tests whether the resulting lift belongs to the desired class. The key observation is that in many cases this check can

performed *before* actually constructing the lift. In the case of bipartite graphs, the results of Sections 5.2 and 5.3 provide sieving conditions that make the search more efficient, by discarding a large number of lifts before actually constructing them.

With the aid of this modified procedure, new record bipartite graphs have been found. The updated table of bipartite graphs is shown in Table 5.1. The table shows the orders of the largest known bipartite graphs. The entries in bold are the ones that have been obtained with the aid of our methods. For more details see [103] and [63]; there are adjacency lists, the actual voltage assignments used to obtain the graphs, references, etc. Note that the values at (14,8) and (15,8) are the same, but the latter can be pushed a little bit further by vertex duplication [48].

The table appears in [103] and [63], while the results about metabelian groups, and their application to get sieves for constructing large bipartite graphs, are contained in the paper [49].

Δ/D	3	4	5	6	7	8	9	10
3	14	30	56	126	168	256	**506**	800
4	26	80	160	728	840	**2 184**	**4 970**	**11 748**
5	42	170	**336**	2 730	**3 110**	**9 234**	**27 936**	**90 068**
6	62	312	**684**	7812	8310	**29 790**	**117 360**	**452 032**
7	80	346	**1134**	8992	23436	**80 940**	**400 160**	**1 987 380**
8	114	800	**1710**	39216	40586	**201 480**	**1 091 232**	**6 927 210**
9	146	1170	**2496**	74898	117648	**449 480**	**2 961 536**	**20 017 260**
10	182	1 640	4 000	132 860	224 694	1 176 480	**7 057 400**	**50 331 156**
11	186	1 734	5 850	142 464	398 580	2 246 940	**15 200 448**	**130 592 354**
12	266	2 928	8 200	354 312	664 300	4 650 100	**30 001 152**	**300 383 050**
13	270	3 064	11 480	374 452	1 062 936	5 314 680	**50 990 610**	**617 330 936**
14	366	4 760	14 760	804 468	1 771 560	14 172 480	**95 087 738**	**1 213 477 190**
15	370	4 946	20 496	842 048	2 480 184	14 172 480	**168 016 334**	**2 300 326 510**
16	394	5 134	27 300	884 062	4 022 340	36 201 060	**288 939 118**	**4 119 507 330**

Table 5.1: Orders of the largest known bipartite graphs

5.4 Some open problems

By now it is a widely documented fact that the farther a group is from abelian, the better results we can get with it [51, 53, 104, 60, 62]. Since metabelian groups are not very far from abelian (they are solvable with derived length 2), it is reasonable to expect that our results can be improved by using other solvable groups with a greater derived length, or outright unsolvable.

Exoo and Jajcay use *perfect groups* to construct cages [62]. A group is perfect if it has no nontrivial abelian quotients. In particular, *simple groups* are perfect (a group is simple if it has no nontrivial quotients of any kind). It would be interesting to see what results we can get with perfect groups in the Degree-Diameter Problem. The next step could be simple groups,

if we can find an efficient way to handle them.
In any case, it is important to determine the scope and limits of our methods: What fraction of bipartite graphs can be obtained this way? This is a fundamental question that remains unanswered.
Let $G = (V, E)$ be a graph, and $Aut(G)$ its automorphism group. A permutation group Γ acting on V is called *free*, or *semiregular*, if no vertex of G remains fixed by any non-trivial permutation of Γ. If $Aut(G)$ contains a subgroup Γ that is semiregular, then G can be obtained as a lift of a smaller graph.
The classification of graphs possessing a semiregular group of automorphisms is a significant topic in its own right. To guarantee the existence of the desired group of automorphisms, one common approach is to find a *semiregular automorphism* (a fixed-point-free automorphism) in the lift. Research efforts in this direction have only targeted vertex-transitive graphs. It has been conjectured that all vertex-transitive graphs possess a semiregular group of automorphisms. This conjecture is often called the *polycirculant conjecture*, and in the form of a research problem is now known as the *semiregularity problem* [100]. The conjecture is valid for Cayley graphs, but still remains open for arbitrary vertex-transitive graphs.
The existence of a semiregular group of automorphisms for a class of graphs can also be used to enumerate all the graphs of the class [113]. Settling this conjecture will heavily advance the theory of vertex-transitive graphs. In particular, it will contribute significantly to answering the question of whether every connected vertex-transitive graph has a hamiltonian path [100]. In this case, we need to investigate the validity of the polycirculant conjecture for bipartite graphs, vertex-transitive or not.
If we wanted to extend our techniques to other kinds of graphs, similar questions arise: e.g. how can we generate vertex-transitive (or planar) graphs?, can all planar graphs be obtained by voltage assignment from a smaller graph?, etc.
In the realm of group theory, another fundamental problem is to determine the order of a product of two elements, given the order of an element. As we mentioned before, if γ_1 and γ_2 commute, $o(\gamma_1\gamma_2)$ divides $\text{lcm}(o(\gamma_1), o(\gamma_2))$. However, if γ_1 and γ_2 do not commute, anything can happen: For any positive integers $i, j, k > 1$, there exists a (non-abelian) group Γ, and elements $\gamma_1, \gamma_2 \in \Gamma$, such that $o(\gamma_1) = i$, $o(\gamma_2) = j$, and $o(\gamma_1\gamma_2) = k$ (e.g. the triangle groups). So, the best we can do is investigate this problem in *classes of groups* in an ad hoc manner.
In the case of a semidirect product of abelian groups we have obtained some bounds for the order of a product, but can we get more precise? This would lead to better sieves for our algorithm, and hence we could narrow our search space even further.
In analogy with graph theory, define the *order sequence* of a group as the ordered list of all the element orders. If two groups are isomorphic, it is obvious that they have the same order sequence, but the converse is not true in general. In the case of abelian groups, the order sequence determines the group up to isomorphism [15]. However, we can find two metacyclic groups that are non-isomorphic but have the same order sequence. Therefore, we would like to know if there are other classes of groups that are completely determined by their order sequence.
Kavitha [92] gives an algorithm for computing in linear time the order sequence of a group given by its multiplication table, and he uses the order sequence to determine nilpotence of a group, and isomorphism of two abelian groups in linear time. However, the multiplication table is usually not given, therefore it would be interesting to extend these methods to other representation methods, like finite presentations.

Part III

The Degree-Diameter Subgraph Problem

6

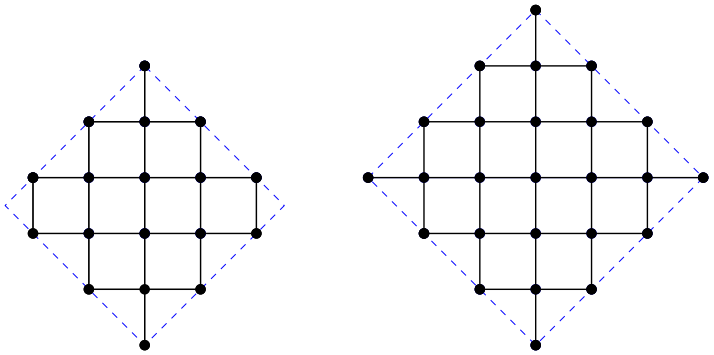
The Degree-Diameter Subgraph Problem in the mesh

In Chapter 3 we introduced the Degree-Diameter Subgraph Problem, discussed its computational complexity, and its relationship with the 'classical' Degree-Diameter Problem. In this chapter we study the Degree-Diameter Subgraph Problem in an important class of host graphs: the k-dimensional mesh or grid. The results of this chapter have been published in [117].

6.1 DDS in the k-dimensional mesh

In this chapter we will assume that the host graph G is an infinite k-dimensional mesh, and we are looking for a subgraph of maximum degree $\Delta \leq 2k$, and diameter D. We can associate our mesh with an L^1 metric space in dimension k. Pick an arbitrary point in this k-dimensional L^1 metric space as the center of a coordinate system. Now, the vertices of the mesh are the points with integer coordinates (*lattice points*), and two lattice points are joined iff they are at distance 1. The largest subgraph of degree $\Delta = 2k$ and diameter D corresponds to a closed ball of radius $D/2$. The number of lattice points contained in a ball of radius $D/2$ is variable, and depends on the location of the center of the ball. The maximum number of lattice points is achieved when the center of the ball is a lattice point itself, for even D, and when the center of the ball is the midpoint between two adjacent lattice points, for odd D. Such balls will be called *maximal*, and will be denoted $B_k(p)$, where $p = \lfloor \frac{D}{2} \rfloor$, depending on its parity, and $k \geq 1$. Figure 6.1 depicts two maximal balls in dimension two, with diameters 5 and 6, respectively.

For more details about the shape and location of maximal balls, we refer to the comprehensive study by Dougherty and Faber [53], where the same problem appears in a slightly different context. In that paper, the infinite k-dimensional mesh is interpreted as the Cayley graph of the free abelian group $\mathbb{Z}^k$, and the points with integer coordinates contained in the ball $B_k(p)$



(a) (b)

Figure 6.1: Maximal balls in the two-dimensional L^1 metric space

correspond to words of length at most D in the canonical generators of $\mathbb{Z}^k$. The aim of [53] was to construct large Cayley graphs on abelian groups, with given degree and diameter.

In order to simplify notation, we will also use $B_k(p)$ to denote the set of points with integer coordinates contained in the closed ball $B_k(p)$. The order of the largest subgraph S of degree $\Delta \leq 2k$ and diameter $D = 2p$ or $D = 2p + 1$, that can be constructed on the k-dimensional mesh, will be denoted $N_k(\Delta, p)$. Alternatively we could use the notations $B_k(D)$ and $N_k(\Delta, D)$, specifying whether D is even or odd. If $k' > k$, the following inequalities are straightforward:

$$|B_k(p)| \leq N_{k'}(2k, p) \leq |B_{k'}(p)| \tag{6.1}$$

The first inequality tells us that if we go to a higher dimension, keeping Δ and p constant, we can construct larger subgraphs. The reason for that is that we can move along the extra dimensions in order to avoid 'collisions'. Figure 6.2 is an example of one such construction in dimension $k = 3$, of a subgraph with degree $\Delta = 4$, diameter $D = 4$ (i.e. $p = 2$), and 18 vertices, whereas $|B_2(2)| = 13$.

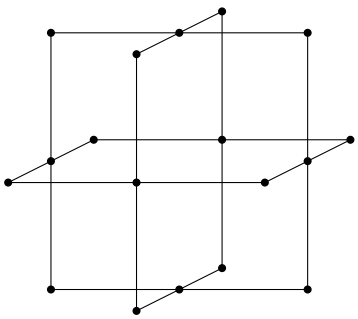


Figure 6.2: Construction for $\Delta = 4$ and $D = 4$ in the 3D mesh

From (6.1) follows the importance of determining the numbers $|B_k(p)|$. Counting the number of lattice points contained in a circle is a problem that goes back to Gauss, and there is an

extensive literature on the subject, including several approximate results for the number of lattice points in balls and other sets, e.g. [6, 13, 11, 12, 138, 150]. Regarding the *exact* number of lattice points contained in closed balls in the L^1 metric in arbitrary dimension, the main reference seems to be a paper by Vassilev-Missana and Atanassov [143]. The following result was given in [44], and we reproduce it here with slight modifications.

★ **Theorem 6.1** *The cardinality of $B_k(p)$ is*

$$|B_k(p)| = \begin{cases} \sum_{i=0}^{p} \binom{k}{i}\binom{k+p-i}{p-i} = \sum_{i=0}^{p} \binom{k}{p-i}\binom{k+i}{i} & \text{if } D = 2p \\ 2\sum_{i=0}^{p} \binom{k-1}{i}\binom{k+p-i}{p-i} = 2\sum_{i=0}^{p} \binom{k-1}{p-i}\binom{k+i}{i} & \text{if } D = 2p+1 \end{cases} \tag{6.2}$$

Proof:

The ball $B_k(p)$ can be constructed as the union of a smaller ball in the same dimension k, plus two balls in dimension $k-1$, as shown in Figure 6.3.

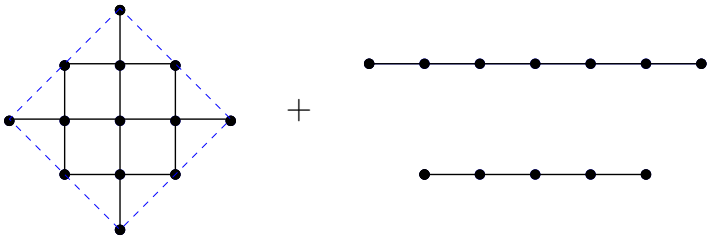


Figure 6.3: Decomposition of the ball of diameter 6

From this decomposition we get the recurrence relation

$$f(k,p) = f(k,p-1) + f(k-1,p) + f(k-1,p-1) \tag{6.3}$$

where $f(k,p)$ denotes the number of lattice points in $B_k(p)$. The boundary conditions are:

$$f(k,0) = \begin{cases} 1 & \text{if } D \text{ is even} \\ 2 & \text{if } D \text{ is odd} \end{cases}$$

$$f(1,p) = \begin{cases} 2p+1 & \text{if } D \text{ is even} \\ 2(p+1) & \text{if } D \text{ is odd} \end{cases} \tag{6.4}$$

We want to find the generating function $A_k(z) = \sum_{p\geq 0} f(k,p)z^p$. Multiplying (6.3) by z^p and summing over $p \geq 1$ we get

$$A_k(z) - A_k(0) = zA_k(z) + (A_{k-1}(z) - A_{k-1}(0)) + zA_{k-1}(z) \tag{6.5}$$

whence

$$A_k(z) = \frac{1+z}{1-z} A_{k-1}(z) \tag{6.6}$$

With the aid of the boundary conditions (6.4) we get

$$A_k(z) = \begin{cases} \frac{(1+z)^k}{(1-z)^{k+1}} & \text{if } D = 2p \\ 2\frac{(1+z)^{k-1}}{(1-z)^{k+1}} & \text{if } D = 2p+1 \end{cases} \tag{6.7}$$

For even D, $A_k(z)$ is the product of $(1+z)^k = \sum_p \binom{k}{p} z^p$ and $1/(1-z)^{k+1} = \sum_p \binom{k+p}{p} z^p$. Then, the series of $A_k(z)$ can be obtained as the convolution of the respective factor series. The series of $A_k(z)$ for odd D can be obtained in the same manner. □

For even D, the numbers $|B_k(p)|$ turn out to be the *Delannoy numbers* (sequence A008288 of [123]), which appear in a variety of combinatorial and geometric problems [141]. This particular interpretation of Delannoy numbers was first given by Vassilev-Missana and Atanassov [143], and later rediscovered by Schröder [136], and then by us. Our formulation and proof are different from the ones in [143, 136]. For odd D, the numbers $|B_k(p)|$ are known as a Riordan array of coordination sequences (sequence A113413 of [123]). Tables 6.1 and 6.2 show the first few values of $|B_k(p)|$ for even and odd D. They can be constructed in a Pascal-like fashion, with the convention that $|B_0(p)| = 1$.

	p								
k	0	1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	1	1	1
1	1	3	5	7	9	11	13	15	17
2	1	5	13	25	41	61	85	113	145
3	1	7	25	63	129	231	377	575	833
4	1	9	41	129	321	681	1289	2241	3649

Table 6.1: Some values of $|B_k(p)|$ for even D

	p								
k	0	1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	1	1	1
1	2	4	6	8	10	12	14	16	18
2	2	8	18	32	50	72	98	128	162
3	2	12	38	88	170	292	462	688	978
4	2	16	66	192	450	912	1666	2816	4482

Table 6.2: Some values of $|B_k(p)|$ for odd D

It is known that Delannoy numbers have no closed form, meaning that they cannot be represented as a linear combination of a fixed number of hypergeometric terms (which can be verified with the aid of the methods developed in [125]). However, we can extract asymptotic information from the generating function $A_k(z)$ in the proof of Theorem 6.1 above. Recall that α is an *algebraic singularity* of the function f if f can be written near α as

$$f(z) = f_0(z) + \frac{g(z)}{(1 - z/\alpha)^\omega} \tag{6.8}$$

where f_0 and g are analytic near α, g is nonzero near α, and ω is a real number different from $0, -1, -2, \ldots$. We readily recognize that 1 is an algebraic singularity of $A_k(z)$, since $A_k(z)$ can be written in the above form, with $g(z) = (1+z)^k$ for D even, and $g(z) = 2(1+z)^{k-1}$ for D odd, and all the other conditions are satisfied. Now we can readily apply Theorem 3 of [108]:

Theorem 6.2 *Suppose that for some real $r > 0$, $A(z)$ is analytic in the region $|z| < r$, and has a finite number $t > 0$ of singularities on the circle $|z| = r$, all of which are algebraic. Let α_i, ω_i, and g_i be the values of α, ω, and g in (6.8), corresponding to the i-th such singularity. Then $A(z)$ is the generating function for a sequence $\langle a_n \rangle$ satisfying*

$$a_n = \frac{1}{n} \sum_{i=1}^{t} \frac{g_i(\alpha_i) n^{\omega_i}}{\Gamma(\omega_i) \alpha_i^n} + o(r^{-n} n^{\Omega - 1})$$

where Ω is the maximum of the ω_i and Γ denotes the Gamma function.

We get

★ **Corollary 6.1**

$$|B_k(p)| = \frac{(2p)^k}{\Gamma(k+1)} + o(p^k) = \frac{(2p)^k}{k!} + o(p^k)$$

□

6.2 Subgraphs of degree 4 in the 3-dimensional mesh

An interesting special case is $k = 3$ and $\Delta = 4$. In this case, the inequalities (6.1) translate to $|B_2(p)| \leq N_3(4, p) \leq |B_3(p)|$, or

$$\begin{array}{ll} 2p^2 + 2p + 1 \leq N_3(4,p) \leq 4p^3/3 + 2p^2 + 8p/3 + 1 & \text{if } D = 2p \\ 2(p^2 + 2p + 1) \leq N_3(4,p) \leq 4p^3/3 + 4p^2 + 14p/3 + 2 & \text{if } D = 2p+1 \end{array}$$

The following theorem shows that the lower bounds are in fact a lot closer to the upper bounds:

★ **Theorem 6.3**

$$\begin{array}{lcccll} 4p^3/3 + 2p^2 - 4p/3 + 3 & \leq & N_3(4,p) & \leq & 4p^3/3 + 2p^2 + 8p/3 + 1 & \textit{if } D = 2p \\ 4p^3/3 + 4p^2 + 2p/3 & \leq & N_3(4,p) & \leq & 4p^3/3 + 4p^2 + 14p/3 + 2 & \textit{if } D = 2p+1 \end{array} \tag{6.9}$$

Proof:

Let $D = 2p$, with $p \geq 2$, and let us go back to our L^1 metric space model in dimension two. W.l.o.g. we can pick the center of our coordinate system as the center of all our balls and constructions. Now let $E_2(p)$ be the graph obtained from $B_2(p)$ by removing all the edges along the y-axis, and the two vertices that are left isolated (see Figure 6.4).

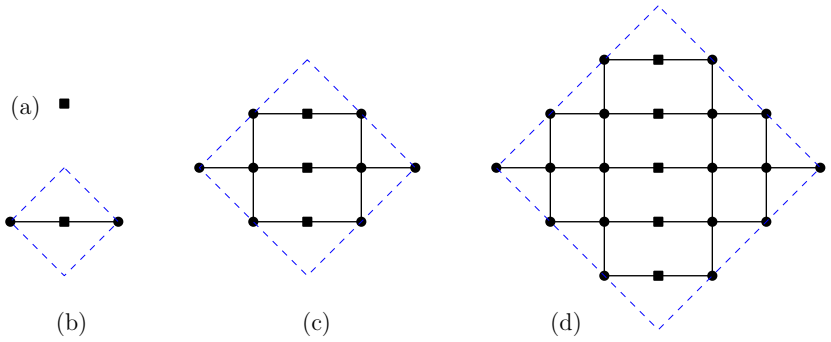


Figure 6.4: Construction for $\Delta = 4$ and even diameter in the 3D mesh

Note that in $E_2(p)$ there are two classes of nodes: the ones along the y-axis, shown as black squares, and all the other ones, shown as black circles. The interior circular nodes have degree 4, while the interior square nodes only have degree 2, and we will use them to move along the third dimension z.

Our three-dimensional graphs $H_3(p)$ (with $p \geq 2$) will consist of layers of $E_2(i)$ connected via the square nodes. Figure 6.5 describes schematically the layout of these layers in three-dimensional space.

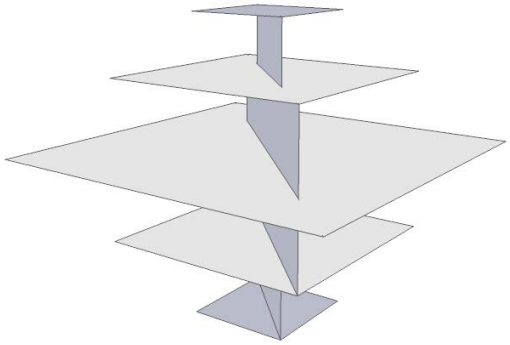


Figure 6.5: Arrangement of layers in three-dimensional space

At $z = 0$ we have $E_2(p)$, and at $z = \pm i$ we have $E_2(p - i)$, for $1 \leq i \leq p$. It is straightforward to check that $H_3(p)$ has diameter $D = 2p$, so let us concentrate on the number of nodes.

The number of vertices on each layer $E_2(i)$ is $|B_2(i)|$ minus two 'missing' vertices. Therefore

$$\begin{aligned} |E_2(p)| &= 2p^2 + 2p - 1 \\ |H_3(p)| &= 2\sum_{i=1}^{p-1}(2i^2 + 2i - 1) + (2p^2 + 2p - 1) \\ &= 4p^3/3 + 2p^2 - 4p/3 + 3 \end{aligned}$$

Now let $D = 2p+1$, with $p \geq 2$. The construction here is also made with layers that are $B_2(p)$ with the central 'spine' suppressed (i.e. the edges on the y-axis and the two tip vertices). Let

us call these graphs $O_2(p)$. Figure 6.6 shows $O_2(1)$, $O_2(2)$, and $O_2(3)$. The three-dimensional graph $Q_3(p)$ has a layer of type $O_2(p-i)$ at $z = \pm i$, for $0 \leq i \leq p-1$.

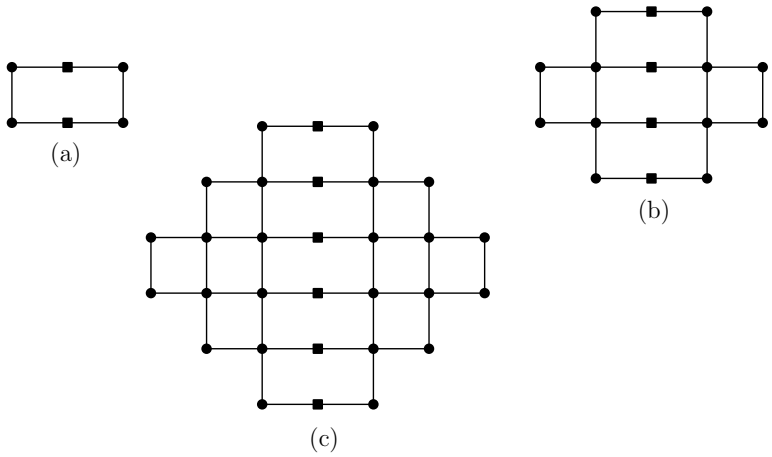


Figure 6.6: Construction for $\Delta = 4$ and odd diameter in the 3D mesh

Again, it is easy to check that the diameter is correct. Regarding the number of nodes, in every $O_2(i)$ there are only two nodes missing from $B_2(i)$. Hence

$$\begin{aligned} |Q_3(p)| &= 4\sum_{i=1}^{p-1}(i^2+2i) + 2p^2 + 4p \\ &= (4p^3 + 12p^2 + 2p)/3 \end{aligned}$$

□

Note that the above constructions are asymptotically optimal, since they agree up to the second term with the upper bounds. They are also optimal in another sense, as shown by the following

★ **Corollary 6.2** *The average degree $\hat{\delta}$ of the constructions in Theorem 6.3 tends to 4 as p approaches infinity.*

Proof:

$\hat{\delta} = 2A/V$, where A is the number of edges and V is the number of vertices. The number of edges is $8p^3/3 + 4p/3 + 2$ for $D = 2p$, and $8p^3/3 + 4p^2 - 2p/3$ for $D = 2p+1$. □

It is also very likely that these constructions can be extended to higher dimensions.

6.3 Subgraphs of degree 3 in the 2-dimensional mesh

The smallest case of $\Delta < 2k$ that makes sense is $\Delta = 3$ in dimension two. In this case we will also show that the lower bounds are quite close to the upper bounds. We have the following

★ **Theorem 6.4**

$$\begin{array}{lcll} 2p^2-2p+1 & \leq N_2(3,p) \leq & 2p^2+2p+1 & \text{if } D=2p \\ 8r^2+2r=2p^2+p & \leq N_2(3,p) \leq & 2p^2+4p+2 & \text{if } D=4r+1 \\ 8r^2+10r+6=2p^2+p+3 & \leq N_2(3,p) \leq & 2p^2+4p+2 & \text{if } D=4r+3 \end{array} \quad (6.10)$$

Proof:

As in Section 6.2, we will give two constructions that achieve the new bounds. In this case, a rigorous mathematical description of our constructions will be quite cumbersome; instead, we will use some geometric analogies to describe them. What we do is that we take the balls $B_2(p)$ and we try to fill them up in a convenient way with as many 'building blocks' as possible. Our construction elements will be 2×2 square blocks, and 2×1 half-blocks, or rectangular 'bricks'. We have to pack them up in such a way that no four edges meet at a single point.

Let us start with even D: We place as many square blocks as possible along the x and y axes, and if there is some space left at the end, we fill it up with bricks, as shown in Figure 6.7 (a). On each horizontal semi-axis we will have to use $(p-1)/2$ square blocks, and on each vertical semi-axis we will need $(p-3)/2$ square blocks.[1]

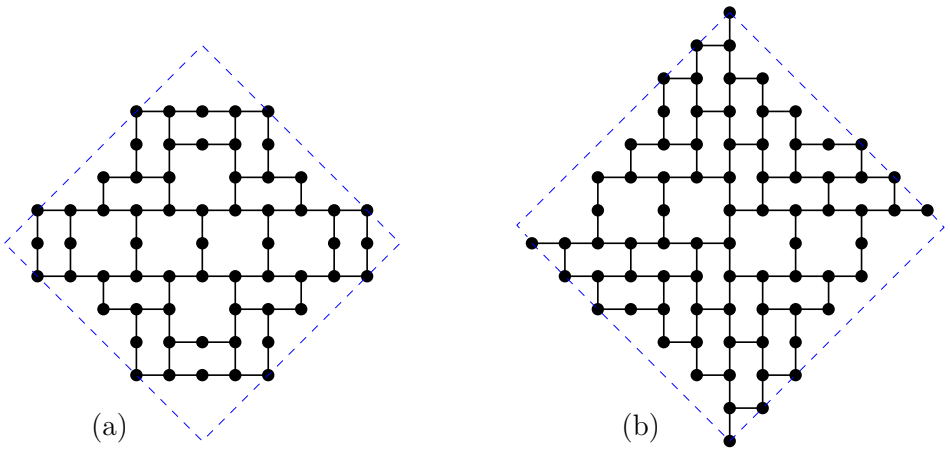


Figure 6.7: Constructions for $\Delta = 3$ in the two-dimensional mesh

Now we are left with four empty triangular regions that will have to be occupied. They are isosceles triangles, with side $p-2$. We will fill them with bricks, forming a pattern of interwoven horizontal and vertical bricks, as shown in Figure 6.8. Starting from the innermost corner there is only one way to do this, since every other vertex on the triangle sides already has three edges attached to it. Moreover, all triangles constructed this way will be the same, save rotations and reflections. We will make use of this fact again in the case of odd D. The construction ensures that no vertex has degree 4.

Next we have to show that the diameter is $D = 2p$. First consider the blocks along the axes. We shall call the vertices belonging to one of those blocks 'block vertices'. It is clear that starting from any block vertex we can reach any other block vertex in at most $2p$ steps. Hence we only have to include the vertices in the triangular regions, and show that for any such

[1] A fraction here means that we have to complete with a brick at the end.

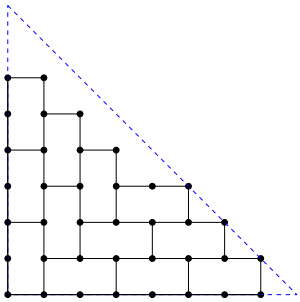


Figure 6.8: Brick pattern in a triangular region

vertex, we can reach any other vertex in at most $2p$ steps. The procedure is quite simple: First note that we can reach the innermost vertex of the triangle (which is also a block vertex) in at most $p-2$ steps (the length of the triangle sides). Call this vertex a 'corner'. From there we can reach any block vertex in at most $p+2$ steps, and we can reach any other corner in at most 4 steps, which means that we can reach any other vertex located in a triangular region in at most $2p$ steps, as desired.

Now we are going to count the number of vertices: If p is even we have $2(p-3)$ square blocks, plus four bricks, one at the end of each semi-axis. The square blocks contribute $10(p-3)+3$ points, and we have to add 12 additional points corresponding to the four bricks. Each triangle contributes $2\lfloor (p-2)/2 \rfloor^2 - 2\lfloor (p-2)/2 \rfloor = [(p-3)^2 - 1]/2$ points. Hence, the total number of vertices is $2p^2 - 2p + 1$. If p is odd we have $2(p-2)$ square blocks, and no bricks at the end of the semi-axes. The square blocks contribute $10(p-2)+3$ points. Finally, each triangle contributes $2\lfloor (p-2)/2 \rfloor^2 = (p-3)^2/2$ points. Hence, the total number of vertices is also $2p^2 - 2p + 1$.

Now consider the construction for odd D. In this case we have a central 'spine' of length $D = 2p+1$ (the y-axis), and two horizontal semi-axes of length p. We place $\lceil (p-2)/2 \rceil$ square blocks on each horizontal semi-axis, as shown in Figure 6.7 (b). Again, we are left with four triangular regions that are filled up with interwoven bricks, in the same manner as before. In this case, two triangles have sides of length p, and the other two have sides of length $p-1$.

We show that the diameter is $D = 2p+1$. We proceed as in the case of even D. In this case we distinguish block vertices and 'axis vertices' (i.e. vertices belonging to the central spine or to the horizontal semi-axes). We also distinguish two central vertices at the intersection of the vertical spine and the two horizontal semi-axes. It is clear that starting from any block or axis vertex we can reach the closest central vertex in at most p steps. Now, if we start at a vertex belonging to one of the larger triangles, of side length p, we can reach the corresponding corner (which is one of the central vertices) in at most p steps. Starting at a vertex in one of the smaller triangles, of side length $p-1$, we can reach the corresponding corner in at most $p-1$ steps, and the nearest central vertex is just one step away. Since both central vertices are adjacent, the proof of the diameter follows.

Regarding the number of vertices, we have $2p+2$ vertices on the vertical spine, plus $2p$ vertices from the horizontal semi-axes, plus $6\lceil (p-2)/2 \rceil$ block vertices, plus the triangles. If p is even, the two large triangles contribute $p(p-2)$ extra vertices, and the two smaller triangles contribute $(p-2)^2$ extra vertices. If p is odd, the two large triangles contribute $(p-1)^2$ extra

vertices, and the two smaller triangles contribute $(p-1)(p-3)$ extra vertices. Hence the total number of vertices is $2p^2+p$ if p is even, and $2p^2+p+3$ if p is odd.

□

These constructions are also asymptotically optimal in the sense of the average degree:

★ **Corollary 6.3** *The average degree $\hat{\delta}$ of the constructions in Theorem 6.4 tends to 3 as p approaches infinity.*

Proof:

The number of edges is

$$\begin{array}{ll} 3p^2-6p+6 & \text{if } D=4r \\ 3p^2-p-1 & \text{if } D=4r+1 \\ 3p^2-6p+4 & \text{if } D=4r+2 \\ 3p^2-p+3 & \text{if } D=4r+3 \end{array} \tag{6.11}$$

□

Despite being asymptotically optimal, these constructions can be improved without much effort for particular values of D, but extracting a general pattern and counting the vertices for arbitrary D may prove a challenge. In Table 6.3 we give the order of the largest graphs that have been constructed for some small values of D. The actual graphs can be seen in [124]. Note that for diameters $D=2,\ldots,6$, the upper bound given in the table is smaller than the upper bound given by Theorem 6.4, and the largest graphs are optimal. These graphs have been obtained by exhaustive computer search.

Diameter	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Upper bound	4	6	10	14	22	32	41	50	61	72	85	98	113	128	145
Order of largest graph	4	6	10	14	22	28	37	44	52	68	77	90	104	124	135

Table 6.3: Orders of the largest known subgraphs of the two-dimensional mesh, with $\Delta=3$ and $2\leq D\leq 16$.

6.4 Conclusions and open problems

To us, it is always amazing that a simple combinatorial setting like this one provides such a wealth of interesting and difficult problems. In the preceding two sections we have seen constructions that result in lower bounds for $N_k(\Delta,p)$, that are asymptotically close to the upper bounds, for small values of k and Δ, with $\Delta<2k$. Now, is that also possible for arbitrary k and $\Delta<2k$? How close can we get to the upper bounds? The constructions in Section 6.2 could be generalized to higher dimensions, more precisely to the case $\Delta=2k-2$. The constructions in Section 6.3 might also be extended to $\Delta=5$ in dimension 3, but it does not appear to be easy.

There are several additional properties that could be considered for every construction, like connectivity (*sic* fault-tolerance), average path length, symmetry, etc. For example, in parallel

computing applications, our constructions of Section 6.2 might not be the best ones, as they impose a relatively high communication overhead on the square vertices.

On the other hand, what is the computational complexity of finding the largest degree&diameter bounded subgraph in the mesh? Does it remain $\mathcal{NP}-$hard for all dimensions?, for some dimensions?, for any dimension? If it remains $\mathcal{NP}-$hard, then, can it be approximated to within a constant ratio?

This same study can be carried out for other host networks of theoretical and/or practical importance: the hypercube, the butterfly, the cube-connected cycles, Cayley graphs, etc. In Chapter 7 we discuss the honeycomb grid and the hypercube, and in Chapter 8 there is a preliminary heuristic study involving some random networks, but other than that, the field remains totally virgin.

7

The Degree-Diameter Subgraph Problem in other host networks

In this chapter we continue investigating the Degree-Diameter Problem in other classes of host networks of practical importance, namely the honeycomb grid (Section 7.1), and the hypercube (Section 7.2). This chapter represents work in progress. The contents of Section 7.1 will appear in [88], and Section 7.2 is the main topic of [87].

7.1 DDS in honeycomb grids

In Chapter 6 we addressed the Degree-Diameter Problem in the k-dimensional mesh or grid. Now, the infinite two-dimensional mesh is just one of the three *regular tilings* of the plane (see [79]). Another such tiling is the *hexagonal grid*, and we can apply the methods developed for the mesh to obtain some results for the hexagonal grid as well. As a model for parallel architectures, the hexagonal grid, or *honeycomb network*, offers some advantages in routing and broadcasting (see [140]).

7.1.1 Volume of maximal balls

Let H denote the infinite hexagonal grid in dimension two. Note that H is regular of degree 3. Although there is no hexagonal tiling of the three-dimensional space (or spaces of higher dimensions), we can also generalize the hexagonal grid to higher dimensions in a certain way. Note that the k-dimensional mesh is the cartesian product of the $(k-1)$-dimensional mesh with the infinite path P_∞. In analogy with that, we define the k-dimensional hexagonal grid $H^{(k)}$ for $k > 2$, as the Cartesian product of the $(k-1)$-dimensional hexagonal grid $H^{(k-1)}$ with P_∞.

Since the distance function in a connected graph is a metric in the usual sense, we can talk about open and closed balls in H, just like we did in the mesh. Again, we are interested in the

volume or number of vertices contained in a ball, and we are looking for the *maximal closed balls*, i.e. the closed balls with the largest volume.

The maximal closed ball of radius $\lfloor D/2 \rfloor$ and diameter D in dimension k will be denoted by $H_D^{(k)}$. If $k = 2$ we can omit the superscript. H_D is centered in a fixed vertex if D is even, or it has an edge as the center if D is odd. Figure 7.1 depicts closed balls with even diameters 2,4,6, and Figure 7.2 depicts the closed balls H_1, H_3 and H_5. We denote the volume of H_D by $|H_D|$.

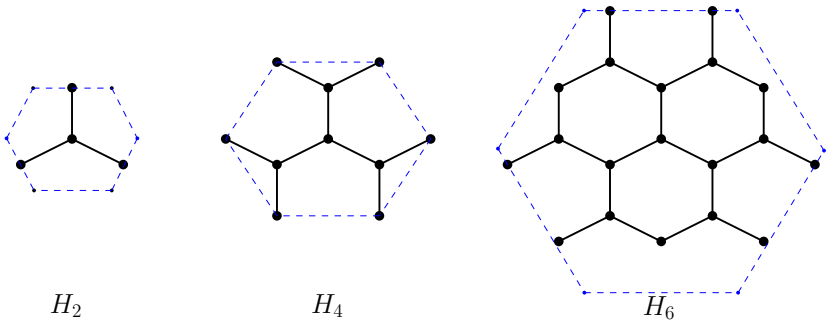


Figure 7.1: Maximal balls of even diameter in the two-dimensional hexagonal grid

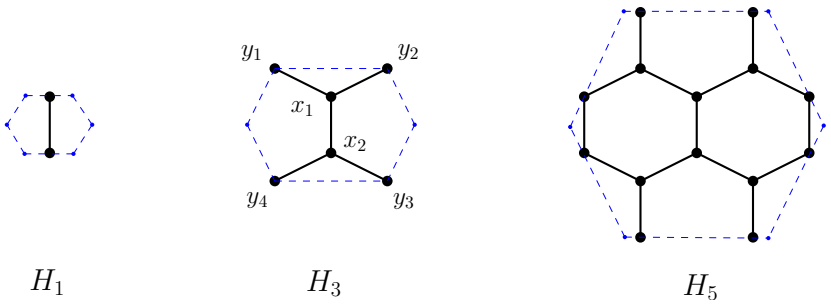


Figure 7.2: Maximal balls of odd diameter in the two-dimensional hexagonal grid

★ **Lemma 7.1** *Let $D = 4q + r$, where $q, r \in \mathbb{N}$, $r \in \{0, 1, 2, 3\}$. Then*

$$|H_D| = \begin{cases} 6q^2 + 3q + 1 & \text{when } r = 0, \\ 6q^2 + 6q + 2 & \text{when } r = 1, \\ 6q^2 + 9q + 4 & \text{when } r = 2, \\ 6q^2 + 12q + 6 & \text{when } r = 3. \end{cases} \tag{7.1}$$

Proof:

First consider the case of D even. Then, counting the vertices in horizontal rows we get

$$\begin{aligned} |H_D| &= (2q + 1) + (2q + 3) + \cdots + (2q + 2q + 1) + (2q + 2q - 1) + \cdots + (2q + 1) + (q + 1) = \\ &= 6q^2 + 3q + 1, \end{aligned}$$

when $D \equiv 0(\text{mod } 4)$, i.e., $D = 4q$, $q \in \mathbb{N}$, and

$$|H_D| = (q+1)(2q+3) + (2q+5) + \cdots + (2q+2q+3) + (2q+2q+1) + \cdots + (2q+3) = \\ = 6q^2 + 9q + 4,$$

when $D \equiv 2(\text{mod } 4)$, i.e., $D = 4q+2$, $q \in \mathbb{N}$.

For odd D we also count the vertices in horizontal rows, and we get

$$|H_D| = (q+1) + (2q+3) + (2q+5) \cdots + (2q+2q+1) + \\ + (2q+2q+1) + (2q+2q-1) + \cdots + (2q+3) + (q+1) = \\ = 6q^2 + 6q + 2,$$

when $D \equiv 1(\text{mod } 4)$, i.e., $D = 4q+1$, $q \in \mathbb{N}$, and

$$|H_D| = (2q+3) + (2q+5) + \cdots + (2q+2q+3) + (2q+2q+3) + (2q+2q+1) \cdots + (2q+3) = \\ = 6q^2 + 12q + 6,$$

when $D \equiv 3(\text{mod } 4)$, i.e., $D = 4q+3$, $q \in \mathbb{N}$.

□

From this we can obtain generating functions for $|H_D|$:

★ **Lemma 7.2** *The ordinary generating functions for $|H_D|$ are:*

$$\frac{1-x^3}{(x-1)^4} \quad \textit{for even values of } D \tag{7.2}$$

$$-\frac{2(x^2+x+1)}{(x-1)^2(x^2-1)} \quad \textit{for odd values of } D \tag{7.3}$$

$$-\frac{x^4+x^2+1}{(x-1)^2(x^4-1)} \quad \textit{for all } D \tag{7.4}$$

Proof: The subsequence for even D is well-known: the Centered Triangular Numbers (sequence A005448 of OEIS). For odd D we consider the two sub-cases separately: For $D = 4k+1$, the ordinary generating function (o.g.f) is $-\frac{2(1+4x+x^2)}{(x-1)^3}$, whereas for $D = 4k+3$, the o.g.f. is $-\frac{6(1+x)}{(x-1)^3}$. Now, if $f(x)$ is the o.g.f. for the sequence $a_0, a_1, a_2, \ldots$, then $f(x^2)$ is the o.g.f for the sequence $a_0, 0, a_1, 0, a_2, 0 \ldots$, and $xf(x^2)$ is the o.g.f for the sequence $0, a_0, 0, a_1, 0, a_2, 0 \ldots$. Therefore, in order to obtain the o.g.f for odd D we just have to add up $-\frac{2(1+4x^2+x^4)}{(x^2-1)^3}$ and $-\frac{6(1+x^2)}{(x^2-1)^3}$. We can do the same with (7.2) and (7.3) to get (7.4).

□

Now, from Equation 7.4 and with the aid of Mathematica we can get a general formula for $|H_D|$:

★ **Corollary 7.1**

$$|H_D| \leq \frac{6D^2 + 12D - 2ii^D + 2i(-i)^D + 3(-1)^D + 13}{16} \tag{7.5}$$

Proof: This result can be demonstrated independently from Mathematica, by checking that Equation 7.5 specializes to Equation 7.1 in each one of the four cases $r = 0, 1, 2, 3$.

□

In order to get formulas for $|H_D^{(k)}|$ in arbitrary dimension k, we note that $|H_D^{(k)}|$ can be decomposed according to the following recurrence formula:

$$|H_D^{(k)}| = |H_{D-2}^{(k)}| + |H_D^{(k-1)}| + |H_{D-2}^{(k-1)}| \tag{7.6}$$

This situation is similar to that of the k-dimensional grid, which is explained in Chapter 6. By taking $p = \lfloor D/2 \rfloor$, and making $f(k, p) = |H_D^{(k)}|$, we get the recurrence

$$f(k, p) = f(k-1, p) + f(k, p-1) + f(k-1, p-1) \tag{7.7}$$

which is easier to handle, though here we would have to distinguish two cases: even and odd D. There is a well-known sequence that satisfies Eq. 7.7, namely the sequence of Delannoy numbers. Subsequently we define the *Generalized Delannoy Numbers* as any number sequence satisfying Eq. 7.7. In Chapter 6 we have already discussed the 'classical' Delannoy numbers, and solved Equation 7.7 for particular initial values of f. Let $A_k(x)$ be the generating function for $f(k, p)$, i.e. $A_k(x) = \sum_{p \geq 0} f(k, p) x^p$. Then

$$A_k(x) = \frac{1+x}{1-x} A_{k-1}(x) = \ldots = \frac{(1+x)^{k-2}}{(1-x)^{k-2}} A_2(x) \tag{7.8}$$

Therefore, using Eq. 7.8 and Lemma 7.2 we prove the following

★ **Theorem 7.1**

$$A_k(x) = \begin{cases} \frac{(1-x^3)(1+x)^{k-2}}{(x-1)^4(1-x)^{k-2}} = \frac{(1+x)^{k-2}}{(1-x)^{k-1}} & \text{for even } D \\ -\frac{2(x^2+x+1)(1+x)^{k-2}}{(x-1)^2(x^2-1)(1-x)^{k-2}} = \frac{2(x^2+x+1)(1+x)^{k-3}}{(1-x)^{k+1}} & \text{for odd } D \end{cases} \tag{7.9}$$

□

Using the same methodology that we used in the proof of Lemma 7.2 we should also be able to obtain a unified formula for $A_k(x)$ in arbitrary diameter, at least in principle, but the expressions that we get are too cumbersome for general usage, and we will not derive them here. Nevertheless, we can obtain the unified formula for dimension three:

★ **Corollary 7.2** *The ordinary generating functions for* $|H_D^{(3)}|$ *are:*

$$A_3(x) = \begin{cases} \frac{(1-x^3)(1+x)}{(1-x)^5} & \text{for even values of } D \\ \frac{2(x^2+x+1)(1+x)}{(1-x)^3(x^2-1)} & \text{for odd values of } D \\ \frac{(x^2-x+1)(x^2+x+1)}{(x-1)^4(x+1)^2} & \text{for all } D \end{cases} \tag{7.10}$$

□

As we did in dimension 2, we can now get a general formula for $|H_D^{(3)}|$:

★ **Corollary 7.3**

$$|H_D^{(3)}| = \frac{(1+D)(2D^2+4D+13+3(-1)^D)}{16} \tag{7.11}$$

Proof: The partial fraction decomposition of Equation 7.10 for all D is

$$\frac{3}{16(x+1)^2} + \frac{13}{16(x-1)^2} + \frac{3}{4(x-1)^3} + \frac{3}{4(x-1)^4}$$

Therefore,

$$|H_D^{(3)}| = \frac{3}{16}(-1)^D(D+1) + \frac{13}{16}(D+1) - \frac{3}{8}(D+1)(D+2) + \frac{1}{8}(D+1)(D+2)(D+3)$$

□

It is quite surprising that this formula turns out to be simpler than in the two-dimensional case, in the sense that it does not involve the imaginary unit. In terms of cases, Equation 7.5 distinguishes four cases, while Equation 7.11 only makes a distinction between two cases: even and odd D. For practical purposes we can decompose Equation 7.11 into four cases as well:

★ **Corollary 7.4** *Let $D = 4q + r$, where $q, r \in \mathbb{N}$, $r \in \{0, 1, 2, 3\}$. Then*

$$|H_D^{(3)}| = \begin{cases} 8q^3 + 6q^2 + 5q + 1 & \text{if } r = 0, \\ 8q^3 + 12q^2 + 8q + 2 & \text{if } r = 1, \\ 8q^3 + 18q^2 + 17q + 6 & \text{if } r = 2, \\ 8q^3 + 24q^2 + 26q + 10 & \text{if } r = 3. \end{cases} \tag{7.12}$$

□

Figure 7.3 depicts the ball $H_4^{(3)}$, and Figure 7.4 depicts the ball $H_5^{(3)}$.

Now, for arbitrary D we can also obtain asymptotic information about $|H_D^{(k)}|$, just as we did for $|B_k(p)|$ in Section 6.1. Again, let $p = \lfloor D/2 \rfloor$; applying Theorem 6.2 we get

★ **Corollary 7.5**

$$|H_D^{(k)}| = \begin{cases} \frac{(2p)^{k-2}}{(k-2)!} + o(p^{k-2}) & \text{if } D \text{ is even}, \\ \frac{3(2p)^k}{4k!} + o(p^k) & \text{if } D \text{ is odd}. \end{cases}$$

□

Note that the sequence for odd D grows quite faster than the sequence for even D.

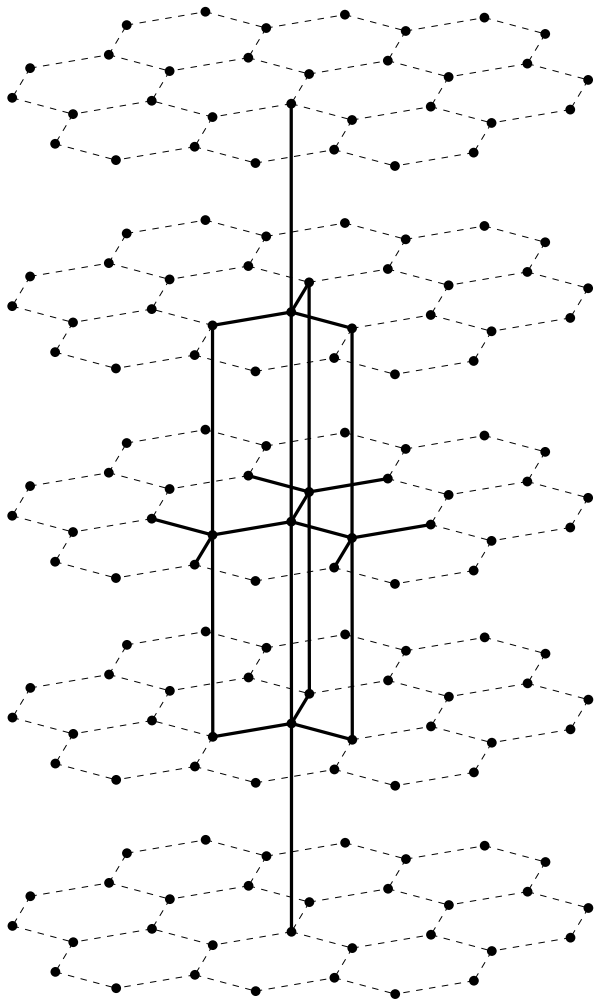


Figure 7.3: Maximal ball of diameter 4 in the three-dimensional hexagonal grid

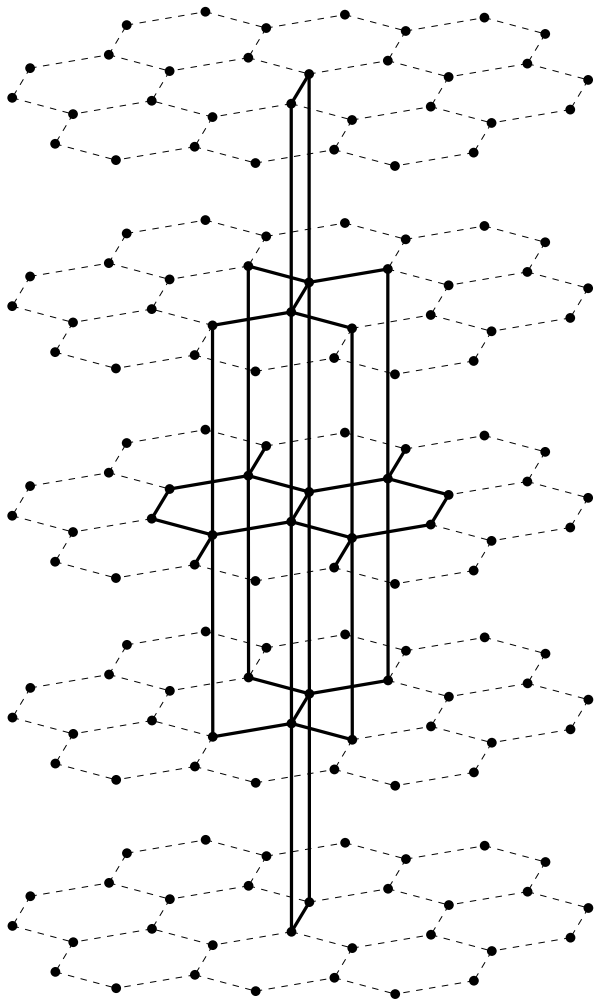


Figure 7.4: Maximal ball of diameter 5 in the three-dimensional hexagonal grid

7.1.2 Lower bounds in the two-dimensional grid

In general, we are looking for a largest subgraph of $H^{(k)}$ of given diameter D and given maximum degree Δ, where $\Delta \leq 2k-1$. We will denote by $N_k(\Delta, D)$ the order of such a subgraph. In Subsection 7.1.1 we discussed upper bounds for $N_k(\Delta, D)$, and now let us see some constructions that provide lower bounds. We start with dimension 2, and as we did before, we omit the subscript in this case.

Since $\Delta(H) = 3$, it makes sense to consider $\Delta \leq 3$. If $\Delta = 1$, then the largest subgraph of H is an edge and hence $N(1, D) = 2$. If $\Delta = 3$, then the largest subgraph of H is H_D, hence $N(3, D) = |H_D|$. Thus it remains to find a lower bound on $N(\Delta, D)$ for $\Delta = 2$. Since the diameter D of S is given and H is bipartite, the largest subgraph of H is a path of length D, for $D < 3$, or a cycle of length $2D$.

★ **Theorem 7.2** *Let D be a positive integer and let H be the infinite honeycomb grid. Then*

$$N(2, D) = \begin{cases} D+1 & \text{if } D \leq 4, \\ 2D & \text{otherwise.} \end{cases}$$

Proof: From Figures 7.1 and 7.2 it follows that, for $D \leq 4$, the graph H_D contains no cycle. Hence a largest subgraph of H of diameter D and maximum degree 2 is a path on $D+1$ vertices. Clearly such a path exists for each $D \in \{1, 2, 3, 4\}$.

Let us suppose that $D \geq 5$. Since H is bipartite, the largest possible subgraph of H of given diameter D and $\Delta = 2$ is a cycle C on $2D$ vertices. We distinguish the following posibilities:

Case 1: $D > 4$ is even.

Consider a border cycle C of the two horizontal rows of hexagons which contain the central vertex x (see the red cycle in Figure 7.5). If $D = 4q$, $q \in \mathbb{N}$, then C has $2q + 2q - 3 + 2q + 2q - 1 + 4 = 8q$ vertices. If $D = 4q + 2$, $q \in \mathbb{N}$, then C contains $2q + 2q - 1 + 2q + 2q - 3 + 4 = 8q$ vertices. Thus, for every even $D > 4$ there is a cycle C in H_D of length $2D$.

Case 2: $D > 4$ is odd.

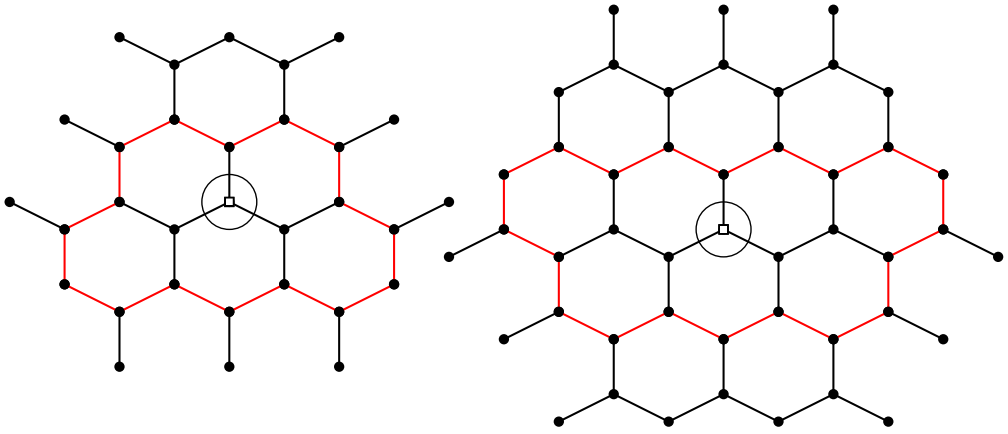
Suppose that $D = 4q + 1$, $q \in \mathbb{N}$. A border cycle C of the central horizontal row of hexagons (see Figure 7.6) contains $2q$ hexagons, i.e., $8q+2$ vertices. Now suppose that $D = 4q + 3$, $q \in \mathbb{N}$. Consider three horizontal rows of hexagons: the central one, the upper neighbouring one and the lower neighbouring one (see Figure 7.6). A border cycle C of these rows contains $2q + 2q - 1 + 2q + 2q - 1 + 8 = 8q + 6$. Thus, for every odd $D > 4$, C contains $2D$ vertices.

□

7.1.3 Lower bounds on $N_G(\Delta, D)$ for $\Delta = 4$.

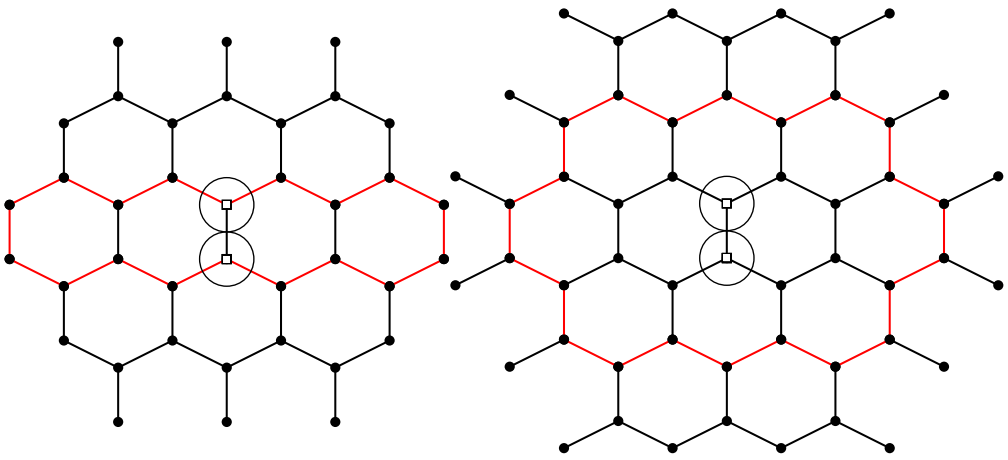
We start with the following statements.

★ **Proposition 7.1** *Let $D \geq 4$ be an even integer and H_D a largest subgraph of the infinite honeycomb grid of diameter D. Let x denote the central vertex of H_D and e denote any edge incident to x. Then there are $D/2 + 1$ vertices u such that $dist_{H_D}(u, x) \geq D - 1$ and $dist_{H_D - e}(u, x) > D$.*



$D = 4q$ $D = 4q + 2$

Figure 7.5: The set Y for even diameter



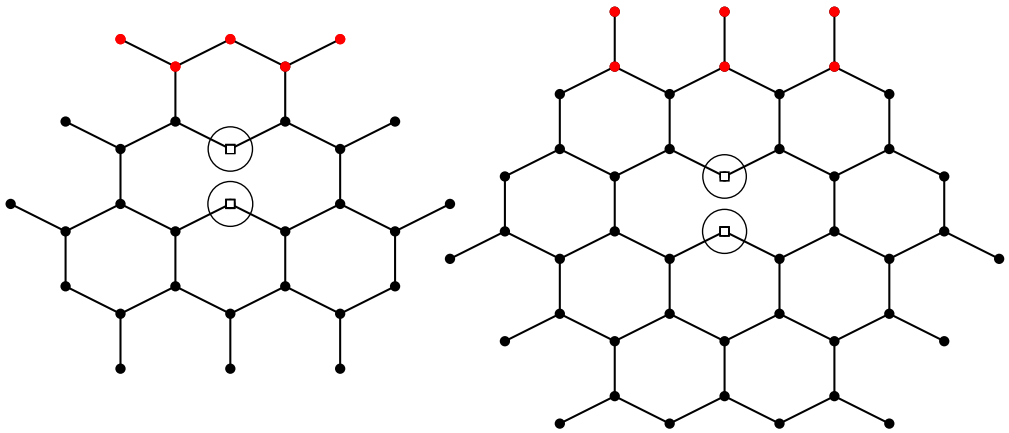
$D = 4q + 1$ $D = 4q + 3$

Figure 7.6: The set Y for odd diameter

Proof: Consider the graph H_D (see Figure 7.1) after deleting the upper edge incident to the central vertex x. First suppose that $D/2$ is even. Hence $D = 4q$, $q \in \mathbb{N}$ and $q \geq 1$. A set Y of vertices u of H_D such that $\text{dist}_{H_D}(u, x) \geq D - 1$ and $\text{dist}_{H_D - e}(u, x) > D$ are vertices of the most up path in H_D (see Figure 7.7, the vertex x is encircled and Y consists of all the red vertices). Thus $|Y| = 2q + 1 = D/2 + 1$.

Now suppose that $D/2$ is odd. A set Y of vertices u of H_D such that $\text{dist}_{H_D}(u, x) \geq D-1$ and $\text{dist}_{H_D - e}(u, x) > D$ are vertices of the most up row in H_D and their neighbours (see Figure 7.7, the vertex x is encircled and Y consists of all the red vertices). Thus $|Y| = 2\frac{D+2}{4} = D/2 + 1$.

□



$D = 4q$ $D = 4q + 2$

Figure 7.7: The set Y for even diameter

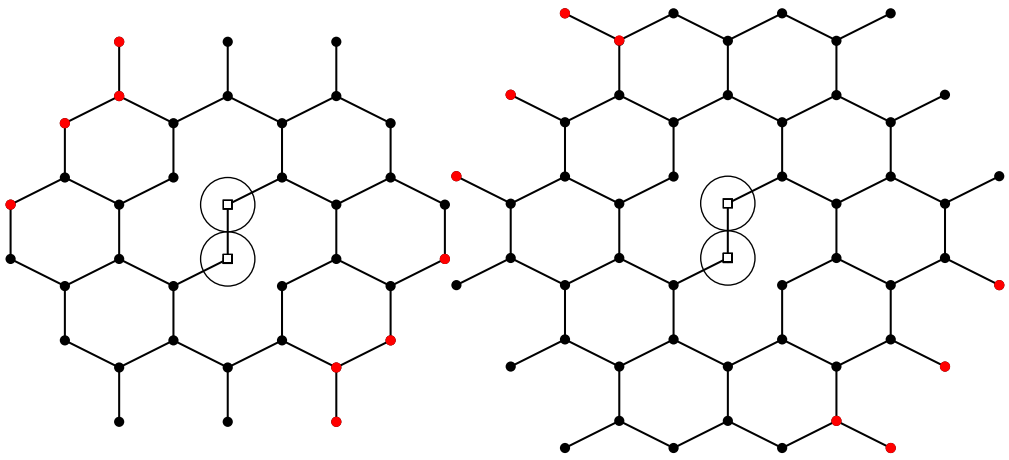
★ **Proposition 7.2** *Let $D \geq 5$ be an odd integer and H_D a largest subgraph of the infinite honeycomb grid of diameter $D \geq 4$. Let x^1x^2 denote the central edge of H_D, e_1 denote any edge incident to x^1 and e_2 the edge incident to x^2 in an opposite position. Then there are $D/2 + 1$ vertices u such that $dist_{H_D}(u, x) \geq D - 1$ and $dist_{H_D - e}(u, x) > D$.*

Proof: Consider the graph H_D (see Figure 7.2), keeping notation of the vertices x^1 and x^2. Now we delete the upper left edge e_1 incident to x^1 and the lower right edge e_2 incident to x^2. The subgraph of H_D so obtained will be denoted by $\bar{H}_D$. Let Y denote a set of vertices u of $\bar{H}_D$ such that $\text{dist}_{H_D}(u, x^1x^2) \geq D - 1$ and $\text{dist}_{\bar{H}_D}(u, x^1x^2) > D$.

Let $D = 4q + r$, $q \in \mathbb{N}$, $r \in \{1, 3\}$. We show that $|Y| = 2(q + 2)$. The set Y consists of the upper left row, the rightmost bottom row, and two extra vertices, as can be seen in Figure 7.8 (vertices of the central edge are encircled, and Y consists of all the red vertices). Clearly $|Y| = 2(q + 2)$.

□

★ **Theorem 7.3** *Let $D \geq 2$ be a positive integer such that $D = 4q + r$, $q \in \mathbb{N}$, $r \in \{0, 2\}$. Then*



$D = 4q + 1$ $D = 4q + 3$

Figure 7.8: The set Y for odd diameter

$$N_3(4, D) \leq \begin{cases} 5 & \text{if } D = 2 \\ 16 & \text{if } D = 4 \\ 8q^3 + 6q^2 + 3q - 3 & \text{if } r = 0 \text{ and } q \geq 2, \\ 8q^3 + 18q^2 + 15q + 1 & \text{if } r = 2 \text{ and } q \geq 1. \end{cases} \tag{7.13}$$

Proof: By the assumptions, D is even. Consider a graph $H_D^{(3)}$. Note that $H_D^{(3)}$ consists of the layers $L_0, \dots L_D$. Let x_i denote the central vertex of L_i $(i = 0, \dots, D)$ and let y_i^j denote the neighbours of x_i in L_i, $(i = 1, \dots, D-1$ and $j = 1, 2, 3)$. Note that the vertices y_i^j correspond to the vertices y_{i+1}^j, $i = 1, \dots D-2$ and $j = 1, 2, 3$, i.e., the vertices y_i^j, y_{i+1}^j are at the same positions in corresponding layers L_i, L_{i+1}. From the graph $H_D^{(3)}$, for each odd $i = 1, 3, \dots, D-3$ we delete all edges between some vertex of L_i at even distance from x_i and corresponding vertex in L_{i+1}. Similarly for each even $i = 2, 4, \dots, D-2$, we delete all edges between some vertex of L_i at odd distance from x_i and corresponding vertex in L_{i+1}.

If $r = 0$, then $e = x_{D/2}y_{D/2}^1$, otherwise $e = x_{D/2}y_{D/2}^2$. Let Y denote a set of all vertices u of $L_{D/2}$ such that $\text{dist}_{LD/2}(u, x_{D/2}) \geq D-1$ and $\text{dist}_{L(D/2)-e}(u, x_{D/2}) \geq D+1$. Now we delete the vertices y_2^2, y_{D-1}^1, all the vertices of Y and all edges incident to any of the deleted vertices in $H_D^{(3)}$. We also delete all the edges $x_iy_i^1$ whenever i is odd, all the edges $x_iy_i^2$ whenever i is even.

Such a graph will be denoted by G_D. Let $\bar{L}_0 \dots, \bar{L}_D$ denote layers of G_D, i.e., $\bar{L}_i = G_D \cap L_i$, $i = 0, \dots, D$.

It is easy to see that $\Delta(G_D) = 4$. Now we show that the diameter of G_D is D, i.e., for each $u \in V(G_D)$

$$\text{dist}_{G_D}(u, x_{D/2}) \leq D/2. \tag{7.14}$$

Let i denote the index of the layer $\bar{L}_i$ which contains vertex u. We suppose that $i = D/2$, i.e., the vertex u belongs to the middle layer. Clearly $\bar{L}_{D/2}$ is connected and $\mathrm{dist}_{G_D}(u, x_{D/2}) \leq D/2$ except the case when $D = 6$. In this case, $\bar{L}_3$ consists of two components. The component containing vertex $x_{D/2}$ clearly satisfy condition (7.14), hence it remains to consider the other component which contains only vertex y_3^1. But then the path $P = y_3^1, y_4^1, x_4, x_3$ has length 3, hence we satisfy condition (7.14). Note that y_4^1 denote the corresponding vertex of y_3^1 in $\bar{L}_4$ and clearly $y_4^1 x_4 \in E(G_D)$.

Now we consider the following possibilities:

Case 1: $i < D/2$.

First we suppose that i is odd. By the construction of G_D, $y_i^q y_{i+1}^q \in E(G_D)$, $q = 1, 2, 3$. Let P denote a shortest path between u and any of the vertices y_i^q, say y_i^l, $l \in \{1, 2, 3\}$. Clearly $\mathrm{dist}_{\bar{L}_i}(u, y_i^l) \leq i - 1$. If $y_i^l x_i \in E(G_D)$, then we add the edges $y_i^l x_i$, $x_i x_{i+1}$ to P. If $y_i^l x_i \notin E(G_D)$, then we add the edges $y_i^l y_{i+1}^l$, $y_{i+1}^l x_{i+1}$ to P. In both cases we have shown that the length of P is at most $i + 1$. Since $\mathrm{dist}_{G_D}(x_{i+1}, x_{D/2}) = D/2 - i - 1$, we obtain $\mathrm{dist}_{G_D}(u, x_{D/2}) \leq i + 1 + D/2 - i - 1 = D/2$.

Now we suppose that i is even. If there is a u, x_i-path P of length at most i in $\bar{L}_i$, we add the edge $x_i x_{i+1}$ to P, P has length at most $i + 1$ and we easily satisfy condition (7.14). Hence suppose that there is no such path P in $\bar{L}_i$. Let u' denote the corresponding vertex of u in $\bar{L}_{i+1}$ and let P' denote a shortest $u' x_{i+1}$-path in $\bar{L}_{i+1}$. Note that, by the construction of G_D, such the vertex and the path exist, and the length of P' is at most i. Let z' denote the vertex of P' at distance 2 from x_{i+1} on P', and let z denote the corresponding vertex of z' in $\bar{L}_i$. Since i is even, $zz' \in E(G_D)$. Let Q denote the copy of the $u'z'$-subpath of P' in $\bar{L}_i$. Note that Q has length at most $i - 2$. Then the u, x_{i+1}-path P consisting of Q, the edge zz' and the $z' x_{i+1}$-subpath of P' has length at most $i + 1$. And, since $\mathrm{dist}_{G_D}(x_{i+1} x_{D/2}) = D/2 - i - 1$, we obtain $\mathrm{dist}_{G_D}(u, x_{D/2}) \leq i + 1 + D/2 - i - 1 = D/2$.

Case 2: $i > D/2$.

Similarly as in the previous case we can show that $\mathrm{dist}_{G_D}(u, x_{i-1}) \leq D - i + 1$ interchanging the role of the parity of i. And, since $\mathrm{dist}_{G_D}(x_{i-1} x_{D/2}) = i - 1 - D/2$, we obtain $\mathrm{dist}_{G_D}(u, x_{D/2}) \leq D - i + 1 + i - 1 - D/2 = D/2$.

Finally we count the number of vertices of G_D from the number of vertices of $H_D^{(3)}$. From Corollary 7.11 we have $|V(H_D^3)| = 8q^3 + 6q^2 + 5q + 1$ when $D = 4q$, $q \in \mathbb{N}$ and $|H_D^3| = 8q^3 + 18q^2 + 17q + 6$ when $D = 4q + 2$, $q \in \mathbb{N}$. If $D = 2$, then we deleted one vertex from L_1 to get $\bar{L}_1$, thus we have $|V(G_D)| = 5$. If $D = 4$, we deleted three vertices from L_2 to get $\bar{L}_2$ and one vertex from L_3 to get $\bar{L}_3$, thus we obtain $|V(G_D)| = 16$. For all $D \geq 6$, we deleted one vertex from L_2 to get $\bar{L}_2$, one vertex from L_{D-1} to get $\bar{L}_{D-1}$ and $D/2 + 1$ vertices from $L_{D/2}$ to get $\bar{L}_{D/2}$ by Proposition 7.1. Thus we obtain $|V(G_D)| = 8q^3 + 6q^2 + 3q - 3$ when $D \equiv 0(\mod 4)$ and that $|V(G_D)| = 8q^3 + 18q^2 + 15q + 1$ when $D \equiv 2(\mod 4)$.

□

★ **Theorem 7.4** *Let $G = P \times H$, let $D \geq 2$ be a positive integer such that $D = 4q + r$, $q \in \mathbb{N}$, $r \in \{1, 3\}$. Then*

$$N_G(D,4) \leq \begin{cases} 2 & \text{if } D = 1 \\ 8 & \text{if } D = 3 \\ 22 & \text{if } D = 5 \\ 8q^3 + 12q^2 + 6q - 2 & \text{if } r = 1 \text{ and } q \geq 2, \\ 8q^3 + 24q^2 + 24q + 6 & \text{if } r = 3 \text{ and } q \geq 1. \end{cases}$$

It is easy to observe that D is odd. Consider a graph $H_D^{(3)}$ for odd D and note that $H_D^{(3)}$ consists of layers $L_0, \ldots, L_{D-1}$. Let x_i^1, x_i^2 denote the end vertices of the central edge in L_i $(i = 0, \ldots D-1)$, let y_i^j denote the neighbours of x_i^1 or x_i^2 in L_i $(i = 1, \ldots, D-2$ and $j = 1, 2, 3, 4)$. We will keep the labelling of these vertices in L_i as shown in Figure 7.2. Note that the vertices x_i^1, x_i^2 and y_i^j $(j = 1, 2, 3, 4)$ are at same positions in all the mentioned layers.

From the graph H_D we delete all edges between any vertex of L_i at even (nonzero) distance from $x_i^1 x_i^2$ and corresponding vertex in L_{i+1} for each odd $i = 3, \ldots, D-4$. Similarly for each even $i = 2, 4, \ldots, D-3$, we delete all edges between any vertex of L_i at odd distance from $x_i^1 x_i^2$ and corresponding vertex in L_{i+1}. If $r = 1$, then we set $e_1 = x_i^1 y_i^1$ and $e_2 = x_i^2 y_i^3$, otherwise we set $e_1 = x_i^1 y_i^2$ and $e_2 = x_i^2 y_i^4$. Let Y denote a set of all vertices u of $L_{(D-1)/2}$ such that $\mathrm{dist}_{L_{(D-1)/2}}(u, x_i^1 x_i^2) \geq D-1$ and $\mathrm{dist}_{L_{(D-1)/2} - \{e_1, e_2\}}(u, x_i^1 x_i^2) \geq D+1$. Now we delete all the vertices of Y and all the edges incident to at least one vertex of Y from $H_D^{(3)}$. We also delete the edges $x_i^1 y_i^1$, $x_i^2 y_i^3$ whenever i is odd and different from $D-2$, and the edges $x_i^1 y_i^2$, $x_i^2 y_i^4$ whenever i is even and different from 2. If $D = 5$, we delete the edges $x_2^1 y_2^2$ and $x_2^2 y_2^4$, otherwise we delete the central edge $x_2^1 x_2^2$. Similarly, if $D = 3$, then we delete the edges $x_1^1 y_1^1$ and $x_1^2 y_2^3$, otherwise we delete the central edge $x_{D-2}^1 x_{D-2}^2$.

The resulting graph will be denoted by G_D. Let $\bar{L}_0, \ldots, \bar{L}_{D-1}$denote layers of G_D, i.e., $\bar{L}_i = G_D \cap L_i$, $i = 0, \ldots, D-1$.

Clearly $\Delta(G_D) = 4$. Now we show that the diameter of G_D is D, i.e., for each $u \in G_D$ the distance from u to $x_{(D-1)/2}^1 x_{(D-1)/2}^2$ is at most $(D-1)/2$.

Let i denote the index of the layer L_i which contains vertex u. If $i = (D-1)/2$, i.e., the vertex u belongs to the middle layer, then the distance from arbitrary vertex of L_i to $x_{(D-1)/2}^1 x_{(D-1)/2}^2$ is at most $(D-1)/2$ in L_i except the case when $D = 7$, and $u = y_3^1$ or $u = y_3^3$. But then, up to symmetry, say for $u = y_1^1$, the path consisting of edges uy_4^1, $y_4^1 x_4^1$ and $x_4^1 x_3^1$ has length $3 = (D-1)/2$. Clearly $\mathrm{dist}_{G_D}(u, x_i^1) \neq \mathrm{dist}_{G_D}(u, x_i^2)$, therefore, up to symmetry, we can assume that $\mathrm{dist}_{G_D}(u, x_i^1) < \mathrm{dist}_{G_D}(u, x_i^2)$.

Now we consider the following possibilities:

Case 1: $i < (D-1)/2$.

We start with odd i. By the construction of G_D, $y_i^q y_{i+1}^q \in E(G)$ for every $q = 1, 2, 3, 4$. Consider a shortest path P between u and any of the vertices y_i^q, say y_i^l, $l \in \{1, 2\}$. Clearly $\mathrm{dist}_{G_D}(u, y_i^l) \leq i-1$. If $y_i^l x_i^1 \in E(G_D)$, then we add the edges $y_i^l x_i^1$ and $x_i^1 x_{i+1}^1$ to P. Otherwise we add the edges $y_i^l y_{i+1}^l$ and $y_{i+1}^l x_{i+1}^1$ to P. In both possibilities we have shown that the length of P is at most $i+1$. Since $\mathrm{dist}_{G_D}(x_{i+1}^1, x_{(D-1)/2}^1 = (D-1)/2 - (i+1)$, the distance of u from $x_{(D-1)/2}^1 x_{(D-1)/2}^2$ is at most $(D-1)/2$.

Now we suppose that i is even. If $i = 2$, then every vertex of L_2 is at distance at most 2 from $x_2^1 x_2^2$ and the distance of u from $x_{(D-1)/2}^1 x_{(D-1)/2}^2$ is at most $(D-1)/2$. Let us suppose that $i > 2$. If there is a ux_i^1-path P in L_i of length at most $i-1$, then

the distance of u from $x^1_{(D-1)/2}x^2_{(D-1)/2}$ is at most $(D-1)/2$. Hence suppose that there is no such path P in L_i. Let u' denote the corresponding vertex of u in L_{i+1} and let P' be a shortest $u'x^1_{i+1}$-path in L_{i+1}. By the construction of G_D such a path and such a vertex exist and the length of P' is at most i. Let z' denote the vertex of P' at distance two from x^1_{i+1} on P', and let z denote the corresponding vertex of z' in L_i. Since i is even, $zz' \in E(G_D)$. Let Q be the copy of the $u'z'$-subpath of P' in L_i. Clearly the length of Q is at most $i-2$. Then the u, x^1_{i+1}-path P consisting of Q, the edge zz' and the z', x^1_{i+1}-subpath of P' has length at most $i+1$. And, since $\text{dist}_{G_D}(x^1_{i+1}, x^1_{(D-1)/2}) = (D-1)/2 - (i+1)$, the distance of u from $x^1_{(D-1)/2}x^2_{(D-1)/2}$ is at most $(D-1)/2$.

Case 2: $i < (D-1)/2$.

Similarly as in Case 1 we can show that $\text{dist}_{G_D}(u, x^1_{i-1}) \leq (D-1)-i+1$ interchanging the role of the parity of i. And, since $\text{dist}_{G_D}(x^1_{i-1}x^1_{(D-1)/2}) = i-1-(D-1)/2$, we obtain $\text{dist}_{G_D}(u, x^1_{(D-1)/2}) \leq (D-1)-i+1+i-1-(D-1)/2 = (D-1)/2$.

Finally we count the number of vertices in G_D from the number of vertices of $H^{(3)}_D$. From Corollary 7.11 we have $|H^3_D| = 8q^3+12q^2+8q+2$ when $D = 4q+1$, $q \in \mathbb{N}$ and $|H^3_D| = 8q^3+24q^2+26q+10$ when $D = 4q+3$. For $D = 1$ we have $|G_1| = 2$ since G_1 consists of only one edge. If $D = 3$, then we deleted two vertices in H_3 to obtain the middle layer L, thus $|G_3| = 8$. If $D = 5$, then we deleted six vertices from H_5 to obtain L, implying that $|G_5| = 22$. If $D \geq 7$, then we obtained L from H_D deleting $2(q+2)$ vertices by Proposition 7.2. Thus we obtain $|G_D| = 8q^3+12q^2+6q-2$ when $D = 4q+1$ and $|G_D| = 8q^3+24q^2+24q+6$ when $D = 4q+3$.

□

7.2 DDS in the k-dimensional hypercube

The hypercube is one of the network topologies that arise very frequently in parallel architectures. The k-dimensional hypercube Q_k can be defined recursively as

$$Q_2 = K_2$$
$$Q_k = Q_{k-1} \times K_2 \quad \text{for } k > 2.$$

Q_k is a k-regular graph of 2^k vertices and $k2^{k-1}$ edges, and diameter k. It is bipartite and vertex-transitive for all $k \geq 1$, and hamiltonian for $k \geq 2$. The vertices of Q_k can be seen as the binary strings of length k, with two strings connected iff they differ in just one bit. The *Hamming distance* between two vertices is the number of bits in which they differ. Therefore, Q_k can also be seen as the *Hasse diagram* of a *finite boolean algebra*. Sometimes we will make use of this duality, and we might refer to Q_k as a graph or as a boolean algebra interchangeably.

7.2.1 First case: Smaller diameter

As usual, let us denote by $N_k(\Delta, D)$ the order of the largest subgraph of degree Δ and diameter D in Q_k. Let us assume that $\Delta > 2$ and $D \leq \Delta \leq k$.

From the definition of Q_k it can be seen that Q_k contains Q_i as a subgraph, for every $2 \leq i \leq k-1$. This fact provides a straightforward lower bound for $N_k(\Delta, D)$ when $D \leq \Delta$, since Q_k contains a subcube Q_Δ, and in turn, Q_Δ contains a subgraph of order

$$\Phi_\Delta(D) = \sum_{i=0}^{D} \binom{\Delta}{i} \tag{7.15}$$

made up by the first $D+1$ levels of Q_Δ.

The function $\Phi_\Delta(D)$ represents the volume of the Hamming ball of radius D. There seems to be no simple combinatorial closed form for (7.15) but it is well known that

$$\frac{1}{\sqrt{8D(1-\frac{D}{\Delta})}} 2^{H(\frac{D}{\Delta})\Delta} \leq \Phi_\Delta(D) \leq 2^{H(\frac{D}{\Delta})\Delta} \tag{7.16}$$

where $H(x) = -x\log x - (1-x)\log(1-x)$ is the *binary entropy function* [33]. Moreover, it can be shown that $\Phi_\Delta(D) \leq (\frac{\Delta e}{D})^D$ [98].

If $D \leq \Delta = k$, then it is clear that $\Phi_\Delta(D)$ is also an upper bound for $N_k(\Delta, D)$. However, as we move to higher dimensions, keeping Δ and D fixed, we should be able to construct larger graphs. Again, the reason for that is that we can move along the extra dimensions in order to avoid 'collisions'. In other words, if $D \leq \Delta = k < k'$, then

$$\Phi_\Delta(D) \leq N_{k'}(\Delta, D) \leq \Phi_{k'}(D) \tag{7.17}$$

So, let $k = \Delta + r$, with $r \in \mathbb{N}$. We are interested in finding the smallest value of r such that $N_{\Delta+r}(\Delta, D)$ becomes strictly larger than $\Phi_\Delta(D)$. For example, we know that for $r = (\Delta-1)\lfloor D/2 \rfloor$ we can construct a full tree of degree Δ and height $\lfloor D/2 \rfloor$, with diameter $2\lfloor D/2 \rfloor$, and order

$$M_{\Delta,\lfloor D/2 \rfloor} = 1 + \Delta\,\frac{(\Delta-1)^{\lfloor D/2 \rfloor} - 1}{\Delta - 2} = \frac{\Delta(\Delta-1)^{\lfloor D/2 \rfloor} - 2}{\Delta - 2}, \tag{7.18}$$

which grows faster than $\Phi_\Delta(D)$. Therefore, we have the following lower bound:

★ **Proposition 7.3** *If $\Delta \geq D$, $N_k(\Delta, D) \geq \max\{\Phi_\Delta(D), M_{\Delta,\lfloor D/2 \rfloor}\}$*

□

It remains to be determined if the tree construction can also be achieved for smaller r, and the exact values of Δ and D for which $M_{\Delta,\lfloor D/2 \rfloor} > \Phi_\Delta(D)$. We have determined those values empirically for $3 \leq \Delta \leq 12$, $1 \leq D \leq 12$, and results are shown in Table 7.1. The actual values of $M_{\Delta,\lfloor D/2 \rfloor}$ and $\Phi_\Delta(D)$ can be consulted in the appendices.

		D										
		1	2	3	4	5	6	7	8	9	10	11	12
	3			+									
	4			+	+								
	5					+							
	6					+	+						
Δ	7					+	+	+					
	8					+	+	+	+				
	9					+	+	+	+	+			
	10					+	+	+	+	+	+		
	11					+		+	+	+	+	+	
	12					+		+	+	+	+	+	+

Table 7.1: Values of Δ and D for which $\frac{\Delta(\Delta-1)^{\lfloor D/2 \rfloor}-2}{\Delta-2} > \Phi_\Delta(D)$

7.2.2 Second case: Larger diameter

The case $\Delta < D \leq k$ seems to be more difficult than the former, but at least for $D = \Delta+1 = k$ we can give a construction that achieves the upper bound 2^D:

★ **Theorem 7.5** *Let $k = D = \Delta + 1 \geq 4$. Then $N_k(\Delta, D) = N_D(D-1, D) = \Phi_D(D) = 2^D$.*

Proof: We know that Q_k consists of four copies of Q_{k-2}, with edges between them. Let us arbitrarily label the copies of Q_{k-2} with the letters A, B, C, D. Copy A is joined to B and C, but not D. In our construction we will take the four copies of Q_{k-2}, and we will add some of the 'inter-copy' edges. Pick an arbitrary vertex v of A, and partition the vertices of A into two classes, one consisting of the vertices at even distance from v (including v itself), and the other one consisting of the vertices at odd distance from v. We will call them the 'even' and 'odd' class, respectively. Now we add the edges joining the even class to B, and the edges joining the odd class to C, and thereafter there is only one way to join B and C with D, keeping the degree equal to $\Delta = k - 1$. Figures 7.9 and 7.10 show the construction for $\Delta = 3$ and $\Delta = 4$, respectively.

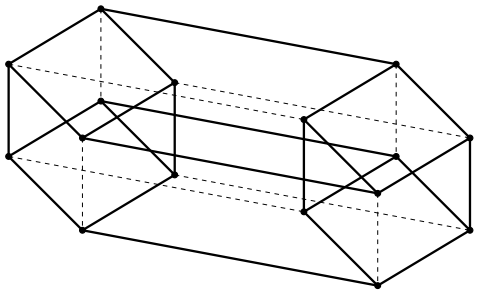


Figure 7.9: Construction for $\Delta = 3$ and $D = 4$

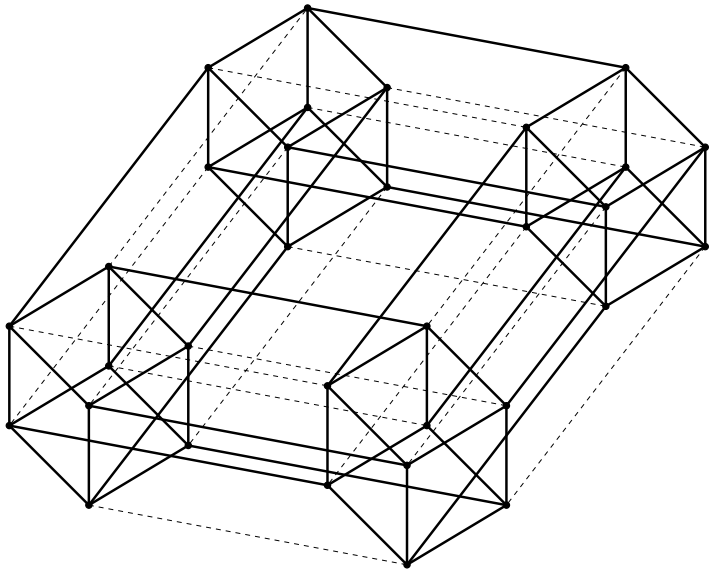


Figure 7.10: Construction for $\Delta = 4$ and $D = 5$

Now we only need to show that the resulting graph has diameter $\Delta + 1 = k$. W.l.o.g. the proof reduces to show that from a vertex $v \in A$ we can reach any other vertex w in at most k steps. If $w \in A$ we can reach it in at most $k-2$ steps, since Q_{k-2} has diameter $k-2$. Now let $w \in B \cup C$. Assume w.l.o.g that $w \in B$. If w is connected to A, then we can trivially reach v from w in at most $k-1$ steps. If w is not connected to A, then the $k-2$ neighbours of w belonging to B are connected to A. Each one of the neighbours of w is connected to a different vertex of A, and at least one of these must be reachable from v in less than $k-2$ steps, since there is only one vertex in A at distance $k-2$ from v. [1] There are, in fact, $k-3$ disjoint paths of length $< k$ from v to w.

Finally let $w \in D$. Either w is connected to B or w is connected to C; let us assume that it is connected to $u \in B$. Then we have $k-3$ paths of length at most $k-1$ from v to u, and the result follows.

□

7.3 What next?

This chapter leaves many questions open. In contrast with the square grid or mesh, which was its own dual tiling, the honeycomb grid has a dual tiling that is different from itself, namely the *triangular tiling* (Figure 7.11). It is interesting to replicate the study made for the square mesh and the honeycomb grid with the triangular tiling, thus completing the picture for regular tilings. Above all, it would be interesting to establish a connection between the results for the honeycomb grid and the triangular tiling, and the fact that they are duals. Then, the next logical move could be towards semiregular tilings of the plane, and spaces of higher dimension.

[1] This is the reason why we have to take $k > 3$.

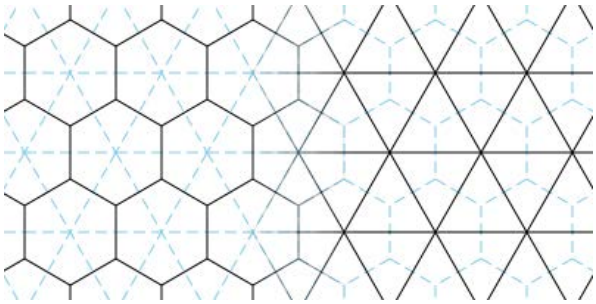


Figure 7.11: Duality between hexagonal and triangular tilings

The issues discussed in Section 6.4 also apply here. In particular, there is a fertile ground for devising *practical* constructions, i.e. constructions that possess additional nice properties, besides low degree and diameter (e.g. symmetry and fault-tolerance).

In the study of the hypercube we have yet to find upper bounds for $N_k(\Delta, D)$ other than the trivial bounds. In the case $D \leq \Delta \leq k$, we need to refine the lower bounds, and solve the inequality $\frac{\Delta(\Delta-1)^{\lfloor D/2 \rfloor}-2}{\Delta-2} > \Phi_\Delta(D)$ more precisely. We also need lower bounds for $\Delta < D-1 \leq k$.

8

Computational approach to MaxDDBS

After studying the Degree-Diameter Subgraph Problem from the mathematical viewpoint in different classes of host graphs, in this Chapter we address the problem from the computational viewpoint, and we propose a heuristic algorithm to deal with it, given its intractable nature. We analyze the performance of the algorithm both theoretically and experimentally. These results have been published in [44].

8.1 Heuristic algorithm for MaxWDDBS

The problems MaxDDBS and MaxWDDBS have not been studied in their full generality from the computational perspective yet, but some special cases and single-constraint restrictions have. For example, [95] gives an approximation algorithm for a special case: Finding the minimum-diameter spanning tree with bounded degree. That paper also considers edge weights, but restricts the host graph to be a complete graph.

A variant of the Largest Degree-Bounded Subgraph Problem was studied in [5], where the goal was to maximize the number of edges, and the problem was proved to be not in Apx for any $\Delta \geq 2$, and two polynomial-time approximation algorithms were given: one with ratio $\min\{m/\log n, n\Delta/(2\log n)\}$ for unweighted graphs, and another one with ratio $\min\{n/2, m/2\}$ for general weighted graphs. Obviously, the latter can also be used for unweighted graphs by taking all the weights equal to 1.

Our goal here is to maximize the number of vertices, and we give a greedy algorithm for the unweighted case (Algorithm 8 below). The algorithm works with two sets of vertices: dead and alive. A vertex is dead if it has already reached the maximum degree Δ, or if it has no more incident edges that can be added; otherwise it is alive. The algorithm grows the desired subgraph from a star, by adding edges incident to the live vertices.

Some remarks are due in Algorithm 8. First of all we need to clarify the somewhat vague expressions 'if possible' (Step 4), and 'if necessary' (Step 5), which are meant to simplify the

Algorithm 8: Unweighted bounded-degree subgraph

Input : A connected undirected graph $G = (V, E)$, and an integer $2 \leq \Delta \leq |V|$.
Output: A subgraph $S = (V_S, E_S)$ of G, such that $\deg_S(v) \leq \Delta$ for all $v \in V_S$, and $|V_S|$ is maximized.

1 Initially, take S as any Δ-star (a star of degree Δ) of G;
2 Mark the central vertex of S as killed (the other vertices of G are alive);
3 **repeat**
4 Pick a live vertex $u \in V_S$ (with $\deg_S(u) < \Delta$) and add an incident edge $e = (u, v)$ to S (if possible);
5 Mark u or v (or both) as killed, if necessary;
6 **until** *no more edges can be added*;
7 Return $S = (V_S, E_S)$;

structure of the algorithm. In Step 4, if there exists a live vertex u, with $\deg_S(u) < \Delta$, and having an incident edge $e = (u, v)$, where v is also alive, then we pick both u and e. This could also involve adding v to S, if $v \notin S$. If there is no such e, or if it exists, and $\deg_S(u)$ reaches $\min(\Delta, \deg_G(u))$ after the addition of e, then we mark u as killed in Step 5. Similarly, v is marked as killed if $\deg_S(v)$ reaches $\min(\Delta, \deg_G(v))$ after the addition of e.

Note also that Steps 1 and 4 of Algorithm 8 are left somewhat undetermined. The choice of the initial Δ-star in Step 1, and the live vertex u and the incident edge e in Step 4, can greatly affect the outcome. In our implementation, reported in Section 8.2, we chose in Step 4 the vertex u with the highest $\deg_S(u)$, and picked the incident edge $e = (u, v)$ so that v was not in S, or with the lowest $\deg_S(v)$. This choice of e is obvious, since we want S to have as many vertices as possible. This combination of choices results in an approximately breadth-first exploration of the graph, which performed better than depth-first or random heuristics. In the next section we give some experimental results that support our claim.

As for the initial Δ-star, a natural choice is to construct it around the vertex with the highest degree in G; nevertheless we discuss some alternatives below. Whatever the heuristic, in the worst case we can guarantee the following result:

★ **Lemma 8.1** *Algorithm 8 produces a polynomial-time approximation of the Largest Degree-Bounded Subgraph, with a worst-case approximation ratio* $\frac{\min(n, N_{\Delta,D})}{\Delta+1}$.

Proof: First let us check that the algorithm is correct, and it runs in polynomial time. The initial Δ-star is guaranteed to exist by the assumption that Δ does not exceed the maximum degree of G. Then, the 'killed' tags guarantee that no vertex of S will exceed the upper bound Δ. Finally, the way that S is grown (by adding edges incident to S) ensures that S will be connected, given that G is itself connected. The number of iterations is bounded above by the number of edges of G, hence the algorithm runs in polynomial time in the size of G.

Now, for the approximation ratio, we have that $|V_S| \geq \Delta + 1$ in the worst case (that is, when S consists only of the initial Δ-star), and we know that the number of vertices of the optimal solution is at most $\min(n, N_{\Delta,D})$. Therefore, $\frac{OPT}{ALG} \leq \min(\frac{n}{\Delta+1}, \frac{N_{\Delta,D}}{\Delta+1}) = \frac{\min(n, N_{\Delta,D})}{\Delta+1}$. That completes the proof. □

This approximation ratio is tight, as shown by the following example:

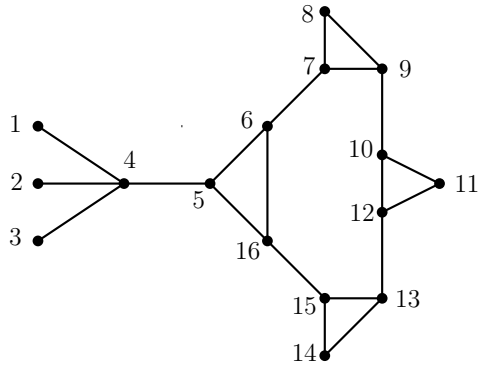


Figure 8.1: A worst-case input for Algorithm 8

Figure 8.1 depicts a worst-case input for Algorithm 8, assuming that $\Delta = 3$. The vertex with the highest degree is 4, so the initial 3-star could be made up by the vertices 1, 2, 3, and 4, and after that, no more vertices may be added. On the other hand, an optimal solution would consist of all the vertices, except 2. A remedy for this situation would be to modify Step 1, so as to choose a vertex with higher centrality around which to construct the initial Δ-star. For example, in a tradeoff between cost and effectiveness we could compute G^2, and take the vertex x with the highest degree in G^2, having $\deg_G(x) \leq \Delta$. However, with this heuristic it is also possible to construct an example with roughly the same approximation ratio, as shown in Figure 8.2.

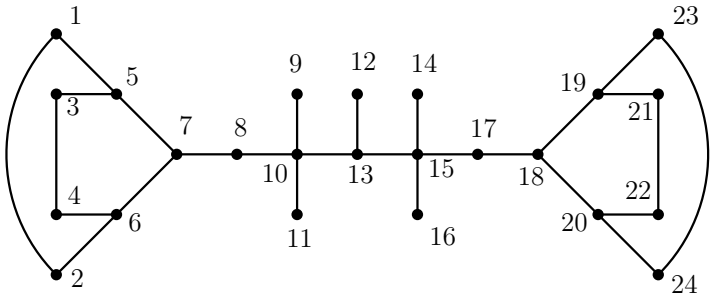


Figure 8.2: Another worst-case input for Algorithm 8

Again, let $\Delta = 3$. In this case, vertex 13 will have the largest degree in G^2, and will thus be used as the center of the initial 3-star. Then, the algorithm might return $S = \{9, 10, 11, 12, 13, 14, 15, 16\}$, whereas an optimal solution consists of all the vertices, except 11 and 16.

Nevertheless, in practice the algorithm seems to behave better than this, but an average-case analysis of its performance still remains open.

Another single-constraint restriction of MaxDDBS, the Largest Diameter-Bounded Subgraph Problem, was studied in [7], where it was also proved not to be in Apx, and a polynomial-time approximation algorithm with ratio $\mathcal{O}(n^{1/2})$ was given (Algorithm 9). This algorithm can be

generalized in a straightforward manner to the case where $G = (V, E)$ is a weighted graph. First we have to define the d-th power of G appropriately: We compute the distance between every pair of vertices $u, v \in V$, and then we add a new edge (u, v), with weight equal to $dist(u, v)$, if $dist(u, v) \leq d$. If the weight of some edge (u, v) is greater than d, we delete (u, v) from G. Now, with G^d so defined, we can apply Algorithm 9 directly.

Algorithm 9: Diameter-bounded subgraph

Input : A connected weighted undirected graph $G = (V, E)$, and an integer $1 \leq D \leq |V|$.
Output: A subgraph $T = (V_T, E_T)$ of G, such that $diam(T) \leq D$, and $|V_T|$ is maximized.

1 Compute the $\lfloor D/2 \rfloor$-th power, $G^{\lfloor D/2 \rfloor}$, of G;
2 Find the vertex $v \in G^{\lfloor D/2 \rfloor}$ with the maximum degree, and let $T^* = (V_T, E_T)$ be the subgraph of $G^{\lfloor D/2 \rfloor}$ induced by v and its neighbours;
3 Output $T = (V_T, E \cap E_T)$;

For arbitrary weights, it is very difficult to predict the size of the subgraph T produced by Algorithm 9. However, for practical purposes it is not unreasonable to assume that D is large compared to the weights. If D is large enough, larger than the weight of any single edge, then no edges will be deleted from G in the computation of $G^{\lfloor D/2 \rfloor}$, and we get the following

Proposition 8.1 *[7] Under the assumptions above, Algorithm 9 produces a polynomial-time approximation of the Largest Diameter-Bounded Subgraph, with approximation ratio* $\mathcal{O}(n^{1/2})$.

We can also get a bound for the *best case*:

★ **Proposition 8.2** *Let* $\Delta > 2$ *be maximum degree of the host graph* G. *Then, the order of the subgraph constructed by Algorithm 9 is bounded above by* $\frac{\Delta(\Delta-1)^{\lfloor D/2 \rfloor}-2}{\Delta-2}$.

Proof: Reasoning as in Proposition 3.1, we can limit ourselves w.l.o.g. to the version without weights. The order of the subgraph $S \subseteq G$ constructed by Algorithm 9 equals the order of the largest star in $G^{\lfloor D/2 \rfloor}$. Let x be the center of this star. Now, if we expand back this star as a subgraph of G, we will get at most Δ vertices at distance one from x, each one of them having at most $\Delta - 1$ descendants, and so on. Therefore, the total number of vertices of S will be at most $M_{\Delta,\lfloor D/2 \rfloor} = \frac{\Delta(\Delta-1)^{\lfloor D/2 \rfloor}-2}{\Delta-2}$. □

The latter result translates directly into a lower bound for the approximation ratio, provided that we have a lower bound for the order of an optimal solution in G. This is the case for $G = K_n$, and $3 \leq \Delta \leq 20$, $2 \leq D \leq 10$ [63], and when G is the two-dimensional square grid, for $\Delta = 3$ and $6 \leq D \leq 16$ [124].

Note also that, because of the floor function, Algorithm 9 returns a solution of the same order for $D = 2p$ and $D = 2p + 1$.

Clearly, if we perform Algorithm 8 on a graph G, and then Algorithm 9 on the resulting subgraph S, we get a heuristic polynomial-time algorithm for MaxDDBS. If G is a weighted

graph, in the first stage we can make all the weights equal to one and apply Algorithm 8. Then, in the second stage we restore the original weights, and perform Algorithm 9 with weights, thus getting an approximation of MaxWDDBS. Putting it all together we get the following

★ **Theorem 8.1** *If D is large enough, the greedy algorithm outlined above approximates* MaxWDDBS *in polynomial time, with a worst-case approximation ratio* $\frac{\min(n,N_{\Delta,D})}{\Delta+1}$.

Proof: As we said before, the algorithm works by performing Algorithm 8 on G without weights, and then Algorithm 9 on the output subgraph S with weights. The output S of Algorithm 8 contains a Δ-star, and so does the output T of Algorithm 9. Hence T contains at least $\Delta+1$ vertices, and again, $\frac{OPT}{ALG} \leq \min(\frac{n}{\Delta+1}, \frac{N_{\Delta,D}}{\Delta+1}) = \frac{\min(n,N_{\Delta,D})}{\Delta+1}$.

It is clear that the algorithm runs in polynomial time. □

We should remark again that $N_{\Delta,D}$ is not known exactly for most combinations of Δ and D. We have upper bounds (equal or very close to the Moore bound in most cases), and a table of lower bounds, i.e. the largest graphs that have been found so far [63]. Assuming that G is large enough, so that $n > N_{\Delta,D}$, and replacing $N_{\Delta,D}$ by $M_{\Delta,D}$, we get that $\frac{OPT}{ALG} \leq \mathcal{O}(\Delta^{D-1})$. Again, this bound is tight, since Algorithm 9 works on the subgraph produced by Algorithm 8, which in the worst case is the largest admissible star. This does not look like a very promising result, but the algorithm works in practice much better than this, as we shall see in the next section.

On the other hand, as a consequence of Proposition 8.2, if G is the complete graph K_n, with $n \geq M_{\Delta,D}$, then an optimal solution will have order at most $M_{\Delta,\lfloor D/2 \rfloor}$, and the best approximation ratio will be $\frac{M_{\Delta,D}}{M_{\Delta,\lfloor D/2 \rfloor}} = \mathcal{O}((\Delta-1)^{D/2})$, since in this case we do know that an optimal solution will have order $\approx M_{\Delta,D}$. This is also likely to happen in large dense graphs, other than K_n.

8.2 Experimental results

In Section 8.1 we have provided a heuristic strategy to solve MaxWDDBS, and we proved some approximation ratios for the solution. But how good is the algorithm in practice? We have implemented our algorithm and tested it on a variety of unweighted networks. There are currently two implementations of the algorithm: one in MATLAB®, written by myself, and another one in Java™, written by A. Dekker, who also assisted with experiments 1, 3, and 4. The second experiment is mine, and experiments 1 and 3 were independently confirmed by myself with the MATLAB® program. Additionally, I carried out other experiments on Erdös-Rényi random graphs, which are not included here because they do not add any new insight, just confirm the results of Experiment 3. Our experiments show that the average performance of the algorithm is better than the worst case performance given in Theorem 8.1.

8.2.1 First experiment

Our first experiment supports the tie-breaking rules that we chose for Algorithm 8, as explained in Section 8.1. We computed an approximation of MaxDDBS on three 100-vertex host graphs,

with $\Delta = 3$ and $D = 6$, using four different rules for choosing $u \in V_S$ and v. The three input graphs were: the antiprism, the complete graph K_{100}, and a toroidal square grid. The value of 13 obtained for the antiprism is optimal. Note that the fourth rule, which approximates breadth-first search, produces better results than the other three.

	Input Graph		
Preferred (u, v) choice	Antiprism	K_{100}	Toroidal square grid
Low degree u	13	4	12
High degree u	11	4	12
Low degree v and u	12	17	17
Low degree v, high u	13	22	20

Table 8.1: Average orders of subgraphs found for $\Delta = 3$ and $D = 6$, with different choices of u and v

8.2.2 Second experiment: Performance on the mesh

For an arbitrary graph we only have a very loose upper bound of the optimal solution, namely $\min(\Delta+1, N_{\Delta,D})$, which in turn leads to pessimistic approximation ratios. However, in Section 6.1 we derived better upper bounds for the mesh, which we can now use to get more accurate approximation ratios.

Thus, we have tested the algorithm on a sufficiently large two-dimensional square grid, and computed the MaxDDBS for $\Delta = 3$ and $2 \leq D \leq 20$. The results are summarized in Table 8.2. For even diameter, the approximation ratio lies between 1.3 and 1.17, with a tendency to decrease as D grows. For odd diameter the approximation ratios are somewhat larger, but still good and decreasing. Overall, the results compare reasonably well with the largest known subgraphs, depicted in [124].

Diameter	2	3	4	5	6	7	8	9	10	11
Upper bound	5	8	13	18	25	32	41	50	61	72
Algorithm result	4	4	10	10	20	20	33	33	49	49
Approximation ratio	1.25	2	1.3	1.8	1.25	1.6	1.24	1.51	1.25	1.47
Diameter	12	13	14	15	16	17	18	19	20	
Upper bound	85	98	113	128	145	162	181	200	221	
Algorithm result	69	69	93	93	121	121	153	153	189	
Approximation ratio	1.23	1.42	1.21	1.38	1.2	1.3	1.18	1.3	1.17	

Table 8.2: Algorithm performance on the two-dimensional grid, with $\Delta = 3$ and $2 \leq D \leq 20$.

Figure 8.3 also shows the performance of the algorithm graphically. The smooth upper curve represents the upper bound of the MaxDDBS, while the jagged curve below corresponds to the actual performance of the algorithm. The subgraphs constructed by the algorithm follow an interesting pattern; Figure 8.4 shows the one for $D = 16$.

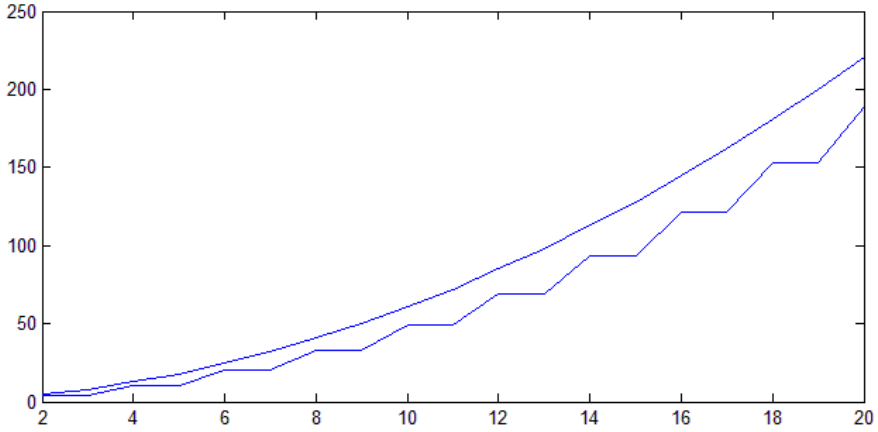


Figure 8.3: Algorithm performance on the two-dimensional grid, with $\Delta = 3$ and $2 \leq D \leq 20$

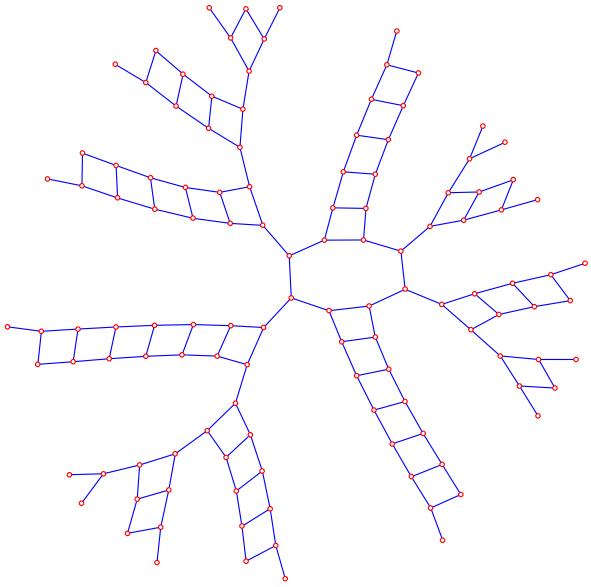


Figure 8.4: Subgraph constructed by the algorithm on the two-dimensional grid, with $\Delta = 3$ and $D = 16$

8.2.3 Third experiment: Some random graphs

In our third experiment we run the algorithm with $\Delta = 3$ and $D = 6$ on a range of input graphs of order N (ranging from 100 to 300), produced by the Watts rewiring process [149] on an N-vertex antiprism (i.e. with average degree 4). This process relocates edges at random with probability p. Figure 8.5 illustrates an example, where the probability p of an edge being rewired is 0.05. For such small values of p, the graphs produced are known as 'Small World' graphs or Watts-Strogatz graphs. For $p = 1$, the process produces graphs very close to Erdős-Rényi random graphs.

For comparison, we also use Barabási-Albert 'Scale Free' graphs, which are created by a preferential attachment process, which generates an approximately power-law distribution of node degrees [10, 9]. For comparison to the rewired antiprisms, these were constructed to also have average degree 4. In addition, toroidal square grids, and complete graphs were also used. Since the rewiring and preferential attachment processes are random in nature, Table 8.3 and Figure 8.6 show the average results over 50 runs. For each result, Table 8.3 also gives an upper bound of the approximation ratio; this is the number in parentheses right below.

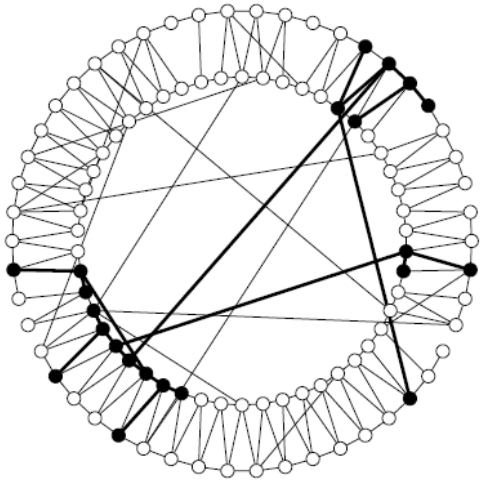


Figure 8.5: A 100-vertex antiprism rewired with a probability $p = 0.05$ of edges being randomly relocated. The 22 vertices of the $\Delta = 3, D = 6$ subgraph identified by our algorithm are marked in black. This figure was provided by Anthony Dekker for [44].

Note that for the 'Small World' graphs with $0.01 \leq p \leq 0.02$, results improve noticeably with the size of the input graph (increasing by 10–16%), presumably because of the greater probability of finding a segment of input graph with a topology that suits the algorithm (this is statistically significant at the 10^{-15} level). The effect is smaller for $0.05 \leq p \leq 0.5$ and for 'Scale Free' graphs (though still equally significant). However, the increase occurs primarily as the order N of the input graph increases from 100 to 200: beyond this point results begin to converge to an upper limit.

Subgraphs in this experiment were largest for random and approximately random networks with $0.2 \leq p \leq 1$ (statistically significant at the 10^{-15} level). For these graphs, the limiting

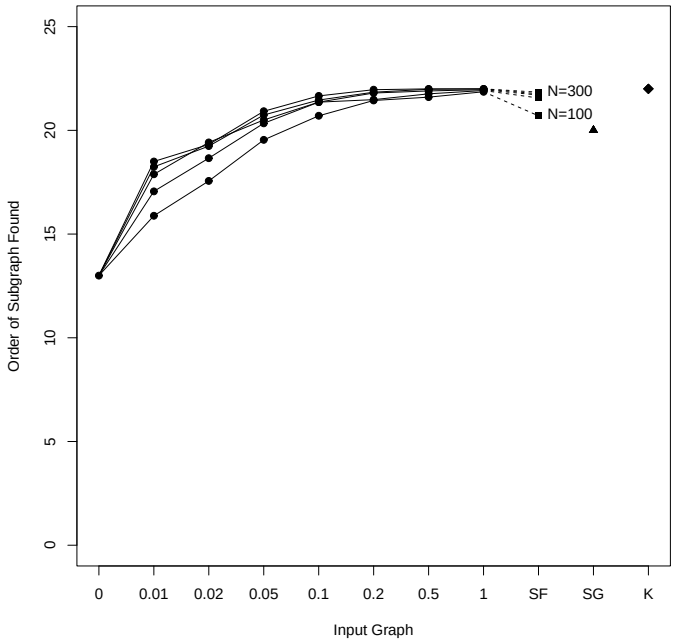


Figure 8.6: Average orders of subgraphs found for $\Delta = 3$ and $D = 6$ on various input graphs with order N ranging from 100 to 300: rewired antiprisms with $0 \leq p \leq 1$, 'Scale Free' graphs (SG), toroidal square grids (SG), and complete graphs (K). Courtesy of Anthony Dekker.

	Watts rewiring probability p										
	0	0.01	0.02	0.05	0.1	0.2	0.5	1	SF	SG	K_N
100	13	15.9	17.6	19.5	20.7	21.4	21.6	21.9	20.7	20	22
	(1)	(6.3)	(5.7)	(5.1)	(4.8)	(4.7)	(4.6)	(4.6)	(4.8)	(1.25)	(4.5)
150	13	17.1	18.7	20.3	21.4	21.5	21.8	21.9	21.6	20	22
	(1)	(8.8)	(8.0)	(7.4)	(7.0)	(7.0)	(6.9)	(6.8)	(7.0)	(1.25)	(6.8)
N 200	13	17.9	19.4	20.5	21.4	21.8	21.9	22.0	21.7	20	22
	(1)	(10.5)	(9.7)	(9.2)	(8.8)	(8.6)	(8.6)	(8.6)	(8.7)	(1.25)	(8.5)
250	13	18.2	19.2	20.7	21.5	21.8	22.0	22.0	21.8	20	22
	(1)	(10.3)	(9.8)	(9.1)	(8.8)	(8.6)	(8.6)	(8.5)	(8.6)	(1.25)	(8.5)
300	13	18.5	19.3	20.9	21.7	22.0	22.0	22.0	21.8	20	22
	(1)	(10.2)	(9.7)	(9.0)	(8.7)	(8.6)	(8.5)	(8.5)	(8.6)	(1.25)	(8.5)

Table 8.3: Average orders of subgraphs found for $\Delta = 3$ and $D = 6$ on rewired antiprisms and other graphs. SF refers to 'Scale Free' graphs, and SG to toroidal square grids. The numbers in parentheses are upper bounds of the approximation ratios.

value is 22, the result obtained for complete graphs K_N, which agrees with the prediction provided by Proposition 8.2. In other words, sufficiently large random or near-random input graphs with average degree only slightly more than the desired subgraph degree Δ behave like complete graphs: the additional edges in K_N provide no benefit.

The 13-vertex result for the unrewired antiprism ($p = 0$) is optimal. For the toroidal square grid, the optimal subgraph would have order 22. Apart from these two cases, an upper bound on the order of optimal subgraphs is $\min(N, 188)$, and we can thus calculate the upper bounds on approximation ratios in Table 8.3. Particularly for low values of p, these bounds are likely to be overly pessimistic. However, since a 132-vertex graph with $\Delta = 3$ and $D = 6$ is known [58], the approximation ratios for K_{200} and K_{300} are bounded below by 6.

8.2.4 Fourth experiment: A naturally occurring network

Our fourth experiment involves a graph of 516 genetic diseases, taken from the data displayed at `diseasome.eu`, and shown in Figure 8.7. This graph has 1188 edges, linking diseases that share one or more genes, and a diameter of 15. Applying our algorithm finds clusters of related diseases, of varying size. This illustrates the performance of our algorithm on naturally occurring graphs, as well as suggesting possible applications in medical research and systems biology.

Table 8.4 shows the orders of subgraphs found by the algorithm when applied to the human diseasome graph, for various values of Δ and D. Comparing to the known upper bounds for the degree-diameter problem [63] or to the order of the input graph (where that is lower), yields the upper bounds on approximation ratios. These are shown in parentheses right below each result. For $\Delta = 6$ and $D = 4$, the ratio of 17.2 results from comparing the subgraph of order 30 found by the algorithm with the order of the input graph, which is 516.

In summary, although our algorithm produces optimal results in some cases (such as the antiprism without rewiring), and good suboptimal results in other cases (such as the square

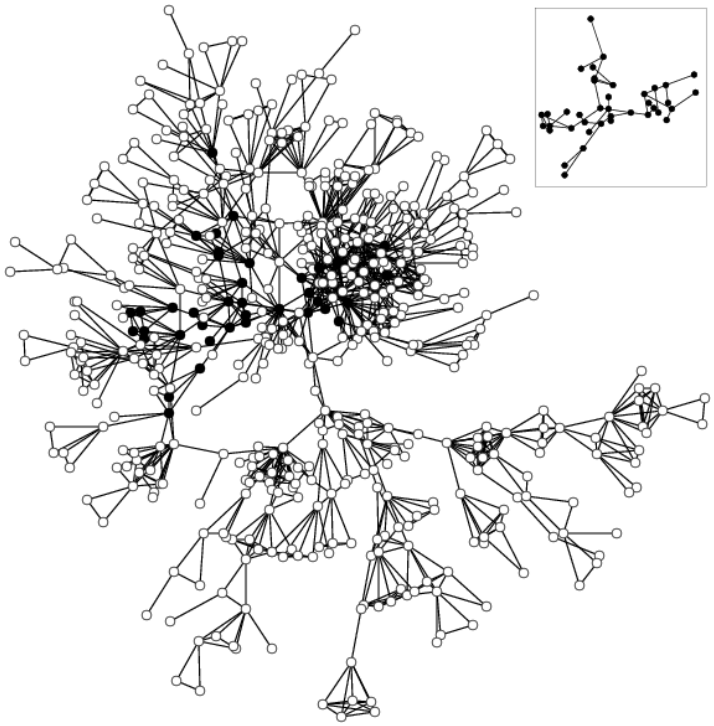


Figure 8.7: The human diseasome graph. The 39 vertices of the $\Delta = 3, D = 8$ subgraph identified by our algorithm are marked in black and highlighted in the inset on the upper right. Courtesy of Anthony Dekker.

	Diameter D								
	4	6	8	10	12	14	16	18	20
3	10	22	39	64	93	117	134	151	167
	(3.8)	(8.5)	(13.2)	(8.1)	(5.5)	(4.4)	(3.9)	(3.4)	(3.1)
4	17	40	74	118	167	210	246	284	309
	(9.4)	(12.9)	(7.0)	(4.4)	(3.1)	(2.5)	(2.1)	(1.8)	(1.7)
Δ 5	26	64	115	184	260	335	399	439	452
	(16.3)	(8.1)	(4.5)	(2.8)	(2.0)	(1.5)	(1.3)	(1.2)	(1.1)
6	30	79	145	240	315	402	442	461	462
	(17.2)	(6.5)	(3.6)	(2.2)	(1.6)	(1.3)	(1.2)	(1.1)	(1.1)
7	36	93	170	272	352	414	456	472	472
	(14.3)	(5.5)	(3.0)	(1.9)	(1.5)	(1.2)	(1.1)	(1.1)	(1.1)

Table 8.4: Orders of subgraphs found for the human diseasome graph.

grid), approximation ratios in the worst case (for the range of graphs tested) can rise to the range of 10 to 20. Typically, results lie between these two extremes. In general, performance on the complete graph K_N provides a limit to performance on randomly constructed input graphs of lower degree.

8.3 Some open questions

In this chapter we have studied the Maximum Degree&Diameter Bounded Subgraph Problem (MaxDDBS), and its weighted version (MaxWDDBS), from the computational perspective. Given that MaxDDBS and MaxWDDBS are computationally hard, we have proposed a heuristic algorithm for dealing with them, analysed its approximation ratio, and conducted a series of experiments to check its performance. Though the approximation ratio is quite high in the worst case, experimental results show that it can be a lot smaller in several practical situations.

These preliminary studies open up several interesting research lines for future development; here we summarize a few of them:

1. In several cases, the theoretical approximation ratio that we have found lies considerably above the real performance of the algorithm, as reflected in the experiments of Section 8.2. A more accurate theoretical analysis of the approximation ratio may involve computing $N_{\Delta,D}$, which remains elusive so far. On the other hand, for particular classes of host graphs it is possible to find tighter upper bounds for MaxDDBS, rather than using $M_{\Delta,D}$, as we have done in the case of the mesh and the hexagonal grid. Besides being interesting in their own right, these upper bounds also provide the means to refine the approximation ratio, and evaluate the performance of the algorithm more accurately in that particular class of host graphs. This remains open for hypercubes, and for other classes of host graphs of practical interest, such as vertex-transitive, cube-connected cycles, butterflies, etc.

2. It is necessary to conduct a more comprehensive set of experiments on larger random and deterministic host networks of different types. This experimental analysis faces several practical issues as well, such as computing an upper bound for MaxDDBS in that particular network (other than $M_{\Delta,D}$), and generating large realistic connected random networks.

3. Regarding computational complexity, we already know that the problems MaxDDBS and MaxWDDBS are $\mathcal{NP}-$hard and not in Apx. Now, where do they stand in relation with other complexity class hierarchies (e.g. parallel, randomized, fixed-parameter, etc.)?. For example, are these problems fixed-parameter tractable?, and if so, devise an efficient data-reduction algorithm to deal with them.

4. So far we have concentrated on two parameters only: Δ and D. However, in order to address realistic situations, future algorithms should also take into account other parameters, like the ones discussed in Section 1.2 (connectivity, fault-tolerance, etc.).

9 Conclusion

In this final chapter we strive to make sense of all the different strands of our work, enumerate our main activities and results during the candidature, and summarize the most important research questions that remain open.

9.1 Summary of results

- In the classical Degree-Diameter Problem (DDP) we have contributed to raise the lower bounds with the aid of analytic constructions (graph compounding), and algebraic techniques (Cayley graphs and voltage assignment). The first technique yields large graphs of diameter 6, while the second class of techniques yields large bipartite graphs. In the case of algebraic techniques we have paved the way for further applications in the construction of other types of graphs. We have focused on the use of finitely presented groups and rewriting systems, thus departing from the more traditional approach based on permutation groups.
- As a natural generalization of the classical Degree-Diameter Problem we have introduced the Degree-Diameter Subgraph Problem (DDS). Then we have fixed three types of host networks, which are very popular as models for parallel architectures: the mesh, the honeycomb grid, and the hypercube. For these host networks we have computed Moore-like upper bounds, and we have constructed large subgraphs with the aid of *ad hoc* techniques. In our view, the main theoretical contribution of the thesis resides here.
- Furthermore, we have discussed complexity and algorithmic issues related to DDS and its weighted version (MaxWDDBS), which is a more realistic model in practical situations. We have designed a first practical algorithm to deal with the problem, and studied its performance. Theoretically, our algorithm performs rather poorly in the worst case, but it behaves much better in practice, as shown by our experiments. Our algorithm can be the basis for future, more efficient algorithms.

So far these are the results connected with our main research goals. As a bonus we have obtained some accessory results:

1. We have contributed to the formalization and organization of current knowledge about construction techniques in the Degree-Diameter Problem, and algebraic techniques in particular. We have commented some recent advances in the field, and highlighted the connection between seemingly unrelated problems and sources.

2. We have made some modest contributions to the theory of rewriting systems, which can be used for speeding up algebraic computations related with DDP.

3. We have implemented the following algorithms and computational procedures, thus expanding our arsenal of software tools for problem solving:

   - The Knuth-Bendix completion procedure for string-rewriting systems.
   - Construction of the derived graph, given a base graph and a finite group.
   - Algorithms for computing and estimating the diameter of a graph.
   - Our heuristic algorithm for solving MaxDDBS.
   - Generation of different graph classes, such as toroidal grids, antiprisms, and some random graphs.

4. Together with Eyal Loz, Geoffrey Exoo, and Guillermo Pineda-Villavicencio, we have set up the website http://combinatoricswiki.org/wiki, a collaborative repository on Combinatorics, which has already attracted several other researchers to publish their problems and results online.

9.2 Summary of research problems

Below we summarize the main research directions that either arise as a result of the thesis, or have been discussed here, but still contain open questions. In this summary we just give an overview of the open questions; for more details see the section on open problems at the end of each chapter. We follow the same order as in the body of the thesis, and start with algebraic techniques.

Hopefully, the reader is by now convinced about the potential of algebraic techniques, especially voltage assignment, for constructing large graphs with given degree and diameter. However, in order to unleash the full power of that technique we need to further develop the theory of rewriting systems and group theory. Let us start by enumerating the problems in those two areas, which are, in general, very difficult. Recall that we are mainly concerned with finitely presented groups.

- In order to apply the theory of rewriting systems here it is necessary to compute complete rewriting systems for several classes of groups, and efficient algorithms to find the normal form of any word, that do not rely on repeated application of the reduction rules. That would translate directly into efficient algorithms for computing the exact diameter of the corresponding Cayley graph.

- Regarding the generalization of Cayley graphs, voltage assignment, there already exist crude estimates for the diameter of the derived graph [155], and for its girth [62]. We need to improve the estimates for the diameter. In terms of group theory, that means computing the order of arbitrary elements.
- The above results have enabled to identify certain classes of groups that are suitable for constructing cages (e.g. perfect groups). It is interesting to apply those same groups to the Degree-Diameter Problem, and identify additional classes of groups that could be useful.
- In general, we want to obtain conditions on the base digraph and the voltage group, that enable us to predict properties of the derived graph (such as diameter, bipartiteness, planarity, vertex-transitivity, edge-transitivity, distance-transitivity, $k-$connectivity, etc.). With the aid of these conditions, develop a methodology to construct large graphs of some particular class via voltage assignment.

The Degree-Diameter Subgraph Problem was introduced in this thesis, and thus it is a virgin area for research. There is plenty to be done here:

- The work we have done for the mesh and the honeycomb grid can be extended to other regular and semiregular tilings in the plane and higher dimensions, and to other classes of host graphs of practical application, like the hypercube and its variants.
- In mid-term, we can investigate larger classes of host graphs, such as Cayley, vertex-transitive, bipartite, and planar. The idea is to establish a connection between the order of the largest subgraph of given degree and diameter and other graph parameters (e.g. eigenvalues, girth, chromatic number, etc.)
- For practical applications we must consider additional parameters, such as connectivity and fault-tolerance, symmetry, and others.
- Complex random networks are pervasive today, and DDS offers a broad scope for research here.

Finally, there are a lot of algorithmic and complexity issues, both in DDP and DDS:

- First of all, there is the issue of the complexity of DDP, which remains open.
- Computational techniques for solving DDP have not been studied systematically. There have only been isolated attepmts, that mostly make use of the techniques developed for other problems (e.g. local search with 2-Opt [1]). There is no comprehensive study, and no specific heuristics tailored for DDP.
- In the case of DDS, we must find better algorithms than the one we propose here, which is a combination of an algorithm for solving the Bounded-Degree Spanning Subgraph, and another one for the Largest Diameter-Bounded Subgraph. Heuristics must play an important role here, since MaxDDBS is not only $\mathcal{NP}$-hard, but it cannot be approximated either within a constant factor in polynomial time.

- We already know that MaxDDBS and MaxWDDBS are $\mathcal{NP}$-hard and not in Apx, but where do they stand in relation with other complexity classes? Furthermore, it is important to identify classes of host graphs where these problems can be solved, or can be approximated in polynomial time.

Bibliography

[1] Allwright, J. New (δ, d) graphs discovered by heuristic search. *Discrete Applied Mathematics 37-38* (1992), 3–8.

[2] Alspach, B. Cayley graphs with optimal fault tolerance. *IEEE Transactions on Computers 41*, 10 (1992), 1337–1339.

[3] Alspach, B. Cayley graphs. In *Topics in Algebraic Graph Theory*, L. W. Beineke, R. J. Wilson, and P. J. Cameron, Eds. Cambridge University Press, 2004, pp. 156–178.

[4] Alspach, B. Cayley graphs. In *Handbook of Graph Theory*, J. L. Gross and J. Yellen, Eds. CRC Press, 2004, pp. 505–515.

[5] Amini, O., Peleg, D., Perennes, S., Sau, I., and Saurabh, S. Degree-constrained subgraph problems: Hardness and approximation results. In *Proceedings of ALGOWAOA* (2008), vol. LNCS 5426, Springer Verlag, pp. 29–42.

[6] Arkhipova, L. Number of lattice points in a sphere. *Moscow Univ. Math. Bull. 63* (2008), 214–215.

[7] Asahiro, Y., Miyano, E., and Samizo, K. Approximating maximum diameter-bounded subgraphs. In *Proceedings LATIN* (2010), vol. LNCS 6034, Springer Verlag, pp. 615–626.

[8] Bang-Jensen, J., and Gutin, G., Eds. *Digraphs. Theory, Algorithms and Applications.* Springer Verlag, Berlin, 2007.

[9] Barabási, A. *Linked: The New Science of Networks.* Perseus Publishing, Cambridge, Massachussets, 2002.

[10] Barabási, A.-L., and Albert, R. Emergence of scaling in random networks. *Science 286* (1999), 509–512.

[11] Barvinok, A. Lattice points, polyhedra, and complexity. *IAS/Park City Math. Series, Amer. Math. Soc. 13* (2007), 19–62.

[12] Barvinok, A. *Integer Points in Polyhedra.* Eur. Math. Soc., Zurich, 2008.

[13] Barvinok, A., and Woods, K. Short rational generating functions for lattice point problems. *J. Amer. Math. Soc. 16* (2003), 957–979.

[14] Baskoro, E., Brankovič, L., Miller, M., Plesník, J., Ryan, J., and Širáň, J. Large digraphs with small diameter: A voltage assignment approach. *Journal of Combinatorial Mathematics and Combinatorial Computing 24* (1997), 161–176.

[15] Benson, S. Students ask the darnedest things: A result in elementary group theory. *Mathematics Magazine 70* (1997), 207–211.

[16] Bermond, J., Delorme, C., and Farhi, G. Large graphs with given degree and diameter III. In *Graph theory (Cambridge, 1981)*, vol. 62 of *North-Holland Mathematics Studies*. North-Holland, Amsterdam, 1982, pp. 23–31. doi:10.1016/S0304-0208(08)73544-8.

[17] Bermond, J., Delorme, C., and Farhi, G. Large graphs with given degree and diameter II. *Journal of Combinatorial Theory, Series B 36*, 1 (1984), 32–48. doi:10.1016/0095-8956(84)90012-1.

[18] Bermond, J., Delorme, C., and Quisquater, J. Tables of large graphs with given degree and diameter. *Information Processing Letters 15* (1982), 10–13.

[19] Biggs, N. *Algebraic Graph Theory*, 2nd. ed. Cambridge University Press, Cambridge, 1993.

[20] Blum, C., and Roli, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys 35* (2003), 268–308.

[21] Bollobás, B. *Modern Graph Theory*. Springer Verlag, New York, 1998.

[22] Bondy, J., and Delorme, C. New large bipartite graphs with given degree and diameter. *Ars Combinatoria 25C* (1988), 123–132.

[23] Bondy, J., and Murty, U. *Graph Theory*, vol. 244 of *Graduate Texts in Mathematics*. Springer Verlag, New York, 2008.

[24] Book, R., and Otto, F. *String Rewriting Systems*. Springer Verlag, 1993.

[25] Borges-Trenard, M., and Borges-Quintana, M. Gröbner bases property on elimination ideal in the noncommutative case. In *Gröbner Bases and Applications. Proceedings of the Conference '33 Years of Gröbner Bases'* (1998), Cambridge University Press, pp. 323–337.

[26] Borges-Trenard, M., Pérez-Rosés, H., and Borges-Quintana, M. Gröbner basis property on elimination ideals in finite group theory. In *Proceedings of the Conference 'Logic, Mathematics and Computer Science: Interactions'* (2002), K. Nakagawa, Ed., pp. 61–69.

[27] Borges-Trenard, M., Pérez-Rosés, H., and Puente-Maury, L. Elimination in string rewriting systems. In *Proceedings of the Conference JMDA-08* (2008), J. Gimbert, Ed., pp. 493–500.

[28] Brankovič, L., Miller, M., Plesník, J., Ryan, J., and Širáň, J. Large graphs with small diameter: A voltage assignment approach. *Australasian Journal of Combinatorics 18* (1998), 65–76.

[29] Canale, E., and Gómez, J. Asymptotically large (Δ, D)-graphs. *Discrete Applied Mathematics 152*, 1-3 (2005), 89–108. doi:10.1016/j.dam.2005.03.008.

[30] Capobianco, M., and Molluzzo, J. *Examples and Counterexamples in Graph Theory*. North Holland, New York, 1978.

[31] Chwa, K.-Y. *Circulant Graphs and their Application to Communication Networks*. PhD thesis, Korea Advanced Institute of Science and Technology, 1992.

[32] Cohen, D. String rewriting - a survey for group theorists. In *Geometric Group Theory*, G. Niblo and M. Roller, Eds. Cambridge University Press, 1993, pp. 37–47.

[33] Cohen, G., Honkala, I., Litsyn, S., and Lobstein, A. *Covering Codes*. Elsevier, Amsterdam, 1997.

[34] Comellas, F. Table of the largest known graphs for maximum degrees $3 \leq \delta \leq 16$ and diameters $2 \leq d \leq 10$. http://www-mat.upc.es/grup_de_grafs/table_g.html. website.

[35] Comellas, F., and Fiol, M. Using simulated annealing to design interconnection networks. Tech. Rep. 05-290, Departament de Matemàtica Aplicada i Telemàtica. Universitat Politècnica de Catalunya, 1990.

[36] Comellas, F., and Gómez, J. New large graphs with given degree and diameter. In *Graph theory, combinatorics, and algorithms, (Kalamazoo, MI, 1992)*, vol. 1,2 of *Wiley-Interscience Publication*. Wiley, New York, 1995, pp. 221–233.

[37] Cooperman, G., and Finkelstein, L. New methods for using Cayley graphs in interconnection networks. *Discrete Applied Mathematics 37-38* (1992), 95–118.

[38] Coxeter, H., and Moser, W. *Generators and relations for discrete groups*. Springer Verlag, 1984.

[39] Crescenzi, P., Kann, V., Halldórsson, M., Karpinski, M., and Woeginger, G. A compendium of NP optimization problems. http://www.nada.kth.se/~viggo/wwwcompendium/. website.

[40] de Bruijn, N. A combinatorial problem. *Koninklijke Nederlandse Akademie v. Wetenschappen, Proc. Ser A 49* (1946), 758–764.

[41] Dekker, A. Simulating network robustness for critical infrastructure networks. In *Proceedings 28th Australasian Computer Science Conference* (2005), pp. 59–67.

[42] Dekker, A., and Colbert, B. Network robustness and graph topology. In *Proceedings 27th Australasian Computer Science Conference* (2004), pp. 359–368.

[43] Dekker, A., and Colbert, B. The symmetry ratio of a network. In *Proceedings 11th Computing: The Australasian Theory Symposium* (2005), pp. 13–20.

[44] Dekker, A., Pérez-Rosés, H., Pineda-Villavicencio, G., and Watters, P. The Maximum Degree & Diameter-Bounded Subgraph and its applications. *Journal of Mathematical Modelling and Algorithms* (2012). DOI:10.1007/s10852-012-9182-8.

[45] Delorme, C. Grands graphes de degré et diamètre donnés. *European Journal of Combinatorics 6* (1985), 291–302.

[46] Delorme, C. Examples of products giving large graphs with given degree and diameter. *Discrete Applied Mathematics 37/38* (1992), 157–167. doi:10.1016/0166-218X(92)90130-3.

[47] Delorme, C., and Gómez, J. Some new large compound graphs. *European Journal of Combinatorics 23*, 5 (2002), 539–547. doi:10.1006/eujc.2002.0581.

[48] Delorme, C., Gómez, J., and Quisquater, J. On large bipartite graphs. unpublished manuscript, 2011.

[49] Delorme, C., Loz, E., Pérez-Rosés, H., and Pineda-Villavicencio, G. Some properties of metabelian groups for constructing large graphs by voltage assignment. *in preparation* (2012).

[50] Diestel, R. *Graph Theory*, 3rd. ed., vol. 173 of *Graduate Texts in Mathematics*. Springer Verlag, New York, 2005.

[51] Dinneen, M., and Hafner, P. New results for the degree/diameter problem. *Networks 24* (1994), 359–367.

[52] Dixon, J., and Mortimer, B. *Permutation groups*. Springer Verlag, 1996.

[53] Dougherty, R., and Faber, V. The Degree-Diameter problem for several varieties of Cayley graphs i: The abelian case. *SIAM J. Disc. Math. 17* (2004), 478–519.

[54] Duch, J., and Arenas, A. Community identification using extremal optimization. *Physical Review E 72*, 027104 (2005).

[55] Durrett, R. *Random Graph Dynamics*. Cambridge University Press, New York, 2007.

[56] Elspas, B. Topological constraints on interconnection-limited logic. In *Proceedings of the IEEE Fifth Symposium on Switching Circuit Theory and Logical Design* (1964), vol. S–164, pp. 133–147.

[57] Exoo, G. Some new Ramsey colorings. *The Electronic Journal of Combinatorics 5*, R29 (1998), 000–000.

[58] Exoo, G. A family of graphs and the degree/diameter problem. *Journal of Graph Theory 37*, 2 (2001), 118–124.

[59] Exoo, G. Voltage graphs, group presentations, and cages. *The Electronic Journal of Combinatorics 11*, N2 (2004), 000–000.

[60] Exoo, G., and Jajcay, R. Properties of graphs for the cage and degree diameter problems. *Electronic Notes in Discrete Mathematics 34* (2009), 341–345.

[61] Exoo, G., and Jajcay, R. Dynamic cage survey. *The Electronic Journal of Combinatorics* (2011), 1–54. Dynamic survey DS16.

[62] Exoo, G., and Jajcay, R. On the girth of voltage graph lifts. *European Journal of Combinatorics 32* (2011), 554–562.

[63] Exoo, G., Loz, E., Pérez-Rosés, H., and Pineda-Villavicencio, G. Combinatorics wiki: The degree/diameter problem. http://combinatoricswiki.org/wiki/The_Degree/Diameter_Problem. website.

[64] Faradžev, I. Constructive enumeration of combinatorial objects. *Problèmes combinatoires et théorie des graphes. Colloq. Internat. CNRS, Paris 260* (1978), 131–135.

[65] Fiol, M., and Fábrega, J. Some compound graphs. *Stochastica 7*, 2 (1983), 137–143.

[66] Fischer, T., and Merz, P. A memetic algorithm for the optimal communication spanning tree problem. In *Proceedings of the 4th International Workshop on Hybrid Metaheuristics* (2007), vol. 4771, Springer Verlag, pp. 170—-184.

[67] Gao, S., and Novick, B. Fault tolerance of Cayley graphs. *Annals of Combinatorics 11* (2007), 161–171.

[68] Garey, M., and Johnson, D. *Computers and Intractability. A Guide to the Theory of NP-completeness.* W.H. Freeman and Sons., 1979.

[69] Gilman, R. Presentations of groups and monoids. *Journal of Algebra 57* (1979), 544–554.

[70] Gómez, J. Generalized compound graphs. *Ars Combinatoria 29B* (1990), 33–53.

[71] Gómez, J. On large $(\Delta, 6)$-graphs. *Networks 46* (2005), 82–87.

[72] Gómez, J. Some new large $(\delta, 3)$-graphs. *Networks 53* (2009), 1–5.

[73] Gómez, J., and Fiol, M. Dense compound graphs. *Ars Combinatoria 20A* (1985), 211–237.

[74] Gómez, J., Fiol, M., and Serra, O. On large (Δ, D)-graphs. *Discrete Mathematics 114* (1993), 219–235. doi:10.1016/0012-365X(93)90368-4.

[75] Gómez, J., and Miller, M. Two new families of large compound graphs. *Networks 47* (2006), 140–146.

[76] Gómez, J., Pelayo, I., and Balbuena, C. New large graphs with given degree and diameter six. *Networks 34*, 2 (1999), 154–161.

[77] Gross, J., and Tucker, T. *Topological Graph Theory.* Dover Publications, Minneola, New York, 2001.

[78] Gross, J., and Yellen, J., Eds. *Handbook of Graph Theory.* Discrete Mathematics and its Applications. CRC Press, Boca Raton, Florida, 2004.

[79] Grunbaum, B., and Shepard, G. *Tilings and Patterns.* W.H. Freeman and Sons., 1987.

[80] Hayes, J., Mudge, T., Stout, Q., Colley, S., and Palmer, J. A microprocessor-based hypercube supercomputer. *IEEE Micro 6* (1986), 6–17.

[81] Hempel, C. Metacyclic groups. *Communications in Algebra 28* (2000), 3865–3897.

[82] Heuberger, C. On the planarity and colorability of circulant graphs. *Discrete Mathematics 268* (2003), 153–169.

[83] Heydemann, M., and Ducourthial, B. *Cayley graphs and interconnection networks*. Rapports de recherche. Université de Paris-Sud, Centre d'Orsay, Laboratoire de Recherche en Informatique, 1997.

[84] Hoffman, A., and Singleton, R. On Moore graphs with diameter 2 and 3. *IBM Journal of Research and Development 4* (1960), 497–504.

[85] Holt, D. KBMAG (Knuth-Bendix in monoids and automatic groups). 1992.

[86] Holt, D., Eick, B., and O'Brien, E. *Handbook of Computational Group Theory*. Discrete Mathematics and its Applications. CRC Press, Boca Raton, 2005.

[87] Holub, P., Miller, M., Pérez-Rosés, H., and Ryan, J. The Maximum Degree & Diameter-Bounded Subgraph in the hypercube. *in preparation* (2012).

[88] Holub, P., Miller, M., Ryan, J., and Pérez-Rosés, H. The Degree-Diameter problem in honeycomb networks. *in preparation* (2012).

[89] Johnson, D. The NP-completeness column: An ongoing guide (14). *Journal of Algorithms 6* (1985), 145–159.

[90] Karp, R. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, J. W. T. R. E. Miller, Ed. Plenum Press, 1972, pp. 85–103.

[91] Kautz, W. Bounds on directed (d, k)-graphs. *Theory of Cellular Logic Networks and Machines Final Report AFCRL-68-0668* (1968), 20–28.

[92] Kavitha, T. Linear time algorithms for abelian group isomorphism and related problems. *Journal of Computer and System Sciences 73* (2007), 986–996.

[93] Knuth, D., and Bendix, P. Simple word problems in universal algebras. In *Computational Problems in Abstract Algebra*, J. Leech, Ed. Pergamon Press, 1970, pp. 263–297.

[94] Koblitz, N., Ed. *Algebraic Aspects of Cryptography*, vol. 3 of *Algorithms and Computation in Mathematics*. Springer Verlag, New York, 1998.

[95] Könemann, J., Levin, A., and Sinha, A. Approximating the degree-bounded minimum diameter spanning tree problem. *Algorithmica 41* (2005), 117–129.

[96] Korte, B., Lovász, L., and Schrader, R. *Greedoids*. Springer Verlag, New York, 1991.

[97] Kostrikin, A. I. *Introduction to Algebra*. Mir Publishers, Moscow, USSR, 1978.

[98] Kumar, N. Bounding the volume of Hamming balls. http://cstheory.wordpress.com/2010/08/13/bounding-the-volume-of-hamming-balls. website.

[99] Kurzweil, H., and Stellmacher, B. *The theory of Finite Groups: An Introduction*. Springer Verlag, New York, 2004.

[100] Kutnar, K., and Marušič, D. Recent trends and future directions in vertex-transitive graphs. *Ars Mathematica Contemporanea 1* (2008), 112–125.

[101] Lakshmivarhan, S., Jwo, J.-S., and Dhall, S. Symmetry in interconnection networks based on Cayley graphs of permutation groups: A survey. *Parallel Computing 19* (1993), 361–407.

[102] Ljubić, I. *Exact and Memetic Algorithms for two Network Design Problems.* PhD thesis, Technische Universitaet Wien, 2004.

[103] Loz, E. *The Degree-Diameter and Cage Problems: A Study in Structural Graph Theory.* PhD thesis, University of Auckland, New Zealand, 2009.

[104] Loz, E. Graphs of given degree and diameter obtained as abelian lifts of dipoles. *Discrete Mathematics 309* (2009), 3125–3130.

[105] Loz, E., Macaj, M., Miller, M., Šiagiová, J., Širáň, J., and Tomanová, J. Small vertex-transitive and Cayley graphs of girth six and given degree: an algebraic approach. *Journal of Graph Theory 68*, 4 (2011), 265–284.

[106] Loz, E., and Pineda-Villavicencio, G. New benchmarks for large scale small-world networks with given degree and diameter. *The Computer Journal 53* (2010), 1092–1105.

[107] Loz, E., and Širáň, J. New record graphs in the Degree-Diameter problem. *Australasian Journal of Combinatorics 41* (2008), 63–80.

[108] Lueker, G. Some techniques for solving recurrences. *ACM Computing Surveys 12* (1980), 419–436.

[109] Macbeth, H., Šiagiová, J., and Širáň, J. Cayley graphs of given degree and diameter for cyclic, abelian, and metacyclic groups. *Discrete Mathematics 312*, 1 (2012), 94–99.

[110] Macbeth, H., Šiagiová, J., Širáň, J., and Vetrík, T. Large Cayley graphs and vertex-transitive non-Cayley graphs of given degree and diameter. *Journal of Graph Theory 64*, 2 (2010), 87–98.

[111] McCool, J. The power problem for groups with one defining relator. *Proceedings of the American Mathematical Society 28* (1971), 427–430.

[112] McKay, B., Miller, M., and Širáň, J. A note on large graphs of diameter two and given maximum degree. *Journal of Combinatorial Theory, Series B 74* (1998), 110–118.

[113] McKay, B., and Royle, G. The transitive graphs with at most 26 vertices. *Ars Combinatoria 30* (1990), 161–176.

[114] Meijer, P. Connectivities and diameter of circulant graphs. Master's thesis, Simon Fraser University, 1991.

[115] Merz, P. *Memetic Algorithms for Combinatorial Optimization Problems: Fitness Landscapes and Effective Search Strategies.* PhD thesis, Universitaet-Gesamthochschule Siegen, 2000.

[116] Mestre, J. Greedy in approximation algorithms. In *Proceedings of ESA'06*, Y. Azar and T. Erlebach, Eds. Springer Verlag, 2006, pp. 528–539.

[117] Miller, M., Pérez-Rosés, H., and Ryan, J. The Maximum Degree & Diameter-Bounded Subgraph in the mesh. *Discrete Applied Mathematics 160* (2012), 1782—-1790.

[118] Miller, M., Tang, J., and Lin, Y. HSAGA and its application for the construction of near-Moore digraphs. *Journal of Discrete Algorithms 6* (2008), 73–84.

[119] Miller, M., and Širáň, J. Moore graphs and beyond: A survey of the degree/diameter problem. *The Electronic Journal of Combinatorics* (2005), 1–61. Dynamic survey DS14.

[120] Mora, T. Gröbner bases for non-commutative algebras. In *Proceedings of the ISSAC 88* (1989), Springer Verlag, pp. 150–161.

[121] Moscato, P. On evolution, search, optimization, genetic algorithms, and martial arts: Towards memetic algorithms. Tech. Rep. 826, California Institute of Technology, 1989.

[122] Moscato, P. Memetic algorithms - a short introduction. In *New Ideas in Optimization*, F. Corne and M. Dorigo, Eds. McGraw-Hill Publishing Co., 1999, pp. 37–47.

[123] OEIS-Foundation, T. OEIS: The on-line encyclopedia of integer sequences. http://oeis.org/classic/index.html. website.

[124] Pérez-Rosés, H. Combinatorics wiki: MaxDDBS in the mesh. http://combinatoricswiki.org/wiki/MaxDDBS_in_the_mesh. website.

[125] Petkovšek, M., Wilf, H., and Zeilberger, D. *A = B*. AK Peters, New York, 1996.

[126] Pineda-Villavicencio, G. *Topology of Interconection Networks With Given Degree and Diameter*. PhD thesis, University of Ballarat, Australia, 2009.

[127] Pineda-Villavicencio, G., Gómez, J., Miller, M., and Pérez-Rosés, H. New largest known graphs of diameter 6. *Networks 53* (2009), 315–328.

[128] Pineda-Villavicencio, G., and Miller, M. On graphs of maximum degree 3 and defect 4. *Journal of Combinatorial Mathematics and Combinatorial Computing 65* (2008), 25–31.

[129] Preparata, F., and Vuillemin, J. The Cube-Connected Cycles: A versatile network for parallel computation. *Communications of the ACM 24* (1981), 300–309.

[130] Quisquater, J. Structures d'interconnexion: Constructions et applications. Master's thesis, LRI, Université de Paris Sud, Orsay, France, Jul 1987.

[131] Ravi, R., Marathe, M., Ravi, S., Rosenkrantz, D., and III, H. B. H. Approximation algorithms for degree-constrained minimum-cost network-design problems. *Algorithmica 1* (2001), 58–78.

[132] Rosen, K., Michaels, J., Gross, J., Grossman, J., and Shier, D., Eds. *Handbook of Discrete and Combinatorial Mathematics*. CRC Press, Boca Raton, Florida, 2000.

[133] Sabidussi, G. On a class of fixed-point-free graphs. *Proceedings of the American Mathematical Society 9* (1958), 800–804.

[134] Sampels, M. Large networks with small diameter. In *Proceedings of the 23rd International Workshop on Graph-Theoretic Concepts in Computer Science* (London, Jun 1997), vol. 1335 of *Lecture Notes in Computer Science*, Springer-Verlag, pp. 288–302.

[135] Schibell, S., and Stafford, R. Processor interconnection networks from Cayley graphs. *Discrete Applied Mathematics 40* (1992), 333–357.

[136] Schröder, J. Generalized Schröder numbers and the rotation principle. *Journal of Integer Sequences 10* (2007), 07.7.7.

[137] Sims, C. *Computation with Finitely Presented Groups*. Cambridge University Press, 1994.

[138] Skriganov, M., and Sobolev, A. Variation of the number of lattice points in large balls. *Acta Arithmetica 120* (2005), 245–267.

[139] Sohaee, N., and Forst, C. Bounded diameter clustering scheme for protein interaction networks. In *Proceedings of the World Congress on Engineering and Computer Science* (2009), vol. I.

[140] Stojmenovic, I. Honeycomb networks: Topological properties and communication algorithms. *IEEE Trans. Parallel and Distributed Systems 8* (1997), 1036—-1042.

[141] Sulanke, R. Objects counted by the central Delannoy numbers. *Journal of Integer Sequences 6* (2003), 03.1.5.

[142] Trevisan, L. Inapproximability of combinatorial optimization problems. *Electronic Colloqium on Computational Complexity 00* (2004), 1–39.

[143] Vassilev-Missana, M., and Atanassov, K. On Delanoy [*sic*] numbers. *Annuaire Univ. Sofia Fac. Math. Inform. 81* (1987), 153–162.

[144] Vazirani, V. *Approximation Algorithms*. Springer Verlag, 2003.

[145] Vetrík, T., Simanjuntak, R., and Baskoro, E. Large bipartite Cayley graphs of given degree and diameter. *Discrete Mathematics 311* (2011), 324–326.

[146] von Dyck, W. Gruppentheoretische Studien. *Mathematische Annalen 20* (1882), 1–44.

[147] Šiagiová, J. A note on the McKay-Miller-Širáň graphs. *Journal of Combinatorial Theory, Series B 81* (2001), 205–208.

[148] Šiagiová, J., and Vetrík, T. Large vertex-transitive and Cayley graphs with given degree and diameter. *Electronic Notes in Discrete Mathematics 28* (2007), 365–369.

[149] Watts, D., Ed. *Six Degrees: The Science of a Connected Age*. W. W. Norton & Company, New York, 2003.

[150] Widmer, M. Lipschitz class, Narrow class, and counting lattice points. Tech. Rep. 13, Graz University of Technology, 2010.

[151] Williamson, D., and Shmoys, D. *The Design of Approximation Algorithms*. Cambridge University Press, 2010.

[152] Winkler, F. Reducing the complexity of the Knuth-Bendix completion algorithm: a 'unification' of different approaches. In *Proceedings of the European Conference on Computer Algebra (EUROCAL '85)* (1985), vol. 2, Springer Verlag, pp. 378–389.

[153] Wohlmuth, O. A new dense group graph discovered by an evolutionary approach. In *Proceedings of the Workshop ueber wissenschaftliches Rechnen* (1996), pp. 21–27.

[154] Xiao, W., and Parhami, B. Cayley graphs as models of deterministic small-world networks. *Information Processing Letters 97* (2006), 115–117.

[155] Zlatoš, A. The diameter of lifted digraphs. *Australasian Journal of Combinatorics 19* (1999), 73–82.

Appendices

Tables of combinatorial functions

		D											
		1	2	3	4	5	6	7	8	9	10	11	12
	3	4	4	10									
	4	5	5	17	17								
	5	6	6	26	26	106							
	6	7	7	37	37	187	187						
Δ	7	8	8	50	50	302	302	1814					
	8	9	9	65	65	457	457	3201	3201				
	9	10	10	82	82	658	658	5266	5266	42130			
	10	11	11	101	101	911	911	8201	8201	73811	73811		
	11	12	12	122	122	1222	1222	12222	12222	122222	122222	1222222	
	12	13	13	145	145	1597	1597	17569	17569	193261	193261	2125873	2125873

Table 1: Values of $\frac{\Delta(\Delta-1)^{\lfloor D/2 \rfloor}-2}{\Delta-2}$ for $3 \leq \Delta \leq 12$, $1 \leq D \leq 12$, and $D \leq \Delta$.

		D												
		0	1	2	3	4	5	6	7	8	9	10	11	12
	0	1												
	1	1	2											
	2	1	3	4										
	3	1	4	7	8									
	4	1	5	11	15	16								
	5	1	6	16	26	31	32							
Δ	6	1	7	22	42	57	63	64						
	7	1	8	29	64	99	120	127	128					
	8	1	9	37	93	163	219	247	255	256				
	9	1	10	46	130	256	382	466	502	511	512			
	10	1	11	56	176	386	638	848	968	1013	1023	1024		
	11	1	12	67	232	562	1024	1486	1816	1981	2036	2047	2048	
	12	1	13	79	299	794	1586	2510	3302	3797	4017	4083	4095	4096

Table 2: Values of $\Phi_\Delta(D)$ for $0 \leq D \leq \Delta \leq 12$

List of abbreviations and acronyms

ALG	The value of the objective function returned by some approximation algorithm for some particular instance of a combinatorial optimization problem
BFS	Breadth-First Search
DDP	The Degree-Diameter Problem
DFS	Depth-First Search
gcd	Greatest common divisor
iff	if, and only if
lcm	Least common multiple
HSAGA	Hybrid Simulated Annealing and Genetic Algorithm
MaxDDBS	The Maximum Degree & Diameter Bounded Subgraph Problem
MaxWDDBS	The Maximum Weighted Degree & Diameter Bounded Subgraph Problem
OPT	Optimal value of the objective function for some instance of a combinatorial optimization problem
PTAS	Polynomial Time Approximation Scheme
s.t.	such that
TSP	Travelling Salesman Problem
w.l.o.g.	without loss of generality

Index