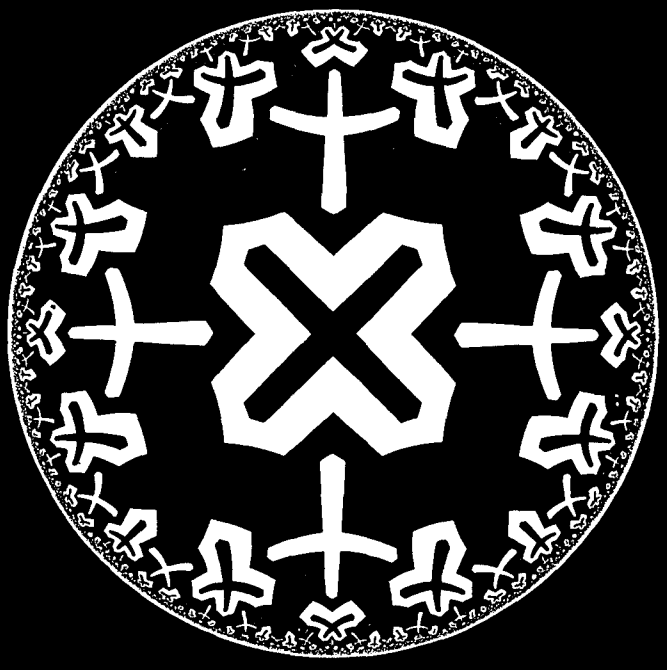
Efficient Network Topologies for Extensible Massively Parallel Computers



Ferdinand Peper

10. Stellingen bij een proefschrift dienen niet te genuanceerd te zijn: nuances worden vanzelf aangebracht in de discussie over de stellingen.

11. Uit veiligheidsoverwegingen gaat deze stelling niet over de Islam.

Ferdinand Peper
Augustus 1989

Stellingen

behorende bij het proefschrift

Efficient Network Topologies
for
Extensible
Massively Parallel Computers

1. Beschouw het 'vuurtorenalgoritme', beschreven in [1], dat de verbonden componenten van een eindige ongerichte graaf G met behulp van $n=|V(G)|$ processoren in $2n-2-\deg(G)$ stappen kan bepalen op een shared memory SIMD-machine. Zij K_m ($m \in \mathbb{N}$) de klasse van grafen waarvan voor elk element G geldt:

   Als $C_1,\ldots,C_q$ de verbonden componenten van G zijn, dan is $\mathrm{diam}(C_i) \times \deg(C_i) \leq m$ voor $i=1,\ldots,q$.

   Er geldt dan dat voor elke $G \in K_m$ de verbonden componenten van G in m stappen bepaald kunnen worden door het vuurtorenalgoritme op een shared memory SIMD-machine met $|V(G)|$ processoren.

   [1] F. Peper,
   "Determining connected components in linear time by a linear number of processors", Information Processing Letters, Vol. 25, July 1987, pp. 401-406.

2. Zij Γ de onderliggende graaf van een (d,h)-net met een d-cube als bouwsteen (zie [2] en hoofdstuk 1), dan geldt:

   $$\exp(\Gamma) \geq 2^{(d-2)/(d+1)}.$$

   [2] K. Hwang, J. Ghosh,
   "Hypernet: A Communication-Efficient Architecture for Constructing Massively Parallel Computers", IEEE Transactions on Computers, Vol. C-36, No. 12, December 1987, pp. 1450-1466.

3. Voor de uniforme exponentialiteiten van supersymmetrische grafen geldt:
   - $\overline{\exp}(S_{6,k}) = \overline{\exp}(S_{4,k+1}) = \overline{\exp}(S_{3,k+3})$,
   - $\lim_{k\to\infty} \dfrac{\overline{\exp}(S_{3,k})}{k-4} = 1$,
   - $\lim_{k\to\infty} \dfrac{\overline{\exp}(S_{4,k})}{k-2} = 1$,
   - $\lim_{k\to\infty} \dfrac{\overline{\exp}(S_{g,k})}{k-1} = 1$ voor $g \geq 6$.

4. Beschouw het polynoom $f_{gk}(t)$, beschreven in theorema 7.41. Er geldt:
   - Als t_0 een nulpunt is van $f_{gk}(t)$, dan is $1/t_0$ ook een nulpunt van $f_{gk}(t)$,
   - De grootste reële (positieve) wortel $\hat{t}$ van $f_{gk}(t)$ is enkelvoudig, en is de enige wortel van $f_{gk}(t)$ in $\mathbb{C}$ met radius $\hat{t}$,
   - De volgende getallen zijn wortels van $f_{gk}(t)$:

     1 als $(g,k)=(6,3)$ of $(g,k)=(4,4)$ of $(g,k)=(3,6)$,

     -1 als $g \equiv 0 \pmod 4$ en $k=4$, of als $g \equiv 2 \pmod 4$,

     i en $-i$ als $g \equiv 3 \pmod 8$ en $(g,k) \neq (3,6)$.

5. Het aantal door Nederlandse wetenschappers geproduceerde artikelen zou groter zijn indien al in het voortgezet onderwijs meer aandacht zou worden besteed aan schrijfvaardigheid, en in het bijzonder aan het opzetten van een coherent betoog.

6. Het van te voren nauwkeurig specificeren van te behalen resultaten in een wetenschappelijk project kan belemmerend werken.

7. Het kabaal waarmee paranormale genezers en andere kwakzalvers hun gelijk opeisen wanneer een deel van hun kunsten een wetenschappelijke fundering dreigt te krijgen bewijst nu juist hun onwetenschappelijkheid.

8. Een grootscheeps onderzoek onder PC-bezitters moet welhaast tot de conclusie leiden dat personal computers voornamelijk gebruikt worden voor tekstverwerking en het verbeteren van schietvaardigheid.

9. Ten onrechte wordt vaak gemeend dat de kwaliteit van een geluidsinstallatie slechts door de componenten wordt bepaald en niet door de verbindingen tussen de componenten.

Efficient Network Topologies for Extensible Massively Parallel Computers

Efficient Network Topologies for Extensible Massively Parallel Computers

---

Efficiënte Netwerk Topologieën
voor
Uitbreidbare
Grootschalig Parallelle Computers

Proefschrift

Ter verkrijging van de graad van doctor
aan de Technische Universiteit Delft
op gezag van de Rector Magnificus,
Prof. drs. P.A. Schenck,
in het openbaar te verdedigen
ten overstaan van een commissie
aangewezen door het College van Dekanen
op maandag 18 september 1989 te 16.00 uur

door

Ferdinand Peper

wiskundig ingenieur
geboren te Hengelo(O)

Dit proefschrift is goedgekeurd door de promotor:

Prof. dr. S.C. van Westrhenen

en door de leden van de promotiecommissie:

Prof. dr. ir. L. Dekker
Prof. dr. ir. A.J. van de Goor
Prof. dr. I.S. Herschberg
Prof. dr. J. van Leeuwen
Prof. dr. D.J. McConalogue
Ir. R. Sommerhalder

A man must have a certain amount of intelligent ignorance to get anywhere.

Charles F. Kettering (1946)

Picture on cover: Circle limit II, M.C. Escher, March 1959,
see also paragraph 6.2 (figure 6.4),
Collection "Haags Gemeentemuseum", the Hague,

Acknowledgements

Many people have helped me during the preparation of this dissertation in the last few years. First of all, I would like to thank my promoter Prof. S.C. van Westrhenen for the helpful discussions and comments and for the many hours he has spent reading my manuscripts. Furthermore, I would like to thank Ruud Sommerhalder for the helpful discussions and his comments. Thanks go also to the following persons:

Robbert Fokking for proving lemma 7.39 at the last moment,
Hans Tonino for the discussions we had about the proof of theorem 8.4,
Haixiang Lin for the many discussions we had about implementations of ray tracing algorithms on the tree-mesh,
Prof. H.J.A. Duparc for providing the idea for the proof of theorem 7.47,
Prof. J. van Leeuwen of the University of Utrecht for pointing me to references [Parber] and [Hilber],
M. Veldhorst of the University of Utrecht for pointing me to reference [Vitany],
Prof. A.W. Grootendorst for providing references to the theorems of Descartes (7.43) and Fourier-Budan (7.44),
S.J. van Strien for pointing me to the Perron-Frobenius theorems concerning eigenvalues of non-negative matrices,
Gert Jan van Loo for the many discussions we had about parallel computing and communication networks,
Prof. D.J. McConalogue for his advices about writing in English,
Ton Biegstraaten for keeping the system running and for his advises concerning the use of troff,
Koos Schonewille for his support concerning copying this dissertation and for drawing figure 4.1,
J.W. Muilman and W.J.P. van Nimwegen of the Electrical Engineering's Drafting Department for furnishing all other figures, which were quite a lot,

Acknowledgements

Trudie Stoute for her advices about English and for typing half of the many letters I had to send during the preparation of this dissertation,
Lieneke van der Kwaak for typing the other half of the letters,
Mrs. and Mr. Veldhuysen of Cordon Art for giving permission to use Escher's Circle limit II for the cover,
My colleagues in the section theoretical computer science for the pleasant environment, and especially my "room-mate" Cees Witteveen for the interchange of many jokes and ironical remarks,
And last but not least, my parents, family, and friends for their support and encouragement during the past years.

Contents

Introduction

Is it progress if a cannibal uses knife and fork?

Stanislaw Lec (1962)

Developments in integrated circuit technology have resulted in chips with large numbers of electronic components. Up to now the number of components that can fit onto a single chip has approximately been doubled every two years. Current top of the line chips contain about 10^6 gate equivalents, and this number is expected to grow to 10^7 or even 10^8. It is making feasible processors ever-decreasing in size and ever-increasing in speed.

Advances in technology not only cause but also *limit* performance growth of computer systems. Attempts to hide from this have resulted in costly high-performance sequential computers, such as the Cray I, the Cyber 205, etc. Up to now there was a wide and successful application of such pipelined vector computers, especially in a field like physical modeling. Nevertheless, limits to performance improvements of such computers have been reached. Yet to obtain faster computers, parallelism is employed. Up to now, this resulted in the Cray XMP, Cray II, Alliant, etc.

Parallelism in these computers is, however, only applied on a small scale, since their processors are very expensive. Even for these computers there remain problems which cannot be solved in reasonable time. In fields such as speech analysis, image processing, computer vision, machine inference, weather modeling, seismic exploration, and nuclear fusion research, performance increase of a factor 1000 is not an excessive luxury. Computation intensive jobs often consist of many highly parallel subtasks. It is a reason for existence of parallel computers equipped with very many cheap processors.

This introduces the first of three themes in this dissertation, i.e. *massive parallelism.* Computers equipped with large ensembles of processors are called massively parallel. They consist of at least 100 to 1000, in future possibly 10^6 to 10^8, processors. Major increases in performance can only be achieved by massively parallel computers. Ever-decreasing sizes and prices of processors make such computers highly cost-effective - although their huge number of processors will always keep them expensive.

The second theme in this dissertation is *interprocessor communication.* Interprocessor communication is one of the crucial issues in parallel computers. In order to communicate efficiently, a computer should have a communication mechanism being able to transport messages fast, and being able to handle a large number of messages simultaneously.
A special class of communication mechanisms are networks. Interprocessor communication by networks has been subject to many studies in literature (see paragraph 1.4.4). In this dissertation we are interested in a special kind of networks, i.e. statical point-to-point topologies. In such structures all processors have a local memory and they communicate via immutable interprocessor connections.

Many computer systems run out of capacity a few years after purchase. A reason for this is strikingly reflected by the first sentence of [AgJaPa] which says: 'As ever more powerful computers were developed, so did the demands made upon them'.
A company confronted with lack of capacity as years pass by, will not be very eager to buy a new computer, mainly for two reasons. First of all, costs of such an operation are high. This is even more true for the replacement of an expensive massively parallel computer. Second, there is no guarantee that the new computer is completely software-compatible with the old one.
At this point the third theme of this dissertation comes in view, i.e. *extensibility.* A parallel computer is extensible if it can be extended by addition of processors, while its structure and characteristics are maintained. The structure of a computer is maintained when extension does not essentially change its architecture. It guarantees software compatibility. By maintenance of characteristics we mean that a machine's characteristics, such as the worst-case communication time, do not change drastically upon extension.
Clearly, an extensible computer system is a solution for the company mentioned above. Extensibility is also an attractive prospect to, for example, (parts of) companies which expect to grow, or technical customers who use a small-sized configuration for development of software and add processors when the system is put into use.

This dissertation deals with the following two questions.

1. Which problems can be expected in the design of efficient extensible massively parallel computers, and how can these problems be solved?
2. How can efficient statical point-to-point topologies be designed for extensible massively parallel computers?

Chapter 1 gives an introduction to massive parallelism, interprocessor communi-

cation and extensibility. Chapter 2 describes a method to construct efficient point-to-point topologies for extensible parallel computers, and describes measures to evaluate such networks. In chapter 3 we face some unpleasant physical consequences of low communication times in extensible massively parallel computers. It appears that low communication times in extensible computers give rise to space deficiencies. We meet this physical dilemma by proposing a teasing solution, which amounts to designing, building and putting on sale of computers, while most of their chips still have to be designed and cannot even be manufactured by the state of technology at that moment.
Thereupon, the construction method of chapter 2 is used to construct two infinite classes of extensible networks. The first class, which is dealt with in chapters 4 and 5, consists of networks which are very likely suitable for efficient implementation of algorithms. Chapters 6, 7, and 8 describe the second class. This class contains networks which are planar and which enable low nearly optimal worst-case communication times.
Notions and notations used in this dissertation can be found in appendix A.

PART I

General

1

Massive parallelism, communication issues, and extensibility

Good communication is stimulating as black coffee,
and just as hard to sleep after.

Anne Morrow Lindbergh (1955)

1.1 Introduction

This chapter deals in more detail with the three main themes in this dissertation: massive parallelism, communication, and extensibility.
Paragraph 1.2 sketches how massively parallel computers should be used in order to be an attractive alternative to vector computers.
Paragraph 1.3 deals with the processors in an extensible massively parallel computer. It discusses whether the processes should be MIMD or SIMD, how powerful the processors should be, whether they should have their own local memory, whether they should be identical, and whether each processor should have its own clock.
Paragraph 1.4 deals with communication between processors. It states some demands on communication mechanisms, discusses the impact of physics on communication, gives a brief overview of communication mechanisms, and discusses statical point-to-point topologies.
Paragraph 1.5 deals with extensibility. It starts to explain why computers should be extensible, relates extensibility to graph theory, and states some demands on extensibility. Thereupon, it discusses some limitations to performance of extensible computers, describes in what configurations an extensible computer should be used, and gives a short comment on extensibility.
Paragraph 1.6 describes some important factors affecting performance of a parallel computer. In particular, it discusses the scheduling and mapping of jobs on extensible computers based on statical interconnection networks.

1.2 Massive parallelism as an alternative to pipelined vector computers

In the last 10 years massive parallelism has been subject of extensive research. Some prototype massively parallel computers have been built, such as the Massively Parallel Processor ([Potter]), the Connection Machine ([Hillis]), the ICL DAP ([HocJes]), the New York University's Ultracomputer ([GoGrKr]), the CHiP ([Snyder]), the Non-Von ([Shaw]), the TRAC ([LipMal]), and the GF11 ([BeDeWe]). There was and is a small market for massively parallel computers. Nevertheless, they are still waiting for a commercial breakthrough. Massively parallel computers are mainly applied in academical environments. Their success is overshadowed by that of other supercomputers, in particular pipelined vector computers. In order to have a right to exist, massively parallel computers should offer at least the same as vector computers, preferably against lower costs.
Vector computers have a high peak-performance, but also a high price. Vector computers are often employed as a multi-user system. A large number of users guarantees constant supply of jobs, preventing the computer system from being idle for a while. It enables a continuous utilization of the processing capacity. As a result, vector computers have a high throughput of jobs - that is the number of jobs processed per unit of time. Disregarding the overhead of the multi-user environment, the cost-effectiveness of vector computers is high.
Massively parallel computers not only have their high performance, but also their high price in common with vector computers. To be competitive with vector computers, massively parallel computers should have a high throughput of jobs too. Furthermore, they should be employable as multi-user systems.

In order to achieve a high throughput, it is necessary to have some knowledge about the load the jobs impose to a system. In general, there is much variety to the load. The well-known 20/80-rule appears to be valid, i.e. 20 percent of the jobs use 80 percent of the resources. For jobs processed on a parallel supercomputer, the amount of parallelism they exhibit is of importance. There are jobs exhibiting few parallelism, but there are also jobs occupying a large part of the computer. Furthermore, the amount of parallelism in a job may be time dependent. Assuming the universal validity of the 20/80-rule, we conclude that 20 percent of the jobs use 80 percent of a parallel computer's resources (processors) in a concurrent way.
Jobs exhibiting only a small amount of parallelism, are just able to use only a small part of the processors concurrently. However, even for jobs exhibiting much parallelism it is sometimes not very meaningful to use the maximal number of processors concurrently. It would drastically degrade the efficiency (see appendix A) by which the processors are used. Consider Amdahl's law. If f_s is the fraction of sequential instructions executed in a computation on p processors, then the speedup S_p (see appendix A) is bounded from above according to the

formula

$$S_p \leq \frac{1}{f_s + (1-f_s)/p}.$$

It states that speedup of a parallel program is mainly determined by the part of the instructions, which are not parallelizable. If the speedup to be attained is at least p/2, then it is easily deduced (see [Quinn]) that

$$p \leq 1 + 1/f_s.$$

Consequently, in order to achieve a reasonable efficiency ($\geq$½), the number of processors should be limited if a part of the instructions is to be executed sequentially.
So, the claims that jobs make to processors in a concurrent way are due to much variety. They depend of the amount of parallelism exhibited in the job as well as the fraction of non-parallelizable instructions.

In order to cope with the variety in the load jobs impose to a massively parallel computer, the computer should be able to execute several jobs concurrently. A job exhibiting few parallelism will claim only a small part of the processors. If it is impossible to execute such a job concurrently with other jobs, it will block the whole computer. This reduces the efficiency of a massively parallel computer substantially.

We conclude that massively parallel computers with a high peak-performance are competitive to pipelined vector computers, if, employed as multi-user system, they are able to execute several jobs concurrently. In the remaining paragraphs of this chapter massively parallel computers are assumed to have these characteristics. The requirement that a massively parallel computer should be able to execute jobs concurrently, will have impact on the architecture of the computer.

1.3 The processors

An important issue in the design of parallel computers concerns the processors. Major decisions to be made in the design are:

1. Should the processors be fed by one single instruction stream or by several streams? (SIMD versus MIMD)
2. Should the processors be powerful, or should their power be traded against their number?
3. Should each processor have its own local memory, or should a global memory be used?

And if an MIMD-concept is adopted:

4. Should the processors be identical?
5. Should the processors work under a single global clock, or should each of them have its own clock?

In the previous subparagraph we concluded that a general-purpose massively parallel computer should process several jobs concurrently, in order to be efficiently applicable. Running several jobs concurrently on one computer can only be achieved when all processors are able to run different programs. So, it is plausible to choose for an MIMD concept.

In all parallel systems designed up to now the power of processors range from simple 1-bit processors as in the ICL-DAP to highly pipelined vectorized processors as in the CRAY X-MP, the CRAY 2, and the CRAY Y-MP. Quinn ([Quinn]) denotes these approaches by the army-of-ants and the herd-of-elephants approach respectively. Both approaches have advantages and disadvantages.
Application of simple processors results in a more efficient use of the hardware. Compared with an advanced processor, a simple processor consists of very few transistors. The number of processors which can be built up from a fixed amount of transistors is maximized for simple processors. Furthermore, most of the transistors in an advanced processor are idle while the processor is running. So, the efficiency of the use of a fixed number of transistors is maximal if they are implemented as simple rather than advanced processors.
On the other hand, it gives much trouble to keep all processors running, especially when the program to be run contains large pieces of non-parallelizable code. Amdahl's law implies that powerful processors are more lucrative in that case. So, although transistors are more efficiently used in the army-of-ants approach, keeping all processors running causes so much trouble that the herd-of-elephant approach is preferable.

Which approach should be preferred for massively parallel computers?

Application of very powerful processors in massively parallel computers is very costly. On the other hand, simple processors degrade the peak-performance. Therefore, we propose an option somewhere between these two extremes. The processors in a massively parallel computer should at most be as powerful as a processor fitting on a single chip, but at least be as powerful so as to be able to run with a reasonable speed. This definition is rather vague, but for a general discussion of these matters more specificness can not be expected.

Whether each processor should have its own local memory depends on the decisions made for the communication medium between the processors. This subject will be dealt with in more detail in subparagraph 1.4.4.

The choice for MIMD leaves open the question whether all processors must be identical. Before giving an answer to this we make a distinction between logical and physical identity.
Two processors are *logically identical* if they behave identical. That is, execution of the same program with identical inputs produces identical outputs. Several levels of logical identity can be distinguished. High level identity occurs when the program is written in some high-level language, and input/output are strings of characters without concern how these characters are represented in bits. As a matter of fact, many processors are logically identical at this level. On the other hand, two processors exhibit low level identity whenever they are identical at bit level, i.e. when they are pin-compatible. Low level identity is more strict than high level identity. We shall only qualify two processors as logical identical whenever they don't differ at bit level.
Two processors are *physically identical* whenever they are logically identical, have the same layout in VLSI and are equally sized.

From many points of view, it is beneficial for a parallel computer to have logically identical processors. It simplifies the computer's structure. Logically identical processors are able to take over tasks of each other, easing the consequences of defective processors. It also enables a flexible schedule of jobs onto processors. For, in a computer with logically identical processors scheduling software does not have to take into account processor inhomogeneities. So, logical identity results in simplicity of (system) software.
Another benefit of logical identity is that it simplifies the analyses of the characteristics and performance of a parallel computer.
Although parallel computers with non-identical processors don't have the advantages of identity, they are sometimes preferred to homogeneous computers. Addition of special-purpose processors for database operations, matrix operations or other specific operations often results in a significant improvement of a parallel computer's total performance. However, it might have a negative influence on a computer's efficiency for two reasons.
First, although the high speed of special-purpose processors results in performance improvements, it might be overshadowed by the time needed for data-transport to and from the processors.
Second, if special-purpose processors are not frequently used, then the yield of the investment in them is low. In that case, it is more cost-effective to spend the money to continuously used general-purpose processors (see also [LipMal; pp.11, 29, 30]).

In addition to all advantages of logical identity, physical identity offers even more. It implicates the identity of many electronic components, thus resulting in larger series and lower prices. Furthermore, a limited number of different kinds

of electronic components in a computer implies that only a small number of components has to be designed. As a result of this, the design costs will be lower, and they can be spread among larger series.

In this dissertation we shall only consider parallel computers consisting of logically identical processors, without willing to qualify computers with logically different processors as useless. Whether processors should be physically identical will be discussed in chapter 3.

Finally, we are confronted with the question whether all processors should work under a global clock or not. Processors working under local clocks are sometimes said to work asynchronously. This kind of asynchronism differs from the asynchronism denoting that a computer is a member of the class MIMD. By definition, programs are executed asynchronously by MIMD processors, even if they are controlled by a global clock. Consequently, two levels of synchronism can be distinguished. To differentiate between them we decided to use the terms global and local clock instead of synchronous and asynchronous.

In practice, global clock designs are preferred because they require less complicated hardware. Data exchange between two directly connected processors is complicated when both have their own local clocks. As a consequence of local clock designs, hardware provisions have to be made for synchronization of data transfer.
On the other hand, when systems become physically large, or when their size can not be predicted in advance, the advantages of local clock designs begin to mount. If a system's physical size is large, application of a global clock runs up against difficulties. There is the delay of a signal through a wire, caused by the wire's resistance and capacity. In a computer with many processors, transportation of a clock pulse to a processor lying far away takes a longer time than transport to a near processor. As a consequence, the processors don's run in lock step. To cope with this, the clock frequency can be decreased until the situation arises that no processor receives a new pulse, while others haven't yet received the previous one. This causes a degradation of the computer's performance.
Alternatively, a so-called 'clock-tree' may be used, which is able to transport a signal from the root to its leaves. If all wires in the tree have equal length, this device guarantees simultaneous arrivals of all the pulses in its leaves. When a provision is made for pipelining the pulses among the tree, the system doesn't suffer from long delays, enabling a higher clock pulse.

However, not only clock pulses but also data transmission between processors is hindered by large physical distances. In global clock designs, the time between two consecutive pulses should be large enough to enable a package of data to flow from a source processor to its destination. Clearly, large physical distances force

the clock frequency to be low. We conclude that extensibility and massiveness causes complications in globally clocked computers (see also [FisKun] and [Wan-Fra]). For this reason and for reasons which will become clear in chapter 3 we prefer locally clocked schemes.

1.4 Communication issues

1.4.1 Introduction

One of the crucial issues in parallel computer design is interprocessor communication. The design of a parallel computer stands or falls with the efficiency of its communication. In the previous decade, interprocessor communication has been the subject of much research. This paragraph will not give an exhaustive overview of this research. We shall rather describe the conditions that communication mechanisms should satisfy in order to be fully fledged (see subparagraph 1.4.2). In addition, we shall describe some measurements on the basis of which strategies for communication can be evaluated.
For communication models, and in particular the efficient ones among them, a number of restrictions imposed by nature's laws hold. The consequences of these restrictions will be described in subparagraph 1.4.3.
In the search for ways to communicate many proposals have been done. Subparagraph 1.4.4 will give an overview of well-known communication mechanisms.
Thereupon, in subparagraph 1.4.5 one of the communication mechanisms, statical networks, will be considered in more detail.

1.4.2 Demands on communication mechanisms

To be efficiently applicable in practice, communication mechanisms should satisfy a number of conditions. Before dealing with these conditions in this subparagraph, we shall describe two measurements enabling evaluation of communication mechanisms.
The first metric is the *worst-case communication time.* It is the longest time needed to send a single message in isolation of a processor to another, considered over any two processors in a computer. During the mailing of the message no other communication in the computer takes place.
The second metric is the so-called *communication bandwidth* (e.g. see [Levita]). It is the total number of messages that can be sent or received by the processors in the system in one unit of time (O(1)). The time needed for one CPU operation or instruction is considered as unit of time. In a computer with n processors the communication bandwidth is never greater than n.

Having defined these two metrics, we can deal with the demands on communication mechanisms. Two groups of demands are distinguished.

- Preconditions to communication.
- Preconditions to efficiency of communication.

Essential demands in the first group are

1. The communication mechanism should be capable to establish communication between any two processors (trivial).
2. The communication method should not conflict with physical laws.
   This trivial requirement appears to have more repercussions than would be expected on the face of it. More details about this can be found in subparagraph 1.4.3 and chapter 3.
3. The communication mechanism must technologically be feasible.
   As an example, a totally connected network with a large number of processors is infeasible with current technology. (In such a network any two processors are directly connected, resulting in $n-1$ connections per processor, where n is the number of processors.) The number of I/O-ports needed in each processor is very large in totally connected networks.
   In the near future a totally connected network with 1000 processors seems to be technologically feasible: In [DeFrSm] a parallel computer with 1000 processors is proposed in which each processor has direct write-access to an output buffer belonging to it and each processor has direct read-access to 1000 input buffers, each of which corresponds uniquely to a processor. The total number of input buffers is 10^6 and the contents of each output buffer can be transported via optical links to 1000 of the input buffers.

Preconditions to efficiency of communication are

4. The amount of communication hardware should be kept small, and the hardware should not be too complex.
   Since most of the hardware of a parallel computer is needed for communication, any attempt to reduce it is welcome. This demand does not necessarily result in faster communication, but it does enable the spared components to be used for other purposes, such as extra computational hardware. Furthermore, simplicity of communication hardware results in lower design costs.
5. The communication mechanism should be applicable in massively parallel computers.
   Communication mechanisms which can only be used efficiently when the

number of processors is limited, or which require extensive hardware provisions for large numbers of processors, can not be applied in massively parallel computers.

6. Communication software should be simple.
   Simplicity of the software dealing with the sending of messages in a computer will result in lower software design costs, and will often result in faster execution of the software.

7. The structure of the communication mechanism should be regular.
   This demand does not necessarily result in faster communication, but it does result in a more well-organized communication process.
   It promotes simplicity of the software running on the computer, in particular the system-software dealing with the sending of messages. For, the software does not have to take into account inhomogeneities in the communication structure if the latter is regular.
   A regular communication structure will also relieve the job of someone analyzing the performance of the computer. In general, absence of inhomogeneities results in a more orderly course of the data streams in a computer, simplifying the forecast of the behaviour of these streams and the analysis of the load they impose to the system.
   Finally, regularity of a communication mechanism results in a simpler layout of the computer. This is not only an advantage for a user, a designer will also profit from this. The ever-repeated patterns in a regular structure need to be designed only once, resulting in lower design-costs.

8. The worst-case communication time should be low.
   Too high a worst-case communication time slows down the execution of jobs and causes processors to wait long for messages. The best worst-case communication time that can be achieved by most practicable communication mechanisms is $O(\log n)$, where n is the total number of processors.

9. The communication-bandwidth should be large.
   One important pursuit in the design of communication mechanisms is to maximize the number of processors capable to communicate simultaneously. Too small a bandwidth may cause congestion of the communication mechanism. As a consequence of congestion a system's performance will decrease.

10. The processes in a job should be scheduled on the processors such that interprocessor communication is minimized.
    Direct processor to processor communication in an asynchronous MIMD computer causes serious problems ([Uhr; p. 51]). The operating system

running on such a computer has to execute many thousand instructions to send a message between processors. It takes thousands of times longer to send information than to operate on that information. Hence, minimization of interprocessor communication grades up the efficiency with which jobs are executed.

Since it is hard to forecast communication between processors within one job, the above demand is not easily satisfied. Therefore, we propose the following requirement as alternative.

11. The processes in a job should be scheduled on the processors such that computation and interprocessor communication are well-balanced.
    An excess of processors used for a job will tip the scale to considerable interprocessor communication times. On the other hand, using too small a number of processors results in a more efficient use of them, but also in a lower speedup. An intermediate course between these alternatives results in a reasonable speedup and a reasonable efficiency of the use of the processors. Setting the number of processors working on a job such that the efficiency of their use is, say, ½ seems to be reasonable.

12. Communication within a job should not be interfered by communication in other jobs.
    The routes via which processes in a job send their messages should not pass via processors executing another job. There is no mutual communication between jobs. The only communication taking place should be the communication between processes belonging to the same job. Dependency of jobs with respect to communication decreases the speed and efficiency of their execution. It causes the communication times in a job to be not only dependent of the job itself, but also of external factors.

The first class of demands should be satisfied unconditionally. The second class is more interesting: it is a challenge to design a model for communication satisfying these demands as close as possible.

1.4.3 Physical aspects of communication

In early computers a large part of the costs was constituted by components rather than wires. Because of the ever-advancing VLSI technology, this situation has drastically changed. Nowadays the fundamental limitation to computers are the high costs of communication relative to logic and storage. These costs are mainly encountered in (see [Seitz], [Hilber]):

- Consumption of chip area.
  The layers of a chip are primarily covered by wires, whereas at most 5 percent

of its area is swallowed up by transistors on the lowest layer.

- Power consumption.
  The energy supplied to a chip is mainly absorbed by the circuits that switch signal nodes. It is almost entirely used to charge the capacitance of internal and interchip wires, rather than the capacitance of transistor gates.

- Time delays, both on chips as between chips.
  First, the delay of a signal when transmitted through a wire is high compared to the switching time of a modern transistor. The delay is due to the poor relation between transistor driving capabilities and the high parasitic capacitance of a wire.
  Second, delay of a wire caused by resistance and parasitic capacitance of wires, is becoming increasingly significant at smaller geometries. The delay depends of the ratio between a wire's length and its width. The delay of a short wire is logarithmic in its length. Beyond a certain length, dependent of the wire's width, the delay time grows linearly with the length of the wire.
  Third, amplification of an on-chip signal to off-chip levels causes a delay comparable to a clock period. Amplification is necessary to bridge the differences between internal and external signal energies. External signal energies are higher than internal energies, to cope with the large capacitances of package pins and interchip wiring.

The above discussion implies that communication constitutes the main part of the costs of VLSI chips. We conclude that communication costs favour architectures in which communication is minimized or at least localized.

Physics have their impact to still other communication issues. In the previous subparagraph, a low worst-case communication time - $O(\log n)$, where n is number of processors - is considered important. To realize models, exhibiting such performance on communication time, some physical aspects have to be taken into account. Nature's laws appear to impose severe restriction on such models.
In the discussion in this subparagraph we will assume that the physical space required by a processor is constant, and that the time to transmit a signal through a wire scales linearly to the wire's length.

In [Vitany] Vitányi discusses the physical realizability of communication models with logarithmic worst-case communication times. He concludes that under the constant space assumption for processors such models are infeasible, modulo major advances in physics, when a system contains a large number of processors. This is easily seen. If a computer consists of n processors of unit size each, then the tightest they can be packed is in a 3-dimensional sphere of volume n. If this

ball has radius R, then

$$n = 4/3\pi R^3.$$

The maximum physical distance between two processors in the ball is

$$2.R = O(n^{1/3}).$$

Since the worst-case communication time is linearly proportional to this distance, it is of the same order. Hence, communication models with communication times less than asymptotically $O(n^{1/3})$ time and assuming O(1) space per processor are infeasible. If the computer is embedded in only 2 dimensions, the situation is even worse. Worst-case communication times of $O(n^{1/2})$ are the optimum which can be attained in that case.
Communication mechanisms with communication times less than $O(n^{1/3})$ (respectively $O(n^{1/2})$) suffer from space deficiencies, if they are applied in massively parallel computers assuming constant space per processor. It is emphasized that this result holds for any communication mechanism. Results equivalent to those in [Vitany] but less general can be found in [Mazumd] and [Fisher]. We refer to chapter 3 for more details about space deficiencies.

Quite a different physical limitation to communication mechanisms is the limitation to the rate at which processors can send messages to other processors in a uniform or purely random pattern. This limitation follows from considerations which are well-known in VLSI complexity theory (see [HarUll]). Again, we assume a parallel computer with n processors of O(1) physical size each and an arbitrary communication mechanism.
Suppose that the processors are tightly packed in d-dimensional Euclidean space (d=2 or 3), and constitute a parallel computing system with physical size $S_d(n) = O(n)$. If we assume that the space occupied by the system is convex (which is a reasonable assumption, because the packing of the processors must be tight), then the space taken up by the d-dimensional circumscribed rectangle is also O(n).
Let's consider the hyperplanes - i.e. (d−1)-dimensional subspaces of a d-dimensional space - dividing this rectangle in two parts, each part containing approximately half of the processors. Of all these hyperplanes, the ones perpendicular to a longest side of the rectangle have the smallest 'space' (i.e. 'area' for d=3 and 'line segment' for d=2) in common with the rectangle. The size of this space is at most $O(n^{(d-1)/d})$ - compare: shortest side of 2-dimensional rectangle with area O(n) is at most $O(n^{1/2})$. Therefore, at most that order of communication lines can cross this space. Since each wire is capable to send only a limited number of messages per unit of time, the rate at which messages cross the space is at most $O(n^{(d-1)/d})$.

Suppose, communication between processors in the system is determined by a purely random or uniform pattern. This pattern depends of course of the programs running on the computer, and the matching of programs onto the computer. If an efficient matching between programs and processors is possible, communication may be very locally bounded. In that case the above supposition does not hold. In many other cases, however, an efficient matching is impossible. In particular, in AI programs data streams exhibit a somewhat random behaviour. So, the assumption of random or uniform patterns is not purely unrealistic.
In the communication-pattern supposed, about n/4 processors in each half of the computer will send a message to the other half through the hyperplane. This results in approximately $n/2 = \Omega(n)$ messages wishing to cross the hyperplane per unit of time. However, only $O(n^{(d-1)/d})$ are allowed to pass per unit of time, causing a communication bottleneck.
The inconvenience experienced from this bottleneck strongly depends of the communication pattern. It is a ground to aim at schedulings of jobs on a computer so that local communication patterns will arise. For other reasons, i.e. to minimize communication times experienced by a job, there already was wide agreement of the necessity of local communication patterns (see also demand no. 10 in subparagraph 1.4.2).

1.4.4 Brief overview of communication mechanisms

This subparagraph gives a short overview of the main communication mechanisms. We shall not give a complete survey of these models, but rather intend to supply the unexperienced reader with a global impression and the main references to the literature.

Communication mechanisms can roughly be divided into three classes:

- Shared memories.
- Shared busses.
- Networks.

Combinations of them can often be found in parallel computers.

In a shared memory model processors communicate via a shared random access memory, writable and readable for all of them. In its simplest construction at most one processor at a time is allowed to refer to it. Trivially, the shared memory is a flagrant bottleneck in that case. For this reason, alternative shared memory models have been introduced, which permits simultaneous reference of many processors to it.

These models exhibit larger communication bandwidths. Nevertheless, conflicts will arise if several processors simultaneously refer to one memory location. The most tedious conflicts are those in which several processors simultaneously try to write to one memory location. We call them *write-conflicts.*
The situation in which one processor writes to a certain memory location and others simultaneously read from it is denoted by the term *read-/write-conflicts.* As a consequence of read-/write-conflicts read values are not defined uniquely: they may be equal to the written value, but they may also be equal to the contents of the memory location before it was written to.
Finally, *read-conflicts* occur when several processors read from one location. There are no principal drawbacks of such conflicts: permitting them in theoretical models causes no disasters. Practice is different, however. Memories permitting read-conflicts in their locations are not easily designed and are expensive.

The forementioned conflicts have resulted in a multitude of shared memory SIMD models ([Quinn]) each permitting a particular combination of conflicts. We mention the SIMDAG ([Goldsc]), P-RAM, PP-RAM, SP-RAM ([ShiVi2]), RP-RAM ([ShiVi1]), CRCW P-RAM ([Quinn]), CREW P-RAM ([Quinn]), and the EREW P-RAM. No one of these models have actually been realized as computer.
The only practicably applicable model permits no read- and write-conflicts. To prevent bottlenecks in such models shared memory is divided in memory banks. Memory banks are memories which can independently be accessed by all processors, each bank permitting one access at a time. So, division of a shared memory in m banks enables at most m simultaneous accesses to the memory.

How should the processors be connected to the memory banks?

The processors are connected by a communication mechanism which sends access instructions to the proper memory bank and returns eventual data to the proper processor. There are roughly two possibilities for such a mechanism. It can be a bus or it can be a network. These are the two alternatives to the shared memory model. We conclude that shared memory models should be combined by busses or networks in order to be applicable.

Are there examples of shared memory computers which have actually been built?

Yes, there are. Most pipelined vector computers (though they are strictly speaking not parallel) are equipped with memories divided in banks and connected by busses to the processor(s).
In tightly coupled multiprocessors - these are MIMD computers with a shared memory and shared memory address space - busses as well as networks may occur. Encore's Multimax ([Quinn]), and Sequent's Balance 8000 ([Quinn])

consist of processors connected by a bus to a shared memory divided in banks. Examples of shared memory multiprocessors using networks are Carnegie-Mellon's C.mmp ([HwaBri]), Denelcor's Hep ([Kowali]) and New York University's Ultracomputer ([GoGrKr]).

Quite a different class of parallel computers are the so-called loosely coupled multiprocessors. As in tightly coupled multiprocessors, they have a shared address space. In this architecture, however, the memory banks are directly accommodated with the processors. Each processor is directly connected to one memory bank. Together they constitute a module. Intermodule communication is done by a bus or by a network. An example of a loosely coupled multiprocessor connected by a network is the Bolt, Beranek, and Newman Butterfly™ Parallel Processor ([Quinn]). An example of a hybrid loosely coupled multiprocessor using busses as well as a network is the Cm* multiprocessor ([Quinn]).

The next class of communication mechanisms to be dealt with are shared busses. They are used to interconnect processors to processors and processors to memories. It is not very efficient to connect a multitude of processors to a single bus. Only so many processors can share the bus before it becomes saturated. Busses are very common in computers, but their use as central communication mechanism in parallel computers is not very customary. Two computers using busses as part of their communication mechanism are the forementioned Multimax and Balance 8000 tightly coupled multiprocessors and the Cm* loosely coupled multiprocessor.

Finally, we discuss the class of networks. A network consists of a set of nodes that are connected by edges according to some pattern. A node in a network may consist of

- a processor,
- a memory,
- a bus,
- a switching element,
- a comparator with switching element,
- etc,

or a combination of these elements. Edges model connections between these elements.
The most common network models have processors and/or switching elements as their nodes, and wires as their edges. These kinds of networks can be divided into the class of circuit-switched and packet-switched networks.

To mail data in circuit-switched networks, a physical path is established from source to destination. It endures as long as there is a data stream from source to destination. All nodes on the path deal with a data stream as long as it is not terminated.
On the contrary, packet-switched networks put data into packets which are routed through the network. So, no physical paths between source and destination are established, and of all nodes on the route followed, only one (or two consecutive) node(s) deal with sending of the data.
In general, circuit-switching is more suitable for bulk data transmission, and packet switching for mailing of short messages.

Another classification of networks which is of interest, is that according to their ability to change their structure. We distinguish two categories of networks:

- Dynamical networks.
- Statical networks.

In dynamical networks connections between processors can be shifted, so they may change in time. Opposite to this are statical structures, which can not be reconfigured.

Dynamical networks are networks capable to establish a connection between any two processors. The connections between processors in dynamical networks pass via a number of consecutive switching elements. Dynamical networks may be viewed as statical networks of which the nodes contain switching elements. The input and the output terminals of the switching elements are connected to other switching elements or connected to the processors. Dynamical networks known up to now can be grouped into three categories:

- Single-stage networks.
- Multi-stage networks.
- Crossbar switch.

A single-stage network consists of one stage of switching elements, each being able to make a direct connection to one of a limited number of elements in the stage. A connection between two processors may pass through several elements in the stage. A single-stage network may be viewed as a statical network with a combination of a processor and a switching element in each of its nodes. The perfect shuffle ([Stone1]) is an example of a single-stage network.

Multi-stage networks are composed of more than one stage of switching elements. The processors are connected to the input terminals of the first stage and the output terminals of the last stage, or, in so-called one-sided networks, all

processors are connected to input and output terminals of one of the stages. In literature many multi-stage networks have been described, such as the SW-Banyan, the Omega ([Lawrie]), the Flip ([Batche]), the Delta, the Baseline ([WuFeng]), the Benes ([Benes]), and the Clos network. Most of these networks require O(log n) stages of O(n) switching elements to connect n processors. Consequently, their time delay is O(log n) and the hardware costs of such networks are O(n.log n).

A crossbar switch ([Feng]) consists of a number, say n, of vertical parallel input lines and the same number of horizontal parallel output lines. On the intersections of these lines switching elements are placed, which can establish a connection between the lines. Each line can be connected to only one line perpendicular to it by such an element. A crossbar switch can be used as interconnection network between processors by connecting each processor to one input and one output line. Since a network with n processors requires n^2 switching elements, a crossbar switch is unsuited for parallel computers with many processors.

The remaining class of networks are the statical networks. Due to the unchangeability of these networks, their structure is often referred to by the term 'topology'. In the course of time many different topologies have been proposed in literature. The best-known among them are (see figure 1.1) the mesh ([BaBrKa]) of which the 2-dimensional and the 3-dimensional versions are well-known, the binary tree, the totally connected network ([Aupper]), the ring, the hypercube, the cube-connected-cycles ([PreVui]), which is based on the cube, the shuffle-exchange ([Stone1], [LanSto]), and the linear array ([Kung]). In these networks each processor is equipped with a local memory.

The above topologies are often denoted as point-to-point topologies, since each of their edges connects one node to one node. These structures can very well be modeled by graph theory. Nodes in a network correspond to nodes in a graph, and edges in a network to edges in a graph.
Networks which are less well-known but which attract increasing attention are those based on hypergraphs ([Berge]). In a hypergraph, each edge is able to connect more than two nodes. An edge in a hypergraph is not viewed as a wire, but as a relation between all nodes it connects. Networks based on hypergraphs might consist of processors on the nodes, connected by memories or busses modeled by the edges.

A more elaborate general overview of communication mechanisms can be found in [Quinn]. For a survey of networks we refer to [Feng].

Finally, we shall make a choice between the communication mechanisms. This choice is the base for the rest of this dissertation. For this we recall some of the

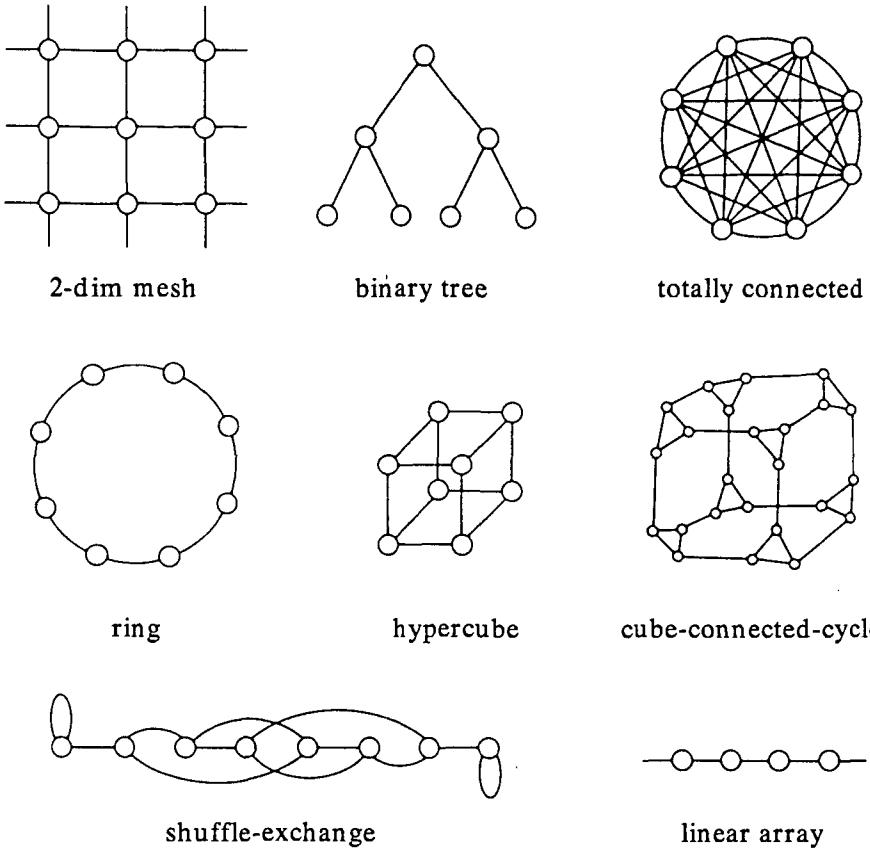


Figure 1.1 Some well-known topologies.

demands stated in subparagraph 1.4.2.

- The communication mechanism should be applicable in massively parallel computers.
- The amount of hardware supporting communication should be small.
- The hardware should not be too complex.
- The structure of the communication mechanism should be regular.
- The communication bandwidth should be large.
- Communication software should be simple.
- The processes should be scheduled on the processors such that interprocessor communication is minimized.

A bus can not efficiently be applied as communication mechanism for massively parallel computers, because it does not support a large communication bandwidth.

A shared memory model in combination with a network is more attractive. This model seems to be a bit more complex than the pure network model, especially when it is combined with extensibility. For reasons of simplicity of the hardware, we shall drop shared memory models, and concentrate on pure network models. In subparagraph 1.4.3 we concluded that both cost and performance metrics favour architectures in which communication is localized. This is the reason to adopt network models in which each processor has its own local memory.

Should we prefer dynamical or statical networks?

A dynamical single-stage network should not be used, because simultaneous connections between more than one processor pair may result in conflicts. For, there is only one stage via which connections between processors are allowed to pass.
Dynamical multi-stage networks and crossbar switches require more complex and a larger amount of hardware than statical networks. Furthermore, to control the switches of a dynamical network complex software is needed. This pleads for statical networks.
Whether statical networks satisfy the demands concerning their structure and communication bandwidth depends of their topologies. There are no principal impediments to regularity and high communication bandwidth in a statical network; the forementioned statical networks in this subparagraph all satisfy these requirements.

How about the remaining demands, i.e. can statical networks be applied in massively parallel computers and can schedulings minimizing communication be established?

Statical networks can surely be applied in massively parallel computers.
The demand that jobs should be scheduled on processors such that communication is minimized is less easy to meet for statical networks. The load imposed to a system by communication will be minimal if communication patterns can be matched optimally onto the network. This favours dynamical networks, because they can be adapted to the communication patterns. For statical networks the situation is less beneficial: only communication patterns isomorphic to the topology of the network can be mapped optimally.

Due to the advantages of statical networks, and in spite of the advantages of dynamical networks, we prefer statical networks. The main argument for this decision is the simplicity of statical networks. Research in a rather unexplored field as extensible computers should be initiated at the simplest cases. The choice

for statical structures leaves open the question whether hypergraph networks will be considered in addition to point-to-point topologies. Because of their simplicity in mathematical sense as well as in their realization as computers, we prefer point-to-point topologies and reject hypergraph models. Statical point-to-point topologies will be considered in more detail in the next subparagraph.

1.4.5 More about statical point-to-point topologies

In this subparagraph a set of demands specific for statical networks based on point-to-point topologies will be formulated. Due to the analogy of statical point-to-point topologies to graphs these demands will be stated in graph theoretical terminology.

The first demand on statical networks concerns the *degree* (see appendix A). The degree of a node in a network is linearly proportional to the number of I/O-ports of the corresponding processor. I/O-ports require a considerable part of the area of a chip. In addition, a large number of I/O-ports on a chip causes the chip to have many pins, especially when the I/O-ports are parallel ports. To limit the costs of I/O-ports, the degree of the processors should be low. Furthermore, since all processors are logically identical, they should all have the same degree. So, the demand on the degree is:

1. The degree of a statical network should be low, and all nodes should have the same degree.

Let's consider the worst-case communication time. Concerning demand no. 8 in subparagraph 1.4.2 the worst-case communication time should be low. The worst-case communication time in a statical network is determined by two factors:

- The *diameter* of the network.
  The diameter of a network is the maximum of the lengths of all shortest paths in it (see appendix A).
- The *routing function* of the network.
  The routing function of the network determines the routes by which data is sent between processors.

In order to obtain a low worst-case communication time, the diameter of a network should be small, and the routing function should not route via long detours. To get an impression of the minimal value of a graph's diameter we consider the so-called Moore-bound ([BeBoPa], [BeDeQu], [Uhr; p.138]). In an undirected graph of degree k the number of nodes, n(k,r), within distance r from an arbitrary node is limited according to the following formulas:

$n(2,r) \leq 2.r+1$,
$n(k,r) \leq (k.(k-1)^r-2)/(k-2)$ if $k \geq 3$.

The distance r is at most equal to the diameter of the graph. Substituting the diameter for r gives the Moore-bound. The diameter D of graphs of degree k can never be less than $\Omega(\log n(k,D))$.
The Moore-bound for $k=2$ is obtained by considering a circuit graph of $2.r+1$ nodes.
The Moore-bound for $k \geq 3$ is obtained by starting with a tree of which all non-leaf-nodes have degree k, and which consists of $r+1$ complete levels. The diameter of such a tree is twice the distance from any leaf to the root, since the root lies on some leaf-to-leaf paths. By adding edges to the tree until all leaves have degree k rather than 1, at best the diameter can be reduced to the root-leaf distance. The latter distance is not influenced by addition of the edges, since no new edges are added to nonleaf nodes.

It has been proved by several authors (see for example [Biggs; chapter 23]) that only three nontrivial graphs other than circuit graphs achieve the Moore-bound: the Petersen graph ($r=2$, $k=3$; see figure 1.2), the Hoffman-Singleton graph ($r=2$, $k=7$), and possibly a graph with parameters $r=2$ and $k=57$, which has not yet been discovered.

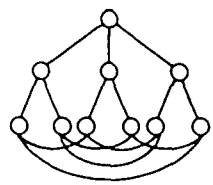


Figure 1.2 The Petersen graph.

Networks with a logarithmic diameter are indeed possible - for example the binary tree. So, it is not unrealistic to reformulate the demand for a low diameter as:

2. The diameter of a statical network should be a logarithmic function of the number of nodes.

The other factor which determines the worst-case communication time in a statical network is the routing function. Before stating a demand on it, we first define it formally.

(1.1) Definition. Let Γ be a connected graph and $r_\Gamma: V(\Gamma) \times V(\Gamma) \rightarrow 2^{V(\Gamma)}$ be a function for which $r_\Gamma(v,v)=\{v\}$ and $\varnothing \subset r_\Gamma(u,v) \subseteq \Gamma_1(u)$ if $u \neq v$, where $\Gamma_1(u)$ is the set of neighbours of node u (see appendix A).

Let $R_{\Gamma i}: V(\Gamma) \times V(\Gamma) \to 2^{V(\Gamma)}$ be defined by

$$R_{\Gamma 0}(u,v) := \{u\}$$

$$R_{\Gamma i+1}(u,v) := \bigcup_{w \in R_{\Gamma i}(u,v)} r_\Gamma(w,v) \quad \text{for } i=1,2,....$$

Then, r_Γ is a *routing function* of Γ if for all $u,v \in V(\Gamma)$ there exists an integer N such that for all $i \geq N$: $R_{\Gamma i}(u,v) = \{v\}$. $r_\Gamma(u,v)$ will be supposed to route from u to v.
$R_{\Gamma i}$ is called the *routing trace* of r_Γ in Γ.
An *optimal* routing function is a routing function for which $R_{\Gamma d(u,v)}(u,v) = \{v\}$ for all $u,v \in V(\Gamma)$.

A routing function, as defined by this definition, routes from a node u to a node v by determining all neighbouring nodes $R_{\Gamma 1}(u,v)$ of u which are to lie on a route from u to v. Applying the routing function to one of the nodes in $R_{\Gamma 1}(u,v)$ and v results in a set $R_{\Gamma 2}(u,v)$ of nodes, each of which is a neighbour of at least one node in $R_{\Gamma 1}(u,v)$. Repeating this process N times results in the sets $R_{\Gamma 1}(u,v)$, $R_{\Gamma 2},\ldots, R_{\Gamma N}(u,v)$. This sequence traces all routes between u and v via which r_Γ may route. If $N = d(u,v)$, then the nodes in $\Gamma_i(u,v)$ have distance $d(u,v)-i$ from v. In that case, the routing function routes via shortest paths from u to v, and is optimal.

The consequence of non-optimality of a routing function is that the worst-case communication time may not be linearly proportional to the diameter of a network. The following should hold in order that the worst-case communication time is linearly proportional to the diameter:

> The length of any route between two arbitrary nodes determined by the routing function is of the same order as the distance between those two nodes.

Though this demand guarantees a logarithmic worst-case communication time in a network with a logarithmic diameter, it does not guarantee a low worst-case communication time. Therefore, we demand

3. The routing function of a statical network should be optimal.

Demand no. 9 in subparagraph 1.4.2 stated that the *communication bandwidth* of a communication mechanism should be high. Whether this demand is satisfied for a particular statical network depends of the structure of the network. The precise relation between the bandwidth and a network's structure is not clear, but communication bandwidth appears to be very high for most networks. The reason for this is that each processor can often find a recipient to send a message to, simultaneously to the mailing of other processors. This seems to be even more valid, when the network's degree is high. So, the demand for a high

communication bandwidth does not impose a very severe restriction onto statical networks.

A metric somewhat similar to communication bandwidth is the *connectivity* of a network (see appendix A). As with the communication bandwidth, the connectivity should be high. High connectivity results in lots of disjunct paths between any two nodes - and in most graphs even much more non-disjunct paths between those nodes. This implies that there are many paths by which messages can be routed between processors.
High connectivity has a positive influence on the total number of messages which can be handled by a network per unit of time. Consequently, it limits congestion in a network. Another advantage of high connectivity comes into prominence when there are faults in the network, i.e. when processors or connections between processors are defective. High connectivity eases bypassing of messages in that case. So, we demand:

4. The connectivity of a statical network should be high.

Concerning the structure of communication mechanisms, demand no. 7 in subparagraph 1.4.2 stated that it should be *regular.* Applied to statical networks this demand becomes:

5. A statical network should have a regular topology.

In addition to the motives for this demand mentioned in subparagraph 1.4.2, there are some motives specifically tailored to statical networks.
The first motive to demand regularity of a network concerns the routing of messages in a network. Irregular structures may need a table in each processor for routing data. For large networks such tables consume much memory space, because each node requires a table with size linearly proportional to the total number of nodes. It is preferable to route by using a function. This function should not be complicated. Such a property will be reflected by simplicity of the software controlling communication (see demand no. 6 in subparagraph 1.4.2). Which conditions must be satisfied to obtain networks with simple routing functions is currently unknown. At least it may be clear that the aspirations to a simple routing function are not hindered by a regular network structure.
The second motive to demand a network to be regular concerns its realization in hardware. A regular structure often results in an efficient packing of components on chips as well as on printed circuit boards. Furthermore, a regular structure of a network simplifies the design of chips and printed circuit boards.

The question remains how regularity is represented in graph theoretical terms. We use automorphism groups for this (see appendix A). The size of a graph's automorphism group gives an indication of its structure. The larger the group,

the more regular is the graph. Appendix A distinguishes four main categories of graphs classified to regularity: node-transitive graphs, edge-transitive graphs, symmetric graphs and distance transitive graphs. Symmetric graphs in particular will be of significant importance in this dissertation.

We conclude the list of demands with a demand concerning the routing function of a statical network. It was implicitly stated above in the discussion about regularity of a statical network:

6. The routing function of a statical network should be simple.

This concludes the formulation of the demands on statical networks, as well as this paragraph about communication. In the next paragraph another important aspect of massively parallel computers will be considered.

1.5 Extensibility

1.5.1 Why extensibility?

The need for extensible computers is a direct consequence of the ever-growing demands of computer users on computation capacity. The most common way to satisfy these demands is the purchase of new computers in substitution for older ones. This is an expensive solution. Not only should hardware, actually not yet outdated, be replaced, it is also very likely that existing software must be adapted to a new computer.
Extensibility of a computer will relieve these difficulties, provided the concept is implemented properly. A computer is considered properly extensible if its infrastructure need not be changed when the computer's capacity is increased.
Extensibility will not only result in lower direct costs, design costs of a computer will also be lower. For, they can be spread over a multitude of (identical) components. As a result, the hardware design bottleneck will be broadened.
Extensibility will also broaden the software design bottleneck. Extensibility results in a longer life-cycle of the software running on a computer, provided the computer's underlying structure is maintained under extension. An architecture preserved by extensions promotes software compatibility. An additional advantage of preservation of a computer's structure under extension is that software can run on computers of different sizes. This implies that the costs for developing software can be spread over more computers (see also [LipMal; p.42]).

The merits of extensibility open up interesting perspectives to computer designers. Not surprisingly, literature shows a growing interest in extensibility of networks, also denoted by expandability, scalability or inductivity. Agrawal, Janakiram and Pathak give in [AgJaPa] an overview of some networks and their

characteristics. They consider extension capability as an important property of networks. Extensibility is more elaborately discussed by Lipovski and Malek in [LipMal]. They make a distinction between plainly extensible computers and *inductive* computers. Extensibility as defined by them satisfies less stringent demands than inductivity. Their inductivity concept looks like our concept of extensibility still to be described, with one main difference: an inductive computer can be extended by 1 processor. This property is of no use for massively parallel computers, as will be shown in demand no. 2 in subparagraph 1.5.3.

As far as I know, Lipovski's and Malek's book is the most elaborate treatment of extensibility in literature. Articles dealing with extensible architectures are [Parber], [DesPat], [GooSeq], [FrHeHe], [HoKuRe], [HwaGho], [Snyder], and [Shaw]. The ones among them based on statical point-to-point topologies will be described in subparagraph 1.5.5. In the next subparagraph, extensibility is related to graph theory. Subparagraph 1.5.3 deals with some demands on extensibility. Thereupon, subparagraph 1.5.4 discusses some architectural limitations to the performance of extensible computers. Subparagraph 1.5.6 deals with the devices that are to be connected to an extensible network of processors. Finally, subparagraph 1.5.7 gives some marginal notes to extensibility.

1.5.2 Extensibility and graph theory

This subparagraph starts by defining extensible networks in terms of graph theory. Thereupon, it introduces some graph theoretical notions related with extensible networks. Because of the similarity between networks and graphs, the terms 'extensible network' and 'extensible graph' will be used indifferently.

An extensible network is viewed as a finite graph being an element of a sequence of finite graphs $(\Delta_0,\Delta_1,\Delta_2,\ldots)$, for which:

$$\Delta_i \subset \Delta_{i+1} \quad (i \geq 0),$$

that is, Δ_i is a proper subgraph of Δ_{i+1}.

The sequence $(\Delta_0,\Delta_1,\Delta_2,\ldots)$ is called an *extension sequence* of Δ_i. An extensible graph may have several extension sequences. Throughout this dissertation it is assumed that whenever a particular extensible graph is denoted by Δ_i (or any other symbol with subscript), its extension sequence is defined to be $(\Delta_0,\Delta_1,\Delta_2,\ldots)$ (or a sequence based on another symbol with subscript). The extension sequence of Δ_i will be denoted by $S_E(\Delta_i)$. When discussing some extensible graph Δ_i in this dissertation, i being an open variable, it is assumed to be an *arbitrary* element of its extension sequence $S_E(\Delta_i)$.

The result of extending a graph Δ_i is determined by its extension sequence.

Extending Δ_i results in a graph Δ_j ($j>i$), called an *extension* of Δ_i. Extending Δ_i by a minimal number of nodes results in Δ_{i+1}. The set $V(\Delta_{i+1})-V(\Delta_i)$ is the set of nodes added to Δ_i in that case. The number of nodes in this set is called the extension complexity of Δ_i.

(1.2) Definition. The *extension complexity* of an extensible graph Δ_i is defined by

$$C_E(\Delta_i) := |V(\Delta_{i+1})-V(\Delta_i)|.$$

The extension complexity of Δ_i is *optimal* whenever $C_E(\Delta_i) = O(1)$.

So, the extension complexity of an extensible graph Δ_i is optimal when it can be extended by addition of a *constant* number of nodes.

Extensible statical networks differ from plain statical networks, in that some of the edges of the former are *loose*, i.e. incident to only one node. Loose edges are necessary to extend a network.
Different kinds of nodes in an extensible network can be distinguished: nodes incident to loose edges, called the *border nodes*, and the other nodes, called the *internal nodes*. Nodes that still have to be added to the network are called *external nodes*.
The processors corresponding to the border nodes in an extensible network have open I/O-ports, i.e. ports not connected to other processors. These ports will be used for transmission of jobs and data to and from the network.

1.5.3 Demands on extensibility

Subparagraph 1.5.1 laid the emphasis on *proper* extensibility. Which demands should be satisfied by a computer to be proper extensible will be discussed in the current subparagraph. It is assumed that the communication mechanism used is a statical point-to-point network.

Some general demands, i.e. demands not related to the communication mechanism used, are:

1. The size of extensions should be limited.
   The minimal number of processors needed to extend a computer should be of the same or lower order as the number of processors already existent in the computer. Hardly any customer will be interested in a computer of which the smallest extension is many times more expensive than the computer itself. Extensions at most doubling a computer's size seem to be reasonable. Applied to statical networks this demand becomes
   $$C_E(\Delta_i) = O(|V(\Delta_i)|).$$

2. The size of extensions need not be extremely small.
   Actually, this is not a demand but a relaxation of the previous demand. Lipovski and Malek propagate in [LipMal] extensibility by 1 processor. Their preference for that small extensions might be explained by the fact that it simplifies analysis of matters like performance improvements as result of extension, etc. From a practical point of view, their demand has no sense for massively parallel computers and is much too strict. This can simply be made clear by posing the following question:

   > Why should one extend a 100000-processor system to a 100001-processor system?

   The increase of the computation capacity of a massively parallel computer as a result of extension by 1 processor is negligible in terms of percentage. So, the number of processors in an extension should be in proportion to the number of processors already existent in the computer. Applied to statical networks this demand becomes $C_E(\Delta_i) = \Omega(1)$.

3. There should be no upper bound to the number of processors in an extensible computer.
   If there is such an upper bound, the computer is not truly extensible. From a practical point of view, a computer with a very large upper bound is of course extensible. For reasons of convenience we shall assume that no constant bounds the number of processors in an extensible computer. Any extensible computer should permit extension by an unlimited number of processors. The extension sequence of a statical network satisfying this demand, is an infinite sequence.

4. Software which runs on a computer should also run on the extended computer. In particular, this should be true for system software.
   An extensible computer not satisfying this demand is not very meaningful. This demand is related to the next demand, which refers to the structure of an extensible computer.

5. The structure of an extensible computer should be maintained under extension.
   There are two motives for this demand.
   The first motive refers to the previous demand, concerning software compatibility under extension.
   The second motive is that it enables an easier and more precise analysis of performance of extensible computers. This is important, both from a theoretical as from a practical point of view.
   In another context, the above demand is encountered once again in this list

of demands (see no. 13). That demand is specifically tailored to the structure of statical extensible networks.

6. An extensible computer should take advantage of advances in technology, i.e. it should remain technologically up-to-date.
   The life of a parallel computer is not only determined by its capability of being extended by new processors, but also by the technological standard of these components. If the processors used for extension are of the same technological standard as those in the original computer, the latter will become out-of-date in the course of time, since technology is always advancing. To prevent this, a computer should always be extended by the most up-to-date components.

7. The performance of an extensible computer should be linearly or almost linearly related to its number of processors.
   Actually, this is a demand on software, and in particular to system software running on the computer. The amount of pure MIPS increases linearly with the number of processors, but the degree to which these MIPS manifest themselves in the computer's performance is determined by software. If this demand is not satisfied, a disproportionate amount of money must be invested to increase a computer's performance.

8. The efficiency with which a fixed sized job is executed should not decrease radically when the computer used is extended.
   The increase of a job's overhead as a consequence of extension should be limited. If the above demand is not satisfied, an extensible computer can not be applied efficiently.
   This demand correlates slightly to demand no. 12 in subparagraph 1.4.2, which stated that communication in different jobs should not interfere with each other. If data streams in different jobs are not independent of each other, and the number of jobs run on the computer is linearly proportional to the number of processors, then extension of the computer causes an increase of communication overhead in the jobs.

Demands no. 9 up to 15 are more specific for statical point-to-point networks:

9. The degree of the network and of all its extensions should be bounded from above by a fixed positive constant.
   Subparagraph 1.4.5 stated that a node's degree is linearly proportional to the number of I/O-ports on the corresponding processor. This number is not allowed to increase just like that. A statical network not satisfying this demand is the k-cube. Extension of a k-cube to a $(k+1)$-cube causes the degree to increase from k to $k+1$.

10. A network's diameter, expressed as function of the number of nodes, should remain of the same order under extension.
    Some particular worst-case communication time of an extensible network can only be guaranteed when this demand is satisfied. A logarithmic diameter should also be a logarithmic diameter after extension.

Demands similar to the previous one can also be stated for other characteristics which express the performance of a network. For communication bandwidth this results in demand no. 11:

11. A network's communication bandwidth, expressed as function of the number of nodes, should remain of the same order under extension.

12. The connectivity of the network should not decrease when it is extended.
    A high connectivity, the benefits of which are pointed out in subparagraph 1.4.5, should be maintained under extension. For this reason extension by only one processor, as propagated by Lipovski and Malek, may even be disadvantageous. If the newly added processor is connected to only one other processor in the network, the resulting edge-connectivity will be 1. Naturally, subsequent extensions with other processors might restore the connectivity, but why then add them not all at once? It would result in an 'extension-module', that, when added, would keep the network's connectivity intact.

The first two of the following demands were encountered before in a more general form.

13. The underlying structure of the network should be maintained under extension.
    An explanation of this demand can be found with demand no. 5 in this list. Networks not satisfying the above demand are the shuffle-exchange and the cube-connected cycles. Not only processors should be added to extend them; the shuffle-exchange and the cube-connected-cycles should also be restructured.

14. The routing function should be maintained under extension.
    The description of a routing function should be left unchanged by extensions. As a consequence of this demand, the description of a routing function should be very general. It should even be so general as to be able to establish a route from a processor in a network to a processor not yet added to the network.
    The routing function is the base for the system software dealing with communication. So, the demand is a special case of the fourth demand of this list.

15. The routing function should only route via paths inside the network.
    A routing function is worthless, if it establishes routes between processors in the network which pass through processors not yet added to the network. A message sent via these routes would never arrive. This is a trivial demand for non-extensible networks. However, this demand is harder to be satisfied by extensible networks. It requires the shape of an extensible network to be adapted to its routing function (and the other way around).

We conclude this subparagraph with an example of two well-known networks which satisfy all demands concerning extensibility. Those networks are the mesh and the binary tree. Though both satisfy the demands in this subparagraph, they are not the kind of networks we are looking for. They don't satisfy all demands on statical point-to-point topologies (see subparagraph 1.4.5). The diameter of the mesh is not logarithmic and the connectivity of the binary tree is only 1. For a survey of extensible networks that do satisfy all demands on statical point-to-point topologies, we refer to subparagraph 1.5.5.

1.5.4 Architectural limitations to the performance of extensible computers

In many parallel computers bottlenecks occur, affecting the performance. Such bottlenecks are mainly caused by a shared resource, such as a common bus, a common memory, etc. If an extensible computer contains such an inherent bottleneck, it is not properly extensible. An example of such a computer is an installation which consists of separate computers connected to an Ethernet. This computer can be extended indefinitely. In most applications, however, the efficiency of the installation will be low (see also [LipMal; p.36]). It is caused by the limited capacity of the bus.
An extensible computer should not contain this kind of bottlenecks. Unfortunately, a bottleneck can be less explicit than above. In this subparagraph they are visualized.

In [LipMal; p. 45, Theorem 2] Lipovski and Malek give a characterization of bottlenecks in extensible computers. It amounts to:

> Any resource, that is unique in an extensible system, and is called on by all processors with probability bounded from below by some fixed positive constant, and that can not be used concurrently, restricts the performance of the system.

As the number of processors in the system increases, the load of the resource will increase up to the point that the resource is fully used. Beyond that point the system is too large for that one resource, as a consequence of which all requests of the processors can not be handled any more. This affects the efficiency and

usefulness of the system.

The unique resource can have several manifestations, such as a shared bus, or a memory which can only be accessed simultaneously by one processor. The resource can also be less explicit, such as in the linear array at which software imposes a purely random or uniform communication pattern. Under such a communication pattern, the probability that a communication path passes through a node somewhere close to the middle of the array is larger than a fixed positive constant. When the number of processors grows, the node in the middle will behave like a bottleneck.
This problem is not restricted to a linear array. It will occur in all connected extensible statical networks. It is caused by the communication pattern, not that so much by the architecture. Communication patterns mainly exhibiting locality, will result in considerably fewer bottlenecks. We conclude that not only features of the architecture, but also factors like the communication pattern may impose a restriction to the performance of extensible computers.

In addition to the factors mentioned above, a computer's age, and related to that its technological outdatedness, may also cause bottlenecks in a system. Demand no. 6 in subparagraph 1.5.3 stated that extensible computers should be extended by modern components in order to remain up-to-date. However, whether or not a computer is extended by modern components, its core remains always the same. Consequently, the core will be outdated after some years. This causes the core to become the slow part of an extensible computer.
The age of the core may decrease the performance of the computer, though that is not inevitable. If the core lies centrally in the network, uniform or random communication patterns will cause many communication paths to pass through it. The resulting bottleneck is underlined by the age of the core. However, if we succeed to extend the computer in such a way, that the old core is located at a non-central part somewhere in the computer, and the modern components are directed to the main parts, then a decrease of performance will less strongly be felt.

In addition to the restrictions dealt with in this subparagraph, there are also practical reasons for reservations against extensibility. In order not to discourage the reader, we postpone their treatment to subparagraph 1.5.7.

1.5.5 Overview of extensible statical point-to-point networks

This subparagraph gives an overview of extensible statical point-to-point networks. All networks discussed here have the following characteristics:

1. Their degree is bounded from above by a fixed positive constant.
2. Their diameter is logarithmic with respect to their number of nodes.
3. Their connectivity is larger than a tree's connectivity, i.e. larger than 1.
4. Their structure is more or less regular.

The following networks will be considered: the butterfly (e.g. see [Ullman] or [Quinn]), the cube-connected-lines ([Parber]), the X-tree ([DesPat]), the hyper-tree ([GooSeq]), the pyramid ([FrHeHe]), the binary cluster tree ([HoKuRe]), and the hypernet ([HwaGho]).

The butterfly network (see figure 1.3) consists of $(r+1).2^r$ nodes divided over $r+1$ rows each containing $q=2^r$ nodes.

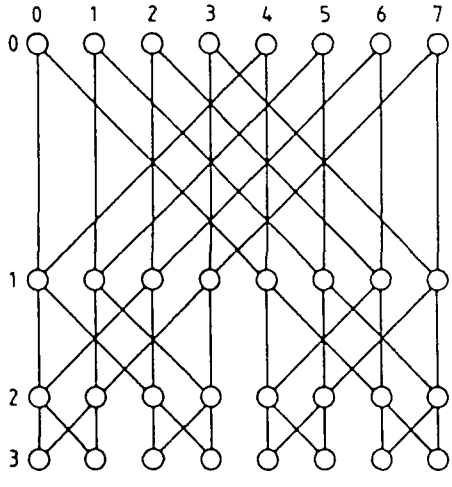


Figure 1.3 The butterfly.

The rows are labeled from 0 up to r and the columns are labeled from 0 up to $q-1$. The nodes in a butterfly are labeled by 2-tuples. Label (i,j) refers to the node in row i and column j, where $0 \leq i < r$ and $0 \leq j < q$. Node (i,j) in row $i>0$ is connected to two nodes in row $i-1$, i.e. nodes $(i-1,j)$ and $(i-1,m)$, where m is obtained from j by inverting the i^{th} most significant bit of j. The degree of any node in the butterfly is at most 4.
A butterfly with $(r+1).2^r$ nodes is extended to a butterfly of $(r+2).2^{r+1}$ nodes, by taking a copy of the first one and add a node to each of the nodes in row 0 of the copy and the original. Thereupon, the rows are renumbered: the newly added nodes get row label 0 and the other row labels are increased by 1. Finally, the nodes just added to the original/copy are connected to the nodes with row label 1

in the copy/original, following the 'butterfly connecting recipe'. This results in a new butterfly of $2(r+1).2^r+2.2^r = (r+2).2^{r+1}$ nodes. It is easily verified that the extension complexity of a butterfly with $(r+1).2^r$ nodes is $(r+3).2^r$. This is of the same order as the number of nodes in the network.

The cube-connected-lines (see figure 1.4) of Parberry is actually a cube-connected-cycles of which each cycle lacks an edge. An r-cube-connected-lines consists of 2^r lines of r nodes each ($r \geq 1$). So, its total number of nodes is $r.2^r$ and the degree of the nodes is at most 3.

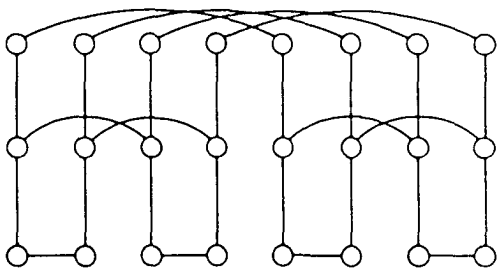


Figure 1.4 The cube-connected-lines.

An $(r+1)$-cube-connected-lines is obtained from two r-cube-connected-lines and 2^{r+1} nodes, by lengthening each of the 2.2^r lines in both r-cube-connected-lines with one node. Thereupon, the just added node in the i^{th} line of the first r-cube-connected-lines is connected to the just added node in the i^{th} line of the other r-cube-connected-lines. From the above extension procedure it is easily deduced that the extension complexity of the r-cube-connected-lines is equal to $(r+2).2^r$. This is of the same order as the number of nodes in an r-cube-connected-lines.

The X-tree is a complete binary tree, augmented with extra edges. The reason to attach these edges, is to obtain a connectivity larger than that of a binary tree (1). Despain and Patterson propose in [DesPat] several patterns to install these edges, resulting in the threaded tree, the double threaded tree, the half-ringed tree, and the half-ringed tree with shuffle. These structures all have degree 4. They also introduce X-trees with degree 5, such as the full-ringed tree with shuffle and without shuffle (see figure 1.5).
Except in the threaded trees the augmented edges only connect nodes at the same tree levels. X-trees are extended by increasing their number of levels. Since each level of a complete binary tree consists of the same order of nodes as the total number of nodes at a higher level, the extension complexity of an X-tree is of the same order as the total number of nodes in it. Most X-trees described in [DesPat] are not very regular.

X-trees somewhat more regular are introduced by Goodman and Sequin in

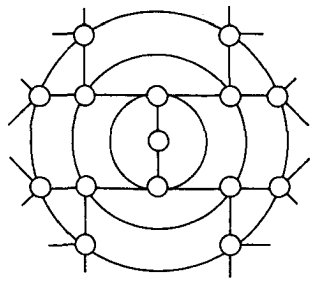


Figure 1.5 The full-ringed tree without shuffle.

[GooSeq], and called hypertrees. Again, in such trees all augmented edges only connect nodes lying at the same tree levels. The edges are defined by first labeling a binary tree in the standard way, i.e. the root label is 1 and the two children of a node with label x have labels 2x and 2x+1 respectively. A hypertree is a tree in which the augmented edges connect only nodes of which the labels differ by 1 bit. The actual number of edges to be augmented depends of the node degree aimed at. If the degree is 4 respectively 5, then at most 1 respectively 2 edges are attached to each node, resulting in the hypertree I respectively the hypertree II. Since each node label often differs by 1 bit from several other node labels at the same tree level, several hypertrees are possible. The hypertrees considered by Goodman and Sequin exhibit a regular pattern. As with the X-trees, hypertrees are easily extended by increasing their number of levels. The extension complexity of a hypertree is of the same order as the total number of nodes in it.

The well-known pyramid (see figure 1.6) described in [FrHeHe] is based on a tree of which all nonleaf nodes have four sons. On each level the nodes are connected as a 2-dimensional mesh. The degree of its nodes is at most 9. The pyramid is often used for image processing, pattern perception, and other areas of AI where information should be simultaneously transformed and converged, or, moving in the other direction, broadcast and diverged (see [Uhr; p. 114]). A pyramid is extended by increasing its number of levels. The extension complexity of a pyramid is of the same order as its number of nodes.

The binary cluster tree is a binary tree, with its root replaced by a regular structured cluster of 2t nodes and each of the other nodes replaced by a regular structured cluster of 3t nodes. Each cluster is connected to its ancestor cluster through t edges, and to each of its child clusters through t edges. Hosseini, Kuhl and Reddy choose in [HoKuRe] a circuit as cluster (we call the resulting structure the *binary cycle tree*). A binary cycle tree with r levels of cluster consists of $2t+6t(2^{r-1}-1)$ nodes and has degree 3. Figure 1.7 depicts a binary cycle tree with parameters t=2 and r=3. A binary cluster tree can easily be extended by

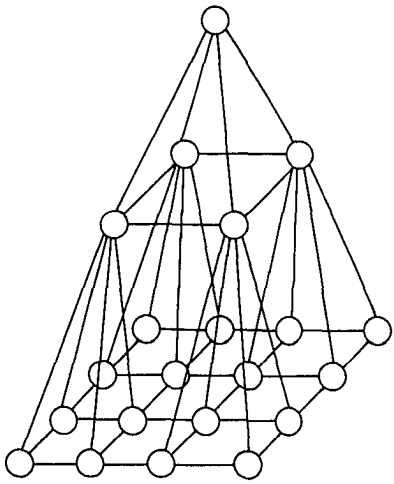


Figure 1.6 The pyramid.

increasing its number of cluster levels. Extension of a binary cluster tree with r levels to one with r+1 levels takes $6t.2^{r-1}$ nodes to add.

A hypernet or (d,h)-net, described in [HwaGho], is a network that is recursively built up of smaller hypernets, (d,h−1)-nets. The smallest hypernets possible, (d,1)-nets, are the building blocks. A building block in a (d,h)-net can be a d-cube, a complete binary tree of d levels or 2^d processors connected to a bus. Only one type is used simultaneously in a hypernet. To the nodes in building blocks loose edges are attached. They are used for I/O or to make connections to other building blocks.
The degree of a (d,h)-net depends of the building block used. If the building block is a cube then the degree is d+1; if it is a tree then the degree is 4; if it is a bus with 2^d processors then the degree is 2. The number of nodes in a (d,h)-net is equal to

$$N(d,h) = 2^{2^{h-1}(d-2)+h+1}.$$

Extension of a (d,h)-net to a (d,h+1)-net requires

$$2^{2^{h-1}(d-2)+h+1}\{2^{2^{h-1}(d-2)+1}-1\}$$

nodes to add, which is approximately the square of the number of nodes in a (d,h)-net. Hence, the extension complexity of a (d,h)-net is not of the same order as the total number of nodes in a (d,h)-net. That is to say, hypernets don't satisfy demand no. 1 in subparagraph 1.5.3.

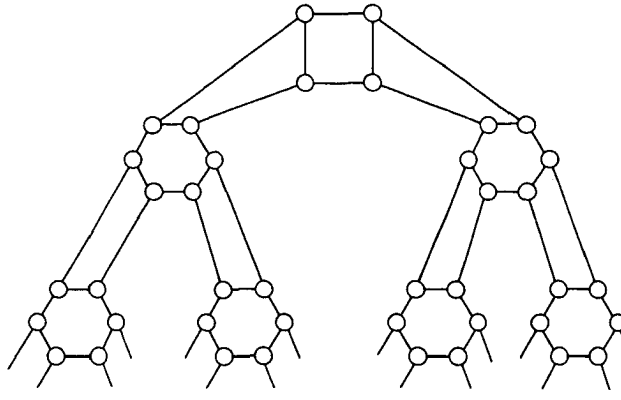


Figure 1.7 The binary cycle tree (t=2 and r=3).

1.5.6 Hosts and peripherals of extensible computers

This subparagraph deals with configurations based on extensible computers. Obviously, the main part of such a configuration is constituted by an extensible network of processors each of which is equipped with a local memory. The loose edges of the network will be used as interconnections to so-called *peripheral computers.* Peripheral computers present jobs to the extensible computer and receive the output of the jobs (see figure 1.8).
Presenting jobs and receiving their output will not be done by a single host. The reason for the absence of a host is straightforward: a host causes a bottleneck in the sense of [LipMal; p.45, Theorem 2] (see subparagraph 1.5.4). It is called on by each processor with a probability larger than a fixed positive constant.
If a sufficient number of peripheral computers is available, no input and output bottlenecks will arise. As will be seen in this dissertation, extensible computers based on a statical network with logarithmic diameter can be connected to sufficiently many peripheral computers. The number of loose edges in extensible networks exhibiting logarithmic diameters, will appear to be of the same order as the total number of processors in the network. So, the number of peripherals that can be connected to a network is allowed to grow linearly with the number of processors. This is indeed necessary when the supply of jobs increases in the course of time, and the average sizes of the jobs remain equal. In practice this situation is not very likely to occur, however. To be sure, there is a steady increase of the number of jobs as a computer's capacity increases, but it is very likely to be exceeded by the increase of computing capacity required by each single job. In that case, the number of peripherals only need to grow sublinearly with the number of processors.
The choice for several peripherals instead of one host has consequences for the

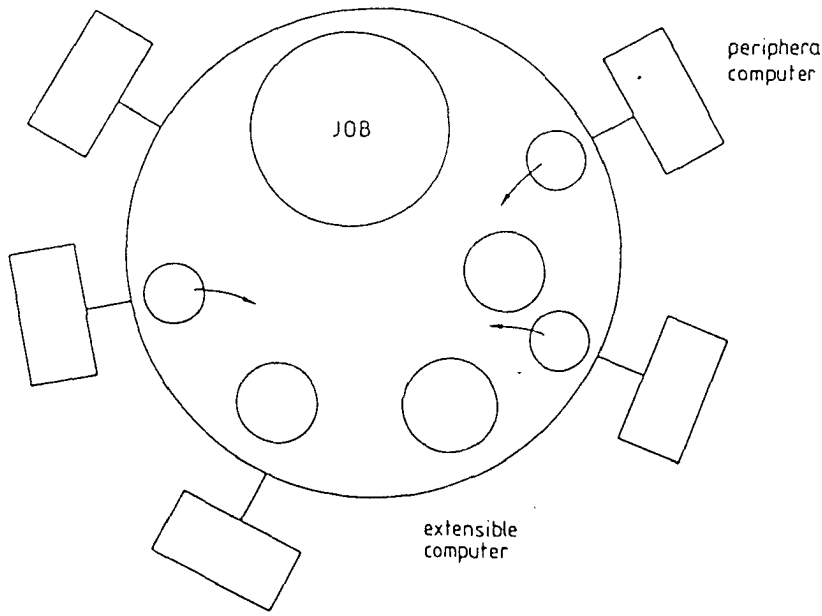


Figure 1.8 Extensible computer with its peripherals.

control of an extensible computer. Controlling any computer is most easily done by a single host. Control by a multitude of (different) peripherals is much harder, however. Concurrent execution of the operating system on the peripherals causes a lot of complications. Therefore, extensible computers should control themselves.

1.5.7 Marginalia to extensibility

We conclude the paragraph about extensibility with some marginal notes. Though extensibility has much to offer, it has some disadvantages. Subparagraph 1.5.4 dealt with some factors that limit the performance of extensible computers. They were mainly caused by the hardware and software of extensible computers. There are also external factors which may subdue the enthusiasm for extensible computers, however.

A supplier's opinion about extensibility might be negative, because he might experience a decrease of his sales by it. In current practice a customer can often be persuaded to purchase a (conventional) computer that is large enough to comply with his computing demands over the next two or so years. However, to the owner of an extensible computer, no more hardware can be sold than is strictly needed by him, provided the extensions are reasonably small. The owner will

wait procuring additional computing capacity up to the moment he really needs it. There is still another factor that may decrease the sales. Whereas completely new computers could often be sold when a customer's needs for computing capacity increased, for extensible computers only extensions will be sold. An extension will represent a lower economical value than a complete computer. Consequently, though extensibility may result in higher sales measured in units of computing capacity, the sales in economical units are very likely to be lower.

The opinion of customers about the above matters will be opposite to that of suppliers. The interests of both are incompatible. However, objections to extensibility can also be expected from customer's sides.
To a customer, the long life-cycle of extensible computers is disadvantageous. According as a computer is extended ever more, switching to another supplier becomes more and more difficult. A big investment in a multiply extended computer would be wasted. The advantages of extensible computers are clouded by the bogey of years (or even decades) of dependency of a particular supplier.
A drawback to the long life cycle of extensible computers, which was mentioned before (subparagraph 1.5.4), concerns the technological advancement of extensions. Extension of a computer by processors being of the same technological standard as the original computer will result in a technologically outdated computer. Even if up-to-date components are used for extension, the user is confronted with an outdated core of the computer, which decreases its performance. The longer the life-cycle of a computer is, the more serious this problem is.

Although the advantages of extensibility are accompanied by some disadvantages, the concept is worth to be investigated in more detail.

1.6 Can extensible computers based on statical networks be efficient?

1.6.1 Introduction

The preceding paragraphs dealt with several aspects of extensible massively parallel computers. First, it was pointed out that massively parallel computers should be capable of executing jobs of different sizes, and moreover, to execute them in parallel. Thereupon, three important issues in massively parallel computers were treated: the processors, communication between them, and extensibility. In these paragraphs a number of demands on extensible massively parallel computers were stated. It is unclear, however, whether computers satisfying these demands are able to process jobs efficiently. The current paragraph gives an indication of which factors the efficiency, by which jobs are processed by massively parallel computers, depend.

1.6.2 Factors influencing performance of parallel computers

Agrawal, Janakiram and Pathak distinguish in [AgJaPa] three factors having bearing on performance of parallel computers with statical communication networks:

1. The mechanism for detecting parallelism and partitioning each job into processes.
2. The topology of the interconnection network.
3. The scheduling and mapping of the processes on the architecture.

The first issue is of concern for any parallel computer irrespective of its communication mechanism and its other characteristics. Automatic detection of parallelism is quite involved and beyond the scope of this dissertation.
In some systems, the mechanism for automatic detection of parallelism is simply omitted. It is left to the users to specify parallelism in their jobs. For example, in the OCCAM-language the job of specifying data dependencies between parallel subtasks is on account of users. They are responsible for the optimal parallelization of their programs.

The efficiency with which parallelism in a job is transformed in speedup depends of the other two factors in the list, i.e. the network topology and the scheduling of jobs on it.
Construction of efficient network topologies is one of the main issues of this dissertation and runs to chapters 2, and 4 up to and including 8.

Scheduling of the processes on the architecture is divided into two subscheduling problems:

1. Scheduling jobs on the processors.
2. Scheduling processes within jobs on the processors.

These two subscheduling problems are not the topics of this dissertation. Nevertheless, the next two subparagraphs deal with the two subscheduling problems.

1.6.3 Scheduling jobs on the processors

The scheduler of jobs on processors establishes which processors are to be used for each job. The set of processors on which a particular job is scheduled is called the *chunk* of processors belonging to that job.
The demands to which chunks and jobs should conform are listed below. The first three demands have been encountered before in the context of whole

extensible networks, rather than chunks in them.

1. Processors executing a particular job should be packed tightly, i.e. the diameter of the chunk should be low.
   As a result of a low diameter of a chunk, the processes within the corresponding job lie close together (see also demand no. 8 in subparagraph 1.4.2, and demand no. 2 in subparagraph 1.4.5).

2. Communication within a job should be independent of the communication in other jobs (see demand no. 12 in subparagraph 1.4.2).

3. The time a processor is computing should be well-balanced with the time it is communicating or waiting for communication (see demand no. 11 in subparagraph 1.4.2). This demand is closely related to the next demand.

4. The number of processors in a chunk should be dynamically adaptable to the needs of a job.
   At moments that the number of parallel processes in a job increases, the corresponding chunk should expand up to the point that the balance between communication and computation is restored (see demand no. 11 in subparagraph 1.4.2). Analogously, if the number of processes decreases, then the chunk should be reduced. By releasing processors at moments at which they are not needed by a job, they will become available for other jobs.
   When this demand is satisfied, the efficiency by which processors are used in a job will improve. Nevertheless, the speedup of the job will remain approximately equal. Amdahl's law can not be sidestepped.

5. The maximal sized chunk which fits in a network, should fit exactly in the network, i.e. the shapes of the maximal sized chunk and the network are equal.
   A computation intensive highly parallel job should be able to allocate all processors of a network. This demand is not as trivial as it seems to be. A chunk is not allowed to have any arbitrary shape, neither does a network (see demand no. 15 in subparagraph 1.5.3). The factors which determine the shapes of networks and chunks will be elaborated in chapter 2.

6. Jobs should be allocatable anywhere in the computer.
   This demand amounts to requiring that the network is identical to each processor. Stated in another way, the network should be node-transitive (see appendix A), and all processors should be logically identical. The demand guarantees flexibility of scheduling.

7. Jobs should be distributed homogeneously among the network.
   This demand discourages situations in which processors are overloaded while

others are idle.

There exists a relationship between the shape of extensible networks and the shape of chunks. This relationship will become clear in paragraph 2.7. In particular, the questions

1. Which shape should a chunk have, in order to have a low diameter?
2. Which shape should a chunk have, in order that its communication is independent of communication in other chunks?

are respectively equivalent with the questions

1. Which shape should an extensible network have, in order to have a low diameter?
2. Which shape should an extensible network have, in order that there exists a routing function, which is maintained under extension, and which only routes via paths inside the network?

The latter two questions, and so, also the former two questions, will be answered in the next chapter.

1.6.4 Scheduling processes within jobs on the processors

The second subscheduling problem concerns the mapping of processes within a job onto the corresponding chunk of processors. The diameter and shape of this chunk are given. They are set by the first subscheduling process and can not be influenced directly by the second. The only task of the second subscheduling process is to find an optimal allotment of the processes among the processors so that the speedup or some other performance parameter is optimized. For this, processes frequently exchanging data should be mapped onto processors which lie close together, or even better, are directly connected. A low diameter of the chunk is of help to minimize the average distance over which processes communicate.
The structure determined by the processes and their communication interrelationships should be mapped optimally onto the chunk of processors. To complicate these matters, in most general purpose computers no information is available on the nature of the processes to be executed on it. The mapping should be done at run time. Unfortunately, dynamical mapping pushes down the total performance of a system.
In literature a limited number of articles have appeared describing research on scheduling and mapping of processes on processors. In [Stone2] Stone describes an optimal assignment of processes to a two-processor system. Bokhari proposed in [Bokhar] a more complete solution for an arbitrary number of processors. A

method applicable to large homogeneous multiprocessor systems, called *wave-scheduling* is described by van Tilborg and Wittie in [TilWit]. It assumes that any process can be executed on any one of the processors. A more elaborate treatment of these methods may be found in [AgJaPa] or in the original articles. Finally, we mention [Hilber], in which Hilbers describes a theory about mappings of processes onto processors. These mappings minimize the largest distance between the processors at which two neighbour processes are mapped.

2

A method to construct efficient extensible networks

Géronte: It seems to me you are locating them wrongly: the heart is on the left and the liver is on the right.
Sganarelle: Yes, in the old days that was so, but we have changed all that, and now we practise medicine by a completely new method.

Molière (1622-1673)

2.1 Introduction

This chapter describes a method suitable to construct extension sequences of which each element has the following properties:

1. There is no bound to the number of extensions that can be made to the network.
2. The degree of the network and of all its extensions is bounded from above by a fixed positive small constant.
3. The network's diameter is logarithmic and remains logarithmic after extension.
4. The network's connectivity is reasonably high and is maintained under extension.
5. The network's structure is regular and is maintained under extension.
6. Any optimal routing function of the network routes only via paths inside the network, and is maintained under extension.
7. The network's extension complexity is minimal with respect to the above properties.

Furthermore, the relationship between extensible networks and chunks, which were introduced in subparagraph 1.6.3, is discussed.

The next paragraph first gives a rough outline of the construction method and then relates it to the properties listed above. It appears that the first two properties are easily provided by the method. The subsequent paragraphs put in the other details of the method. Paragraph 2.3 discusses the preconditions to obtain extensible networks with logarithmic diameters. Regularity and connectivity, which will appear to be closely related, are studied in paragraph 2.4. Paragraph 2.5 gives more details about the actual shape of the network and relates it to routing functions. The method itself is described in full detail in paragraph 2.6. Paragraph 2.7 relates the shapes of chunks to the shapes of extensible networks constructed with the method.

2.2 Fundamentals of constructing extensible networks

This paragraph describes the fundamentals of constructing extensible networks with the properties listed in the introduction. First of all, a method to construct such networks is sketched. The method consists of two stages:

- Construct an infinite connected graph Γ.
- Cut out subgraphs of Γ.

The subgraphs cut out of Γ will constitute the extensible networks. To obtain properly extensible networks, the subgraphs are cut out such that they constitute an extension sequence $S_E(\Delta_i) = (\Delta_0,\Delta_1,\Delta_2,\ldots)$, each member of which is an *induced* subgraph (see appendix A) of Γ. Γ will be called an *underlying* graph of the sequence $S_E(\Delta_i)$. The way the extensible networks are cut out of Γ determines their *shape.*
In the remainder of this paragraph we deal briefly with the conditions that Γ and Δ_i should satisfy in order to have the properties listed in the introduction. The precise conditions they should satisfy will become clear in the course of this chapter.

The first property, stated in the introduction, concerns unboundedness of extensibility. It is easily incorporated in extensible networks constructed by the construction method. The sequence $S_E(\Delta_i)$ can be of infinite length, since Γ is an infinite connected graph. Consequently, there is no bound to the number of times Δ_i can be extended.

The second property concerns the degree of extensible networks. This property is also easily imposed to the members of an extension sequence $S_E(\Delta_i)$, by assuming the degree of Γ to be finite and low. In addition to this we assume that all nodes of Γ have the same degree. Clearly, the degree of any node in any Δ_i is bounded from above by the degree of Γ, since the total number of loose and non-loose edges any node in Δ_i is incident with, is equal to $\deg(\Gamma)$. This implies that the

degree of any node in any Δ_i is finite and low.

The third property concerns the diameter of extensible networks. The diameter of an extensible network Δ_i depends on two factors:

- The node density of Γ.
- The shape of Δ_i.

Informally, the node density of Γ is the number of nodes within some distance from some node in Γ. This number is bounded from above by the Moore-bound (see subparagraph 1.4.5). The node density of Γ is high if the above number lies close to the Moore-bound. In paragraph 2.3 a formal measure for node density of Γ will be defined. Furthermore, in that paragraph it is proved that a high node density of Γ is a precondition to construct extensible graphs with low diameters.
The shape of Δ_i is determined by the way it is cut out of Γ. The relation between the shape of Δ_i and its diameter will be elaborated in paragraph 2.5. In particular, this paragraph points out how extensible networks with low diameters should be constructed.

The fourth property, stated in the introduction, concerns the connectivity of Δ_i. The connectivity of Δ_i depends on four factors:

- The structure of Γ.
- The connectivity of Γ.
- The shape of Δ_i.
- The placement of Δ_i on Γ.

In general, the connectivity of Δ_i is neither bounded from below nor bounded from above by the connectivity of Γ.
If Δ_i is placed on a part of Γ of which all nodes have high mutual local connectivity relative to other parts of Γ, then the connectivity of Δ_i might be larger than the connectivity of Γ. For, Γ's connectivity is determined by its smallest separating set (see appendix A). This set is located at the parts of Γ of which the nodes have relatively low mutual local connectivities.
On the other hand, if Δ_i is placed on a smallest separating set of Γ then its connectivity is not greater than Γ's connectivity.

If Γ's connectivity is optimal, then it is always greater than the connectivity of Δ_i. For, the degrees of the border nodes of Δ_i are lower than $\deg(\Gamma)$, and the connectivity of Δ_i can never be greater than $\deg^-(\Delta_i)$. In the case of optimal connectivity of Γ, Δ_i's connectivity is less sensitive to the structure of Γ and the placement of Δ_i on Γ.

A high connectivity of Γ promotes a high connectivity of Δ_i. For this reason, we aim at optimization of Γ's connectivity. In paragraph 2.4 some sufficient conditions are deduced for optimal connectivity of Γ.
No general results about the precise connectivity of Δ_i shall be given in this dissertation. We confine ourselves to remarking that each Δ_i constructed by the method has a relatively high connectivity. It is a result of the prescription of Δ_i's shape by the requirements concerning the diameter and the routing function of Δ_i.

The fifth property, stated in the introduction, concerns the structure of Δ_i. The structure of Δ_i depends on two factors:

- The structure of Γ.
- The shape of Δ_i.

The regularity of Γ's structure is described by automorphism groups (see appendix A). It will appear that it is very convenient to choose Γ to be symmetric.
Classification based on automorphism groups does not apply to Δ_i. Any Δ_i is not even node-transitive, since the loose edges of Δ_i can not be matched onto the other edges of Δ_i. Therefore, in order to obtain a regular shape of Δ_i, we shall concentrate on the way Δ_i is cut out of Γ. This will be investigated in paragraph 2.5.

The sixth property, stated in the introduction, concerns the routing function. The routes via which an optimal routing function is allowed to route depend on the shape of Δ_i. Fortunately, the shapes allowing optimal routing functions to route inside extensible networks, will appear to guarantee maintenance of optimal routing functions under extension (see paragraph 2.5).

The seventh property, stated in the introduction, concerns the extension complexity. The extension complexity of Δ_i can not be made arbitrary small. Extension of Δ_i by too small a number of nodes may result in a loss of (a part of) the properties of Δ_i. Extension complexity will be investigated in paragraph 2.5.

From the above remarks it may be concluded that there should exists a close relationship between Γ and Δ_i. Only then the impositions made on Δ_i by Γ are strict. Which conditions should be satisfied by Γ and Δ_i in order to make their relationship as close as possible is made clear by theorem 2.2. First, some additional notions are defined.

The underlying graph Γ is called a *minimal* underlying graph of the sequence $S_E(\Delta_i) = (\Delta_0,\Delta_1,\Delta_2,\ldots)$ if Γ is a (not necessarily proper) subgraph of every

underlying graph of $S_E(\Delta_i)$. Every extension sequence has a uniquely defined minimal underlying graph. To prove this we first need a lemma.

(2.1) Lemma. Γ is a minimal underlying graph of an extension sequence $S_E(\Delta_i) = (\Delta_0,\Delta_1,\Delta_2,\ldots)$ if and only if for each node u in Γ there exists some natural number N such that $\forall\, i \geq N$: $u \in V(\Delta_i)$.

Proof.

- Suppose Γ is a minimal underlying graph of $S_E(\Delta_i)$.
  Suppose that there exists a node u in Γ such that

  $$\forall N \in \mathbb{N}\ \exists i \geq N: u \notin V(\Delta_i).$$

  It is to be proved that this condition implies a contradiction.
  If $i \geq N$, then $\Delta_i \supseteq \Delta_N$. Hence, if $i \geq N$ and $u \notin V(\Delta_i)$, then $u \notin V(\Delta_N)$.
  Since for any N there exists such an $i \geq N$ we obtain: $u \notin V(\Delta_i)$ for $i = 0,1,2,\ldots$.
  Then, $\Gamma - \{u\}$ is an underlying graph of the sequence $S_E(\Delta_i)$ contradicting with the minimality of Γ.

- Suppose for each node u in Γ there exists some N such that $\forall\, i \geq N$: $u \in V(\Delta_i)$.
  If Γ is not minimal then there exists some nonempty subgraph Λ of Γ such that $\Gamma - \Lambda$ is a minimal underlying graph of $S_E(\Delta_i)$. Then, for all $i = 0,1,2,\ldots$ and for all $w \in V(\Lambda)$: $w \notin V(\Delta_i)$, otherwise $\Gamma - \Lambda$ would not be an underlying graph of $S_E(\Delta_i)$. This is a contradiction. □

(2.2) Theorem. Every extension sequence $S_E(\Delta_i)$ has a unique minimal underlying graph Γ, defined by

$$\Gamma = \lim_{j \to \infty} \Delta_j.$$

Proof. For all $i \in \mathbb{N}$: $\Delta_i \subset \lim_{j \to \infty} \Delta_j$, since $\Delta_i \subset \Delta_{i+1}$.
Hence, $\Gamma = \lim_{j \to \infty} \Delta_j$ is an underlying graph of $S_E(\Delta_i)$.
Trivially, each node u in Γ lies in Δ_N for some N, and so u lies also in Δ_i for all $i \geq N$. Then, by lemma 2.1, Γ is a minimal underlying graph of $S_E(\Delta_i)$.
In order to prove that Γ is unique, suppose Γ' is a minimal underlying graph of $S_E(\Delta_i)$, different from Γ. Then, lemma 2.1 implies that for each node u in Γ' there exists an N such that

$$u \in V(\Delta_i) \quad \text{for all } i \geq N.$$

Hence, each node u of Γ' lies in $\lim_{j \to \infty} \Delta_j = \Gamma$,

which implies $\Gamma' \subseteq \Gamma$, and so, $\Gamma' = \Gamma$. □

If an underlying graph Γ used for the construction of an extension sequence $S_E(\Delta_i)$ is not the minimal underlying graph of $S_E(\Delta_i)$, then $S_E(\Delta_i)$ does not take full advantage of all features of Γ. Though this might result in extensible networks with favourable characteristics, we prefer to construct extension sequences by using underlying graphs which are minimal. Each extension sequence constructed in this dissertation is obtained from its minimal underlying graph.

2.3 Node density

This paragraph describes a measure for node density in an infinite connected graph, and proves some properties of the measure. The measure is called the *exponentiality* of a graph. Informally, it considers the function description of the number of nodes lying within a distance d from some node v as function of d. If this function is an exponential function, then the value of the base determines the exponentiality. If this function is not exponential, then the exponentiality is set equal to 1.
In the formal definition of the exponentiality of a graph, the sequence $D = (d_0,d_1,d_2,...)$ consists of all distances d at which the function is considered. D is assumed to be an infinite increasing sequence of positive integers.

(2.3) Definition. The *exponentiality* of an infinite connected graph Γ is defined by

$$\exp(\Gamma) := \sup\{ b \mid \exists v\in V(\Gamma)\ \exists c\in\mathbb{R}^+\ \exists D\ \forall d_j\in D: \sum_{i=0}^{d_j} |\Gamma_i(v)| \geq c.b^{d_j} \}.$$

Γ is *exponential* if $\exp(\Gamma)>1$ and *non-exponential* if $\exp(\Gamma)=1$. Γ will be called *super-exponential* if no supremum exists.

Clearly, superexponential graphs have not a bounded degree.
Whenever we speak in the rest of this dissertation of the exponentiality of a graph, $D = (d_0,d_1,d_2,...)$ will supposed to be a sequence of integers that, when substituted in the above definition, results in the supremum equal to the exponentiality of the graph. A special case occurs when $D = (1,2,3,...)$.

(2.4) Definition. The *uniform exponentiality* of an infinite connected graph Γ is defined by

$$\overline{\exp}(\Gamma) := \sup\{ b \mid \exists v\in V(\Gamma)\ \exists c\in\mathbb{R}^+\ \forall n\in\mathbb{N}: \sum_{i=0}^{n} |\Gamma_i(v)| \geq c.b^n \}.$$

Γ is *uniform exponential* if $\overline{\exp}(\Gamma)>1$ and *uniform non-exponential* if $\overline{\exp}(\Gamma)=1$. Γ will be called *uniform superexponential* if no supremum exists.

As a matter of fact, uniform exponentiality never exceeds the exponentiality of a graph.

The above definitions are not restricted to a specific node v. An arbitrary choice for v can be made, as the following lemma shows.

(2.5) Lemma. The value of $\exp(\Gamma)$ is independent of the choice for v in definition 2.3.

Proof. Let v be a node for which the supremum equals $\exp(\Gamma)$ in definition 2.3, let u be an arbitrary node of $V(\Gamma)$ and let $d=d(u,v)$. Define a new sequence $D':=(d'_0,d'_1,d'_2,\ldots)$ by $d'_j:=d_j+d$ and a new c' by $c':=c/b^d$.

Then $\sum_{i=0}^{d'_j} |\Gamma_i(u)| = \sum_{i=0}^{d_j+d} |\Gamma_i(u)| \geq \sum_{i=0}^{d_j} |\Gamma_i(v)| \geq c.b^{d_j} = c'.b^{d'_j}$

Hence, the replacement of v by u in definition 2.3 results in the same value of $\exp(\Gamma)$. □

An analogous lemma can be formulated for uniform exponentiality.

(2.6) Lemma. The value of $\overline{\exp}(\Gamma)$ is independent of the choice for v in definition 2.4.

Proof. Let v, u, d, c, and n be as in lemma 2.5, define n' by $n':=n$, and define c' by

$$c' := \min\left(\frac{c}{b^d} , \min_{1\leq r<d} \left\{ \frac{1}{b^r} \sum_{i=0}^{r} |\Gamma_i(u)| \right\} \right).$$

The definition of c' is more complicated than in lemma 2.5 to cope with the case $n'<d$. In a similar way to lemma 2.5 we deduce:

$$\sum_{i=0}^{n'} |\Gamma_i(u)| \geq c'.b^{n'} \quad \text{for } n'=1,2,\ldots.$$

□

If extensible networks have a high node density then the exponentiality of the corresponding minimal underlying graph is high. A high exponentiality might be an indication that the diameters of the extensible networks corresponding to the underlying graph are low. Might be, because the diameters also depend on another factor (see paragraph 2.5). Nevertheless, a high exponentiality of an underlying graph is of help to construct extensible networks with low diameters. In that perspective it is important to know some upper bounds to the exponentiality of a graph. The next theorem gives such an upper bound.

(2.7) Theorem. If Γ is an infinite graph of degree k (k>2), then $\exp(\Gamma) \leq k-1$.

Proof. Let v and D be as in definition 2.3. Define $(\Delta_0, \Delta_1, \Delta_2, \ldots)$ to be a sequence of induced subgraphs of Γ, for which

$$V(\Delta_i) := \bigcup_{i=0}^{d_i} \Gamma_i(v).$$

Since Γ is exponential there exist constants $c \in \mathbb{R}^+$ and $b \in (1,\infty)$ such that

$$|V(\Delta_i)| \geq c.b^{d_i} \quad \text{for all } i=0,1,2,\ldots.$$

$|V(\Delta_i)|$ is bounded from above by $n(k,d_i)$ (see subparagraph 1.4.5), giving

$$|V(\Delta_i)| \leq \frac{k(k-1)^{d_i}-2}{k-2}.$$

Combining these two expressions and rewriting gives:

$$c(k-2)-k(\frac{(k-1)}{b})^{d_i}+2.b^{-d_i} \leq 0$$

If $b>k-1$ then the left hand of this expression will be larger than 0 as i approaches ∞, which is a contradiction. Hence, the supremum of all b's satisfying this expression is not larger than $k-1$. □

The next theorem is useful for determining a graph's exponentiality. It gives a relationship between the exponentialities of two graphs.

(2.8) Theorem. If Γ and Λ are two infinite connected graphs, then $\Gamma \subset \Lambda$ implies

- $\exp(\Gamma) \leq \exp(\Lambda)$.
- $\overline{\exp}(\Gamma) \leq \overline{\exp}(\Lambda)$.

Proof. Let $\Gamma \subset \Lambda$ and v be a node in Γ, then for all $n \in \mathbb{N}$:

$$\bigcup_{i=0}^{n} \Gamma_i(v) \subseteq \bigcup_{i=0}^{n} \Lambda_i(v).$$

Then, definitions 2.3 and 2.4 directly imply the required results. □

We shall now determine the exponentialities of the minimal underlying graphs of a tree and a mesh, and of extensible networks considered in subparagraph 1.5.5. The first two networks, to be considered, are the 2-dimensional mesh, and the infinite tree of which all nodes have degree k. It is easily seen that the mesh is not exponential. The tree has uniform exponentiality $k-1$, which is optimal. We remark that halving such a tree by removing one edge does not affect its exponentiality. Hence, infinite trees of which all nodes have degree k except one 'root' node, which has degree $k-1$, have uniform exponentiality $k-1$. Many networks

are based on such trees, such as the X-tree, the hypertree and the pyramid (see subparagraph 1.5.5).

How about the exponentiality of the minimal underlying graphs of the extensible networks in subparagraph 1.5.5?

The butterfly is the first network to be considered. Figure 2.1 shows that the infinite tree of which all nodes have degree 3 is a subgraph of the butterfly.

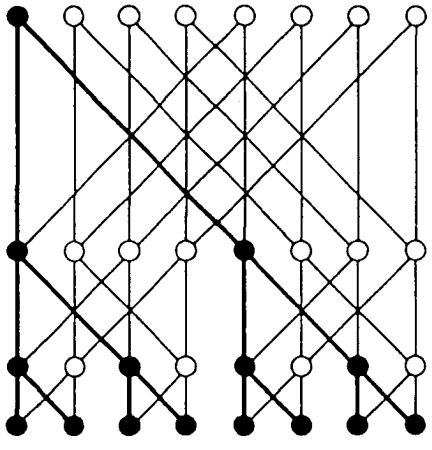


Figure 2.1 Embedding of tree in butterfly.

Theorem 2.8 implies that the uniform exponentiality of the butterfly is at least 2. Furthermore, by theorem 2.7, the uniform exponentiality of the butterfly is at most 3. The exact value of the butterfly's exponentiality is not known, but computation of the quotient

$$\sum_{i=0}^{n+1} |\Gamma_i(v)| \;/\; \sum_{i=0}^{n} |\Gamma_i(v)| \quad (\approx \overline{\exp}(\Gamma))$$

for the values $n=1,2,\ldots,8$ yields a sequence of numbers which seems to approach 2 from above. This indicates that the uniform exponentiality of the butterfly is 2.

The exponentiality of the cube-connected-lines is at most 2 (theorem 2.7). Considering the above quotient for $n=1,2,\ldots,7$, yields a sequence which seems to approach 1.7.

The X-trees in [DesPat] are all based on an infinite tree of which all nodes have degree 3 except the root node which has degree 2. All but the threaded and the double threaded tree have only extra edges between nodes at the same level. These edges don't affect the exponentiality. We conclude that the underlying graphs of all X-trees but the threaded and double threaded trees have uniform

exponentiality 2. This is not optimal since the degrees of X-trees are 4 or 5.

The underlying graphs of the hypertrees in [GooSeq] have uniform exponentiality 2, since they are based on the same tree as the X-tree, and augmented edges only connect nodes at the same level. Since the degree of a hypertree is 4 or 5, this is not optimal.

The pyramid in [FrHeHe] is based on an infinite tree of which all nodes have degree 5 except the root node which has degree 4. Mesh connections in the pyramid connect only nodes at the same tree level. This implies that the pyramid has uniform exponentiality 4. This is far from optimal because the maximum node degree in the pyramid is 9.

Computing the exponentiality of the underlying graph of the binary cluster tree in [HoKuRe] is a bit harder. We notice the following facts:

- Let a and b be nodes in a binary tree and C_a and C_b the corresponding clusters in the binary cluster tree. Then the minimal distance between C_a and C_b is at least $2.d(a,b)-1$.
- The number of nodes in a cluster is bounded by a constant.
- The original binary tree has uniform exponentiality 2.

From this we conclude that the underlying structure of the binary cluster tree has uniform exponentiality at most $2^{1/2}$ (doubling of distances causes the uniform exponentiality to be square rooted).

The next theorem shows the main reason for the introduction of exponentiality. It proves that extensible networks can only have logarithmic diameters when they have an exponential underlying graph.

(2.9) Theorem. Let $S_E(\Delta_i) = (\Delta_0,\Delta_1,\Delta_2,...)$ be an infinite extension sequence with underlying graph Γ. If $\mathrm{diam}(\Delta_i) \leq c.\log_2 |V(\Delta_i)|$ for some positive constant c, then Γ is exponential.

Proof. First, remove Δ_0 from $S_E(\Delta_i)$. As a result of this, the diameters of all elements in $S_E(\Delta_i)$ are positive.
Thereupon, consider the diameters of the remaining elements of $S_E(\Delta_i)$. If there are elements in $S_E(\Delta_i)$ with an equal diameter, then remove those elements except one from $S_E(\Delta_i)$. As a result of this, the diameters of all elements in $S_E(\Delta_i)$ are different.
Finally, renumber the remaining elements of $S_E(\Delta_i)$, such that they constitute a new extension sequence $(\Delta_0,\Delta_1,\Delta_2,...)$. This sequence has infinite length.
Let $d_i=\mathrm{diam}(\Delta_i)$, then $0<d_0<d_1<....$ Hence, $D=(d_0,d_1,d_2,...)$ is an infinite

increasing sequence of positive integers.
Let v be a node of $V(\Delta_0)$, then $v \in V(\Delta_i)$ and

$$\sum_{j=0}^{d_i} |\Gamma_j(v)| = \sum_{j=0}^{\mathrm{diam}(\Delta_i)} |\Gamma_j(v)| \geq |V(\Delta_i)| \geq (2^{1/c})^{\mathrm{diam}(\Delta_i)} = (2^{1/c})^{d_i}$$

for $i = 0,1,2,\ldots$. This implies that Γ is exponential. □

The reverse of theorem 2.9 is not true as the following example shows.

(2.10) Example. Let T_k be an infinite tree of which all nodes have degree k and let Δ_i be a subgraph of T_k, defined as a path in T_k of length i, beginning in some fixed node v of T_k. T_k is exponential but $\mathrm{diam}(\Delta_i)=i$, which is not logarithmic.

Up to now it remained unclear why uniform exponentiality was defined in addition to exponentiality. The reason is that exponentiality is not strict enough to our purposes. There may be huge 'gaps' between subsequent elements of the sequence D belonging to a graph. These gaps may even be so big that an exponential graph is uniform non-exponential. A graph belonging to such a sequence is not suitable as underlying graph for extensible networks with low diameter. The reason for this is three-fold.
First, there would be huge gaps between the diameters of subsequent elements of an extension sequence based on such an underlying graph. For, theorem 2.9 implies that if there are no large gaps between the diameters of the elements of the extension sequence, then there are no large gaps in D. Huge gaps between the diameters cause the extension complexity of each element of the sequence to be no longer a linear or sublinear function of the number of nodes.
Second, there would be much variety in the node density of extensible graphs. This would conflict with the desire of regularity of extensible graphs.
Third, uniform non-exponentiality troubles establishing of logarithmic diameters of extensible networks by a theorem (2.19) which shall be dealt with in paragraph 2.5.
So, when constructing underlying graphs we should always aim at uniform exponential graphs.

In order to ease the construction of uniform exponential graphs, it would be convenient to have some knowledge about the relationship between exponential and uniform exponential graphs. This relationship will be elucidated in the remaining part of this paragraph. It appears to be a very close relationship for node-transitive graphs. To prove this we first need a lemma. It gives a lower bound for $\overline{\exp}(\Gamma)$ expressed in terms of $\exp(\Gamma)$ and the gaps between the elements in D.

(2.11) Lemma. If $D = (d_0,d_1,d_2,\ldots)$ is a sequence belonging to an exponential graph Γ, and $a\in(1,\infty)$ is a constant bounding the elements of D according to

$$d_{j+1} < a.d_j \quad \text{for all } j=0,1,2,\ldots$$

then $\overline{exp}(\Gamma) \geq \exp(\Gamma)^{1/a}$

Proof. Define a number b by

$$b := \inf\{\, a \mid a > d_{j+1}/d_j \,\},$$

then, $d_{j+1} \leq b.d_j$ and $\forall t>1 \;\; \forall a > d_{j+1}/d_j \;\; \exists \epsilon>0\colon \; (t-\epsilon)^{1/b} = t^{1/a}$.
Setting $t=\exp(\Gamma)$ gives:

$$\forall a > d_{j+1}/d_j \;\; \exists \epsilon>0 \;\; \forall v\in V(\Gamma) \;\; \exists c\in\mathbb{R}^+ \;\; \sum_{i=0}^{d_j} |\Gamma_i(v)| \geq c.(t-\epsilon)^{d_j} \geq$$

$$c.((t-\epsilon)^{1/b})^{d_{j+1}} = c.(t^{1/a})^{d_{j+1}}.$$

Moreover, for all integers $m\in[0,d_{j+1}-d_j]$:

$$\sum_{i=0}^{d_j+m} |\Gamma_i(v)| \geq \sum_{i=0}^{d_j} |\Gamma_i(v)| \quad \text{and} \quad c.(t^{1/a})^{d_{j+1}} \geq c.(t^{1/a})^{d_j+m}.$$

This implies

$$\forall a > d_{j+1}/d_j \;\; \forall v\in V(\Gamma) \;\; \exists c\in\mathbb{R}^+ \;\; \forall m\in[0,d_{j+1}-d_j]\colon$$

$$\sum_{i=0}^{d_j+m} |\Gamma_i(v)| \geq c.(t^{1/a})^{d_j+m},$$

and so,

$$\forall a > d_{j+1}/d_j \;\; \forall v\in V(\Gamma) \;\; \exists c\in\mathbb{R}^+ \;\; \forall r\in\mathbb{Z}^+\colon \; \sum_{i=0}^{r} |\Gamma_i(v)| \geq c.(t^{1/a})^r. \qquad \square$$

This lemma shows that an exponential graph is uniform exponential if d_j is limited by an exponential function in j.

(2.12) Theorem. Every exponential node-transitive graph is uniform exponential.

Proof. Let Γ be an exponential node-transitive graph, then there exists a node $v\in V(\Gamma)$, there exist constants b, $c\in\mathbb{R}^+$, $b>1$, and there exists an infinite increasing sequence of positive integers $D=(d_0,d_1,\ldots)$ such that

$$\sum_{i=0}^{d_j} |\Gamma_i(v)| \geq c.b^{d_j} \quad \text{for all } d_j\in D.$$

If $c>1$, c will be decreased to 1 (notice that this does not cause b, v and D to change). In the rest of this proof we assume $c\leq 1$. Suppose D is maximal with

respect to b and c, i.e. for each $r \notin D$

$$\sum_{i=0}^{r} |\Gamma_i(v)| < c.b^r.$$

If D is not maximal, then add to D all numbers r not yet in D which do not satisfy the above equation. This results in a new infinite increasing sequence D of positive integers. The new D is maximal with respect to b and c.
If $D=(1,2,3,\ldots)$ then Γ is uniform exponential, in which case we are finished. So, suppose Γ is not uniform exponential, then lemma 2.11 implies

$$\forall a \in (1,\infty)\ \exists j \in \{1,2,3,\ldots\}:\ d_j \geq a.d_{j-1}.$$

Let $a=4$ and select a j for which $d_j \geq 4.d_{j-1}$. The expression $d_j \geq 4.d_{j-1}$ will be used extensively in the rest of the proof. We now distinguish four independent stages:

(1) Inasmuch as $d_j \geq 4.d_{j-1}$, we have $d_{j-1}+1 \neq d_j$ which implies $d_{j-1}+1 \notin D$. Hence,

$$|\Gamma_{d_{j-1}+1}(v)| = \sum_{i=0}^{d_{j-1}+1} |\Gamma_i(v)| - \sum_{i=0}^{d_{j-1}} |\Gamma_i(v)| < c.b^{d_{j-1}+1} - c.b^{d_{j-1}} = c.\frac{b-1}{b} b^{d_{j-1}+1}.$$

(2) Let $n \in \{0,1,\ldots,j-1\}$, then d_n is an element of D for which $d_n \leq d_{j-1}$. Then,

$$d_{j-1}+d_n+1 \leq 2.d_{j-1}+1 < 4.d_{j-1} \leq d_j,$$

implying

$$\forall n \in \{0,1,\ldots,j-1\}:\ d_j-(d_{j-1}+1) \neq d_n.$$

Hence, the number $d_j-(d_{j-1}+1)$ is not an element of D. This implies

$$\sum_{i=0}^{d_j-(d_{j-1}+1)} |\Gamma_i(v)| < c.b^{d_j-(d_{j-1}+1)}$$

(3) Let Δ be an induced subgraph of Γ with node set

$$V(\Delta) = \bigcup_{u \in U_j} \bigcup_{i=0}^{d_j-(d_{j-1}+1)} \Gamma_i(u), \quad \text{where } U_j = \Gamma_{d_{j-1}+1}(v).$$

Clearly, Δ is a subgraph of the ball with radius $d_j-(d_{j-1}+1)+d_{j-1}+1 = d_j$ around node v. Trivially,

$$d_j \geq 4.d_{j-1} \geq 2.d_{j-1}+2,$$

implying

$$d_j-(d_{j-1}+1) \geq d_{j-1}+1 \quad (j=1,2,\ldots).$$

Hence, each ball with radius $d_j-(d_{j-1}+1)$ around any node u of U_j *overlaps* U_j. Therefore, each node in U_j lies also in Δ. From this we conclude that Δ is the ball with radius d_j around node v.

(4) Node-transitivity of Γ implies that for all $i \geq 0$ and any node u in Γ:

$$|\Gamma_i(u)| = |\Gamma_i(v)|.$$

From these four results we obtain:

$$\sum_{i=0}^{d_j} |\Gamma_i(v)| \overset{(3)}{=} |V(\Delta)| \overset{(3)+(4)}{\leq} |\Gamma_{d_{j-1}+1}(v)| \cdot \sum_{i=0}^{d_j-(d_{j-1}+1)} |\Gamma_i(v)| \overset{(1)+(2)}{<}$$

$$c.\frac{b-1}{b}b^{d_{j-1}+1}.c.b^{d_j-(d_{j-1}+1)} = c^2\frac{b-1}{b}b^{d_j} < c.b^{d_j},$$

from which we conclude that $d_j \notin D$, which is a contradiction. Hence Γ is uniform exponential. □

Node-transitivity turns an exponential underlying graph automatically into a uniform exponential graph. We might wonder whether regularity of Γ has other positive side effects. Indeed, this will be the case.
In the next paragraph we will focus our attention onto the regularity and connectivity of Γ.

2.4 Regularity and connectivity

Distance-transitivity (see appendix A) is the highest degree of regularity considered in this dissertation, so it will be the first candidate structure. Macpherson gives in [Macphe] a characterization of infinite locally finite distance-transitive graphs (see appendix A), proving a conjecture of C.D. Godsil. To state Macpherson's theorem, it is necessary to first introduce the notion of standard graphs.

'Any *standard* graph arises in the following way. Let T be an infinite tree in which the vertices have alternating degree s and t+1. Then there is any standard graph G with parameters s and t whose vertices are the vertices of T with degree s, two vertices of G being connected if they lie at distance 2 in T. Informally, a standard graph is a tree-like form of complete graphs. Clearly, any standard graph is distance-transitive.' (Macpherson, [Macphe]). An infinite tree of which all nodes have degree k is standard with parameters s=k and t=1. We notice that the node-connectivity of a standard graph is 1.

(2.13) Theorem. (Macpherson, 1982). If Γ is an infinite locally finite distance-transitive graph, then it is standard.

Proof. See [Macphe]. □

This theorem shows that infinite locally finite distance-transitive graphs are exponential, except the one with degree 2. The low connectivity of standard graphs makes them unsuited for our purposes, however.
The next class of regular graphs is the class of symmetric graphs (see appendix A). An example of a symmetric graph is the infinite d-dimensional mesh. Symmetric graphs are a bit less regular than distance-transitive graphs. On the other hand, symmetric graphs can be highly connective, as will be shown. To study connectivity of symmetric graphs we consider a class of graphs to which symmetric graphs belong: the class of edge-transitive graphs.

Let's start with a theorem exploring *finite* edge-transitive graphs. It was independently proved by Mader and Watkins, and shows that finite connected edge-transitive graphs have optimal connectivity.

(2.14) Theorem. (Mader, 1970; Watkins, 1970). If Γ is finite, connected, edge-transitive and $|V(\Gamma)| \geq 2$, then $\kappa(\Gamma) = \deg^-(\Gamma)$.

Proof. See [Mader] or [Watkin]. □

This theorem is not valid for infinite locally finite edge-transitive graphs as is shown by Macpherson's theorem. It can be adapted by adding an extra condition.

(2.15) Theorem. If Γ is infinite locally finite, connected, edge-transitive and $\kappa_\infty(\Gamma) > \deg^-(\Gamma)$, then $\kappa(\Gamma) = \deg^-(\Gamma)$.

(For an explanation of $\kappa_\infty(\Gamma)$ see the description of *coherency* in appendix A). The proof of this theorem is analogous to Mader's proof in [Mader]. Mader subdivided his proof into two lemmas ([Mader; Lemma 1 and Lemma 2]) and the main theorem ([Mader, Satz 2]). The extra condition has only impact to his first lemma. Only the (small) change in the proof of this lemma will be outlined here. First, some additional notions are given.
Let S be a minimal separating set of Γ, then $\Gamma - S$ consists of the components $C_1, C_2, \ldots, C_i$ ($i \geq 2$). Let $\mu_\Gamma(S)$ and $\mu(\Gamma)$ be defined by

$$\mu_\Gamma(S) := \min_{j=1,\ldots,i} |V(C_j)|$$

$$\mu(\Gamma) := \min \{ \mu_\Gamma(S) \mid S \text{ is a minimal separating set of } \Gamma \}$$

Clearly, $|V(S)|=\kappa(\Gamma)$. Let S_0 be an S for which $\mu_\Gamma(S)$ is minimal, then $\mu_\Gamma(S_0)=\mu(\Gamma)$. Let C be a component of $\Gamma-S_0$ with the minimal number of nodes, i.e. $|V(C)|=\mu_\Gamma(S_0)$. Then C is called a *fragment.* $G(\Gamma)$ will denote the automorphism group of Γ.
The first lemma used by Mader in [Mader] states:

(2.16) Lemma. (Mader, [Mader; Lemma 1])
Let S be the separating set belonging to a fragment C of the finite graph Γ. Then, for each $\phi \in G(\Gamma)$ satisfying $\phi(S)\cap V(C)\neq\varnothing$ the condition $V(C)\subseteq\phi(S)$ holds.

In the proof of this lemma a set of nodes $P=V(C)\cap\phi(S)$ and a number $p=|P|$ are defined. The proof operates by proving that the condition $p<|V(C)|$ gives a contradiction. This procedure does not work when Γ is an infinite locally finite graph. Problems will arise when a finite minimal separating set S separates Γ into components which are all infinite. In that case $|V(C)|=\infty$, obstructing the deduction of a contradiction. This flaw can be repaired by adding an extra condition.

(2.17) Lemma. Let S be the separating set belonging to a fragment C of the infinite locally finite graph Γ for which $\kappa_\infty(\Gamma)>\deg^-(\Gamma)$. Then, for each $\phi\in G(\Gamma)$ satisfying $\phi(S)\cap V(C)\neq\varnothing$, the condition $V(C)\subseteq\phi(S)$ holds.

Proof. We first show that C is a finite component.
If C is an infinite component, then S separates T into sheer infinite components, since C is the smallest component of Γ. This implies $\kappa(\Gamma)=\kappa_\infty(\Gamma)$. Combining this with the added condition in lemma 2.17, we obtain $\kappa(\Gamma)>\deg^-(\Gamma)$, which is impossible.
The essential point is that C is a finite component. This enables the deduction of a contradiction in a similar way as in lemma 2.16. The rest of this proof proceeds in the same way as Mader's proof of lemma 2.16.† □

The proofs of Lemma 2 and Satz 2 in [Mader] don't have to be changed to apply to infinite locally finite graphs. Together with lemma 2.17 they constitute the proof of theorem 2.15.
In chapters 4 and 6 infinite locally finite symmetric graphs with high node-coherency will be constructed. Theorem 2.15 permits us to immediately decide to optimal node-connectivity of these graphs.

In the second stage of the construction method described in paragraph 2.2,

† Watkins' proof can not be adapted in this way. The main problem seems to be the use of $|R_1|$ and $|R_2|$ in the proof of [Watkin; lemma 3.3], which are infinite for infinite graphs, even if the extra condition is added.

extensible networks are cut out of an underlying graph. This implies that the networks have border nodes with loose edges. Hence, the cut process results in a decrease of the degree of these nodes. Consequently, the connectivity of the actual networks is not as high as the connectivity of the underlying graph. If the shape of the networks is selected with some care low local connectivity (see appendix A) can often be restricted to the border nodes. This will indeed be the case for the networks constructed in chapter 4 (and probably also the networks in chapter 8), as will be shown in chapter 5. In the next paragraph we give some attention to the shape of extensible networks in general.

2.5 Cutting out subgraphs of Γ

In paragraph 2.2 we pointed out that the shapes of extensible networks, determined by the way they are cut out of Γ, have impact on their following characteristics:

- Their diameters.
- Their routing functions.
- Their structure.
- Their connectivities.

In the current paragraph, the diameter, the routing functions, and the structure will be investigated in more detail. In particular, a recipe is given to cut the elements of an extension sequence $S_E(\Delta_i)$ out of Γ, such that the diameters of the elements of $S_E(\Delta_i)$ are low, the elements all have the same optimal routing function which only routes via paths inside each network, and each cutout is regular. Furthermore, it will be pointed out, how to construct a new extension sequence from $S_E(\Delta_i)$ with the same properties but lower or equal extension complexities.
No general results about the connectivity of extensible networks will be given in this chapter. It seems, however, that the shape of Δ_i determined by the first two characteristics, results in a reasonable connectivity. In chapter 5 it is shown that the local connectivities of a class of extensible networks, constructed by our method, are optimal.

The first characteristic to be dealt with is the diameter of an extensible network. Example 2.10 showed an exponential underlying graph of an extension sequence of which all elements have a non-logarithmic diameter. Apparently, exponentiality of an underlying graph is not a sufficient condition to impose a logarithmic diameter onto extensible networks. The reason for the high diameters of the networks in example 2.10 is their 'oblongness': their 'length' is much larger than their 'width'. A formal measure for the oblongness of an extensible network is

its so-called R/r-ratio.

(2.18) Definition. The *R/r-ratio* of a subgraph Δ of a graph Γ is defined to be the quotient R_Δ/r_Δ, where R_Δ and r_Δ are the radii of the circumscribed and the inscribed ball (see appendix A) respectively of Δ in Γ.

This concept plays an important role in this dissertation. A high value of the R/r-ratio of Δ indicates that Δ is oblong, and a low R/r-ratio indicates the opposite. Trivially, the R/r-ratio of a graph is at least 1. The following theorem states that if the subgraphs Δ_i of a uniform exponential graph are not too oblong, then their diameter is logarithmic.

(2.19) Theorem. Let Γ be an uniform exponential graph, then for all induced subgraphs Δ of Γ having a non-trivial inscribed ball:

$$\mathrm{diam}(\Delta) = O\left(R_\Delta/r_\Delta \cdot \log |V(\Delta)| \right).$$

Proof. Since Γ is uniform exponential there exists some $b>1$ for which $\underline{\exp}(\Gamma)>b$.
Suppose v_Δ is the centre node (see appendix A) of the inscribed ball of Δ, then

$$|V(\Delta)| \geq \sum_{i=0}^{r_\Delta} |\Gamma_i(v_\Delta)| \geq c.b^{r_\Delta} \quad \text{for some positive constant } c.$$

So, $r_\Delta \leq \log_b (|V(\Delta)| / c)$.
Moreover, $\mathrm{diam}(\Delta) \leq 2.R_\Delta \leq 2 . R_\Delta/r_\Delta . r_\Delta \leq 2 . R_\Delta/r_\Delta . \log_b (|V(\Delta)| / c)$,
which gives the required result. □

This theorem guarantees a low diameter of Δ when the R/r-ratio of Δ is low and the uniform exponentiality of Δ is high.
The important implication of this theorem is, that if the R/r-ratios of the elements of an extension sequence are bounded from above by a constant, then the diameters of the elements are logarithmic, provided that they have a uniform exponential underlying graph. That is, their diameters are not only logarithmic; they remain logarithmic under extension.

The reader might wonder: why not set an extensible network Δ_i equal to a ball with radius i in the underlying graph? The R/r-ratios of a ball is 1, which is optimal. The reason not to do this, is that the shape of Δ_i should also support ease of routing.
More in particular, any optimal routing function of Δ_i should only route via paths inside Δ_i, and it should be maintained under extension. These demands can both be satisfied by putting one single imposition onto the shapes of the elements of an extension sequence. Informally, this imposition states that there should be no

'holes' and 'inlets' in an extensible network. Holes and inlets are obstacles to easy routing of messages through the network. Lack of holes and inlets is denoted by *convexity.*

(2.20) Definition. An induced subgraph Δ of Γ is *convex* if for each two nodes $u,v \in V(\Delta)$ the condition $d(u,w)+d(w,v)=d(u,v)$ for any $w \in V(\Gamma)$ implies $w \in V(\Delta)$.

Stated in another way, Δ is convex if all shortest paths between u and v lie in Δ. Convexity as defined above is equivalent to the so-called 'g-convexity of a set of nodes in a graph', described in [FarJam] or 'd-convexity' in [SolChe].
Convexity is not an absolute characteristic of a subgraph, but depends on the graph of which it is a subgraph. A subgraph can be convex in one graph and non-convex in another. Extensible networks will be called *convex* in short whenever they are convex subgraphs of their minimal underlying graph.

Any optimal routing function r_Γ of a graph Γ can be used to route in a convex subgraph Δ of Γ. Since r_Γ routes via shortest paths between any two nodes in Γ, it also routes via shortest paths between any two nodes in Δ. Hence, each route established by r_Γ between any two nodes of Δ lies inside Δ. More formally:

(2.21) Theorem. Let Γ be a graph and Δ a convex induced subgraph of Γ. If r_Γ is an optimal routing function of Γ, then the following two conditions hold:

1. For all $u,v \in V(\Delta)$:

   $$R_{\Gamma i}(u,v) \subseteq V(\Delta),\ (i=0,1,2,\ldots,d(u,v)),$$

   i.e. r_Γ routes inside Δ.

2. r_Γ is an optimal routing function for Δ.

Proof. Let u and v be nodes of Δ.

1. Optimality of the routing function r_Γ implies that for all $w \in R_{\Gamma i}(u,v)$:

   $$d(u,w) = i \text{ and } d(w,v) = d(u,v)-i.$$

   That is, $d(u,w)+d(w,v) = d(u,v)$.
   This implies $w \in V(\Delta)$, since Δ is convex.
   So, $R_{\Gamma i}(u,v) \subseteq V(\Delta)$, $(i=0,1,2,\ldots,d(u,v))$.

2. If $u,v \in V(\Delta)$, then by optimality of r_Γ: $R_{\Gamma d_\Gamma(u,v)}(u,v)=\{v\}$. Since Δ is convex

$d_\Delta(u,v)=d_\Gamma(u,v)$. Hence, $R_{\Gamma d_\Delta(u,v)}(u,v)=\{v\}$, implying that r_Γ is an optimal routing function of Δ. □

The implication of this theorem is that if an extension sequence $S_E(\Delta_i)$ consists of sheer convex elements, then an optimal routing function of the minimal underlying graph Γ is an optimal routing function of any element Δ_i. Hence, optimal routing functions of Δ_i are maintained under extension of Δ_i.

In addition to simplifying routing, convexity may also reduce the R/r-ratios of the networks. Convexity guarantees networks to be free of holes and inlets. The radius of a network's inscribed ball is substantially reduced by holes and inlets. They hinder an easy conclusion of logarithmic diameter by use of theorem 2.19. Though convexity may be of help to achieve a low R/r-ratio, it does not guarantee a low R/r-ratio. The extensible networks in example 2.10 are convex but have a high R/r-ratio (which is even ∞).
Balls have a low R/r-ratio but they may be non-convex. In order to obtain convex networks with a low R/r-ratio, we shall 'convexify' balls. For this, we need an additional notion.

(2.22) Definition. The *convex hull* of a subgraph Δ of Γ, denoted by $[\Delta]$, is a convex subgraph of Γ for which

1. Δ is a subgraph of $[\Delta]$,
2. Δ is not a subgraph of any proper convex subgraph of $[\Delta]$.

Informally, the convex hull of a subgraph Δ of Γ is the smallest convex subgraph of Γ containing Δ.
A ball is convexified by determining its convex hull. The following theorem shows that convex hulls of balls have low R/r-ratios.

(2.23) Theorem. For every subgraph Δ of a graph Γ there exists a ball B in Γ such that $R_{[B]}/r_{[B]} \leq R_{[\Delta]}/r_{[\Delta]}$.

Proof. Let B be the inscribed ball of $[\Delta]$, then $r_{[B]} \geq r_B = r_{[\Delta]}$. Furthermore, $[B] \subseteq [\Delta]$ since $B \subseteq [\Delta]$, giving $R_{[B]} \leq R_{[\Delta]}$. □

This theorem does not imply that the R/r-ratio of the convex hull of *every* ball is smaller than the R/r-ratio of the convex hull of any subgraph Δ of Γ. Nevertheless, it indicates that convex hulls of balls are a good choice to obtain subgraphs of Γ with low R/r-ratios, and so, to obtain subgraphs of Γ with low diameters.

We are now in a position to cut extension sequences with favourable characteristics out of underlying graphs. Consider the sequence $(\Delta_0,\Delta_1,\Delta_2,\ldots)$ of which the

elements are defined by $\Delta_i := [B_i(v)]$, where $B_i(v)$ is a ball with radius i around some node v in the underlying graph. All elements of this sequence are convex and have relatively low R/r-ratios. This sequence is an extension sequence since $\Delta_i \subset \Delta_{i+1}$. We notice that defining extensible networks in this way guarantees the underlying graph to be minimal.
In order to be a suitable extension sequence, the R/r-ratios of all elements of $S_E(\Delta_i)$ must be bounded from above by a fixed small constant (≥ 1). Whether the R/r-ratios of convex hulls of balls are indeed bounded by a fixed small constant depends on the underlying graph Γ. The conditions which should be satisfied by Γ with respect to this, will be investigated later on in this paragraph.

Convex hulls of balls are regular cutouts of Γ, contributing to regularity of the elements in $S_E(\Delta_i)$. So, the third characteristic of Δ_i, regular structure, is automatically imposed by the other two characteristics of Δ_i, a low diameter and maintenance of optimal routing functions under extension.

How about the extension complexity of the elements of $S_E(\Delta_i)$ defined above?

The extension complexity of Δ_i depends on the R/r-ratio of Δ_i and the structure of Γ. No general results will be given for the extension complexity. Instead, we show a technique to reduce the extension complexity of extensible networks.
$C_E(\Delta_i)$ can be reduced in the following way:

- Find convex subgraphs Ψ of Γ for which

  $\exists i \in \mathbb{N}: \Delta_i \subset \Psi \subset \Delta_{i+1}$.

- Find sequences $\Psi_{i0}, \Psi_{i1}, \Psi_{i2}, \ldots, \Psi_{ir}$ among these subgraphs for which

  $\exists i \in \mathbb{N}: \Delta_i \subset \Psi_{i0} \subset \Psi_{i1} \subset \ldots \subset \Psi_{ir} \subset \Delta_{i+1}$.

- Add the longest sequences to $S_E(\Delta_i)$, giving a new sequence $S_E(\Delta_i')$.

Though this procedure reduces extension complexities, it may increase R/r-ratios. For, suppose R_i/r_i is the R/r-ratio of Δ_i. Then, the R/r-ratio of Δ_i' is at most R_{i+1}/r_i, which is of course larger. If, however, the increase of r_i as function of i is limited and the quotient R_i/r_i is bounded from above by a constant c, then the quotient R_{i+1}/r_i is also bounded from above by a constant. This can easily be seen. Suppose, $r_{i+1} \leq d.r_i$ for some constant d, then $R_{i+1}/r_i \leq c.d$.

Summarizing, suitable networks can be constructed by

1. Constructing an extension sequence $S_E(\Delta_i)$ consisting of convex hulls of balls.

2. Filling the gaps between the elements of $S_E(\Delta_i)$ by additional (finite) extension sequences.

Whether the extensible networks constructed by this procedure have good characteristics depends on the underlying graph used. The procedure guarantees that, given the underlying graph, the constructed extension sequences can hardly be improved. If the underlying graph is not selected with some care, the resulting extension sequences do not have desirable properties. Whether a particular underlying graph Γ enables construction of good extensions sequences depends on the R/r-ratios of the convex hulls of balls in Γ: these should be low.
It remains to be investigated on which factors the R/r-ratio of a ball's convex hull depends. Theorem 2.26 gives an answer to this. First, we define a number which tells something about the complexity of convex hulls of subgraphs of Γ (see [HarNie]).

(2.24) Definition. Let Γ be a graph, Δ be a subgraph of Γ, and let (Δ) be the induced subgraph of Γ with node set all nodes lying on a shortest path between two nodes of Δ. Let ${}^0\Delta := \Delta$ and ${}^{i+1}\Delta := ({}^i\Delta)$. The *geodetic iteration number* of Δ, denoted by $\mathrm{gin}(\Delta)$, is the minimum n for which ${}^{n+1}\Delta = {}^n\Delta$.

This definition may be clarified by remarking that 'geodesic' is just another word for 'shortest path' in graph theory. Trivially, $[\Delta] = {}^{\mathrm{gin}(\Delta)}\Delta$.

(2.25) Lemma. For every ball $B_r(v)$ with radius r around a node v in a graph Γ:

$$\forall k \in \mathbb{N} \quad \forall u \in {}^kB_r(v): \ d(u,v) \leq r.2^k.$$

Proof. (By induction on k).
The lemma is trivially true for $k=0$.
Suppose it is true for all $k \leq n$. Let w be a node in ${}^{n+1}B_r(v)$, then w lies on a geodesic between two nodes u_1 and u_2 in ${}^nB_r(v)$.
Then,

$$d(u_1,u_2) \leq d(u_1,v)+d(v,u_2) \leq r.2^{n+1}.$$

Hence,

$$d(u_1,w) \leq r.2^n \text{ or } d(u_2,w) \leq r.2^n.$$

From this we conclude

$$d(w,v) \leq d(w,u_i)+d(u_i,v) \leq r.2^{n+1} \quad (i=1 \text{ or } 2). \qquad \square$$

(2.26) Theorem. Let $\Delta = B_r(v)$ be a ball with radius r around a node v in a graph Γ such that $\mathrm{gin}(\Delta) < \infty$, then $R_{[\Delta]}/r_{[\Delta]} \leq 2^{\mathrm{gin}(\Delta)}$.

Proof. Lemma 2.21 implies

$$\forall u \in [\Delta]: \; d(u,v) \leq r.2^{gin(\Delta)} \leq r_{[\Delta]}.2^{gin(\Delta)}.$$

Hence, $R_{[\Delta]} \leq r_{[\Delta]}.2^{gin(\Delta)}$, giving the required result. □

From this theorem we conclude that the geodetic iteration number of balls in an infinite connected graph Γ should be small or at least be finite in order that Γ is suitable to be used as underlying graph for extensible networks. Unfortunately, it is not known which graphs guarantee the geodetic iteration number of their balls to be small, or even to be finite.

Two infinite classes of underlying graphs, to be constructed in chapters 4 and 6, contain only convex hulls of balls with finite geodetic iteration numbers. In chapter 6 the iteration numbers will even appear to be at most 1. This provides networks with low R/r-ratios.

2.6 The construction method

In this paragraph the preceding results are put systematically together, to constitute a recipe for the construction of efficient extensible networks. The recipe consists of three stages:

1. Find an infinite connected graph Γ which satisfies the following conditions:
   - $\deg(\Gamma) = \deg^-(\Gamma)$ is finite and low.
   - $\exp(\Gamma)$ lies close to $\deg(\Gamma) - 1$.
   - Γ is node-transitive and edge-transitive.
   - $\kappa_\infty(\Gamma) > \deg(\Gamma)$.
   - The geodetic iteration numbers of the balls in Γ are bounded from above by a fixed finite constant.
2. Cut subgraphs out of Γ in the following way:
   - Construct an extension sequence $S_E(\Delta_i)$ of Γ defined by

     $$\Delta_i := [B_i(v)],$$

     where $B_i(v)$ is a ball with radius i around some node v in Γ.
   - Consider $C_E(\Delta_i)$. If it is sufficiently low, then continue with step 3. Otherwise find extension sequences $(\Psi_{i0}, \Psi_{i1}, \ldots, \Psi_{ir})$ of convex elements satisfying

     $$\Delta_i \subset \Psi_{i0} \subset \Psi_{i1} \subset \ldots \subset \Psi_{ir} \subset \Delta_{i+1},$$

and r is maximal.
Add those sequences to $S_E(\Delta_i)$, resulting in a new $S_E(\Delta_i)$. If the resulting extension complexities are still too large, or if there are no sequences between consecutive elements of $S_E(\Delta_i)$, then try another Γ.

3. Construct an optimal routing function of Γ. It is also an optimal routing function of all elements of $S_E(\Delta_i)$.

For an explanation of this method we notice the following.

1. Concerning Γ:
   - A finite and low value of $\deg(\Gamma)$ guarantees a finite and low value of $\deg(\Delta_i)$.
   - The condition $\deg(\Gamma)=\deg^-(\Gamma)$ implies that all nodes of Γ have the same degree.
   - Exponentiality combined with node-transitivity of Γ guarantees Γ to be uniform exponential (see theorem 2.12).
   - High uniform exponentiality of Γ enables low diameters of Δ_i (see theorem 2.19).
   - Edge-transitivity of Γ combined with the condition $\kappa_\infty > \deg(\Gamma)=\deg^-(\Gamma)$ implies that $\kappa(\Gamma)=\lambda(\Gamma)=\deg(\Gamma)$ (see theorem 2.15). This guarantees a reasonable connectivity of subgraphs Δ_i of Γ, as constructed in stage 2.
   - Finite geodetic iteration numbers of balls in Γ guarantee the existence of finite subgraphs Δ_i of Γ, as constructed in stage 2 (see definition 2.24).
   - Boundedness of the finite geodetic iteration numbers guarantees the R/r-ratios of the Δ_i to be bounded from above by a constant (see theorem 2.26).

   Instead of assuming Γ to be node-transitive and edge-transitive, we assume it to be symmetric. Symmetry implies node-transitivity and edge-transitivity (but not the other way around).

2. Concerning Δ_i:
   - Convexity of Δ_i guarantees any optimal routing function of Γ to be an optimal routing function of any Δ_i (see theorem 2.21).
   - Being a convex hull of a ball guarantees convexity and a relatively low R/r-ratio of Δ_i (see theorem 2.23).

- A low R/r-ratio guarantees a low diameter in an underlying graph with a high uniform exponentiality (see theorem 2.19).
- Being a convex graph interjacent to two subsequent convex hulls of balls guarantees convexity and may guarantee a low R/r-ratio. Furthermore, it reduces extension complexity.

3. Concerning the routing function:
   - Optimality guarantees a low worst-case communication time in networks with low diameters.
   - Optimality guarantees a routing function of Γ to be an optimal routing function of all convex subgraphs of Γ (see theorem 2.21).

This completes the description of the construction method. It will be used to construct two infinite classes of extensible networks with desirable properties.
The first class, the tree-mesh, which will be described in chapters 4 and 5, does not perform much better with respect to diameters than the extensible networks described in subparagraph 1.5.5. However, a tree-mesh has meshes and trees as subgraphs, enabling the implementation of many parallel algorithms on it.
The second class, supersymmetric graphs which will be described in chapters 6, 7 and 8, consists of networks being almost optimal. Their uniform exponentialities lie very close to the upper bound stated in theorem 2.7 and the R/r-ratios of convex hulls of balls in them lie very close to 1. In addition, they are planar and have optimal extension complexity.

2.7 Relationship between extensible networks and chunks

We conclude this chapter with a discussion of the relationship between chunks of processors (see subparagraph 1.6.3) and extensible networks which can be constructed by our method.

Let Δ_i be an extensible network having the properties listed in the introduction of this chapter, and v be some fixed node in Δ_i. Suppose that the minimal underlying graph Γ of $S_E(\Delta_i)$ is node-transitive, and u is an arbitrary node of Γ. Then, there is always a placement of Δ_i on Γ such that u and v coincide. Consequently, Δ_i can be placed anywhere on Γ if Γ is node-transitive.
Let γ be a chunk in Δ_i having the same shape as Δ_j for some $j \leq i$. Then, γ can be placed anywhere on Γ. Consequently, γ can also be placed anywhere on Δ_i, provided $\gamma \subseteq \Delta_i$. There is at least one placement of γ on Δ_i for which the condition $\gamma \subseteq \Delta_i$ holds, since $\Delta_j \subseteq \Delta_i$ for $j \leq i$.
Clearly, a chunk defined in this way inherits all the properties of the members of $S_E(\Delta_i)$. In particular:

- The diameter of the chunk will be logarithmic, since the diameter of any Δ_j is logarithmic.
- The communication within a chunk is not disturbed by the communication of chunks disjunct from it, since any optimal routing function of Δ_j only routes via paths inside Δ_j.

In addition, a chunk has the following properties:

- The size of a chunk can be adapted dynamically by reducing it to Δ_{j-p} (for some p) or expanding it to Δ_{j+p} (for some p).
- The maximal sized chunk fits exactly in the network, because it has the shape of Δ_i.
- A job can be allocated anywhere on Δ_i, because its corresponding chunk can be placed anywhere in Δ_i.

Inspection of the list of demands in subparagraph 1.6.3 yields that demands no. 1 up to and including 6 are satisfied in this way. To satisfy demand no. 7, a strategy is needed to spread chunks homogeneously among the network. The development of such a strategy is beyond the scope of this dissertation.

3

A technological solution to space deficiencies

O God! I could be bounded in a nut-shell,
and count myself a king of infinite space,
were it not that I have bad dreams.

William Shakespeare (1564-1616)

3.1 Introduction

Paragraph 1.4.3 showed that in any communication model worst-case communication times of less than $O(n^{1/3})$ can not be combined with the constant space assumption for processors. In literature many attempts have been made to cope with this problem in network models. All approaches assume constant space for processors. As a consequence, resulting networks can not be extended unlimitedly, or the lengths of wires between processors are not constant. In models with variable length wires two approaches can be distinguished.

1. Wires become shorter as we move away from some central processor. This is the most common approach ([HorZor]). This restricts extensibility of the resulting networks, however. At some point wires can not be made shorter than a certain minimum length.
   Furthermore, the space problem for processors is not really solved in this way. In fact, it is even made worse.

2. Wires become longer as we move away from some central processor. At least this solution may create enough space for the processors. The philosophy behind this solution is that delay-times of wires are very small with respect to the times needed in processors to start up send and receive actions. According to this, it is the distance in graph-theoretical sense rather than the physical distance which determines communication times.
   The question remains, however, whether enough physical space is available for the wires, becoming longer and longer. If it is not, we throw out the baby

with the bath-water by replacing space deficiencies for processors by the even worse space deficiencies for wires.
Another thwart is that as delay times grow with the length of wires they will dominate the total time needed to send and receive. This situation inescapably occurs at some point. The resulting communication times are not the low ones we hoped to achieve.

We conclude that variation of wires lengths offers no satisfactory solution for space deficiencies in networks with low communication times. Nevertheless, in chapter 2 we made the first preparations to construct such networks. Were these preliminaries useless? Not at all, there exist other possibilities to achieve low communication times. For this, we should abandon the constant-space assumption for processors. It amounts to the use of processors ever decreasing in size as a computer is extended. This chapter deals with the consequences of this.

3.2 Extensibility, technological advances, and computation capacity

In the introduction of this chapter we pointed out that parallel computers with physical identical processors connected by extensible networks with logarithmic communication times are not feasible. Stated in another way, exponential interconnection networks cannot be applied in parallel computers which satisfy the constant space assumption and for which the (physical) lengths of edges do not exceed a fixed constant. The number of nodes increases as an exponential function of the graph-distance from some central node v. Inasmuch as the length of edges is bounded from above by a constant, the number of nodes also increases as an exponential function of the *physical* distance from v. This is incompatible with the constant space assumption, since we live in Euclidean space. The only alternative is to decrease the space available for a processor as we move away from v. It follows that processors can not be physical identical under the above assumptions.
In paragraph 1.3 we pleaded for identity of processors. There are no fundamental impediments to logical identity. Because of space deficiencies the situation is quite different for physical identity. This is the reason why we differentiated between the two kinds of identity.

In practice, it is impossible to fill in an infinite exponential network with processors ever decreasing in size as they lie more to the 'outside' of the network. Even when infinitely many processors were available, the state of technology at that moment would put a lower bound to the size of processors, permitting only a finite part of the network to be realized. Instead of infinite networks we shall consider finite subnetworks of them. They are the base of parallel extensible computers.

As more extensions are made, the size of the processors added to a computer should decrease. This is only possible when we make use of technological advances in the manufacture of chips. The ever-increasing number of electronic components per unit area enables the implementation of processors on ever-decreasing areas. It results in smaller processor chips or more processors per chip.

What are the consequences of physical non-identity of processors in a computer?

One of the most important incentives for continued advances in VLSI technology is gain in processor speed. Increasing integration density by a factor 100 enables a 10-fold increase of a processor's clock frequency in the ideal case ([Seitz]). Since a processor's clock frequency is directly related to its computational speed, higher integration density results in more powerful processors.
We can make use of this effect to obtain a more than proportional increase in computation capacity while extending a parallel computer. The extension of a computer with low communication times requires processors ever-decreasing in size, thus enabling the processors to work under increased clock frequency. This means that newly added processors have the potency to run faster than processors already existent in the computer.
Use of this effect results in the satisfaction of demand no. 6 in subparagraph 1.5.3 to extensible computers. It concerned the long life-cycle of extensible computers, and related with that the inability to keep these computers technologically up to date. When components used for extension are technologically as advanced as the core-computer purchased 10 years ago, the resulting computer runs of course faster, but not as fast as would be possible with more modern components. Consequently, a computer should be extended by the most up to date components deliverable. The effect described above is compatible with this need.
Whether smaller clock times are indeed put on newly added processors depends of the timing model used. If all processors work under a global clock no speed gain will be obtained. On the other hand local clocks enable optimal adaptation of clock frequency to a processor's power. We conclude that local timing enables us to reap the fruits of technological advances in extensible computers with low communication times. A more quantitative description of the additional increase in computation capacity of an extensible computer by extension with processors with higher clock frequencies shall be given in the next paragraph.
In paragraph 1.3 a global clock appeared to have major disadvantages for extensible computers. As described above other disadvantages emerge from a global clock. It affirms again our preference for local timing of processors.

As a consequence of the need to extend computers with processors ever

decreasing in size, the situation may arise that a customer buys a computer of which the processors needed for extension are not yet deliverable, because they still have to be designed and can not even be manufactured given the state of technology at that moment. The customer will be faced with the situation that his computer can only be extended when state of technology permits it. So, the rate at which technology is developing sets the pace at which a parallel computer with low communication times can be extended. The relation between these two rates will be investigated in paragraph 3.4.

In this paragraph we referred to advances in chip-technology. As a matter of fact, not only advances in VLSI should be considered but also developments in off-chip technology, such as chip-packages and pins. For, most space consumed by chips is on account of pins and packages, and to make it worse, off-chip technology is developing less quickly than VLSI. This can only be met partly by integrating more and more processors on chips becoming larger and larger.

3.3 Effective exponentiality

In the previous paragraph we pointed out that as a result of decrease of the size of processors their clock frequency may increase. Consequently, the computation capacity of an extensible computer with low communication times is super-linear in its number of processors. In this paragraph a measure will be developed to quantify the additional increase in computation capacity with respect to the number of processors. It will be denoted by the *effective exponentiality* of a network of processors. We shall deduce the relation between this measure and the exponentiality of a network.

First of all, we make three assumptions for the rest of this chapter.
The first assumption is that the technology used has the same characteristics as our current technology. Results in this chapter are based on the features of VLSI-technology, and in particular of CMOS-technology. The specific properties of other technologies, such as optics, are so divergent that phenomena we make use of do not occur, or occur at least at a different scale in those alternative technologies. For example, the relation between clock frequency and area of a processor in optical technology will be quite different from that in CMOS-technology.
The second assumption is that phenomena, such as the effect of a processor's area on its clock frequency, occur in the same degree in chips with higher integration densities. Hence, we extrapolate characteristics of the technology used to smaller scales.
The third assumption is that the networks in this chapter are *uniform* exponential. This does not impose a severe restriction to the validity of the analyses in this

chapter and simplifies them to a great extent. It is just the uniformity of exponentiality that makes networks interesting for application in parallel computers. All exponential networks constructed in this dissertation are uniform exponential.

To obtain a usable definition of effective exponentiality we put an imaginary case of filling an infinite exponential network with processors just small enough to fit into space. As is noted before, this is impossible in practice. Although only a finite part of the network is needed for a computer, the effective exponentiality is determined by all processors in an infinite network. Consequently, to define effective exponentiality we use a hypothetical infinite network entirely filled with processors.
The features of all processors in the infinite network are not known in advance. Yet to make meaningful predictions about those features, we use our second assumption. That is, the features of the processors are extrapolated from the features of processors customary today.
For the definition of effective exponentiality a unit of computation capacity is needed. The progress of computation capacity of one processor with respect to others is only a relative notion. We are not interested in an absolute notion of computation capacity. So, it makes not much difference what measure should be chosen as unit. We therefore define the unit of computation capacity in a network of processors to be the computation capacity of the *least* powerful processor. This processor will be denoted by v in this chapter.
In the following definitions $D=(d_0,d_1,\ldots)$ denotes an infinite increasing sequence of positive integers, and cap(u) denotes the computation capacity of the processor at node u with respect to the computation capacity of v (hence, for all nodes u: $cap(u)\geq cap(v)=1$).

(3.1) Definition. The *effective exponentiality* of a network N_Γ filled with processors and based on the graph Γ is defined by

$$\exp_c(N_\Gamma) := \sup\{\, b \mid \exists c\in\mathbb{R}^+ \ \exists D \ \forall d_j\in D: \sum_{i=0}^{d_j} \sum_{u\in\Gamma_i(v)} cap(u) \geq c.b^{d_j} \,\}.$$

(3.2) Definition. The *uniform effective exponentiality* of a network N_Γ filled with processors and based on the graph Γ is defined by

$$\overline{\exp_c}(N_\Gamma) := \sup\{\, b \mid \exists c\in\mathbb{R}^+ \ \forall n\in\mathbb{N}: \sum_{i=0}^{n} \sum_{u\in\Gamma_i(v)} cap(u) \geq c.b^{n} \,\}.$$

Effective exponentiality depends on the clock frequencies of the processors in a network with respect to each other, and of course of the exponentiality of the network itself. The clock frequency of a processor is inversely proportional to the

square root of its area ([Seitz]) in the ideal case, where the number of gate equivalents per processor is assumed to be constant. According to Seitz ([Seitz; p.1251]), to attain these figures in practice a few things go wrong, 'but', as he points out, 'these are only difficulties, not disasters'. There is no fundamental reason why these performances are impossible. Therefore, we shall assume that the above relation holds. We should realize, however, that these marks represent an ideal situation. The effective exponentialities, determined in this chapter, are upper bounds to expected increases of computation capacity of processors, when their sizes are reduced.

Inasmuch as the area of a processor is related to its clock frequency, it is also related to the effective exponentiality of a network. Hence, to determine the effective exponentiality of a network, the available area per processor should be known. For this reason we are interested in the sizes of processor's areas in this paragraph.
The size of the area available for a processor depends of the way it is embedded in Euclidean space. To obtain its value, it is important to know the dimension of the space the network is embedded in. In 2-dimensional space a processor will have to be content with less area than in 3-dimensional space.
The structure of the network determines the efficiency of the usage of the space available. So, it indirectly influences the size of the space available for processors. For reasons of simplicity we suppose that the structure of the network has no impact on the space available for processors. That is, the available space is distributed in the most efficient way among the processors.

We are now in the position to relate the effective exponentiality to the exponentiality of a network. First of all, we shall consider the 2-dimensional case.

3.3.1 Effective exponentiality in 2-dimensional space

The processors are placed onto the plane such that each of them lying at graph distance r_g from v lies on a circle with physical radius r_f around v. Since physical lengths of edges are constant we have

$$r_f \approx c_1 . r_g \quad \text{for some constant } c_1.$$

If the uniform exponentiality of the network is e, then the number of nodes at distance r_g from v is about $p_1(r_g).e^{r_g}$, where p_1 is a subexponential function. For simplicity we suppose that p_1 is a polynomial. Hence, about $p_1(r_g).e^{r_g}$ processors lie on the circumference of the circle with radius r_f. Consequently, the 'piece of circumference' available for each processor is

$$2\pi r_f / (p_1(r_g).e^{r_g}).$$

Assuming that a processor is a square and one of its sides lies against the circle, its area is about

$$4\pi^2 r_f^2 / (p_1^2(r_g).e^{2r_g}).$$

Suppose the clock frequency of this processor is at most K. We shall determine the maximal clock frequency of processors lying at distance r_g+1 from v. They lie with one of their sides against a circle with radius about r_f+c_1. Hence, the area of such a processor is about

$$4\pi^2(r_f+c_1)^2 / (p_1^2(r_g+1).e^{2(r_g+1)}).$$

Applying the inverse relation between the square root of a processor's area and its clock speed we obtain a clock frequency of these processors of at most

$$\{\frac{p_1^2(r_g+1)}{p_1^2(r_g)}.\frac{r_f^2}{(r_f+c_1)^2}.e^{2(r_g+1-r_g)}\}^{1/2}.K.$$

For large r_f (and large r_g) both quotients in this expression approximate 1, because p_1 is a polynomial. Hence, the clock frequency of processors at distance r_g+1 from v is at most e.K. We conclude that the clock frequency of a processor is multiplied by the factor e as we move one graph step away from v.
Inasmuch as the computation capacity of a processor is directly proportional to its clock frequency, the total computation capacity of the joint processors lying at graph-distance at most r from v is about

$$\sum_{i=0}^{r} p_1(i).e^i.e^i = \sum_{i=0}^{r} p_1(i).e^{2i}$$

(Notice that the computation capacity of v equals 1).

We conclude that for a network of processors with uniform exponentiality e, when optimally embedded in 2-dimensional space, the uniform effective exponentiality is about e^2.

3.3.2 Effective exponentiality in 3-dimensional space

In the 3-dimensional case we assume that the processors lying at graph-distance r_g from v are 'glued' onto a ball around v with physical radius r_f. That is, the surface of the ball lies locally parallel to the surface of the processor. This assumption seems to be a reasonable one. For, the height of a chip will not decrease much by technological developments, in contrary to the area of a chip. Since the area available for a chip on a ball decreases as r_f becomes larger, the most efficient way to embed a chip in space is the above one.
In a way similar to the 2-dimensional case we deduce that the area of a processor

lying at graph-distance r_g from v is about

$$4\pi r_f^2 / (p_1(r_g).e^{r_g}).$$

Analogously to the 2-dimensional case it can be deduced that the clock frequencies of processors are multiplied by a factor of at most $e^{1/2}$ when we move one graph step away from v in a network with uniform exponentiality e. Consequently, a network of processors with uniform exponentiality e has, when optimally embedded in 3-dimensional space, uniform effective exponentiality of about $e^{3/2}$.

In paragraph 3.3 we aimed to show that space deficiencies don't have sheer negative effects. We tried to make a virtue of necessity. Though shortage of space results in serious problems in the construction of computers, it also results in computation capacities which are larger than we can expect from only the exponentialities of underlying networks. The figures deduced in this paragraph should not be considered as absolute standards. They should rather give an impression how space deficiencies can be explored.
Although embeddings of networks in 2-dimensional space result in larger effective exponentialities than in 3-dimensional space, there is no reason to prefer embeddings in the plane. For, the higher effective exponentialities just show that embeddings in the plane suffer more seriously from space deficiencies. From that point of view embeddings in 3-dimensional space should rather be preferred.

Having described some virtues of space deficiencies, we shall investigate their influence upon the rate at which customers may wish to extend their computers.

3.4 Restrictions to the growth of extensible computers

In paragraph 3.2 we pointed out that the rate at which a computer can be extended is limited by the rate at which technology develops. In the current paragraph the relationship between these two rates will be considered in more detail. The rate of a computer's growth and of technological development will be expressed by functions. These functions are first related to the exponentiality and uniform exponentiality of the underlying graph at which a computer is based. Thereupon, they are related to each other. Exponentiality and effective exponentiality will appear to play hardly any role in the relations between these two rates. On the other hand R/r-ratios of the networks under consideration do have influence on the relations.

Before determining these relations, we first define the growth functions of extensible computers. There are two kinds of growth functions. The first one specifies a customer's needs for processors in his computer as a function of time, and

the second one specifies a customer's need for computation capacity as a function of time.

(3.3) Definition. The function $growth_p$: $\mathbb{R} \rightarrow \mathbb{N}$ specifies the number of processors as required by the owner of a computer as a function of time (in years).

(3.4) Definition. The function $growth_c$: $\mathbb{R} \rightarrow \mathbb{R}$ specifies the computation capacity as required by the owner of a computer as function of time (in years).

There is a clear difference between both functions. As described in the previous paragraph, processors may have a higher clock frequency when their area is smaller. In that case, a computer's computation capacity is super-linear in the number of processors. As a result, the two growth functions will behave differently.

We assume that both growth functions behave as an exponential function in time. This assumption seems to be a reasonable one for the following reasons. If a computer consists of, say, 10000 processors, then it is not very meaningful to extend it with 1, 10, 100 or even 1000 processors (see also demand no. 2 in subparagraph 1.5.3). In all cases the increase in computation capacity is very small in proportion to the computation capacity already existent. In practice extension of the capacity by these small numbers of processors will scarcely occur. We conclude that the size of extensions should be of the same order as the size of the computer. Furthermore, it is to be expected that extensions to a computer are made in a regular pattern, say every two or so years. The above is just a different way of stating that the number of processors and the capacity of a computer grow exponentially in time.

It is difficult to give an exact specification of growth functions. We supposed that they contain an exponential function as factor, but do they also contain, for example, polynomial factors? Since we only aim to gather a global impression of the relation between growth functions and the rate of development of technology, and since it is hard to specify all details of growth functions, we assume that the growth functions can be expressed as simple exponential functions. That is,

$$growth_p(t) = c_p . g_p^t,$$
$$growth_c(t) = c_c . g_c^t,$$

where c_p, c_c, g_p and g_c are positive constants (g_p and g_c larger than 1).

The function which describes the rate at which technology develops is defined as follows.

(3.5) Definition. The function tech: $\mathbb{R} \to \mathbb{R}$ specifies the number of electronic components per unit area as function of time (in years).

Up to now, this function has always been an exponential function. Hence we express it by

$$\text{tech}(t) = c_a.a^t.$$

where c_a and a are positive constants ($a>1$). The constant c_a depends on the technology used and the choice of the unit of area. Several values for a are given in literature. In [Leiser] and [Seitz] integration density is assumed to double every two years, which corresponds to $a=2^{1/2}$. In [Queyss] integration density quadruples every three years, resulting in $a=4^{1/3}$, and in [Uhr; p. 27] integration density is even assumed to double every year ($a=2$).

To obtain relations between the growth functions and the tech function, we first relate the growth functions to the exponentiality and the effective exponentiality of the network at the basis of the computer. It is assumed that a computer is made as large as is needed to meet its owner's demands for processing power. That is, the number of processors in the computer is larger than or equal to the number of processors required by its owner at that time. The same holds for the computation capacity of a computer.
Suppose the computer is based on an extensible network having an inscribed ball with radius r. The number of processors in this network depends on its R/r-ratio, the exponentiality e of the underlying network, and the radius r of the ball. This number is about

$$p_2(r).e^{\alpha r},$$

where $1 \leq \alpha \leq R/r$, and p_2 is a subexponential function. It follows that

$$c_p.g_p^t \leq p_2(r).e^{\alpha r}.$$

Similarly,

$$c_c.g_c^t \leq p_2(r).e^{\beta\alpha r},$$

where e^β is the effective exponentiality of the network (embedding the network in 2-dimensional space yields $\beta=2$ and embeddings in 3 dimensions yield $\beta=3/2$). The above relations imply

$$\textbf{(3.6)} \qquad r \geq \frac{1}{\alpha}.\log_e\frac{c_p}{p_2(r)} + \frac{1}{\alpha}.t.\log_e g_p,$$

and

$$\textbf{(3.7)} \qquad r \geq \frac{1}{\alpha.\beta}.\log_e\frac{c_c}{p_2(r)} + \frac{1}{\alpha.\beta}.t.\log_e g_c.$$

Concerning the relation between technological development and effective exponentiality of a network we make two suppositions.
First, the number of electronic components in a processor is assumed to be a constant n_p, independent of the processor's size. The ground for this assumption is that all processors are logically identical. This implies that they all have the same complexity without regard to their sizes.
Second, a computer can not be made larger than technology permits (notice that integration density increases as a computer's size increases). That is, the number of electronic components per unit area anywhere in the computer is less than or equal to integration density permitted by technology at that moment.

Integration density depends of the dimension of the space the computer is embedded in. We differentiate between the 2-dimensional and the 3-dimensional case.

3.4.1 Relations for 2-dimensional embeddings

From the area available for a processor (see paragraph 3.3.1) it is easily deduced that the number of electronic components per unit area in a processor, which lies at the inscribed ball of the network, equals

$$\frac{p_1^2(r).n_p}{4\pi^2 r_f^2}.e^{2r}.$$

Integration density enabled by technology at that moment must exceed this. Hence,

$$c_a.a^t \geq \frac{p_1^2(r).n_p}{4\pi^2 r_f^2}.e^{2r}.$$

Substituting $r=r_g$ and applying relations 3.6 and 3.7 respectively to the exponent of e results in:

$$a^t \geq \frac{p_1^2(r_g).n_p}{c_a 4\pi^2 r_f^2} \cdot \left(\frac{c_p}{p_2(r_g)}\right)^{\frac{2}{\alpha}} \cdot g_p^{\frac{2}{\alpha}.t},$$

and

$$a^t \geq \frac{p_1^2(r_g).n_p}{c_a 4\pi^2 r_f^2} \cdot \left(\frac{c_c}{p_2(r_g)}\right)^{\frac{2}{\alpha.\beta}} \cdot g_c^{\frac{2}{\alpha.\beta}.t} = \frac{p_1^2(r_g).n_p}{c_a 4\pi^2 r_f^2} \cdot \left(\frac{c_c}{p_2(r_g)}\right)^{\frac{1}{\alpha}} \cdot g_c^{\frac{1}{\alpha}.t},$$

because $\beta=2$ in 2-dimensional space.
Inasmuch as p_1 and p_2 are subexponential functions and $r=r_g$ is a sublinear function of t in both 3.6 and 3.7, the above expressions can be rewritten to

$$a^t \geq \frac{n_p.c_p^{\frac{2}{\alpha}}}{c_a 4\pi^2} \cdot q_{2p}(t) \cdot g_p^{\frac{2}{\alpha}.t},$$

and

$$a^t \geq \frac{n_p \cdot c_c^{\frac{1}{\alpha}}}{c_a 4\pi^2} \cdot q_{2c}(t) \cdot g_c^{\frac{1}{\alpha}.t},$$

where $q_{2p}(t) = p_1^2(r_g)/((p_2(r_g))^{\frac{2}{\alpha}}.r_f^2)$ and $q_{2c}(t) = p_1^2(r_g)/((p_2(r_g))^{\frac{1}{\alpha}}.r_f^2)$ are subexponential functions in t.

The above inequalities can only be valid for all t if

(3.8) $g_p \leq a^{\alpha/2}$

and

(3.9) $g_c \leq a^{\alpha}$.

The least favourable value of α for the growth factors is $\alpha=1$. The networks constructed in chapter 8 have an R/r-ratio lying very close to 1. The choice $\alpha=1$ is justifiable for them. For networks which have a larger R/r-ratio, and consequently, which may have a larger value of α, setting α to 1 results in upper bounds to the growth factors, which hold for the least favourable case.
Substituting $a=2^{1/2}$ and $\alpha=1$, the most conservative estimates, results for the 2-dimensional case in

$$g_p \leq 2^{1/4}$$

and

$$g_c \leq 2^{1/2}.$$

Hence, the rate of technological development permits doubling of the number of processors every 4 years in a 2-dimensional computer and doubling of the computation capacity every 2 years.

3.4.2 Relations for 3-dimensional embeddings

Analogously to the 2-dimensional case it is deduced that

$$c_a.a^t \geq \frac{p_1(r).n_p}{4\pi r_f^2}.e^r.$$

Substituting $r=r_g$ and applying relations 3.6 and 3.7 respectively to the exponent of e results in

$$a^t \geq \frac{p_1(r_g).n_p}{c_a 4\pi r_f^2} \cdot \left(\frac{c_p}{p_2(r_g)}\right)^{\frac{1}{\alpha}} \cdot g_p^{\frac{1}{\alpha}.t}$$

and

$$a^t \geq \frac{p_1(r_g).n_p}{c_a 4\pi r_f^2} \cdot \left(\frac{c_c}{p_2(r_g)}\right)^{\frac{2}{3\alpha}} \cdot g_c^{\frac{2}{3\alpha}.t}$$

(notice that $\beta = 3/2$ in this case).

These inequalities can be rewritten to

$$a^t \geq \frac{n_p \cdot c_p^{\frac{1}{\alpha}}}{c_a 4\pi} \cdot q_{3p}(t) \cdot g_p^{\frac{1}{\alpha}.t}$$

and

$$a^t \geq \frac{n_p \cdot c_c^{\frac{2}{3\alpha}}}{c_a 4\pi} \cdot q_{3c}(t) \cdot g_c^{\frac{2}{3\alpha}.t},$$

where $q_{3p}(t) = p_1(r_g)/((p_2(r_g))^{\frac{1}{\alpha}}.r_f^2)$ and $q_{3c}(t) = p_1(r_g)/((p_2(r_g))^{\frac{2}{3\alpha}}.r_f^2)$ are subexponential functions in t.

The above inequalities can only be valid for all t if

(3.10) $g_p \leq a^\alpha$

and

(3.11) $g_c \leq a^{3\alpha/2}$.

Once again setting $a = 2^{1/2}$ and $\alpha = 1$ gives

$g_p \leq 2^{1/2}$

and

$g_c \leq 2^{3/2}$.

Hence, the rate of technological development permits doubling of the number of processors every 2 years in a 3-dimensional computer and increasing its computation capacity to the eightfold every 2 years.

3.5 Concluding remarks

In this chapter we studied the consequences of space deficiencies. The only solution to these problems which maintains low communication times under extension, is a decrease of processor's sizes. From a technological point of view, this solution has positive as well as negative consequences.
Diminution of processors positively effects their computation capacity, since it enables higher clock frequencies. On the other hand, the need to scale down processors prevents a computer from being unrestrainedly extensible. The rate at

which extension can take place is limited by the rate of technological development.

The more space deficiencies come into prominence, the more their positive aspects become explicit. Consequently, embedding a network of processors in 2-dimensional Euclidean space results in a larger effective exponentiality than its embedding in 3-dimensional space. In both cases effective exponentiality exceeds the corresponding exponentiality. As a consequence, the computation capacity of a computer with low communication times is a super-linear function of its number of processors.

Concerning the technological limitation to a computer's growth, the value of the corresponding computer's exponentiality or effective exponentiality plays no role. Of more influence is the rate at which chips are scaled down, a network's R/r-ratio, and the dimension of the space the network is embedded in. Given a technological development factor of about $2^{1/2}$, which corresponds to doubling integration density every two years, and the least favourable R/r-ratio (i.e. 1), the number of processors in a 2-dimensional computer can be doubled every 4 years and the computation capacity can be doubled every 2 years. These marks are even more propitious for 3-dimensional computers. Doubling the number of processors every 2 years and increasing the computation capacity to the eightfold every 2 years is in prospect for these computers.

The marks determined in this chapter should not be considered as ultimate truths. They came into being by considerable simplifications, and rather should give a first global impression of the consequences of space deficiencies. The finality in regard to this matter has not yet been reached.

PART II

The tree-mesh

4

Construction of the tree-mesh and its convex subgraphs

You can't be suspicious of a tree,
or accuse a bird or a squirrel of subversion
or challenge the ideology of a violet.

Hal Borland (1964)

4.1 Introduction

In this chapter we use our construction method to design an infinite class of extensible networks, the tree-meshes. Tree-meshes are a crossbreeding between trees and meshes. They combine the optimal connectivity of the minimal underlying graphs of meshes with the logarithmic diameters of trees.

An important tool used in this chapter is the Cartesian graph product, which was defined by Sadibussy in [Sadibu].

(4.1) Definition. The Cartesian product of the graphs Γ_1 and Γ_2 is the graph $\Gamma_1 \times \Gamma_2$ with node set $V(\Gamma_1) \times V(\Gamma_2)$, and two nodes (u_1,u_2) and (v_1,v_2) being adjacent whenever $(u_1=v_1$ and $(u_2,v_2) \in E(\Gamma_2))$ or $(u_2=v_2$ and $(u_1,v_1) \in E(\Gamma_1))$.

An example of a Cartesian product graph is the $n \times m$-mesh: it is the Cartesian product of a linear array of n nodes and a linear array of m nodes.
We have the following convention for Cartesian products. Let Γ be the Cartesian product of $\Gamma_1,\Gamma_2,\ldots,\Gamma_m$ $(\Gamma=\Gamma_1 \times \Gamma_2 \times \ldots \Gamma_m)$, then any node u of Γ is represented by an m-tuple $(u_1,u_2,\ldots,u_m)$, where u_i is a node in Γ_i. The i^{th} coordinate of u will be denoted by $\text{coord}_i(u)$.

The Cartesian product is used in paragraph 4.2 to construct symmetric uniform exponential underlying graphs with fixed degree and optimal connectivity. Paragraph 4.3 describes the shapes of convex hulls of balls and convex graphs

between convex hulls of balls in these underlying graphs. Finally, in paragraph 4.4 optimal routing functions for the underlying graphs are constructed.

4.2 The underlying graph

In this paragraph infinite connected graphs, denoted by Γ, with the following properties are constructed:

- Γ is the Cartesian product of two infinite connected graphs.
- $\deg(\Gamma)=\deg^-(\Gamma)$ is a composite number.
- $\overline{\exp}(\Gamma)>1$.
- Γ is symmetric.
- $\kappa_\infty(\Gamma)=\infty$.
- The geodetic iteration numbers of the balls in Γ are bounded from above by a finite constant.

Γ is constructed by applying the Cartesian product operator to two infinite connected exponential graphs. We start with a theorem about the degree of Cartesian product graphs.

(4.2) Theorem. $\deg(\Gamma_1\times\Gamma_2) = \deg(\Gamma_1)+\deg(\Gamma_2)$.

Proof. Directly from definition 4.1. □

We next consider the exponentiality of Cartesian product graphs. It is determined according to the following theorem.

(4.3) Theorem. $\exp(\Gamma^{(1)}\times\Gamma^{(2)}) = \max(\exp(\Gamma^{(1)}),\exp(\Gamma^{(2)}))$.

Proof. Let $\Gamma=\Gamma^{(1)}\times\Gamma^{(2)}$, $t_m=\exp(\Gamma^{(m)})$ $(m=1,2)$, and $t=\max(t_1,t_2)$.

1. Γ has subgraphs isomorphic to $\Gamma^{(1)}$ and $\Gamma^{(2)}$ respectively, so by theorem 2.8: $\exp(\Gamma)\geq t$.
2. Since $\exp(\Gamma^{(m)})=t_m$ and $\overline{\exp}(\Gamma^{(m)})\leq t_m$ $(m=1,2)$:

$$\forall v_m\in V(\Gamma^{(m)})\ \ \forall\epsilon\in\mathbb{R}^+\ \ \exists c_m\in\mathbb{R}^+\ \forall r\in\mathbb{Z}^+\ \sum_{i=0}^{r}|\Gamma_i^{(m)}(v_m)| < c_m(t_m+\epsilon)^r.$$

Then,

$$\forall v=(v_1,v_2)\in V(\Gamma)\ \ \forall\epsilon\in\mathbb{R}^+\ \exists c_m\in\mathbb{R}^+\ \forall r\in\mathbb{Z}^+\ \sum_{i=0}^{r}|\Gamma_i(v)| =$$

$$\sum_{i=0}^{r}\Gamma_i^{(1)}(v_1)\ \sum_{j=0}^{r-i}\Gamma_j^{(2)}(v_2) < \sum_{i=0}^{r}c_1(t_1+\epsilon)^i\sum_{j=0}^{r-i}c_2(t_2+\epsilon)^j \leq$$

$$c_1c_2 \sum_{i=0}^{r} (t+\epsilon)^i \sum_{j=0}^{r-i} (t+\epsilon)^j = c_1c_2 \sum_{i=0}^{r} (t+\epsilon)^i \left(\frac{(t+\epsilon)^{r-i+1}-1}{t+\epsilon-1}\right) <$$

$$\frac{c_1c_2}{t+\epsilon-1} \sum_{i=0}^{r} (t+\epsilon)^{r+1} = \frac{c_1c_2}{t+\epsilon-1}(r+1).(t+\epsilon)^{r+1}.$$

Since this is valid for all $\epsilon>0$, $\exp(\Gamma)\leq t$.

From 1 and 2 we conclude that $\exp(\Gamma)=\max(t_1,t_2)$. □

The equivalent of this theorem for uniform exponentiality is:

(4.4) Theorem. $\overline{\exp}(\Gamma^{(1)}\times\Gamma^{(2)}) = \max(\overline{\exp}(\Gamma^{(1)},\overline{\exp}(\Gamma^{(2)})$.

Proof. Identical to the proof of theorem 4.3 with exp replaced by $\overline{\exp}$. □

Next, we consider the structure of Cartesian product graphs. For this we first introduce an additional notion:

(4.5) Definition. The m^{th} Cartesian power of a graph Γ, Γ^m, is defined by

$\Gamma^1 := \Gamma$
$\Gamma^m := \Gamma^{m-1}\times\Gamma$ if $m\geq 2$.

Symmetry of graphs is preserved under this kind of involution.

(4.6) Theorem. If Γ is symmetric, then Γ^m is symmetric ($m\geq 1$).

Proof. Trivially, Γ^1 is symmetric, so we have to prove the theorem for $m\geq 2$.
It is sufficient to prove that Γ^m is node-transitive and the vertex stabilizer G_v of any node $v\in V(\Gamma^m)$ is transitive on the set of neighbours of v (see appendix A).
Let $u=(u_1,\ldots,u_m)$ and $v=(v_1,\ldots,v_m)$ be two arbitrary nodes in Γ^m and $\alpha_1,\ldots,\alpha_m$ be automorphisms in $G(\Gamma)$ for which $\alpha_i(u_i)=v_i$.
Let $\beta_1,\ldots,\beta_m$ be defined by

$$\beta_i((x_1,\ldots,x_m))=(x_1,\ldots,x_{i-1},\alpha_i(x_i),x_{i+1},\ldots,x_m).$$

Then, β_i is an automorphism in $G(\Gamma^m)$. Hence, $\beta_1.\beta_2.\ldots.\beta_m$ is also an automorphism in $G(\Gamma^m)$ and it maps u to v. From this we conclude that Γ^m is node-transitive.
To prove symmetry of Γ^m, again let $v=(v_1,\ldots,v_m)$ be a node in Γ^m. Suppose $u=(u_1,\ldots,u_m)$ and $w=(w_1,\ldots,w_m)$ are neighbours of v ($u\neq w$). We have to find an automorphism $\alpha\in G(\Gamma^m)$ for which $\alpha(v)=v$ and $\alpha(u)=w$. Since u and w are neighbours of v they differ in one coordinate from v. So, u and w differ in at most two coordinates from each other. Suppose without loss of generality that $u_j=v_j$ and $w_j=v_j$ for $1\leq j\leq m-2$.

There are two main cases.

1. Nodes u and w differ in the same coordinate from v.
   Suppose without loss of generality that $u_m \neq v_m$ and $w_m \neq v_m$. Then, $u_{m-1} = w_{m-1} = v_{m-1}$.
   Since Γ is symmetric there exists an $\alpha' \in G(\Gamma)$ such that $\alpha'(v_m) = v_m$ and $\alpha'(u_m) = w_m$.
   Let $\alpha((x_1, x_2, \ldots, x_{m-1}, x_m)) = (x_1, x_2, \ldots, x_{m-1}, \alpha'(x_m))$.
   Since α preserves adjacencies, $\alpha \in G(\Gamma^m)$.

2. Nodes u and w differ in a distinct coordinate from v. Suppose without loss of generality that $u_{m-1} \neq v_{m-1}$ and $w_m \neq v_m$. Then, $u_m = v_m$ and $w_{m-1} = v_{m-1}$. Within this case there are two subcases:

   a. $v_m = v_{m-1}$.
      Let $\beta: V(\Gamma^m) \to V(\Gamma^m)$ be defined by

      $$\beta((x_1, \ldots, x_{m-2}, x_{m-1}, x_m)) = (x_1, \ldots, x_{m-2}, x_m, x_{m-1}).$$

      Since $(u_{m-1}, v_{m-1}) \in E(\Gamma)$ and $(w_m, v_{m-1}) = (w_m, v_m) \in E(\Gamma)$, and Γ is symmetric, there exists a $\gamma' \in G(\Gamma)$ for which

      $$\gamma'(v_{m-1}) = v_{m-1}$$
      $$\gamma'(u_{m-1}) = w_m$$

      Let $\gamma: V(\Gamma^m) \to V(\Gamma^m)$ be defined by

      $$\gamma((x_1, \ldots, x_{m-1}, x_m)) = (x_1, \ldots, x_{m-1}, \gamma'(x_m)).$$

      Since both β and γ preserve adjacencies $\beta, \gamma \in G(\Gamma^m)$. It is easily verified that $\gamma.\beta(v) = v$ and $\gamma.\beta(u) = w$. Hence, $\alpha = \gamma.\beta$ suffices.

   b. $v_m \neq v_{m-1}$.
      Since Γ is node transitive, there exists a $\delta' \in G(\Gamma)$ such that $\delta'(v_m) = v_{m-1}$.
      Define δ by

      $$\delta((x_1, \ldots, x_{m-1}, x_m)) = (x_1, \ldots, x_{m-1}, \delta'(x_m)).$$

      Since δ preserves adjacencies, $\delta \in G(\Gamma^m)$, so $\delta^{-1} \in G(\Gamma^m)$. After applying δ, the same situation as in 2a occurs.
      Let β be defined as in 2a and γ by

      $$\gamma((x_1, \ldots, x_{m-1}, x_m)) = (x_1, \ldots, x_{m-1}, \gamma'(x_m))$$

      where $\gamma'(v_{m-1}) = v_{m-1}$ and $\gamma'(u_{m-1}) = \delta'(w_m)$. Since $(w_m, v_m) \in E(\Gamma)$, $(\delta'(w_m), v_{m-1}) = (\delta'(w_m), \delta'(v_m)) \in E(\Gamma)$. So, γ' is a rotation around v_{m-1} preserving adjacencies. Application of $\delta^{-1}.\gamma.\beta.\delta$ to u gives:

$$\delta^{-1}.\gamma.\beta.\delta(u) = \delta^{-1}.\gamma.\beta.\delta((u_1,\ldots,u_{m-1},v_m)) =$$
$$\delta^{-1}.\gamma.\beta((u_1,\ldots,u_{m-1},\delta'(v_m))) =$$
$$\delta^{-1}.\gamma((u_1,\ldots,u_{m-2},\delta'(v_m),u_{m-1})) =$$
$$\delta^{-1}((u_1,\ldots,u_{m-2},\delta'(v_m),\gamma'(u_{m-1}))) =$$
$$\delta^{-1}((u_1,\ldots,u_{m-2},v_{m-1},\delta'(w_m))) =$$
$$(u_1,\ldots,u_{m-2},v_{m-1},w_m) = w.$$

In a similar way it is deduced that $\delta^{-1}.\gamma.\beta.\delta(v)=v$.
Hence, $\alpha=\delta^{-1}.\gamma.\beta.\delta$ suffices. □

Finally, we consider the connectivity of a Cartesian product graph. Cattermole proved the following theorem in [Catte2] (see also [Catte1]).

(4.7) Theorem. (Cattermole, 1972). $\kappa(\Gamma_1\times\Gamma_2) \geq \kappa(\Gamma_1)+\kappa(\Gamma_2)$.

Proof. See [Catte2]. □

This theorem implies optimal connectivity of $\Gamma_1\times\Gamma_2$ in case Γ_1 and Γ_2 are optimal connective, otherwise it gives a reasonable lower bound of $\kappa(\Gamma_1\times\Gamma_2)$. If $\Gamma_1\times\Gamma_2$ is edge-transitive we can do better, however. To see this we first prove a lemma.

(4.8) Lemma. If Γ_1 and Γ_2 are connected infinite locally finite graphs, then $\kappa_\infty(\Gamma_1\times\Gamma_2) = \infty$.

Proof. Suppose $\kappa_\infty(\Gamma_1\times\Gamma_2)$ is finite, then there is a minimum node set K with cardinality $\kappa_\infty(\Gamma_1\times\Gamma_2)$, dividing $\Gamma_1\times\Gamma_2$ into at least 2 components each being an infinite graph. Let R, S_1, S_2 and T be defined by

$$R = \{(v_1,v_2)\in V(\Gamma_1\times\Gamma_2) \mid \forall x\in V(\Gamma_1)\colon (x,v_2)\notin K \text{ and } \forall y\in V(\Gamma_2)\colon (v_1,y)\notin K\}$$
$$S_1 = \{(v_1,v_2)\in V(\Gamma_1\times\Gamma_2) \mid \exists x\in V(\Gamma_1)\colon (x,v_2)\in K \text{ and } \forall y\in V(\Gamma_2)\colon (v_1,y)\notin K\}$$
$$S_2 = \{(v_1,v_2)\in V(\Gamma_1\times\Gamma_2) \mid \forall x\in V(\Gamma_1)\colon (x,v_2)\notin K \text{ and } \exists y\in V(\Gamma_2)\colon (v_1,y)\in K\}$$
$$T = \{(v_1,v_2)\in V(\Gamma_1\times\Gamma_2)-K \mid \exists x\in V(\Gamma_1)\colon (x,v_2)\in K \text{ and } \exists y\in V(\Gamma_2)\colon (v_1,y)\in K\}$$

All nodes of R belong to the same component C of $\Gamma_1\times\Gamma_2-K$. For, if $u=(u_1,u_2)\in R$ and $v=(v_1,v_2)\in R$ and $u_i=u_{i,0},u_{i,1},\ldots,v_i$ is a path between u_i and v_i in Γ_i $(i=1,2)$, then

$$(u_1,u_2)=(u_{1,0},u_2),(u_{1,1},u_2),\ldots,(v_1,u_2)=(v_1,u_{2,0}),(v_1,u_{2,1}),\ldots,(v_1,v_2)$$

is a path from u to v not separated by K.
All nodes of S_1 and S_2 also belong to C. For, if $u=(u_1,u_2)\in S_1$ and $v=(v_1,v_2)\in S_2$

then $w=(u_1,v_2)\in R$. The path

$$u=(u_1,u_2)=(u_1,u_{2,0}),\ (u_1,u_{2,1}),\ldots,(u_1,v_2)=w$$

is not separated by K. There is also such a path from v to w. So, all nodes of S_1 and S_2 belong to C.
$|T|$ is finite since $|K|$ is finite ($|T|\leq|K|^2$). Since $V(\Gamma_1\times\Gamma_2)=R\cup S_1\cup S_2\cup T\cup K$, and $\Gamma_1\times\Gamma_2-K$ must contain at least two infinite components, we obtain a contradiction. □

Theorem 2.15 directly implies the following corollary.

(4.9) Corollary. If an infinite locally finite edge-transitive graph Γ is the Cartesian product of two connected infinite locally finite graphs, then

$$\kappa(\Gamma)=\lambda(\Gamma)=\deg^-(\Gamma).$$

From theorem 4.6 and corollary 4.9 we obtain

(4.10) Corollary. If Γ is a connected infinite locally finite symmetric graph, then

$$\kappa(\Gamma^m)=\lambda(\Gamma^m)=\deg(\Gamma^m) \quad \text{for } m\geq 2.$$

By using theorems 4.2, 4.4, 4.6, and corollary 4.10 we can now construct suitable underlying graphs. If Γ is a connected symmetric exponential infinite graph with bounded degree, but not necessarily with high connectivity, then Γ^m ($m\geq 2$) has the same qualifications and there is the added advantage of optimal connectivity. Let T_k be an infinite tree of which all nodes have degree k ($k\geq 3$), then T_k^m ($m\geq 2$) has the following properties:

1. $\deg(T_k^m)=k.m$ (theorem 4.2).
2. $\overline{\exp}(T_k^m)=k-1$ (theorem 4.4).
3. It is symmetric, because T_k is distance-transitive (theorem 4.6).
4. $\kappa(T_k^m)=\lambda(T_k^m)=\deg(T_k^m)$ (corollary 4.10).

T_k^m will be called an *m-dimensional tree-mesh*, not to be confused with the mesh of trees and the tree of meshes, both introduced by Leighton in [Leight]. Clearly, T_2^m is the m-dimensional mesh. Trivially, T_k is a subgraph of T_k^m. Since T_2 (the two-sided infinite linear array) is a subgraph of T_k, T_2^m is a subgraph of T_k^m.
A Part of the 2-dimensional tree-mesh T_3^2 is depicted in figure 4.1.
We notice that of all the optimal connective tree-meshes, T_k^2 has the largest exponentiality with respect to its degree.

Without proof we mention that the geodetic iteration number of balls in T_k^m is m,

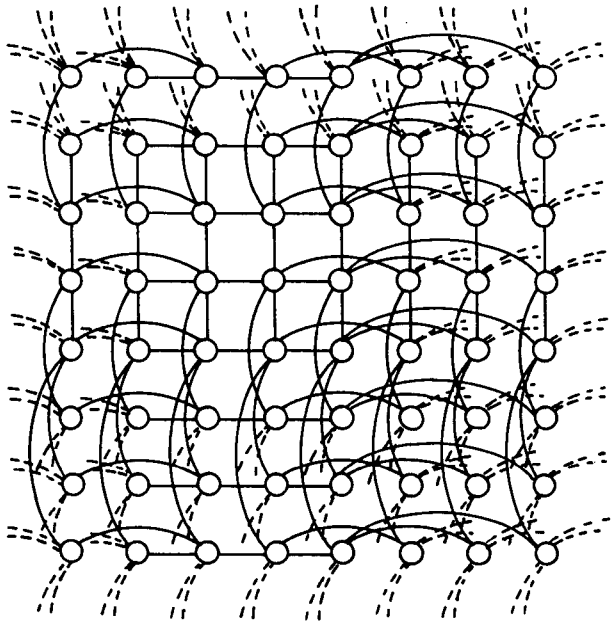


Figure 4.1 A piece of T_3^2.

giving an upper bound of 2^m to the R/r-ratio of convex hulls of balls in T_k^m. In the next paragraph, the R/r-ratio will appear to be much lower, namely m.

4.3 Cutting out suitable subgraphs

In this paragraph we determine suitable convex subgraphs of the tree-mesh. We have seen (theorem 2.23) that convex hulls of balls are a good choice for such subgraphs. Before determining the convex hull of a ball in a tree-mesh, we give two lemmas which simplify this job.
The next lemma facilitates the construction of convex subgraphs of the Cartesian product of two graphs.

(4.11) Lemma. Let Γ_1 and Γ_2 be graphs and Δ_1 and Δ_2 be subgraphs of Γ_1 and Γ_2 respectively, then $\Delta_1 \times \Delta_2$ is convex in $\Gamma_1 \times \Gamma_2$ if and only if Δ_1 is convex in Γ_1 and Δ_2 is convex in Γ_2.

Proof.

1. Let $\Delta_1 \times \Delta_2$ be convex in $\Gamma_1 \times \Gamma_2$. If Δ_1 is not convex in Γ_1 then there are two nodes $u,v \in V(\Delta_1)$ and a node $w \in V(\Gamma_1) - V(\Delta_1)$ such that w lies on a shortest path between u and v. Let $z \in V(\Delta_2)$, then node (w,z) lies on a shortest path

between nodes (u,z) and (v,z). However, $(u,z),(v,z) \in V(\Delta_1 \times \Delta_2)$ and $(w,z) \notin V(\Delta_1 \times \Delta_2)$, which is a contradiction. In the same way it is proved that Δ_2 is convex in Γ_2.

2. Let Δ_1 and Δ_2 be convex in Γ_1 and Γ_2 respectively. Suppose $\Delta_1 \times \Delta_2$ is not convex in $\Gamma_1 \times \Gamma_2$. Then there exist two nodes $u=(u_1,u_2)$ and $v=(v_1,v_2)$ in $\Gamma_1 \times \Gamma_2$ for which $u_i,v_i \in V(\Delta_i)$ $(i=1,2)$, and a shortest path P in $\Gamma_1 \times \Gamma_2$ between them not lying entirely in $\Delta_1 \times \Delta_2$. That is, there exists a $w=(w_1,w_2)$ on P such that $w \in V(\Gamma_1 \times \Gamma_2) - V(\Delta_1 \times \Delta_2)$. Then $w_1 \notin V(\Delta_1)$ or $w_2 \notin V(\Delta_2)$. Suppose $w_1 \notin V(\Delta_1)$. Let P be:

$$(u_1,u_2)=(u_{1,0},u_{2,0}), (u_{1,1},u_{2,1}),\ldots, (u_{1,r},u_{2,r})=(v_1,v_2).$$

Consider the sequence of nodes $P_1=u_{1,0},\ldots,u_{1,r}$, and call it P_1. Construct a new sequence P_1' of nodes in Δ_1, being equal to $P_1'=x_0,x_1,\ldots,x_{k_1}$, by joining identical nodes of P_1 to one node in P_1', and leaving the ordering intact. Then, P_1' is a path between u_1 and v_1 in Δ_1.
In the same way a path P_2' equal to $y_0,y_1,\ldots,y_{k_2}$ is constructed from the second coordinates of the nodes in P. P_2' is a path between u_2 and v_2 in Δ_2, and $r=k_1+k_2$.
Moreover, P_1' is a shortest path between u_1 and v_1, for otherwise there would be a path $Q=q_0,\ldots,q_s$ between u_1 and v_1 of length $s<k_1$. Then,

$$(q_0,y_0), (q_1,y_0),\ldots, (q_s,y_0), (q_s,y_1), (q_s,y_2), \ldots, (q_s,y_{k_2})$$

would be a path between u and v of length $s+k_2<r$, which is impossible.
Node w_1 lies on P_1', since w lies on P. However, Δ_1 is convex and $u_1,v_1 \in V(\Delta_1)$, so $w_1 \in V(\Delta_1)$, which is a contradiction.
In a similar way it is proved that $w_2 \notin V(\Delta_2)$ is impossible. So, $\Delta_1 \times \Delta_2$ is convex in $\Gamma_1 \times \Gamma_2$. □

(4.12) Lemma. Let Δ be a subgraph of $\Gamma=\Gamma_1 \times \Gamma_2$. If Δ is convex in Γ then there exist two convex subgraphs Δ_1 and Δ_2 of Γ_1 and Γ_2 respectively such that $\Delta=\Delta_1 \times \Delta_2$.

Proof. Let Δ be a convex subgraph of Γ and suppose there are no Δ_1 and Δ_2 for which $\Delta=\Delta_1 \times \Delta_2$. Then there exist nodes (u_1,u_2) and (v_1,v_2) in Δ such that $(u_1,v_2) \notin V(\Delta)$. This node lies on a shortest path between (u_1,u_2) and (v_1,v_2) in $\Gamma_1 \times \Gamma_2$, implying that Δ is not convex. From this contradiction we conclude that Δ can be written as $\Delta=\Delta_1 \times \Delta_2$. Then, lemma 4.11 implies that Δ_1 and Δ_2 are convex. □

We are now in a position to determine the shapes of convex hulls of balls in tree-meshes.
Let $\beta_i(v_0)$ be a ball with radius i around some node v_0 in T_k. Then, the m-tuple $v=(v_0,\ldots,v_0)$ is a node in T_k^m, and $\beta_i(v_0)^m$ is a subgraph of T_k^m. Clearly, the degree of $\beta_i(v_0)^m$ is k.m and its number of nodes is

$$|V(\beta_i(v_0)^m)| = (\frac{k(k-1)^i-2}{k-2})^m.$$

By lemma 4.11 $\beta_i(v_0)^m$ is convex in T_k^m, because β_i is convex in T_k. Moreover, $\beta_i(v_0)^m$ is the convex hull of its inscribed ball as is proven by the following theorem.

(4.13) Theorem. $\beta_i(v_0)^m$ is the convex hull of its inscribed ball.

Proof. It follows from the definition of Cartesian product that the ball with radius i around node v lies inside $\beta_i(v_0)^m$. Hence, the radius of the inscribed ball of $\beta_i(v_0)^m$ is at least i.
Let Θ be the convex hull of the inscribed ball of $\beta_i(v_0)^m$. Then, by lemma 4.12 there exist convex subgraphs $\Theta_1, \Theta_2,\ldots, \Theta_m$ of T_k such that $\Theta=\Theta_1\times\Theta_2\times\ldots\times\Theta_m$. Suppose Θ is a proper subgraph of $\beta_i(v_0)^m$, then Θ_j is a proper subgraph of $\beta_i(v_0)$ in T_k for at least one j in $\{1,\ldots,m\}$. Then, the nodes in $\beta_i(v_0)-\Theta_j$ lie in the outer layer of $\beta_i(v_0)$, otherwise Θ_j would not be a connected graph in T_k (notice that Θ_j is convex in T_k, and T_k is a tree). The distances of these nodes to v_0 are i. This implies that the inscribed ball of $\beta_i(v_0)^m$, which has radius at least i, is not a subgraph of Θ. This is a contradiction. Hence, Θ is not a proper subgraph of $\beta_i(v_0)^m$, implying $\Theta=\beta_i(v_0)^m$. □

In the proof of the previous theorem it was stated that the inscribed ball of $\beta_i(v_0)^m$ has radius at least i. Moreover, the radius of the inscribed ball is at most i. For, let u_0 be a node in T_k for which $d(u_0,v_0)=i+1$. Then, $(u_0,v_0,\ldots,v_0)$ lies at distance i+1 from v_0, but it does not lie in $\beta_i(v_0)^m$. We conclude that the radius of the inscribed ball of $\beta_i(v_0)^m$ is i.

To obtain the radius of the circumscribed ball of $\beta_i(v_0)^m$, we notice the following. If u_1 and v_1 are nodes in a graph Γ_1 and u_2 and v_2 are nodes in a graph Γ_2, then (u_1,u_2) and (v_1,v_2) are nodes in $\Gamma_1\times\Gamma_2$, and

$$d_{\Gamma_1\times\Gamma_2}((u_1,u_2),(v_1,v_2)) = d_{\Gamma_1}(u_1,v_1)+d_{\Gamma_2}(u_2,v_2)$$

(see also part 2 of the proof of lemma 4.11 for this). Hence, if w_0 is a node in T_k for which $d(v_0,w_0)=i$, then $d((v_0,\ldots,v_0),(w_0,\ldots,w_0))=m.i$. Furthermore, node $(w_0,\ldots,w_0)$ lies inside $\beta_i(v_0)^m$, which implies that the radius of the circumscribed

ball of $\beta_i(v_0)^m$ is at least m.i.
If there exists a node in $\beta_i(v_0)^m$ at distance greater than m.i from v, then there exists a node in $\beta_i(v_0)$ at distance greater than i from v_0 by the above formula describing $d_{\Gamma_1 \times \Gamma_2}$, which is impossible. From the above discussion we conclude that the radius of the circumscribed ball of $\beta_i(v_0)^m$ is m.i. This implies that the R/r-ratio of $\beta_i(v_0)^m$ is m.i / i = m. Then by theorem 2.19 the diameter of $\beta_i(v_0)^m$ is logarithmic. In fact,

$$\operatorname{diam}(\beta_i(v_0)^m) = 2.m.i.$$

The next stage in the construction of suitable subgraphs of T_k^m is to find convex subgraphs of T_k^m interjacent between two consecutive members of $S_E(\beta_i(v_0)^m)$. For this we use the following theorem.

(4.14) Theorem. Let $S_E(\alpha_i)$ be an extension sequence of convex networks with underlying graph Γ and $C_E(\alpha_i) \leq O(|V(\alpha_i)|)$. If $(\Delta_0, \Delta_1, \Delta_2, \ldots)$ is a sequence of subgraphs of Γ^m defined by

$$\Delta_i := \alpha_{\lfloor i/m \rfloor}^{m-(i \bmod m)} \times \alpha_{\lfloor i/m \rfloor + 1}^{i \bmod m} \quad (m \in \mathbb{N}),$$

then

1. $S_E(\Delta_i)$ is an extension sequence of which the elements are convex in Γ^m.
2. Γ^m is the minimal underlying graph of $S_E(\Delta_i)$.
3. $C_E(\Delta_i) = C_E(\alpha_{\lfloor i/m \rfloor}).O(|V(\Delta_i)|^{(m-1)/m})$.

Proof.

1. It is easily verified that Δ_i is a proper subgraph of Δ_{i+1} (consider $i \equiv m-1 \pmod m$ as a separate case). Convexity of Δ_i follows directly from lemma 4.11.
2. Consider the sequence $(\Delta_0, \Delta_m, \Delta_{2m}, \ldots)$. If there is a node $v \in \Gamma^m$ such that $v = (v_1, \ldots, v_m)$ is not in $\Delta_{im} = \alpha_i^m$ for all i, then there exists a $j \in [1,m]$ such that for all i: $v_j \notin \alpha_i$, implying non-minimality of Γ with respect to the α_i. So, Γ^m is the minimal underlying graph of $(\Delta_0, \Delta_m, \Delta_{2m}, \Delta_{3m}, \ldots)$ and hence of $(\Delta_0, \Delta_1, \Delta_2, \ldots)$.
3. $C_E(\Delta_i) = |V(\Delta_{i+1})| - |V(\Delta_i)| =$

$$|V(\alpha_{\lfloor (i+1)/m \rfloor})|^{m-(i+1 \bmod m)}.|V(\alpha_{\lfloor (i+1)/m \rfloor + 1})|^{i+1 \bmod m} -$$
$$|V(\alpha_{\lfloor i/m \rfloor})|^{m-(i \bmod m)}.|V(\alpha_{\lfloor i/m \rfloor + 1})|^{i \bmod m} =$$
$$C_E(\alpha_{\lfloor i/m \rfloor}).|V(\alpha_{\lfloor i/m \rfloor})|^{m-1-(i \bmod m)}.|V(\alpha_{\lfloor i/m \rfloor + 1})|^{i \bmod m}$$

(Consider $i \equiv m-1 \pmod m$ as a separate case to verify this).

Since $C_E(\alpha_{\lfloor i/m \rfloor}) \leq O(|V(\alpha_{\lfloor i/m \rfloor})|)$ we have $|V(\alpha_{\lfloor i/m \rfloor+1})| = O(|V(\alpha_{\lfloor i/m \rfloor})|)$.

Moreover,

$$|V(\alpha_{\lfloor i/m \rfloor})| \leq |V(\Delta_i)^{1/m}|,$$

which implies

$$|V(\alpha_{\lfloor i/m \rfloor+1})| = O(|V(\Delta_i)|^{1/m}.$$

Substituting $|V(\alpha_{\lfloor i/m \rfloor})|$ and $|V(\alpha_{\lfloor i/m \rfloor+1})|$ in the above expression of $C_E(\Delta_i)$ gives the required result. □

Consider any connective subgraph Δ of T_k. Since there exists only one shortest path between any two nodes in Δ and since Δ is connective, this path lies in Δ. Hence, Δ is convex in T_k. This implies the existence of extension sequences $S_E(\alpha_i)$ consisting of convex subgraphs of T_k for which $C_E(\alpha_i)=1$. In fact, $\beta_i(v_0)$ can be extended in $k.(k-1)^i$ steps of 1 node to $\beta_{i+1}(v_0)$, all the intermediate subgraphs being convex. Let $(\gamma_{i,0},\gamma_{i,1},\ldots,\gamma_{i,k(k-1)^i})$ be the corresponding extension sequence, where $\beta_i(v_0)=\gamma_{i,0}$ and $\beta_{i+1}(v_0)=\gamma_{i,k(k-1)^i}$. Putting the extension sequences $(\gamma_{0,0},\gamma_{0,1},\ldots,\gamma_{0,k})$, $(\gamma_{1,0},\gamma_{1,1},\ldots,\gamma_{1,k(k-1)})$,..., in succession and adjoining them (notice that $\gamma_{i,k(k-1)^i}=\gamma_{i+1,0}$) results in a new extension sequence with extension complexity 1, the elements of which are convex subgraphs of T_k. Let $S_E(\alpha_i)$ be this sequence, α_i being equal to $\gamma_{p,q}$, where

$$i = 1+k\frac{(k-1)^p-1}{k-2}+q.$$

Then, by theorem 4.14 the sequence $(\Delta_0,\Delta_1,\ldots)$, defined as in the theorem, is an extension sequence of convex elements with minimal underlying graph T_k^m and extension complexity

$$C_E(\Delta_i) = C_E(\alpha_{\lfloor i/m \rfloor}).O(|V(\Delta_i)|^{(m-1)/m}) = O(|V(\Delta_i)|^{(m-1)/m}).$$

For $m=2$ we obtain $C_E(\Delta_i) = O(|V(\Delta_i)|^{1/2})$, and a larger extension complexity for larger m. We conclude that, with respect to the extension complexity, T_k^2 is an optimal tree-mesh. This is a pleasant coincidence with the fact that T_k^m has the largest exponentiality with respect to its degree for $m=2$ (see paragraph 4.2).

The R/r-ratio of Δ_i ($i \geq m$) is at most

$$m(\lfloor i/m \rfloor+1)/\lfloor i/m \rfloor = m(1+1/\lfloor i/m \rfloor) \approx m \quad \text{for large i.}$$

Hence, the diameter of Δ_i is logarithmic.

Finally, we consider the connectivity of Δ_i. Since $\kappa(\alpha_i)=1$, by theorem 4.7 the connectivity of Δ_i is at least m. Moreover, the lowest degree in α_i is 1 (a leaf

node), so the lowest degree in Δ_i is m (a 'corner' node). Thus, $\kappa(\Delta_i)=m$.
The rather large difference between the connectivities of Δ_i and T_k^m is not as serious as it seems to be. The lowest local connectivity (see appendix A) is only confined to the borders of Δ_i. Local connectivity between nodes in the interior of Δ_i, i.e. the nodes at distance at least 1 from any border of Δ_i, is k.m. We shall prove this in chapter 5. Moreover, it is proved that local connectivity between any two nodes in Δ_i is optimal.

The tree-mesh is not brand new. As far as I know it appeared once in literature, namely in the book by Akl [Akl; section 2.2], where it is used for a parallel enumeration sorting algorithm. Akl uses an induced subgraph of T_3^2 with a shape differing from Δ_i for this. Akl's network is the Cartesian product of a complete binary tree with itself. In contrast to α_i in T_k, the complete binary tree has two kinds of border nodes, i.e. leaf nodes and the root. Consequently, compared to Δ_i in T_3^2, there is more variety in the nodes of Akl's network. From this cause the latter is a bit less regular than our subgraphs of T_3^2. Nevertheless, it is convex, local connectivity between any two of its nodes is optimal, and its diameter is only slightly larger than that of our networks. Akl applied his network not because of its extensibility but simply because of its low diameter and its ability to route data efficiently in the described sorting algorithm.

4.4 An optimal routing function for the tree-mesh

In the last stage of the construction method we construct an optimal routing function of the tree-mesh. In order to obtain a routing function we first need to have a labeling of the tree-mesh to address the nodes. A labeling of a network must satisfy the following requirements, in order that we can use it in an efficient way to address processors:

1. Each node has a unique label.
2. Given two labels of the network, it is easy to establish whether the nodes corresponding to them are connected by an edge.
3. The labeling is efficient, i.e. the memory space required to represent each label is of the same order as the space needed to represent the number of nodes in the network.
4. Given two labels it is easy to find a shortest path between the nodes corresponding to them.

In order to obtain an efficient labeling of a tree-mesh we shall first design an efficient labeling for T_k. Having done this, an efficient labeling for T_k^m is easily obtained by assigning m-tuples to the nodes in accordance with the conventions in

definition 4.1.

In order to obtain efficient labelings of infinite trees, we use a method, described in [DoRaSl; Theorem 8], to label finite trees. To construct an efficient labeling of T_k, T_k is first split in k infinite subtrees by removing one of its nodes. Call this node v, and let S_1, S_2,..., S_k be the resulting subtrees. They are k-ary trees, i.e. the degree of the root is $k-1$, and the degree of all other nodes is k. Thereupon, we label each S_i in the usual way, i.e. the label of the root of S_i is 1 and if the label of a node in S_i is r then its childs are labeled by the labels $k(r-1)+2$, $k(r-1)+3$,..., $k(r-1)+k+1$. We denote the label of a node x in S_j by $f_j(x)$. A labeling g on T_k is defined by

$$g(x) := k.f_j(x)-(j-1) \quad \text{if } x \in V(S_j).$$

$$g(v) := 0.$$

Figure 4.2 shows such a labeling for T_4.

It is easily seen that this is a unique labeling of T_k, since f_j is a unique labeling of S_j. So, requirement 1 is satisfied.
Requirement 2 is also satisfied, because it is satisfied for S_j.
Concerning requirement 3, the labeling of S_j is efficient since for a complete finite k-ary subtree of S_j with r levels, all labels in the range 1,2,..., $(k^{r+1}-1)/(k-1)$ are used. That is, there are no 'holes' in the labeling of S_j. From this it is easily proved that the labeling g of T_k has neither holes. There is no labeling being more efficient than labeling g. So, the labeling g satisfies requirement 3.
In order to prove that the labeling g also meets requirement 4 we design an optimal routing function for T_k^m. First, an optimal routing function is designed for routing in T_k.

To obtain an optimal routing function for routing between two arbitrary nodes u and w in T_k, we distinguish two cases:

1. The nodes u and w lie in the same subtree S_j.
2. Otherwise, i.e. each path from u to w passes through v.

It is easy to distinguish between these cases, for, u and w lie in the same subtree if and only if $g(u) \equiv g(w) \pmod k$.
The first case amounts to routing with an optimal routing function in a subtree S_j. It is well-known how this can be done.
If the second case occurs, we only need to route from u to the root of the subtree to which u belongs. From the root we route via v to the root of the subtree to which w belongs. From that point we route to w.

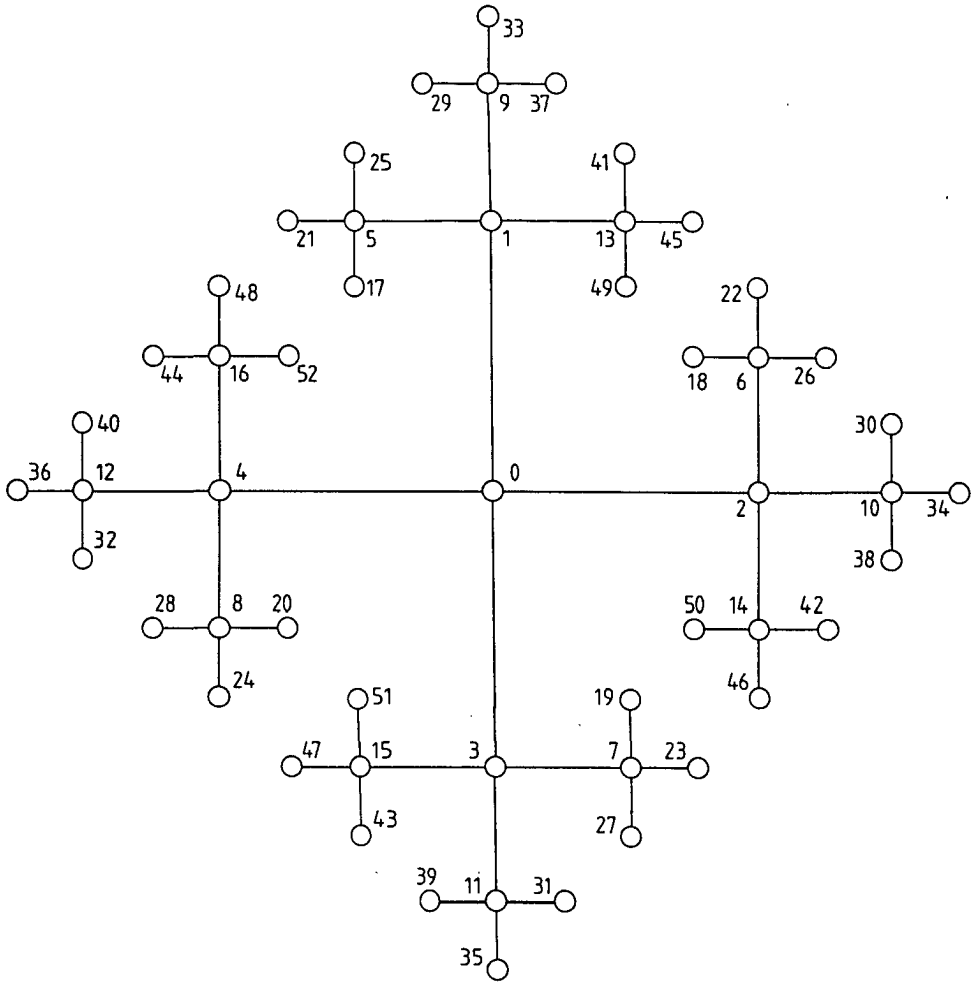


Figure 4.2 Labeling of T_4.

An optimal routing function of T_k^m can easily be obtained from the routing function of T_k. To see this, we express a shortest path between two arbitrary nodes $u=(u_1,\ldots,u_m)$ and $v=(v_1,\ldots,v_m)$ in T_k^m as a subsequence of shortest paths between u_i and v_i $(i=1,\ldots,m)$. A shortest path from u to v is

$$(u_1,u_2,\ldots,u_m),P_1,(v_1,u_2,\ldots,u_m),P_2,(v_1,v_2,u3,\ldots,u_m),\ldots,P_m,(v_1,v_2,\ldots,v_m)$$

where P_i is a shortest path between u_i and v_i in T_k. Hence, optimal routing from u to v can be done by first routing via the first 'dimension', then routing via the

second 'dimension', etc. Trivially, this is not the only optimal routing strategy. There are many others: they are easily obtained from the paths P_i ($i=1,\ldots,m$).

4.5 Concluding remarks

Tree-meshes have many good properties such as symmetry and optimal connectivity. The uniform exponentiality of tree-meshes is lower than we wished: $\overline{exp}(T_k^m)=k-1$ while $deg(T_k^m)=k.m$. The latter is a motive to keep m in T_k^m small, i.e. to let $m=2$. A pleasant side-effect of this choice is that if Δ_i is defined as in paragraph 4.3, i.e. if Δ_i is a convex hull of a ball in T_k^2 or a convex subgraph of T_k^2 interjacent between the convex hulls of two subsequent balls, then the elements of $S_E(\Delta_i)$ have small extension complexity.
The extension complexity of extensible networks based on tree-meshes is not optimal, but small enough to meet our requirements. It is a sublinear function of the number of nodes in the networks.
The reason to describe tree-meshes after all, was three-fold:

1. To illustrate the use of operations on graphs in the construction of underlying graphs.
2. The straightforward labelings and optimal routing functions of tree-meshes enable the simple design of algorithms on tree-meshes.
3. A tree-mesh has meshes and trees as subgraphs, enabling the implementation of many algorithms on it.

In the next chapter we continue with tree-meshes and determine the exact values of the local connectivities in convex subgraphs of tree-meshes.

5

Local connectivities in convex subgraphs of the tree-mesh

It is to be noted that when any part of this paper appears dull there is a design in it.

Sir Richard Steele (1672-1729)

5.1 Introduction

In the previous chapter it was proved that the tree-mesh has optimal connectivity. In this chapter we investigate the local connectivity (see appendix A) in convex subgraphs of the tree-mesh. Unfortunately, local connectivity between nodes in convex subgraphs of T_k^m is not as high as the connectivity of T_k^m itself. The main culprit to this is the fact that the degrees of the border nodes of the subgraphs are less than $\deg(T_k^m)$. In this chapter conditions are determined which must be satisfied by convex subgraphs of the tree-mesh in order that the local connectivity between any two nodes of the subgraph is optimal. In addition, it will be proved that convex subgraphs satisfy these conditions, provided their shape is chosen with some care. The following two kinds of subgraphs of tree-meshes, which were introduced in the previous chapter, have a proper shape.

β_i^m for $i=1,2,3,\ldots,$

where β_i is a ball with radius i in T_k, and

$\Delta_i=\alpha_{\lfloor i/m\rfloor}^{m-(i \bmod m)}\times\alpha_{\lfloor i/m\rfloor+1}^{i \bmod m}$ for $i\geq k^2m+1,$

where α_j is as defined in paragraph 4.3.

Determining the local connectivity between two different nodes u and v amounts to finding the maximum number of paths between u and v having only u and v in common. We call such paths node-disjoint (see appendix A). If u and v lie in a subgraph Δ of a graph Γ, then the local connectivity of u and v in Δ, $\kappa_\Delta(u,v)$, is determined by the maximum number of node-disjoint paths between u and v

lying entirely inside Δ. If u or v is a border node in Δ, then a part of the node-disjoint paths between u and v in Γ do not lie inside Δ. If both u and v are internal nodes in Δ and their distances from the border nodes in Δ are sufficiently large, then there exists a set of node-disjoint paths between u and v in Γ that do lie inside Δ.

In this chapter it will appear that for all convex Δ (except for some pathological cases) and all u and v in Δ lying at distance at least 1 from each border node of Δ, there exists a set of node-disjoint paths between u and v lying entirely inside Δ. To prove this we consider the convex hull of the induced subgraph of Δ based on the subgraph of Δ with node set $\{u,v\}$. This convex hull is denoted by $[<\{u,v\}>]$ (see also appendix A). Thereupon, a set of node-disjoint paths between u and v is constructed such that

- the number of paths in the set inside $[<\{u,v\}>]$ is maximal,
- the paths in the set outside $[<\{u,v\}>]$ lie as 'close' as possible to $[<\{u,v\}>]$.

The paths inside $[<\{u,v\}>]$ lie inside Δ, since Δ is convex. The paths outside $[<\{u,v\}>]$ lie inside Δ, if the distance of any node on them to $[<\{u,v\}>]$ is less than the minimal distance of u and v to any border node of Δ. A measure of 'closeness' of a path to $[<\{u,v\}>]$ is the so-called deflection. For a formal definition of this notion we first need an extension of the distance function.

(5.1) Definition. The *distance* of a *node* u to a *set* A of *nodes* is defined by

$$d(u,A) := \min_{v \in A} d(u,v).$$

(5.2) Definition. The *deflection* δ of a *subgraph* Δ to a *subgraph* Θ of Γ is defined by

$$\delta(\Delta,\Theta) := \max_{u \in V(\Delta)} d(u,V(\Theta)).$$

Notice that δ is not a symmetric function, i.e. $\delta(\Delta,\Theta) \neq \delta(\Theta,\Delta)$. In its most common use, the deflection of a path P to $[<\{u,v\}>]$ is the maximum distance of any node on P to $[<\{u,v\}>]$. The definition of deflection is local to this chapter, and will not be used in other chapters. Other definitions of which the use is limited to this chapter are the following definitions.

(5.3) Definition. If $\Gamma = \Gamma_1 \times \Gamma_2 \times \ldots \times \Gamma_m$, then the *distance* in *dimension* i between two nodes u and v in Γ is defined to be

$$d_i(u,v) := d_{\Gamma_i}(\mathrm{coord}_i(u),\mathrm{coord}_i(v)).$$

(5.4) Definition. The *distance* in *dimension* i of a *node* u to a *set* A of *nodes* is defined to be

$$d_i(u,A) := \min_{v \in A} d_i(u,v).$$

(5.5) Definition. The *deflection* in *dimension* i of a *subgraph* Δ to a *subgraph* Θ is defined to be

$$\delta_i(\Delta,\Theta) := \max_{u \in V(\Delta)} d_i(u,V(\Theta)).$$

(5.6) Definition. Two paths $P_1 = x_1,x_2,\ldots,x_r$ and $P_2 = y_1,y_2,\ldots,y_r$ in Γ are *parallel* if

$$\forall t \in \{1,\ldots,m\}\ \forall i,j \in \{1,\ldots,r\}\ d_t(x_i,y_i) = d_t(x_j,y_j).$$

(5.7) Definition. The *dimension difference* between two nodes u and v in Γ is defined by

$$D(u,v) := \{ i \mid d_i(u,v) > 0 \}.$$

Dimension differences are depicted graphically as in figure 5.1. The configuration in figure 5.1 (a) denotes $D(x,y) = \{i\}$. Figure 5.1 (b) means that the condition

$$\forall x \in X\ \exists y \in Y\colon D(x,y) = \{i\} \text{ and } \forall y \in Y\ \exists x \in X\colon D(x,y) = \{i\}$$

holds.

(a) (b)

Figure 5.1 Dimension differences.

(5.8) Definition. The sets of nodes N(u), $N_i(u)$, NN(u) and $NN_i(u)$ in T_k^m are defined by

$$N(u) := \{v \mid d(u,v) = 1\},$$
$$N_i(u) := \{v \mid d(u,v) = 1 \text{ and } i \in D(u,v)\},$$
$$NN(u) := \{v \mid d(u,v) = 2\},$$
$$NN_i(u) := \{v \mid d(u,v) = 2 \text{ and } i \in D(u,v)\}.$$

In paragraph 5.2 we determine a set of node-disjoint paths between any two

nodes u and v in T_k^m ($m \geq 2$) such that the deflection of each of the paths to $[<\{u,v\}>]$ is minimal. The results of that paragraph are used in paragraph 5.3 to determine the local connectivities between non-border nodes of convex subgraphs of T_k^m ($m \geq 2$). In paragraph 5.4 the same is done for border nodes of convex subgraphs of T_k^m. Finally, in paragraph 5.5 the results of paragraphs 5.3 and 5.4 are applied to the convex subgraphs of T_k^m ($m \geq 2$) mentioned in the introduction of this chapter, β_i^m and Δ_i.

5.2 Node-disjoint paths and their deflections to convex hulls

In this paragraph we make the preparations to construct the maximum number of node-disjoint paths between any two nodes in a convex subgraph Δ of T_k^m ($m \geq 2$) which lie entirely in Δ. In order to determine such paths we start with two arbitrary nodes u and v in T_k^m, and construct a set of node-disjoint paths between u and v such that the deflection of each path to $[<\{u,v\}>]$ is minimal. In paragraphs 5.3 and 5.4 it will appear that such constructs guarantee the placement of u and v in a convex subgraph Δ of T_k^m such that the number of node-disjoint paths between u and v fitting into Δ is maximal, even if u or v lie close to but not onto border nodes of Δ.

We mark off three cases:

A. case $m \geq 2$, $|D(u,v)| \geq 2$.

B. case $m \geq 3$, $|D(u,v)| = 1$.

C. case $m = 2$, $|D(u,v)| = 1$.

In all cases and the rest of this chapter we assume that the dimensions are ordered such that $D(u,v) = \{1,2,\ldots,|D(u,v)|\}$ $(u \neq v)$.

A. case $m \geq 2$, $|D(u,v)| \geq 2$

In the next two lemmas we determine a set of node-disjoint paths between u and v lying in an n-dimensional mesh T_2^n at minimal deflection from $[<\{u,v\}>]$, where $|D(u,v)| = n$ and $n \geq 2$.

(5.9) Lemma. If u and v are two nodes in an n-dimensional mesh T_2^n ($n \geq 2$) such that $d_i(u,v) = 1$ ($i = 1,\ldots,n$), then there are 2.n node-disjoint paths from u to v, n of which lie inside $[<\{u,v\}>]$ and n of which have deflection at most 1 from $[<\{u,v\}>]$ in each of the n dimensions.

Proof. Clearly, $[<\{u,v\}>]$ is an n-cube. Cubes are distance-transitive, implying that they have connectivity n (see theorem 2.14). Hence, there are n node-disjoint paths lying inside $[<\{u,v\}>]$. For convenience in the proof of lemma 5.2 we denote these paths by $Q_1,\ldots,Q_n$.
To prove the existence of yet other n node-disjoint paths with deflection at most 1 from $[<\{u,v\}>]$ in all n dimensions, we assume that the nodes in T_2^n are identified by n-tuples in the usual way in Cartesian graph products (i.e. the tuples corresponding to two adjacent nodes differ by 1 in exactly one element). Suppose $u=0$ (all zero tuple) and $v=1$ (all one tuple). Let e_i be the tuple of which all coordinates are 0 except the i^{th} which is 1. The e_i will be used for easy denotation of nodes. Any node can be represented in a unique way by a sum of e's.
Let $f:\{1,\ldots,n\}\rightarrow\{1,\ldots,n\}$ be a bijective function for which $f(i)\neq i$. Suppose $j=f(i)$. We construct n paths $P_1,\ldots,P_n$, having deflection at most 1 from $[<\{u,v\}>]$ in all n dimensions. P_i is defined by

$$u = 0,\ -e_i,\ e_j-e_i,\ 2e_j-e_i,\ 2e_j,\ e_1+2e_j,\ e_1+e_2+2e_j,\ldots,\ e_1+\ldots+e_{j-1}+2e_j,$$
$$e1+\ldots+e_{j-1}+2e_j+e_{j+1},\ldots,\ e_1+\ldots+e_{j-1}+2e_j+e_{j+1}+\ldots+e_n,\ \sum_{t=1}^{n} e_t = v.$$

Then, for each intermediate node x on path P_i:

$$(\mathrm{coord}_i(x)=-1 \text{ or } \mathrm{coord}_{f(i)}(x)=2)$$

and

$$(\mathrm{coord}_t(x)\neq -1 \text{ for } t\neq i \text{ and } \mathrm{coord}_t(x)\neq 2 \text{ for } t\neq f(i))$$

Using the above relation it is easily verified that all paths $P_1,\ldots,P_n$ are node-disjoint.
Furthermore, P_i does not pass through $[<\{u,v\}>]$, and it has deflection 1 from $[<\{u,v\}>]$ in dimensions i and f(i) and deflection 0 from $[<\{u,v\}>]$ in all other dimensions. □

As we have seen in this lemma, a path in a mesh can be described by a sequence of additions or subtractions of e_i $(i=1,\ldots,n)$. Such a sequence will be called the *e-sequence* of the corresponding path. The e-sequence of P_i in the previous lemma is "$-e_i, +e_j, +e_j, +e_i, +e_1, +e_2,\ldots, +e_{j-1}, +e_{j+1},\ldots, +e_n, -e_j$".

(5.10) Lemma. If u and v are two nodes in an n-dimensional mesh T_2^n $(n\geq 2)$ such that $|D(u,v)|=n$, then there are 2.n node-disjoint paths from u to v, n of which lie inside $[<\{u,v\}>]$ and n of which have deflection at most 1 from $[<\{u,v\}>]$ in each of the n dimensions.

Proof. The proof is similar to that of lemma 5.9. Suppose $u=(0,\ldots,0)$ and $v=(d_1,\ldots,d_n)$ $(d_j \geq 1)$, d_j being defined by $d_j = d_j(u,v)$. Let $E_1,\ldots,E_n$ be the e-sequences of the paths $Q_1,\ldots,Q_n$ respectively inside the n-cube in lemma 5.9. We shall construct a set of n node-disjoint paths $Q_1',\ldots,Q_n'$ between u and v inside $[<\{u,v\}>]$ which are based on $Q_1,\ldots,Q_n$. This is done by substituting the subsequence "$+e_t$" in E_i (see proof of lemma 5.9) by the subsequence "$+e_t,\ldots,+e_t$" of length d_t for all $t=1,\ldots,n$. It results in e-sequences E_i' corresponding to the paths Q_i' from u to v inside $[<\{u,v\}>]$. None of these paths coincide, for otherwise the Q_i in lemma 5.1 wouldn't be node-disjoint.

To prove the existence of the remaining n node-disjoint paths with deflection at most 1 from $[<\{u,v\}>]$ in all n dimensions, we suppose that $f:\{1,\ldots,n\}\to\{1,\ldots,n\}$ is again a bijective function for which $f(i)\neq i$ and $j=f(i)$.
Let P_i be a path from u to v defined by:

$$u = 0,\ -e_i,\ e_j-e_i,\ 2e_j-e_i,\ldots,\ (d_j+1)e_j-e_i,\ (d_j+1)e_j,\ e_1+(d_j+1)e_j,$$
$$2e_1+(d_j+1)e_j,\ldots,\ d_1e_1+(d_j+1)e_j,\ldots,\ \sum_{t=1}^{j-1} d_te_t+(d_j+1)e_j,\ldots,$$
$$\sum_{t=1}^{j-1} d_te_t+(d_j+1)e_j+\sum_{t=j+1}^{n} d_te_t,\ \sum_{t=1}^{n} d_te_t = v.$$

We notice that for each intermediate node x on P_i:

$$(\mathrm{coord}_i(x)=-1 \text{ or } \mathrm{coord}_{f(i)}(x)=d_{f(i)}+1)$$

and

$$(\mathrm{coord}_t(x)\neq -1 \text{ for } t\neq i \text{ and } \mathrm{coord}_t(x)\neq d_t+1 \text{ for } t\neq f(i))$$

which gives an easy verification of the P_i being node-disjoint.
Furthermore, P_i does not pass through $[<\{u,v\}>]$, and it has deflection 1 from $[<\{u,v\}>]$ in dimensions i and f(i) and deflection 0 in all other dimensions. □

We can now establish a set of node-disjoint paths between two nodes in T_k^m $(k\geq 2,\ m\geq 2)$ such that each path has minimal deflection to $[<\{u,v\}>]$. To obtain these paths, we notice that meshes are proper subgraphs of tree-meshes, provided the dimension of the former does not exceed that of the latter. We use this observation in the proof of the following lemma.

(5.11) Lemma. If u and v are nodes in T_k^m $(m\geq 2,\ k\geq 2)$ for which $|D(u,v)|\geq 2$, then there are k.m node-disjoint paths from u to v, $|D(u,v)|$ of which lie inside $[<\{u,v\}>]$ and the rest having deflection at most 1 from $[<\{u,v\}>]$ in each of the m dimensions.

Proof. Let w_i $(1 \leq i \leq |D(u,v)|)$ be a node in T_k^m defined by

$$D(u,w_i) := \{i\}$$
$$D(v,w_i) := D(u,v) - \{i\}$$

This is a unique characterisation of w_i. Let A_i and B_i be defined by

$$A_i := N_i(u) - V([<\{u,w_i\}>])$$
$$B_i := N_i(w_i) - V([<\{u,w_i\}>])$$

Then, A_i consists of all neighbours of u differing in dimension i from u and not lying on the shortest path between u and w_i. B_i is determined analogously. Hence, $|A_i| = |B_i| = k-1$.

Let $a_{i1}, a_{i2}, \ldots, a_{ik-1}$ be the elements of A_i and $b_{i1}, b_{i2}, \ldots, b_{ik-1}$ be the elements of B_i. Let R_{ij} $(1 \leq j \leq k-1)$ be a two way infinite path in T_k^m through a_{ij}, b_{ij} and $V([<\{u,w_i\}>])$ such that for each node x in R_{ij}: $D(u,x) \subseteq \{i\}$. Note that the part of R_{ij} outside $V([<\{u,w_i\}>]) \cup \{a_{ij}, b_{ij}\}$ is not uniquely determined by this. We know that $j_1 \neq j_2$ implies $R_{ij_1} \cap R_{ij_2} = [<\{u,w_i\}>]$ $(1 \leq i \leq |D(u,v)|)$, since all R_{ij} lie in a particular subtree T_k of T_k^m.

We define meshes M_j $(j = 1, \ldots, k-1)$ to be subgraphs of T_k^m equal to

$$M_j := R_{1j} \times R_{2j} \times \ldots \times R_{|D(u,v)|j}.$$

M_j is a mesh of dimension $|D(u,v)|$ comprising a_{ij}, b_{ij} and $[<\{u,w_i\}>]$ for $i = 1, \ldots, |D(u,v)|$. By lemma 5.10 there are $2.|D(u,v)|$ paths from u to v in M_j, $|D(u,v)|$ of which lie inside $[<\{u,v\}>]$ and the rest having deflection at most 1 from $[<\{u,v\}>]$ in dimensions 1 to $|D(u,v)|$. All $(k-1).|D(u,v)|$ paths from u to v in all M_j $(1 \leq j \leq k-1)$ outside $[<\{u,v\}>]$ are node-disjoint since the M_j are disjoint outside $[<\{u,v\}>]$. For, $j_1 \neq j_2$ implies

$$M_{j_1} \cap M_{j_2} = (R_{1j_1} \times \ldots R_{|D(u,v)|j_1}) \cap (R_{1j_2} \times \ldots R_{|D(u,v)|j_2} =$$
$$(R_{1j_1} \cap R_{1j_2}) \times \ldots \times (R_{|D(u,v)|j_1} \cap R_{|D(u,v)|j_2}) =$$
$$[<\{u,w_1\}>] \times \ldots \times [<\{u,w_{|D(u,v)|}\}>] = [<\{u,v\}>].$$

Since there are $k-1$ of such meshes, there are $(k-1).|D(u,v)|$ node-disjoint paths from u to v outside $[<\{u,v\}>]$ having deflection at most 1 from $[<\{u,v\}>]$ in each of the m dimensions (in fact, the deflection is 1 in exactly two dimensions). Since the meshes coincide inside $[<\{u,v\}>]$, there are only $|D(u,v)|$ node-disjoint paths from u to v inside $[<\{u,v\}>]$.

The remaining $k.(m - |D(u,v)|)$ paths from u to v are constructed as follows.

Define $x_{i1}, x_{i2}, \ldots, x_{ik}$ to be the nodes in $N_i(u)$ and $y_{i1}, y_{i2}, \ldots, y_{ik}$ to be the nodes in $N_i(v)$ $(|D(u,v)| < i \leq m)$, ordered in such a way that $i \notin D(x_{ij}, y_{ij})$ $(1 \leq j \leq k)$. Such an ordering is possible, since $i > D(u,v)$. We notice that by this definition $i_1 \neq i_2$ or

$j_1 \neq j_2$ implies $D(x_{i_1 j_1}, x_{i_2 j_2}) \neq \varnothing$.
Let P be a shortest path from u to v (so, P lies inside $[<\{u,v\}>]$) and let P_{ij} be the path parallel to P and starting in x_{ij} (and so ending in y_{ij}). All P_{ij} are node-disjoint with respect to each other, since they are parallel and they have different start nodes. Moreover, P_{ij} has deflection 1 from $[<\{u,v\}>]$ in dimension i and 0 in all other dimensions. Each P_{ij} differs in dimension i from the $k.|D(u,v)|$ paths constructed above. The start- and end-nodes of P_{ij} are incident to u and v respectively, resulting in the $k.(m-|D(u,v)|)$ paths from u to v which remained to be constructed. □

B. case $m \geq 3$, $|D(u,v)| = 1$

The construction of node-disjoint paths between u and v in this section is more direct than in the previous case.

(5.12) Lemma. If u and v are nodes in T_k^m ($m \geq 3$, $k \geq 2$) for which $|D(u,v)| = 1$, then there are k.m node-disjoint paths from u to v having deflection at most 1 from $[<\{u,v\}>]$ in each of the m dimensions.

Proof. There is one shortest path P from u to v. Let $x_{i1}, x_{i2}, \ldots, x_{ik}$ be the nodes in $N_i(u)$ and $y_{i1}, y_{i2}, \ldots, y_{ik}$ be the nodes in $N_i(v)$ ($2 \leq i \leq m$), ordered in such a way that $i \notin D(x_{ij}, y_{ij})$ ($1 \leq j \leq k$). Let P_{ij} be the path parallel to P and starting in x_{ij} (and so ending in y_{ij}). There are $k(m-1)$ of such paths. All P_{ij} are node-disjoint with respect to each other and to P, and P_{ij} has deflection 1 from $[<\{u,v\}>] = P$ in dimension i and 0 in all other dimensions. The start- and end-nodes of P_{ij} are incident to u and v respectively, which completes the construction of the paths through $N_i(u)$ and $N_i(v)$ ($i = 2, \ldots, m$).

Define sets of nodes A and B as

$$A := N_1(u) - V([<\{u,v\}>]),$$

$$B := N_1(v) - V([<\{u,v\}>]).$$

Then, $|A| = |B| = k-1$. There remains $k-1$ paths to be constructed from u to v passing through A and B. These paths must be node-disjoint from the paths constructed above.
Let C be a subset of $N_2(u)$ with cardinality $k-1$. Let A′ be defined as the unique subset of

$$\{x \in V(T_k^m) \mid \exists y \in A: D(x,y) = \{2\} \text{ and } \exists z \in C: D(x,z) = \{1\}\}$$

such that for each two nodes x and y in A′: $D(x,y) = \{1,2\}$. Analogously, define B′ as the unique subset of

$\{x \in V(T_k^m) \mid \exists y \in B: D(x,y) = \{2\} \text{ and } \exists z \in C: D(x,z) = \{1\}\}$

such that for each two nodes x and y in B': $D(x,y) = \{1,2\}$.

Then $|A'| = |B'| = k-1$, $d_1(x,u) = d_2(x,u) = 1$, $d_i(x,u) = 0$ $(2 \leq i \leq m)$ for each node $x \in A'$, and $d_1(x,v) = d_2(x,v) = 1$, $d_i(x,v) = 0$ $(2 \leq i \leq m)$ for each node $x \in B'$. All shortest paths from A' to B' coincide with the paths P_{2j} $(j = 1,\ldots,k)$. So, to construct the remaining $k-1$ node-disjoint paths between u and v, a diversion in dimension 3 is needed.
Let w be a node in $N_3(u)$. A'' and B'' are defined by

$A'' := \{x \in V(T_k^m) \mid \exists y \in A': D(x,y) = \{3\} \text{ and } D(x,w) = \{1,2\}\}$

$B'' := \{x \in V(T_k^m) \mid \exists y \in B': D(x,y) = \{3\} \text{ and } D(x,w) = \{1,2\}\}$

Then $|A''| = |B''| = k-1$, $d_1(x,u) = d_2(x,u) = d_3(x,u) = 1$, $d_i(x,u) = 0$ $(3 \leq i \leq m)$ for each node $x \in A''$, and $d_1(x,v) = d_2(x,v) = d_3(x,v) = 1$, $d_i(x,v) = 0$ $(3 \leq i \leq m)$ for each node $x \in B''$. See figure 5.2.

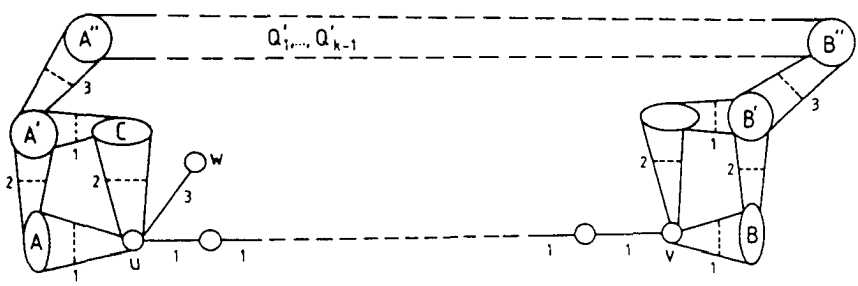


Figure 5.2 Situation occurring in lemma 5.12.

Now, there are $k-1$ paths $Q_1,\ldots,Q_{k-1}$ from A to B having different start- and end-nodes. These paths have P in common. Let Q_t' be a path starting in A'' and parallel to Q_t $(t = 1,\ldots,k-1)$. This path ends up in B''. Each node in A'' and each node in B'' lies in one of these paths. Every node in Q_t' differs at least in dimensions 2 and 3 from P. So, Q_t' is node-disjoint from P. Since each P_{ij} differs only in one dimension from P, Q_t' is node-disjoint from P_{ij}. Moreover, if $t_1 \neq t_2$ then Q_{t_1}' is disjoint from Q_{t_2}'. They differ at least in dimension 2, since their start nodes (in A'') differ in dimension 2. Concatenation of

- the shortest paths from u to A'' via A and A',
- the Q_t' $(t = 1,\ldots,k-1)$, and

- the shortest paths from B'' to v via B' and B

completes the construction of the remaining $k-1$ paths. These paths are node-disjoint from P and P_{ij} ($i=2,\ldots,m$; $j=1,\ldots,k$) since A, A', A'', B, B' and B'' are node-disjoint from P and P_{ij}. □

C. case $m=2$, $|D(u,v)|=1$

This case is similar to the previous case, except that there is no third dimension for diversion of node-disjoint paths between u and v, passing through the sets A'' and B''. Therefore, we make a longer detour through dimension 2.

(5.13) Lemma. If u and v are nodes in T_k^2 ($k\geq 2$) for which $|D(u,v)|=1$, then there are 2.k node-disjoint paths from u to v having deflection at most 1 from $[<\{u,v\}>]$ in dimension 1 and deflection at most 2 from $[<\{u,v\}>]$ in dimension 2.

Proof. The proof is similar to that of lemma 5.12 up to the point of the introduction of C. For the remaining $k-1$ paths there is no third dimension via which the paths can be diverted. The only alternative is the second dimension but this will increase the deflection by 1 in dimension 2.
Suppose S is a set of $k-1$ nodes such that for each $s\in S$: $D(u,s)=\{2\}$ and $d(u,s)=2$. Let A'' be defined as the unique subset of

$$\{x\in V(T_k^m) \mid \exists y\in A: D(x,y)=\{2\} \text{ and } \exists s\in S: D(x,s)=\{1\}\}$$

such that for each two nodes x and y in A'': $D(x,y)=\{1,2\}$ (No A' is defined in this proof). Then, $|A''|=k-1$ and for each node z in A there is exactly one node z'' in A'' only differing in dimension 2 from z. Consider the shortest path from node z to z''. There are exactly $k-1$ of such paths from A to A'' and they differ in dimension 1 from each other.
Again, let Q_t' be a path starting in A'' and parallel to Q_t ($t=1,\ldots,k-1$). The paths $Q_1',\ldots, Q_{k-1}'$ end up in a set that will be denoted by B''. Q_t' has deflection 1 from P in dimension 1 and deflection 2 in dimension 2. Furthermore, Q_t' is disjoint from P and P_{2j} ($j=1,\ldots,k$), and Q_{t_1}' is disjoint from Q_{t_2}', if $t_1\neq t_2$.
As with the shortest paths from A to A'', there are $k-1$ mutually disjoint shortest paths from B to B'' differing in dimension 1. Concatenation of

- the shortest paths from u to A'' via A,
- the Q_t' ($t=1,\ldots,k-1$), and
- the shortest paths from B'' to v via B,

completes the construction of the remaining $k-1$ paths. As in the previous lemma these paths are node-disjoint from P and P_{2j} $(j=1,\ldots,k)$. □

We are now in the position to formulate necessary conditions for optimal connectivities in convex subgraphs of the tree-mesh.

5.3 Non-border nodes

Lemmas 5.11, 5.12, and 5.13 will be used in this paragraph to prove that the local connectivity between two non-border nodes in any convex subgraph Δ of T_k^m $(k\geq 2, m\geq 2)$ is optimal, except for a few pathological subgraphs Δ.

(5.14) Theorem. Let u and v be non-border nodes in a convex subgraph Δ of T_k^m $(k\geq 2)$. If $m\geq 2$ and $|D(u,v)|\geq 2$, then $\kappa_\Delta(u,v)=k.m$.

Proof. If u and v are non-border nodes of Δ then the deflection of each border node to $[<\{u,v\}>]$ is at least 1 in each of the m dimensions. This means that any path having deflection at most 1 from $[<\{u,v\}>]$ lies inside Δ. Then, lemma 5.11 implies the existence of k.m node-disjoint paths from u to v in Δ, giving the required result. □

(5.15) Theorem. Let u and v be non-border nodes in a convex subgraph Δ of T_k^m $(k\geq 2)$. If $m\geq 3$ and $|D(u,v)|=1$, then $\kappa_\Delta(u,v)=k.m$.

Proof. Similar to the proof of theorem 5.14. □

If u and v are nodes in a convex subgraph Δ of T_k^2 $(k\geq 2)$ at distance 1 from any border node of Δ and $D(u,v)=\{1\}$, we might conclude that $\kappa_\Delta(u,v)$ is not optimal, since by lemma 5.13 distance 2 from any border is needed for optimality. This conclusion is justifiable. Special shape of Δ, however, results in optimality of $\kappa_\Delta(u,v)$ in case of unit distance of u and v from any border node of Δ.

(5.16) Theorem. Let u and v be non-border nodes in a convex subgraph Δ of T_k^2 $(k\geq 2)$. If $|D(u,v)|=1$ and

$$|(NN(u)-NN_1(u))\cap V(\Delta)| \geq k-1,$$

then $\kappa_\Delta(u,v)=2.k$.

Proof. We let $S \subseteq (NN(u)-NN_1(u)) \cap V(\Delta)$ such that $|S|=k-1$. Then, the constructions in the proof of lemma 5.13 are applicable, leaving all node-disjoint paths between u and v inside Δ. □

5.4 Border nodes

If one or both of the nodes u or v is a border node of a subgraph Δ of T_k^m ($k \geq 2$, $m \geq 2$), then $\kappa_\Delta(u,v) < \kappa(T_k^m)$. Optimality of $\kappa_\Delta(u,v)$ is less easily proved in that case. In order to prove that $\kappa_\Delta(u,v)$ is optimal, we have to construct $\min(|N(u) \cap V(\Delta)|, |N(v) \cap V(\Delta)|)$ node-disjoint paths from u to v lying entirely inside Δ. We start with the case in which nodes u and v differ in only one dimension.

(5.17) Theorem. Let u and v be nodes in a convex subgraph Δ of T_k^m ($k \geq 2$, $m \geq 2$). If u or v is a border node of Δ, $|D(u,v)|=1$, and

$$|(NN(u)-NN_1(u)) \cap V(\Delta)| \geq \min(|N_1(u) \cap V(\Delta)|, |N_1(v) \cap V(\Delta)|) - 1,$$

then $\kappa_\Delta(u,v)$ is optimal.

Proof. We have to prove that the number of node-disjoint paths between u and v lying entirely inside Δ is equal to

$$\min(|N(u) \cap V(\Delta)|, |N(v) \cap V(\Delta)|).$$

First, we notice that the shortest path from u to v lies inside Δ. Call this path P. Second, we construct the paths via $N_i(u) \cap V(\Delta)$ and $N_i(v) \cap V(\Delta)$ (for all $i \geq 2$). Let $w \in N_i(u)$ and $w' \in N_i(v)$, such that $i \notin D(w,w')$. Then $w \in V(\Delta)$ if and only if $w' \in V(\Delta)$, since Δ is convex. If $w \in V(\Delta)$, then a path from w to w', parallel to P, is constructed in the same way as the paths P_{ij} in lemma 5.12 are constructed. Doing this for all nodes w in $N_i(u) \cap V(\Delta)$, we obtain a set of node-disjoint paths from u to v via $N_i(u) \cap V(\Delta)$ and $N_i(v) \cap V(\Delta)$ ($i=2,\ldots,m$).

The paths passing through the sets of nodes $(N_1(u)-V([<\{u,v\}>])) \cap V(\Delta)$ and $(N_1(v)-V([<\{u,v\}>])) \cap V(\Delta)$ remain to be constructed. We have

$$|(NN(u)-NN_1(u)) \cap V(\Delta)| = |(NN(v)-NN_1(v)) \cap V(\Delta)|,$$

since Δ is convex. Suppose without any loss of generality that $|N_1(u) \cap V(\Delta)| \leq |N_1(v) \cap V(\Delta)|$. Let A'' be defined as a subset of

$$\{x \in V(T_k^m) \mid \exists y \in (N_1(u)-V([<\{u,v\}>])) \cap V(\Delta): 1 \notin D(x,y) \text{ and } \exists z \in (NN(u)-NN_1(u)) \cap V(\Delta): D(x,z)=\{1\}\},$$

such that for each two nodes x and y in A'':

$$D(x,y)-\{1\} \neq \varnothing$$

In the same way as in lemmas 5.12 and 5.13 we construct $|N_1(u)\cap V(\Delta)|-1$ paths starting in A'' and being one-by-one parallel to the $|N_1(u)\cap V(\Delta)|-1$ shortest paths starting in the set of nodes $(N_1(u)-V([<\{u,v\}>]))\cap V(\Delta)$ and ending in the set of nodes $(N_1(v)-V([<\{u,v\}>]))\cap V(\Delta)$. These paths have different start- and end-nodes and so are node-disjoint. They have deflection 2 from $[<\{u,v\}>]$ in dimension 2 if $m=2$. If $m\geq 3$ then the deflection of a path from $[<\{u,v\}>]$ is 1 in two dimensions or 2 in one dimension. It is possible to construct these paths since

$$|(NN(u)-NN_1(u))\cap V(\Delta)| \geq \min(|N_1(u)\cap V(\Delta)|,|N_1(v)\cap V(\Delta)|)-1,$$

implying that the set A'' consisting of $\min(|N_1(u)\cap V(\Delta)|,|N_1(v)\cap V(\Delta)|)-1$ nodes is constructible.
All paths from u to v inside Δ which we have constructed, are node-disjoint because of the similarity with lemmas 5.12 and 5.13. We conclude that $\kappa_\Delta(u,v)$ is optimal. □

(5.18) Theorem. Let u and v be nodes in a convex subgraph Δ of T_k^m ($k\geq 2$, $m\geq 2$). If u or v is a border node of Δ, $|D(u,v)|\geq 2$, and for all i in the set $\{1,\ldots,|D(u,v)|\}$:

$$|(NN(u)-NN_i(u))\cap V(\Delta)| \geq \min(|N_i(u)\cap V(\Delta)|,|N_i(v)\cap V(\Delta)|)-1,$$

then $\kappa_\Delta(u,v)$ is optimal.

Proof. In lemma 5.11 three classes of paths from u to v in T_k^m are distinguished:

1. The paths inside $[<\{u,v\}>]$,
2. The paths with deflection 1 from $[<\{u,v\}>]$ in two of the dimensions in $\{1,\ldots,|D(u,v)|\}$ and deflection 0 in all other dimensions,
3. The paths with deflection 1 from $[<\{u,v\}>]$ in one of the dimensions in $\{|D(u,v)|+1,\ldots,m\}$ and deflection 0 in all other dimensions.

The paths in class 1 lie inside Δ since Δ is convex.
The number of class 2 paths inside Δ is determined by

$$\min_{x\in\{u,v\}} |((\bigcup_{i=1}^{|D(u,v)|} N_i(x))-V([<\{u,v\}>]))\cap V(\Delta)|.$$

Assume u is a border node of Δ. If v is not a border node, the minimum is achieved for $x=u$, otherwise we assume without any loss of generality that $x=u$ gives the minimum.

We shall relate each node in $(N_i(u)-V([<\{u,v\}>]))\cap V(\Delta)$ $(i=1,\ldots,|D(u,v)|)$ to a unique node in $(N_j(v)-V([<\{u,v\}>]))\cap V(\Delta)$ for some j not equal to i

$(1 \leq j \leq |D(u,v)|)$, if possible. In this way node-disjoint paths in Δ with deflection 1 in dimensions i and j are constructed in the same way as in lemmas 5.9 to 5.11. This might cause a problem, however. The problem arises if there exists an i in the set $\{1,\ldots,|D(u,v)|\}$ for which

$$|(N_i(u)-V([<\{u,v\}>]))\cap V(\Delta)| > |((\bigcup_{t\neq i} N_t(v))-V([<\{u,v\}>]))\cap V(\Delta)|,$$

where t runs from 1 to $|D(u,v)|$.
In this case not enough nodes in the set S_i, defined by

$$S_i := ((\bigcup_{t\neq i} N_t(v))-V([<\{u,v\}>]))\cap V(\Delta),$$

are available for paths from u to v through $(N_i(u)-V([<\{u,v\}>]))\cap V(\Delta)$ for some i.
We have to find alternative routes for the paths passing through the set of nodes $(N_i(u)-[<\{u,v\}>])\cap V(\Delta)$ and not passing through S_i. First, we notice that there is only one i causing trouble (if there is one). For, if there are two, say i_1 and i_2 $(i_1 \neq i_2)$, then

$$|(N_{i_1}(u)-V([<\{u,v\}>]))\cap V(\Delta)| + |(N_{i_2}(u)-V([<\{u,v\}>]))\cap V(\Delta)| >$$

$$|((\bigcup_{t\neq i_1} N_t(v))-V([<\{u,v\}>]))\cap V(\Delta)|+|((\bigcup_{t\neq i_2} N_t(v))-V([<\{u,v\}>])\cap V(\Delta)| \geq$$

$$|((\bigcup_{t=1}^{|D(u,v)|} N_t(v))-V([<\{u,v\}>]))\cap V(\Delta)| \geq$$

$$|((\bigcup_{t=1}^{|D(u,v)|} N_t(u))-V([<\{u,v\}>]))\cap V(\Delta)| \geq$$

$$|(N_{i_1}(u)-V([<\{u,v\}>])\cap V(\Delta)| + |(N_{i_2}(u)-V([<\{u,v\}>])\cap V(\Delta)|,$$

which is a contradiction.

Suppose i_0 is the one causing trouble. Then, we must construct paths passing through $N_{i_0}(u)-V([<\{u,v\}>])$ and $N_{i_0}(v)-V([<\{u,v\}>])$. Let R_0 be the subset of nodes of $N_{i_0}(u)-V([<\{u,v\}>])$ through which no one of the paths passing through S_{i_0} passes. The paths to be constructed should pass through R_0. In order to obtain these paths we first construct

$$\min(|N_{i_0}(u)\cap V(\Delta)|,|N_{i_0}(w_{i_0})\cap V(\Delta)|)-1$$

paths from u to w_{i_0} in exactly the same way as the paths through $N_1(u)\cap V(\Delta)$ and $N_1(v)\cap V(\Delta)$ in theorem 5.17, where w_{i_0} is a node uniquely defined by

$$D(u,w_{i_0}) := \{i_0\}$$
$$D(v,w_{i_0}) := D(u,v)-\{i_0\}.$$

These paths have deflection 2 from $[<\{u,w_{i_0}\}>]$ and hence from $[<\{u,v\}>]$ in one dimension different from i_0. It is possible to construct these paths, since

$$|NN(u)-NN_{i_0}(u))\cap V(\Delta)| \geq \min(|N_{i_0}(u)\cap V(\Delta)|,|N_{i_0}(v)\cap V(\Delta)|)-1$$

and (since Δ is convex)

$$|N_{i_0}(w_{i_0})\cap V(\Delta)| = |N_{i_0}(v)\cap V(\Delta)|.$$

We are only interested in the subset of these paths of which the elements pass through R_0. We select the paths through R_0 and break them off in $N_{i_0}(w_{i_0})$. This results in a set R_1 of $|R_0|$ node-disjoint paths from u to a subset of $N_{i_0}(w_{i_0})\cap V(\Delta)$ with separate end-nodes. Each path in R_1 will be extended by a path, starting in an end-node in $N_{i_0}(w_{i_0})\cap V(\Delta)$ and being parallel to a shortest path between w_{i_0} and v. An extended path ends up in $N_{i_0}(v)\cap V(\Delta)$, which is incident to v. The set of extended paths is denoted by R_2. Concatenation of R_1, R_2 and the edges between v and the end nodes in $N_{i_0}(v)\cap V(\Delta)$ of the paths in R_2 results in the required set of node-disjoint paths. We call this set R in the rest of this proof. All paths in R are node-disjoint.

Finally, the paths in the third class, having deflection 1 in one of the dimensions of $\{|D(u,v)|+1,\ldots,m\}$ via $N_i(u)\cap V(\Delta)$ and $N_i(v)\cap V(\Delta)$ $(i>|D(u,v)|)$, are constructed in the same way as in theorem 5.17.

Because of the similarity with lemma 5.11 all constructed paths from u to v are node-disjoint, with a possible exception for the paths in R.
As a matter of course a path in R is node-disjoint from class 1 paths.
Disjointness of the paths in R from class 2 paths not being paths in R is proved by noting that for each node x in any path of R:

a. $d_i(x,V([<\{u,v\}>]))=2$, for some $i\in\{1,\ldots,m\}-\{i_0\}$, and for each node x lying on a subpath in R_1 (notice that $d_i(x,V([<\{u,w_i\}>]))=2$),

b. $d_{i_0}(x,V([<\{u,v\}>]))=1$, for each node x lying on a subpath in R_2.

Condition a immediately implies that any class 2 path P not being a path in R is node-disjoint from any subpath in R_1, since no node in P has distance 2 from $V([<\{u,v\}>])$ in any dimension.
Concerning condition b, let z be a node on a subpath in the set of subpaths R_2. Then, $d_{i_0}(z,V([<\{u,v\}>]))=1$. This implies that for each node y in a class 2 path P not being a path in R: $i_0\in D(z,y)$, otherwise P and R would have common nodes in $N_{i_0}(u)-V([<\{u,v\}>])$ and $N_{i_0}(v)-V([<\{u,v\}>])$, which is impossible since P does not pass through R_0.

For each node y on a class 3 path: $d_i(y,V([<\{u,v\}>]))=1$ for exactly one $i>|D(u,v)|$. For any other i we have $d_i(y,V([<\{u,v\}>]))=0$. It is easily seen that no node on a path in R satisfies these conditions. □

Putting lemmas 5.17 and 5.18 together we obtain the following corollary.

(5.19) Corollary. Let u and v be nodes in a convex subgraph Δ of T_k^m ($k\geq 2$, $m\geq 2$). If u or v is a border node of Δ and for all $i\in\{1,\ldots,|D(u,v)|\}$:

$$|(NN(u)-NN_i(u))\cap V(\Delta)| \geq \min(|N_i(u)\cap V(\Delta)|,|N_i(v)\cap V(\Delta)|)-1,$$

then $\kappa_\Delta(u,v)$ is optimal.

5.5 Local connectivities of some convex subgraphs of T_k^m

In this paragraph the theorems of the preceding paragraphs are applied to the convex subgraphs in chapter 4, i.e. the subgraphs β_i^m and Δ_i of T_k^m in paragraph 5.1.

(5.20) Lemma. Let Δ_i ($i\geq k^2m+1$) be a convex subgraph of T_k^m as defined in the introduction ($k\geq 2$, $m\geq 2$). Then, for each node u in Δ_i and for each $j\in\{1,\ldots,m\}$:

$$|(NN(u)-NN_j(u))\cap V(\Delta_i)| \geq k-1.$$

Proof. Let γ_r denote a subgraph of T_k having a ball β_r with radius r as subgraph and being a proper subgraph of a ball β_{r+1} with radius $r+1$, both balls centered around the same node. Then, for each node z in γ_r ($r\geq 2$):

$$|NN(z)\cap V(\gamma_r)| \geq k-1.$$

Let $\tau_n(u)$ be an induced subgraph of T_k^m with node set $\{v|\ D(u,v)\subseteq\{n\}\}$. Clearly, $\tau_n(u)$ is isomorphic to T_k. Then,

$$NN(u)\cap\tau_n(u) = NN(u)-\bigcup_{t\in\{1,\ldots,m\}-\{n\}} NN_t(u) \subseteq NN(u)-NN_j(u) \quad (j\neq n).$$

Hence,

$$NN(u)\cap\tau_n(u)\cap V(\Delta_i) \subseteq (NN(u)-NN_j(u))\cap V(\Delta_i) \quad (j\neq n),$$

for any subgraph Δ_i of T_k^m. If $i\geq k^2m+1$ and n is equal to $m-1$ or m, then $\tau_n(u)\cap\Delta_i$ has a subgraph which is a ball with radius at least 2. For, if $i\geq k^2m+1$, then $\alpha_{k^2+1}^{m-2}\times\alpha_{k^2+2}^2=\alpha_{k^2+1}^{m-2}\times\beta_2^2$ is a subgraph of Δ_i. Hence, for $n=m-1,m$:

$$|NN(u)\cap(\tau_n(u)\cap V(\Delta_i))| \geq k-1.$$

Combining this with

$$|(NN(u)-NN_j(u))\cap V(\Delta_i)| \geq |NN(u)\cap(\tau_n(u)\cap V(\Delta_i))|$$

we obtain

$$|(NN(u)-NN_j(u))\cap V(\Delta_i)| \geq k-1 \text{ for all } j=1,\ldots,m. \quad \square$$

(5.21) Corollary. If u and v are nodes in the convex subgraph Δ_i ($i\geq k^2m+1$) of T_k^m ($k\geq 2$, $m\geq 2$), then $\kappa_{\Delta_i}(u,v)$ is optimal.

Proof. Corollary 5.19 and lemma 5.20 directly imply that $\kappa_{\Delta_i}(u,v)$ is optimal. $\square$

(5.22) Lemma. If u and v are nodes in the convex subgraph Δ_i ($1\leq i\leq m.(k+1)$) of T_k^m ($k\geq 2$, $m\geq 2$), then $\kappa_{\Delta_i}(u,v)$ is optimal.

Proof. There is only one non-border node in Δ_i, since $i\leq m(k+1)$. Hence, at least one of u and v is a border node. Assume without loss of generality that u is a border node.
Then, for all $j\in\{1,\ldots,|D(u,v)|\}$: $|N_j(u)\cap V(\Delta_i)|=1$.
Hence, $\min(|N_j(u)\cap V(\Delta_i)|,|N_j(v)\cap V(\Delta_i)|)-1=0$, implying that

$$|(NN(u)-NN_i(u))\cap V(\Delta_i)| \geq \min(|N_i(u)\cap V(\Delta_i)|,|N_i(v)\cap V(\Delta_i)|)-1,$$

for all $j=1,\ldots,|D(u,v)|$. Then, $\kappa_{\Delta_i}(u,v)$ is optimal by corollary 5.19. $\square$

Lemma 5.20 and corollary 5.21 give rise to the following corollary.

(5.23) Corollary. Let u and v be nodes in the convex subgraph Δ_i ($i\leq m.(k+1)$ or $i\geq k^2m+1$) of T_k^m ($k\geq 2$, $m\geq 2$), then $\kappa_{\Delta_i}(u,v)$ is optimal.

This corollary immediately implies the following.

- Any two nodes in $\Delta_i=\beta_j^m$ ($j=1,2,\ldots$) have optimal local connectivity, where

  $$i=m.k.\frac{(k-1)^j-1}{k-2}+m.$$

  For, if $j=1$, then $i=m(k+1)$, and if $j=2$, then $i=k^2m+m>k^2m+1$.

- Any two nodes in $\Delta_i=\alpha_{\lfloor i/m\rfloor}^{m-(i \bmod m)}\times\alpha_{\lfloor i/m\rfloor+1}^{i \bmod m}$ have optimal local connectivity if $i\geq k^2m+1$.

5.6 Concluding remarks

Local connectivities between any two nodes in Δ_i are optimal if $i\geq k^2m+1$. Thence, the sequence $(\Delta_{k^2m+1},\Delta_{k^2m+2},\Delta_{k^2m+3},\ldots.)$ is a proper choice to obtain extensible networks with optimal local connectivities. The local connectivities of the networks in the sequence $(\beta_1^m,\beta_2^m,\beta_3^m,\ldots.)$ are also optimal.
If the theorems in sections 5.3 and 5.4 are applied to the networks of Akl (see the

end of paragraph 4.3), which are defined as the Cartesian product of a complete binary tree with itself, optimal local connectivities are also achieved, provided the tree has height at least 3. The 3×3 network of Akl has also optimal connectivity, but the 7×7 network has not.

Local connectivity between two internal nodes of Δ_i ($i \geq k^2m+1$) attains the high connectivity of T_k^m. Local connectivity between any two nodes of a network in the above sequences is not as large as $\kappa(T_k^m)$, if not both of the nodes lie in the interior of the network. Unfortunately, the interior of a network constitutes only a small part of the total number of nodes. Especially the exponential character of an underlying graph reduces the interior substantially. Consequently, only a small percentage of the nodes in Δ_i have mutual local connectivity equal to $\kappa(T_k^m)$. For example, the fraction of interior nodes in β_i^m is $|V(\beta_i^m)| \,/\, |V(\beta_{i-1}^m)| \approx (k-1)^{-m}$ ($i \geq 1$). That is, for $k=3$ and $m=2$ only 25 percent of the nodes of β_i^m have mutual local connectivity 6.

PART III

Planar extensible networks

6

Supersymmetric graphs and some of their properties

To the artist the communication he offers is a by-product.

W. Somerset Maugham (1938)

6.1 Introduction

In the next three chapters our construction method will be used to design extensible networks which are planar (see appendix A) in addition to all the properties stated in paragraph 2.1. Planarity of a network eases its implementation on chips and printed circuit boards. It stimulates minimization of the number of wires that need to cross. Such crossings can easily be made on a chip, by burrowing wires down into different layers. However, crossings are expensive. They enlarge the area of a chip and complicate the design of a chip (see for example [Leiser]). For these reasons we are interested in planarity of networks.

The current chapter describes underlying graphs for planar extensible networks. We assume underlying graphs to be planar. Planarity of underlying graphs directly implies planarity of extensible networks, since any subgraph of a planar graph is planar.
We are merely interested in symmetric planar underlying graphs, and in particular in a subclass of them, the so-called supersymmetric graphs. They are introduced in paragraph 6.2 after describing some general properties of symmetric planar graphs. In paragraph 6.3 supersymmetric graphs are classified according to a theorem of Grünbaum and Shephard. Thereupon, in paragraphs 6.4 and 6.5 preparations are made to chapter 7, in which the uniform exponentialities of supersymmetric graphs are determined. In chapter 8 convex subgraphs are cut out of supersymmetric graphs, resulting in the planar extensible networks.

In chapters 6, 7, and 8 we shall often use a notion which is defined as follows.

(6.1) Definition. A circuit C in an infinite connected planar graph Γ *surrounds* a node u in Γ whenever every 1-way infinite path (see appendix A) starting in u intersects C in some node. C surrounds a subgraph Δ of Γ whenever C surrounds every node of Δ. The subgraph Δ is *properly surrounded* by C if it is surrounded by C and disjunct from C.

6.2 Symmetric planar graphs

In this paragraph the class of supersymmetric graphs, which will be used as underlying graphs, are introduced. First, we discuss some general properties of symmetric planar graphs. The symmetric planar graphs discussed here are assumed to have degree at least 3; the only infinite symmetric planar graph with degree 2, the 2-way infinite path, is of no use to us. Additionally, symmetric planar graphs are assumed to have optimal connectivity. This implies that their connectivity is at least 3. The next two lemmas give a hint about the structure of planar graphs with connectivities at least 3.

(6.2) Lemma. (See [FleImr; p.99]).
If Γ is a planar graph for which $\kappa(\Gamma)\geq 3$, then any two facial circuits of Γ have at most one edge in common.

Proof. Let F_1 and F_2 be two facial circuits in Γ which have two edges in common. If these edges are adjacent, then the situation in figure 6.1 (a) occurs (the two adjacent edges are the ones between nodes u and v).
If the edges are non-adjacent, then the situation in figure 6.1 (b) occurs.

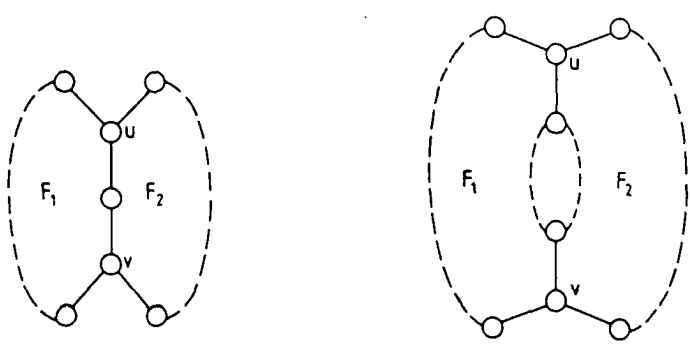


Figure 6.1 Common edges in facial circuits F_1 and F_2.

In both cases, deletion of nodes u and v disconnects Γ, contradicting with the fact that Γ has connectivity at least 3. □

In an analogous way the following lemma is proved.

(6.3) Lemma. If Γ is a planar graph for which $\kappa(\Gamma)\geq 3$, then any two nodes common to two facial circuits are adjacent, and the edge incident to the nodes is common to both circuits.

Having some knowledge about the positions of facial circuits with respect to each other in planar graphs, we are interested in the facial circuits themselves, and in particular in the number of different kinds of facial circuits that may occur in symmetric planar graphs. It will appear that for any $g\geq 3$ there exists a symmetric planar graph with facial circuits of length g. There is not much variety in the length of facial circuits in one particular symmetric planar graph, however. To show this we consider a lemma about edge-transitive graphs.

(6.4) Lemma. Let Γ be a planar edge-transitive graph with $\deg^-(\Gamma)\geq 2$, then

1. All facial circuits of Γ are of at most two different lengths.
2. If Γ contains a node of odd degree, then all facial circuits of Γ have the same length.

Proof.

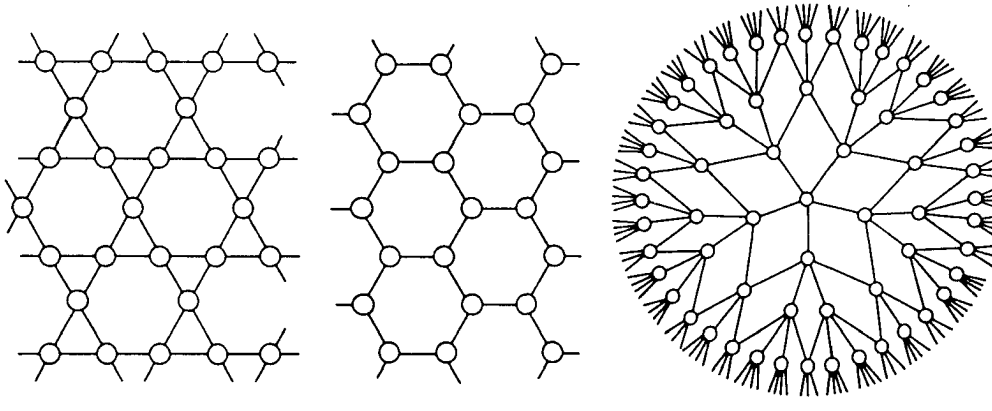
1. Suppose Γ contains facial circuits of at least three different lengths. Then, there exist facial circuits F_1 and F_2 of different lengths, having an edge e_1 in common. But also, there exist facial circuits F_3 and F_4 of different lengths with a common edge e_2 such that the length of F_3 is different from the lengths of F_1 and F_2. This implies that e_2 cannot be mapped onto e_1, contradicting with the edge-transitivity of Γ.
2. Let there be facial circuits of two different lengths and suppose v is a node in Γ with odd degree. Then there exist at least two adjacent facial circuits of the same length, both containing v. However, the edge lying on both circuits cannot be mapped onto an edge lying in two adjacent facial circuits of different length. This contradicts with the edge-transitivity of Γ. □

Since symmetric graphs are edge-transitive, the following holds for a symmetric planar graph Γ with optimal connectivity and degree at least 3.

- All nodes of Γ have the same degree.

- Every two facial circuits in Γ have at most one edge in common.
- If two facial circuits in Γ have two nodes in common, then these nodes are adjacent and the edge incident to the nodes is common to both circuits.
- The facial circuits in Γ are of at most two different lengths.
- If $\deg(\Gamma)$ is odd, then all facial circuits in Γ have the same length.

Figures 6.2 (a), (b) and (c) show some infinite locally finite symmetric planar graphs.



(a) (b) (c)

Figure 6.2 Infinite locally finite symmetric planar graphs.

The 2-dimensional mesh and the infinite tree T_k, described in chapter 4, are other examples of symmetric planar graphs.
Symmetric planar graphs of which all facial circuits have the same length will be called *supersymmetric graphs.* Figures 6.2 (b) and (c) depict such graphs. Clearly, supersymmetric graphs are highly regular, which is the reason for their name. We are interested in supersymmetric graphs with finite facial circuit length. This length is called the *girth* of a supersymmetric graph. The girth of a supersymmetric graph will be denoted by g and the degree of the graph by k. A supersymmetric graph with parameters g and k will be denoted by S_{gk}.

In order to be suitable candidates for underlying graphs, supersymmetric graphs should be exponential. Though exponentiality of supersymmetric graphs will be investigated in chapter 7, we give an impression about their exponentiality in this paragraph.
Consider supersymmetric graphs of degree 3. The two extremes of these graphs are S_{63} and $S_{\infty 3}$ ($g<6$ is impossible if $k=3$, as will be shown in paragraph 6.3).

The hexagonal grid, S_{63}, is not exponential. The infinite tree $S_{\infty 3}$, however, is exponential. The important difference between them is their girth. Apparently, increase of the girth results in exponentiality. For supersymmetric graphs with degree 3, a girth equal to 7 is sufficient to effect exponentiality - the uniform exponentiality of S_{73} will appear to be about 1.55603. Exponentiality of S_{73} may also be concluded from the space deficiencies of the outer nodes in figure 6.3. S_{73} can not be embedded in the Euclidian plane without meeting 'space problems'.

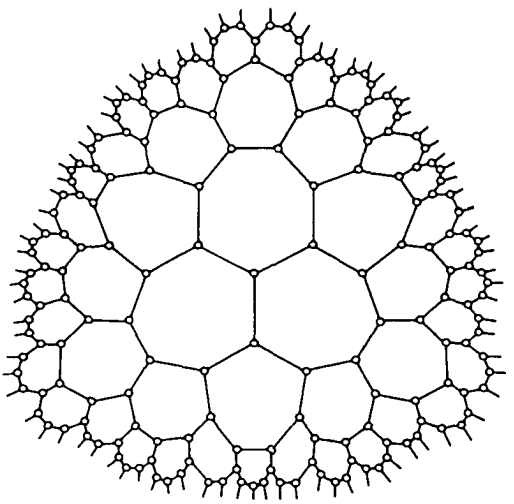


Figure 6.3 S_{73}.

The supersymmetric graph S_{83} can be obtained from Escher's picture at the cover of this dissertation by placing nodes on all points adjoining to three different colours, and connecting them by edges according to figure 6.4.

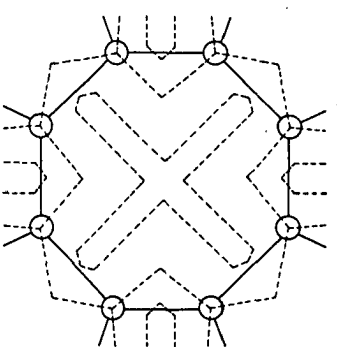


Figure 6.4 Construction of S_{83} from Escher's Circle limit II.

S_{83} also suffers from space deficiencies, which indicates that it is exponential. Indeed, $\overline{exp}(S_{83}) \approx 1.72208$.

The observation of space deficiencies is of course not a formal proof that a graph is exponential. The graph in figure 6.5 indicates that some care must be exercised to conclude exponentiality from space deficiencies. This graph cannot be embedded into the Euclidian plane, but embedding it into 3-dimensional Euclidian space raises no problems.

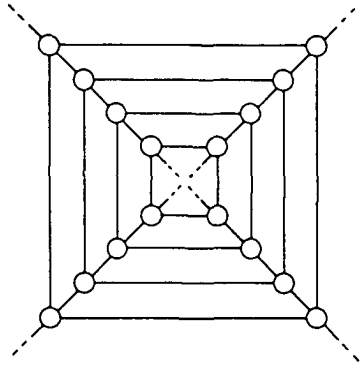


Figure 6.5 A non-exponential planar graph.

An opposite observation can be made without risk, however: if a graph can be embedded in the Euclidian plane without space problems, then it is not exponential. For this reason, we directly conclude that S_{63} (the hexagonal grid), S_{44} (the 2-dimensional mesh), and S_{36} (the triangular grid) are not exponential.

6.3 Classification of supersymmetric graphs

In this paragraph supersymmetric graphs are classified by using a theorem of Grünbaum and Shephard, described in [GruShe]. In their book, Grünbaum and Shephard encounter supersymmetric graphs in the context of tilings. We can establish a one-to-one relation between tilings and supersymmetric graphs by making each tile uniquely correspond to a facial circuit.

Before stating the theorem of Grünbaum and Shephard, we formally define the absence of space problems of a graph when drawn on the Euclidian plane.

(6.5) Definition. A planar graph is *uniformly bounded* if there exist two fixed positive numbers p and P for which the graph can be drawn on a Euclidian plane, such that each face contains a circle of radius p and is contained in a circle of radius P.

(6.6) Theorem. (Grünbaum, Shephard; 1987).
For every pair of positive integers g and k with

(6.7) $1/g + 1/k \leq 1/2$

there exists a supersymmetric graph with girth g and degree k. Such a graph is only uniformly bounded if equality holds in 6.7, that is if (g,k) is (3,6), (4,4) or (6,3). If g and k do not satisfy inequality 6.7, then no supersymmetric graph with girth g and degree k exists.

Proof. See [GruShe; 4.7.1]. □

In order to prove that a supersymmetric graph with parameters g and k exists, Grünbaum and Shephard give an explicit construction for it. The construction is used in the next paragraph, a reason to describe it here.

First, the case $g=3$, $k\geq 6$ is considered. With a node v as centre, draw circles C_1, C_2, C_3,... of radii 1, 2, 3,..., construct k nodes on C_1 and connect them to v by edges. Each of the circle segments between two nodes on C_1 will be considered as an edge.
Suppose the construction has proceeded as far as determining all the vertices and edges in the closed disk determined by the circle C_i. Let X_i respectively W_i be the set of nodes on C_i which are connected to 1 respectively 2 nodes on C_{i-1} by edges ($i\geq 2$). Let X_1 consist of all nodes in C_1 and let W_1 be empty. Let $x_i = |X_i|$, $w_i = |W_i|$ ($i\geq 1$).
Now, construct

$$w_{i+1} = x_i + w_i$$

nodes on C_{i+1} and join each of them by edges to two adjacent nodes on C_i, such that each node on C_i is connected to two nodes on C_{i+1} and no edges intersect. Thereupon construct for each node u in X_i $k-5$ nodes on C_{i+1} lying between the two nodes on C_{i+1} adjacent to u, and join them by edges to u. In an analogous way, for each node u in W_i we construct $k-6$ nodes on C_{i+1} lying between the two nodes on C_{i+1} adjacent to u and connect the $k-6$ nodes to u. The number of nodes introduced in the latter two stages is

$$x_{i+1} = (k-5)x_i + (k-6)w_i.$$

Each of the line segments and arcs of C_{i+1} determined by the nodes we have constructed will be an edge of the graph. This process goes on indefinitely whenever $k\geq 6$.

For $g=3$ and $k=7$ this process results in the construction in figure 6.6.

In the case $g\geq 4$ again circles C_1, C_2, C_3,... are drawn around a node v, and k

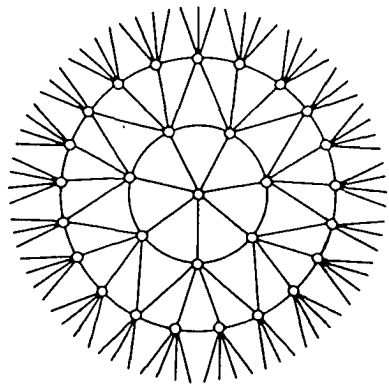


Figure 6.6 Construction of S_{37}.

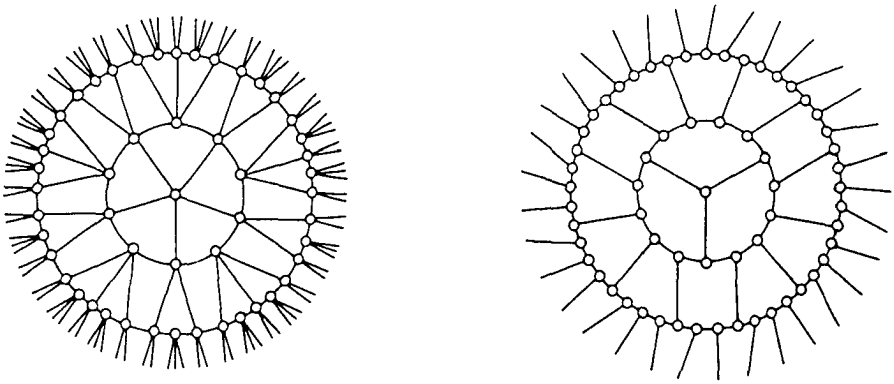
nodes on C_1 are constructed and connected by edges to v. Let X_1 be the set of this nodes. Thereupon, between each two neighbouring nodes in X_1 we construct $g-3$ nodes on C_1. The resulting $k.(g-3)$ nodes on C_1 constitute the set Y_1. Each of the circle segments between two nodes on C_1 will be considered as an edge. Suppose the construction has proceeded as far as determining all the vertices and edges in the closed disk determined by the circle C_i. Let X_i be the set of nodes on C_i that are joined by edges to nodes on C_{i-1}, and Y_i be the set of nodes on C_i not so joined ($i \geq 2$). Let $x_i = |X_i|$, $y_i = |Y_i|$ for $i \geq 1$.
Then construct for each node u in X_i respectively Y_i, $k-3$ respectively $k-2$ nodes on C_{i+1}, and connect each member of such a group to u by edges so that no edges intersect. The group members lie consecutively on C_{i+1}. This results in

$$x_{i+1} = (k-3)x_i + (k-2)y_i$$

nodes on C_{i+1}. Next, we choose $g-4$ or $g-3$ nodes on C_{i+1} between each two adjacent nodes of the x_{i+1} so far constructed. It is done in such a way that each of the x_{i+1} regions lying between C_i and C_{i+1} has g nodes.
It is easily seen that the number of nodes introduced at this stage is

$$y_{i+1} = (g-3)(x_{i+1} - (x_i + y_i)) + (g-4)(x_i + y_i).$$

As in the previous case each of the line-segments and arcs of C_{i+1} determined by the nodes constructed will be edges of the graph. This process goes on indefinitely whenever $g \geq 4$ and $k \geq 4$ or $g \geq 6$ and $k \geq 3$.
Figure 6.7 illustrates the graphs S_{45} and S_{73} constructed in this way. The reader is invited to compare these graphs with the graphs in figures 6.2 (c) and 6.3 respectively.



(a) S_{45} (b) S_{73}

Figure 6.7 Constructions of Grünbaum and Shephard.

Up to this point we didn't make profound use of the interpretation of Grünbaum and Shephard of supersymmetric graphs as tilings. Inasmuch as supersymmetric graphs with finite length facial circuits can be viewed as a tiling with finite tiles, they have infinite node- and edge-coherency (see appendix A). For, a plane tiled with finite tiles, each node being incident with but finitely many tiles, can only be split into two disjunct infinite subplanes by removing infinitely many tiles. From this we immediately conclude by using theorem 2.15, that supersymmetric graphs have optimal connectivity.

6.4 Extreme nodes

Having characterized supersymmetric graphs we are appointed to the task of determining their uniform exponentiality. For this purpose we shall draft a system of recurrence equations in chapter 7 to express the number of nodes at distance d from some node v in a supersymmetric graph as function of the number of nodes at distance $d-1$ from v. The reason that a system of equations rather than a single equation must be drafted is that not all nodes can be dealt with identically. To cope with this, the nodes are grouped into classes; nodes within one class 'behave' in a uniform way with respect to v. Each equation describes the number of nodes in one of the classes at distance d from v.
Every node will be marked by a combination of labels according to some algorithm. Each combination is the representative of a class. There appears to be only a limited number of classes for supersymmetric graphs. To determine all possible combinations of labels we need to do some work in advance. This work is done in the remainder of this chapter. The actual computation of the uniform

exponentialities of supersymmetric graphs will be done in chapter 7.
In the current paragraph we consider a property of supersymmetric graphs which will appear to be very useful in the deduction of all possible combinations of labels on nodes in supersymmetric graphs. This property is called smoothness and is described formally in the following definition.

(6.8) Definition. Let u and v be nodes in a graph Γ, then u is an *extreme node with respect to* v, if u has no neighbour w in Γ lying at distance $d(u,v)+1$ from v. A node will be called an *extreme node* in short if it is an extreme node with respect to some node in Γ. A *smooth* graph is a graph without extreme nodes.

Trivially, each component of a smooth graph has infinitely many nodes. Figure 6.8 illustrates a (symmetric) graph with extreme nodes. The black nodes are extreme with respect to the node v.

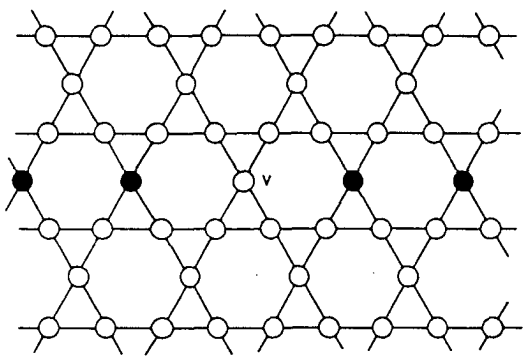


Figure 6.8 Extreme nodes in a symmetric graph.

In this paragraph we prove the following theorem.

(6.9) Theorem. Supersymmetric graphs are smooth.

The proof of this theorem is subdivided in the following cases:

A. case $g=3, k\geq 6$.

B. case $g\geq 4, k\geq 4$.

C. case $g\geq 6, k=3$.

   1. subcase $g\geq 7, k=3$.

   2. subcase $g=6, k=3$.

By theorem 6.6 these cases cover all admitted g,k-values of supersymmetric graphs. We shall first prove some lemmas for these cases and then give the proof

of theorem 6.9.
The general strategy followed in all cases is to prove that for each two nodes v and u in a supersymmetric graph Γ there exists a node x in Γ such that $d(v,x)=d(v,u)+1$. In all cases we assume Γ to consist of concentric circles C_1, C_2, C_3,..., around a node v with cross-connections as defined by the constructions of Grünbaum and Shephard in theorem 6.6.

A. case $g=3$, $k\geq 6$

(6.10) Lemma. If u is a node on C_i ($i\geq 1$) in S_{3k} ($k\geq 6$), then $d(u,v)=i$.

Proof. The construction of Grünbaum and Shephard yields immediately that each node on C_i is adjacent to (and only to) nodes on C_{i-1}, C_i and C_{i+1} ($i\geq 2$). Hence, a shortest path from v to u hits each C_j for $1\leq j\leq i$ exactly once. □

For the remaining cases (($g\geq 4$, $k\geq 4$) and ($g\geq 6$, $k=3$)) we first prove an additional lemma.

(6.11) Lemma. Let u_i, w_i be nodes on C_i, and u_{i+1}, w_{i+1} be nodes on C_{i+1} ($i\geq 1$) in S_{gk} ($g\neq 3$), such that $(u_i,u_{i+1}),(w_i,w_{i+1})\in E(\Gamma)$. Let P_{i+1} be a shortest path on C_{i+1} between u_{i+1} and w_{i+1}. Then there exists a shortest path P_i on C_i between u_i and w_i such that each node on C_{i+1} adjacent to a node on P_i lies on P_{i+1}.

Proof. Let Q_j be the alternative path between u_j and w_j on C_j ($j=i$ or $i+1$). From the construction for the case $g\neq 3$ in theorem 6.6 we directly conclude that the situation in figure 6.9 will not arise, unless the length of P_{i+1} equals the length of Q_{i+1}.

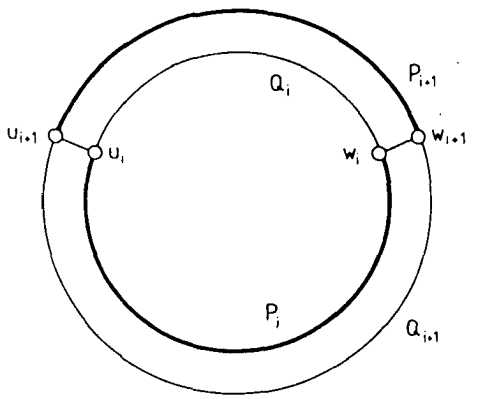


Figure 6.9 Situation occurring in lemma 6.11.

In that case, however, the length of P_i is equal to the length of Q_i, so that their

names can be interchanged. Hence, the shortest paths P_i and P_{i+1} pass via the 'same sides' of C_i and C_{i+1} respectively. □

B. case $g \geq 4$, $k \geq 4$

(6.12) Lemma. Let u_i and w_i be nodes on C_i and u_{i+1} and w_{i+1} be nodes on C_{i+1} ($i \geq 1$) in S_{gk} ($g \geq 4$, $k \geq 4$), such that $(u_i,u_{i+1}),(w_i,w_{i+1}) \in E(\Gamma)$. Then,

$$d_{C_{i+1}}(u_{i+1},w_{i+1}) \geq d_{C_i}(u_i,w_i).$$

Proof. Let $d = d_{C_i}(u_i,w_i)$.
If $d=1$, then $d_{C_{i+1}}(u_{i+1},w_{i+1}) \geq 1$, otherwise there would be a node on C_{i+1} adjacent to two nodes on C_i.
If $d>1$, there lie at least $d-1$ nodes between u_i and w_i on C_i, thence the number of nodes on C_{i+1} between u_{i+1} and w_{i+1} is at least $(d-1).(k-3)$. (Notice that lemma 6.11 is used implicitly here). From this we immediately deduce that $d_{C_{i+1}}(u_{i+1},w_{i+1}) \geq (d-1).(k-3)+1 \geq d$. □

(6.13) Lemma. Let u and w be two nodes on C_i ($i \geq 1$) in S_{gk} ($g \geq 4$, $k \geq 4$), then there is no shortest path between them through C_j for $j>i$.

Proof. Define $n := \max \{ j \mid$ shortest path between u and w passes through $C_j \}$.
Suppose $n>i$, and let P be a shortest path between u and w passing through C_n. Let u_n and w_n be nodes on $C_n \cap P$ having neighbours u_{n-1} and w_{n-1} respectively on $C_{n-1} \cap P$.
As a matter of fact such nodes u_n and w_n exist. The previous lemma yields $d_{C_n}(u_n,w_n) \geq d_{C_{n-1}}(u_{n-1},w_{n-1})$ from which we conclude that the path from u_{n-1} to w_{n-1} via C_n is a roundabout route. This is in contradiction with the definition of P. □

(6.14) Lemma. If $u_i \in C_i$ ($i \geq 1$) in S_{gk} ($g \geq 4$, $k \geq 4$), then $d(v,u_{i+1}) = d(v,u_i)+1$ for each $u_{i+1} \in C_{i+1}$ for which $(u_i,u_{i+1}) \in E(\Gamma)$.

Proof. Let $d = d(v,u_i)$ and suppose $d(v,u_{i+1}) \leq d$ for a neighbour u_{i+1} of u_i on C_{i+1}. Let P be a shortest path from v to u_{i+1}. Then P does not pass through C_j for $j>i+1$, because of the previous lemma, and P does not pass through u_i, because $d(v,u_i)=d$. Suppose P lies on C_{i+1} from u_{i+1} up to a node w_{i+1}. Then there is a node w_i on $C_i \cap P$ adjacent to w_{i+1}.
Since w_{i+1} lies on P and P is a shortest path, the following condition holds:

$$d(v,w_{i+1}) + d(w_{i+1},u_{i+1}) = d(v,u_{i+1}).$$

In addition, $d(u_{i+1},w_{i+1}) = d_{C_{i+1}}(u_{i+1},w_{i+1})$, giving

$$d(v,w_{i+1}) \leq d - d_{C_{i+1}}(u_{i+1},w_{i+1}).$$

From this it follows immediately that

$$d(v,w_i) \leq d - d_{C_{i+1}}(u_{i+1},w_{i+1}) - 1.$$

Furthermore, by Cauchy-Schwartz

$$d(v,u_i) \leq d(v,w_i) + d_{C_i}(w_i,u_i).$$

Combining the last two formulas we obtain

$$d(v,u_i) \leq d(v,w_i) + d_{C_i}(w_i,u_i) \leq$$
$$d - d_{C_{i+1}}(u_{i+1},w_{i+1}) - 1 + d_{C_{i+1}}(w_{i+1},u_{i+1}) = d - 1,$$

which is a contradiction. □

C. case $g \geq 6$, $k = 3$

Supersymmetric graphs of degree 3 need a special treatment, because some of the nodes on C_i have no neighbours on C_{i+1}.
We remark the following:

1. Of each two adjacent nodes on C_i at least one has a neighbour on C_{i+1} ($i \geq 1$).
2. Two kinds of facial circuits between C_i and C_{i+1} can be distinguished ($i \geq 1$):
   a. Circuits with 3 nodes on C_i and $g-3$ nodes on C_{i+1}.
   b. Circuits with 2 nodes on C_i and $g-2$ nodes on C_{i+1}.

   We shall denote them by *type a* and *type b* circuits respectively.

(6.15) Lemma. Let u_i, w_i be nodes on C_i, and u_{i+1}, w_{i+1} be nodes on C_{i+1} ($i \geq 1$) in S_{g3} ($g \geq 6$), such that $(u_i,u_{i+1}),(w_i,w_{i+1}) \in E(\Gamma)$. Let P_i be a shortest path on C_i between u_i and w_i and let r be the number of nodes on P_i which have no neighbours on C_{i+1}. Then,

$$d_{C_{i+1}}(u_{i+1},w_{i+1}) \geq (d_{C_i}(u_i,w_i) - 2.r)(g-3) + r(g-4)$$

and

$$r \leq d_{C_i}(u_i,w_i)/2.$$

Proof. There are $d_{C_i}(u_i,w_i) - 2.r$ facial circuits of type b between C_i and C_{i+1} having 2 nodes in common with P_i. Each of them contributes $g-2-1$ to $d_{C_{i+1}}(u_{i+1},w_{i+1})$. Furthermore, there are r facial circuits of type a between C_i and C_{i+1} having 3 nodes in common with P_i, each of them contributing $g-3-1$ to $d_{C_{i+1}}(u_{i+1},w_{i+1})$ (lemma 6.11 is used implicitly here). The first formula can

immediately be deduced from this.
For the second part we notice that of each two adjacent nodes on C_i at least one has a neighbour on C_{i+1} (remark 1). Hence, each node on C_i having no neighbour on C_{i+1} is adjacent to a node on C_i which has a neighbour on C_{i+1}. So, $r \leq d_{C_i}(u_i,w_i) / 2$. □

An easy calculation shows that

$$d_{C_{i+1}}(u_{i+1},w_{i+1}) \geq \frac{g-4}{2}.d_{C_i}(u_i,w_i)$$

for $r \geq 1$, and

$$d_{C_{i+1}}(u_{i+1},w_{i+1}) \geq (g-3).d_{C_i}(u_i,w_i)$$

otherwise. Using this relations, we can easily prove the following two lemmas.

(6.16) Lemma. Let u and w be two nodes on C_i ($i \geq 1$) in S_{g3} ($g \geq 6$), then there is no shortest path between them through C_j for $j > i$.

Proof. Identical to the proof of lemma 6.13. □

For nodes on C_i having a neighbour on C_{i+1} we have the following lemma.

(6.17) Lemma. If $u_i \in C_i$ has a neighbour $u_{i+1} \in C_{i+1}$ ($i \geq 1$) in S_{g3} ($g \geq 6$), then $d(v,u_{i+1}) = d(v,u_i)+1$.

Proof. Analogously to the proof of lemma 6.14. □

For the nodes on C_i having *no* neighbours on C_{i+1} we need to distinguish two subcases.

1. subcase $g \geq 7$.
2. subcase $g = 6$.

In both subcases we assume the remarks stated at the beginning of case C to be valid.

C1. subcase $g \geq 7$, $k = 3$

(6.18) Lemma. No two circuits of type a between C_i and C_{i+1} lie side by side in S_{g3} ($g \geq 7$), i.e. two circuits of type a have no edge in common between C_i and C_{i+1} ($i \geq 1$).

Proof. There are 3 nodes on C_1 having no neighbour in C_2, thence 3 circuits of type 1 lie between C_1 and C_2. It is easily verified that they don't lie side by side. For the case $i>1$ let F_1 and F_2 be two circuits of type a lying side by side between C_i and C_{i+1}. Let m_j be the middlemost node of the 3 nodes of F_j on C_i $(j=1,2)$, then it has no neighbours on C_{i+1}, otherwise F_j wouldn't be facial. Hence, m_j has a neighbour on C_{i-1}. Let's call it m_j' (see figure 6.10).

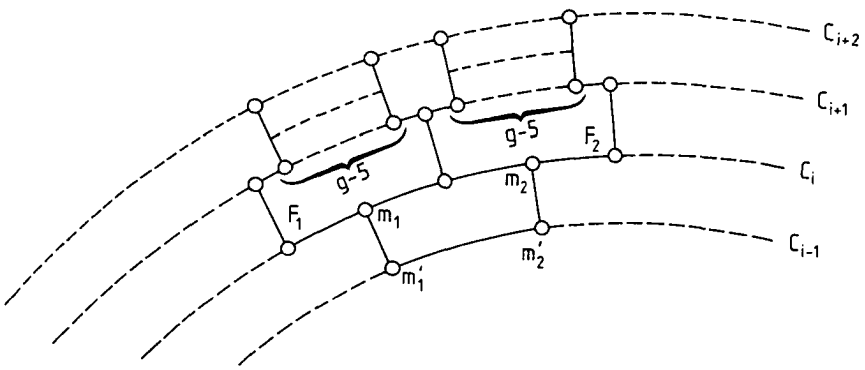


Figure 6.10 Situation occurring in lemma 6.18.

Clearly, there are no nodes between m_1' and m_2' on C_{i-1} with neighbours on C_i. Hence, the circuit between C_i and C_{i-1} containing m_1, m_2, m_1' and m_2' is facial. This circuit contains only 3 nodes on C_i. If it is a circuit of type a, then $g-3=3$, which is impossible. If it is a circuit of type b, then $g-2=3$, which is also impossible. We conclude that F_1 and F_2 don't lie side by side. □

(6.19) Lemma. If $u_i \in C_i$ has no neighbours on C_{i+1} $(i \geq 1)$ in S_{g3} $(g \geq 7)$, then there exists a node $x_i \in C_i$ for which $(x_i,u_i) \in E(\Gamma)$ and $d(v,x_i)=d(v,u_i)+1$.

Proof. The case $i=1$ is easily verified.
For the case $i>1$ we notice that u_i is part of two facial circuits between C_i and C_{i-1}. The previous lemma implies that at least one of them contains $g-2$ nodes on C_i. Let F be a circuit satisfying this, and let x_i be the neighbour of u_i on C_i lying on F. Let u_{i-1} be the neighbour of u_i on C_{i-1}.
Let $d=d(v,u_i)$ and suppose $d(v,x_i) \leq d$. This causes a contradiction in the following way. Let P be a shortest path between x_i and v. It does not intersect C_j for $j>i$ because of lemma 6.16. P crosses C_i from x_i through at least one node which has a neighbour in C_{i-1}. Let y_i be the unique node on $P \cap C_i$ with a neighbour on C_{i-1}, y_i lying at minimal distance from x_i $(y_i \neq u_i)$. Clearly, y_i lies on F and $d(x_i,y_i)=g-4$ (see figure 6.11). Since P is a shortest path, we obtain

$$d(v,y_i) = d(v,x_i) - d(x_i,y_i) \leq d-(g-4).$$

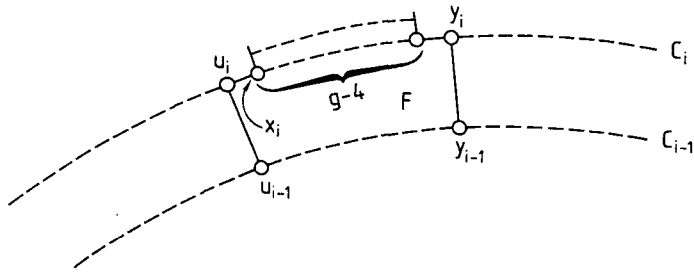


Figure 6.11 Situation occurring in lemma 6.19.

Lemma 6.17 implies $d(v,y_{i-1}) \leq d-(g-3)$ and $d(v,u_{i-1}) = d-1$. Hence,

$$d(u_{i-1},y_{i-1}) \geq d(v,u_{i-1}) - d(v,y_{i-1}) \geq d-1-(d-(g-3)) = g-4.$$

Each node between u_{i-1} and y_{i-1} on C_{i-1} has no neighbours on C_i, because F is facial. Then, by remark 1 there exists at most 1 node on C_{i-1} between u_{i-1} and y_{i-1} having no neighbours on C_i. Hence, $d(u_{i-1},y_{i-1}) \leq 2$ implying $g-4 \leq 2$, which is a contradiction. □

C2. subcase $g=6$, $k=3$

To cope with the subcase $g=6$, first some general lemmas must be proved.

(6.20) Lemma. Let Γ be an infinite locally finite planar graph in which all facial circuits have finite even length. Then all finite circuits have even length.

Proof. Suppose Γ contains an finite odd circuit C. Then C is not a facial circuit. Hence, there exist two nodes u and w on C connected by a path P which is surrounded (see definition 6.1) by C. P divides C into an odd and an even circuit, i.e. one of the circuits being a proper subgraph of C∪P has odd length and the length of the other is even. Repeating the division of the odd circuit, results again in an odd and an even circuit. Since C surrounds only a finite number of facial circuits, this process must stop at some moment, resulting in an odd facial circuit. This is a contradiction. □

This lemma will also be used in the proof of lemma 6.33.

(6.21) Lemma. Let x, y and z be nodes in an infinite locally finite planar graph with facial circuits of even length, z being a neighbour of x, then $d(y,z) \neq d(y,x)$.

Proof. Suppose $d(y,z)=d(y,x)$. Let P_z and P_x denote the shortest paths from y to z and x respectively. Let y' be the unique node in $P_z \cap P_x$ having the maximum distance to y ($P_z \cap P_x \neq \emptyset$ since it contains y). Let P_z' be the part of P_z lying between y' and z, and let P_x' be defined analogously. Then, the circuit $y'P_z'zxP_x'y'$ has odd length, which is a contradiction. □

(6.22) Corollary. If x is extreme with respect to y in the previous lemma, then $d(y,z)=d(y,x)-1$.

This corollary will be used in lemma 6.23, which is tailored to the subcase $k=3$, $g=6$. The hexagonal grid distinguishes itself by the other supersymmetric graphs of degree 3 in that two circuits of type a lying between C_i and C_{i+1} are allowed to lie side by side (see figure 6.12).

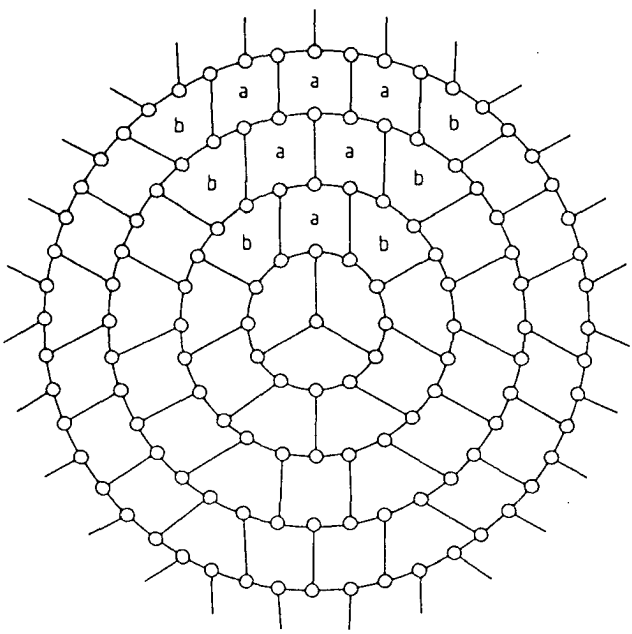


Figure 6.12 Type a and b circuits in the hexagonal grid.

(6.23) Lemma. If the node $u_i \in C_i$ has no neighbours on C_{i+1} ($i \geq 1$) in S_{63}, then $d(v,x_i)=d(v,u_i)+1$ for each neighbour x_i of u_i on C_i.

Proof. The case $i=1$ is easily verified.
For the case $i>1$ let F_i be the facial circuit between C_i and C_{i+1} containing u_i. Then, F_i is a circuit of type a and u_i is the middlemost node of the 3 nodes on

$F_i \cap C_i$.

Let $d=d(v,u_i)$, and suppose x_i is a neighbour of u_i on C_i not lying at distance $d+1$ from v. Then $d(v,x_i)=d-1$ because of corollary 6.22. On account of remark 1 x_i has a neighbour on C_{i+1}, and so, x_i has no neighbours on C_{i-1}. It is proved that the condition $d(v,x_i)=d-1$ causes a contradiction in the following way.

Let y_i be the neighbour of x_i on C_i not equal to u_i. The shortest path from x_i to v does not pass through C_j $(j>i)$ because of lemma 6.16, and it does not pass through u_i either. Hence, it passes through y_i, which implies $d(v,y_i)=d-2$. Let t_{i-1} be the neighbour of u_i on C_{i-1}. Then, $d(v,t_{i-1})=d-1$ (lemma 6.17). The facial circuit between C_{i-1} and C_i which contains u_i, x_i, y_i and t_{i-1} will be denoted by F_{i-1}.

We shall prove that F_{i-1} is a circuit of type a, i.e. that the situation in figure 6.13 occurs. If F_{i-1} is of type b, then four of its nodes lie on C_i. Then y_i does not have neighbours on C_{i-1} but y_i does have a neighbour z_i on C_i not equal to x_i, and z_i has a neighbour on C_{i-1}. We shall denote the latter by s_{i-1}.

Inasmuch as the shortest path between x_i and v passes through z_i, $d(v,z_i)=d-3$. Lemma 6.17 implies $d(v,s_{i-1})=d-4$. Nodes s_{i-1} and t_{i-1} are both part of F_{i-1}, thence $d(t_{i-1},s_{i-1})=1$, implying $d(v,t_{i-1})\leq d-4+1=d-3$, which is a contradiction. Hence, F_{i-1} is a facial circuit of type a and y_i has a neighbour on C_{i-1} (which shall be denoted by x_{i-1}).

Since F_{i-1} is of type a, there exists a node on C_{i-1} between t_{i-1} and x_{i-1}. It will be denoted by u_{i-1} (see figure 6.13). From lemma 6.17 we conclude $d(v,x_{i-1})=d-3$. Together with the knowledge that $d(v,t_{i-1})=d-1$ this implies $d(v,u_{i-1})=d-2$.

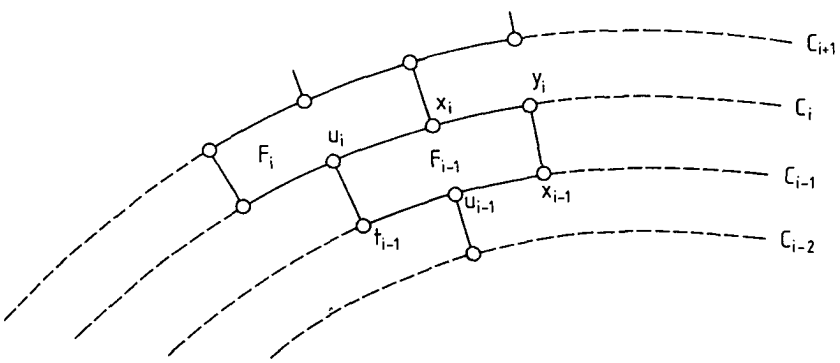


Figure 6.13 Situation occurring in lemma 6.23.

We just proved that if u_i is a node on C_i $(i\geq 2)$ without neighbours on C_{i+1} and with a neighbour x_i on C_i for which $d(v,x_i)=d(v,u_i)-1$, then there exists a node

u_{i-1} on C_{i-1} without neighbours on C_i and with a neighbour x_{i-1} on C_{i-1} for which $d(v,x_{i-1})=d(v,u_{i-1})-1$. Using Descente Infinie (see appendix A) we deduce that there exists a node u_1 on C_1 with a neighbour x_1 on C_1, for which $d(v,x_1)=d(v,u_1)-1$. Since u_1 has no neighbours on C_2, it is adjacent to v. Hence $d(v,x_1)=0$, which is a contradiction. We conclude that $d(v,x_i)=d+1$. □

At this point we can prove theorem 6.9.

Proof of theorem 6.9.
Let v be an arbitrary node in a supersymmetric graph Γ. It is considered the centre node of the circle constructions of Grünbaum and Shephard. We reconsider the main cases ($i\geq 1$):

A. case $g=3$, $k\geq 6$.
It is easily verified that each node u_i on C_i has a neighbour u_{i+1} on C_{i+1}. Lemma 6.10 implies $d(v,u_{i+1})=d(v,u_i)+1$.

B. case $g\geq 4$, $k\geq 4$.
Lemma 6.14 states that each node u_i on C_i has a neighbour u_{i+1} on C_{i+1} for which $d(v,u_{i+1})=d(v,u_i)+1$.

C. case $g\geq 6$, $k=3$.
Lemma 6.17 states that for each node u_i on C_i having a neighbour u_{i+1} on C_{i+1} the following holds: $d(v,u_{i+1})=d(v,u_i)+1$. If u_i has no neighbours on C_i, the following subcases must be considered.

C1. subcase $g\geq 7$, $k=3$.
Lemma 6.19 states that for each node u_i on C_i having no neighbours on C_{i+1} there exists a neighbour x_i of u_i on C_i for which $d(v,x_i)=d(v,u_i)+1$.

C2. subcase $g=6$, $k=3$.
Lemma 6.23 states that for each node u_i on C_i having no neighbours on C_{i+1} and each neighbour x_i of u_i on C_i the following holds: $d(v,x_i)=d(v,u_i)+1$.

In all cases we conclude that for each node u in Γ there exists a node x in Γ such that $d(v,x)=d(v,u)+1$. This implies that Γ has no extreme nodes with respect to v. Hence, Γ is smooth. □

The interest of theorem 6.9 will become clear after the next lemma, which is a powerful extension of our equipment. This lemma will be used in the remainder of this chapter and in chapters 7 and 8.

(6.24) Lemma. Let Γ be an infinite locally finite connected smooth graph and S be a separating set in Γ which separates a finite component K from Γ. Then for each node $x \in V(\Gamma) - V(K)$ and each node y in K:

$$d(x,y) < \max_{z \in S} d(x,z).$$

Proof. Suppose there exist an $x \in V(\Gamma) - V(K)$ and a $y \in V(K)$ such that

$$d(x,y) \geq \max_{z \in S} d(x,z).$$

The absence of extreme nodes in Γ implies the existence of a neighbour u_1 of y for which $d(x,u_1) = d(x,y) + 1$. Inasmuch as

$$d(x,u_1) > \max_{z \in S} d(x,z),$$

u_1 does not lie on S. Then, it lies in K since y does not lie in S. Analogously, u_1 has a neighbour u_2 for which $d(x,u_2) = d(x,u_1) + 1$, implying that u_2 lies in K. Continuing this process results in a 1-way infinite path $y,u_1,u_2,u_3,\ldots$ lying entirely in K. This causes a contradiction since K consists of only finitely many nodes. □

Lemma 6.24 is applicable to supersymmetric graphs since they are smooth. It will be used in the following way.
Let C be a non-facial finite circuit in a supersymmetric graph Γ. Since Γ is planar, C is a separating set separating a finite component K from Γ. By definition C surrounds K. Lemma 6.24 rules out all configurations which give rise to a node v in Γ and a circuit C properly surrounding a node w such that

$$d(v,w) \geq \max_{u \in V(C)} d(v,u).$$

Such configurations will frequently occur in a large number of lemmas in paragraph 6.5 and chapters 7 and 8 as a result of suppositions made in the lemmas. Then, the contradictions caused by lemma 6.24 prove that these suppositions are false.

6.5 Relative Distance labelings of smooth planar graphs

In this paragraph we define labelings for supersymmetric graphs and deduce some properties of the labelings. As was stated in the previous paragraph, these labelings will be used to differentiate between nodes in supersymmetric graphs. This is necessary because not all nodes 'behave' identically with respect to a particular node v. Labelings ease the draft of recurrence equations describing the number of nodes as function of the distance to some central node v. The labelings will also be used in chapter 8 to construct convex hulls of balls in supersymmetric graphs.

(6.25) Definition. The *Relative Distance labeling* (abbreviated as *RD-labeling*) of a planar connected graph Γ *with respect to* a node $v \in V(\Gamma)$ is a labeling which attaches to each tuple (u,F), F being a facial circuit in Γ and $u \in V(F)$, a label RD-l defined by

$$\text{RD-l}(u,F) := d(u,v) - \min_{w \in V(F)} d(w,v).$$

This label is called the *RD-label* of u in F. If $u \notin V(F)$, then RD-l(u,F) is not defined.

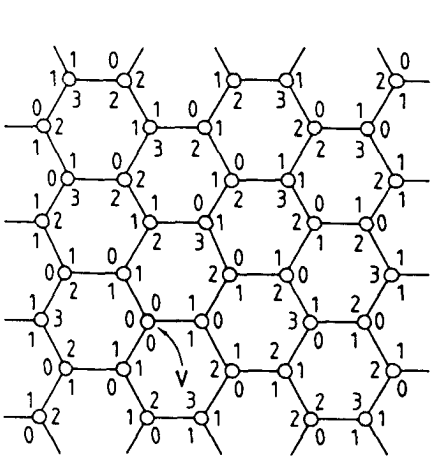
Figure 6.14 shows the RD-labelings of four planar graphs with respect to a node v.
All RD-labelings in this dissertation are labelings with respect to a particular node v. In each facial circuit F in a graph Γ, an RD-label 0 occurs at the nodes of F lying the most nearby to v. These nodes will be called the *zeros* of F. Trivially, the RD-labels of all other nodes in F are greater than 0.
We are interested in the combinations of the RD-labels attached to the nodes of supersymmetric graphs. In the graphs in figure 6.14 some combinations occur at many nodes and others do not occur at all. Which combinations occur in supersymmetric graphs will be investigated in detail in chapter 7. To determine all feasible combinations, we need to have some knowledge about the RD-labels in facial circuits. This will be investigated in the subsequence of this paragraph. We shall consider the RD-labels in the finite facial circuits of smooth planar graphs.
Let's first consider the zeros in facial circuits of smooth planar graphs.

(6.26) Lemma. If Γ is a smooth planar graph, then every two zeros of a facial circuit F in Γ are adjacent.

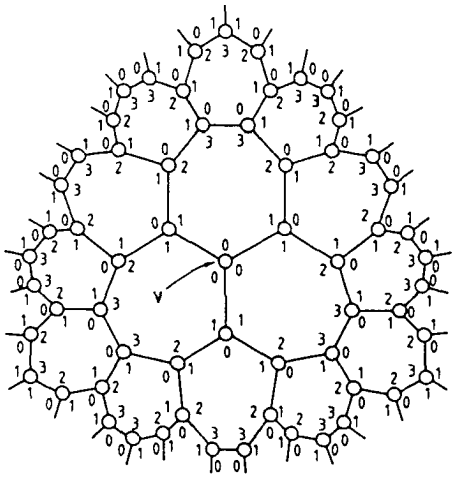
Proof. Suppose z_1 and z_2 are two non-adjacent zeros in F. Let P and Q be the two paths connecting z_1 and z_2 on F. Let G_1 and G_2 be the shortest paths from z_1 and z_2 respectively to v, and v' be the unique node on $G_1 \cap G_2$ with maximum distance from v. The subpaths of G_j from v' to z_j will be called G_j' (j=1,2). This way of constructing a node v' and subpaths G_1' and G_2' from the paths G_1 and G_2 will very frequently occur in proofs in the remainder of this chapter and in chapter 7.
Two cases can be distinguished:

1. Q is surrounded by the circuit $v'G_1'z_1Pz_2G_2'v'$.
2. P is surrounded by the circuit $v'G_1'z_1Qz_2G_2'v'$,

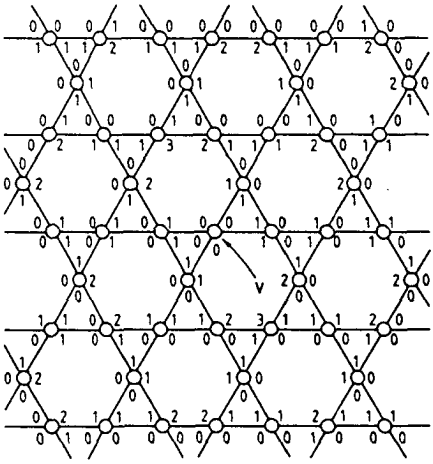
Without loss of generality we assume the first case occurs (see figure 6.15).
Let x be the node with the largest distance from v, of all the nodes on Q not equal



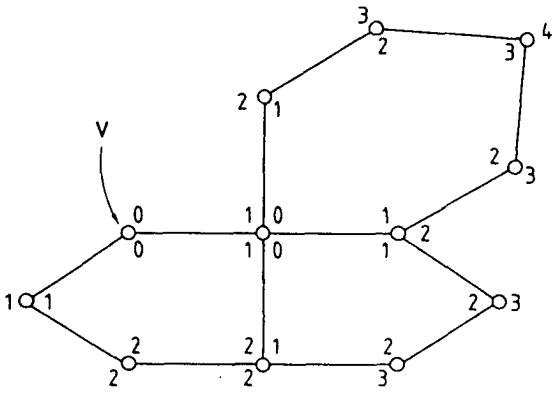
(a) S_{63}



(b) S_{73}



(c) a symmetric graph



(d) a finite planar graph

Figure 6.14 RD-labelings of four graphs with respect to node v.

to z_1 and z_2. Then, $d(v,x) \geq d(v,z_j)$ $(j=1,2)$, because z_1 and z_2 are zeros in F. Since $d(v,u) \leq d(v,z_j)$ for each node u on G_j' $(j=1,2)$, x is a node on the circuit $v'G_1'z_1Qz_2G_2'v'$ lying at maximal distance from v. Since Γ is smooth there exists a node y which is adjacent to x and for which $d(v,y)=d(v,x)+1$. This node does not lie on the circuit $v'G_1'z_1Qz_2G_2'v'$.

Since F is a facial circuit, node y is properly surrounded by the circuit

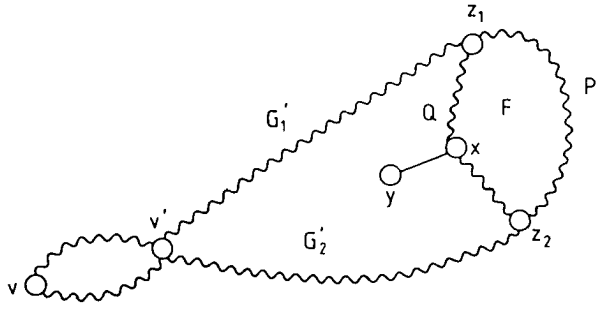


Figure 6.15 Situation occurring in lemma 6.26.

$v'G_1'z_1Qz_2G_2'v'$. Applying lemma 6.24 we obtain a contradiction. Hence, z_1 and z_2 are adjacent. □

(6.27) Lemma. If Γ is a smooth planar graph, then every facial circuit F in Γ contains at most two zeros.

Proof. If F contains more than 2 zeros, then by the previous lemma it is a triangle. Let x, y and z be the nodes of F and G_1, G_2 and G_3 respectively the shortest paths from v to them. One of the nodes of F is properly surrounded by the circuit made up by the shortest paths to the other two nodes and the edge which connects these two (see figure 6.16).

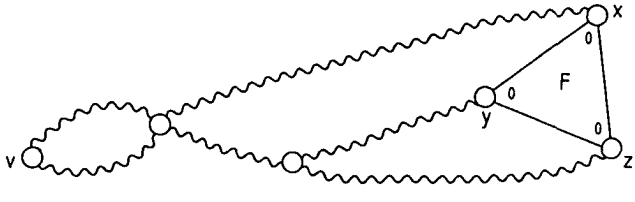


Figure 6.16 Situation occurring in lemma 6.27.

Furthermore,

$$d(v,x)=d(v,y)=d(v,z)\geq d(v,u)$$

for any node u on G_1, G_2 or G_3. Application of lemma 6.24 causes a contradiction. □

In the subsequence of this chapter and in chapters 7 and 8, the maximum RD-label in a facial circuit will be denoted by 'm'.

(6.28) Lemma. If Γ is a smooth planar graph, then each node with RD-label i, $(i \leq m-1)$ in a facial circuit F in Γ has a neighbour with RD-label $i+1$ in F.

Proof. The case $i=0$ is directly implied by lemma 6.27.
In the case $i>0$, we suppose that there exists a node y with RD-label i in F having two neighbours x_1 and x_2 with RD-labels $i-1$ or i in F. Let G_1 and G_2 be the shortest paths from v to x_1 and x_2 respectively. We define v', G_1' and G_2' in the usual way. The path on F between x_1 and x_2 not passing through y contains a node with RD-label m. This node is different from x_1 and x_2, because $i \leq m-1$. The circuit $v'G_1'x_1yx_2G_2'v'$ does not surround this node because of lemma 6.24. Hence, this circuit properly surrounds a neighbour z of y at distance $d(v,y)+1$ from v (see figure 6.17). There must be such a neighbour since Γ is smooth.

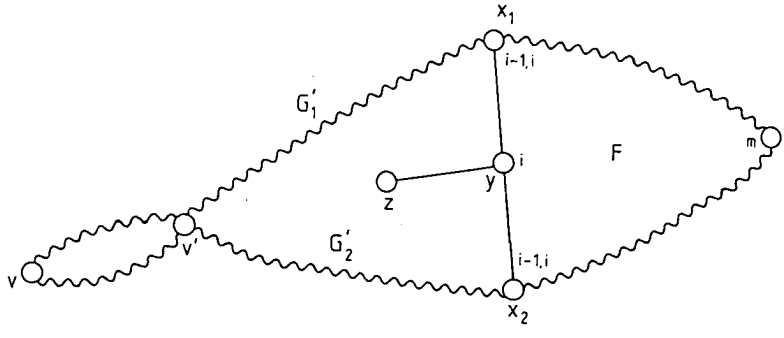


Figure 6.17 Situation occurring in lemma 6.28.

Then, application of lemma 6.24 gives a contradiction. From this contradiction we conclude that y has a neighbour with RD-label $i+1$ in F. □

(6.29) Lemma. If Γ is a smooth planar graph, then each node with RD-label i, $(1 \leq i \leq m-1)$ in a facial circuit F in Γ has a neighbour with RD-label $i-1$ in F.

Proof. Let y be a node with RD-label i in F having two neighbours x_1 and x_2 with RD-label i or $i+1$ in F. Assume that z is a zero on F and G_1 and G_2 are shortest paths from v to y and z respectively. G_1', G_2' and v' are defined in the usual way. Let P_j be the path $y, x_j, \ldots, z$ on F $(j=1,2)$. Two cases can occur:

1. P_1 is surrounded by the circuit $v'G_1'yP_2zG_2'v'$.
2. P_2 is surrounded by the circuit $v'G_1'yP_1zG_2'v'$.

Without any loss of generality assume the first case occurs (see figure 6.18).
Repeated application of lemma 6.28 yields a node on P_1 with RD-label m in F. It is properly surrounded by the circuit $v'G_1'yP_2zG_2'v'$ and its distance to v is larger

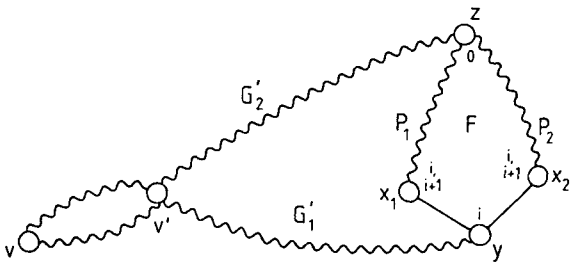


Figure 6.18 Situation occurring in lemma 6.29.

than or equal to the distance to v of any node of the circuit, which is impossible by lemma 6.24. From this contradiction we conclude that y has a neighbour with RD-label i−1 in F. □

(6.30) Lemma. If Γ is a smooth planar graph, then every two nodes with RD-label m in a facial circuit F are adjacent.

Proof. Let x and y be two non-adjacent nodes with RD-label m in F. Let P_1 and P_2 be the two paths between x and y on F and z_j be a node on P_j (j=1,2), not equal to x and y, z_j having the smallest RD-label in F of all nodes on P_j. The nodes z_1 and z_2 exist since x and y are not adjacent. Let G_j be a shortest path from v to z_j (j=1,2) and let G_1', G_2' and v' be defined in the usual way. Then, $d(u,v) \leq d(w,v)$ for any node u on G_j' and any node w on F. Furthermore, $x,y \notin G_j'$ (j=1,2). Hence, one of the following situations occurs:

1. The circuit $v'G_1'z_1$...(via F)...y...(via F)...$z_2G_2'v'$ properly surrounds node x.
2. The circuit $v'G_1'z_1$...(via F)...x...(via F)...$z_2G_2'v'$ properly surrounds node y.

Both situations are impossible by lemma 6.24. Hence, each two nodes with RD-label m in F are adjacent. □

(6.31) Lemma. If Γ is a smooth planar graph, then every facial circuit F of Γ contains at most two nodes with RD-label m.

Proof. The previous lemma implies that if F contains three nodes with RD-label m, then it is a triangle. In that case, however, all nodes in F have RD-label m, which is impossible. □

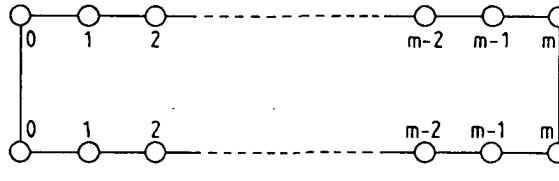
In the preceding lemmas we have proved that for every facial circuit F in a smooth graph Γ:

1. F has at most two zeros.
2. If F has two zeros, they are adjacent.
3. Each node with RD-label i ($1 \leq i \leq m-1$) in F has two neighbours with RD-labels $i-1$ and $i+1$ respectively in F.
4. F contains at most two nodes with RD-label m.
5. If F contains two nodes with RD-label m, then they are adjacent.

This immediately implies the following corollary.

(6.32) Corollary. If Γ is a smooth planar graph, then the only feasible RD-labelings of its even facial circuits are the ones in figure 6.19 and the only feasible RD-labelings of its odd facial circuits are the ones in figure 6.20 (m is the maximum RD-label in the circuit under consideration).

(a) Type I facial circuit



(b) Type II facial circuit

Figure 6.19 Admitted RD-labelings of even facial circuits in a smooth graph.

Smoothness of a graph forces the RD-labelings of its facial circuits to have the description imposed by corollary 6.32. Indeed, the smooth graphs in figures 6.14 (a) and (b) have the labelings as described in corollary 6.32. The graph in figure 6.14 (c) contains facial circuits with three zeros, proving that this graph is not smooth.
Facial circuits with one respectively two zeros will be called *type I* respectively *type II facial circuits.*

(a) Type I facial circuit

(b) Type II facial circuit

Figure 6.20 Admitted RD-labelings of odd facial circuits in a smooth graph.

(6.33) Lemma. If Γ is a smooth planar graph of which all facial circuits have even length, then the facial circuits are of type I.

Proof. Suppose F is an even facial circuit in Γ with two zeros z_1 and z_2. Let G_1 and G_2 be the shortest paths from v to z_1 and z_2 respectively. G_1', G_2' and v' are defined in the usual way. Then, the circuit $v'G_1'z_1z_2G_2'v'$ has odd length, which is in contradiction with lemma 6.20. Hence, F contains only one zero. □

(6.34) Lemma. Let Γ be a smooth planar graph of which all facial circuits have odd length, then Γ contains facial circuits of both types.

Proof. Γ contains type I facial circuits since all facial circuits containing v are of type I. Let C_1 be a type I facial circuit and u and w be the nodes with maximum RD-label in C_1. Let C_2 be the facial circuit having u and w in common with C_1. Then, $RD\text{-}l(u,C_2)=RD\text{-}l(w,C_2)=0$ or $RD\text{-}l(u,C_2)=RD\text{-}l(w,C_2)=m$, where m is the maximum RD-label in C_2.
Suppose that C_2 is of type I, i.e. $RD\text{-}l(u,C_2)=m$. Let z_1 and z_2 be the zeros in C_1 and C_2 respectively and let the shortest paths from v to them be G_1 and G_2 respectively. Define G_1', G_2' and v' in the usual way. Let P_i and Q_i be the paths on C_i from z_i to u and w respectively (i=1,2). Then there are two cases:

- The circuit $v'G_1'z_1P_1uP_2z_2G_2'v'$ properly surrounds w,
- The circuit $v'G_1'z_1Q_1wQ_2z_2G_2'v'$ properly surrounds u.

Both cases give a contradiction by lemma 6.24. Hence, RD-l(u,C_2)=RD-l(w,C_2)=0, i.e. C_2 is of type II. □

Application of corollary 6.32, lemma 6.33, and lemma 6.34 to supersymmetric graphs with girth g yields m=g/2 for even g and m=(g−1)/2 for odd g. Whenever any RD-label m is used in a supersymmetric graph in chapters 7 and 8, it is supposed to be defined in the above way. The characteristics of RD-labels within a facial circuit of a supersymmetric graph, given by corollary 6.32, lemma 6.33, and lemma 6.34, are implicitly assumed and used in chapters 7 and 8.

6.6 Concluding remarks

In this chapter we introduced an infinite class of infinite planar graphs with fixed finite degree and optimal connectivity, the class of supersymmetric graphs. These graphs are highly regular. The uniform exponentialities of these graphs are not known yet: they will be determined in the next chapter. The preliminaries for this have been made in paragraph 6.4, by considering a special property of graphs, smoothness, and in paragraph 6.5 by dealing with special labelings of planar graphs, their RD-labelings.

7

Uniform exponentialities of supersymmetric graphs

Tyger! Tyger! burning bright
In the forest of the night,
What immortal hand or eye
Could frame thy fearful symmetry?

William Blake (1757-1827)

7.1 Introduction

In this chapter the uniform exponentialities of all supersymmetric graphs are determined. This is done in two stages.
In the first stage we draft recurrence equations which describe the number of nodes at a certain distance d from a node v in a supersymmetric graph as function of d. These recurrence equations will appear to be linear equations. Consequently, the system of equations corresponding to the supersymmetric graph S_{gk} can be written as $N(d)=A_{gk}.N(d-1)$, where N(d) is a vector and A_{gk} is a matrix.
In the second stage we consider the eigenvalue of the matrix A_{gk}. The largest eigenvalue of A_{gk} will appear to be equal to the uniform exponentiality of the corresponding supersymmetric graph S_{gk}.

To draft the recurrence equations, we need to differentiate between the nodes of supersymmetric graphs. For this reason we introduced RD-labelings. The nodes are differentiated according to the combinations of RD-labelings attached to them. In general, an RD-labeling attaches deg(u) RD-labels to a node u. A rotation around u, which starts in one of the labels, results in a sequence of deg(u) consecutive RD-labels. We call such a sequence an *RD-combination* of u. Two different RD-combinations of some node which arose from choosing a different starting label or direction of rotation, will be considered identical. More formally:

(7.1) Definition. The *Relative Distance combination* (abbreviated as *RD-combination*) of a node u in a planar connected graph Γ *with respect to* a node $v \in V(\Gamma)$ is a sequence of RD-labels of u with respect to v, (RD-l(u,F_1),RD-l(u,F_2),...,RD-l(u,$F_{deg(u)}$)), such that facial circuit F_i is adjacent to F_{i+1} ($1 \leq i < deg(u)$) and F_1 is adjacent to $F_{deg(u)}$. All possible sequences of RD-labels of u defined in this way are considered to be identical, thus constituting a unique RD-combination of u. The RD-combination of u is denoted by *RD-c(u)*.

All RD-combinations discussed in this dissertation will be RD-combinations with respect to the node v. For convenience, we leave the commas and the brackets out of RD-combinations.
It is easily seen that v is (the only node) completely labeled by zeros. Hence, the RD-combination of v is always $0^{deg(v)}$ (in analogy to language theory a superscript k denotes a k-fold succession of the character being superscripted). Any neighbour of v in the graph in figure 6.14 (c) has RD-combination 0011, which is identical to 1001, 1100 and 0110 but not to 1010.
Each RD-combination will give rise to a variable denoted by 'N' with the RD-combination as subscript. We consider variables identical if the RD-combinations in their subscripts are identical, e.g. $N_{012} \equiv N_{102} \equiv N_{210}$. The number of nodes at distance d from v and labeled with a particular RD-combination *comb* will be denoted by $N_{comb}(d)$. For example, $N_{011}(1)=3$ in figures 6.14 (a) and 6.14 (b). In the recurrence equations corresponding to a supersymmetric graph $N_{comb}(d)$ will be expressed as function of $N_{comb_1}(d-1)$, $N_{comb_2}(d-1)$,..., $N_{comb_r}(d-1)$, where r is the total number of different RD-combinations occurring in the graph and 'comb' is one of these combinations. The variable r appears to be finite for supersymmetric graphs, i.e. each supersymmetric graph exhibits only a finite number of different RD-combinations.
In paragraph 7.2 we make some preparations to determine the feasible RD-combinations. In paragraph 7.3 we determine the feasible RD-combinations of the nodes in supersymmetric graphs with even girth, and draft the systems of recurrence equations corresponding to them. In paragraph 7.4 the same will be done for supersymmetric graphs with odd girth. Thereupon, in paragraph 7.5 the uniform exponentialities of supersymmetric graphs are deduced from the recurrence equations.

7.2 Preparations for the draft of recurrence equations

The examples in figures 6.14 (a) and (b) suggested that only a limited number of RD-combinations may occur in supersymmetric graphs. Indeed, most combinations are not permitted in supersymmetric graphs. We shall deduce a complete description of all feasible RD-combinations in supersymmetric graphs. To

determine all feasible RD-combinations in a particular supersymmetric graph Γ, a kind of induction method will be used. From the RD-combination of a node $u \in \Gamma_{d-1}(v)$ we determine the RD-combinations of all neighbours of u in $\Gamma_d(v)$. When started in v, this process results in the fixed set of all feasible RD-combinations in the supersymmetric graph under consideration.
During this process it is often necessary to rule out infeasible RD-combinations. We prove an RD-combination to be infeasible by considering two of its RD-labels, and proving that they can not occur together in the RD-combination. The two RD-labels considered in an RD-combination are often chosen such that they belong to two *adjacent* facial circuits. If the two RD-labels are i and j, then the resulting configuration will be denoted by *adj ij* (see figure 7.1).

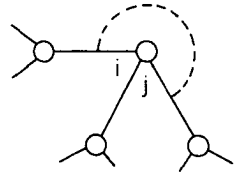


Figure 7.1 Adj ij.

Whenever we refer in this chapter to adj ij for some values of i and j, such a configuration is meant. When proving results about RD-combinations we shall often refer to pictures rather than writing long-winded stories.
The following lemma will be used very often for establishing whether an RD-combination is feasible.

(7.2) Lemma. Let x be a node in a supersymmetric graph Γ and $z_1,\ldots, z_r$ and $y_1,\ldots, y_{k-r}$ be the neighbours of x, and let a, b, c, p, q, i, i+1, j, and j+1 be RD-labels in Γ according to figure 7.2 ($1 \le i \le m-1$, $1 \le j \le m-1$, $1 \le r \le k-2$, $1 \le s \le k-r-1$).

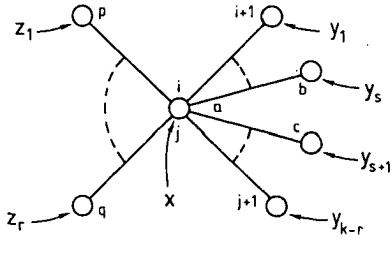


Figure 7.2 Situation occurring in lemma 7.2.

Then, a=0.

Proof. It is sufficient to prove that $b=c=a+1$.
If $s=1$, then $b=a+1$.
If $s>1$, then $b=a+1$ too. For, suppose $b\leq a$. Then, let G_1 and G_2 be the shortest paths from v to z_1 and y_s respectively. Define G_1', G_2', and v' in the usual way (see the proof of lemma 6.26). Trivially, node x does not lie on G_1 and G_2, since $d(v,x)\geq d(v,u)$ for any node u on G_1 or G_2 (notice that $p\leq i$ and $q\leq j$). This implies that $v'G_1'z_1xy_sG_2'v'$ is a circuit. It properly surrounds either y_1 or y_{k-r}, both yielding a contradiction by lemma 6.24. Hence, $b=a+1$.
In an analogous way it is proved that $c=a+1$. Hence, $a=0$. □

This lemma immediately implies the following lemma.

(7.3) Lemma. If $g>3$, then the neighbours of v have RD-combination $0^{k-2}11$.

Proof. Let $r=1$ in lemma 7.2 and z_1 be equal to v. Then, $i=j=1$ and x is a neighbour of v. Since $g>3$, $d(y_1,v)=d(y_{k-1},v)=2$, hence lemma 7.2 can be applied. It implies that $a=0$ for any a placed between y_s and y_{s+1} like in figure 7.2 ($s=1,\ldots,k-2$). Hence, $RD-c(x)=0^{k-2}11$. □

This lemma is valid for supersymmetric graphs with even girth as well as supersymmetric graphs with odd girth. The RD-combinations of nodes at distance at least 2 from v will be determined in the next two paragraphs.

7.3 Equations for supersymmetric graphs with even girth

In this paragraph the recurrence equations for supersymmetric graphs with even girth are designed. Before drafting the equations corresponding to a particular supersymmetric graph, we need to know all feasible RD-combinations in the graph. To determine all feasible RD-combinations in supersymmetric graphs with even girth, we consider the following cases:

A. case $g=4$, $k\geq 4$.

B. case $g\geq 6$, $k\geq 3$.

   1. subcase $g\geq 6$, $k\geq 4$.

   2. subcase $g\geq 6$, $k=3$.

      a. subsubcase $g\geq 8$, $k=3$.

      b. subsubcase $g=6$, $k=3$.

By theorem 6.6 these cases cover all feasible g,k-values for supersymmetric graphs with even girth.

In all the cases we assume that the RD-combinations are with respect to the node v. We know that for all cases, RD-c(v)=0^k and RD-c(w)=$0^{k-2}11$ for each neighbour w of v.

A. case g=4, k≥4

(7.4) Lemma. Adj 22 is not feasible in S_{4k} (k≥4).

Proof. If adj 22 occurs in S_{4k} (k≥4), then the situation in figure 7.3 will arise (nodes u and x_2 coincide if k=4).

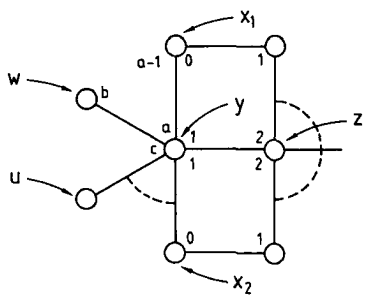


Figure 7.3 Situation occurring in lemma 7.4.

If d(v,z)=2 then x_1 and x_2 would coincide with v (and so they would also coincide with nodes w, u, etc.). In that case the degree of node y would be 2, which is a contradiction. Hence, d(v,z)>2. Let G_j be the shortest path from x_j to v (j=1,2). G_1', G_2' and v' are defined as usual. The circuit $v'G_1'x_1yx_2G_2'v'$ surrounds node w properly, because it does not surround node z properly by lemma 6.24. Applying lemma 6.24 again gives b<a, which implies a=2. In the same way it is proved that c=2.
We have proved that any subcombination adj 22 at a node at distance d from v implies the existence of the same subcombination at a node at distance d−1 from v (d≥3). Then, using Descente Infinie we obtain the subcombination adj 22 at a node at distance 2 from v. This is a contradiction. □

(7.5) Lemma. If w is a node in S_{4k} (k≥4) for which RD-c(w)=$0^{k-2}11$ and d(w,v)≥1, then w has k−1 neighbours at distance d(w,v)+1 from v, of which

- 2 have RD-combination $0^{k-3}121$,
- k−3 have RD-combination $0^{k-2}11$.

The remaining neighbour of w lies at distance d(w,v)−1 from v.

Proof. RD-c(w)=$0^{k-2}11$ induces the situation in figure 7.4. It follows directly

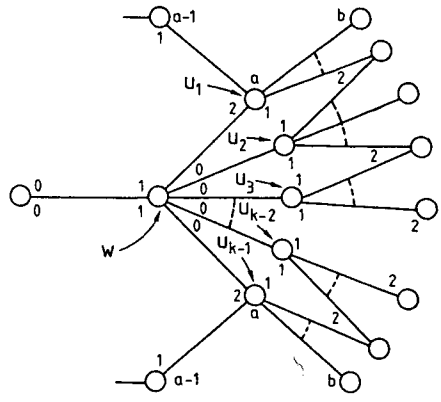


Figure 7.4 Situation occurring in lemma 7.5.

that $d(u_j,v)=d(w,v)+1$ for $j=1,\ldots,k-1$. Clearly, the remaining neighbour of w lies at distance $d(w,v)-1$ from v.
The previous lemma implies $a<2$. Furthermore, since $a-1$ is an RD-label we have $a-1\geq 0$. From this we deduce that $a=1$, and so $b=2$. Then, lemma 7.2 implies $\text{RD-c}(u_1)=\text{RD-c}(u_{k-1})=0^{k-3}121$. Furthermore, lemma 7.2 implies $\text{RD-c}(u_j)=0^{k-2}11$ for $j=2,\ldots,k-2$. □

(7.6) Lemma. If w is a node in S_{4k} ($k\geq 4$) for which $\text{RD-c}(w)=0^{k-3}121$ and $d(w,v)\geq 2$, then w has $k-2$ neighbours at distance $d(w,v)+1$ from v, of which

- 2 have RD-combination $0^{k-3}121$,
- $k-4$ have RD-combination $0^{k-2}11$.

The remaining 2 neighbours of w lie at distance $d(w,v)-1$ from v.

Proof. $\text{RD-c}(w)=0^{k-3}121$ induces the situation in figure 7.5. It follows directly that $d(u_j,v)=d(w,v)+1$ for $j=1,\ldots,k-2$. Clearly, the remaining two neighbours of w lie at distance $d(w,v)-1$ from v.
In a way similar to the previous lemma we obtain $a=1$ and $b=2$. Then, lemma 7.2 implies $\text{RD-c}(u_1)=\text{RD-c}(u_{k-2})=0^{k-3}121$. This lemma also implies $\text{RD-c}(u_j)=0^{k-2}11$ for $j=2,\ldots,k-3$. □

We have proved that RD-combinations $0^{k-2}11$ or $0^{k-3}121$ at nodes in $\Gamma_{d-1}(v)$ ($d\geq 2$) give rise to RD-combinations $0^{k-2}11$ or $0^{k-3}121$ at nodes in $\Gamma_d(v)$. Inasmuch as all neighbours of v have RD-combination $0^{k-2}11$ and these neighbours 'generate' RD-combinations $0^{k-2}11$ and $0^{k-3}121$ at distance 2 from v, the only RD-combinations occurring in supersymmetric graphs with parameters $g=4$ and $k\geq 4$ are $0^{k-2}11$ and $0^{k-3}121$.

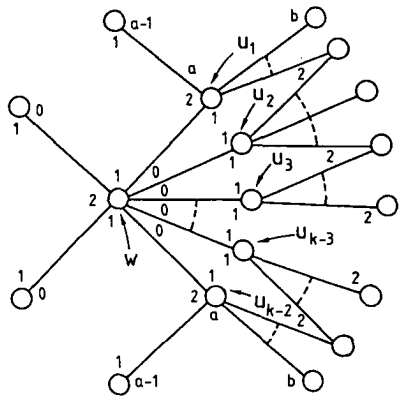


Figure 7.5 Situation occurring in lemma 7.6.

Lemmas 7.5 and 7.6 result in the following remarks.

1. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 2)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-2}11$ is equal to

   - $k-3$ if RD-c(w) $= 0^{k-2}11$,
   - $k-4$ if RD-c(w) $= 0^{k-3}121$.

   Furthermore, each node in $\Gamma_d(v)$ $(d \geq 1)$ with RD-combination $0^{k-2}11$ has exactly 1 neighbour in $\Gamma_{d-1}(v)$. Hence,

   $$N_{0^{k-2}11}(d) = (k-3).N_{0^{k-2}11}(d-1)+(k-4).N_{0^{k-3}121}(d-1).$$

2. Each node $w \in \Gamma_{d-1}(v)$ $(d \geq 2)$ has 2 neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-3}121$. Moreover, each node in $\Gamma_d(v)$ with RD-combination $0^{k-3}121$ has exactly 2 neighbours in $\Gamma_{d-1}(v)$. Hence,

   $$N_{0^{k-3}121}(d) = |\Gamma_{d-1}(v)| = N_{0^{k-2}11}(d-1)+N_{0^{k-3}121}(d-1).$$

If we define the vector N by

$$N(d) := (N_{0^{k-2}11}(d), N_{0^{k-3}121}(d))^T$$

then the system of recurrence equations can be expressed as:

$$N(d) = A_{4k}\, N(d-1) \quad (k \geq 4),$$

where

$$A_{4k} = \begin{bmatrix} k-3 & k-4 \\ 1 & 1 \end{bmatrix}$$

The computation of the uniform exponentiality of supersymmetric graphs with girth 4 and degree $k \geq 4$ from this matrix will be postponed until paragraph 7.5.

B. case $g \geq 6$, $k \geq 3$

(7.7) Lemma. If w is a node in S_{gk} ($g \geq 6$, $k \geq 3$) for which RD-c(w)$=0^{k-2}1i$ ($1 \leq i \leq m-2$) and $d(w,v) \geq 1$, then w has $k-1$ neighbours at distance $d(w,v)+1$ from v, of which

- 1 has RD-combination $0^{k-2}12$,
- 1 has RD-combination $0^{k-2}1i+1$,
- $k-3$ have RD-combination $0^{k-2}11$.

The remaining neighbour of w lies at distance $d(w,v)-1$ from v.

Proof. RD-c(w)$=0^{k-2}1i$ induces the situation in figure 7.6.

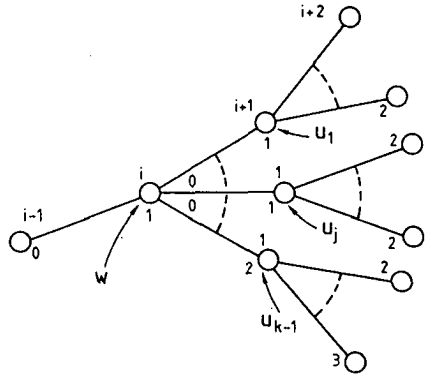


Figure 7.6 Situation occurring in lemma 7.7.

It follows directly that $d(u_j,v)=d(w,v)+1$ for $j=1,\ldots,k-1$. Clearly, the remaining neighbour of w lies at distance $d(w,v)-1$ from v.
Lemma 7.2 immediately implies RD-c(u_1)$=0^{k-2}1i+1$, RD-c(u_{k-1})$=0^{k-2}12$ and RD-c(u_j)$=0^{k-2}11$ for $j=2,\ldots,k-2$. □

(7.8) Lemma. If adj im ($2 \leq i \leq m$) occurs in S_{gk} ($g \geq 6$, $k \geq 3$) at a node at distance d ($d \geq 3$) from v then adj m−1m occurs at a node at distance d−1 from v.

Proof. If adj im occurs at a node z, then the situation in figure 7.7. will arise.
We shall prove that a=m.
If k=3, nodes w and x_1 will coincide, implying b=a−1, and so, a=m.
Otherwise, let G_j be the shortest path from x_j to v (j=1,2). G_1', G_2' and v′ are defined as usual. The circuit $v'G_1'x_1yx_2G_2'v'$ does not surround node z properly by lemma 6.24, since $d(z,v)>d(u,v)$ for any node u on the circuit. Hence, the

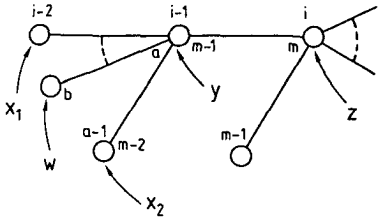


Figure 7.7 Situation occurring in lemma 7.8.

circuit surrounds node w properly. Then, lemma 6.24 implies b<a. From this we conclude that a=m. □

(7.9) Corollary. Adj im ($2 \leq i \leq m$) is infeasible in S_{gk} ($g \geq 6$, $k \geq 3$).

Proof. When i=m−1 is substituted in the previous lemma, we can prove by using Descente Infinie that adj m−1m occurs at a node at distance 2 from v, which is a contradiction, since $m \geq 3$. Then, infeasibility of adj im ($2 \leq i \leq m$) follows directly from infeasibility of adj m−1m. □

(7.10) Lemma. If w is a node in S_{gk} ($g \geq 6$, $k \geq 3$) for which RD-c(w)=$0^{k-2}1m-1$ and $d(w,v) \geq 2$, then w has k−1 neighbours at distance d(w,v)+1 from v, of which

- 1 has RD-combination $0^{k-2}12$,
- 1 has RD-combination $0^{k-3}1m1$,
- k−3 have RD-combination $0^{k-2}11$.

The remaining neighbour of w lies at distance d(w,v)−1 from v.

Proof. RD-c(w)=$0^{k-2}1m-1$ induces the situation in figure 7.8 (If k=3, node x and y coincide).
It follows directly that $d(u_j,v)=d(w,v)+1$ for $j=1,\ldots,k-1$. Clearly, the remaining neighbour of w lies at distance d(w,v)−1 from v.
By corollary 7.9, a<2. From $a-1 \geq 0$, we deduce that a=1. Hence, b=2. Then, lemma 7.2 implies RD-c(u_1)=$0^{k-3}1m1$. Furthermore, lemma 7.2 implies RD-c(u_{k-1})=$0^{k-2}12$ and RD-c(u_j)=$0^{k-2}11$ for $j=2,\ldots,k-2$. □

At this point we differentiate between two subcases.

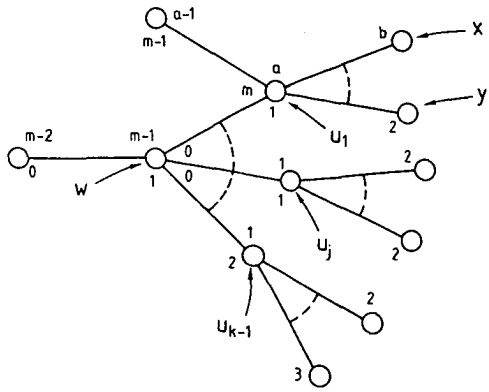


Figure 7.8 Situation occurring in lemma 7.10.

B1. subcase $g \geq 6$, $k \geq 4$

(7.11) Lemma. If w is a node in S_{gk} ($g \geq 6$, $k \geq 4$) for which RD-c(w) $= 0^{k-3}1m1$ and $d(w,v) \geq 3$, then w has $k-2$ neighbours at distance $d(w,v)+1$ from v, of which

- 2 have RD-combination $0^{k-2}12$,
- $k-4$ have RD-combination $0^{k-2}11$.

The remaining 2 neighbours of w lie at distance $d(w,v)-1$ from v.

Proof. RD-c(w) $= 0^{k-3}1m1$ induces the situation in figure 7.9.

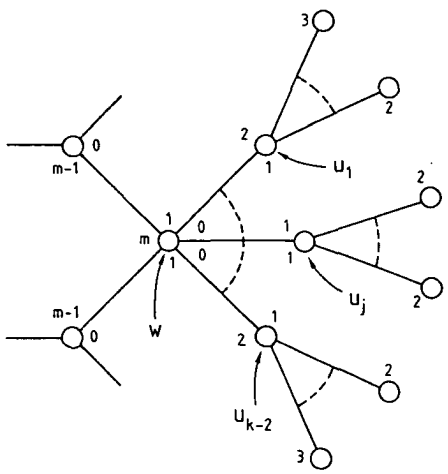


Figure 7.9 Situation occurring in lemma 7.11.

It follows directly that $d(u_j,v) = d(w,v)+1$ for $j = 1,\ldots,k-2$. Clearly, the remaining two neighbours of w lie at distance $d(w,v)-1$ from v.

Lemma 7.2 immediately implies that RD-c(u_1)=RD-c(u_{k-2})=$0^{k-2}12$ and RD-c(u_j)=$0^{k-2}11$ for j=2,...,k−3. □

Lemmas 7.7, 7.10 and 7.11 imply that RD-combinations $0^{k-2}11$, $0^{k-2}12$, $0^{k-2}13$,..., $0^{k-2}1m-1$ and $0^{k-3}1m1$ at nodes in $\Gamma_{d-1}(v)$ (d≥2) give rise to the same set of RD-combinations at nodes in $\Gamma_d(v)$. By lemmas 7.3, 7.7, 7.10 and 7.11 these RD-combinations indeed occur in supersymmetric graphs with girth g≥6 and degree k≥4.
The above lemmas result in the following remarks:

1. Each node in $\Gamma_d(v)$ (d≥1) has 1 neighbour in $\Gamma_{d-1}(v)$ except when its RD-combination is $0^{k-3}1m1$.

2. For each node w∈$\Gamma_{d-1}(v)$ (d≥2) its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-2}11$ is equal to

   - k−3 if RD-c(w)=$0^{k-2}1i$, for i=1,2,...,m−1,
   - k−4 if RD-c(w)=$0^{k-3}1m1$ (i.e. otherwise).

   Together with remark 1 this implies

   $$N_{0^{k-2}11}(d) = (k-3).N_{0^{k-2}11}(d-1) + (k-3).N_{0^{k-2}12}(d-1) + \ldots +$$
   $$(k-3).N_{0^{k-2}1m-1}(d-1) + (k-4).N_{0^{k-3}1m1}(d-1).$$

3. For each node w∈$\Gamma_{d-1}(v)$ (d≥2) its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-2}12$ is equal to

   - 2 if RD-c(w)=$0^{k-2}11$ or $0^{k-3}1m1$,
   - 1 otherwise.

   Together with remark 1 this implies:

   $$N_{0^{k-2}12}(d) = 2.N_{0^{k-2}11}(d-1) + N_{0^{k-2}12}(d-1) + \ldots + N_{0^{k-2}1m-1}(d-1) +$$
   $$2.N_{0^{k-3}1m1}(d-1).$$

4. For each node w∈$\Gamma_{d-1}(v)$ (d≥2) its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-2}1i$ (3≤i≤m−1) is equal to 1 if RD-c(w)=$0^{k-2}1i-1$ and 0 otherwise. Together with remark 1 this implies

   $$N_{0^{k-2}1i}(d) = N_{0^{k-2}1i-1}(d-1).$$

5. For each node w∈$\Gamma_{d-1}(v)$ (d≥2) its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-3}1m1$ is equal to 1 if RD-c(w)=$0^{k-2}1m-1$ and 0 otherwise. Furthermore, each node in $\Gamma_d(v)$ with RD-combination $0^{k-3}1m1$ has 2

neighbours in $\Gamma_{d-1}(v)$ with RD-combination $0^{k-2}1m-1$. Hence,

$$N_{0^{k-3}1m1}(d) = 1/2 \,.\, N_{0^{k-2}1m-1}(d-1).$$

If we define the vector N by

$$N(d) := (N_{0^{k-2}11}(d), N_{0^{k-2}12}(d), \ldots, N_{0^{k-2}1m-1}(d), N_{0^{k-3}1m1}(d))^T,$$

then the system of recurrence equations can be expressed as

$$N(d) = A_{gk}.N(d-1) \ (g \geq 6, k \geq 4),$$

where A_{gk} is an $m \times m$ matrix, equal to:

$$A_{gk} = \begin{bmatrix} k-3 & k-3 & k-3 & k-3 & \ldots & k-3 & k-3 & k-4 \\ 2 & 1 & 1 & 1 & \ldots & 1 & 1 & 2 \\ 0 & 1 & 0 & 0 & \ldots & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 & 0 & 0 \\ . & . & . & . & \ldots & . & . & . \\ . & . & . & . & \ldots & . & . & . \\ 0 & 0 & 0 & 0 & \ldots & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & \ldots & 0 & 1/2 & 0 \end{bmatrix}$$

For the uniform exponentiality of supersymmetric graphs with even girth $g \geq 6$ and degree $k \geq 4$ we refer to paragraph 7.5.

B2. subcase $g \geq 6$, $k = 3$

(7.12) Lemma. If w is a node in S_{g3} ($g \geq 6$) for which RD-c(w) = 11m and $d(w,v) \geq 3$, then w has exactly 1 neighbour at distance $d(w,v)+1$ from v, of which the RD-combination is 022. The other 2 neighbours of w have distance $d(w,v)-1$ from v.

Proof. RD-c(w) = 11m induces the situation in figure 7.10.

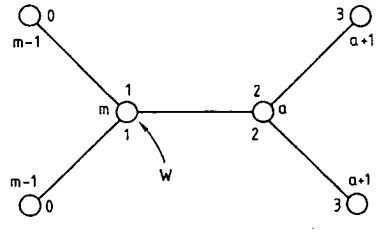


Figure 7.10 Situation occurring in lemma 7.12.

It follows directly that w has 1 neighbour at distance $d(w,v)+1$ from v and 2 neighbours at distance $d(w,v)-1$ from v. Lemma 7.2 directly implies that $a = 0$. □

Again, the hexagonal grid asks for a special treatment. First we deal with the subsubcase $g \geq 8$.

B2a. subsubcase $g \geq 8$, $k=3$

(7.13) Lemma. If $g \geq 8$ and w is a node in S_{g3} ($g \geq 8$) for which RD-c(w)=022 and $d(w,v) \geq 2$, then w has exactly 2 neighbours at distance $d(w,v)+1$ from v. Both have RD-combination 013. The remaining neighbour of w lies at distance $d(w,v)-1$ from v.

Proof. RD-c(w)=022 gives rise to the situation in figure 7.11.

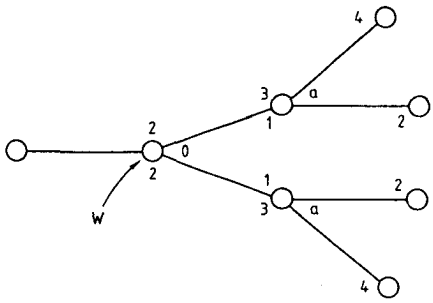


Figure 7.11 Situation occurring in lemma 7.13.

It follows directly from the figure that 2 neighbours of w lie at distance $d(w,v)+1$ from v and the other neighbour of w lies at distance $d(w,v)-1$ from v. Lemma 7.2 implies a=0. □

In supersymmetric graphs with degree 3 the only nodes with RD-combination 011 are the three neighbours of v. We have proved that RD-combinations 022, 012, 013,..., 01m−1 or 11m at nodes in $\Gamma_{d-1}(v)$ ($d \geq 3$) give rise to the same set of RD-combinations at nodes in $\Gamma_d(v)$. Inasmuch as all nodes in $\Gamma_2(v)$ have RD-combination 012 (by lemmas 7.3 and 7.7), the above RD-combinations indeed occur by lemmas 7.7, 7.10, 7.12 and 7.13.
The above lemmas result in the following remarks:

1. Each node in $\Gamma_d(v)$ ($d \geq 1$) except the ones with RD-combination 11m has 1 neighbour in $\Gamma_{d-1}(v)$.
2. For each node $w \in \Gamma_{d-1}(v)$ ($d \geq 3$) its number of neighbours in $\Gamma_d(v)$ with RD-combination 012 is equal to 1 if RD-c(w)=01i for i=2,3,...m−1 and 0 otherwise. Together with remark 1 this implies

$$N_{012}(d) = N_{012}(d-1) + N_{013}(d-1) + N_{014}(d-1) + \ldots + N_{01m-1}(d-1).$$

3. For each node $w \in \Gamma_{d-1}(v)$ ($d \geq 2$) its number of neighbours in $\Gamma_d(v)$ with RD-combination 013 is equal to

- 1 if RD-c(w)=012,
- 2 if RD-c(w)=022,

and 0 otherwise. Together with remark 1 this implies

$$N_{013}(d) = N_{012}(d-1) + 2.N_{022}(d-1).$$

4. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 2)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 01i, $4 \leq i \leq m-1$, is equal to 1 if RD-c(w)=01i−1 and 0 otherwise. Together with remark 1 this implies

$$N_{01i}(d) = N_{01i-1}(d-1), \text{ for } 4 \leq i \leq m-1.$$

5. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 2)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 11m is equal to 1 if RD-c(w)=01m−1 and 0 otherwise. Furthermore, each node in $\Gamma_d(v)$ with RD-combination 11m has 2 neighbours in $\Gamma_{d-1}(v)$ with RD-combination 01m−1. Hence,

$$N_{11m}(d) = 1/2 \,.\, N_{01m-1}(d-1).$$

6. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 2)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 022 is 1 if RD-c(w)=11m and 0 otherwise. Together with remark 1 this implies

$$N_{022}(d) = N_{11m}(d-1).$$

If we define the vector N by

$$N(d) := (N_{012}(d), N_{013}(d), \ldots, N_{01m-1}(d), N_{11m}(d), N_{022}(d))^T,$$

and the system of recurrence equations in the usual way, then we obtain the $m \times m$ matrix A_{g3} $(g \geq 8)$, equal to

$$A_{g3} = \begin{bmatrix} 1 & 1 & 1 & \ldots & 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & \ldots & 0 & 0 & 0 & 2 \\ 0 & 1 & 0 & \ldots & 0 & 0 & 0 & 0 \\ \cdot & \cdot & \cdot & \ldots & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \ldots & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & \ldots & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & \ldots & 0 & 0 & 1 & 0 \end{bmatrix}$$

For the uniform exponentiality of supersymmetric graphs with even girth $g \geq 8$ and degree 3 we refer to paragraph 7.5.

B2b. subsubcase $g=6$, $k=3$

(7.14) Lemma. If w is a node in S_{63} for which RD-c(w)=022 and $d(w,v) \geq 2$, then w has exactly 2 neighbours at distance d(w,v)+1 from v. Both have RD-combination 113. The other neighbour of w has distance d(w,v)−1 from v.

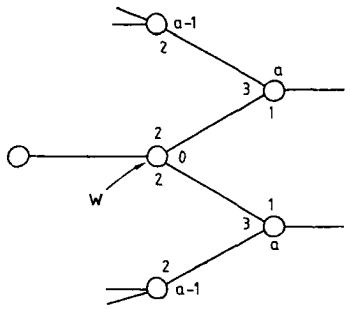


Figure 7.12 Situation occurring in lemma 7.14.

Proof. If RD-c(w)=022, then the situation in figure 7.12 arises.
It follows directly that w has 2 neighbours at distance d(w,v)+1 from v and 1 neighbour at distance d(w,v)−1 from v.
In the same way as in the proof of lemma 7.5 we deduce that a=1. □

We have proved that RD-combinations 012, 022 or 113 at nodes in $\Gamma_{d-1}(v)$ ($d \geq 3$) give rise to the same set of RD-combinations at nodes in $\Gamma_d(v)$. Inasmuch as all nodes in $\Gamma_2(v)$ have RD-combination 012 (by lemmas 7.3 and 7.7) lemmas 7.10, 7.12 and 7.14 imply that the above RD-combinations indeed occur in S_{63}.
The above lemmas result in the following remarks:

1. Each node in $\Gamma_d(v)$ ($d \geq 1$) with RD-combination 012 or 022 has 1 neighbour in $\Gamma_{d-1}(v)$.

2. For each node $w \in \Gamma_{d-1}(v)$ ($d \geq 3$) its number of neighbours in $\Gamma_d(v)$ with RD-combination 012 is 1 if RD-c(w)=012 and 0 otherwise. Together with remark 1 this implies

$$N_{012}(d) = N_{012}(d-1).$$

3. Each node in $\Gamma_{d-1}(v)$ ($d \geq 3$) with RD-combinations 012 respectively 022 has 1 neighbour respectively 2 neighbours in $\Gamma_d(v)$ with RD-combination 113. Each node in $\Gamma_d(v)$ with RD-combination 113 has 2 neighbours in $\Gamma_{d-1}(v)$. Each of them has RD-combination 012 or 022. We conclude that

$$N_{113}(d) = 1/2 \,.\, N_{012}(d-1) + N_{022}(d-1).$$

4. For each node $w \in \Gamma_{d-1}(v)$ ($d \geq 2$) its number of neighbours in $\Gamma_d(v)$ with RD-combination 022 is 1 if RD-c(w)=113 and 0 otherwise. Together with remark 1 this implies

$$N_{022}(d) = N_{113}(d-1).$$

If we define the vector N by

$$N(d) := (N_{012}(d), N_{113}(d), N_{022}(d))^T,$$

and the system of recurrence equations in the usual way, then we obtain the matrix A_{63}, equal to

$$A_{63} = \begin{bmatrix} 1 & 0 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

To prove what we already knew, i.e. that the uniform exponentiality of the hexagonal grid is 1, we refer to paragraph 7.5.

7.4 Equations for supersymmetric graphs with odd girth

To design recurrence equations for supersymmetric graphs with odd girth we use an analogous method as the method in the previous paragraph. The situation for supersymmetric graphs with odd girth is more complex, however, because there exist two types of facial circuits, type I and type II facial circuits. We differentiate between the circuits by underlining the RD-labels in type II facial circuits. Underlined labels can just be handled as conventional numbers, i.e. they can be added, subtracted, etc. For convenience we assume they can also be added to conventional numbers. This results in underlined numbers, i.e. $\underline{1} + 1 = \underline{2}$. The absolute value operator will scrape away the underline, for example $|\underline{2}| = 2$.

To determine all feasible RD-combinations in supersymmetric graphs with odd girth, we consider the following cases:

A. case $g=3, k\geq 6$.

B. case $g=5, k\geq 4$.

C. case $g\geq 7, k\geq 3$.

   1. subcase $g\geq 7, k\geq 4$.

   2. subcase $g\geq 7, k=3$.

      a. subsubcase $g>7, k=3$.

      b. subsubcase $g=7, k=3$.

By theorem 6.6 these cases cover all feasible g,k-values for supersymmetric graphs with odd girth.
For these cases we need a lemma like lemma 7.2. Since some RD-labels may be underlined in supersymmetric graphs with odd girth, we shall use a counterpart of lemma 7.2. It is exactly the same as lemma 7.2 except that the RD-labels p, q, i, i+1, j, and j+1 may be replaced by their underlined equivalents, and i is

limited according to $\underline{0} \leq i \leq \underline{m-1}$ if i is underlined, and j is limited according to $\underline{0} \leq j \leq \underline{m-1}$ if j is underlined. The conclusion remains unchanged, i.e. $a=0$. When referring to the counterpart of lemma 7.2 we shall simply refer to lemma 7.2.

A. case $g=3$, $k \geq 6$

For this case we refer to the construction of Grünbaum and Shephard and to lemma 6.10. From the former we have

$$x_{i+1} = (k-5).x_i + (k-6).w_i$$

$$w_{i+1} = x_i + w_i$$

Lemma 6.10 implies $|\Gamma_i(v)| = x_i + w_i$. Writing the vector $(x_d, w_d)^T$ as $N(d)$ we obtain

$$N(d) = A_{3k} \,.\, N(d-1), \quad (k \geq 6),$$

where

$$A_{3k} = \begin{bmatrix} k-5 & k-6 \\ 1 & 1 \end{bmatrix}$$

For the exponentiality of supersymmetric graphs with girth 3 and degree $k \geq 6$, we refer to paragraph 7.5.

B. case $g=5$, $k \geq 4$

(7.15) Lemma. If w is a node in S_{5k} $(k \geq 4)$ for which RD-c(w) $= 0^{k-2}11$ and $d(w,v) \geq 1$, then w has $k-1$ neighbours at distance $d(w,v)+1$ from v, of which

- 2 have RD-combination $\underline{0}0^{k-3}12$,
- $k-3$ have RD-combination $0^{k-2}11$.

The remaining neighbour of w lies at distance $d(w,v)-1$ from v.

Proof. RD-c(w) $= 0^{k-2}11$ induces the situation in figure 7.13.
It follows directly that $d(u_j,v) = d(w,v)+1$ for $j=1,\ldots,k-1$. Clearly, the remaining neighbour of w lies at distance $d(w,v)-1$ from v.
Lemma 7.2 implies RD-c(u_1) = RD-c$(u_{k-1}) = \underline{0}0^{k-3}12$ and RD-c$(u_j) = 0^{k-2}11$ for $j=2,\ldots,k-2$. □

(7.16) Lemma. If w is a node in S_{5k} $(k \geq 4)$ for which RD-c(w) $= \underline{0}0^{k-3}12$ and $d(w,v) \geq 2$, then w has $k-2$ neighbours at distance $d(w,v)+1$ from v, of which

- 1 has RD-combination $\underline{0}0^{k-3}12$,

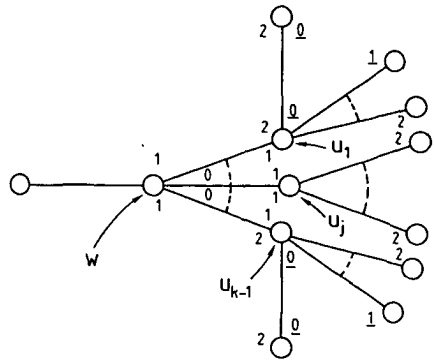


Figure 7.13 Situation occurring in lemma 7.15.

- 1 has RD-combination $0^{k-2}1\underline{1}$,
- $k-4$ have RD-combination $0^{k-2}11$.

One of the remaining two neighbours of w lies at distance $d(w,v)-1$ from v and one lies at distance $d(w,v)$ from v.

Proof. RD-$c(w)=\underline{0}0^{k-3}12$ induces the situation in figure 7.14.

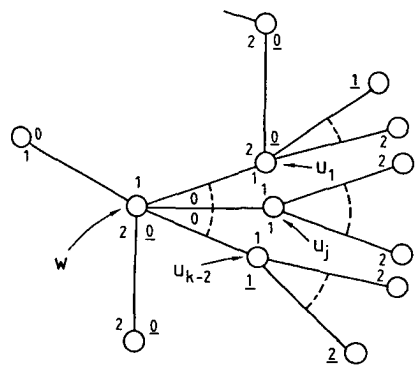


Figure 7.14 Situation occurring in lemma 7.16.

It follows directly that $d(u_j,v)=d(w,v)+1$ for $j=1,\ldots,k-2$. Clearly, 1 of the remaining 2 neighbours of w lies at distance $d(w,v)-1$ from v, and the other lies at distance $d(w,v)$ from v.
Lemma 7.2 implies RD-$c(u_1)=\underline{0}0^{k-3}12$, RD-$c(u_{k-2})=0^{k-2}1\underline{1}$ and RD-$c(u_j)=0^{k-2}11$ for $j=2,\ldots,k-3$. □

To investigate the next RD-combination we need some lemmas first.

(7.17) Lemma. If adj i2 (i=2 or i=$\underline{2}$) occurs at a node in S_{5k} (k≥4) at distance d (d≥3) from v then adj $\underline{22}$ occurs at a node at distance d−1 from v.

Proof. Adj i$\underline{2}$ induces the situation illustrated by figure 7.15 (if k=4 then u and w coincide).

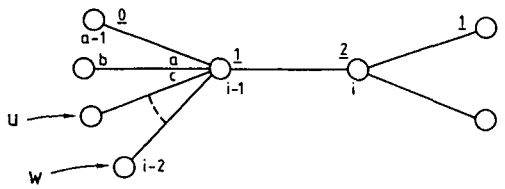


Figure 7.15 Situation occurring in lemma 7.17.

We obtain a=$\underline{2}$ and c=$\underline{2}$ by applying lemma 6.24 in the usual way (see for example proofs of lemmas 7.4 and 7.8). □

(7.18) Corollary. Adj $\underline{22}$ is infeasible in S_{5k} (k≥4).

Proof. When i=$\underline{2}$ is substituted in the previous lemma, Descente Infinie yields adj $\underline{22}$ at distance 2 from v. However, then there exists a type II facial circuit containing v, which is a contradiction. □

(7.19) Corollary. Adj 2$\underline{2}$ is infeasible in S_{5k} (k≥4).

(7.20) Lemma. Adj $\underline{12}$ is infeasible in S_{5k} (k≥4).

Proof. Adj $\underline{12}$ results in the situation illustrated by figure 7.16 (if k=4 then u and w coincide).

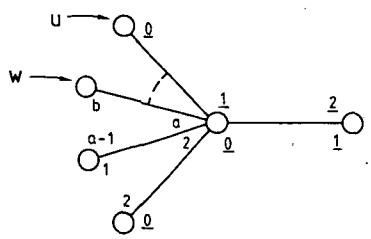


Figure 7.16 Situation occurring in lemma 7.20.

We obtain b<a in the usual way by lemma 6.24. Hence, a=$\underline{2}$. This results in the infeasible combination adj 2$\underline{2}$. □

We can now investigate the third RD-combination for the case g=5, k≥4.

(7.21) Lemma. If w is a node in S_{5k} ($k \geq 4$) for which RD-c(w)$=0^{k-2}1\underline{1}$ and $d(w,v) \geq 1$, then w has $k-1$ neighbours at distance $d(w,v)+1$ from v, of which

- 1 has RD-combination $\underline{0}0^{k-3}12$,
- 1 has RD-combination $0^{k-3}1\underline{2}1$,
- $k-3$ have RD-combination $0^{k-2}11$.

The other neighbour of w lies at distance $d(w,v)-1$ from v.

Proof. RD-c(w)$=0^{k-2}1\underline{1}$ induces the situation in figure 7.17.

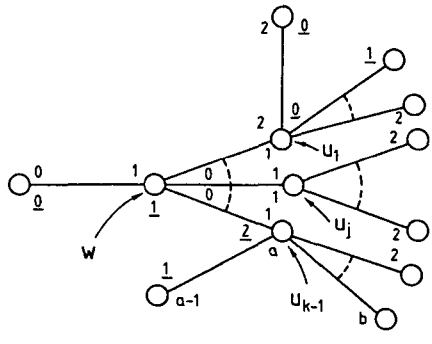


Figure 7.17 Situation occurring in lemma 7.21.

It follows directly that $d(u_j,v)=d(w,v)+1$ for $j=1,\ldots,k-1$. Clearly, the remaining neighbour of w lies at distance $d(w,v)-1$ from v.
Corollaries 7.18 and 7.19 imply $|a|<2$. Since $|a-1| \geq 0$ we obtain $a=1$ or $a=\underline{1}$. The latter is impossible by lemma 7.20. Hence, $a=1$ and $b=2$. Then, by lemma 7.2 RD-c(u_1)$=\underline{0}0^{k-3}12$, RD-c(u_{k-1})$=0^{k-3}1\underline{2}1$ and RD-c(u_j)$=0^{k-2}11$ for $j=2,\ldots,k-2$. □

(7.22) Lemma. If w is a node in S_{5k} ($k \geq 4$) for which RD-c(w)$=0^{k-3}1\underline{2}1$, then w has $k-2$ neighbours at distance $d(w,v)+1$ from v, of which

- 2 have RD-combination $\underline{0}0^{k-3}12$,
- $k-4$ have RD-combination $0^{k-2}11$.

The other neighbour of w lies at distance $d(w,v)-1$ from v.

Proof. RD-c(w)$=0^{k-3}1\underline{2}1$ induces the situation in figure 7.18.
It follows directly that $d(u_j,v)=d(w,v)+1$ for $j=1,\ldots,k-2$. Clearly, the remaining 2 neighbours of w lie at distance $d(w,v)-1$ from v.

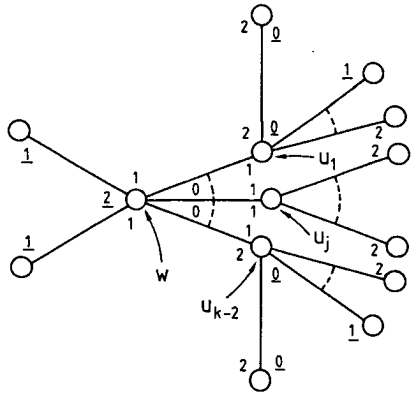


Figure 7.18 Situation occurring in lemma 7.22.

Lemma 7.2 implies RD-c(u_1)=RD-c(u_{k-2})=$\underline{0}0^{k-3}12$ and RD-c(u_j)=$0^{k-2}11$ for $j=2,\ldots,k-3$. □

Lemmas 7.15, 7.16, 7.21 and 7.22 imply that RD-combinations $0^{k-2}11$, $\underline{0}0^{k-3}12$, $0^{k-2}1\underline{1}$ and $0^{k-3}1\underline{2}1$ at nodes in $\Gamma_{d-1}(v)$ ($d\geq 2$) give rise to the same set of RD-combinations at nodes in $\Gamma_d(v)$. By lemmas 7.3, 7.15, 7.16, 7.21 and 7.22 these RD-combinations indeed occur in supersymmetric graphs with girth $g=5$ and degree $k\geq 4$.
The above lemmas result in the following remarks.

1. Each node in $\Gamma_d(v)$ ($d\geq 1$) has 1 neighbour in $\Gamma_{d-1}(v)$ except when it has RD-combination $0^{k-3}1\underline{2}1$.
2. For each node $w\in\Gamma_{d-1}(v)$ ($d\geq 2$) its number of neighbours in $\Gamma_d(v)$ with RD-combinations $0^{k-2}11$ is equal to
   - $k-3$ if RD-c(w)=$0^{k-2}11$ or RD-c(w)=$0^{k-2}1\underline{1}$,
   - $k-4$ if RD-c(w)=$\underline{0}0^{k-3}12$ or RD-c(w)=$0^{k-3}1\underline{2}1$.

   Together with remark 1 this implies

$$N_{0^{k-2}11}(d) = (k-3).N_{0^{k-2}11}(d-1) + (k-4).N_{\underline{0}0^{k-3}12}(d-1) + (k-3).N_{0^{k-2}1\underline{1}}(d-1) + (k-4).N_{0^{k-3}1\underline{2}1}(d-1).$$

3. For each node $w\in\Gamma_{d-1}(v)$ ($d\geq 2$) its number of neighbours in $\Gamma_d(v)$ with RD-combination $\underline{0}0^{k-3}12$ is equal to
   - 2 if RD-c(w)=$0^{k-2}11$ or RD-c(w)=$0^{k-3}1\underline{2}1$,
   - 1 if RD-c(w)=$\underline{0}0^{k-3}12$ or RD-c(w)=$0^{k-2}1\underline{1}$.

Together with remark 1 this implies

$$N_{\underline{0}0^{k-3}12}(d) = 2.N_{0^{k-2}11}(d-1) + N_{\underline{0}0^{k-3}12}(d-1) + N_{0^{k-2}1\underline{1}}(d-1) + 2.N_{0^{k-3}12\underline{1}}(d-1).$$

4. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 3)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-2}1\underline{1}$ is equal to 1 if RD-c(w)$=\underline{0}0^{k-3}12$ and 0 otherwise. Together with remark 1 this implies

$$N_{0^{k-2}1\underline{1}}(d) = N_{\underline{0}0^{k-3}12}(d-1).$$

5. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 2)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-3}12\underline{1}$ is equal to 1 if RD-c(w)$=0^{k-2}1\underline{1}$ and 0 otherwise. Furthermore, each node in $\Gamma_d(v)$ with RD-combination $0^{k-3}12\underline{1}$ has 2 neighbours in $\Gamma_{d-1}(v)$ with RD-combination $0^{k-2}1\underline{1}$. Hence,

$$N_{0^{k-3}12\underline{1}}(d) = 1/2 \,.\, N_{0^{k-2}1\underline{1}}(d-1).$$

If we define the vector N by

$$N(d) := (N_{0^{k-2}11}(d), N_{\underline{0}0^{k-3}12}(d), N_{0^{k-2}1\underline{1}}(d), N_{0^{k-3}12\underline{1}}(d))^T,$$

and the system of recurrence equations in the usual way, we obtain the 4×4 matrix A_{5k} $(k \geq 4)$ equal to:

$$A_{5k} = \begin{bmatrix} k-3 & k-4 & k-3 & k-4 \\ 2 & 1 & 1 & 2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1/2 & 0 \end{bmatrix}$$

For the exponentiality of supersymmetric graphs with girth 5 and degree $k \geq 4$ we refer to paragraph 7.5.

C. case $g \geq 7$, $k \geq 3$

(7.23) Lemma. If w is a node in S_{gk} $(g \geq 7, k \geq 3)$ for which RD-c(w)$=0^{k-2}1i$ $(1 \leq |i| \leq m-2)$ and $d(w,v) \geq 1$, then w has $k-1$ neighbours at distance $d(w,v)+1$ from v, of which

- 1 has RD-combination $0^{k-2}12$,
- 1 has RD-combination $0^{k-2}1i+\underline{1}$,
- $k-3$ have RD-combination $0^{k-2}11$.

(Notice that i may be an underlined RD-label). The remaining neighbour of w lies at distance $d(w,v)-1$ from v.

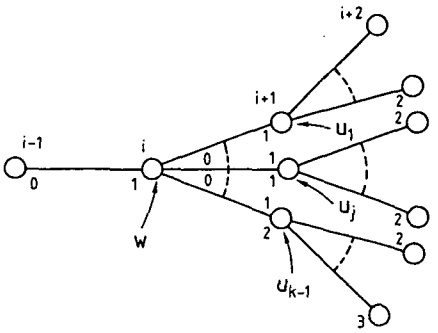


Figure 7.19 Situation occurring in lemma 7.23.

Proof. RD-c(w)=0^{k-2}1i induces the situation in figure 7.19.
It follows directly that $d(u_j,v)=d(w,v)+1$ for $j=1,\ldots,k-1$. Clearly, the remaining neighbour of w lies at distance $d(w,v)-1$ from v.
Lemma 7.2 implies RD-c(u_1)=0^{k-2}1i+1, RD-c(u_j)=0^{k-2}11 for $j=2,\ldots,k-2$ and RD-c(u_{k-1})=0^{k-2}12. □

(7.24) Lemma. If w is a node in S_{gk} ($g\geq 7$, $k\geq 3$) for which RD-c(w)=0^{k-2}1m−1 and $d(w,v)\geq 2$, then w has k−1 neighbours at distance $d(w,v)+1$ from v, of which

- 1 has RD-combination 0^{k-2}12,
- 1 has RD-combination $\underline{0}0^{k-3}$1m,
- k−3 have RD-combination 0^{k-2}11.

The remaining neighbour of w lies at distance $d(w,v)-1$ from v.

Proof. RD-c(w)=0^{k-2}1m−1 induces the situation in figure 7.20.
It follows directly that $d(u_j,v)=d(w,v)+1$ for $j=1,\ldots,k-1$. Clearly, the remaining neighbour of w lies at distance $d(w,v)-1$ from v.
Lemma 7.2 implies RD-c(u_1)=$\underline{0}0^{k-3}$1m, RD-c(u_{k-1})=0^{k-2}12 and RD-c(u_j)=0^{k-2}11 for $j=2,\ldots,k-2$. □

To investigate the RD-combination $0^{k-2}1\underline{m-1}$ in S_{gk} ($g\geq 7$, $k\geq 3$), we need some lemmas first.

(7.25) Lemma. If adj i$\underline{m}$ ($2\leq|i|\leq m$ and i may be an underlined RD-label) occurs in S_{gk} ($g\geq 7$, $k\geq 3$) at a node at distance d ($d\geq 3$) from v, then adj $\underline{m-1}$m occurs at a node at distance d−1 from v.

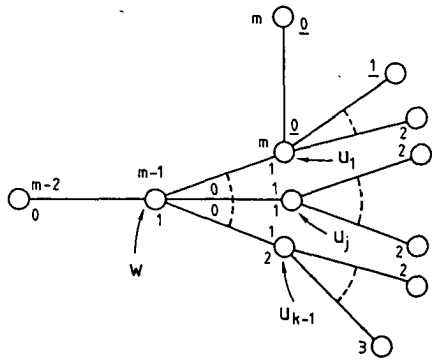


Figure 7.20 Situation occurring in lemma 7.24.

Proof. Similar to proof of lemma 7.8. □

(7.26) Lemma. Adj im ($2 \leq |i| \leq m$; i may be an underlined RD-label) is infeasible in S_{gk} ($g \geq 7$, $k \geq 3$).

Proof. In a similar way as in the proof of lemma 7.8 it is deduced that adj im ($2 \leq |i| \leq m$) at a node at distance d from v implies the subcombination adj $m-1\underline{m}$ at a node at distance $d-1$ from v. Then, by lemma 7.25 adj $\underline{m-1m}$ occurs at a node at distance $d-2$ from v.
Substitution of $i=\underline{m-1}$ in lemma 7.25 and using Descente Infinie, we easily deduce that adj $\underline{m-1m}$ is infeasible in S_{gk} ($g \geq 7$, $k \geq 3$). Hence, adj im ($2 \leq |i| \leq m$) is infeasible. □

(7.27) Lemma. Adj $\underline{1m}$ is infeasible in S_{gk} ($g \geq 7$, $k \geq 3$).

Proof. Adj $\underline{1m}$ induces the situation in figure 7.21.

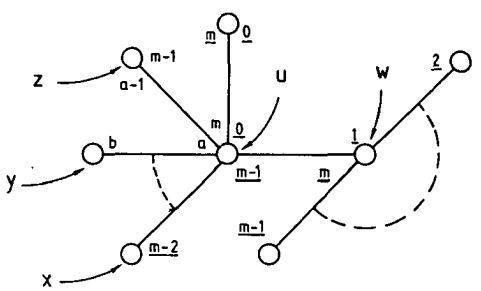


Figure 7.21 Situation occurring in lemma 7.27.

If $k=3$, then x, y and z coincide, resulting in adj $\underline{m-1}m$, which is infeasible by lemma 7.26. Hence, adj $\underline{1m}$ is infeasible in S_{g3} ($g \geq 7$).

If $k=4$ then x and y coincide, from which we directly conclude $b=a-1$. This implies $a=\underline{m}$, inducing the subcombination adj $\underline{m}m$ at node u, which is infeasible by lemma 7.26.
If $k>4$, then construct shortest paths G_1 and G_2 from v to x and z respectively. We define G_1', G_2' and v' in the usual way. The circuit $v'G_1'xuzG_2'v'$ does not properly surround node w by lemma 6.24. Hence, it properly surrounds node y, implying $b=a-1$. Hence $a=\underline{m}$, inducing the subcombination adj $\underline{m}m$ at node u, which is infeasible by lemma 7.26. □

(7.28) Lemma. If w is a node in S_{gk} ($g\geq 7$, $k\geq 3$) for which RD-c(w)$=0^{k-2}1\underline{m-1}$ and $d(w,v)\geq 2$, then w has $k-1$ neighbours at distance $d(w,v)+1$ from v, of which

- 1 has RD-combination $0^{k-2}12$,
- 1 has RD-combination $0^{k-3}1\underline{m}1$,
- $k-3$ have RD-combination $0^{k-2}11$.

The remaining neighbour of w lies at distance $d(w,v)-1$ from v.

Proof. RD-c(w)$=0^{k-2}1\underline{m-1}$ induces the situation in figure 7.22 (if $k=3$, nodes x and y coincide).

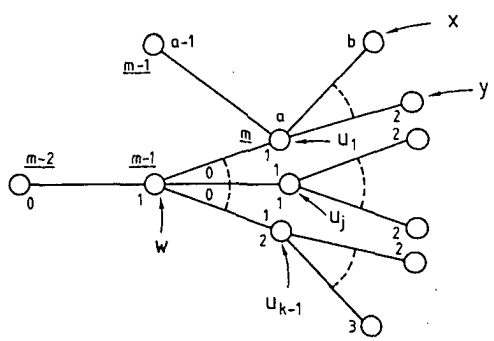


Figure 7.22 Situation occurring in lemma 7.28.

It follows directly that $d(u_j,v)=d(w,v)+1$ for $j=1,\ldots,k-1$. Clearly the remaining neighbour of w lies at distance $d(w,v)-1$ from v.
By lemma 7.26, $|a|<2$. From $|a-1|\geq 0$, we deduce that $a=1$ or $a=\underline{1}$. The latter

is impossible because of lemma 7.27. Hence, a=1 and b=2. Then, lemma 7.2 implies RD-c(u_1)=$0^{k-3}1\underline{m}1$, RD-c(u_{k-1})=$0^{k-2}12$ and RD-c(u_j) = $0^{k-2}11$ for j=2,...,k−2. □

C1. subcase g≥7, k≥4

(7.29) Lemma. If w is a node in S_{gk} (g≥7, k≥4) for which RD-c(w)=$\underline{0}0^{k-3}1m$ and d(w,v)≥3, then w has k−2 neighbours at distance d(w,v)+1 from v, of which

- 1 has RD-combination $0^{k-2}1\underline{1}$,
- 1 has RD-combination $0^{k-2}12$,
- k−4 have RD-combination $0^{k-2}11$.

One of the remaining two neighbours of w lies at distance d(w,v)−1 from v and one lies at distance d(w,v) from v.

Proof. RD-c(w)=$\underline{0}0^{k-3}1m$ induces the situation in figure 7.23.

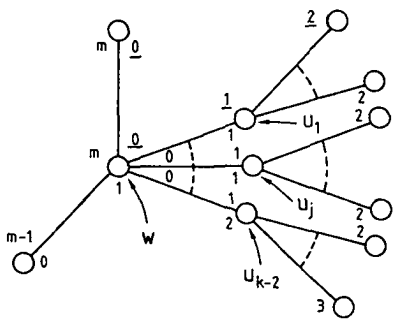


Figure 7.23 Situation occurring in lemma 7.29.

It follows directly that d(u_j,v)=d(w,v)+1 for j=1,...,k−2. Clearly, one of the remaining two neighbours of w lies at distance d(w,v)−1 from v and the other lies at distance d(w,v) from v.
Lemma 7.2 immediately implies RD-c(u_1)=$0^{k-2}1\underline{1}$, RD-c(u_{k-2})=$0^{k-2}12$ and RD-c(u_j)=$0^{k-2}11$ for j=2,...,k−3. □

(7.30) Lemma. If w is a node in S_{gk} (g≥7, k≥4) for which RD-c(w)=$0^{k-3}1\underline{m}1$ and d(w,v)≥3, then w has k−2 neighbours at distance d(w,v)+1 from v, of which

- 2 have RD-combination $0^{k-2}12$,

- $k-4$ have RD-combination $0^{k-2}11$.

The remaining 2 neighbours of w lie at distance $d(w,v)-1$ from v.

Proof. RD-c(w)$=0^{k-3}1\underline{m}1$ induces the situation in figure 7.24.

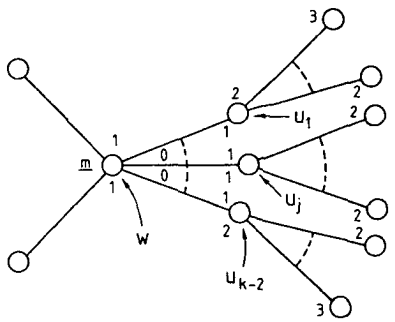


Figure 7.24 Situation occurring in lemma 7.30.

It follows directly that $d(u_j,v)=d(w,v)+1$ for $j=1,\ldots,k-2$. Clearly, the remaining 2 neighbours of w lie at distance $d(w,v)-1$ from v.
Lemma 7.2 immediately implies RD-c(u_1)=RD-c(u_{k-2})$=0^{k-2}12$ and RD-c(u_j)$=0^{k-2}11$ for $j=2,\ldots,k-3$. □

Lemmas 7.23, 7.24, 7.28, 7.29 and 7.30 imply that RD-combinations $0^{k-2}11$, $0^{k-2}12,\ldots$, $0^{k-2}1m-1$, $0^{k-2}1\underline{1}$, $0^{k-2}1\underline{2},\ldots$, $0^{k-2}1\underline{m-1}$, $\underline{0}0^{k-3}1m$ and $0^{k-3}1\underline{m}1$ at nodes in $\Gamma_{d-1}(v)$ ($d\geq 2$) give rise to the same set of RD-combinations at nodes in $\Gamma_d(v)$. By lemma 7.3 and the foremention lemmas these RD-combinations indeed occur in supersymmetric graphs with girth $g\geq 7$ and degree $k\geq 4$.
The above lemmas result in the following remarks.

1. Each node in $\Gamma_d(v)$ ($d\geq 1$) has 1 neighbour in $\Gamma_{d-1}(v)$ except when it has RD-combination $0^{k-3}1\underline{m}1$.
2. For each node $w\in\Gamma_{d-1}(v)$ ($d\geq 2$) its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-2}11$ is equal to
   - $k-3$ if RD-c(w)$=01i$ for $1\leq|i|\leq m-1$,
   - $k-4$ if RD-c(w)$=\underline{0}0^{k-3}1m$ or RD-c(w)$=0^{k-3}1\underline{m}1$.

   Together with remark 1 this implies

$$N_{0^{k-2}11}(d) = (k-3).N_{0^{k-2}11}(d-1) + (k-3).N_{0^{k-2}12}(d-1) + \ldots + (k-3).N_{0^{k-2}1m-1}(d-1) + (k-4).N_{\underline{0}0^{k-3}1m}(d-1) +$$

$$(k-3).N_{0^{k-2}11}(d-1) + (k-3).N_{0^{k-2}12}(d-1) + \ldots +$$
$$(k-3).N_{0^{k-2}1m-1}(d-1) + (k-4).N_{0^{k-3}1m1}(d-1).$$

3. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 2)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-2}12$ is equal to
   - 2 if RD-c(w) $= 0^{k-2}11$ or RD-c(w) $= 0^{k-3}1m1$,
   - 1 if w has another RD-combination.

   Together with remark 1 this implies

$$N_{0^{k-2}12}(d) = 2.N_{0^{k-2}11}(d-1) + N_{0^{k-2}12}(d-1) + \ldots + N_{0^{k-2}1m-1}(d-1) +$$
$$N_{\underline{0}0^{k-3}1m}(d-1) + N_{0^{k-2}\underline{1}1}(d-1) + N_{0^{k-2}\underline{1}2}(d-1) + \ldots +$$
$$N_{0^{k-2}1m-\underline{1}}(d-1) + 2.N_{0^{k-2}\underline{1}m1}(d-1).$$

4. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 3)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-2}1i$, $3 \leq i \leq m-1$, is equal to 1 if RD-c(w) $= 0^{k-2}1i-1$ and 0 otherwise. Together with remark 1 this implies

$$N_{0^{k-2}1i}(d) = N_{0^{k-2}1i-1}(d-1).$$

5. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 2)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination $\underline{0}0^{k-3}1m$ is equal to 1 if RD-c(w) $= 0^{k-2}1m-1$ and 0 otherwise. Together with remark 1 this implies

$$N_{\underline{0}0^{k-3}1m}(d) = N_{0^{k-2}1m-1}(d-1).$$

6. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 2)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-2}1\underline{1}$ is equal to 1 if RD-c(w) $= \underline{0}0^{k-3}1m$ and 0 otherwise. Together with remark 1 this implies

$$N_{0^{k-2}1\underline{1}} = N_{\underline{0}0^{k-3}1m}(d-1).$$

7. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 2)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-2}1\underline{i}$, $2 \leq i \leq m-1$, is equal to 1 if RD-c(w) $= 0^{k-2}1\underline{i-1}$ and 0 otherwise. Together with remark 1 this implies

$$N_{0^{k-2}1\underline{i}} = N_{0^{k-2}1\underline{i-1}}(d-1).$$

8. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 3)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination $0^{k-3}1\underline{m}1$ is equal to 1 if RD-c(w) $= 0^{k-2}1\underline{m-1}$ and 0 otherwise. Furthermore, each node in $\Gamma_d(v)$ with RD-combination $0^{k-3}1\underline{m}1$ has 2 neighbours in $\Gamma_{d-1}(v)$ with RD-combination $0^{k-2}1\underline{m-1}$. Hence,

$$N_{0^{k-3}1\underline{m}1}(d) = 1/2 \; . \; N_{0^{k-2}1\underline{m-1}}(d-1).$$

When the vector N(d) is equal to

$$(N_{0^{k-2}11}(d),\ldots,N_{0^{k-2}1m-1}(d),N_{\underline{0}0^{k-3}1m}(d),N_{0^{k-2}1\underline{1}}(d),\ldots,N_{0^{k-2}1\underline{m-1}}(d),N_{0^{k-3}1m1}(d))^T,$$

and when we write the system of recurrence equations in the usual way, we obtain the $2m \times 2m$ matrix A_{gk} ($g \geq 7$, $k \geq 4$) equal to

$$A_{gk} = \begin{bmatrix}
k-3 & k-3 & k-3 & \ldots & k-3 & k-4 & k-3 & \ldots & k-3 & k-3 & k-4 \\
2 & 1 & 1 & \ldots & 1 & 1 & 1 & \ldots & 1 & 1 & 2 \\
0 & 1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 1 & \ldots & 0 & 0 & 0 & \ldots & 0 & 0 & 0 \\
. & . & . & \ldots & . & . & . & \ldots & . & . & . \\
. & . & . & \ldots & . & . & . & \ldots & . & . & . \\
0 & 0 & 0 & \ldots & 1 & 0 & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 0 & \ldots & 0 & \mathbf{1} & 0 & \ldots & 0 & 0 & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 1 & \ldots & 0 & 0 & 0 \\
. & . & . & \ldots & . & . & . & \ldots & . & . & . \\
. & . & . & \ldots & . & . & . & \ldots & . & . & . \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 1 & 0 & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 & 1/2 & 0
\end{bmatrix}$$

The matrix element in bold font lies in row m+1 and column m. For the exponentiality of the graphs dealt with above we refer to paragraph 7.5.

C2. subcase $g \geq 7$, $k=3$

(7.31) Lemma. If w is a node in S_{gk} ($g \geq 7$, $k=3$) for which RD-c(w)=$\underline{0}$1m and $d(w,v) \geq 3$, then w has 1 neighbour at distance d(w,v)+1 from v, 1 neighbour at distance d(w,v) from v, and 1 neighbour at distance d(w,v)−1 from v. The neighbour at distance d(w,v)+1 from v has RD-combination 0$\underline{1}$2.

Proof. RD-c(w)=$\underline{0}$1m induces the situation in figure 7.25.

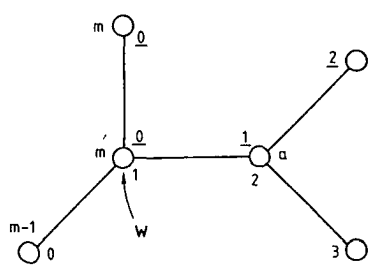


Figure 7.25 Situation occurring in lemma 7.31.

The distances from v of the neighbours of w follow directly from figure 7.25.

Application of lemma 7.2 gives a=0. □

(7.32) Lemma. If w is a node in S_{gk} (g≥7, k=3) for which RD-c(w)=11m and d(w,v)≥3, then w has 1 neighbour at distance d(w,v)+1 from v and 2 neighbours at distance d(w,v)−1 from v. The neighbour of w at distance d(w,v)+1 from v has RD-combination 022.

Proof. RD-c(w)=11m induces the situation in figure 7.26.

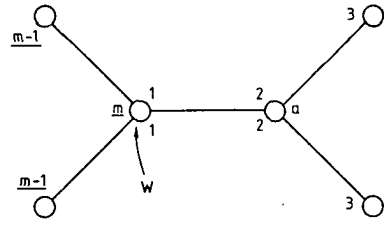


Figure 7.26 Situation occurring in lemma 7.32.

The distances from v of the neighbours of w follow directly from figure 7.26. Application of lemma 7.2 gives a=0. □

C2a. subsubcase g≥9, k=3

(7.33) Lemma. If w is a node in S_{g3} (g≥9) for which RD-c(w)=012 and d(w,v)≥2, then w has 2 neighbours at distance d(w,v)+1 from v, of which

- 1 has RD-combination 012,
- 1 has RD-combination 013.

The remaining neighbour of w lies at distance d(w,v)−1 from v.

Proof. RD-c(w)=012 induces the situation in figure 7.27.

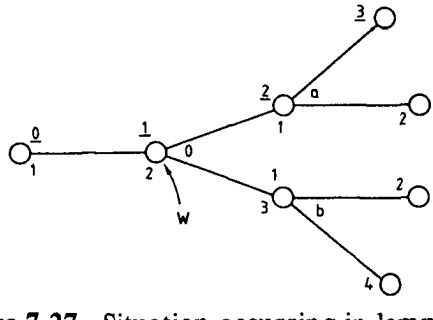


Figure 7.27 Situation occurring in lemma 7.33.

The distances from v of the neighbours of w follow directly from figure 7.27.

Application of lemma 7.2 gives $a=b=0$. □

(7.34) Lemma. If w is a node in S_{g3} ($g\geq 9$) for which RD-c(w)=022 and $d(w,v)\geq 2$, then w has 2 neighbours at distance $d(w,v)+1$ from v and 1 neighbour at distance $d(w,v)-1$ from v. The neighbours of w at distance $d(w,v)+1$ from v have RD-combination 013.

Proof. RD-c(w)=022 induces the situation in figure 7.28.

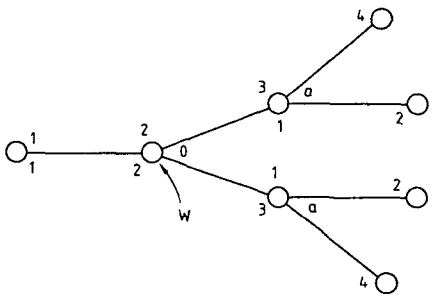


Figure 7.28 Situation occurring in lemma 7.34.

The distances from v of the neighbours of w follow directly from figure 7.28. Application of lemma 7.2 gives $a=0$. □

As noted before, in supersymmetric graphs with degree 3 the only nodes with RD-combination 011 are the three neighbours of v. Lemmas 7.23, 7.24, 7.28, 7.31, 7.32, 7.33 and 7.34 imply that RD-combinations 012, 013,..., 01m−1, 01m, 012, 012, 013,..., 01m−1, 11m and 022 at nodes in $\Gamma_{d-1}(v)$ ($d\geq 3$) give rise to the same set of RD-combinations at nodes in $\Gamma_d(v)$. Inasmuch as all nodes in $\Gamma_2(v)$ have RD-combination 012 (by lemmas 7.3 and 7.23), the above RD-combinations indeed occur in supersymmetric graphs with girth $g\geq 9$ and degree 3.
The above lemmas result in the following remarks.

1. Each node in $\Gamma_d(v)$ ($d\geq 1$) has 1 neighbour in $\Gamma_{d-1}(v)$ except when it has RD-combination 11m.
2. For each node $w\in\Gamma_{d-1}(v)$ ($d\geq 3$) its number of neighbours in $\Gamma_d(v)$ with RD-combination 012 is equal to 1 if RD-c(w)=01i for $2\leq|i|\leq m-1$ and 0 otherwise. Together with remark 1 this implies
$$N_{012}(d) = N_{012}(d-1) + \ldots + N_{01m-1}(d-1) + N_{012}(d-1) + \ldots + N_{01m-1}(d-1).$$
3. For each node $w\in\Gamma_{d-1}(v)$ ($d\geq 3$) its number of neighbours in $\Gamma_d(v)$ with RD-combination 013 is equal to

- 1 if RD-c(w)=012 or RD-c(w)=0$\underline{1}$2,
- 2 if RD-c(w)=022,

and 0 otherwise. Together with remark 1 this implies

$$N_{013}(d) = N_{012}(d-1) + N_{0\underline{1}2}(d-1) + 2.N_{022}(d-1).$$

4. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 4)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 01i $(4 \leq i \leq m-1)$ is equal to 1 if RD-c(w)=01i−1 and 0 otherwise. Together with remark 1 this implies

$$N_{01i}(d) = N_{01i-1}(d-1).$$

5. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 3)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination $\underline{0}$1m is equal to 1 if RD-c(w)=01m−1 and 0 otherwise. Together with remark 1 this implies

$$N_{\underline{0}1m}(d) = N_{01m-1}(d-1).$$

6. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 4)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 0$\underline{1}$2 is equal to 1 if RD-c(w)=$\underline{0}$1m and 0 otherwise. Together with remark 1 this implies

$$N_{0\underline{1}2}(d) = N_{\underline{0}1m}(d-1).$$

7. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 3)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 01$\underline{2}$ is equal to 1 if RD-c(w)=0$\underline{1}$2 and 0 otherwise. Together with remark 1 this implies

$$N_{01\underline{2}}(d) = N_{0\underline{1}2}(d-1).$$

8. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 3)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 01$\underline{i}$, $3 \leq i \leq m-1$, is equal to 1 if RD-c(w)=01$\underline{i-1}$ and 0 otherwise. Together with remark 1 this implies

$$N_{01\underline{i}}(d) = N_{01\underline{i-1}}(d-1).$$

9. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 3)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 11$\underline{m}$ is equal to 1 if RD-c(w)=01$\underline{m-1}$ and 0 otherwise. Furthermore, each node in $\Gamma_d(v)$ with RD-combination 11$\underline{m}$ has 2 neighbours in $\Gamma_{d-1}(v)$ with RD-combination 01$\underline{m-1}$. Hence,

$$N_{11\underline{m}}(d) = 1/2 \,.\, N_{01\underline{m-1}}(d-1).$$

10. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 4)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 022 is equal to 1 if RD-c(w)=11$\underline{m}$ and 0 otherwise. Together with remark 1 this implies

$$N_{022}(d) = N_{11\underline{m}}(d-1).$$

When the vector N(d) is equal to

$$(N_{012}(d),\ldots,N_{01m-1}(d),N_{\underline{0}1m}(d),N_{0\underline{1}2}(d),N_{01\underline{2}}(d),\ldots,N_{01\underline{m-1}}(d),N_{11\underline{m}}(d),N_{022}(d))^T,$$

and when we write the system of recurrence equations in the usual way, we obtain the $2m \times 2m$ matrix A_{g3} ($g \geq 9$) equal to

$$A_{g3} = \begin{bmatrix}
1 & 1 & 1 & \ldots & 1 & 0 & 0 & 1 & 1 & \ldots & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & \ldots & 0 & 0 & 1 & 0 & 0 & \ldots & 0 & 0 & 0 & 2 \\
0 & 1 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 \\
0 & 0 & 1 & \ldots & 0 & 0 & \mathbf{0} & 0 & 0 & \ldots & 0 & 0 & 0 & 0 \\
. & . & . & \ldots & . & . & . & . & . & \ldots & . & . & . & . \\
. & . & . & \ldots & . & . & . & . & . & \ldots & . & . & . & . \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & \ldots & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 1/2 & 0 & 0 \\
0 & 0 & 0 & \ldots & 0 & 0 & 0 & 0 & 0 & \ldots & 0 & 0 & 1 & 0
\end{bmatrix}$$

The matrix element in bold font lies in row 4 and column m. For the exponentiality of S_{g3} ($g \geq 9$) we refer to paragraph 7.5.

C2b. subsubcase $g=7$, $k=3$

(7.35) Lemma. If w is a node in S_{73} for which RD-c(w)$=0\underline{1}2$ and $d(w,v) \geq 2$, then w has 2 neighbours at distance $d(w,v)+1$ from v, of which

- 1 has RD-combination $01\underline{2}$,
- 1 has RD-combination $\underline{0}13$.

The remaining neighbour of w lies at distance $d(w,v)-1$ from v.

Proof. RD-c(w)$=0\underline{1}2$ induces the situation in figure 7.29.

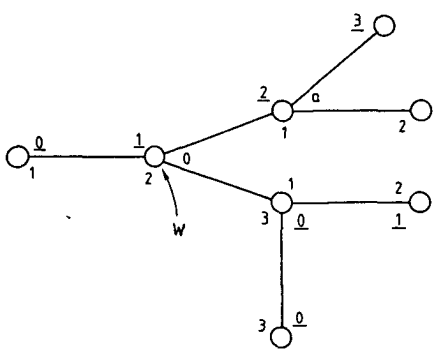


Figure 7.29 Situation occurring in lemma 7.35.

The distances from v of the neighbours of w follow directly from figure 7.29. Application of lemma 7.2 gives a=0. □

(7.36) Lemma. If w is a node in S_{73} for which RD-c(w)=022 and $d(w,v)\geq 2$, then w has 2 neighbours at distance d(w,v)+1 from v and 1 neighbour at distance d(w,v)−1 from v. The neighbours of w at distance d(w,v)+1 from v have RD-combinations $\underline{0}13$.

Proof. RD-c(w)=022 induces the situation in figure 7.30.

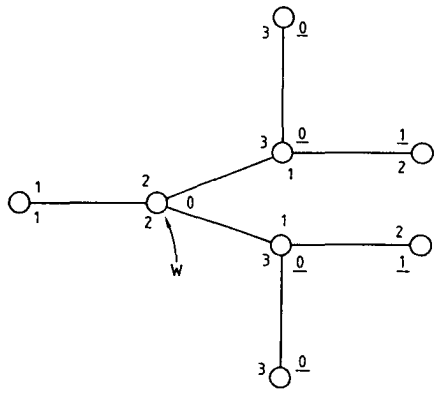


Figure 7.30 Situation occurring in lemma 7.36.

The distances from v of the neighbours of w follow directly from figure 7.30. □

The only nodes with RD-combination 011 are the three neighbours of v. Lemmas 7.23, 7.24, 7.28, 7.31, 7.32, 7.35 and 7.36 imply that RD-combinations 012, $\underline{0}13$, $01\underline{2}$, 012, $11\underline{3}$ and 022 at nodes in $\Gamma_{d-1}(v)$ ($d\geq 3$) give rise to the same set of RD-combinations at nodes in $\Gamma_d(v)$. Inasmuch as all nodes in $\Gamma_2(v)$ have RD-combination 012 (by lemmas 7.3 and 7.23), the above RD-combinations indeed occur in S_{73}.
The above lemmas result in the following remarks.

1. Each node in $\Gamma_d(v)$ ($d\geq 1$) has 1 neighbour in $\Gamma_{d-1}(v)$ except when it has RD-combination $11\underline{3}$.
2. For each node $w\in\Gamma_{d-1}(v)$ ($d\geq 3$) its number of neighbours in $\Gamma_d(v)$ with RD-combination 012 is equal to 1 if RD-c(w)=012 or RD-c(w)=$01\underline{2}$ and 0 otherwise. Together with remark 1 this implies

$$N_{012}(d) = N_{012}(d-1) + N_{01\underline{2}}(d-1).$$

3. For each node $w\in\Gamma_{d-1}(v)$ ($d\geq 3$) its number of neighbours in $\Gamma_d(v)$ with RD-combination $\underline{0}13$ is equal to

- 1 if RD-c(w)=012 or RD-c(w)=0$\underline{1}$2,
- 2 if RD-c(w)=022,

and 0 otherwise. Together with remark 1 this implies

$$N_{\underline{0}13}(d) = N_{012}(d-1) + N_{0\underline{1}2}(d-1) + 2.N_{022}(d-1).$$

4. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 4)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 0$\underline{1}$2 is equal to 1 if RD-c(w)=$\underline{0}$13 and 0 otherwise. Together with remark 1 this implies

$$N_{0\underline{1}2}(d) = N_{\underline{0}13}(d-1).$$

5. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 3)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 01$\underline{2}$ is equal to 1 if RD-c(w)=0$\underline{1}$2 and 0 otherwise. Together with remark 1 this implies

$$N_{01\underline{2}}(d) = N_{0\underline{1}2}(d-1).$$

6. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 3)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 11$\underline{3}$ is equal to 1 if RD-c(w)=01$\underline{2}$ and 0 otherwise. Furthermore, each node in $\Gamma_d(v)$ with RD-combination 11$\underline{3}$ has 2 neighbours in $\Gamma_{d-1}(v)$ with RD-combinations 01$\underline{2}$. Hence,

$$N_{11\underline{3}}(d) = 1/2 \,.\, N_{01\underline{2}}(d-1).$$

7. For each node $w \in \Gamma_{d-1}(v)$ $(d \geq 4)$ its number of neighbours in $\Gamma_d(v)$ with RD-combination 022 is equal to 1 if RD-c(w)=11$\underline{3}$ and 0 otherwise. Together with remark 1 this implies

$$N_{022}(d) = N_{11\underline{3}}(d-1).$$

When we define the vector N by

$$N(d) := (N_{012}(d), N_{\underline{0}13}(d), N_{0\underline{1}2}(d), N_{01\underline{2}}(d), N_{11\underline{3}}(d), N_{022}(d))^T,$$

and the system of recurrence equations in the usual way, we obtain the 6×6 matrix A_{73}, equal to

$$A_{73} = \begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 2 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$$

For the exponentiality of S_{73} we refer to the next paragraph.

7.5 The uniform exponentialities of supersymmetric graphs

In this paragraph we deduce the uniform exponentialities of supersymmetric graphs from their recurrence equations. This will be done in the following way.
First, we deal with the solution of a general system of linear recurrence equations $N(d)=A.N(d-1)$, describing the number of nodes in a graph Γ as function of the distance to a certain node. It will be pointed out how the solution can be related to the uniform exponentiality of the corresponding graph.
Second, a general result is deduced about the 'behaviour' of the solutions of such a system of recurrence equations. It will appear that a real eigenvalue of the matrix A in the interval $[1,\infty)$ determines the solution. This eigenvalue equals the uniform exponentiality of Γ.
Third, the characteristic polynomial of A_{gk} will be determined. It appears to have one root in the interval $[1,\infty)$. Consequently, the uniform exponentiality of S_{gk} is equal to the largest root of the characteristic polynomial of A_{gk}.
This chapter is concluded by a theorem which states that the exponentiality of S_{gk} goes asymptotically to $k-1$ as g increases.

We start by considering the solution of the system of linear recurrence equations $N(d)=A.N(d-1)$. It is well-known that the solution can be written as

$$N(d) = \sum_{j=1}^{s} \sum_{n=0}^{m(r_j)-1} c_{jn}\, v_{jn}\, d^n\, r_j^d,$$

where r_j is one of the s eigenvalues of A, $m(r_j)$ is the multiplicity of r_j, v_{jn} is an eigenvector belonging to r_j, and c_{jn} is a constant of which the value depends on the boundary conditions of the system. Defining $N_\Sigma(d)$ as the sum of all vector elements of N(d), it can be expressed as

$$\textbf{(7.37)} \qquad N_\Sigma(d) = \sum_{j=1}^{s} \sum_{n=0}^{m(r_j)-1} c'_{jn}\, d^n\, r_j^d,$$

c'_{jn} being a constant.
We notice that $|\Gamma_d(v)| = N_\Sigma(d)$ for $d \geq 2$. Hence, if for a certain j

1. $|r_j|$ is maximal, and
2. $c'_{jn} \neq 0$ for some value of n,

then r_j determines the behaviour of $|\Gamma_d(v)|$. That is, if r and c are the r_j and c'_{jn} respectively for which conditions 1 and 2 hold, then (see appendix A for the notation)

$$|\Gamma_d(v)| \sim c.d^n.r^d \sim d^n.r^d.$$

We now have the following lemma.

(7.38) Lemma. If $|\Gamma_d(v)| \sim d^n.r^d$, for a certain node v in Γ, for $r \in [1,\infty)$, $n \in \mathbb{N}$, and $d \in \mathbb{Z}^+$, then $\overline{exp}(\Gamma) = r$.

Proof. The condition $|\Gamma_d(v)| \sim d^n.r^d$ implies

$$\forall v \in V(\Gamma)\ \forall \epsilon > 0\ \exists c > 0\ \forall d \in \mathbb{Z}^+: |\Gamma_d(v)| < c.(r+\epsilon)^d.$$

This implies

$$\forall v \in V(\Gamma)\ \forall \epsilon > 0\ \exists c > 0\ \forall d \in \mathbb{Z}^+: \sum_{i=0}^{d} |\Gamma_i(v)| < c.\frac{(r+\epsilon)^{d+1}-1}{r+\epsilon-1},$$

from which we conclude that $\overline{exp}(\Gamma) \leq r$. Furthermore,

$$\sum_{i=0}^{d} |\Gamma_i(v)| \geq |\Gamma_d(v)|,$$

which implies $\overline{exp}(\Gamma) \geq r$. □

Hence, to obtain the exponentiality of Γ, we need to study $|\Gamma_d(v)|$.
We might assert that $|\Gamma_d(v)|$ is determined by the eigenvalue of A with maximal modulus. This is not necessarily true, however. The constant coefficients of the terms corresponding to this eigenvalue in the description of $N_\Sigma(d)$ might vanish, because of the boundary conditions of the system of equations. So, the exponentiality of Γ is determined by the eigenvalue with the largest modulus of A occurring in a term in $N_\Sigma(d)$ of which the constant coefficient is non-zero. In lemma 7.40 we prove that this is a real eigenvalue in the interval $[1,\infty)$. To prove this, we first need a lemma.

(7.39) Lemma. If for the n real numbers $\phi_1,\ldots,\phi_n$ and the n real numbers $c_1,\ldots,c_n$

- $\phi_j \neq 2k\pi$ for $j=1,\ldots,n$ $(k \in \mathbb{Z})$,
- $\exists d \in \mathbb{N}: \sum_{j=1}^{n} c_j \cos d\phi_j > 0$,

then there exist infinitely many $d \in \mathbb{N}$ such that

$$\sum_{j=1}^{n} c_j \cos d\phi_j < 0.$$

Proof. We notice that the sequence $(\sum_{d=0}^{m} c_j \cos d\phi_j)_m$ is bounded for all $m \in \mathbb{N}$. For,

$$\sum_{d=0}^{m} c_j \cos d\phi_j + i \sum_{d=0}^{m} c_j \sin d\phi_j = \sum_{d=0}^{m} c_j e^{id\phi_j} = c_j.\frac{1-e^{(m+1)i\phi_j}}{1-e^{i\phi_j}}.$$

The norm of this quotient is bounded for all $m \in \mathbb{N}$, because $\phi_j \neq 2k\pi$, and so, the real part of the quotient is bounded for all $m \in \mathbb{N}$.
Hence,

$$\sum_{d=0}^{\infty} c_j \cos d\phi_j$$

is bounded. Clearly,

$$\sum_{j=1}^{n} \sum_{d=0}^{\infty} c_j \cos d\phi_j = \sum_{d=0}^{\infty} \sum_{j=1}^{n} c_j \cos d\phi_j$$

is also bounded. It is given that

$$\sum_{j=1}^{n} c_j \cos d\phi_j > \epsilon > 0$$

for at least one $d \in \mathbb{N}$ and some $\epsilon \in \mathbb{R}^+$. Let d_0 be this d, and ϵ_0 be a corresponding ϵ. For reasons to become clear, we need to find infinitely many d for which the left-hand side of the above expression is larger than the (fixed) ϵ_0. To obtain such d, we notice that the topological compact space $X = \mathbb{R}/\mathbb{Z} \times \ldots \times \mathbb{R}/\mathbb{Z}$ together with the continuous map $T: X \to X$, defined by

$$T(x_1,\ldots,x_n) := (\phi_1 + x_1, \ldots, \phi_n + x_n), \ (x_1,\ldots,x_n) \in X,$$

constitute a dynamical system (X,T). In addition, this system is what is called a Kronecker system by Furstenberg in [Furste]. Then, by theorem 1.2 in [Furste] every point in X is recurrent ([Furste; definition 1.1]). In particular, for the point $(d_0\phi_1,\ldots,d_0\phi_n)$ there exists a $q \in \mathbb{Z}^+$ and an $\epsilon_1 \in \mathbb{R}^+$ such that

$$\sum_{j=1}^{n} c_j \cos (d_0+q)\phi_j > \epsilon_1 > \epsilon_0 > 0.$$

Let d_1 be defined by $d_1 = d_0 + q$. In this way, an infinite increasing sequence $(d_0, d_1, d_2, \ldots)$ can be constructed such that

$$\sum_{j=1}^{n} c_j \cos d_i\phi_j > \epsilon_0 \quad \text{for } i = 0,1,2,\ldots.$$

Summing this expression over i gives ∞ in the right-hand side. Since the sum

$$\sum_{d=0}^{\infty} \sum_{j=1}^{n} c_j \cos d\phi_j$$

is bounded, there must be infinitely many d for which

$$\sum_{j=1}^{n} c_j \cos d\phi_j < 0,$$

which was to be proved. □

(7.40) Lemma. If Γ is an infinite locally finite connected graph for which $|\Gamma_d(v)|$ is an increasing function of d for some node $v \in V(\Gamma)$, and $N(d) = A \,.\, N(d-1)$ is a system of linear recurrence equations such that the sum $N_\Sigma(d)$ of all the elements of the vector N(d) is equal to $|\Gamma_d(v)|$, then the matrix A has a positive real eigenvalue $r \geq 1$ such that $N_\Sigma(d) \sim d^n.r^d$, for some $n \in \{0,1,\ldots,m(r)-1\}$, where m(r) is the multiplicity of r.

Proof. We notice the following:

1. The matrix elements of A are real, because of the nature of the system of equations. Consequently, the coefficients of the terms of the characteristic polynomial f(t) of A are real. This implies that if $a+bi$ $(b \neq 0)$ is a root of f(t), then $a-bi$ is also a root of f(t).

2. If the term $c.d^n(a+bi)^d$ $(b \neq 0)$ occurs in $N_\Sigma(d)$, then the term $c.d^n(a-bi)^d$ occurs in $N_\Sigma(d)$ too, since $N_\Sigma(d)$ is a real positive number for all d. Summation of these two terms gives

   $$c.d^n(a+bi)^d + c.d^n(a-bi)^d = 2.c.d^n(a^2+b^2)^{d/2}.\cos d\phi,$$

   where $\phi = \arccos a / (a^2+b^2)^{1/2}$. Hence, an eigenvalue $a+bi$ $(b \neq 0)$ with multiplicity $m(a+bi)$ gives rise to terms like

   $$2.c_j.d^j(a^2+b^2)^{d/2}\cos d\phi$$

   in $N_\Sigma(d)$ $(0 \leq j \leq m(a+bi)-1)$.

3. If r is a real negative eigenvalue of A with multiplicity m(r), then it gives rise to terms like

   $$c_j\, d^j\, r^d = c_j\, d^j\, |r|^d\, \cos d\pi$$

   in $N_\Sigma(d)$ $(0 \leq j \leq m(r)-1)$.

Let $t_{r,n}(d)$ be the term $c'_{jn}d^n r_j^d$ occurring in expression 7.37, and let $T_{\rho n}(d)$ be defined by

$$T_{\rho n}(d) := \sum_{|r|=\rho} t_{rn}(d).$$

Then, by 2 and 3, $T_{\rho n}(d)$ can be written as

$$T_{\rho n}(d) = d^n \rho^d \sum_{j=1}^{h_{\rho n}} c_{j\rho n} \cos d\phi_{j\rho n},$$

$c_{1\rho n}, c_{2\rho n}, \ldots$ being $h_{\rho n}$ real numbers, and $\phi_{1\rho n}, \phi_{2\rho n}, \ldots$ being $h_{\rho n}$ real numbers $(h_{\rho n} \in \mathbb{Z}^+)$. Let $\hat{\rho}$ be the maximum ρ for which there exists an n such that $T_{\rho n}(d)$

is not identical to 0, and let $\hat{n}$ be the maximum n for which $T_{\hat{\rho}n}(d)$ is not identical to 0. Then,

$N_\Sigma(d) \sim T_{\hat{\rho}\hat{n}}(d) \sim d^{\hat{n}}\hat{\rho}^d$.

Since $N_\Sigma(d)>0$ for all $d\in\mathbb{N}$, there are infinitely many d for which

$$\sum_{j=1}^{h_{\hat{\rho}\hat{n}}} c_{j\hat{\rho}\hat{n}} \cos d\phi_{j\hat{\rho}\hat{n}} > 0.$$

If $\phi_{j\hat{\rho}\hat{n}} \neq 2k\pi$ $(k\in\mathbb{Z})$ for $j=1,\ldots,h_{\hat{\rho}\hat{n}}$, then by lemma 7.39 $T_{\hat{\rho}\hat{n}}(d) < 0$ for infinitely many d. This means that $N_\Sigma(d)<0$ for an infinite subset of these d, which is impossible. Hence, $\phi_{j\hat{\rho}\hat{n}}=2k\pi$ for some j, implying that $\hat{\rho}$ is a real positive eigenvalue of A. Furthermore, $\hat{\rho}\in(0,1)$ is impossible, since it would imply that $N_\Sigma(d)$ is a decreasing function of d. Hence, $\hat{\rho}\in[1,\infty)$. □

The previous lemma implies that some real eigenvalue of A_{gk} in the interval $[1,\infty)$ determines the exponentiality of S_{gk}. To obtain the eigenvalues of A_{gk}, we shall determine the characteristic polynomial of A_{gk}. This job can be done by standard techniques. Some of the matrices A_{gk} determined in the previous two paragraphs require a special treatment, because some values of k might give rise to matrices of which some matrix elements vanish. For example, the case g even, $g\geq 6$, $k\geq 4$ will ask for a special treatment of the subcase $k=4$ because A_{gk} contains some elements equal to $k-4$ in that case.
We shall not describe the computation of $f_{gk}(t)$ from A_{gk}, but merely give the result.

(7.41) Theorem. The characteristic polynomial of A_{gk} is equal to

- $f_{gk}(t) = (-1)^m(t^m-(k-2)\sum_{i=1}^{m-1} t^i+1)$ $(m=g/2)$, if g is even,
- $f_{gk}(t) = t^{2m}-(k-2)\sum_{i=m+1}^{2m-1} t^i-(k-4)t^m-(k-2)\sum_{i=1}^{m-1} t^i+1$ $(m=(g-1)/2)$, if g is odd.

Proof. By standard techniques. □

For convenience, we use two polynomials f_{1gk} and f_{2gk} instead of f_{gk}. They are defined by

$f_{1gk}(t) := (-1)^m f_{gk}(t)$ if g is even,

$f_{2gk}(t) := f_{gk}(t)$ if g is odd.

Trivially, this doesn't affect any root. Both polynomials will appear to have

exactly one root $r \geq 1$. To prove this, we first define the notion of sign changes.

(7.42) Definition. The number of *sign changes* in the row $\alpha_1, \alpha_2, \ldots, \alpha_n$ of real non-zero numbers is equal to

$$| \{ i \mid \alpha_i \cdot \alpha_{i+1} < 0, \ 1 \leq i \leq n-1 \} |$$

Insertion of zeros in this row will leave the number of sign changes intact.

(7.43) Lemma. (Descartes).
If the row of coefficients $a_n, a_{n-1}, \ldots, a_0$ of the n^{th} degree equation

$$a_n x^n + a_{n-1} x^{n-1} + \cdots + a_0 = 0 \quad (a_n \neq 0, \ a_0 \neq 0)$$

has p sign changes, then the number of positive roots of the above equation is $p-2s$ for some s $(0 \leq s \leq p/2)$, where an n-fold root is counted n times.

Proof. See for example [Perron]. □

From this theorem we directly conclude that $f_{igk}(t)$ $(i=1,2)$ has at most two positive roots for all g and k except possibly when $k=3$ and g is odd - the coefficients of the characteristic polynomial have 4 sign changes in that case. Furthermore,

$$f_{igk}(0) = 1 > 0 \quad (i=1,2),$$

$$f_{1gk}(1) = -(m-1)(k-2)+2 < 0 \quad \text{for } (g,k) \neq (4,4) \text{ and } (g,k) \neq (6,3),$$

$$f_{2gk}(1) = -2(m-1)(k-2)-(k-6) < 0 \quad \text{for } (g,k) \neq (3,6),$$

$$f_{igk}(t) > 0 \quad \text{for } t \to \infty \ (i=1,2).$$

Hence, for (g,k) not equal to (3,6), (4,4) or (6,3) or for $k \neq 3$ or g is even, both the intervals (0,1) and $(1,\infty)$ contain exactly one root. The cases (g,k) = (3,6), (4,4) or (6,3) yield a root at $t=1$ and no roots in $(1,\infty)$, again showing that the three corresponding grids are not exponential. Supersymmetric graphs with odd girth and degree 3 remain to be considered. For this we use a lemma which is a generalization of lemma 7.43.

(7.44) Lemma. (Fourier-Budan).
Let f(x) be an n^{th} degree polynomial with real coefficients and let α and β be two real numbers for which $\alpha < \beta$ and $f(\alpha) \neq 0$ and $f(\beta) \neq 0$. Let $R_f(x)$ denote the row

$$f(x), f'(x), \ldots, f^{(n)}(x).$$

If p respectively q denotes the number of sign changes in $R_f(\alpha)$ respectively $R_f(\beta)$, then $p \geq q$ and the number of roots of f(x) in the interval (α,β) is equal to $p-q-2s$ for some s $(0 \leq s \leq (p-q)/2)$, where an n-fold root is counted n times.

Proof. See for example [Perron]. □

This theorem enables us to determine the number of roots of $f_{2g3}(t)$ in the interval $(1,\infty)$. First, we notice that $f_{2g3}^{(n)}(t) > 0$ when $t\to\infty$ for $n=0,1,\ldots,2m$. Second, we have the following values of $f_{2g3}^{(n)}(1)$ for $n=0,1,\ldots,2m$.

$$f_{2g3}(1) = -2m+4,$$
$$f_{2g3}^{(n)}(1) = n!\{\binom{2m}{n} - \binom{2m}{n+1} + \binom{m+1}{n+1} + \binom{m}{n} - \binom{m}{n+1}\} \quad \text{for } 1\le n\le m-1,$$
$$f_{2g3}^{(m)}(1) = m!\{\binom{2m}{m} - \binom{2m}{m+1}\},$$
$$f_{2g3}^{(n)}(1) = n!\{\binom{2m}{n} - \binom{2m}{n+1}\} \quad \text{for } m+1\le n\le 2m-1,$$
$$f_{2g3}^{(2m)}(1) = (2m)!.$$

It is easily established that $f_{2g3}(1) < 0$ (notice that $m\ge 3$), and $f_{2g3}^{(n)}(1) > 0$ for $m\le n\le 2m$. For other n we first need a lemma.

(7.45) Lemma. For all m and n for which $1\le n\le m-2$, or $n=m-1$ and $m\ge 4$

$$\frac{\binom{2m}{n}}{\binom{m}{n}} > \frac{2n+2}{2m-2n-1}.$$

Proof. Let $n=m-r$, then $1\le r\le m-1$. This yields an inequality which can be proved by induction on m. The induction basis is $m=r+1$ for $r\ge 2$ and $m=4$ for $r=1$. □

The value of $f_{2g3}^{(n)}(1)$ for $1\le n\le m-1$ can be rewritten to

$$n!\binom{m}{n}\{\frac{\binom{2m}{n}}{\binom{m}{n}}\frac{-2m+2n+1}{n+1}+2\}.$$

The previous lemma implies that this value is negative for all m and n for which $1\le n\le m-2$, or $n=m-1$ and $m\ge 4$. The reader may verify that the above value is positive for the remaining m and n, i.e. for $m=3$ and $n=2$. In all cases there is one sign change in $R_{f_{2g3}}(1)$. Hence, $f_{2g3}(t)$ has exactly one singular root in the interval $[1,\infty)$.

So, we have the following corollary.

(7.46) Corollary. The uniform exponentiality of S_{gk} is equal to the largest real eigenvalue of A_{gk}.

Proof. By lemmas 7.43, 7.44, and 7.45, $f_{gk}(t)$ has one root in the interval $[1,\infty)$ for all feasible g and k. Then, by lemma 7.40 and lemma 7.38 this root is equal to the uniform exponentiality of S_{gk}. □

Finally, we investigate the behaviour of the largest root of $f_{gk}(t)$ for fixed k when g increases.

(7.47) Theorem. For the largest root $\hat{t}$ of $f_{gk}(t)$ there exist a constant c only depending on k, and a constant $\delta>0$, such that

$$k-1-\hat{t} < c.(1+\delta)^{-m},$$

where $m=g/2$ for even g, and $m=(g-1)/2$ for odd g.

Proof. We distinguish between even and odd girth.

Even g. Suppose $(g,k)\neq(4,4)$ and $(g,k)\neq(6,3)$. Multiplying $f_{1gk}(t)$ by $t-1$ and rewriting gives

$$k-1-t = \frac{(k-1)t-1}{t^m} < (k-1)t^{1-m}.$$

Now, let δ be such that $f_{1gk}(1+\delta)<0$ for all $(g,k)\neq(4,4)$ and $(g,k)\neq(6,3)$ ($\delta=1/2$ suffices). Then, $1+\delta<\hat{t}$ because $f_{1gk}(1)<0$ for $(g,k)\neq(4,4),(6,3)$, and $f_{1gk}(t)>0$ for $t\to\infty$. Hence,

$$k-1-\hat{t} < (k-1)(1+\delta)(1+\delta)^{-m}.$$

Setting $c=(k-1)(1+\delta)$ proves the theorem for $k\geq 5$.
For $k=4$ set c equal to some number larger than $3(1+\delta)$ that makes the theorem valid for $g=4$. Clearly, this c also makes the theorem valid for $k=4$ and $g\geq 5$.
The hexagonal grid is dealt with in a similar way.

Odd g. In this case multiplying $f_{2gk}(t)$ by $t-1$ and rewriting gives

$$k-1-t = \frac{2t^{m+1}-2t^m+(k-1)t-1}{t^{2m}} < (k+1)t^{1-m}.$$

The rest of this case is analogous to the case even g. (We have to deal separately with $(g,k)=(3,6)$). □

We conclude that increase of g yields the uniform exponentiality of S_{gk} to get nearer to $k-1$. This result is just what we expected, because $S_{\infty k}$ is a tree with uniform exponentiality $k-1$. More importantly, the approach of the uniform exponentiality of S_{gk} to $k-1$ as g increases is pretty fast, as can also be seen in the table in appendix B.

It is easily established that $f_{1gk}(k-1)>0$ and $f_{2gk}(k-1)>0$, proving that the uniform exponentiality of S_{gk} is strictly less than $k-1$ for finite g.

7.6 Concluding remarks

This chapter dealt with the uniform exponentialities of supersymmetric graphs. Knowing the two parameters of a supersymmetric graph, its girth g and its degree k, we can easily establish its uniform exponentiality. The uniform exponentiality of S_{gk} is equal to the largest root of the polynomial $f_{gk}(t)$ defined by theorem 7.41. The uniform exponentiality of S_{gk} approaches $k-1$ for large g. From this point of view supersymmetric graphs are 'almost optimal'.

In the next chapter we investigate how to cut finite convex subgraphs with low R/r-ratios out of supersymmetric graphs.

8

Convex subgraphs of supersymmetric graphs

Measure a thousand times and cut once.

Turkish Proverb

8.1 Introduction

In this chapter we construct extension sequences consisting of convex subgraphs of supersymmetric graphs. For this, two approaches will be followed, resulting in two classes of extension sequences.
In the first approach, described in paragraph 8.2, we make use of the constructions of Grünbaum and Shephard in paragraph 6.3. It results in extensible networks of which the R/r-ratios are bounded from above by a fixed constant. The R/r-ratios of the networks are equal to 1, if the underlying graph is S_{3k} ($k \geq 6$). That is optimal. If other underlying graphs are used in the first approach, then the R/r-ratios of the constructed networks are not very close.
In the second approach, described in paragraph 8.3, we construct convex hulls of balls in supersymmetric graphs. Theorem 2.23 guarantees that such subgraphs have low R/r-ratios. The R/r-ratios of convex hulls of balls are 1 in S_{3k} ($k \geq 6$), 2 in S_{44}, and go asymptotically to 1 as r goes to ∞ in the other supersymmetric graphs.

8.2 Convex subgraphs of S_{gk}, a first approach

In this paragraph we design convex subgraphs of S_{gk} which are based on the constructions of Grünbaum and Shephard in paragraph 6.3. These constructions will appear to be optimal constructions for supersymmetric graphs with girth 3, i.e. the R/r-ratios of the constructs based on S_{3k} will appear to be 1. Other supersymmetric graphs result in networks with less favourable R/r-ratios, as will be seen.

In order to obtain the networks, we consider the largest subgraph of S_{gk} surrounded by C_i (see paragraph 6.3). We call this subgraph σ_i. So, σ_i consists of

all nodes on C_i plus all nodes properly surrounded by C_i. To prove that σ_i is convex we refer to lemmas 6.13 and 6.16 which stated that between any two nodes on C_i there exists no shortest path between them passing through C_j for $j>i$ ($i\geq 1$, $g>3$). These lemmas immediately imply that σ_i is convex. For S_{3k} ($k\geq 6$) a similar lemma is valid. To see this, we first state a lemma which is the analogue of lemmas 6.12 and 6.15.

(8.1) Lemma. Let u_i and w_i be nodes on C_i in S_{3k} ($k\geq 6$) and u_{i+1} and w_{i+1} be nodes on C_{i+1} ($i\geq 1$), such that $(u_i,u_{i+1}),(w_i,w_{i+1})\in E(\Gamma)$. Then,

$$d_{C_{i+1}}(u_{i+1},w_{i+1}) \geq d_{C_i}(u_i,w_i).$$

Proof. Let $d=d_{C_i}(u_i,w_i)$. It is easily established that

$$d_{C_{i+1}}(u_{i+1},w_{i+1}) \geq (d-1)(k-4)+2 \geq d. \qquad \square$$

For S_{3k} ($k\geq 6$) we now have the following lemma.

(8.2) Lemma. Let u and w be two nodes on C_i in S_{3k} ($k\geq 6$), then there is no shortest path between them through C_j for $j>i$ ($i\geq 1$).

Proof. Similar to the proof of lemma 6.13. $\square$

This lemma implies that any subgraph σ_i in S_{3k} ($k\geq 6$) as defined above is convex. So, $S_E(\sigma_i)$ is an extension sequence in S_{gk} consisting of convex networks. The R/r-ratio of σ_i is given by the following theorem.

(8.3) Theorem. The R/r-ratio of σ_i ($i\geq 1$) is determined by

(a) $R_{\sigma_i}/r_{\sigma_i}=1$ if $g=3$ and $k\geq 6$,

(b) $R_{\sigma_i}/r_{\sigma_i}=\lfloor g/2\rfloor$ if $g\geq 4$ and $k\geq 4$,

(c) $R_{\sigma_i}/r_{\sigma_i}=\lfloor g/2\rfloor$ and $R_{\sigma_i}/r_{\sigma_i}=(2.\lfloor g/2\rfloor+(i-2)(\lfloor g/2\rfloor-1))/(2i-1)$ ($i\geq 2$) if g is odd, $g\geq 7$ and $k=3$.

(d) $R_{\sigma_i}/r_{\sigma_i}=(g/2+(i-1)(g/2-1))/(2i-1)$ if g is even, $g\geq 6$ and $k=3$.

Proof.

(a) Lemma 6.10 states that all nodes on C_i have distance i from v ($i\geq 1$). From this we conclude $R_{\sigma_i}=r_{\sigma_i}=i$, giving an R/r-ratio of 1.

(b) In S_{gk} ($g\geq 4$, $k\geq 4$) there exist nodes on C_i at distance i from v ($i\geq 1$). Hence, $r_{\sigma_i}=i$.

To obtain R_{σ_i} first construct a node u_1 on C_1 at maximal distance from v. Since $k \geq 4$ and $g \geq 4$, there exists a facial circuit between C_1 and C_2 with u_1 as its only node on C_1. Let f_1 be this circuit. Construct a node u_2 on C_2 lying on f_1 at maximal distance from u_1 (see figure 8.1). Node u_1 lies on the shortest path from v to u_2.

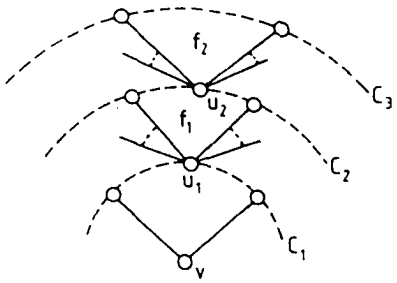


Figure 8.1 Node u_i and facial circuit f_i in S_{gk} ($g \geq 4$, $k \geq 4$).

Continuing this process results in a node u_i on C_i and a facial circuit f_i such that u_i is the only node on $C_i \cap f_i$ ($i=1,2,\ldots$). It is easily verified that u_i is a node on C_i at maximal distance from v. Furthermore, the distance between u_{i-1} and u_i is equal to $\lfloor g/2 \rfloor$. Hence, $R_{\sigma_i} = \lfloor g/2 \rfloor .i$, giving the required R/r-ratios.

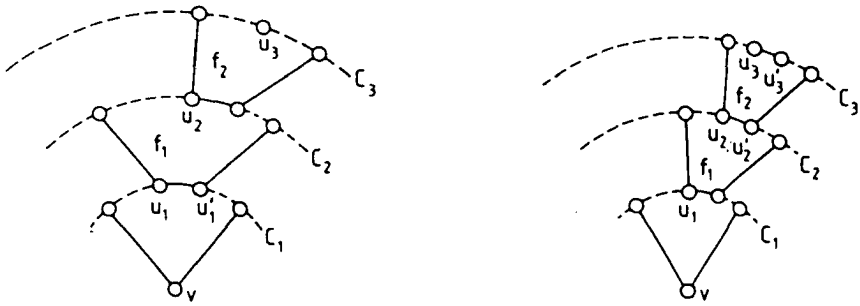
(c) In S_{g3} ($g \geq 7$) each node on C_i ($i \geq 2$) is connected to either a node on C_{i-1} or a node on C_{i+1} but not to both. From this it is easily deduced that $r_{\sigma_i} = 2i-1$.

To obtain R_{σ_i} we first construct a node u_1 on C_1 at maximal distance from v. Node u_1 has a neighbour u_1' on C_1 at distance $d(v,u_1)$ from v. Let f_1 be the facial circuit between C_1 and C_2 containing both u_1 and u_1'. Let u_2 be the node on $f_1 \cap C_2$ at maximal distance from u_1 (and u_1'), let f_2 be one of the facial circuits between C_2 and C_3 containing u_2, and let u_3 be the node on $f_2 \cap C_3$ at maximal distance from u_2 (see figure 8.2 (a)).

More in general, let f_i be one of the facial circuits between C_i and C_{i+1} containing u_i, and let u_{i+1} be the node on $f_i \cap C_{i+1}$ at maximal distance from u_i ($i=2,3,\ldots$). Though u_i does *not* lie on a shortest path from v to u_j ($j>i$), there exists no node on C_i lying at distance larger than $d(v,u_i)$ from v.

It is easily established that $d(v,u_1) = \lfloor g/2 \rfloor$. With some more effort we obtain

$$d(v,u_i) = 2.\lfloor g/2 \rfloor + (i-2)(\lfloor g/2 \rfloor - 1) \quad (i \geq 2).$$



(a) Odd g. (b) Even g.

Figure 8.2 Node u_i and facial circuit f_i in S_{g3} ($g \geq 6$).

Hence, $R_{\sigma_1} = \lfloor g/2 \rfloor$ and $R_{\sigma_i} = 2\lfloor g/2 \rfloor + (i-2)(\lfloor g/2 \rfloor - 1)$ for $i \geq 2$, giving the required R/r-ratios.

(d) Similarly to (c) we obtain $r_{\sigma_i} = 2i-1$ in S_{g3} ($g \geq 6$).
R_{σ_i} is determined as follows.
Node u_1 is determined in a similar way as in (b) (and (c)). Let f_1 be one of the facial circuits between C_1 and C_2 containing u_1, and let u_2' be the node on $f_1 \cap C_2$ at maximal distance from u_1 (see figure 8.2 (b)). Node u_i' has two neighbours on C_i ($i=2,3,\ldots$). Let u_i be the one with the largest distance from v, and let f_i be the facial circuit between C_i and C_{i+1} containing both u_i and u_i'. Furthermore, let u_{i+1}' be the node on $f_i \cap C_{i+1}$ at maximal distance from u_i. Node u_i lies on a shortest path from v to u_j ($1 \leq i < j$), and lies at maximal distance from v of all nodes on C_i. Then,

$$R_{\sigma_i} = d(v,u_i) = g/2 + (i-1)(g/2-1) \quad (i=1,2,\ldots),$$

which gives the required R/r-ratios. □

We conclude that the R/r-ratios of all σ_i are bounded from above by a constant. The R/r-ratios of σ_i in S_{3k} ($k \geq 6$) are 1, and so, optimal. The R/r-ratios of σ_i in supersymmetric graphs with girth $g > 3$ can be improved, as will be seen in the next paragraph.

8.3 Convex subgraphs of S_{gk}, a second approach

The convex subgraphs of S_{gk} constructed in this paragraph are convex hulls of balls. Theorem 2.23 indicated that such constructs promote low R/r-ratios. We shall prove that the R/r-ratios of convex hulls of balls

- are 1 if the underlying graph is S_{3k} ($k \geq 6$),
- are 2 if the underlying graph is S_{44},
- go asymptotically to 1 as r goes to ∞, for all other supersymmetric underlying graphs.

In order to prove this we construct a subgraph $\phi_r(v)$ of S_{gk}, which will appear to be equal to $[B_r(v)]$ for $(g,k) \neq (4,4)$. It is constructed as follows. Let $B_r(v)$ be the ball with radius r around node v in S_{gk}. Consider all facial circuits of S_{gk} having nodes in $\Gamma_r(v)$, in $\Gamma_{r-1}(v)$ and in $\Gamma_{r+1}(v)$. Let f be such a facial circuit. We call f a *closed* circuit with respect to $B_r(v)$ if there is a shortest path through f outside $B_r(v)$ between the two nodes in $f \cap \Gamma_r(v)$. We define $\phi_r(v)$ as the subgraph of S_{gk} consisting of $B_r(v)$ plus all facial circuits which are closed with respect to $B_r(v)$. Figure 8.3 depicts $B_4(v)$ and $\phi_4(v)$ in S_{73}.

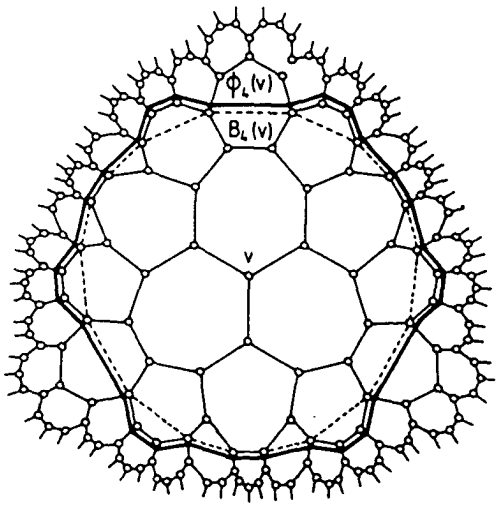


Figure 8.3 $B_4(v)$ and $\phi_4(v)$ in S_{73}.

In order to investigate the relationship between $\phi_r(v)$ and $[B_r(v)]$ we make extensive use of results of chapter 6, especially lemma 6.24 and knowledge about RD-labelings. For convenience, the feasible RD-combinations at nodes in $\Gamma_d(v)$ ($d \geq 2$), deduced in chapter 7, are summarized in table 8.1.

We shall prove the following theorem.

Even g:

$k=3, g=6$: 012, 113, 022
$k=3, g\geq 8$: 012, 013,..., 01m−1, 11m, 022
$k\geq 4, g=4$: $0^{k-2}11$, $0^{k-3}121$
$k\geq 4, g\geq 6$: $0^{k-2}11$, $0^{k-2}12$,..., $0^{k-2}1m-1$, $0^{k-3}1m1$

Odd g:

$k=3, g=7$: 012, $\underline{0}13$, $0\underline{1}2$, $01\underline{2}$, $11\underline{3}$, 022
$k=3, g\geq 9$: 012,..., 01m−1, $\underline{0}1m$, $0\underline{1}2$, $01\underline{2}$,..., $01\underline{m-1}$, $11\underline{m}$, 022
$k\geq 4, g=5$: $0^{k-2}11$, $\underline{0}0^{k-3}12$, $0^{k-2}\underline{1}1$, $0^{k-3}12\underline{1}$
$k\geq 4, g\geq 7$: $0^{k-2}11$,..., $0^{k-2}1m-1$, $\underline{0}0^{k-3}1m$, $0^{k-2}\underline{1}1$,..., $0^{k-2}1\underline{m-1}$, $0^{k-3}1m\underline{1}$

Table 8.1 The feasible RD-combinations in S_{gk} $(g>3)$.

(8.4) Theorem. The convex hull of $B_r(v)$ $(r=0,1,2,...)$ in S_{gk} is determined by

$$[B_r(v)]=\phi_r(v) \quad \text{if } (g,k)\neq(4,4).$$

(A yet incomplete) Proof. Theorem 8.3 implies that $[B_r(v)]=B_r(v)$ for $g=3$. It is easily seen that $\phi_r(v)=B_r(v)$, when $g=3$, proving theorem 8.4 for supersymmetric graphs with girth 3.
In the case $g>3$ the ball $B_r(v)$ is not convex in S_{gk}. Then there exist two nodes u_1 and u_2 on $\Gamma_r(v)$ and a shortest path P between them which lies entirely outside $B_r(v)$. Let Π be the subgraph of S_{gk} consisting of $B_r(v)\cup P$ and all the nodes properly surrounded by $B_r(v)\cup P$. Let F be the set of facial circuits in Π having one or more edges in common with P. We shall prove that F consists of one facial circuit. By definition of P and F, this facial circuit is closed with respect to $B_r(v)$. This result directly implies theorem 8.4. □

So, in order to complete the proof of theorem 8.4, we need to prove that any set F as defined above, consists of exactly one facial circuit. The proof that F consists of one facial circuit covers lemmas 8.5 to 8.16. We assume $g>3$ in these lemmas. In the first lemma we prove that all facial circuits in F lie side by side, as in figure 8.4 (a).

(8.5) Lemma. For every two different facial circuits $f_1,f_2\in F$: $P\cup f_1$ does not surround f_2.

Proof. Suppose f_1 and f_2 are two different facial circuits in F such that $P\cup f_1$ surrounds f_2. Then, there exists a subgraph α of S_{gk} surrounded by $P\cup f_1$, containing f_2 and not surrounding f_1. Let x_1 and x_2 be the two nodes on $f_1\cap P\cap\alpha$ (see figure 8.4 (b)). We know that $d_{f_1}(x_1,x_2)=d_{f_1\cap\alpha}(x_1,x_2)\leq\lfloor g/2\rfloor$, otherwise P

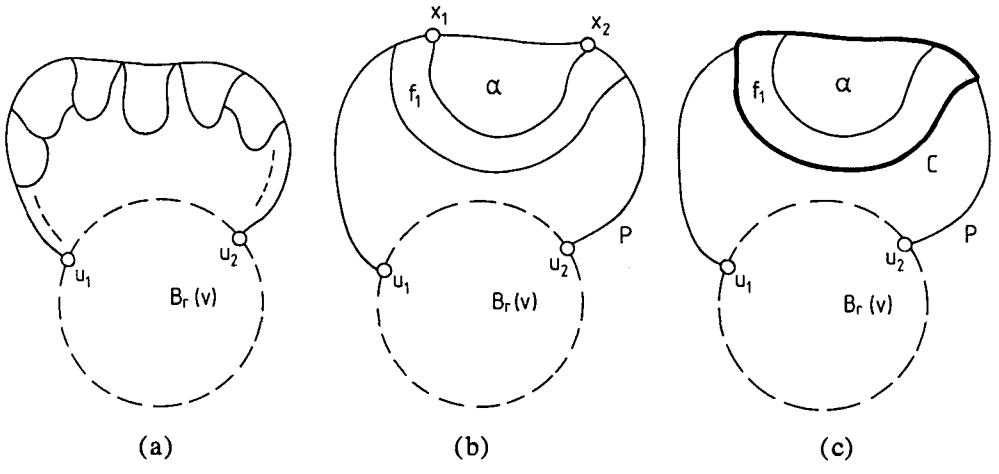


Figure 8.4 Illustration for lemma 8.5.

wouldn't pass through x_1 and x_2, but through the side of f_1 not on α. Furthermore, $d_P(x_1,x_2) \leq d_{f_1}(x_1,x_2)$, since P is a shortest path. Inasmuch as $(P \cap \alpha) \cup (f_1 \cap \alpha)$ is a circuit and the length of each circuit in S_{gk} is at least g, we obtain

$$d_P(x_1,x_2) + d_{f_1 \cap \alpha}(x_1,x_2) \geq g.$$

Substituting $d_P(x_1,x_2)$ and $d_{f_1 \cap \alpha}(x_1,x_2)$ in this expression, we conclude that g is even and $d_P(x_1,x_2) = d_{f_1 \cap \alpha}(x_1,x_2) = g/2$. Then, the smallest circuit C surrounding both f_1 and α has length g but is not facial, which is a contradiction (see figure 8.4 (c)). □

In order to reduce the number of facial circuits in F to one, we consider a node on P lying at maximal distance from v, and investigate its position with respect to the facial circuits in F.
Let w be a node on P at maximal distance from v (there may be more of such nodes). Any node properly surrounded by $B_r(v) \cup P$ lies at distance smaller than d(w,v) from v, since S_{gk} is smooth (use lemma 6.24 in the standard way to prove this).
Knowing that the facial circuits in F lie side by side, we conclude that w lies on P in one of the three ways depicted in figure 8.5.
We now have the following lemmas.

(8.6) Lemma. The situation in figure 8.5 (a) is impossible.

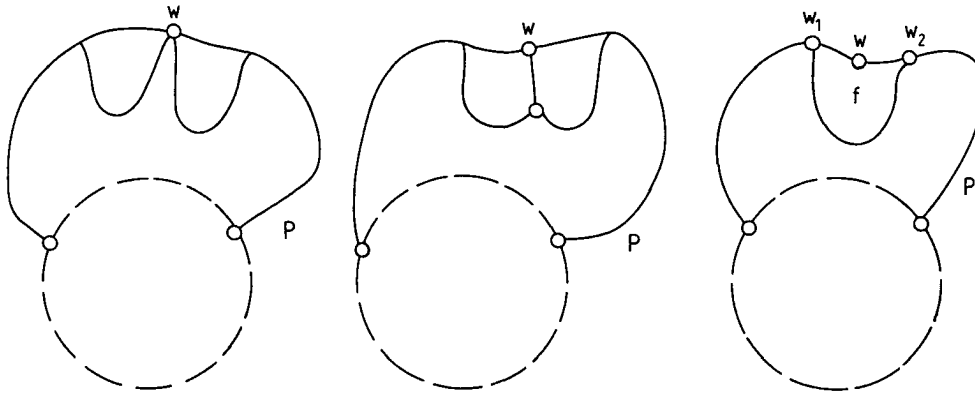


Figure 8.5 Impossible (a,b) and possible (c) positions of w.

Proof. Let f_1 and f_2 be the two facial circuits in F containing w. Since w is a node in Π at maximum distance from v,

$$\text{RD-l}(w,f_i)=m \text{ or } \underline{m} \quad (i=1,2).$$

Then, two m's, two $\underline{m}$'s or an m and an $\underline{m}$ occur simultaneously in the RD-combination of w, which is impossible for all g and k in table 8.1. □

(8.7) Lemma. The situation in figure 8.5 (b) is impossible.

Proof. Similar to the proof of the previous lemma. □

The situation in figure 8.5 (c) remains to be considered. Let f be the facial circuit in F to which w belongs. Clearly, RD-l(w,f) is m or $\underline{m}$. Let w_1 and w_2 be the nodes on $P \cap f$ having only one neighbour in $P \cap f$. For w_1 and w_2 we have the following lemma.

(8.8) Lemma. If g is even, or g is odd and f is a type I circuit, then

$$\text{RD-l}(w_1,f)+\text{RD-}(w_2,f) \geq \lceil g/2 \rceil.$$

If g is odd and f is a type II circuit, then

$$\text{RD-l}(w_1,f)+\text{RD-l}(w_2,f) \geq \underline{\lfloor g/2 \rfloor}.$$

Proof. Since P is a shortest path, we have $d_{f\cap P}(w_1,w_2) \leq d_{f-P}(w_1,w_2)$. This implies

$$d_{f\cap P}(w_1,w_2) \leq g/2.$$

Since w lies on $f \cap P$, we obtain

$m-RD\text{-}l(w_1,f)+m-RD\text{-}l(w_2,f) \leq g/2$, if g is even,

$m-RD\text{-}l(w_1,f)+1+m-RD\text{-}l(w_2,f) \leq g/2$, if g is odd and f is a type I circuit,

$\underline{m}-RD\text{-}l(w_1,f)+\underline{m}-RD\text{-}l(w_2,f) \leq \underline{\lfloor g/2 \rfloor}$, if g is odd and f is a type II circuit.

Substitution of $m=g/2$ for even g and $m=(g-1)/2$ for odd g and some calculations gives the required results. □

To prove that f is the only facial circuit in F, we have to prove that F contains no facial circuits lying next to f, i.e. that $w_1 \in B_r(v)$ and $w_2 \in B_r(v)$. If $w_i \notin B_r(v)$ $(i=1,2)$ then there exists a facial circuit f_i in F next to f and containing w_i. For this facial circuit we have the following lemma.

(8.9) Lemma. If f_i exists, then $|RD\text{-}l(w_i,f_i)| \geq |RD\text{-}l(w_i,f)|$

Proof. Suppose the lemma is not true, i.e. $|RD\text{-}l(w_i,f_i)|<|RD\text{-}l(w_i,f)|$. Let u_i be a node in f_i which has RD-label m or $\underline{m}$ in f_i. Then $d(u_i,w_i)>d(w,w_i)$, implying that $d(u_i,v)>d(w,v)$. This is a contradiction, since u_i lies in Π. □

This lemma will be of help to rule out several situations in the rest of the proof of theorem 8.4.
In the subsequence we prove that $w_1 \in B_r(v)$ and $w_2 \in B_r(v)$. It directly implies that F consists of only one circuit, which is the result we need. We consider the following cases:

A. case $g \geq 5$.

B. case $g=4$.

The first case is covered by lemmas 8.10 to 8.15.

A. case $g \geq 5$

(8.10) Lemma. If $(g,k) \neq (6,3)$ and $g \geq 5$, then the two conditions

- $w_1 \in B_r(v)$
- $w_2 \in B_r(v)$

are equivalent.

Proof. If $w_1 \in B_r(v)$ and $w_2 \notin B_r(v)$, then f_2 exists. We consider two cases.

Case $k \geq 4$. Lemma 8.9 implies that if $|RD\text{-}l(w_2,f)|>1$ then $|RD\text{-}l(w_2,f_2)|>1$. This results in an RD-combination on w_2 containing two RD-labels larger than 1, which is impossible by table 8.1. Hence,

$$\text{RD-l}(w_2,f) \leq 1 \text{ or } \text{RD-l}(w_2,f) \leq \underline{1}.$$

If $\text{RD-l}(w_2,f) \leq 1$, then $\text{RD-l}(w_1,f) \geq \lceil g/2 \rceil - 1$ by lemma 8.8. Analogously, $\text{RD-l}(w_2,f) \leq \underline{1}$ implies $\text{RD-l}(w_1,f) \geq \underline{\lfloor g/2 \rfloor - 1}$. In both cases we have $\text{RD-l}(w_1,f) \geq \text{RD-l}(w_2,f)$, implying that $d(w_1,v) \geq d(w_2,v)$. Therefore, $w_2 \in B_r(v)$.

Case $k=3$. Using lemma 8.9 and table 8.1 we obtain in a similar way to the previous case:

$$\text{RD-l}(w_2,f) \leq 2 \text{ or } \text{RD-l}(w_2,f) \leq \underline{1}.$$

The first part implies $\text{RD-l}(w_1,f) \geq \lceil g/2 \rceil - 2$ and the second part implies $\text{RD-l}(w_1,f) \geq \underline{\lfloor g/2 \rfloor - 1}$ by lemma 8.8.

We conclude that $\text{RD-l}(w_1,f) \geq \text{RD-l}(w_2,f)$ for all $g \geq 7$, implying that $d(w_1,v) \geq d(w_2,v)$. Therefore, $w_2 \in B_r(v)$.

In a similar way it is derived that $w_2 \in B_r(v)$ implies $w_1 \in B_r(v)$. □

In order to show that F consists of only the facial circuit f, we have to prove that $w_1 \in B_r(v)$ or $w_2 \in B_r(v)$, for $(g,k) \neq (6,3)$ and $g \geq 5$. In the next four lemmas we prove this.

(8.11) Lemma. If $k \geq 4$ and $g \geq 5$, then $w_1 \in B_r(v)$ or $w_2 \in B_r(v)$.

Proof. Suppose $w_1 \notin B_r(v)$ and $w_2 \notin B_r(v)$. Then f_1 and f_2 exist. By the relations in lemma 8.8

$$\text{RD-l}(w_1,f) \geq \underline{\lfloor g/2 \rfloor}/2 \quad \text{or} \quad \text{RD-l}(w_2,f) \geq \underline{\lfloor g/2 \rfloor}/2,$$

if g is odd and f is a type II circuit, and

$$\text{RD-l}(w_1,f) \geq \lceil g/2 \rceil/2 \quad \text{or} \quad \text{RD-l}(w_2,f) \geq \lceil g/2 \rceil/2$$

otherwise. Let's assume without loss of generality that

$$\text{RD-l}(w_1,f) \geq \underline{\lfloor g/2 \rfloor}/2 \geq \underline{1},$$

if g is odd and f is a type II circuit, and

$$\text{RD-l}(w_1,f) \geq \lceil g/2 \rceil/2 \geq 2,$$

otherwise. From table 8.1 we directly conclude that $\text{RD-l}(w_1,f_1)$ must be 0, $\underline{0}$ or 1, which is in contradiction with lemma 8.9. □

(8.12) Lemma. If $k=3$ and $g \geq 9$, then $w_1 \in B_r(v)$ or $w_2 \in B_r(v)$.

Proof. Suppose $w_1 \notin B_r(v)$ and $w_2 \notin B_r(v)$. Then f_1 and f_2 exist. By the relations in lemma 8.8

$$\text{RD-l}(w_1,f) \geq \underline{\lfloor g/2 \rfloor/2} \quad \text{or} \quad \text{RD-l}(w_2,f) \geq \underline{\lfloor g/2 \rfloor/2},$$

if g is odd and f is a type II circuit, and

$$\text{RD-l}(w_1,f) \geq \lceil g/2 \rceil/2 \quad \text{or} \quad \text{RD-l}(w_2,f) \geq \lceil g/2 \rceil/2,$$

otherwise. Let's assume without loss of generality that

$$\text{RD-l}(w_1,f) \geq \underline{\lfloor g/2 \rfloor/2} \geq 2,$$

if g is odd and f is a type II circuit, and

$$\text{RD-l}(w_1,f) \geq \lceil g/2 \rceil/2 \geq 3,$$

otherwise. From table 8.1 we directly conclude that RD-l(w_1,f_1) must be 0 or 1, which is in contradiction with lemma 8.9. □

(8.13) Lemma. If k=3 and g=8, then $w_1 \in B_r(v)$ or $w_2 \in B_r(v)$.

Proof. Suppose $w_1 \notin B_r(v)$ and $w_2 \notin B_r(v)$. Then f_1 and f_2 exist. We know that RD-l(w_i,f)≠4 (i=1,2), since f contains only one node with maximal RD-label, i.e. node w. Furthermore,

RD-l(w_1,f)=3 and RD-l(w_2,f)≥1 or

RD-l(w_1,f)≥2 and RD-l(w_2,f)≥2 or

RD-l(w_1,f)≥1 and RD-l(w_2,f)=3,

otherwise P is not a shortest path.
If RD-l(w_i,f)=3 for i = 1 or 2, then RD-l(w_i,f_i) must be 0 or 1 (see table 8.1), which is in contradiction with lemma 8.9. Hence, RD-l(w_1,f)=2 and RD-l(w_2,f)=2. Then, lemma 8.9 and table 8.1 imply

RD-l(w_1,f_1)=2 and RD-l(w_2,f_2)=2.

Hence, the nodes with RD-label 4 in f_1 and f_2 lie at distance d(w,v) from v (notice that RD-l(w,f)=4). This implies that all conditions we deduced for f can also be deduced for f_1 and f_2.
Doing the same for f_i as we did for f results in a facial circuit other than f next to f_i, of which both nodes with RD-label 2 lie on P. Continuing this process results in a sequence of facial circuits in F, of which all nodes with RD-label 2 lie on P (see figure 8.6).
We conclude that for the two facial circuits in F containing u_1 and u_2 (which are called g_1 and g_2 respectively)

RD-l(u_i,g_i)=2 (i=1,2).

Since u_i lies in $B_r(v)$, all nodes with RD-label 2 in a facial circuit of F lie in $B_r(v)$.

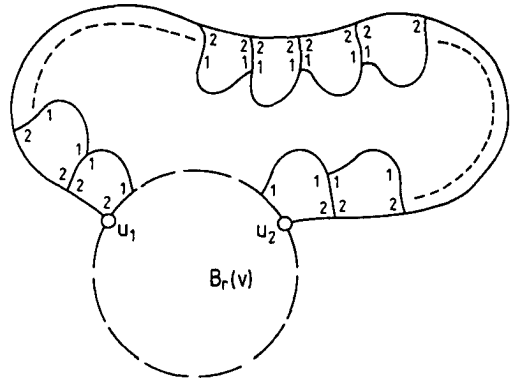


Figure 8.6 Illustration for lemma 8.13.

This is a contradiction, since these nodes lie on P and P lies entirely outside $B_r(v)$. □

(8.14) Lemma. If k=3 and g=7, then $w_1 \in B_r(v)$ or $w_2 \in B_r(v)$.

Proof. Suppose $w_1 \notin B_r(v)$ and $w_2 \notin B_r(v)$. Then f_1 and f_2 exist. If f is a type II circuit, then w is the only node in f with maximal RD-label. Hence, RD-l(w_i,f) $\neq \underline{3}$ (i=1,2). Furthermore,

RD-l(w_1,f) $\geq$ 1 and RD-l(w_2,f) = 3 or
RD-l(w_1,f) $\geq$ 2 and RD-l(w_2,f) $\geq$ 2 or
RD-l(w_1,f) = 3 and RD-l(w_2,f) $\geq$ 1 or
RD-l(w_1,f) $\geq \underline{1}$ and RD-l(w_2,f) = $\underline{2}$ or
RD-l(w_1,f) = $\underline{2}$ and RD-l(w_2,f) $\geq \underline{1}$,

otherwise P is not a shortest path.
If RD-l(w_i,f)=3 for i = 1 or 2, then RD-l(w_i,f_i) must be $\underline{0}$ or 1 (see table 8.1), which is in contradiction with lemma 8.9. If RD-l(w_i,f) = $\underline{2}$ for i = 1 or 2, then RD-l(w_i,f_i) must be 0 or 1 (see table 8.1), which is also in contradiction with lemma 8.9.
Hence, RD-l(w_1,f)=2 and RD-l(w_2,f)=2. From this point on we obtain a contradiction in the same way as in the proof of lemma 8.13. □

The previous five lemmas proved that F consists of the facial circuit f only, for g$\geq$5 and (g,k)$\neq$(6,3). The hexagonal grid must be considered as a separate case. Lemma 8.10 cannot be applied to it, but in this case we can manage without it.

(8.15) Lemma. If $k=3$ and $g=6$, then $w_1 \in B_r(v)$ and $w_2 \in B_r(v)$.

Proof. Suppose $w_1 \notin B_r(v)$, then f_1 exists. We know that RD-l$(w_i,f) \neq 3$ $(i=1,2)$, since f contains only one node with maximal RD-label, i.e. node w. Furthermore,

RD-l$(w_1,f)=2$ and RD-l$(w_2,f) \geq 1$ or

RD-l$(w_1,f) \geq 1$ and RD-l$(w_2,f)=2$,

otherwise P is not a shortest path.
In order to prove that RD-l$(w_2,f)=2$, we consider all facial circuits in F from f up to the facial circuit in F containing u_1 (call the latter facial circuit g_1). If RD-l$(w_2,f)=1$, then each node with RD-label 1 in any of these circuits does not lie on P. To see this, assume that f' is a facial circuit in F between f and g_1 having a node x with RD-label 1 on P, such that there is no facial circuit in F between f and f' having a node with RD-label 1 on P. Then, node x does not lie on the facial circuit in F next to f' between f' and f, for otherwise the distance to v of the node with RD-label 3 in f' would be greater than d(w,v). So, node x lies on f' in the way illustrated by figure 8.7. Then, the shortest path between x and w_2 is not the one lying on P, but the one passing through the zeros of all facial circuits in F between f' and f. This is a contradiction.

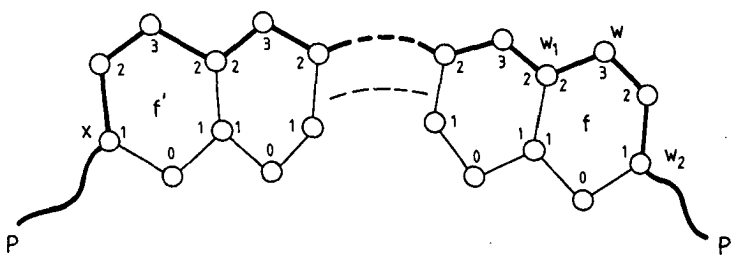


Figure 8.7 Situation occurring in lemma 8.15.

We conclude that if RD-l$(w_2,f)=1$, then each node on P in any of the facial circuits from f up to g_1 has RD-label 2 or 3 in that circuit. This implies that RD-l$(u_1,g_1) \geq 2$. From this we conclude that all nodes with RD-label 2 on the facial circuits from g_1 up to f lie at distance at most $d(u,v) \leq r$ to v. So, they lie in $B_r(v)$. Since they also lie on P, we obtain a contradiction. Hence, RD-l$(w_2,f)=2$.

Since $1 \leq$ RD-l$(w_1,f) \leq 2$, the condition $w_2 \in B_r(v)$ implies $w_1 \in B_r(v)$, which is a contradiction. Hence, $w_2 \notin B_r(v)$. Then by arguments of symmetry RD-

$l(w_1,f)=2$. From this point on we obtain a contradiction in an analogous way as in the proof of lemma 8.13. □

From lemmas 8.10, 8.11, 8.12, 8.13, 8.14 and 8.15 we conclude that the condition

$w_1 \notin B_r(v)$ or $w_2 \notin B_r(v)$

can not be satisfied when $g \geq 5$. Hence, F consists of only the facial circuit f when $g \geq 5$. The proof that this is also the case for $g=4$ is as follows.

B. case $g=4$

In this case f consists of four nodes: w, w_1, w_2 and a zero, called z. First of all, we have an almost trivial lemma.

(8.16) Lemma. $w_1 \in B_r(v)$ if and only if $w_2 \in B_r(v)$.

Proof. RD-$l(w_1,f)$=RD-$l(w_2,f)=1$. Hence, $d(w_1,v)=d(w_2,v)$, directly giving the required result. □

(8.17) Lemma. If $g=4$ and $k \geq 5$, then $w_1 \in B_r(v)$ or $w_2 \in B_r(v)$.

Proof. Suppose $w_1 \notin B_r(v)$ and $w_2 \notin B_r(v)$. Node z lies at distance $d(w,v)-2$ from v. Lemmas 7.5 and 7.6 imply that there are at least $k-2$ neighbours of z at distance $d(w,v)-1$ from v. Nodes w_1 and w_2 are two of them. Since $k \geq 5$ there is at least another one. Let x be this node.
Node x does not lie on P because this would imply that the path x, z, w_2 is shorter than the subpath x,..., w_1, w, w_2 of P, or the path x, z, w_1 is shorter than the subpath x,..., w_2, w, w_1 of P. Node x does lie inside $\Pi - B_r(v)$ because S_{4k} is planar, $w_1 \in P$, $w_2 \in P$ and $d(x,v)=d(w_i,v)$ $(i=1,2)$.
Since S_{4k} is smooth, node x has a neighbour y at distance d(w,v) from v. Inasmuch as x lies in $\Pi - P - B_r(v)$ - and so, x is not a border node of Π - node y does not lie outside Π. By lemma 6.24 node y is not properly surrounded by the circuit consisting of the border nodes of $B_r(v) \cup P$, because $d(y,v)=d(w,v)$. Hence, node y lies on P. This implies that node y lies on a facial circuit of F and y has RD-label 2 in this circuit, since $d(y,v)=d(w,v)$. However, the only possible situation is that in figure 8.5 (c) (see lemmas 8.6 and 8.7), i.e. the neighbour(s) of y at distance $d(w,v)-1$ from v lie(s) on P or outside Π. This implies that x lies on P or outside Π, which is a contradiction. □

We conclude that $w_1 \in B_r(v)$ and $w_2 \in B_r(v)$ in the case $g=4$ and $k \geq 5$. Hence, F consists of only the facial circuit f, if $g=4$ and $k \geq 5$.

Reconsidering the incomplete proof of theorem 8.4, we notice that the only open piece to be filled in was proving that F consists of only one facial circuit. Since this is done for the cases $g \geq 5$ and $g=4$ ($(g,k) \neq (4,4)$), the proof of theorem 8.4 is completed.

We may wonder whether $[B_r(v)] = \phi_r(v)$ in S_{44}. This is not true for $r \geq 2$. Figure 8.8 illustrates the case $r=2$.

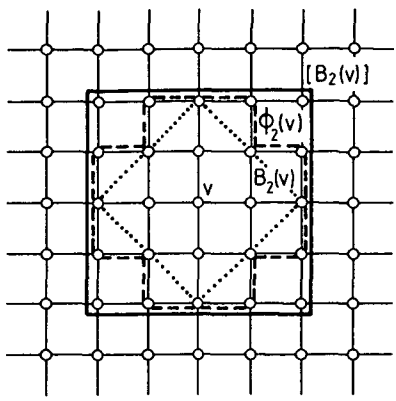


Figure 8.8 $B_2(v)$, $\phi_2(v)$ and $[B_2(v)]$ in S_{44}.

We are now in the position to prove the following theorem.

(8.18) Theorem. If $B_i(v)$ is a ball in S_{gk} ($i \geq 1$), then

(a) $R_{[B_i(v)]}/r_{[B_i(v)]} = 2$ if $g=k=4$,

(b) $R_{[B_i(v)]}/r_{[B_i(v)]} \leq 1 + \lfloor g/4 \rfloor / i$ otherwise.

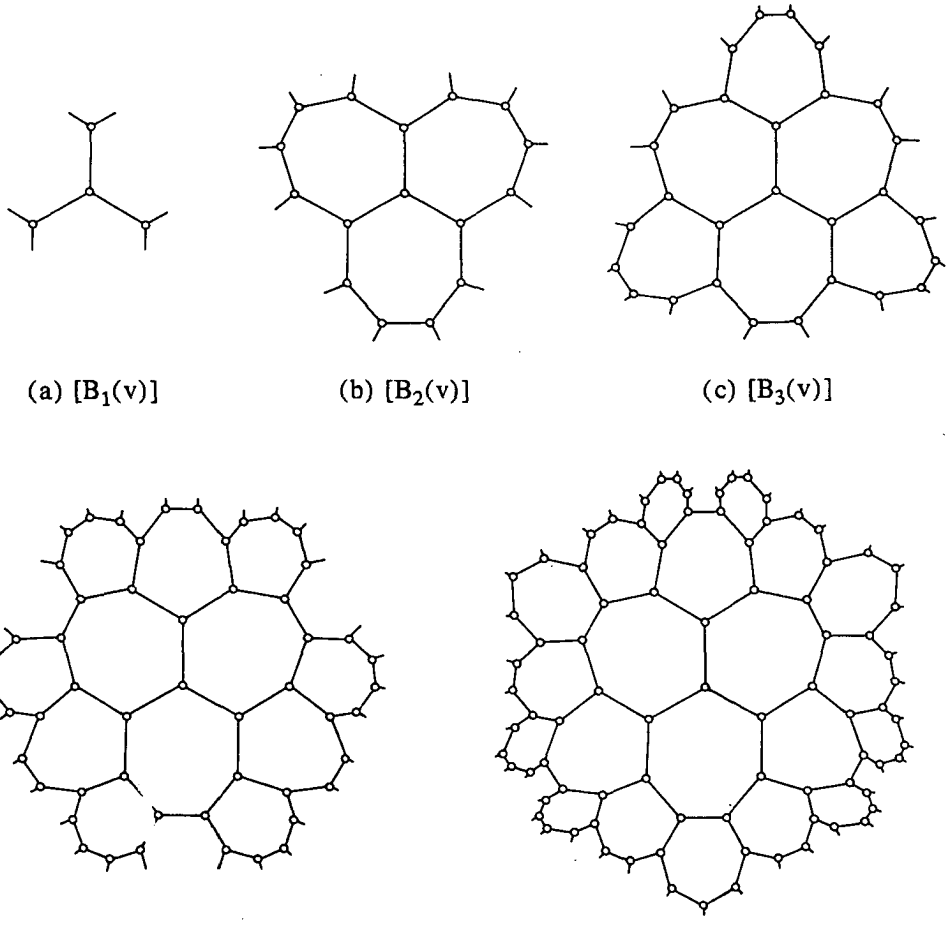
Proof.

(a) S_{44} is identical to the tree-mesh T_2^2. We concluded in paragraph 4.3 that convex hulls of balls in T_2^2 have R/r-ratio 2.

(b) If $g=3$ then from the convexity of $B_r(v)$ in S_{3k} ($k \geq 6$), we directly conclude that $R_{[B_r(v)]}/r_{[B_r(v)]} = 1$. Supersymmetric graphs with girth $g>3$ ($(g,k) \neq (4,4)$) are dealt with as follows.
For $g>3$ and $(g,k) \neq (4,4)$ we have $[B_i(v)] = \phi_i(v)$. Every node on $\phi_i(v) - B_i(v)$ lies on a shortest path between two nodes in $\Gamma_i(v)$ lying on the same facial circuit. The length of this path is at most $(g-1)/2$ for odd g and $2.\lfloor g/4 \rfloor$ for even g.
For odd g the maximum length of the path is achieved for type II circuits.

The distance to v of a node with RD-label m in such a type II circuit does not exceed $\lfloor g/4 \rfloor + i$.
For even g the maximum distance to v of a node on the shortest path is also $\lfloor g/4 \rfloor + i$.
Hence, $R_{[B_i(v)]} \leq i + \lfloor g/4 \rfloor$. Furthermore, $r_{[B_i(v)]} \geq i$, which gives the required result. □

Figure 8.9 shows the subgraphs $[B_1(v)]$, $[B_2(v)]$, $[B_3(v)]$, $[B_4(v)]$ and $[B_5(v)]$ of S_{73}.



(a) $[B_1(v)]$ (b) $[B_2(v)]$ (c) $[B_3(v)]$

(d) $[B_4(v)]$ (e) $[B_5(v)]$

Figure 8.9 Convex hulls of balls in S_{73}.

It is easily verified that $R_{[B_i(v)]} = r_{[b_i(v)]} = 1$, $R_{[B_i(v)]} = i+1$ and $r_{[B_i(v)]} = i$ for

$i=2,3,4,5$.

For $(g,k)\neq(4,4)$ and $g\neq 3$ the R/r-ratio of $[B_i(v)]$ goes asymptotically to 1 as i goes to ∞. That is almost optimal. In chapter 7 we concluded that the exponentiality of S_{gk} is almost optimal too. Considering theorem 2.19, we conclude that the extension sequence $S_E([B_i(v)])$ is not subject to much improvement with respect to the diameters of its elements. The table in appendix C shows the number of nodes in $\phi_r(v)$. The reader is invited to compare $\phi_r(v)$ in S_{gk} with the balls $B_r(v)$ and $B_R(v)$ in $S_{\infty k}$, R being the radius of the circumscribed ball of $\phi_r(v)$ in S_{gk}. In $S_{\infty k}$ balls are optimal extensible networks with respect to their diameter.

The extension complexity of $[B_i(v)]$ in S_{gk} is of the same order as the number of nodes in $[B_i(v)]$. This is easily seen for $(g,k)=(4,4)$ (see paragraph 4.3). To see it for the cases $(g,k)\neq(4,4)$, we first compare the number of nodes in $[B_i(v)]$ with the number of nodes in $B_i(v)$. The number of facial circuits which are closed with respect to $B_i(v)$ does not exceed $|\Gamma_i(v)|$. Furthermore, the number of nodes outside $B_i(v)$ in each of these circuits is at most $(g-3)/2$ for odd g and $2.\lfloor g/4\rfloor-1$ for even g (see proof of theorem 8.18), which is constant. So, the number of nodes in $[B_i(v)]$ is of the same order as the number of nodes in $B_i(v)$. Furthermore, the number of nodes in $B_{i+1}(v)$ is linear in the number of nodes in $B_i(v)$, implying that the number of nodes in $[B_{i+1}(v)]$ is of the same order as the number of nodes in $[B_i(v)]$.

It seems that the extension complexity can even further be improved for subgraphs of supersymmetric graphs with girth $g>3$, $(g,k)\neq(4,4)$ and $(g,k)\neq(6,3)$. To see this we notice that $[B_{i+1}(v)]$ can be obtained from $[B_i(v)]$ by adding the facial circuits to $[B_i(v)]$ which are

- not yet in $[B_{i+1}(v)]$,
- closed with respect to $B_{i+1}(v)$.

The facial circuits can be added one by one without disturbing convexity, whenever $g>3$, $(g,k)\neq(4,4)$ and $(g,k)\neq(6,3)$. This results in convex subgraphs of S_{gk}, being interjacent between $[B_i(v)]$ and $[B_{i+1}(v)]$. These convex subgraphs together with the convex hulls of balls constitute a new extension sequence of which the elements have smaller extension complexity.
Figure 8.10 shows the nodes in $[B_4(v)]-[B_3(v)]$ in S_{73}.
Each of the components in $[B_4(v)]-[B_3(v)]$ can be independently added to $[B_3(v)]$, resulting in a sequence of convex subgraphs of S_{73} interjacent between $[B_3(v)]$ and $[B_4(v)]$. Since the number of nodes in each of the components in $[B_4(v)]-[B_3(v)]$ is constant, we can define a new extension sequence based on $[B_1(v)]$, $[B_2(v)]$,... and their interjacent convex subgraphs. The extension

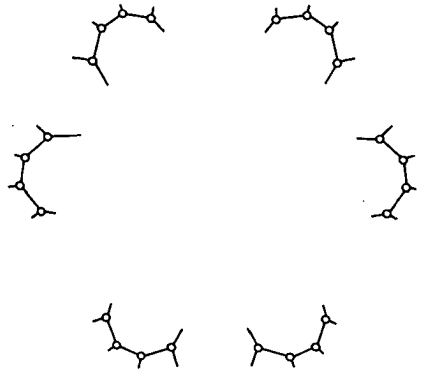


Figure 8.10 $[B_4(v)]-[B_3(v)]$ in S_{73}.

complexity of each of the networks in the new sequence is constant, and so, optimal. This seems to be true in general for supersymmetric graphs for which $g>3$, $(g,k)\neq(4,4)$ and $(g,k)\neq(6,3)$.

8.4 Concluding remarks

In this chapter we have constructed two classes of convex subgraphs of supersymmetric graphs. The subgraphs in both classes have a constant upper bound to their R/r-ratio.

The first class is a residue of the constructions of Grünbaum and Shephard in chapter 6. The upper bound for the R/r-ratio is 1 for the subgraphs of supersymmetric graphs with girth equal to 3. For subgraphs of other supersymmetric graphs the upper bound is not very close, however.

The subgraphs in the second class are convex hulls of balls. Their R/r-ratio is very close to 1 when $(g,k)\neq(4,4)$. Since the exponentiality of supersymmetric graphs is almost optimal, we conclude that the diameter of these subgraphs is very low. The extension complexity of an element of an extension sequence of convex hulls of balls in a supersymmetric graph is of the same order as the number of nodes in the element. It seems that new extension sequences based on convex hulls of balls can be constructed, for which the extension complexity is optimal whenever $g>3$, $(g,k)\neq(4,4)$ and $(g,k)\neq(6,3)$.

APPENDICES

A

Notions and notations used

Graph-theoretical notions and notations

General references for graph theory are [Harary] and [Wilson]. All graph-theoretical notions in this dissertation concern *undirected* graphs.

Node set and edge set. The node set of a graph Γ is denoted by $V(\Gamma)$ and the edge set by $E(\Gamma)$.

Incidence, adjacency. A node v in a graph Γ is *incident* to an edge $e \in E(\Gamma)$ if there exists a node $u \in V(\Gamma)$ such that $(v,u)=e$. An edge e is incident to a node v if v is incident to e.
Two nodes in a graph Γ are *adjacent* if there exists an edge in Γ incident to both nodes. Two adjacent nodes are called *neighbours*. Two edges in Γ are *adjacent* if there exists a node in Γ incident to both edges.

Subgraph, induced subgraph. A *subgraph* of a graph Γ is a graph, all of whose nodes belong to $V(\Gamma)$ and all of whose edges belong to $E(\Gamma)$. The expression $\Delta \subseteq \Gamma$ denotes that Δ is a subgraph of Γ, and $\Delta \subset \Gamma$ denotes that Δ is a *proper* subgraph of Γ, i.e. that $\Delta \subseteq \Gamma$ and $\Delta \neq \Gamma$.
For any set $S \subseteq V(\Gamma)$, the *induced subgraph* $<S>$ is the maximal subgraph of the graph Γ with node set S. So, two nodes of S are adjacent in $<S>$ if and only if they are adjacent in Γ.

Operations on subgraphs. If Δ_1 and Δ_2 are subgraphs subgraphs of a graph Γ, then $\Delta_1 \cap \Delta_2$ is an induced subgraph of Γ having node set $V(\Delta_1 \cap \Delta_2) = V(\Delta_1) \cap V(\Delta_2)$. Furthermore, $\Delta_1 \cup \Delta_2$ is an induced subgraph of Γ with node set $V(\Delta_1 \cup \Delta_2) = V(\Delta_1) \cup V(\Delta_2)$, and $\Delta_1 - \Delta_2$ is an induced subgraph of Γ with node set $V(\Delta_1 - \Delta_2) = V(\Delta_1) - V(\Delta_2)$.

Degree. The *degree* of a node u in a graph Γ, denoted by deg(u), is the number of edges in $E(\Gamma)$ incident to u. The maximum of all node degrees in Γ is denoted

by $\deg(\Gamma)$, and the minimum of all node degrees in Γ is denoted by $\deg^-(\Gamma)$. If u is a node in a subgraph Δ of Γ, then the *degree* of u *in* Δ, denoted by $\deg_\Delta(u)$, is equal to the number of edges in $E(\Delta)$ incident to u.

Infinite graphs. A graph Γ is *infinite* if $|V(\Gamma)|$ or $|E(\Gamma)|$ is infinite. A graph is *infinite locally finite* if it is infinite and all its nodes have finite degrees.

Paths, circuits. The subgraph P of a graph Γ is a *path* if its node set can be written as the set of different nodes $\{u_0,u_1,\ldots,u_n\}$, and its edge set can be written as the set of different edges $\{(u_0,u_1),(u_1,u_2),\ldots,(u_{n-1},u_n)\}$. It is a path between the nodes u_0 and u_n in Γ. If $u_0=u_n$, then P is called a *circuit* in Γ.
The *length* of P is defined to be equal to $|E(P)|$. The length of the smallest circuit of a graph Γ is called the *girth* of Γ.
Two paths between two nodes u and v are *node-disjoint* if they have no nodes but u and v in common. Two paths between u and v are *edge-disjoint* if they have no edges but u and v in common.
An *infinite path* is a path of infinite length. A *2-way infinite path* or *2-way path* or simply *2-path* is an infinite path of which deletion of any node results in two infinite paths. A *1-way infinite path* or *1-way path* or simply *1-path* is an infinite path which is not a 2-path. Clearly, deletion of any node of a 2-path results in two 1-paths.

Distance. The *distance* between two nodes u and v in a graph Γ, denoted by $d(u,v)$, is the length of a shortest path in Γ between u and v. The *distance* between two nodes u and v in a *subgraph* Δ of Γ, denoted by $d_\Delta(u,v)$, is the length of a shortest path in Δ between u and v.

Diameter. The *diameter* of a graph Γ is defined by

$$\operatorname{diam}(\Gamma) = \max_{u,v\in V(\Gamma)} d(u,v).$$

Connectedness, components. A graph Γ is *connected* if there exists a path between each two of its nodes.
A *component* of Γ is a connected maximal subgraph of Γ.

Separating set, disconnecting set. A *separating set* of a connected graph Γ is a set of nodes of Γ whose deletion disconnects Γ (when we delete a node of Γ then we also delete its incident edges). A *disconnecting set* of a connected graph Γ is a set of edges of Γ whose deletion disconnects Γ.

Connectivity. The *node-connectivity* $\kappa(\Gamma)$ of a connected graph Γ is the size of the smallest separating set of Γ. The node-connectivity of a graph is sometimes simply denoted as the *connectivity* of the graph. The *edge-connectivity* $\lambda(\Gamma)$ of a connected graph Γ is the size of the smallest disconnecting set of Γ. For connectivity

the following relation is known: $\kappa(\Gamma) \leq \lambda(\Gamma) \leq \deg^-(\Gamma)$. The node-connectivity of Γ is *optimal* if $\kappa(\Gamma)=\deg^-(\Gamma)$. The edge-connectivity of Γ is *optimal* if $\lambda(\Gamma)=\deg^-(\Gamma)$.

Local connectivity. The *local node-connectivity* of two different nodes u and v in a graph Γ, denoted by $\kappa(u,v)$, is the maximum number of node-disjoint paths between u and v. The *local edge-connectivity* of two different nodes u and v in Γ, denoted by $\lambda(u,v)$, is the maximum number of edge-disjoint paths between u and v. For local connectivity the following relation is known: $\kappa(u,v) \leq \lambda(u,v) \leq \min(\deg(u),\deg(v))$. $\kappa(u,v)$ and $\lambda(u,v)$ are called *optimal* if they are equal to $\min(\deg(u),\deg(v))$.
The *local node-connectivity* of two different nodes u and v in a subgraph Δ of Γ, denoted by $\kappa_\Delta(u,v)$, is the maximum number of node-disjoint paths in Δ between u and v. The *local edge-connectivity* of u and v in Δ, denoted by $\lambda_\Delta(u,v)$, is the maximum number of edge-disjoint paths in Δ between u and v. $\kappa_\Delta(u,v)$ and $\lambda_\Delta(u,v)$ are called *optimal* if they are equal to $\min(\deg_\Delta(u),\deg_\Delta(v))$.

Coherency. The *node-coherency* of an infinite connected graph Γ, denoted by $\kappa_\infty(\Gamma)$, is the size of the smallest node set which separates two infinite subgraphs of Γ.
The *edge-coherency* of an infinite connected graph Γ, denoted by $\lambda_\infty(\Gamma)$, is the size of the smallest edge set which disconnects two infinite subgraphs of Γ.

Balls. Let Γ be a graph and v a node in it, then $\Gamma_i(v)$ denotes the set of nodes having distance i from v. A *ball* with *radius* r and *centre node* v in a graph Γ is an induced subgraph of Γ with node set equal to

$$\bigcup_{i=0}^{r} \Gamma_i(v).$$

Such a ball is denoted by $B_r(v)$. An *inscribed ball* of a subgraph Δ of Γ is a subgraph of Δ being a ball with maximal radius. It is *trivial* if it consists of only one node. A *circumscribed ball* of a subgraph Δ of Γ is a ball with minimal radius having Δ as its subgraph.

Planarity. A graph is *planar* if it can be drawn on a plane so that no two edges intersect. The regions arising from drawing a planar graph in a particular way on the plane are called *faces*. The circuit adjoining a face is a *facial circuit*. Two facial circuits are *adjacent* if they have an edge in common.

Automorphism groups on graphs. An *automorphism* on a graph Γ is a permutation π of $V(\Gamma)$ which has the property that $(u,v) \in E(\Gamma)$ if and only if $(\pi(u),\pi(v)) \in E(\Gamma)$, for all nodes $u,v \in V(\Gamma)$. The *automorphism group* $G(\Gamma)$ of a graph Γ is the group of all automorphisms of Γ. A *stabilizer subgroup* of $G(\Gamma)$,

fixing a node v of Γ, is a subgroup of $G(\Gamma)$, denoted by $G_v(\Gamma)$, such that for each automorphism $\pi \in G_v(\Gamma)$: $\pi(v)=v$. A subgroup $G' \subseteq G(\Gamma)$ is *transitive* on a subset $V' \subseteq V(\Gamma)$ if for each two nodes $u,v \in V'$ there exists an automorphism $\pi \in G'$ such that $\pi(u)=v$.
Γ is *node-transitive* if for each two of its nodes, u and v, there exists an automorphism π such that $u=\pi(v)$.
Γ is *edge-transitive* if for each two of its edges (u,v) and (x,y) there exists an automorphism π such that $u=\pi(x)$ and $v=\pi(y)$ or $u=\pi(y)$ and $v=\pi(x)$.
Γ is *symmetric* if for each four of its nodes u, v, x and y such that $(u,v) \in E(\Gamma)$ and $(x,y) \in E(\Gamma)$ there exists an automorphism π such that $u=\pi(x)$ and $v=\pi(y)$. A node-transitive graph is symmetric if and only if for each $v \in V(\Gamma)$ each stabilizer subgroup $G_v(\Gamma)$ of $G(\Gamma)$ is transitive on $\Gamma_1(v)$.
Γ is *distance-transitive* if for each four of its nodes u, v, x and y such that $d(u,v)=d(x,y)$ there exists an automorphism π such that $u=\pi(x)$ and $v=\pi(y)$.
More details about automorphisms and transitivity can be found in the book by Biggs [Biggs; chapters 15-17].

Parallel computing

Speed-up, efficiency. The *speed-up* achieved by a parallel algorithm running on p processors is the ratio between the time taken by that parallel computer executing the fastest serial algorithm for a problem and the time taken by the same parallel computer executing the parallel algorithm on the same problem using p processors.
The *efficiency* of a parallel algorithm running on p processors is the speed-up divided by p.

General

Order-notation. Let $f,g: \mathbb{R}^+ \to \mathbb{R}^+$ be two functions, then

1. f is of *O-order* or simply *order* g, denoted as $f(x)=O(g(x))$, if and only if for every constant $a \in \mathbb{R}^+$ there exists a real number $m_a \in \mathbb{R}^+$ such that $f(x) \leq a.g(x)$ for all $x \geq m_a$.

2. f is of Ω*-order* g, denoted as $f(x)=\Omega(g(x))$, if and only if for every constant $a \in \mathbb{R}^+$ there exists a real number $m_a \in \mathbb{R}^+$ such that $f(x) \geq a.g(x)$ for all $x \geq m_a$.

Let $f,g: \mathbb{R} \to \mathbb{R}$ be two functions, then f is *proportional* to g, denoted as $f(x) \sim g(x)$, if and only if for every constant $a \in \mathbb{R}^+$ there exist two numbers $\epsilon_a, m_a \in \mathbb{R}^+$ such that

$$|f(x)-a.g(x)| \leq \epsilon_a$$

for all $x \geq m_a$.

Descente Infinie. *Descente Infinie* is an inverse induction method. It is used to prove that a certain logical predicate $T: \mathbb{N} \to \{0,1\}$ is false. It proceeds by proving that

a. $\neg T(a)$ for some $a \in \mathbb{N}$,

b. $T(n) \to T(n-1)$ for all $n \in \mathbb{N}$.

From this it follows directly that $T(n)$ is false for all $n \in \mathbb{N}$.

B

Table with uniform exponentialities of supersymmetric graphs

This appendix contains a table of the uniform exponentialities of supersymmetric graphs.

g\k	3	4	5	6	7	8
3				1.000000	2.618034	3.732051
4		1.000000	2.618034	3.732051	4.791288	5.828427
5		2.296630	3.506068	4.611582	5.677983	6.724268
6	1.000000	2.618034	3.732051	4.791288	5.828427	6.854102
7	1.556030	2.823202	3.897980	4.932826	5.952253	6.964266
8	1.722084	2.890054	3.938691	4.960693	5.972624	6.979836
9	1.831076	2.946995	3.975944	4.987046	5.992235	6.994984
10	1.883204	2.965573	3.985135	4.992273	5.995485	6.997139
11	1.926067	2.983067	3.994094	4.997433	5.998712	6.999286
12	1.946856	2.988825	3.996321	4.998462	5.999249	6.999592
13	1.965636	2.994452	3.998532	4.999488	5.999786	6.999898
14	1.974819	2.996316	3.999083	4.999693	5.999875	6.999942
15	1.983512	2.998163	3.999634	4.999898	5.999964	6.999985
16	1.987793	2.998777	3.999771	4.999939	5.999979	6.999992
17	1.991948	2.999389	3.999908	4.999980	5.999994	6.999998
18	1.994004	2.999593	3.999943	4.999988	5.999997	6.999999
19	1.996027	2.999797	3.999977	4.999996	5.999999	
20	1.997032	2.999864	3.999986	4.999998	5.999999	

g\k	9	10	11	12	13	14
3	4.791288	5.828427	6.854102	7.872983	8.887482	9.898979
4	6.854102	7.872983	8.887482	9.898979	10.908327	11.916080
5	7.758593	8.785151	9.806352	10.823690	11.838144	12.850385
6	7.872983	8.887482	9.898979	10.908327	11.916080	12.922616
7	7.972235	8.977798	9.981838	10.984866	11.987194	12.989023
8	7.984530	8.987757	9.990071	10.991786	11.993092	12.994110
9	7.996574	8.997558	9.998198	10.998633	11.998938	12.999159
10	7.998076	8.998644	9.999010	10.999255	11.999425	12.999547
11	7.999573	8.999729	9.999820	10.999876	11.999912	12.999935
12	7.999760	8.999849	9.999901	10.999932	11.999952	12.999965
13	7.999947	8.999970	9.999982	10.999989	11.999993	12.999995
14	7.999970	8.999983	9.999990	10.999994	11.999996	12.999997
15	7.999993	8.999997	9.999998	10.999999	11.999999	
16	7.999996	8.999998	9.999999	10.999999		
17	7.999999					

C

Table with sizes of $\phi_r(v)$

This appendix contains a table which describes the number of nodes in the subgraph $\phi_r(v)$ of S_{gk} as function of r. In addition to that it describes the value of the radius R of the circumscribed ball of $\phi_r(v)$.
To compute the number of nodes in $\phi_r(v)$, first the number of nodes in $B_r(v)$ is determined. Thereupon, the number of nodes outside $B_r(v)$ in the facial circuits being closed with respect to $B_r(v)$ is determined. The procedure for the latter is straightforward, since from the RD-combination of a node u in $\Gamma_r(v)$, it can easily be established for each of the facial circuits to which u belongs whether the circuit is closed with respect to $B_r(v)$.
For example, if g=12, k=3, and u is a node in $\Gamma_r(v)$ with RD-combination 01i ($3 \leq i \leq 5$), then u belongs to a facial circuit which is closed with respect to $B_r(v)$. If i=2, then each facial circuit to which u belongs is not closed with respect to $B_r(v)$. Noting that each facial circuit contains two nodes with RD-combination 01i ($3 \leq i \leq 5$) in $S_{12,3}$, we obtain

$$|V(\phi_r(v))| = |V(B_r(v))| + 1/2 \{ 5.N_{013}(r) + 3.N_{014}(r) + N_{015}(r) \}$$

for $S_{12,3}$. The table starts at the next page. R denotes the radius of the circumscribed ball of $\phi_r(v)$.

g=6, k=3			g=7, k=3			g=8, k=3			g=9, k=3		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	4	1	1	4	1	1	4	1	1	4
2	3	13	2	3	16	2	4	19	2	2	10
3	4	22	3	4	28	3	5	34	3	4	28
4	5	37	4	5	52	4	6	64	4	5	52
5	6	52	5	6	85	5	7	121	5	6	100
6	7	73	6	7	142	6	8	208	6	8	199
7	8	94	7	8	229	7	9	367	7	9	358
8	9	121	8	9	364	8	10	640	8	10	664
9	10	148	9	10	574	9	11	1102	9	11	1225
10	11	181	10	11	898	10	12	1909	10	12	2236

g=10, k=3			g=11, k=3			g=12, k=3			g=13, k=3		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	4	1	1	4	1	1	4	1	1	4
2	2	10	2	2	10	2	2	10	2	2	10
3	5	31	3	5	34	3	6	37	3	3	22
4	6	58	4	6	64	4	7	70	4	6	58
5	7	112	5	7	124	5	8	136	5	7	112
6	8	217	6	8	244	6	9	268	6	8	220
7	9	418	7	9	478	7	10	529	7	9	436
8	10	784	8	10	919	8	11	1042	8	10	862
9	11	1483	9	11	1774	9	12	2020	9	12	1717
10	12	2800	10	12	3424	10	13	3937	10	13	3358

g=4, k=4			g=5, k=4			g=6, k=4			g=7, k=4		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	2	9	1	1	5	1	1	5	1	1	5
2	3	21	2	2	17	2	3	21	2	3	25
3	4	37	3	4	49	3	4	57	3	4	69
4	5	57	4	5	117	4	5	157	4	5	201
5	6	81	5	6	273	5	6	413	5	6	569
6	7	109	6	7	637	6	7	1089	6	7	1613
7	8	141	7	8	1469	7	8	2853	7	8	4561
8	9	177	8	9	3377	8	9	7477	8	9	12877
9	10	217	9	10	7765	9	10	19577	9	10	36361
10	11	261	10	11	17841	10	11	51261	10	11	102653

g=8, k=4			g=9, k=4			g=10, k=4			g=11, k=4		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	5	1	1	5	1	1	5	1	1	5
2	4	29	2	2	17	2	2	17	2	2	17
3	5	81	3	4	61	3	5	65	3	5	69
4	6	237	4	5	177	4	6	189	4	6	201
5	7	697	5	6	525	5	7	561	5	7	597
6	8	2009	6	8	1565	6	8	1669	6	8	1785
7	9	5813	7	9	4601	7	9	4961	7	9	5333
8	10	16809	8	10	13565	8	10	14705	8	10	15909
9	11	48573	9	11	39985	9	11	43613	9	11	47457
10	12	140389	10	12	117829	10	12	129345	10	12	141573

g=12, k=4			g=13, k=4			g=14, k=4			g=15, k=4		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	5	1	1	5	1	1	5	1	1	5
2	2	17	2	2	17	2	2	17	2	2	17
3	6	73	3	3	53	3	3	53	3	3	53
4	7	213	4	6	177	4	7	181	4	7	185
5	8	633	5	7	525	5	8	537	5	8	549
6	9	1893	6	8	1569	6	9	1605	6	9	1641
7	10	5665	7	9	4701	7	10	4809	7	10	4917
8	11	16949	8	10	14081	8	11	14413	8	11	14745
9	12	50645	9	12	42193	9	12	43193	9	12	44213
10	13	151369	10	13	126325	10	13	129437	10	13	132569

g=4, k=5			g=5, k=5			g=6, k=5			g=7, k=5		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	2	11	1	1	6	1	1	6	1	1	6
2	3	36	2	2	26	2	3	31	2	3	36
3	4	101	3	4	101	3	4	116	3	4	136
4	5	271	4	5	356	4	5	441	4	5	536
5	6	716	5	6	1251	5	6	1646	5	6	2091
6	7	1881	6	7	4396	6	7	6151	6	7	8156
7	8	4931	7	8	15416	7	8	22956	7	8	31801
8	9	12916	8	9	54051	8	9	85681	8	9	123956
9	10	33821	9	10	189516	9	10	319766	9	10	483186
10	11	88551	10	11	664461	10	11	1193391	10	11	1883446

g=8, k=5			g=9, k=5			g=10, k=5			g=11, k=5		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	6	1	1	6	1	1	6	1	1	6
2	4	41	2	2	26	2	2	26	2	2	26
3	5	156	3	4	116	3	5	121	3	5	126
4	6	616	4	5	456	4	6	476	4	6	496
5	7	2441	5	6	1816	5	7	1896	5	7	1976
6	8	9606	6	8	7241	6	8	7561	6	8	7896
7	9	37841	7	9	28776	7	9	30146	7	9	31546
8	10	149056	8	10	114416	8	10	120126	8	10	126001
9	11	587076	9	11	454921	9	11	478721	9	11	503256
10	12	2312321	10	12	1808736	10	12	1907776	10	12	2010056

g=12, k=5			g=13, k=5			g=14, k=5			g=15, k=5		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	6	1	1	6	1	1	6	1	1	6
2	2	26	2	2	26	2	2	26	2	2	26
3	6	131	3	3	106	3	3	106	3	3	106
4	7	516	4	6	446	4	7	451	4	7	456
5	8	2056	5	7	1776	5	8	1796	5	8	1816
6	9	8216	6	8	7096	6	9	7176	6	9	7256
7	10	32841	7	9	28376	7	10	28696	7	10	29016
8	11	131266	8	10	113466	8	11	114761	8	11	116056
9	12	524566	9	12	453731	9	12	458946	9	12	464186
10	13	2096331	10	13	1814236	10	13	1835386	10	13	1856586

g=3, k=6			g=4, k=6			g=5, k=6			g=6, k=6		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	7	1	2	13	1	1	7	1	1	7
2	2	19	2	3	55	2	2	37	2	3	43
3	3	37	3	4	211	3	4	181	3	4	205
4	4	61	4	5	793	4	5	835	4	5	991
5	5	91	5	6	2965	5	6	3853	5	6	4747
6	6	127	6	7	11071	6	7	17779	6	7	22753
7	7	169	7	8	41323	7	8	81991	7	8	109015
8	8	217	8	9	154225	8	9	378109	8	9	522331
9	9	271	9	10	575581	9	10	1743691	9	10	2502637
10	10	331	10	11	2148103	10	11	8041177	10	11	11990863

g=7, k=6			g=8, k=6			g=9, k=6			g=10, k=6		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	7	1	1	7	1	1	7	1	1	7
2	3	49	2	4	55	2	2	37	2	2	37
3	4	235	3	5	265	3	4	199	3	5	205
4	5	1165	4	6	1315	4	5	985	4	6	1015
5	6	5749	5	7	6541	5	6	4915	5	7	5065
6	7	28363	6	8	32437	6	8	24535	6	8	25291
7	8	139921	7	9	160915	7	9	122341	7	9	126277
8	9	690199	8	10	798265	8	10	610123	8	10	630397
9	10	3404641	9	11	3959935	9	11	3042721	9	11	3147115
10	11	16794499	10	12	19644031	10	12	15174187	10	12	15711265

g=11, k=6			g=12, k=6			g=13, k=6			g=14, k=6		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	7	1	1	7	1	1	7	1	1	7
2	2	37	2	2	37	2	2	37	2	2	37
3	5	211	3	6	217	3	3	187	3	3	187
4	6	1045	4	7	1075	4	6	961	4	7	967
5	7	5215	5	8	5365	5	7	4795	5	8	4825
6	8	26065	6	9	26815	6	8	23965	6	9	24115
7	9	130267	7	10	134041	7	9	119815	7	10	120565
8	10	651007	8	11	670027	8	10	599017	8	11	602791
9	11	3253357	9	12	3349087	9	12	2994817	9	12	3013777
10	12	16258435	10	13	16740277	10	13	14972527	10	13	15067987

g=3, k=7			g=4, k=7			g=5, k=7			g=6, k=7		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	8	1	2	15	1	1	8	1	1	8
2	2	29	2	3	78	2	2	50	2	3	57
3	3	85	3	4	379	3	4	295	3	4	330
4	4	232	4	5	1821	4	5	1674	4	5	1933
5	5	617	5	6	8730	5	6	9507	5	6	11264
6	6	1625	6	7	41833	6	7	53992	6	7	65661
7	7	4264	7	8	200439	7	8	306566	7	8	382698
8	8	11173	8	9	960366	8	9	1740677	8	9	2230537
9	9	29261	9	10	4601395	9	10	9883546	9	10	13000520
10	10	76616	10	11	22046613	10	11	56118609	10	11	75772593

g=7, k=7			g=8, k=7			g=9, k=7		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	8	1	1	8	1	1	8
2	3	64	2	4	71	2	2	50
3	4	372	3	5	414	3	4	316
4	5	2220	4	6	2472	4	5	1884
5	6	13217	5	7	14785	5	6	11292
6	7	78674	6	8	88292	6	8	67691
7	8	468301	7	9	527339	7	9	405602
8	9	2787436	8	10	3149616	8	10	2430464
9	10	16591534	9	11	18811458	9	11	14563921
10	11	98757002	10	12	112353781	10	12	87270436

g=10, k=7			g=11, k=7			g=12, k=7			g=13, k=7		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	8	1	1	8	1	1	8	1	1	8
2	2	50	2	2	50	2	2	50	2	2	50
3	5	323	3	5	330	3	6	337	3	3	302
4	6	1926	4	6	1968	4	7	2010	4	6	1842
5	7	11544	5	7	11796	5	8	12048	5	7	11040
6	8	69217	6	8	70764	6	9	72276	6	8	66228
7	9	415010	7	9	424502	7	10	433609	7	9	397356
8	10	2488172	8	10	2546475	8	11	2601362	8	10	2384054
9	11	14917799	9	11	15275562	9	12	15606200	9	12	14303857
10	12	89439456	10	12	91633704	10	13	93625477	10	13	85820050

g=3, k=8			g=4, k=8			g=5, k=8		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	9	1	2	17	1	1	9
2	2	41	2	3	105	2	2	65
3	3	161	3	4	617	3	4	449
4	4	609	4	5	3601	4	5	3017
5	5	2281	5	6	20993	5	6	20289
6	6	8521	6	7	122361	6	7	136441
7	7	31809	7	8	713177	7	8	917465
8	8	118721	8	9	4156705	8	9	6169281
9	9	443081	9	10	24227057	9	10	41483913
10	10	1653609	10	11	141205641	10	11	278948961

g=6, k=8			g=7, k=8			g=8, k=8		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	9	1	1	9	1	1	9
2	3	73	2	3	81	2	4	89
3	4	497	3	4	553	3	5	609
4	5	3417	4	5	3857	4	6	4249
5	6	23417	5	6	26865	5	7	29681
6	7	160513	6	7	187097	6	8	207153
7	8	1100169	7	8	1303009	7	9	1445897
8	9	7540681	8	9	9074489	8	10	10092145
9	10	51684593	9	10	63197169	9	11	70441497
10	11	354251481	10	11	440121913	10	12	491670089

g=9, k=8			g=10, k=8			g=11, k=8		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	9	1	1	9	1	1	9
2	2	65	2	2	65	2	2	65
3	4	473	3	5	481	3	5	489
4	5	3297	4	6	3353	4	6	3409
5	6	23065	5	7	23457	5	7	23849
6	8	161369	6	8	164137	6	8	166929
7	9	1128753	7	9	1148513	7	9	1168393
8	10	7895609	8	10	8036289	8	10	8177929
9	11	55229665	9	11	56231033	9	11	57239649
10	12	386330601	10	12	393456385	10	12	400636649

g=12, k=8			g=13, k=8			g=14, k=8		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	9	1	1	9	1	1	9
2	2	65	2	2	65	2	2	65
3	6	497	3	3	457	3	3	457
4	7	3465	4	6	3233	4	7	3241
5	8	24241	5	7	22617	5	8	22673
6	9	169673	6	8	158305	6	9	158697
7	10	1187649	7	9	1108121	7	10	1110865
8	11	8313097	8	10	7756737	8	11	7775993
9	12	58188265	9	12	54296417	9	12	54431505
10	13	407294097	10	13	380069353	10	13	381017401

g=3, k=9			g=4, k=9			g=5, k=9		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	10	1	2	19	1	1	10
2	2	55	2	3	136	2	2	82
3	3	271	3	4	937	3	4	649
4	4	1306	4	5	6427	4	5	5032
5	5	6265	5	6	44056	5	6	39043
6	6	30025	6	7	301969	6	7	302932
7	7	143866	7	8	2069731	7	8	2350324
8	8	689311	8	9	14186152	8	9	18235207
9	9	3302695	9	10	97233337	9	10	141479560
10	10	15824170	10	11	666447211	10	11	1097682301

g=6, k=9			g=7, k=9			g=8, k=9		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	10	1	1	10	1	1	10
2	3	91	2	3	100	2	4	109
3	4	712	3	4	784	3	5	856
4	5	5617	4	5	6256	4	6	6832
5	6	44218	5	6	49879	5	7	54577
6	7	348139	6	7	397648	6	8	435754
7	8	2740888	7	8	3170161	7	9	3479293
8	9	21578977	8	9	25273252	8	10	27780544
9	10	169890922	9	10	201484306	9	11	221814568
10	11	1337548411	10	11	1606280158	10	12	1771085089

g=9, k=9			g=10, k=9			g=11, k=9		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	10	1	1	10	1	1	10
2	2	82	2	2	82	2	2	82
3	4	676	3	5	685	3	5	694
4	5	5392	4	6	5464	4	6	5536
5	6	43120	5	7	43696	5	7	44272
6	8	344845	6	8	349489	6	8	354160
7	9	2757556	7	9	2795266	7	9	2833138
8	10	22051000	8	10	22356730	8	10	22663909
9	11	176332465	9	11	178810813	9	11	181301572
10	12	1410055624	10	12	1430142400	10	12	1450335088

g=12, k=9			g=13, k=9			g=14, k=9		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	10	1	1	10	1	1	10
2	2	82	2	2	82	2	2	82
3	6	703	3	3	658	3	3	658
4	7	5608	4	6	5302	4	7	5311
5	8	44848	5	7	42400	5	8	42472
6	9	358768	6	8	339184	6	9	339760
7	10	2870065	7	9	2713456	7	10	2718064
8	11	22959874	8	10	21707506	8	11	21744433
9	12	183673450	9	12	173658943	9	12	173954818
10	13	1469343439	10	13	1389262240	10	13	1391633362

g=3, k=10			g=4, k=10			g=5, k=10		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	11	1	2	21	1	1	11
2	2	71	2	3	171	2	2	101
3	3	421	3	4	1351	3	4	901
4	4	2461	4	5	10641	4	5	7911
5	5	14351	5	6	83781	5	6	69501
6	6	83651	6	7	659611	6	7	610591
7	7	487561	7	8	5193111	7	8	5364131
8	8	2841721	8	9	40885281	8	9	47124701
9	9	16562771	9	10	321889141	9	10	413997631
10	10	96534911	10	11	2534227851	10	11	3637031721

g=6, k=10			g=7, k=10			g=8, k=10		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	11	1	1	11	1	1	11
2	3	111	2	3	121	2	4	131
3	4	981	3	4	1071	3	5	1161
4	5	8731	4	5	9621	4	6	10431
5	6	77591	5	6	86381	5	7	93781
6	7	689601	6	7	775511	6	8	842861
7	8	6128811	7	8	6962401	7	9	7575431
8	9	54469711	8	9	62507011	8	10	68086161
9	10	484098581	9	10	561175321	9	11	611941851
10	11	4302417531	10	11	5038118591	10	12	5499984691

g=9, k=10			g=10, k=10			g=11, k=10		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	11	1	1	11	1	1	11
2	2	101	2	2	101	2	2	101
3	4	931	3	5	941	3	5	951
4	5	8361	4	6	8451	4	6	8541
5	6	75231	5	7	76041	5	7	76851
6	8	676931	6	8	684271	6	8	691641
7	9	6090701	7	9	6157541	7	9	6224591
8	10	54801431	8	10	55409501	8	10	56019651
9	11	493079041	9	11	498610391	9	11	504161661
10	12	4436507071	10	12	4486817601	10	12	4537318311

g=12, k=10			g=13, k=10			g=14, k=10		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	11	1	1	11	1	1	11
2	2	101	2	2	101	2	2	101
3	6	961	3	3	911	3	3	911
4	7	8631	4	6	8241	4	7	8251
5	8	77661	5	7	74151	5	8	74241
6	9	698931	6	8	667341	6	9	668151
7	10	6290281	7	9	6006051	7	10	6013341
8	11	56611631	8	10	54054281	8	11	54119971
9	12	509496131	9	12	486486961	9	12	487078841
10	13	4585388461	10	13	4378367971	10	13	4383701471

g=3, k=11			g=4, k=11			g=5, k=11		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	12	1	2	23	1	1	12
2	2	89	2	3	210	2	2	122
3	3	617	3	4	1871	3	4	1211
4	4	4236	4	5	16633	4	5	11870
5	5	29041	5	6	147830	5	6	116403
6	6	199057	6	7	1313841	6	7	1141504
7	7	1364364	7	8	11676743	7	8	11193986
8	8	9351497	8	9	103776850	8	9	109772169
9	9	64096121	9	10	922314911	9	10	1076464566
10	10	439321356	10	11	8197057353	10	11	10556190657

g=6, k=11			g=7, k=11			g=8, k=11		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	12	1	1	12	1	1	12
2	3	133	2	3	144	2	4	155
3	4	1310	3	4	1420	3	5	1530
4	5	12981	4	5	14180	4	6	15280
5	6	128492	5	6	141549	5	7	152681
6	7	1271953	6	7	1412918	6	8	1525272
7	8	12591030	7	8	14103541	7	9	15237575
8	9	124638361	8	9	140779244	8	10	152224480
9	10	1233792572	9	10	1405235646	9	11	1520733270
10	11	12213287373	10	11	14026834834	10	12	15192232661

g=9, k=11			g=10, k=11			g=11, k=11		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	12	1	1	12	1	1	12
2	2	122	2	2	122	2	2	122
3	4	1244	3	5	1255	3	5	1266
4	5	12420	4	6	12530	4	6	12640
5	6	124180	5	7	125280	5	7	126380
6	8	1241615	6	8	1252681	6	8	1263780
7	9	12413886	7	9	12525602	7	9	12637582
8	10	124116488	8	10	125243592	8	10	126373567
9	11	1240941241	9	11	1252311875	9	11	1263712902
10	12	12407176332	10	12	12521878480	10	12	12636901520

g=12, k=11			g=13, k=11			g=14, k=11		
r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$	r	R	$\lvert V(\phi_r(v))\rvert$
1	1	12	1	1	12	1	1	12
2	2	122	2	2	122	2	2	122
3	6	1277	3	3	1222	3	3	1222
4	7	12750	4	6	12266	4	7	12277
5	8	127480	5	7	122640	5	8	122750
6	9	1274780	6	8	1226380	6	9	1227480
7	10	12747681	7	9	12263780	7	10	12274780
8	11	127475602	8	10	122637582	8	11	122747681
9	12	1274743372	9	12	1226373677	9	12	1227475602
10	13	12747307497	10	13	12263714662	10	13	12274743922

References

[AgJaPa] D.P. Agrawal, V.K. Janakiram, G.C. Pathak,
"Evaluating the Performance of Multicomputer Configurations", IEEE Computer, Vol. 19, No. 5, May 1986, pp. 23-37.

[Akl] S.G. Akl,
"Parallel Sorting Algorithms", London: Academic Press, 1985.

[Aupper] E.M. Aupperle,
"MERIT Computer Network: Hardware Considerations", in: R. Rustin (ed.), Computer Networks, New Jersey: Prentice-Hall, Englewood Cliffs, 1972, pp. 49-63.

[BaBrKa] G.H. Barnes, R.M. Brown, M. Kato, D.J. Kuck, D.L. Slotnick, R.A. Stokes,
"The ILLIAC IV Computer", IEEE Transactions on Computers, Vol. C-17, No. 8, August 1968, pp. 746-757.

[Batche] K.E. Batcher,
"The Flip Network in STARAN", Proceedings of the 1976 International Conference on Parallel Processing, August 1976, pp. 65-71.

[BeBoPa] J.C. Bermond, J. Bond, M. Paoli, C. Peyrat,
"Graphs and Interconnection Networks: Diameter and Vulnerability", in: E.K. Lloyd (ed.), Surveys in Combinatorics: Invited Papers for the 9th British Combinatorial Conference, 1983, London Mathematical Society Lecture Notes, Vol. 82, Cambridge: University Press, pp. 1-30.

[BeDeWe] J. Beetem, M. Denneau, D. Weingarten,
"The GF11 Parallel Computer", in: J.J. Dongarra (ed.), Experimental Parallel Computer Architectures, Amsterdam: North-Holland,

1987.

[BeDeQu] J.-C. Bermond, C. Delorme, J.-J. Quisquater,
"Strategies for Interconnection Networks: Some Methods from Graph Theory", Journal of Parallel and Distributed Computing, Vol. 3, No. 4, December 1986, pp. 433-449.

[Benes] V. Benes,
"Mathematical Theory of Connecting Networks", New York: Academic Press, 1965.

[Berge] C. Berge,
"Graphs and Hypergraphs", Amsterdam: North-Holland Publishing Company, 1973.

[Biggs] N. Biggs,
"Algebraic Graph Theory", London: Cambridge University Press, 1974.

[Bokhar] S.H. Bokhari,
"On the Mapping Problem", IEEE Transactions on Computers, Vol. C-30, No. 3, March 1981, pp. 207-214.

[Catte1] K.W. Cattermole,
"Class of communication networks with optimal connectivity", Electronic Letters, Vol. 8, No. 12, June 1972, pp. 316-319.

[Catte2] K.W. Cattermole,
"Bipartite communication networks with optimal connectivity", Electronic Letters, Vol. 8, No. 15, July 1972, pp. 385-388.

[DeFrSm] L. Dekker, E.E.E. Frietman, W. Smit, J.C. Zuidervaart,
"Optical Link in the Delft Parallel Processor - an Example of MOMI-Connection in MIMD-Supercomputers", Future Generations Computer Systems, Vol. 4, 1988, pp. 189-203.

[DesPat] A.M. Despain, D.A. Patterson,
"X-Tree: A Tree Structured Multiprocessor Computer Architecture", Proceedings of the 5th Annual Symposium on Computer Architecture, IEEE, August 1978, pp. 144-151.

[DoRaSl] S.J. Dow, D.F. Rall, P.J. Slater,
"Convex Labelings of Trees", Journal of Graph Theory, Vol. 11, No. 1, Spring 1987, pp. 59-70.

[FarJam] M. Farber, R.E. Jamison,
"On local convexity in graphs", Discrete Mathematics, Vol. 66, No. 3, September 1987, pp. 231-247.

[Feng] T-Y. Feng,
"A Survey of Interconnection Networks", IEEE Computer, Vol. 14, No. 12, December 1981, pp. 12-27.

[FleImr] H. Fleischner, W. Imrich,
"Transitive Planar Graphs", Mathematica Slovaca, Vol. 29, No. 2, 1979, pp. 97-105.

[Fisher] D.C. Fisher,
"Your Favorite Parallel Algorithms Might Not Be as Fast as You Think", IEEE Transactions on Computers, Vol. 37, No. 2, February 1988, pp. 211-213.

[FisKun] A.L. Fisher, H.T. Kung,
"Synchronizing Large VLSI Processor Arrays", IEEE Transactions on Computers, Vol. C-34, No. 8, August 1985, pp. 734-740.

[FrHeHe] H. Fromm, U. Hercksen, U. Herzog, K.H. John, R. Klar, W. Kleinöder,
"Experiences with Performance Measurement and Modeling of a Processor Array", IEEE Transactions on Computers, Vol. C-32, No. 1, January 1983, pp. 15-31.

[Furste] H. Furstenberg,
"Recurrence in Ergodic Theory and Combinatorial Number Theory", Princeton, New Jersey: Princeton University Press, 1981.

[Goldsc] L.M. Goldschlager,
"A universal interconnection pattern for parallel computers", Journal of the ACM, Vol. 29, No. 3, July 1982, pp. 1073-1086.

[GooSeq] J.R. Goodman, C.H. Séquin,
"Hypertree: A Multiprocessor Interconnection Topology", IEEE Transactions on Computers, Vol. C-30, No. 12, December 1981, pp. 923-933.

[GoGrKr] A. Gottlieb, R. Grishman, C.P. Kruskal, K.P. McAuliffe, L. Rudolph, M. Snir,
"The NYU Ultracomputer - Designing an MIMD Shared Memory Parallel Computer", 9th Annual International Conference on Computer Architecture, Austin, 1982, pp. 27-42, see also [LipMal;

Appendix B].

[GruShe] B. Grünbaum, G.C. Shephard,
"Tilings and Patterns", New York: Freeman and Company, 1987.

[Harary] F. Harary,
"Graph Theory", Addison-Wesley, 1969.

[HarNie] F. Harary, J. Nieminen,
"Convexity in Graphs", Journal of Differential Geometry, Vol. 16, 1981, pp. 185-190.

[HarUll] A.C. Hartmann, J.D. Ullman,
"Model Categories For Theories of Parallel Systems", in: Microelectronics and Computer Corporation Technical Report PP-341-86, see also [LipMal; Appendix H].

[Hilber] P.A.J. Hilbers,
"Mappings of Algorithms on Processor Networks", University of Groningen, January 1989.

[Hillis] W.D. Hillis,
"The Connection Machine", Cambridge: The MIT Press, 1985.

[HocJes] R.W. Hockney, C.R. Jesshope,
"Parallel Computers", Bristol: Adam Hilger Ltd, 1981.

[HoKuRe] S.H. Hosseini, J.G. Kuhl, S.M. Reddy,
"Distributed Fault-Tolerance of Tree Structures", IEEE Transactions on Computers, Vol. C-36, No. 11, November 1987, pp. 1378-1382.

[HorZor] E. Horowitz, A. Zorat,
"The Binary Tree as an Interconnection Network: Applications to Multiprocessor Systems and VLSI", IEEE Transactions on Computers, Vol. C-30, No. 4, April 1981, pp. 247-253.

[HwaBri] K. Hwang, F.A. Briggs,
"Computer Architecture and Parallel Processing", New York: McGraw-Hill, 1985.

[HwaGho] K. Hwang, J. Ghosh,
"Hypernet: A Communication-Efficient Architecture for Constructing Massively Parallel Computers", IEEE Transactions on Computers, Vol. C-36, No. 12, December 1987, pp. 1450-1466.

[Kowali] J.S. Kowalik,
"Parallel MIMD Computation: HEP Supercomputers and Its

Applications", Cambridge: The MIT Press, 1985.

[Kung] H.T. Kung,
"The Structure of Parallel Algorithms", in: M.C. Yovits (ed.), Advances in Computers, Vol. 19, New York: Academic Press, 1980. pp. 65-112.

[LanSto] T. Lang, H.S. Stone,
"A Shuffle-Exchange Network with Simplified Control", IEEE Transactions on Computers, Vol. C-25, No. 6, January 1976, pp. 55-65.

[Lawrie] D.H. Lawrie,
"Access and Alignment of Data in an Array Processor", IEEE Transactions on Computers, Vol. C-24, No. 12, December 1975, pp. 1145-1155.

[Leight] F.T. Leighton,
"New Lower Bound Techniques for VLSI", Proceedings of the 22nd Annual IEEE Symposium on Foundations of Computer Science, Nashville, Tennessee, October 1981, pp. 1-12.

[Leiser] C.E. Leiserson,
"Area-Efficient VLSI Computation", Dissertation, Cambridge: The MIT Press, 1982.

[Levita] S.P. Levitan,
"Measuring Communication Structures in Parallel Architectures and Algorithms", in: The Characteristics of Parallel Algorithms, L.H. Jamieson, D.B. Gannon, R.J. Douglas (eds.), Cambridge: The MIT Press, 1987, pp. 101-137.

[LipMal] G.J. Lipovski, M. Malek,
"Parallel Computing, Theory and Comparisons", New York: John Wiley & Sons, 1987.

[Macphe] H.D. Macpherson,
"Infinite distance transitive graphs of finite valency", Combinatorica, Vol. 2, No. 1, 1982, pp. 63-69.

[Mader] W. Mader,
"Uber den Zusammenhang symmetrischer Graphen", Archiv der Mathematik, Vol. 21, 1970, pp. 331-336.

[Mazumd] P. Mazumder,
"Evaluation of On-Chip Static Interconnection Networks", IEEE Transactions on Computers, Vol. C-36, No. 3, March 1987, pp. 365-

369.

[Parber] I. Parberry,
"On Recurrent and Recursive Interconnection Patterns", Information Processing Letters, Vol. 22, No. 6, 1986, pp. 285-289.

[Perron] O. Perron,
"Algebra II, Theorie der algebraischen Gleichungen", Berlin: Walter de Gruyter & Co., 1933.

[Potter] J.L. Potter,
"The Massively Parallel Processor", Cambridge: The MIT Press, 1985.

[PreVui] F. Preparata, J. Vuillemin,
"The Cube-Connected Cycles: A versatile network for parallel computation", CACM, Vol. 24, No. 5, May 1981, pp. 300-309.

[Queyss] D. Queyssac,
"Projecting VLSI's impact on microprocessors", IEEE Spectrum, Vol. 16, May 1979, pp. 38-41.

[Quinn] M.J. Quinn,
"Designing Efficient Algorithms for Parallel Computers", New York: McGraw-Hill, 1987.

[Sadibu] G. Sadibussi,
"Graph Multiplication", Mathematische Zeitschrift, Vol. 72, 1960, pp. 446-457.

[Seitz] C.L. Seitz,
"Concurrent VLSI Architectures", IEEE Transactions on Computers, Vol. C-33, No. 12, December 1984, pp. 1247-1265.

[Shaw] D.E. Shaw,
"Organization and Operation of a Massively Parallel Machine", in: G. Rabbat (ed.), Advanced Semiconductor Technology and Computer Systems, Van Nostrand Reinhold Company, 1987, see also [LipMal; Appendix G].

[ShiVil] Y. Shiloach, U. Vishkin,
"Finding the maximum, merging and sorting in a parallel computation model", Journal of Algorithms, Vol. 2, No. 1, March 1981, pp. 88-102.

[ShiVi2] Y. Shiloach, U. Vishkin,
"An O(log n) parallel connectivity algorithm", Journal of Algorithms, Vol. 3, No. 1, March 1982, pp. 57-67.

[Snyder] L. Snyder,
"Introduction to the Configurable Highly Parallel Computer", IEEE Computer, Vol. 15, No. 1, January 1982, pp. 47-56, see also [LipMal; Appendix F].

[SolChe] V.P. Soltan, V.D. Chepoi,
"Conditions for invariance of set diameters under d-convexification in a graph", Cybernetics, Vol. 19, No. 6, November-December 1983, pp. 750-756.

[Stone1] H.S. Stone,
"Parallel Processing with the Perfect Shuffle", IEEE Transactions on Computers, Vol. C-20, No. 2, February 1971, pp. 153-161.

[Stone2] H.S. Stone,
"Multiprocessor Scheduling with the Aid of Network Flow Algorithms", IEEE Transactions on Software Engineering, Vol. SE-3, No. 1, January 1977, pp. 85-93.

[TilWit] A.M. van Tilborg, L.D. Wittie,
"Wave Scheduling - Decentralized Scheduling of Task Forces in Multicomputers", IEEE Transactions on Computers, Vol. C-33, No. 9, September 1984, pp. 835-844.

[Uhr] L. Uhr,
"Multi-Computer Architectures for Artificial Intelligence, Toward Fast, Robust, Parallel Systems", New York: John Wiley & Sons, 1987.

[Ullman] J.D. Ullman,
"Computational Aspects of VLSI", Maryland: Computer Science Press, 1984.

[Vitany] P.M.B. Vitányi,
"Nonsequential Computation and Laws of Nature", in: F. Makedon, K. Mehlhorn, T. Papatheodorou, P. Spirakis (eds.), Lecture Notes in Computer Science 227, VLSI Algorithms and Architectures, Aegean Workshop on Computing, Loutraki, Greece, July 1986, pp. 108-120.

[WanFra] D.F. Wann, M.A. Franklin,
"Asynchronous and Clocked Control Structures for VLSI Based

Interconnection Networks", IEEE Transactions on Computers, Vol. C-32, No. 3, March 1983, pp. 284-293.

[Watkin] M.E. Watkins,
"Connectivity of Transitive Graphs", Journal of Combinatorial Theory, Vol. 8, No. 1, January 1970, pp. 23-29.

[Wilson] R.J. Wilson,
"Introduction to Graph Theory", New York: Longman, 1985.

[WuFeng] C. Wu, T. Feng,
"On a Class of Multistage Interconnection Networks", IEEE Transactions on Computers, Vol. C-29, No. 8, August 1980, pp. 694-702.

God was satisfied with his own work,
and that is fatal.

Samuel Butler (1912)

Summary

In this dissertation extensible massively parallel computers are discussed. These are computers which consist of many processors, and which can be extended with additional processors without changing their underlying structure. Such computers are in particular suited for computation intensive applications. In addition, their computation capacity can be tailored to the requirements of a user without the need to adapt software.
The mechanism that provides for communication between processors, an important issue in parallel computers, calls the tune in this dissertation. A number of fundamental problems of efficient communication models for extensible massively parallel computers are elucidated, and a few suggestions are made to the solutions of them. Furthermore, statical communication networks, which are often applied as communication mechanism in parallel computers, are studied. It is explained in which way extensible statical communication networks with suitable properties can be constructed.

Besides a general treatment of the three main themes of this dissertation, massive parallelism, communication between processors, and extensibility, in chapter 1 a number of demands to extensible massively parallel computers are formulated.

In chapter 2 a method is described to construct statical communication networks for extensible parallel computers. A network constructed by this method has suitable properties, such as

1. the absence of a bound to the number of times a network can be extended,
2. invariability of the complexity of the processors under extension (fixed degree),
3. the ability to low communication times (low diameter),

4. vulnerability to the dropout of many processors or connections between processors (high connectivity),
5. a regular structure which is maintained under extension,
6. maintenance of the routing function under extension,
7. extensibility by a minimal number of processors while the above characteristics are maintained.

The construction method consists of two stages:

- Construction of an infinite graph,
- Cutting the extensible networks aimed at out of the infinite graph.

A description is given of the properties the infinite graph should have in order to be a suitable basis for extensible networks. Besides it is explained which shapes of cutouts result in efficient extensible networks. In this chapter a measure is introduced which describes the node density of an infinite graph, the so-called exponentiality of the graph. Also a measure is introduced, which gives an indication about the efficiency of a cutout of the infinite graph.

In chapter 3 a fundamental problem of communication mechanisms is considered. It concerns the space deficiencies that arise when processors of fixed sizes, and connection wires of a bounded length are used in an extensible parallel computer exhibiting low communication times. The only solution to these deficiencies maintaining low communication times, is the use of processors with ever-decreasing sizes for subsequent extensions of a computer. The consequences, a superlinear increase of the computation capacity under extension, and a limitation to the pace of extension of a computer by the pace at which technology develops, are considered in a quantitative way.

In chapter 4 the method in chapter 2 is used to construct an infinite class of extensible networks, which are based on the Cartesian graph product. The well-known meshes and trees are subgraphs of these networks, suggesting that the networks are suited for implementation of many standard algorithms, such as sorting, matrix multiplication, etc.
In chapter 5 a particular aspect of these networks is considered, the so-called local connectivities between the nodes.

Thereupon, using the method in chapter 2 a second infinite class of extensible networks is constructed. These networks are planar, have a low nearly optimal diameter, and have a very regular structure.
Chapter 6 describes the infinite graphs which are at the base of these networks, and deduces several properties of the infinite graphs. These properties are used

in chapter 7 to determine the exponentialities of the infinite graphs.
Finally, in chapter 8 efficient cutouts of the infinite graphs are constructed, resulting in the planar networks.

Samenvatting

In dit proefschrift worden uitbreidbare grootschalig parallelle computers besproken. Dit zijn computers die uit veel processoren bestaan en uitgebreid kunnen worden met extra processoren zonder dat hun onderliggende structuur verandert. Deze computers zijn met name geschikt voor rekenintensieve toepassingen. Bovendien kan hun rekencapaciteit op de wensen van een gebruiker toegesneden worden zonder dat software aangepast behoeft te worden.
Het mechanisme dat communicatie tussen de processoren verzorgt, een belangrijk aspect bij parallelle computers, speelt een hoofdrol in dit proefschrift. Er worden een aantal fundamentele problemen belicht van efficiente communicatie-modellen voor uitbreidbare grootschalig parallelle computers, en enkele suggesties gedaan voor de oplossingen daarvan. Verder worden statische communicatie-netwerken, vaak toegepast als communicatie-mechanisme in parallelle computers, bestudeerd. Er wordt uiteengezet hoe uitbreidbare statische communicatie-netwerken met geschikte eigenschappen geconstrueerd kunnen worden.

In hoofdstuk 1 wordt naast een algemene behandeling van de drie hoofdthema's in dit proefschrift, grootschalig parallellisme, communicatie tussen processoren, en uitbreidbaarheid, een aantal eisen geformuleerd ten aanzien van de eigenschappen van uitbreidbare grootschalig parallelle computers.

In hoofdstuk 2 wordt een methode beschreven om statische communicatie-netwerken voor uitbreidbare parallelle computers te construeren. Een netwerk geconstrueerd met deze methode heeft gunstige eigenschappen, zoals

1. het afwezig zijn van een grens aan het aantal malen dat het netwerk uitgebreid kan worden,
2. onveranderlijkheid van de complexiteit van de processoren bij uitbreiding (vaste graad),

3. de mogelijkheid tot lage communicatie-tijden (lage diameter),
4. het bestand zijn tegen uitvallen van veel processoren of verbindingen tussen processoren (hoge connectiviteit),
5. een regelmatige structuur die behouden blijft onder uitbreiding,
6. behoud van de routeringsfunctie onder uitbreiding, en
7. uitbreidbaarheid met een minimaal aantal processoren onder behoud van bovenstaande eigenschappen.

De constructie-methode bestaat uit twee fasen:

- Het construeren van een oneindige graaf,
- Het uitsnijden van de beoogde netwerken uit deze graaf.

Er wordt beschreven welke eigenschappen de oneindige graaf moet hebben om als basis te dienen voor uitbreidbare netwerken. Tevens wordt uiteengezet welke vorm een uitsnijding moet hebben om efficient te zijn als uitbreidbaar netwerk. In dit hoofdstuk wordt een maat geintroduceerd die de knoop-dichtheid van de oneindige graaf beschrijft, de zogenaamde exponentialiteit van de graaf. Tevens wordt een maat geintroduceerd die weergeeft hoe efficient een uitsnijding uit de oneindige graaf is.

In hoofdstuk 3 wordt een fundamenteel probleem van communicatie-mechanismen beschouwd. Het betreft het tekort aan ruimte dat optreedt indien processoren met een vaste grootte, en verbindingsdraden met een begrensde lengte worden gebruikt in een uitbreidbare parallelle computer waarin de communicatietijden laag zijn. De enige oplossing voor dit tekort die lage communicatie-tijden in stand houdt is het gebruik van steeds kleiner wordende processoren voor opeenvolgende uitbreidingen van een computer. De consequenties, een meer dan evenredige toename van de rekencapaciteit bij uitbreiding, en een begrenzing aan het tempo van uitbreiding van een computer door het tempo waarmee de technologie zich ontwikkelt, worden kwantitatief beschouwd.

In hoofdstuk 4 wordt met de methode uit hoofdstuk 2 een oneindige klasse uitbreidbare netwerken geconstrueerd, die gebaseerd zijn op het Cartesisch graafproduct. De welbekende meshes en bomen zijn subgrafen van deze netwerken, wat suggereert dat de netwerken geschikt zijn voor implementatie van veel standaardalgoritmen, zoals sorteren, matrix-vermenigvuldigen, enzovoort.
In hoofdstuk 5 wordt een bepaald aspect van deze netwerken beschouwd, de zogenaamde lokale connectiviteiten tussen de knopen.

Met de methode uit hoofdstuk 2 wordt vervolgens een tweede oneindige klasse

uitbreidbare netwerken geconstrueerd. Deze netwerken zijn planair, hebben een lage bijna optimale diameter, en hebben een zeer regelmatige structuur.
Hoofdstuk 6 beschrijft de oneindige grafen die ten grondslag liggen aan deze netwerken en leidt een aantal eigenschappen van de oneindige grafen af.
Deze eigenschappen worden gebruikt in hoofdstuk 7 om de exponentialiteiten van de oneindige grafen te bepalen.
Tenslotte worden in hoofdstuk 8 efficiente uitsnijdingen uit de oneindige grafen geconstrueerd, wat resulteert in de planaire netwerken.

Curriculum vitae

De schrijver van dit proefschrift werd op 8 juli 1961 geboren in Hengelo(O). In juni 1979 behaalde hij het V.W.O.-diploma aan de Rijksscholengemeenschap te Oud-Beijerland.

In datzelfde jaar begon hij met de studie Wiskunde aan de Technische Universiteit Delft. In december 1983 behaalde hij het kandidaatsexamen Wiskunde met Informatica als hoofdvak, en in november 1985 behaalde hij cum laude het doctoraalexamen Wiskunde, afstudeerrichting Informatica.

Van januari 1984 to juli 1985 was hij werkzaam als student-assistent bij de vakgroep Toegepaste Taalkunde van de Technische Universiteit Delft. Vanaf oktober 1985 verrichtte hij promotie-onderzoek op het gebied van gedistribueerde gegevensverwerking bij de sectie Theoretische Informatica van de Technische Universiteit Delft.

Index